FPGA Generic Library Guide

Summary

This guide contains the naming conventions, detailed description and truth table of all components in the FPGA Generic Library.

Core Reference
CR0118 (v2.202) June 6, 2005

Introduction

FPGA Generic library covers a wide range of commonly used digital components to aid the process of building your system-on-FPGA. This library guide describes the components available in the FPGA Generic integrated library.

Components in this library maintain the policy of FPGA Vendor Independency. This means that you can easily port your design across different platform/architecture FPGA.

The description, functional table and additional information together with their symbolic representation are presented to help you select the correct function to suite your design needs.

Selection Guidelines

The Generic Library components are named following the convention described in Naming Conventions section of this guide.

In the Functional Classes section of this guide each component is listed under their functional category with a short description of their logic behavior.

The Design Components section of this guide lists the components in alphanumeric order with following information on each component:

- Functional Description
- Schematic Symbol
- Truth Table or equation
- Additional notes (if any)

Schematic Symbols

Schematic symbol representation of logic components are shown as they exist in the integrated library. In case where components have large bit size, smaller versions are used to represent their symbolic form.

Naming Conventions

This section contains the naming conventions used to name the components found in the FPGA Generic integrated library. The naming conventions are available for the following functional classes:

- Arithmetic Function
- Buffer
- Bus Joiner
- Clock Divider
- Clock Manager
- Comparator
- Counter
- Decoder
- Encoder
- Flip-Flop
- JTAG
- Latch
- Logic Primitive
- Memory
- Multiplexer
- Numeric Connector
- Shift Register
- Shifter
- Wired Function

Literal Syntax

The naming convention syntax uses the following combinatorial typeface naming conventions.

<*object*>	*object* is compulsory
[*object*]	*object* is optional
object | {*object*}	*object* or combination of *object*s permitted
(*object*)	*object* is literally omitted

Arithmetic Function

The Arithmetic Function naming convention is defined as follows.

<Type>[Registered][Bit-Size][Version]

Type

ACC - Accumulator, Loadable and Cascadable, with Signed and Unsigned Binary operations
ADD - Full Adder, with Signed and Unsigned Binary operations
ADDF - Full Adder, Unsigned
ADSU - Full Adder/Subtracter, with Signed and Unsigned Binary operations
MULT - Multiplier, Signed
MULTU - Multiplier, Unsigned
PAR - Odd/Even Parity Generators/Checker

Registered

R - Registered, ie. Synchronous function
available for ADD, ADDF, ADSU, MULT, MULTU

Bit-Size

1, 2, 4, 8, 16, 32 - for ACC, ADD, ADDF, ADSU
1, 2, 4, 8, 16, 18, 32 - for MULT, MULTU
9 - for PAR

Version

S - Single pin version
B - Bus pin version

Buffer

The Buffers naming convention is defined as follows.

<Type>[Bit-Size][Version]

Type

BUF - Normal Non-inverted Buffer

BUFE - 3-state Output Buffer with Active High Enable

BUFT - 3-state Output Buffer with Active Low Enable

IOBUF - Input/Output Buffer with common control T

IOBUFC - Input/Output Buffer with separated control Ts for each inputs

Bit-Size

(1), 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, 32

Version

S - Single pin version

B - Bus pin version

Bus Joiner

Two conventions are utilized to name the Bus Joiners. **JB** describes the *System Bus Joiner* and the following syntax describes the remaining Bus Joiners:

J<Bit><Port>[Bus-Num]_<Bit><Port>[Bus-Num][Pin-Type]

Bit

2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, 32

Port

S - Single pin

B - Bus

Bus-Num

(1), 2, 4, 8

Pin-Type

X - INOUT

Clock Divider

The Clock Divider naming convention is defined as follows.

CDIV[Num][Duty Cycle]

Num

2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, 20, 24, 32, 64, 128, 256

Programmable versions have the following prefixes:

N_8 - 8-Bit Programmable

N_16 - 16-Bit Programmable

N_32 - 32-Bit Programmable

Duty Circle

DC50 - Duty Cycle of 50%

Clock Manager

The Clock Manager namming convention is defined as follows.

CLKMAN_<Num>

Num

number of operational output ports

Comparator

The Comparator naming convention is defined as follows.

<Type><Bit-Size>[Version]

Type

COMP - Identity Comparator

COMPM - Magnitude Comparator

Bit-Size

2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, 32

Version

S - Single pin version

B - Bus pin version

Counter

The Counter naming convention is defined as follows.

C<Type><Bit-Size><Function|{Function}>[Direction][Version]

Type

B - Cascadable Binary Counter

D - Cascadable Binary-Coded-Decimal (BCD) Counter

J - Johnson Counter

R - Negative-Edge Binary Ripple Counter

Bit-Size

2, 4, 8, 16, 32 - for type B, R

4 - for type D

2, 4, 5, 8, 16, 32 - for type J

Function

C - Asynchronous Clear

R - Synchronous Reset

L - Loadable (Synchronous Load)

E - Clock Enable

Direction

D - Bidirectional (Up/Down)

Version

S - Single pin version

B - Bus pin version

Decoder

Various functional types of Decoders are available to accommodate design needs. The naming convention is defined as follows.

D<Type>[Function][Version]

Type

4_10 - Binary-Coded-Decimal (BCD) Decoder

7SEG - 7-Segment-Display Decoder for Common-Cathode LED (Active High Output)

7SEGN - 7-Segment-Display Decoder for Common-Anode LED (Active Low Output)

n_m - Binary *n*-bit to *m*-bit Decoder,
available in 2_4, 3_8, 4_16, 5_32

Function

E - With Enable. (for 4_10, *n_m* only)

Version

S - Single pin version

B - Bus pin version

Encoder

The Encoder naming convention is defined as follows.

E<Type>[Version]

Type

10_4 - Binary-Coded-Decimal (BCD) Encoder

n_m - *n*-bit to *m*-bit Priority Encoder,
available in 4_2, 8_3, 10_4, 16_4, 32_5

Function

E - With Enable

Version

S - Single pin version

B - Bus pin version

Flip-Flop

The Flip-Flop naming convention is defined as follows.

F<Type>[Bit-Size][Function|{Function}][State][Version]

Type

D - D Flip-Flop
JK - JK Flip-Flop
T - Toggle Flip-Flop

Bit-Size

(1), 2, 4, 8, 16, 32 - for type D
(1) - for other types

Function

C - Asynchronous Clear
R - Synchronous Reset (i.e. Synchronous Clear)
P - Asynchronous Preset
S - Synchronous Set (i.e. Synchronous Preset)
E - Clock Enable

State

_1 - Negative Clock Edge Triggered
N - With Non-inverted and Inverted Outputs

Version

S - Single pin version
B - Bus pin version

JTAG

The JTAG naming convention is defined as follows.

<Type>

Type

NEXUS_JTAG_PORT - Soft Nexus-Chain Connector

Latch

The Latch naming convention is defined as follows.

LD[Bit-Size][Function|{Function}][State][Version]

Bit-Size

(1), 2, 3, 4, 8, 16, 32

Function

C - Clear

P - Preset

E - Gate Enable

State

_1 - Inverted Gate

Version

S - Single pin version

B - Bus pin version

Logic Primitive

The Logic Primitive naming convention is defined as follows.

<Type><Bit-Size>[Function][Version]

Type

AND - AND Gate
NAND - NAND Gate
OR - OR Gate
NOR - NOR Gate
XNOR - Exclusive-NOR Gate
XOR - Exclusive-OR Gate
INV - Inverter
TCZO - True/Complement, Zero/One Element
SOP - Sum of Products

Bit-Size

2, 3, 4, 5, 6, 7, 8, 9, 12, 16, 32 - for AND, NAND, OR, NOR, XNOR, XOR

(1), 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, 32 - for INV

m_n - Applicable to SOP only. Indicates *m* number of *n*-input AND gates in the Sum of Products combination, available in 2_2, 2_3, 2_4, 4_2

Function

(Applicable to AND, NAND, OR, NOR, XNOR and XOR only)

N*m* - *m* inverted inputs
(applicable to `Bit-Size` 2, 3, 4, 5, where *m* is less than or equal to `Bit-Size`)

D - Dual Output (applicable for AND and OR gates with a `Bit-Size` of 2, 3, 4)

Version

S - Single pin version

B - Bus pin version

Memory

The Memory component naming convention is defined as follows.

<Type><Port Type>[Function|{Function}]

Type

RAM - Random Access Memory

ROM - Read Only Memory

Port Type

S - Single Port

D - Dual Port

Function

E - With Enable

R - With Reset

B - Byte Addressable

Multiplexer

The Multiplexer naming convention is defined as follows.

M<Data Width>_<Type>[Function][Select]

Data Width

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, 32

Type

B*n*B1 - *n*-to-1 Multiplexer; *n* number of buses switch to 1 bus, bus size is defined by `Data Width`.

S*n*S1 - *n*-to-1 Multiplexer; *n* groups of single bit pins switch to 1 group, number of pins in a group is defined by `Data Width`.

B*n*S1 - *n*-to-1 Multiplexer; an *n*-bit bus switches to 1-bit single pin, apply for `Data Width` = 1.

B1B*n* - 1-to-*n* DeMultiplexer; 1 bus switch to *n* number of busses, bus size is defined by `Data Width`.

S1S*n* - 1-to-*n* DeMultiplexer; 1 group of single bit pins switch to *n* group, number of pins in a group is defined by `Data Width`.

S1B*n* - 1-to-*n* DeMultiplexer; 1-bit single pin switches to an *n*-bit bus, apply for `Data Width` = 1.

**n* is available in 2, 4, 8, 16

Function

E - With Enable

Select

_SB - With Bus Version Select

Numeric Connector

These components are available for binary logic connections. The naming convention is as follows.

NUM<Hex Value>

Hex Value

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Shift Register

The Shift Register naming convention is defined as follows.

SR<Bit-Size><Function|{Function}>[Direction][Version]

Bit-Size

4, 8, 16, 32

Function

C - Asynchronous Clear

R - Synchronous Reset (i.e. Synchronous Clear)

L - Loadable (Synchronous Load, ie. Parallel In)

E - Clock Enable

Direction

D - Bidirectional (with left or right shift option)

Version

S - Single pin version

B - Bus pin version

Shifter

The Shifter naming convention is defined as follows.

BRLSHFT<Bit-Size><Function>[Version]

Bit-Size

4, 8, 16, 32

Function

M - Fill Mode and direction control

Version

S - Single pin version

B - Bus pin version

Wired Function

The Wired Function naming convention is defined as follows.

<Type>[Bit-Size]<Version>

Type

PULLUP - Pull-up Resistor

PULLDOWN - Pull-down Resistor

Bit-Size

(1), 2, 4, 8, 12, 16, 32

Version

S - Single pin version

B - Bus pin version

Functional Classes

This section lists the name of all components along with a short description. Components are grouped according to functional class; the following classes are available:

Arithmetic Function

Various types of Arithmetic function are available as follows:

- ACC1 — 1-Bit Loadable Cascadable Accumulator with Synchronous Reset
- ACC2B — 2-Bit Loadable Cascadable Accumulator with Synchronous Reset, Bus Version
- ACC2S — 2-Bit Loadable Cascadable Accumulator with Synchronous Reset, Single Pin Version
- ACC4B — 4-Bit Loadable Cascadable Accumulator with Synchronous Reset, Bus Version
- ACC4S — 4-Bit Loadable Cascadable Accumulator with Synchronous Reset, Single Pin Version
- ACC8B — 8-Bit Loadable Cascadable Accumulator with Synchronous Reset, Bus Version
- ACC16B — 16-Bit Loadable Cascadable Accumulator with Synchronous Reset, Bus Version
- ACC32B — 32-Bit Loadable Cascadable Accumulator with Synchronous Reset, Bus Version
- ADD1 — 1-Bit Cascadable Full Adder
- ADD2B — 2-Bit Cascadable Full Adder with Signed and Unsigned Operations, Bus Version
- ADD2S — 2-Bit Cascadable Full Adder with Signed and Unsigned Operations, Single Pin Version
- ADD4B — 4-Bit Cascadable Full Adder with Signed and Unsigned Operations, Bus Version
- ADD4S — 4-Bit Cascadable Full Adder with Signed and Unsigned Operations, Single Pin Version
- ADD8B — 8-Bit Cascadable Full Adder with Signed and Unsigned Operations, Bus Version
- ADD16B — 16-Bit Cascadable Full Adder with Signed and Unsigned Operations, Bus Version
- ADD32B — 32-Bit Cascadable Full Adder with Signed and Unsigned Operations, Bus Version
- ADDF2B — 2-Bit Cascadable Unsigned Binary Full Adder, Bus Version
- ADDF2S — 2-Bit Cascadable Unsigned Binary Full Adder, Single Pin Version
- ADDF4B — 4-Bit Cascadable Unsigned Binary Full Adder, Bus Version
- ADDF4S — 4-Bit Cascadable Unsigned Binary Full Adder, Single Pin Version
- ADDF8B — 8-Bit Cascadable Unsigned Binary Full Adder, Bus Version

- ADDF16B 16-Bit Cascadable Unsigned Binary Full Adder, Bus Version
- ADDF32B 32-Bit Cascadable Unsigned Binary Full Adder, Bus Version
- ADDFR2B 2-Bit Cascadable Unsigned Binary Registered Full Adder, Bus Version
- ADDFR2S 2-Bit Cascadable Unsigned Binary Registered Full Adder, Single Pin Version
- ADDFR4B 4-Bit Cascadable Unsigned Binary Registered Full Adder, Bus Version
- ADDFR4S 4-Bit Cascadable Unsigned Binary Registered Full Adder, Single Pin Version
- ADDFR8B 8-Bit Cascadable Unsigned Binary Registered Full Adder, Bus Version
- ADDFR16B 16-Bit Cascadable Unsigned Binary Registered Full Adder, Bus Version
- ADDFR32B 32-Bit Cascadable Unsigned Binary Registered Full Adder, Bus Version
- ADDR1 1-Bit Cascadable Registered Full Adder
- ADDR2B 2-Bit Cascadable Registered Full Adder with Signed and Unsigned Operations, Bus Version
- ADDR2S 2-Bit Cascadable Registered Full Adder with Signed and Unsigned Operations, Single Pin Version
- ADDR4B 4-Bit Cascadable Registered Full Adder with Signed and Unsigned Operations, Bus Version
- ADDR4S 4-Bit Cascadable Registered Full Adder with Signed and Unsigned Operations, Single Pin Version
- ADDR8B 8-Bit Cascadable Registered Full Adder with Signed and Unsigned Operations, Bus Version
- ADDR16B 16-Bit Cascadable Registered Full Adder with Signed and Unsigned Operations, Bus Version
- ADDR32B 32-Bit Cascadable Registered Full Adder with Signed and Unsigned Operations, Bus Version
- ADSU1 1-Bit Cascadable Full Adder/Subtracter
- ADSU2B 2-Bit Cascadable Full Adder/Subtracter with Signed and Unsigned Operations, Bus Version
- ADSU2S 2-Bit Cascadable Full Adder/Subtracter with Signed and Unsigned Operations, Single Pin Version
- ADSU4B 4-Bit Cascadable Full Adder/Subtracter with Signed and Unsigned Operations, Bus Version
- ADSU4S 4-Bit Cascadable Full Adder/Subtracter with Signed and Unsigned Operations, Single Pin Version
- ADSU8B 8-Bit Cascadable Full Adder/Subtracter with Signed and Unsigned Operations, Bus Version
- ADSU16B 16-Bit Cascadable Full Adder/Subtracter with Signed and Unsigned Operations, Bus Version

- ADSU32B 32-Bit Cascadable Full Adder/Subtracter with Signed and Unsigned Operations, Bus Version
- ADSUR1 1-Bit Cascadable Registered Full Adder/Subtracter
- ADSUR2B 2-Bit Cascadable Registered Full Adder/Subtracter with Signed and Unsigned Operations, Bus Version
- ADSUR2S 2-Bit Cascadable Registered Full Adder/Subtracter with Signed and Unsigned Operations, Single Pin Version
- ADSUR4B 4-Bit Cascadable Registered Full Adder/Subtracter with Signed and Unsigned Operations, Bus Version
- ADSUR4S 4-Bit Cascadable Registered Full Adder/Subtracter with Signed and Unsigned Operations, Single Pin Version
- ADSUR8B 8-Bit Cascadable Registered Full Adder/Subtracter with Signed and Unsigned Operations, Bus Version
- ADSUR16B 16-Bit Cascadable Registered Full Adder/Subtracter with Signed and Unsigned Operations, Bus Version
- ADSUR32B 32-Bit Cascadable Registered Full Adder/Subtracter with Signed and Unsigned Operations, Bus Version
- MULT2B 2x2 Signed Multiplier, Bus Version
- MULT2S 2x2 Signed Multiplier, Single Pin Version
- MULT4B 4x4 Signed Multiplier, Bus Version
- MULT4S 4x4 Signed Multiplier, Single Pin Version
- MULT8B 8x8 Signed Multiplier, Bus Version
- MULT16B 16x16 Signed Multiplier, Bus Version
- MULT18B 18x18 Signed Multiplier, Bus Version
- MULT32B 32x32 Signed Multiplier, Bus Version
- MULTR2B 2x2 Signed Registered Multiplier, Bus Version
- MULTR2S 2x2 Signed Registered Multiplier, Single Pin Version
- MULTR4B 4x4 Signed Registered Multiplier, Bus Version
- MULTR4S 4x4 Signed Registered Multiplier, Single Pin Version
- MULTR8B 8x8 Signed Registered Multiplier, Bus Version
- MULTR16B 16x16 Signed Registered Multiplier, Bus Version
- MULTR18B 18x18 Signed Registered Multiplier, Bus Version
- MULTR32B 32x32 Signed Registered Multiplier, Bus Version
- MULTU2B 2x2 Unsigned Multiplier, Bus Version
- MULTU2S 2x2 Unsigned Multiplier, Single Pin Version
- MULTU4B 4x4 Unsigned Multiplier, Bus Version
- MULTU4S 4x4 Unsigned Multiplier, Single Pin Version
- MULTU8B 8x8 Unsigned Multiplier, Bus Version

- MULTU16B 16x16 Unsigned Multiplier, Bus Version
- MULTU18B 18x18 Unsigned Multiplier, Bus Version
- MULTU32B 32x32 Unsigned Multiplier, Bus Version
- MULTUR2B 2x2 Unsigned Registered Multiplier, Bus Version
- MULTUR2S 2x2 Unsigned Registered Multiplier, Single Pin Version
- MULTUR4B 4x4 Unsigned Registered Multiplier, Bus Version
- MULTUR4S 4x4 Unsigned Registered Multiplier, Single Pin Version
- MULTUR8B 8x8 Unsigned Registered Multiplier, Bus Version
- MULTUR16B 16x16 Unsigned Registered Multiplier, Bus Version
- MULTUR18B 18x18 Unsigned Registered Multiplier, Bus Version
- MULTUR32B 32x32 Unsigned Registered Multiplier, Bus Version
- PAR9B 9-Bit Odd/Even Parity Generators/Checker, Bus Version
- PAR9S 9-Bit Odd/Even Parity Generators/Checker, Single Pin Version

Buffer

Multiple input and tri-state buffers are available as follows:

- BUF 1-bit General Purpose (Non-inverting) Buffer
- BUF2B 2-Bit General Purpose (Non-inverting) Buffer, Bus Version
- BUF2S 2-Bit General Purpose (Non-inverting) Buffer, Single Pin Version
- BUF3B 3-Bit General Purpose (Non-inverting) Buffer, Bus Version
- BUF3S 3-Bit General Purpose (Non-inverting) Buffer, Single Pin Version
- BUF4B 4-Bit General Purpose (Non-inverting) Buffer, Bus Version
- BUF4S 4-Bit General Purpose (Non-inverting) Buffer, Single Pin Version
- BUF5B 5-Bit General Purpose (Non-inverting) Buffer, Bus Version
- BUF5S 5-Bit General Purpose (Non-inverting) Buffer, Single Pin Version
- BUF6B 6-Bit General Purpose (Non-inverting) Buffer, Bus Version
- BUF6S 6-Bit General Purpose (Non-inverting) Buffer, Single Pin Version
- BUF7B 7-Bit General Purpose (Non-inverting) Buffer, Bus Version
- BUF7S 7-Bit General Purpose (Non-inverting) Buffer, Single Pin Version
- BUF8B 8-Bit General Purpose (Non-inverting) Buffer, Bus Version
- BUF8S 8-Bit General Purpose (Non-inverting) Buffer, Single Pin Version
- BUF9B 9-Bit General Purpose (Non-inverting) Buffer, Bus Version
- BUF9S 9-Bit General Purpose (Non-inverting) Buffer, Single Pin Version
- BUF10B 10-Bit General Purpose (Non-inverting) Buffer, Bus Version
- BUF10S 10-Bit General Purpose (Non-inverting) Buffer, Single Pin Version
- BUF12B 12-Bit General Purpose (Non-inverting) Buffer, Bus Version
- BUF12S 12-Bit General Purpose (Non-inverting) Buffer, Single Pin Version
- BUF16B 16-Bit General Purpose (Non-inverting) Buffer, Bus Version
- BUF16S 16-Bit General Purpose (Non-inverting) Buffer, Single Pin Version
- BUF32B 32-Bit General Purpose (Non-inverting) Buffer, Bus Version
- BUF32S 32-Bit General Purpose (Non-inverting) Buffer, Single Pin Version
- BUFE 1-bit 3-state Buffer with Active High Enable
- BUFE2B 2-Bit 3-state Buffer with Active High Enable, Bus Version
- BUFE2S 2-Bit 3-state Buffer with Active High Enable, Single Pin Version
- BUFE3B 3-Bit 3-state Buffer with Active High Enable, Bus Version
- BUFE3S 3-Bit 3-state Buffer with Active High Enable, Single Pin Version
- BUFE4B 4-Bit 3-state Buffer with Active High Enable, Bus Version
- BUFE4S 4-Bit 3-state Buffer with Active High Enable, Single Pin Version
- BUFE5B 5-Bit 3-state Buffer with Active High Enable, Bus Version

- BUFE5S 5-Bit 3-state Buffer with Active High Enable, Single Pin Version
- BUFE6B 6-Bit 3-state Buffer with Active High Enable, Bus Version
- BUFE6S 6-Bit 3-state Buffer with Active High Enable, Single Pin Version
- BUFE7B 7-Bit 3-state Buffer with Active High Enable, Bus Version
- BUFE7S 7-Bit 3-state Buffer with Active High Enable, Single Pin Version
- BUFE8B 8-Bit 3-state Buffer with Active High Enable, Bus Version
- BUFE8S 8-Bit 3-state Buffer with Active High Enable, Single Pin Version
- BUFE9B 9-Bit 3-state Buffer with Active High Enable, Bus Version
- BUFE9S 9-Bit 3-state Buffer with Active High Enable, Single Pin Version
- BUFE10B 10-Bit 3-state Buffer with Active High Enable, Bus Version
- BUFE10S 10-Bit 3-state Buffer with Active High Enable, Single Pin Version
- BUFE12B 12-Bit 3-state Buffer with Active High Enable, Bus Version
- BUFE12S 12-Bit 3-state Buffer with Active High Enable, Single Pin Version
- BUFE16B 16-Bit 3-state Buffer with Active High Enable, Bus Version
- BUFE16S 16-Bit 3-state Buffer with Active High Enable, Single Pin Version
- BUFE32B 32-Bit 3-state Buffer with Active High Enable, Bus Version
- BUFE32S 32-Bit 3-state Buffer with Active High Enable, Single Pin Version
- BUFT 1-Bit 3-state Buffer with Active Low Enable
- BUFT2B 2-Bit 3-state Buffer with Active Low Enable, Bus Version
- BUFT2S 2-Bit 3-state Buffer with Active Low Enable, Single Pin Version
- BUFT3B 3-Bit 3-state Buffer with Active Low Enable, Bus Version
- BUFT3S 3-Bit 3-state Buffer with Active Low Enable, Single Pin Version
- BUFT4B 4-Bit 3-state Buffer with Active Low Enable, Bus Version
- BUFT4S 4-Bit 3-state Buffer with Active Low Enable, Single Pin Version
- BUFT5B 5-Bit 3-state Buffer with Active Low Enable, Bus Version
- BUFT5S 5-Bit 3-state Buffer with Active Low Enable, Single Pin Version
- BUFT6B 6-Bit 3-state Buffer with Active Low Enable, Bus Version
- BUFT6S 6-Bit 3-state Buffer with Active Low Enable, Single Pin Version
- BUFT7B 7-Bit 3-state Buffer with Active Low Enable, Bus Version
- BUFT7S 7-Bit 3-state Buffer with Active Low Enable, Single Pin Version
- BUFT8B 8-Bit 3-state Buffer with Active Low Enable, Bus Version
- BUFT8S 8-Bit 3-state Buffer with Active Low Enable, Single Pin Version
- BUFT9B 9-Bit 3-state Buffer with Active Low Enable, Bus Version
- BUFT9S 9-Bit 3-state Buffer with Active Low Enable, Single Pin Version
- BUFT10B 10-Bit 3-state Buffer with Active Low Enable, Bus Version
- BUFT10S 10-Bit 3-state Buffer with Active Low Enable, Single Pin Version

- BUFT12B 12-Bit 3-state Buffer with Active Low Enable, Bus Version
- BUFT12S 12-Bit 3-state Buffer with Active Low Enable, Single Pin Version
- BUFT16B 16-Bit 3-state Buffer with Active Low Enable, Bus Version
- BUFT16S 16-Bit 3-state Buffer with Active Low Enable, Single Pin Version
- BUFT32B 32-Bit 3-state Buffer with Active Low Enable, Bus Version
- BUFT32S 32-Bit 3-state Buffer with Active Low Enable, Single Pin Version
- IOBUF Input/Output Buffer
- IOBUF2B 2-Bit Input/Output Buffer, Bus Version
- IOBUF2S 2-Bit Input/Output Buffer, Single Pin Version
- IOBUF3B 3-Bit Input/Output Buffer, Bus Version
- IOBUF4B 4-Bit Input/Output Buffer, Bus Version
- IOBUF4S 4-Bit Input/Output Buffer, Single Pin Version
- IOBUF5B 5-Bit Input/Output Buffer, Bus Version
- IOBUF6B 6-Bit Input/Output Buffer, Bus Version
- IOBUF7B 7-Bit Input/Output Buffer, Bus Version
- IOBUF8B 8-Bit Input/Output Buffer, Bus Version
- IOBUF9B 9-Bit Input/Output Buffer, Bus Version
- IOBUF10B 10-Bit Input/Output Buffer, Bus Version
- IOBUF12B 12-Bit Input/Output Buffer, Bus Version
- IOBUF16B 16-Bit Input/Output Buffer, Bus Version
- IOBUF32B 32-Bit Input/Output Buffer, Bus Version
- IOBUFC2B 2-Bit Input/Output Buffer With Separated Control, Bus Version
- IOBUFC2S 2-Bit Input/Output Buffer With Separated Control, Single Pin Version
- IOBUFC3B 3-Bit Input/Output Buffer With Separated Control, Bus Version
- IOBUFC4B 4-Bit Input/Output Buffer With Separated Control, Bus Version
- IOBUFC4S 4-Bit Input/Output Buffer With Separated Control, Single Pin Version
- IOBUFC5B 5-Bit Input/Output Buffer With Separated Control, Bus Version
- IOBUFC6B 6-Bit Input/Output Buffer With Separated Control, Bus Version
- IOBUFC7B 7-Bit Input/Output Buffer With Separated Control, Bus Version
- IOBUFC8B 8-Bit Input/Output Buffer With Separated Control, Bus Version
- IOBUFC9B 9-Bit Input/Output Buffer With Separated Control, Bus Version
- IOBUFC10B 10-Bit Input/Output Buffer With Separated Control, Bus Version
- IOBUFC12B 12-Bit Input/Output Buffer With Separated Control, Bus Version
- IOBUFC16B 16-Bit Input/Output Buffer With Separated Control, Bus Version
- IOBUFC32B 32-Bit Input/Output Buffer With Separated Control, Bus Version

Bus Joiner

Bus joiners are components that allow splitting or merging of buss slices to suite your needs. Various types are available as follows:

- J2B2_4B — 2 x 2-Bit input bus to 1 x 4-bit output bus
- J3B_3S — 3-Bit input bus to 3 Single pin outputs
- J3S_3B — 3 Single pin inputs to single 3-Bit output bus
- J3S_3BX — 3 Single pin IO to single 3-Bit IO bus
- J4B2_8B — 2 x 4-Bit input bus to 1 x 8-bit output bus
- J4B4_16B — 4 x 4-Bit input bus to 1 x 16-bit output bus
- J4B8_32B — 8 x 4-Bit input bus to 1 x 32-bit output bus
- J4B_2B2 — 1 x 4-bit input bus to 2 x 2-Bit output bus
- J4B_2B2X — 1 x 4-bit IO bus to 2 x 2-Bit IO bus
- J4B_4S — 4-Bit input bus to 4 Single pin outputs
- J4S_4B — 4 Single pin inputs to single 4-Bit output bus
- J4S_4BX — 4 Single pin IO to single 4-Bit IO bus
- J5B_5S — 5-Bit input bus to 5 Single pin outputs
- J5S_5B — 5 Single pin inputs to single 5-Bit output bus
- J5S_5BX — 5 Single pin IO to single 5-Bit IO bus
- J6B_6S — 6-Bit input bus to 6 Single pin outputs
- J6S_6B — 6 Single pin inputs to single 6-Bit output bus
- J6S_6BX — 6 Single pin IO to single 6-Bit IO bus
- J7B_7S — 7-Bit input bus to 7 Single pin outputs
- J7S_7B — 7 Single pin inputs to single 7-Bit output bus
- J7S_7BX — 7 Single pin IO to single 7-Bit IO bus
- J8B2_16B — 2 x 8-Bit input bus to 1 x 16-bit output bus
- J8B4_32B — 4 x 8-Bit input bus to 1 x 32-bit output bus
- J8B_4B2 — 1 x 8-bit input bus to 2 x 4-Bit output bus
- J8B_4B2X — 1 x 8-bit IO bus to 2 x 4-Bit IO bus
- J8B_8S — 8-Bit input bus to 8 single pin outputs
- J8S_8B — 8 Single pin inputs to single 8-Bit output bus
- J8S_8BX — 8 Single pin IO to single 8-Bit IO bus
- J9B_9S — 9-Bit input bus to 9 Single pin outputs
- J9S_9B — 9 Single pin inputs to single 9-Bit output bus
- J9S_9BX — 9 Single pin IO to single 9-Bit IO bus
- J10B_10S — 10-Bit input bus to 10 Single pin outputs

- J10S_10B 10 Single pin inputs to single 10-Bit output bus
- J10S_10BX 10 Single pin IO to single 10-Bit IO bus
- J12B_12S 12-Bit input bus to 12 single pin outputs
- J12S_12B 12 Single pin inputs to single 12-Bit output bus
- J12S_12BX 12 single-Bit IO to single 12-Bit IO bus
- J16B2_32B 2 x 16-Bit input bus to 1 x 32-bit output bus
- J16B_4B4 1 x 16-bit input bus to 4 x 4-Bit output bus
- J16B_4B4X 1 x 16-bit IO bus to 4 x 4-Bit IO bus
- J16B_8B2 1 x 16-bit input bus to 2 x 8-Bit output bus
- J16B_8B2X 1 x 16-bit IO bus to 2 x 8-Bit IO bus
- J16B_16S Single 16-Bit input bus to 16 single pin outputs
- J16S_16B 16 Single pin inputs to single 16-Bit output bus
- J16S_16BX 16 Single pin IO to single 16-Bit IO bus
- J32B_4B8 1 x 32-Bit input bus to 8 x 4-Bit output bus
- J32B_4B8X 1 x 32-Bit IO bus to 8 x 4-Bit IO bus
- J32B_8B4 1 x 32-Bit input bus to 4 x 8-Bit output bus
- J32B_8B4X 1 x 32-Bit IO bus to 4 x 8-Bit IO bus
- J32B_16B2 1 x 32-bit input bus to 2 x 16-Bit output bus
- J32B_16B2X 1 x 32-bit IO bus to 2 x 16-Bit IO bus
- JB System BUS Joiner

Clock Divider

General and programmable clock dividers are available as follows:

- CDIV2 Clock Divider by 2
- CDIV2DC50 Clock Divider by 2 with 50% Duty Cycle Output
- CDIV3 Clock Divider by 3
- CDIV4 Clock Divider by 4
- CDIV4DC50 Clock Divider by 4 with 50% Duty Cycle Output
- CDIV5 Clock Divider by 5
- CDIV6 Clock Divider by 6
- CDIV6DC50 Clock Divider by 6 with 50% Duty Cycle Output
- CDIV7 Clock Divider by 7
- CDIV8 Clock Divider by 8
- CDIV8DC50 Clock Divider by 8 with 50% Duty Cycle Output
- CDIV9 Clock Divider by 9
- CDIV10 Clock Divider by 10
- CDIV10DC50 Clock Divider by 10 with 50% Duty Cycle Output
- CDIV12 Clock Divider by 12
- CDIV12DC50 Clock Divider by 12 with 50% Duty Cycle Output
- CDIV16 Clock Divider by 16
- CDIV16DC50 Clock Divider by 16 with 50% Duty Cycle Output
- CDIV20 Clock Divider by 20
- CDIV20DC50 Clock Divider by 20 with 50% Duty Cycle Output
- CDIV24 Clock Divider by 24
- CDIV24DC50 Clock Divider by 24 with 50% Duty Cycle Output
- CDIV32 Clock Divider by 32
- CDIV32DC50 Clock Divider by 32 with 50% Duty Cycle Output
- CDIV64 Clock Divider by 64
- CDIV64DC50 Clock Divider by 64 with 50% Duty Cycle Output
- CDIV128 Clock Divider by 128
- CDIV128DC50 Clock Divider by 128 with 50% Duty Cycle Output
- CDIV256 Clock Divider by 256
- CDIV256DC50 Clock Divider by 256 with 50% Duty Cycle Output
- CDIVN_8 8-Bit Programmable Clock Divider
- CDIVN_16 16-Bit Programmable Clock Divider
- CDIVN_32 32-Bit Programmable Clock Divider

Clock Manager

Various clock manager components are available as follows:

- CLKMAN_1 Single Operational Output Clock Manager
- CLKMAN_2 Dual Operational Output Clock Manager
- CLKMAN_3 Multiple Operational Output Clock Manager
- CLKMAN_4 Multiple Operational Output Clock Manager

Comparator

Magnitude, identity and address comparators are available as follows:

- COMP2B 2-Bit Identity Comparator, Bus Version
- COMP2S 2-Bit Identity Comparator, Single Pin Version
- COMP3B 3-Bit Identity Comparator, Bus Version
- COMP3S 3-Bit Identity Comparator, Single Pin Version
- COMP4B 4-Bit Identity Comparator, Bus Version
- COMP4S 4-Bit Identity Comparator, Single Pin Version
- COMP5B 5-Bit Identity Comparator, Bus Version
- COMP5S 5-Bit Identity Comparator, Single Pin Version
- COMP6B 6-Bit Identity Comparator, Bus Version
- COMP6S 6-Bit Identity Comparator, Single Pin Version
- COMP7B 7-Bit Identity Comparator, Bus Version
- COMP7S 7-Bit Identity Comparator, Single Pin Version
- COMP8B 8-Bit Identity Comparator, Bus Version
- COMP8S 8-Bit Identity Comparator, Single Pin Version
- COMP9B 9-Bit Identity Comparator, Bus Version
- COMP9S 9-Bit Identity Comparator, Single Pin Version
- COMP10B 10-Bit Identity Comparator, Bus Version
- COMP10S 10-Bit Identity Comparator, Single Pin Version
- COMP12B 12-Bit Identity Comparator, Bus Version
- COMP12S 12-Bit Identity Comparator, Single Pin Version
- COMP16B 16-Bit Identity Comparator, Bus Version
- COMP16S 16-Bit Identity Comparator, Single Pin Version
- COMP32B 32-Bit Identity Comparator, Bus Version
- COMPM2B 2-Bit Magnitude Comparator, Bus Version
- COMPM2S 2-Bit Magnitude Comparator, Single Pin Version
- COMPM3B 3-Bit Magnitude Comparator, Bus Version
- COMPM3S 3-Bit Magnitude Comparator, Single Pin Version
- COMPM4B 4-Bit Magnitude Comparator, Bus Version
- COMPM4S 4-Bit Magnitude Comparator, Single Pin Version
- COMPM5B 5-Bit Magnitude Comparator, Bus Version
- COMPM5S 5-Bit Magnitude Comparator, Single Pin Version
- COMPM6B 6-Bit Magnitude Comparator, Bus Version
- COMPM6S 6-Bit Magnitude Comparator, Single Pin Version

- COMPM7B 7-Bit Magnitude Comparator, Bus Version
- COMPM7S 7-Bit Magnitude Comparator, Single Pin Version
- COMPM8B 8-Bit Magnitude Comparator, Bus Version
- COMPM8S 8-Bit Magnitude Comparator, Single Pin Version
- COMPM9B 9-Bit Magnitude Comparator, Bus Version
- COMPM9S 9-Bit Magnitude Comparator, Single Pin Version
- COMPM10B 10-Bit Magnitude Comparator, Bus Version
- COMPM10S 10-Bit Magnitude Comparator, Single Pin Version
- COMPM12B 12-Bit Magnitude Comparator, Bus Version
- COMPM12S 12-Bit Magnitude Comparator, Single Pin Version
- COMPM16B 16-Bit Magnitude Comparator, Bus Version
- COMPM16S 16-Bit Magnitude Comparator, Single Pin Version
- COMPM32B 32-Bit Magnitude Comparator, Bus Version

Counter

Various function and types of counter are available as follows:

- CB2CEB — 2-Bit Cascadable Binary Counters with Clock Enable and Asynchronous Clear, Bus Version
- CB2CES — 2-Bit Cascadable Binary Counters with Clock Enable and Asynchronous Clear, Single Pin Version
- CB2CLEB — 2-Bit Loadable Cascadable Binary Counter with Clock Enable and Asynchronous Clear, Bus Version
- CB2CLEDB — 2-Bit Loadable Cascadable Bidirectional Binary Counter with Clock Enable and Asynchronous Clear, Bus Version
- CB2CLEDS — 2-Bit Loadable Cascadable Bidirectional Binary Counter with Clock Enable and Asynchronous Clear, Single Pin Version
- CB2CLES — 2-Bit Loadable Cascadable Binary Counter with Clock Enable and Asynchronous Clear, Single Pin Version
- CB2REB — 2-Bit Cascadable Binary Counter with Clock Enable and Synchronous Reset, Bus Version
- CB2RES — 2-Bit Cascadable Binary Counter with Clock Enable and Synchronous Reset, Single Pin Version
- CB2RLEB — 2-Bit Loadable Cascadable Binary Counter with Clock Enable and Synchronous Reset, Bus Version
- CB2RLES — 2-Bit Loadable Cascadable Binary Counter with Clock Enable and Synchronous Reset, Single Pin Version
- CB4CEB — 4-Bit Cascadable Binary Counters with Clock Enable and Asynchronous Clear, Bus Version
- CB4CES — 4-Bit Cascadable Binary Counters with Clock Enable and Asynchronous Clear, Single Pin Version
- CB4CLEB — 4-Bit Loadable Cascadable Binary Counter with Clock Enable and Asynchronous Clear, Bus Version
- CB4CLEDB — 4-Bit Loadable Cascadable Bidirectional Binary Counter with Clock Enable and Asynchronous Clear, Bus Version
- CB4CLEDS — 4-Bit Loadable Cascadable Bidirectional Binary Counter with Clock Enable and Asynchronous Clear, Single Pin Version
- CB4CLES — 4-Bit Loadable Cascadable Binary Counter with Clock Enable and Asynchronous Clear, Single Pin Version
- CB4REB — 4-Bit Cascadable Binary Counter with Clock Enable and Synchronous Reset, Bus Version
- CB4RES — 4-Bit Cascadable Binary Counter with Clock Enable and Synchronous Reset, Single Pin Version

- CB4RLEB — 4-Bit Loadable Cascadable Binary Counter with Clock Enable and Synchronous Reset, Bus Version
- CB4RLES — 4-Bit Loadable Cascadable Binary Counter with Clock Enable and Synchronous Reset, Single Pin Version
- CB8CEB — 8-Bit Cascadable Binary Counters with Clock Enable and Asynchronous Clear, Bus Version
- CB8CES — 8-Bit Cascadable Binary Counters with Clock Enable and Asynchronous Clear, Single Pin Version
- CB8CLEB — 8-Bit Loadable Cascadable Binary Counter with Clock Enable and Asynchronous Clear, Bus Version
- CB8CLEDB — 8-Bit Loadable Cascadable Bidirectional Binary Counter with Clock Enable and Asynchronous Clear, Bus Version
- CB8CLEDS — 8-Bit Loadable Cascadable Bidirectional Binary Counter with Clock Enable and Asynchronous Clear, Single Pin Version
- CB8CLES — 8-Bit Loadable Cascadable Binary Counter with Clock Enable and Asynchronous Clear, Single Pin Version
- CB8REB — 8-Bit Cascadable Binary Counter with Clock Enable and Synchronous Reset, Bus Version
- CB8RES — 8-Bit Cascadable Binary Counter with Clock Enable and Synchronous Reset, Single Pin Version
- CB8RLEB — 8-Bit Loadable Cascadable Binary Counter with Clock Enable and Synchronous Reset, Bus Version
- CB8RLES — 8-Bit Loadable Cascadable Binary Counter with Clock Enable and Synchronous Reset, Single Pin Version
- CB16CEB — 16-Bit Cascadable Binary Counters with Clock Enable and Asynchronous Clear, Bus Version
- CB16CES — 16-Bit Cascadable Binary Counters with Clock Enable and Asynchronous Clear, Single Pin Version
- CB16CLEB — 16-Bit Loadable Cascadable Binary Counters with Clock Enable and Asynchronous Clear, Bus Version
- CB16CLEDB — 16-Bit Loadable Cascadable Bidirectional Binary Counter with Clock Enable and Asynchronous Clear, Bus Version
- CB16CLEDS — 16-Bit Loadable Cascadable Bidirectional Binary Counter with Clock Enable and Asynchronous Clear, Single Pin Version
- CB16CLES — 16-Bit Loadable Cascadable Binary Counters with Clock Enable and Asynchronous Clear, Single Pin Version
- CB16REB — 16-Bit Cascadable Binary Counter with Clock Enable and Synchronous Reset, Bus Version
- CB16RES — 16-Bit Cascadable Binary Counter with Clock Enable and Synchronous Reset, Single Pin Version

- CB16RLEB 16-Bit Loadable Cascadable Binary Counter with Clock Enable and Synchronous Reset, Bus Version
- CB16RLES 16-Bit Loadable Cascadable Binary Counter with Clock Enable and Synchronous Reset, Single Pin Version
- CB32CEB 32-Bit Cascadable Binary Counters with Clock Enable and Asynchronous Clear, Bus Version
- CB32CLEB 32-Bit Loadable Cascadable Binary Counters with Clock Enable and Asynchronous Clear, Bus Version
- CB32CLEDB 32-Bit Loadable Cascadable Bidirectional Binary Counter with Clock Enable and Asynchronous Clear, Bus Version
- CB32REB 32-Bit Cascadable Binary Counter with Clock Enable and Synchronous Reset, Bus Version
- CB32RLEB 32-Bit Loadable Cascadable Binary Counter with Clock Enable and Synchronous Reset, Bus Version
- CD4CEB Cascadable BCD Counter with Clock Enable and Asynchronous Clear, Bus Version
- CD4CES Cascadable BCD Counter with Clock Enable and Asynchronous Clear, Single Pin Version
- CD4CLEB Loadable Cascadable BCD Counter with Clock Enable and Asynchronous Clear, Bus Version
- CD4CLES Loadable Cascadable BCD Counter with Clock Enable and Asynchronous Clear, Single Pin Version
- CD4REB Cascadable BCD Counter with Clock Enable and Synchronous Reset, Bus Version
- CD4RES Cascadable BCD Counter with Clock Enable and Synchronous Reset, Single Pin Version
- CD4RLEB Loadable Cascadable BCD Counter with Clock Enable and Synchronous Reset, Bus Version
- CD4RLES Loadable Cascadable BCD Counter with Clock Enable and Synchronous Reset, Single Pin Version
- CJ2CEB 2-Bit Johnson Counter with Clock Enable and Asynchronous Clear, Bus Version
- CJ2CES 2-Bit Johnson Counter with Clock Enable and Asynchronous Clear, Single Pin Version
- CJ2REB 2-Bit Johnson Counter with Clock Enable and Synchronous Reset, Bus Version
- CJ2RES 2-Bit Johnson Counter with Clock Enable and Synchronous Reset, Single Pin Version
- CJ4CEB 4-Bit Johnson Counter with Clock Enable and Asynchronous Clear, Bus Version

- CJ4CES — 4-Bit Johnson Counter with Clock Enable and Asynchronous Clear, Single Pin Version
- CJ4REB — 4-Bit Johnson Counter with Clock Enable and Synchronous Reset, Bus Version
- CJ4RES — 4-Bit Johnson Counter with Clock Enable and Synchronous Reset, Single Pin Version
- CJ5CEB — 5-Bit Johnson Counter with Clock Enable and Asynchronous Clear, Bus Version
- CJ5CES — 5-Bit Johnson Counter with Clock Enable and Asynchronous Clear, Single Pin Version
- CJ5REB — 5-Bit Johnson Counters with Clock Enable and Synchronous Reset, Bus Version
- CJ5RES — 5-Bit Johnson Counters with Clock Enable and Synchronous Reset, Single Pin Version
- CJ8CEB — 8-Bit Johnson Counter with Clock Enable and Asynchronous Clear, Bus Version
- CJ8CES — 8-Bit Johnson Counter with Clock Enable and Asynchronous Clear, Single Pin Version
- CJ8REB — 8-Bit Johnson Counter with Clock Enable and Synchronous Reset, Bus Version
- CJ8RES — 8-Bit Johnson Counter with Clock Enable and Synchronous Reset, Single Pin Version
- CJ16CEB — 16-Bit Johnson Counter with Clock Enable and Asynchronous Clear, Bus Version
- CJ16CES — 16-Bit Johnson Counter with Clock Enable and Asynchronous Clear, Single Pin Version
- CJ16REB — 16-Bit Johnson Counter with Clock Enable and Synchronous Reset, Bus Version
- CJ16RES — 16-Bit Johnson Counter with Clock Enable and Synchronous Reset, Single Pin Version
- CJ32CEB — 32-Bit Johnson Counter with Clock Enable and Asynchronous Clear, Bus Version
- CJ32REB — 32-Bit Johnson Counter with Clock Enable and Synchronous Reset, Bus Version
- CR2CEB — 2-Bit Negative-Edge Binary Ripple Counters with Clock Enable and Asynchronous Clear, Bus Version
- CR2CES — 2-Bit Negative-Edge Binary Ripple Counters with Clock Enable and Asynchronous Clear, Single Pin Version
- CR4CEB — 4-Bit Negative-Edge Binary Ripple Counters with Clock Enable and Asynchronous Clear, Bus Version

- CR4CES — 4-Bit Negative-Edge Binary Ripple Counters with Clock Enable and Asynchronous Clear, Single Pin Version
- CR8CEB — 8-Bit Negative-Edge Binary Ripple Counters with Clock Enable and Asynchronous Clear, Bus Version
- CR8CES — 8-Bit Negative-Edge Binary Ripple Counters with Clock Enable and Asynchronous Clear, Single Pin Version
- CR16CEB — 16-Bit Negative-Edge Binary Ripple Counters with Clock Enable and Asynchronous Clear, Bus Version
- CR16CES — 16-Bit Negative-Edge Binary Ripple Counters with Clock Enable and Asynchronous Clear, Single Pin Version
- CR32CEB — 32-Bit Negative-Edge Binary Ripple Counters with Clock Enable and Asynchronous Clear, Bus Version

Decoder

Various type of decoders are available as follows:

- D2_4B — Binary 2- to 4-Bit Decoder, Bus Version
- D2_4EB — Binary 2- to 4-Bit Decoder with Enable, Bus Version
- D2_4ES — Binary 2- to 4-Bit Decoder with Enable, Single Pin Version
- D2_4S — Binary 2- to 4-Bit Decoder, Single Pin Version
- D3_8B — Binary 3- to 8-Bit Decoder, Bus Version
- D3_8EB — Binary 3- to 8-Bit Decoder with Enable, Bus Version
- D3_8ES — Binary 3- to 8-Bit Decoder with Enable, Single Pin Version
- D3_8S — Binary 3- to 8-Bit Decoder, Single Pin Version
- D4_10B — Binary-Coded-Decimal (BCD) Decoder, Bus Version
- D4_10EB — Binary-Coded-Decimal (BCD) Decoder with Enable, Bus Version
- D4_10ES — Binary-Coded-Decimal (BCD) Decoder with Enable, Single Pin Version
- D4_10S — Binary-Coded-Decimal (BCD) Decoder, Single Pin Version
- D4_16B — Binary 4- to 16-Bit Decoder, Bus Version
- D4_16EB — Binary 4- to 16-Bit Decoder with Enable, Bus Version
- D4_16ES — Binary 4- to 16-Bit Decoder with Enable, Single Pin Version
- D4_16S — Binary 4- to 16-Bit Decoder, Single Pin Version
- D5_32B — Binary 5- to 32-Bit Decoder, Bus Version
- D5_32EB — Binary 5- to 32-Bit Decoder with Enable, Bus Version
- D7SEGB — 7-Segment-Display Decoder for Common-Cathode LED (Active High Output), Bus Version
- D7SEGNB — 7-Segment-Display Decoder for Common-Anode LED (Active Low Output), Bus Version
- D7SEGNS — 7-Segment-Display Decoder for Common-Anode LED (Active Low Output), Single Pin Version
- D7SEGS — 7-Segment-Display Decoder for Common-Cathode LED (Active High Output), Single Pin Version

Encoder

Various type of encoders are available as follows:

- E4_2B 4- to 2-Bit Priority Encoder, Bus Version
- E4_2EB 4- to 2-Bit Priority Encoder with Enable, Bus Version
- E4_2ES 4- to 2-Bit Priority Encoder with Enable, Single Pin Version
- E4_2S 4- to 2-Bit Priority Encoder, Single Pin Version
- E8_3B 8- to 3-Bit Priority Encoder, Bus Version
- E8_3EB 8- to 3-Bit Priority Encoder with Enable, Bus Version
- E8_3ES 8- to 3-Bit Priority Encoder with Enable, Single Pin Version
- E8_3S 8- to 3-Bit Priority Encoder, Single Pin Version
- E10_4B Binary-Coded-Decimal (BCD) Encoder, Bus Version
- E10_4EB Binary-Coded-Decimal (BCD) Encoder with Enable, Bus Version
- E10_4ES Binary-Coded-Decimal (BCD) Encoder with Enable, Single Pin Version
- E10_4S Binary-Coded-Decimal (BCD) Encoder, Single Pin Version
- E16_4B 16- to 4-Bit Priority Encoder, Bus Version
- E16_4EB 16- to 4-Bit Priority Encoder with Enable, Bus Version
- E16_4ES 16- to 4-Bit Priority Encoder with Enable, Single Pin Version
- E16_4S 16- to 4-Bit Priority Encoder, Single Pin Version
- E32_5B 32- to 5-Bit Priority Encoder, Bus Version
- E32_5EB 32- to 5-Bit Priority Encoder with Enable, Bus Version

Flip-Flop

General and multi function flip-flops are available as follows:

- FD — D-Type Flip-Flop
- FD2B — 2-Bit D-Type Flip-Flop, Bus Version
- FD2CB — 2-Bit D-Type Flip-Flop with Asynchronous Clear, Bus Version
- FD2CEB — 2-Bit D-Type Flip-Flop with Clock Enable and Asynchronous Clear, Bus Version
- FD2CES — 2-Bit D-Type Flip-Flop with Clock Enable and Asynchronous Clear, Single Pin Version
- FD2CPB — 2-Bit D-Type Flip-Flop with Asynchronous Preset and Clear, Bus Version
- FD2CPEB — 2-Bit D-Type Flip-Flop with Clock Enable and Asynchronous Preset and Clear, Bus Version
- FD2CPES — 2-Bit D-Type Flip-Flop with Clock Enable and Asynchronous Preset and Clear, Single Pin Version
- FD2CPS — 2-Bit D-Type Flip-Flop with Asynchronous Preset and Clear, Single Pin Version
- FD2CS — 2-Bit D-Type Flip-Flop with Asynchronous Clear, Single Pin Version
- FD2EB — 2-Bit D Flip-Flop with Clock Enable, Bus Version
- FD2ES — 2-Bit D Flip-Flop with Clock Enable, Single Pin Version
- FD2PB — 2-Bit D-Type Flip-Flop with Asynchronous Preset, Bus Version
- FD2PEB — 2-Bit D-Type Flip-Flop with Clock Enable and Asynchronous Preset, Bus Version
- FD2PES — 2-Bit D-Type Flip-Flop with Clock Enable and Asynchronous Preset, Single Pin Version
- FD2PS — 2-Bit D-Type Flip-Flop with Asynchronous Preset, Single Pin Version
- FD2RB — 2-Bit D-Type Flip-Flop with Synchronous Reset, Bus Version
- FD2REB — 2-Bit D-Type Flip-Flop with Clock Enable and Synchronous Reset, Bus Version
- FD2RES — 2-Bit D-Type Flip-Flop with Clock Enable and Synchronous Reset, Single Pin Version
- FD2RS — 2-Bit D-Type Flip-Flop with Synchronous Reset, Single Pin Version
- FD2RSB — 2-Bit D-Type Flip-Flop with Synchronous Reset and Set, Bus Version
- FD2RSEB — 2-Bit D-Type Flip-Flop with Synchronous Reset and Set and Clock Enable, Bus Version
- FD2RSES — 2-Bit D-Type Flip-Flop with Synchronous Reset and Set and Clock Enable, Single Pin Version
- FD2RSS — 2-Bit D-Type Flip-Flop with Synchronous Reset and Set, Single Pin Version

- FD2S 2-Bit D-Type Flip-Flop, Single Pin Version
- FD2SB 2-Bit D-Type Flip-Flop with Synchronous Set, Bus Version
- FD2SEB 2-Bit D-Type Flip-Flop with Clock Enable and Synchronous Set, Bus Version
- FD2SES 2-Bit D-Type Flip-Flop with Clock Enable and Synchronous Set, Single Pin Version
- FD2SRB 2-Bit D-Type Flip-Flop with Synchronous Set and Reset, Bus Version
- FD2SREB 2-Bit D-Type Flip-Flop with Synchronous Set and Reset and Clock Enable, Bus Version
- FD2SRES 2-Bit D-Type Flip-Flop with Synchronous Set and Reset and Clock Enable, Single Pin Version
- FD2SRS 2-Bit D-Type Flip-Flop with Synchronous Set and Reset, Single Pin Version
- FD2SS 2-Bit D-Type Flip-Flop with Synchronous Set, Single Pin Version
- FD4B 4-Bit D-Type Flip-Flop, Bus Version
- FD4CB 4-Bit D-Type Flip-Flop with Asynchronous Clear, Bus Version
- FD4CEB 4-Bit D-Type Flip-Flop with Clock Enable and Asynchronous Clear, Bus Version
- FD4CES 4-Bit D-Type Flip-Flop with Clock Enable and Asynchronous Clear, Single Pin Version
- FD4CPB 4-Bit D-Type Flip-Flop with Asynchronous Preset and Clear, Bus Version
- FD4CPEB 4-Bit D-Type Flip-Flop with Clock Enable and Asynchronous Preset and Clear, Bus Version
- FD4CPES 4-Bit D-Type Flip-Flop with Clock Enable and Asynchronous Preset and Clear, Single Pin Version
- FD4CPS 4-Bit D-Type Flip-Flop with Asynchronous Preset and Clear, Single Pin Version
- FD4CS 4-Bit D-Type Flip-Flop with Asynchronous Clear, Single Pin Version
- FD4EB 4-Bit D Flip-Flop with Clock Enable, Bus Version
- FD4ES 4-Bit D Flip-Flop with Clock Enable, Single Pin Version
- FD4PB 4-Bit D-Type Flip-Flop with Asynchronous Preset, Bus Version
- FD4PEB 4-Bit D-Type Flip-Flop with Clock Enable and Asynchronous Preset, Bus Version
- FD4PES 4-Bit D-Type Flip-Flop with Clock Enable and Asynchronous Preset, Single Pin Version
- FD4PS 4-Bit D-Type Flip-Flop with Asynchronous Preset, Single Pin Version
- FD4RB 4-Bit D-Type Flip-Flop with Synchronous Reset, Bus Version
- FD4REB 4-Bit D-Type Flip-Flop with Clock Enable and Synchronous Reset, Bus Version

- FD4RES 4-Bit D-Type Flip-Flop with Clock Enable and Synchronous Reset, Single Pin Version
- FD4RS 4-Bit D-Type Flip-Flop with Synchronous Reset, Single Pin Version
- FD4RSB 4-Bit D-Type Flip-Flop with Synchronous Reset and Set, Bus Version
- FD4RSEB 4-Bit D-Type Flip-Flop with Synchronous Reset and Set and Clock Enable, Bus Version
- FD4RSES 4-Bit D-Type Flip-Flop with Synchronous Reset and Set and Clock Enable, Single Pin Version
- FD4RSS 4-Bit D-Type Flip-Flop with Synchronous Reset and Set, Single Pin Version
- FD4S 4-Bit D-Type Flip-Flop, Single Pin Version
- FD4SB 4-Bit D-Type Flip-Flop with Synchronous Set, Bus Version
- FD4SEB 4-Bit D-Type Flip-Flop with Clock Enable and Synchronous Set, Bus Version
- FD4SES 4-Bit D-Type Flip-Flop with Clock Enable and Synchronous Set, Single Pin Version
- FD4SRB 4-Bit D-Type Flip-Flop with Synchronous Set and Reset, Bus Version
- FD4SREB 4-Bit D-Type Flip-Flop with Synchronous Set and Reset and Clock Enable, Bus Version
- FD4SRES 4-Bit D-Type Flip-Flop with Synchronous Set and Reset and Clock Enable, Single Pin Version
- FD4SRS 4-Bit D-Type Flip-Flop with Synchronous Set and Reset, Single Pin Version
- FD4SS 4-Bit D-Type Flip-Flop with Synchronous Set, Single Pin Version
- FD8B 8-Bit D-Type Flip-Flop, Bus Version
- FD8CB 8-Bit D-Type Flip-Flop with Asynchronous Clear, Bus Version
- FD8CEB 8-Bit D-Type Flip-Flop with Clock Enable and Asynchronous Clear, Bus Version
- FD8CES 8-Bit D-Type Flip-Flop with Clock Enable and Asynchronous Clear, Single Pin Version
- FD8CPB 8-Bit D-Type Flip-Flop with Asynchronous Preset and Clear, Bus Version
- FD8CPEB 8-Bit D-Type Flip-Flop with Clock Enable and Asynchronous Preset and Clear, Bus Version
- FD8CPES 8-Bit D-Type Flip-Flop with Clock Enable and Asynchronous Preset and Clear, Single Pin Version
- FD8CPS 8-Bit D-Type Flip-Flop with Asynchronous Preset and Clear, Single Pin Version
- FD8CS 8-Bit D-Type Flip-Flop with Asynchronous Clear, Single Pin Version
- FD8EB 8-Bit D Flip-Flop with Clock Enable, Bus Version
- FD8ES 8-Bit D Flip-Flop with Clock Enable, Single Pin Version
- FD8PB 8-Bit D-Type Flip-Flop with Asynchronous Preset, Bus Version

- FD8PEB 8-Bit D-Type Flip-Flop with Clock Enable and Asynchronous Preset, Bus Version
- FD8PES 8-Bit D-Type Flip-Flop with Clock Enable and Asynchronous Preset, Single Pin Version
- FD8PS 8-Bit D-Type Flip-Flop with Asynchronous Preset, Single Pin Version
- FD8RB 8-Bit D-Type Flip-Flop with Synchronous Reset, Bus Version
- FD8REB 8-Bit D-Type Flip-Flop with Clock Enable and Synchronous Reset, Bus Version
- FD8RES 8-Bit D-Type Flip-Flop with Clock Enable and Synchronous Reset, Single Pin Version
- FD8RS 8-Bit D-Type Flip-Flop with Synchronous Reset, Single Pin Version
- FD8RSB 8-Bit D-Type Flip-Flop with Synchronous Reset and Set, Bus Version
- FD8RSEB 8-Bit D-Type Flip-Flop with Synchronous Reset and Set and Clock Enable, Bus Version
- FD8RSES 8-Bit D-Type Flip-Flop with Synchronous Reset and Set and Clock Enable, Single Pin Version
- FD8RSS 8-Bit D-Type Flip-Flop with Synchronous Reset and Set, Single Pin Version
- FD8S 8-Bit D-Type Flip-Flop, Single Pin Version
- FD8SB 8-Bit D-Type Flip-Flop with Synchronous Set, Bus Version
- FD8SEB 8-Bit D-Type Flip-Flop with Clock Enable and Synchronous Set, Bus Version
- FD8SES 8-Bit D-Type Flip-Flop with Clock Enable and Synchronous Set, Single Pin Version
- FD8SRB 8-Bit D-Type Flip-Flop with Synchronous Set and Reset, Bus Version
- FD8SREB 8-Bit D-Type Flip-Flop with Synchronous Set and Reset and Clock Enable, Bus Version
- FD8SRES 8-Bit D-Type Flip-Flop with Synchronous Set and Reset and Clock Enable, Single Pin Version
- FD8SRS 8-Bit D-Type Flip-Flop with Synchronous Set and Reset, Single Pin Version
- FD8SS 8-Bit D-Type Flip-Flop with Synchronous Set, Single Pin Version
- FD16B 16-Bit D-Type Flip-Flop, Bus Version
- FD16CB 16-Bit D-Type Flip-Flop with Asynchronous Clear, Bus Version
- FD16CEB 16-Bit D-Type Flip-Flop with Clock Enable and Asynchronous Clear, Bus Version
- FD16CES 16-Bit D-Type Flip-Flop with Clock Enable and Asynchronous Clear, Single Pin Version
- FD16CPB 16-Bit D-Type Flip-Flop with Asynchronous Preset and Clear, Bus Version
- FD16CPEB 16-Bit D-Type Flip-Flop with Clock Enable and Asynchronous Preset and Clear, Bus Version

- FD16CPES 16-Bit D-Type Flip-Flop with Clock Enable and Asynchronous Preset and Clear, Single Pin Version
- FD16CPS 16-Bit D-Type Flip-Flop with Asynchronous Preset and Clear, Single Pin Version
- FD16CS 16-Bit D-Type Flip-Flop with Asynchronous Clear, Single Pin Version
- FD16EB 16-Bit D Flip-Flop with Clock Enable, Bus Version
- FD16ES 16-Bit D Flip-Flop with Clock Enable, Single Pin Version
- FD16PB 16-Bit D-Type Flip-Flop with Asynchronous Preset, Bus Version
- FD16PEB 16-Bit D-Type Flip-Flop with Clock Enable and Asynchronous Preset, Bus Version
- FD16PES 16-Bit D-Type Flip-Flop with Clock Enable and Asynchronous Preset, Single Pin Version
- FD16PS 16-Bit D-Type Flip-Flop with Asynchronous Preset, Single Pin Version
- FD16RB 16-Bit D-Type Flip-Flop with Synchronous Reset, Bus Version
- FD16REB 16-Bit D-Type Flip-Flop with Clock Enable and Synchronous Reset, Bus Version
- FD16RES 16-Bit D-Type Flip-Flop with Clock Enable and Synchronous Reset, Single Pin Version
- FD16RS 16-Bit D-Type Flip-Flop with Synchronous Reset, Single Pin Version
- FD16RSB 16-Bit D-Type Flip-Flop with Synchronous Reset and Set, Bus Version
- FD16RSEB 16-Bit D-Type Flip-Flop with Synchronous Reset and Set and Clock Enable, Bus Version
- FD16RSES 16-Bit D-Type Flip-Flop with Synchronous Reset and Set and Clock Enable, Single Pin Version
- FD16RSS 16-Bit D-Type Flip-Flop with Synchronous Reset and Set, Single Pin Version
- FD16S 16-Bit D-Type Flip-Flop, Single Pin Version
- FD16SB 16-Bit D-Type Flip-Flop with Synchronous Set, Bus Version
- FD16SEB 16-Bit D-Type Flip-Flop with Clock Enable and Synchronous Set, Bus Version
- FD16SES 16-Bit D-Type Flip-Flop with Clock Enable and Synchronous Set, Single Pin Version
- FD16SRB 16-Bit D-Type Flip-Flop with Synchronous Set and Reset, Bus Version
- FD16SREB 16-Bit D-Type Flip-Flop with Synchronous Set and Reset and Clock Enable, Bus Version
- FD16SRES 16-Bit D-Type Flip-Flop with Synchronous Set and Reset and Clock Enable, Single Pin Version
- FD16SRS 16-Bit D-Type Flip-Flop with Synchronous Set and Reset, Single Pin Version

- FD16SS 16-Bit D-Type Flip-Flop with Synchronous Set, Single Pin Version
- FD32B 32-Bit D-Type Flip-Flop, Bus Version
- FD32CB 32-Bit D-Type Flip-Flop with Asynchronous Clear, Bus Version
- FD32CEB 32-Bit D-Type Flip-Flop with Clock Enable and Asynchronous Clear, Bus Version
- FD32CPB 32-Bit D-Type Flip-Flop with Asynchronous Preset and Clear, Bus Version
- FD32CPEB 32-Bit D-Type Flip-Flop with Clock Enable and Asynchronous Preset and Clear, Bus Version
- FD32EB 32-Bit D Flip-Flop with Clock Enable, Bus Version
- FD32PB 32-Bit D-Type Flip-Flop with Asynchronous Preset, Bus Version
- FD32PEB 32-Bit D-Type Flip-Flop with Clock Enable and Asynchronous Preset, Bus Version
- FD32RB 32-Bit D-Type Flip-Flop with Synchronous Reset, Bus Version
- FD32REB 32-Bit D-Type Flip-Flop with Clock Enable and Synchronous Reset, Bus Version
- FD32RSB 32-Bit D-Type Flip-Flop with Synchronous Reset and Set, Bus Version
- FD32RSEB 32-Bit D-Type Flip-Flop with Synchronous Reset and Set and Clock Enable, Bus Version
- FD32SB 32-Bit D-Type Flip-Flop with Synchronous Set, Bus Version
- FD32SEB 32-Bit D-Type Flip-Flop with Clock Enable and Synchronous Set, Bus Version
- FD32SRB 32-Bit D-Type Flip-Flop with Synchronous Set and Reset, Bus Version
- FD32SREB 32-Bit D-Type Flip-Flop with Synchronous Set and Reset and Clock Enable, Bus Version
- FD_1 D-Type Negative Edge Flip-Flop
- FDC D-Type Flip-Flop with Asynchronous Clear
- FDC_1 D-Type Negative Edge Flip-Flop with Asynchronous Clear
- FDCE D-Type Flip-Flop with Clock Enable and Asynchronous Clear
- FDCE_1 D-Type Negative Edge Flip-Flop with Clock Enable, Asynchronous Clear and Dual output
- FDCEN D-Type Flip-Flop with Clock Enable, Asynchronous Clear and Inverted and Non-Inverted Outputs
- FDCN D-Type Flip-Flop with Asynchronous Clear and Inverted and Non-Inverted Outputs
- FDCP D-Type Flip-Flop with Asynchronous Preset and Clear
- FDCP_1 D-Type Negative Edge Flip-Flop with Asynchronous Preset and Clear
- FDCPE D-Type Flip-Flop with Clock Enable and Asynchronous Preset and Clear

- FDCPE_1 D-Type Negative Edge Flip-Flop with Clock Enable and Asynchronous Preset and Clear
- FDCPEN D-Type Flip-Flop with Clock Enable, Asynchronous Preset, Clear and Inverted and Non-Inverted Outputs
- FDCPN D-Type Flip-Flop with Asynchronous Preset, Clear and Inverted and Non-Inverted Outputs
- FDE D Flip-Flop with Clock Enable
- FDE_1 D Negative Edge Flip-Flop with Clock Enable
- FDEN D Flip-Flop with Clock Enable and Inverted and Non-Inverted Outputs
- FDN D-Type Flip-Flop with Inverted and Non-Inverted Outputs
- FDP D-Type Flip-Flop with Asynchronous Preset
- FDP_1 D-Type Negative Edge Flip-Flop with Asynchronous Preset
- FDPE D-Type Flip-Flop with Clock Enable and Asynchronous Preset
- FDPE_1 D-Type Flip-Flop with Clock Enable and Asynchronous Preset
- FDPEN D-Type Flip-Flop with Clock Enable and Asynchronous Preset and Inverted and Non-Inverted Outputs
- FDPN D-Type Flip-Flop with Asynchronous Preset and Inverted and Non-Inverted Outputs
- FDR D-Type Flip-Flop with Synchronous Reset
- FDR_1 D-Type Negative Edge Flip-Flop with Synchronous Reset
- FDRE D-Type Flip-Flop with Clock Enable and Synchronous Reset
- FDRE_1 D-Type Negative Edge Flip-Flop with Clock Enable and Synchronous Reset
- FDREN D-Type Flip-Flop with Clock Enable Synchronous Reset and Inverted and Non-Inverted Outputs
- FDRN D-Type Flip-Flop with Synchronous Reset and Inverted and Non-Inverted Outputs
- FDRS D-Type Flip-Flop with Synchronous Reset and Set, Single Pin Version
- FDRS_1 D-Type Negative Edge Flip-Flop with Synchronous Reset and Set
- FDRSE D-Type Flip-Flop with Synchronous Reset and Set and Clock Enable
- FDRSE_1 D-Type Negative Edge Flip-Flop with Synchronous Reset and Set and Clock Enable
- FDRSEN D-Type Flip-Flop with Synchronous Reset and Set, Clock Enable and Inverted and Non-Inverted Outputs
- FDRSN D-Type Flip-Flop with Synchronous Reset, Set and Inverted and Non-Inverted Outputs
- FDS D-Type Flip-Flop with Synchronous Set, Single Pin Version
- FDS_1 D-Type Negative Edge Flip-Flop with Synchronous Set
- FDSE D-Type Flip-Flop with Clock Enable and Synchronous Set

- FDSE_1 D-Type Negative Edge Flip-Flop with Clock Enable and Synchronous Set
- FDSEN D-Type Flip-Flop with Clock Enable Synchronous Set and Inverted and Non-Inverted Outputs
- FDSN D-Type Flip-Flop with Synchronous Set and Inverted and Non-Inverted Outputs
- FDSR D-Type Flip-Flop with Synchronous Set and Reset
- FDSR_1 D-Type Negative Edge Flip-Flop with Synchronous Set and Reset
- FDSRE D-Type Flip-Flop with Synchronous Set and Reset and Clock Enable
- FDSRE_1 D-Type Negative Edge Flip-Flop with Synchronous Set and Reset and Clock Enable
- FDSREN D-Type Flip-Flop with Synchronous Set, Reset, Clock Enable and Inverted and Non-Inverted Outputs
- FDSRN D-Type Flip-Flop with Synchronous Set, Reset and Inverted and Non-Inverted Outputs
- FJKC J-K Flip-Flop with Asynchronous Clear
- FJKC_1 J-K Negative Edge Flip-Flop with Asynchronous Clear
- FJKCE J-K Flip-Flop with Clock Enable and Asynchronous Clear
- FJKCE_1 J-K Negative Edge Flip-Flop with Clock Enable and Asynchronous Clear
- FJKCEN J-K Flip-Flop with Clock Enable, Asynchronous Clear and Inverted and Non-Inverted Outputs
- FJKCN J-K Flip-Flop with Asynchronous Clear and Inverted and Non-Inverted Outputs
- FJKCP J-K Flip-Flop with Asynchronous Clear and Preset
- FJKCP_1 J-K Negative Edge Flip-Flop with Asynchronous Clear and Preset
- FJKCPE J-K Flip-Flop with Asynchronous Clear and Preset and Clock Enable
- FJKCPE_1 J-K Negative Edge Flip-Flop with Asynchronous Clear and Preset and Clock Enable
- FJKCPEN J-K Flip-Flop with Asynchronous Clear, Preset, Clock Enable and Inverted and Non-Inverted Outputs
- FJKCPN J-K Flip-Flop with Asynchronous Clear, Preset and Inverted and Non-Inverted Outputs
- FJKP J-K Flip-Flop with Asynchronous Preset
- FJKP_1 J-K Negative Edge Flip-Flop with Asynchronous Preset
- FJKPE J-K Flip-Flop with Clock Enable and Asynchronous Preset
- FJKPE_1 J-K Negative Edge Flip-Flop with Clock Enable and Asynchronous Preset
- FJKPEN J-K Flip-Flop with Clock Enable, Asynchronous Preset and Inverted and Non-Inverted Outputs
- FJKPN J-K Flip-Flop with Asynchronous Preset and Inverted and Non-Inverted Outputs

- FJKRSE — J-K Flip-Flop with Clock Enable and Synchronous Reset and Set
- FJKRSE_1 — J-K Negative Edge Flip-Flop with Clock Enable and Synchronous Reset and Set
- FJKRSEN — J-K Flip-Flop with Clock Enable, Synchronous Reset and Set and Inverted and Non-Inverted Outputs
- FJKSRE — J-K Flip-Flop with Clock Enable and Synchronous Set and Reset
- FJKSRE_1 — J-K Negative Edge Flip-Flop with Clock Enable and Synchronous Set and Reset
- FJKSREN — J-K Flip-Flop with Clock Enable, Synchronous Set and Reset and Inverted and Non-Inverted Outputs
- FTC — Toggle Flip-Flop with Toggle Enable and Asynchronous Clear
- FTC_1 — Negative Edge Toggle Flip-Flop with Toggle Enable and Asynchronous Clear
- FTCE — Toggle Flip-Flop with Toggle and Clock Enable and Asynchronous Clear
- FTCE_1 — Negative Edge Toggle Flip-Flop with Toggle and Clock Enable and Asynchronous Clear
- FTCEN — Toggle Flip-Flop with Toggle, Clock Enable, Asynchronous Clear and Inverted and Non-Inverted Outputs
- FTCLE — Toggle/Loadable Flip-Flop with Toggle and Clock Enable and Asynchronous Clear
- FTCLE_1 — Toggle/Loadable Negative Edge Flip-Flop with Toggle and Clock Enable and Asynchronous Clear
- FTCLEN — Toggle/Loadable Flip-Flop with Toggle, Clock Enable, Asynchronous Clear and Inverted and Non-Inverted Outputs
- FTCN — Toggle Flip-Flop with Toggle Enable, Asynchronous Clear and Inverted and Non-Inverted Outputs
- FTCP — Toggle Flip-Flop with Toggle Enable and Asynchronous Clear and Preset
- FTCP_1 — Negative Edge Toggle Flip-Flop with Toggle Enable and Asynchronous Clear and Preset
- FTCPE — Toggle Flip-Flop with Toggle and Clock Enable and Asynchronous Clear and Preset
- FTCPE_1 — Negative Edge Toggle Flip-Flop with Toggle and Clock Enable and Asynchronous Clear and Preset
- FTCPEN — Toggle Flip-Flop with Toggle, Clock Enable, Asynchronous Clear and Preset and Inverted and Non-Inverted Outputs
- FTCPLE — Loadable Toggle Flip-Flop with Toggle and Clock Enable and Asynchronous Clear and Preset
- FTCPLE_1 — Loadable Negative Edge Toggle Flip-Flop with Toggle and Clock Enable and Asynchronous Clear and Preset

- FTCPLEN — Loadable Toggle Flip-Flop with Toggle, Clock Enable, Asynchronous Clear and Preset and Inverted and Non-Inverted Outputs
- FTCPN — Toggle Flip-Flop with Toggle Enable, Asynchronous Clear, Preset and Inverted and Non-Inverted Outputs
- FTP — Toggle Flip-Flop with Toggle Enable and Asynchronous Preset
- FTP_1 — Negative Edge Toggle Flip-Flop with Toggle Enable and Asynchronous Preset
- FTPE — Toggle Flip-Flop with Toggle and Clock Enable and Asynchronous Preset
- FTPE_1 — Negative Edge Toggle Flip-Flop with Toggle and Clock Enable and Asynchronous Preset
- FTPEN — Toggle Flip-Flop with Toggle, Clock Enable, Asynchronous Preset and Inverted and Non-Inverted Outputs
- FTPLE — Toggle/Loadable Flip-Flop with Toggle and Clock Enable and Asynchronous Preset
- FTPLE_1 — Toggle/Loadable Negative Edge Flip-Flop with Toggle and Clock Enable and Asynchronous Preset
- FTPLEN — Toggle/Loadable Flip-Flop with Toggle, Clock Enable, Asynchronous Preset and Inverted and Non-Inverted Outputs
- FTPN — Toggle Flip-Flop with Toggle Enable, Asynchronous Preset and Inverted and Non-Inverted Outputs
- FTRSE — Toggle Flip-Flop with Toggle and Clock Enable and Synchronous Reset and Set
- FTRSE_1 — Negative Edge Toggle Flip-Flop with Toggle and Clock Enable and Synchronous Reset and Set
- FTRSEN — Toggle Flip-Flop with Toggle, Clock Enable, Synchronous Reset and Set and Inverted and Non-Inverted Outputs
- FTRSLE — Toggle/Loadable Flip-Flop with Toggle and Clock Enable and Synchronous Reset and Set
- FTRSLE_1 — Toggle/Loadable Negative Edge Flip-Flop with Toggle and Clock Enable and Synchronous Reset and Set
- FTRSLEN — Toggle/Loadable Flip-Flop with Toggle, Clock Enable, Synchronous Reset and Set and Inverted and Non-Inverted Outputs
- FTSRE — Toggle Flip-Flop with Toggle and Clock Enable and Synchronous Set and Reset
- FTSRE_1 — Negative Edge Toggle Flip-Flop with Toggle and Clock Enable and Synchronous Set and Reset
- FTSREN — Toggle Flip-Flop with Toggle, Clock Enable, Synchronous Set and Reset and Inverted and Non-Inverted Outputs
- FTSRLE — Toggle/Loadable Flip-Flop with Toggle and Clock Enable and Synchronous Set and Reset

- FTSRLE_1 Toggle/Loadable Negative Edge Flip-Flop with Toggle and Clock Enable and Synchronous Set and Reset
- FTSRLEN Toggle/Loadable Flip-Flop with Toggle, Clock Enable, Synchronous Set and Reset and Inverted and Non-Inverted Outputs

JTAG

- NEXUS_JTAG_PORT Soft Nexus-Chain Connector

Latch

Various latches are available as follows:

- LD — Transparent Data Latch
- LD2B — 2-Bit Transparent Data Latch, Bus Version
- LD2CEB — 2-Bit Transparent Data Latch with Asynchronous Clear and Gate Enable, Bus Version
- LD2CES — 2-Bit Transparent Data Latch with Asynchronous Clear and Gate Enable, Single Pin Version
- LD2S — 2-Bit Transparent Data Latch, Single Pin Version
- LD3B — 3-Bit Transparent Data Latch, Bus Version
- LD3S — 3-Bit Transparent Data Latch, Single Pin Version
- LD4B — 4-Bit Transparent Data Latch, Bus Version
- LD4CEB — 4-Bit Transparent Data Latch with Asynchronous Clear and Gate Enable, Bus Version
- LD4CES — 4-Bit Transparent Data Latch with Asynchronous Clear and Gate Enable, Single Pin Version
- LD4S — 4-Bit Transparent Data Latch, Single Pin Version
- LD8B — 8-Bit Transparent Data Latch, Bus Version
- LD8CEB — 8-Bit Transparent Data Latch with Asynchronous Clear and Gate Enable, Bus Version
- LD8CES — 8-Bit Transparent Data Latch with Asynchronous Clear and Gate Enable, Single Pin Version
- LD8S — 8-Bit Transparent Data Latch, Single Pin Version
- LD16B — 16-Bit Transparent Data Latch, Bus Version
- LD16CEB — 16-Bit Transparent Data Latch with Asynchronous Clear and Gate Enable, Bus Version
- LD16CES — 16-Bit Transparent Data Latch with Asynchronous Clear and Gate Enable, Single Pin Version
- LD16S — 16-Bit Transparent Data Latch, Single Pin Version
- LD32B — 32-Bit Transparent Data Latch, Bus Version
- LD32CEB — 32-Bit Transparent Data Latch with Asynchronous Clear and Gate Enable, Bus Version
- LD_1 — Transparent Data Latch with Inverted Gate
- LDC — Transparent Data Latch with Asynchronous Clear
- LDC_1 — Transparent Data Latch with Asynchronous Clear and Inverted Gate
- LDCE — Transparent Data Latch with Asynchronous Clear and Gate Enable

- LDCE_1 Transparent Data Latch with Asynchronous Clear, Gate Enable, and Inverted Gate
- LDCP Transparent Data Latch with Asynchronous Clear and Preset
- LDCP_1 Transparent Data Latch with Asynchronous Clear and Preset and Inverted Gate
- LDCPE Transparent Data Latch with Asynchronous Clear and Preset and Gate Enable
- LDCPE_1 Transparent Data Latch with Asynchronous Clear and Preset, Gate Enable, and Inverted Gate
- LDE Transparent Data Latch with Gate Enable
- LDE_1 Transparent Data Latch with Gate Enable and Inverted Gate
- LDP Transparent Data Latch with Asynchronous Preset
- LDP_1 Transparent Data Latch with Asynchronous Preset and Inverted Gate
- LDPE Transparent Data Latch with Asynchronous Preset and Gate Enable
- LDPE_1 Transparent Data Latch with Asynchronous Preset, Gate Enable, and Inverted Gate

Logic Primitive

Basic building block logic primitives are available under the following sub classes:

- AND Gates
- Inverters
- NAND Gates
- NOR Gates
- OR Gates
- Sum of Product
- True/Complement
- XNOR Gates
- XOR Gates

AND Gates

Various inputs and dual output AND Gates are available as follows:

- AND2B 2-Input AND Gate, Bus Version
- AND2DB 2-Input AND/NAND Gate, Bus Version
- AND2DS 2-Input AND/NAND Gate, Single Pin Version
- AND2N1B 2-Input AND Gate with Active Low A Input, Bus Version
- AND2N1S 2-Input AND Gate with Active Low A Input, Single Pin Version
- AND2N2B 2-Input AND Gate with Active Low A and B Inputs, Bus Version
- AND2N2S 2-Input AND Gate with Active Low A and B Inputs, Single Pin Version
- AND2S 2-Input AND Gate, Single Pin Version
- AND3B 3-Input AND Gate, Bus Version
- AND3DB 3-Input AND/NAND Gate, Bus Version
- AND3DS 3-Input AND/NAND Gate, Single Pin Version
- AND3N1B 3-Input AND Gate with Active Low A Input, Bus Version
- AND3N1S 3-Input AND Gate with Active Low A Input, Single Pin Version
- AND3N2B 3-Input AND Gate with Active Low A and B Inputs, Bus Version
- AND3N2S 3-Input AND Gate with Active Low A and B Inputs, Single Pin Version
- AND3N3B 3-Input AND Gate with Active Low A, B and C Inputs, Bus Version
- AND3N3S 3-Input AND Gate with Active Low A, B and C Inputs, Single Pin Version
- AND3S 3-Input AND Gate, Single Pin Version
- AND4B 4-Input AND Gate, Bus Version

- AND4DB 4-Input AND/NAND Gate, Bus Version
- AND4DS 4-Input AND/NAND Gate, Single Pin Version
- AND4N1B 4-Input AND Gate with Active Low A Input, Bus Version
- AND4N1S 4-Input AND Gate with Active Low A Input, Single Pin Version
- AND4N2B 4-Input AND Gate with Active Low A and B Inputs, Bus Version
- AND4N2S 4-Input AND Gate with Active Low A and B Inputs, Single Pin Version
- AND4N3B 4-Input AND Gate with Active Low A, B and C Inputs, Bus Version
- AND4N3S 4-Input AND Gate with Active Low A, B and C Inputs, Single Pin Version
- AND4N4B 4-Input AND Gate with Active Low A, B, C and D Inputs, Bus Version
- AND4N4S 4-Input AND Gate with Active Low A, B, C and D Inputs, Single Pin Version
- AND4S 4-Input AND Gate, Single Pin Version
- AND5B 5-Input AND Gate, Bus Version
- AND5N1B 5-Input AND Gate with Active Low A Input, Bus Version
- AND5N1S 5-Input AND Gate with Active Low A Input, Single Pin Version
- AND5N2B 5-Input AND Gate with Active Low A and B Inputs, Bus Version
- AND5N2S 5-Input AND Gate with Active Low A and B Inputs, Single Pin Version
- AND5N3B 5-Input AND Gate with Active Low A, B and C Inputs, Bus Version
- AND5N3S 5-Input AND Gate with Active Low A, B and C Inputs, Single Pin Version
- AND5N4B 5-Input AND Gate with Active Low A, B, C and D Inputs, Bus Version
- AND5N4S 5-Input AND Gate with Active Low A, B, C and D Inputs, Single Pin Version
- AND5N5B 5-Input AND Gate with Active Low A, B, C, D and E Inputs, Bus Version
- AND5N5S 5-Input AND Gate with Active Low A, B, C, D and E Inputs, Single Pin Version
- AND5S 5-Input AND Gate, Single Pin Version
- AND6B 6-Input AND Gate, Bus Version
- AND6S 6-Input AND Gate, Single Pin Version
- AND7B 7-Input AND Gate, Bus Version
- AND7S 7-Input AND Gate, Single Pin Version
- AND8B 8-Input AND Gate, Bus Version
- AND8S 8-Input AND Gate, Single Pin Version
- AND9B 9-Input AND Gate, Bus Version
- AND9S 9-Input AND Gate, Single Pin Version
- AND12B 12-Input AND Gate, Bus Version
- AND12S 12-Input AND Gate, Single Pin Version
- AND16B 16-Input AND Gate, Bus Version
- AND16S 16-Input AND Gate, Single Pin Version

- AND32B 32-Input AND Gate, Bus Version

Inverters

Various inverters are available as follows:

- INV Inverter
- INV2B 2-Bit Inverter, Bus Version
- INV2S 2-Bit Inverter, Single Pin Version
- INV3B 3-Bit Inverter, Bus Version
- INV3S 3-Bit Inverter, Single Pin Version
- INV4B 4-Bit Inverter, Bus Version
- INV4S 4-Bit Inverter, Single Pin Version
- INV5B 5-Bit Inverter, Bus Version
- INV5S 5-Bit Inverter, Single Pin Version
- INV6B 6-Bit Inverter, Bus Version
- INV6S 6-Bit Inverter, Single Pin Version
- INV7B 7-Bit Inverter, Bus Version
- INV7S 7-Bit Inverter, Single Pin Version
- INV8B 8-Bit Inverter, Bus Version
- INV8S 8-Bit Inverter, Single Pin Version
- INV9B 9-Bit Inverter, Bus Version
- INV9S 9-Bit Inverter, Single Pin Version
- INV10B 10-Bit Inverter, Bus Version
- INV10S 10-Bit Inverter, Single Pin Version
- INV12B 12-Bit Inverter, Bus Version
- INV12S 12-Bit Inverter, Single Pin Version
- INV16B 16-Bit Inverter, Bus Version
- INV16S 16-Bit Inverter, Single Pin Version
- INV32B 32-Bit Inverter, Bus Version

NAND Gates

Various input NAND Gates are available as follows:

- NAND2B 2-Input NAND Gate, Bus Version
- NAND2N1B 2-Input NAND Gate with Active Low A Input, Bus Version
- NAND2N1S 2-Input NAND Gate with Active Low A Input, Single Pin Version

- NAND2N2B 2-Input NAND Gate with Active Low A and B Inputs, Bus Version
- NAND2N2S 2-Input NAND Gate with Active Low A and B Inputs, Single Pin Version
- NAND2S 2-Input NAND Gate, Single Pin Version
- NAND3B 3-Input NAND Gate, Bus Version
- NAND3N1B 3-Input NAND Gate with Active Low A Input, Bus Version
- NAND3N1S 3-Input NAND Gate with Active Low A Input, Single Pin Version
- NAND3N2B 3-Input NAND Gate with Active Low A and B Inputs, Bus Version
- NAND3N2S 3-Input NAND Gate with Active Low A and B Inputs, Single Pin Version
- NAND3N3B 3-Input NAND Gate with Active Low A, B and C Inputs, Bus Version
- NAND3N3S 3-Input NAND Gate with Active Low A, B and C Inputs, Single Pin Version
- NAND3S 3-Input NAND Gate, Single Pin Version
- NAND4B 4-Input NAND Gate, Bus Version
- NAND4N1B 4-Input NAND Gate with Active Low A Input, Bus Version
- NAND4N1S 4-Input NAND Gate with Active Low A Input, Single Pin Version
- NAND4N2B 4-Input NAND Gate with Active Low A and B Inputs, Bus Version
- NAND4N2S 4-Input NAND Gate with Active Low A and B Inputs, Single Pin Version
- NAND4N3B 4-Input NAND Gate with Active Low A, B and C Inputs, Bus Version
- NAND4N3S 4-Input NAND Gate with Active Low A, B and C Inputs, Single Pin Version
- NAND4N4B 4-Input NAND Gate with Active Low A, B, C and D Inputs, Bus Version
- NAND4N4S 4-Input NAND Gate with Active Low A, B, C and D Inputs, Single Pin Version
- NAND4S 4-Input NAND Gate, Single Pin Version
- NAND5B 5-Input NAND Gate, Bus Version
- NAND5N1B 5-Input NAND Gate with Active Low A Input, Bus Version
- NAND5N1S 5-Input NAND Gate with Active Low A Input, Single Pin Version
- NAND5N2B 5-Input NAND Gate with Active Low A and B Inputs, Bus Version
- NAND5N2S 5-Input NAND Gate with Active Low A and B Inputs, Single Pin Version
- NAND5N3B 5-Input NAND Gate with Active Low A, B and C Inputs, Bus Version
- NAND5N3S 5-Input NAND Gate with Active Low A, B and C Inputs, Single Pin Version
- NAND5N4B 5-Input NAND Gate with Active Low A, B, C and D Inputs, Bus Version
- NAND5N4S 5-Input NAND Gate with Active Low A, B, C and D Inputs, Single Pin Version
- NAND5N5B 5-Input NAND Gate with Active Low A, B, C, D and E Inputs, Bus Version
- NAND5N5S 5-Input NAND Gate with Active Low A, B, C, D and E Inputs, Single Pin Version
- NAND5S 5-Input NAND Gate, Single Pin Version
- NAND6B 6-Input NAND Gate, Bus Version

- NAND6S 6-Input NAND Gate, Single Pin Version
- NAND7B 7-Input NAND Gate, Bus Version
- NAND7S 7-Input NAND Gate, Single Pin Version
- NAND8B 8-Input NAND Gate, Bus Version
- NAND8S 8-Input NAND Gate, Single Pin Version
- NAND9B 9-Input NAND Gate, Bus Version
- NAND9S 9-Input NAND Gate, Single Pin Version
- NAND12B 12-Input NAND Gate, Bus Version
- NAND12S 12-Input NAND Gate, Single Pin Version
- NAND16B 16-Input NAND Gate, Bus Version
- NAND16S 16-Input NAND Gate, Single Pin Version
- NAND32B 32-Input NAND Gate, Bus Version

NOR Gates

Various input NOR Gates are available as follows:

- NOR2B 2-Input NOR Gate, Bus Version
- NOR2N1B 2-Input NOR Gate with Active Low A Input, Bus Version
- NOR2N1S 2-Input NOR Gate with Active Low A Input, Single Pin Version
- NOR2N2B 2-Input NOR Gate with Active Low A and B Inputs, Bus Version
- NOR2N2S 2-Input NOR Gate with Active Low A and B Inputs, Single Pin Version
- NOR2S 2-Input NOR Gate, Single Pin Version
- NOR3B 3-Input NOR Gate, Bus Version
- NOR3N1B 3-Input NOR Gate with Active Low A Input, Bus Version
- NOR3N1S 3-Input NOR Gate with Active Low A Input, Single Pin Version
- NOR3N2B 3-Input NOR Gate with Active Low A and B Inputs, Bus Version
- NOR3N2S 3-Input NOR Gate with Active Low A and B Inputs, Single Pin Version
- NOR3N3B 3-Input NOR Gate with Active Low A, B and C Inputs, Bus Version
- NOR3N3S 3-Input NOR Gate with Active Low A, B and C Inputs, Single Pin Version
- NOR3S 3-Input NOR Gate, Single Pin Version
- NOR4B 4-Input NOR Gate, Bus Version
- NOR4N1B 4-Input NOR Gate with Active Low A Input, Bus Version
- NOR4N1S 4-Input NOR Gate with Active Low A Input, Single Pin Version
- NOR4N2B 4-Input NOR Gate with Active Low A and B Inputs, Bus Version
- NOR4N2S 4-Input NOR Gate with Active Low A and B Inputs, Single Pin Version
- NOR4N3B 4-Input NOR Gate with Active Low A, B and C Inputs, Bus Version

- NOR4N3S 4-Input NOR Gate with Active Low A, B and C Inputs, Single Pin Version
- NOR4N4B 4-Input NOR Gate with Active Low A, B, C and D Inputs, Bus Version
- NOR4N4S 4-Input NOR Gate with Active Low A, B, C and D Inputs, Single Pin Version
- NOR4S 4-Input NOR Gate, Single Pin Version
- NOR5B 5-Input NOR Gate, Bus Version
- NOR5N1B 5-Input NOR Gate with Active Low A Input, Bus Version
- NOR5N1S 5-Input NOR Gate with Active Low A Input, Single Pin Version
- NOR5N2B 5-Input NOR Gate with Active Low A and B Inputs, Bus Version
- NOR5N2S 5-Input NOR Gate with Active Low A and B Inputs, Single Pin Version
- NOR5N3B 5-Input NOR Gate with Active Low A, B and C Inputs, Bus Version
- NOR5N3S 5-Input NOR Gate with Active Low A, B and C Inputs, Single Pin Version
- NOR5N4B 5-Input NOR Gate with Active Low A, B, C and D Inputs, Bus Version
- NOR5N4S 5-Input NOR Gate with Active Low A, B, C and D Inputs, Single Pin Version
- NOR5N5B 5-Input NOR Gate with Active Low A, B, C, D and E Inputs, Bus Version
- NOR5N5S 5-Input NOR Gate with Active Low A, B, C, D and E Inputs, Single Pin Version
- NOR5S 5-Input NOR Gate, Single Pin Version
- NOR6B 6-Input NOR Gate, Bus Version
- NOR6S 6-Input NOR Gate, Single Pin Version
- NOR7B 7-Input NOR Gate, Bus Version
- NOR7S 7-Input NOR Gate, Single Pin Version
- NOR8B 8-Input NOR Gate, Bus Version
- NOR8S 8-Input NOR Gate, Single Pin Version
- NOR9B 9-Input NOR Gate, Bus Version
- NOR9S 9-Input NOR Gate, Single Pin Version
- NOR12B 12-Input NOR Gate, Bus Version
- NOR12S 12-Input NOR Gate, Single Pin Version
- NOR16B 16-Input NOR Gate, Bus Version
- NOR16S 12-Input NOR Gate, Single Pin Version
- NOR32B 32-Input NOR Gate, Bus Version

OR Gates

Various input OR Gates are available as follows:

- OR2B 2-Input OR Gate, Bus Version
- OR2DB 2-Input OR/NOR Gate, Bus Version

- OR2DS 2-Input OR/NOR Gate, Single Pin Version
- OR2N1B 2-Input OR Gate with Active Low A Input, Bus Version
- OR2N1S 2-Input OR Gate with Active Low A Input, Single Pin Version
- OR2N2B 2-Input OR Gate with Active Low A and B Inputs, Bus Version
- OR2N2S 2-Input OR Gate with Active Low A and B Inputs, Single Pin Version
- OR2S 2-Input OR Gate, Single Pin Version
- OR3B 3-Input OR Gate, Bus Version
- OR3DB 3-Input OR/NOR Gate, Bus Version
- OR3DS 3-Input OR/NOR Gate, Single Pin Version
- OR3N1B 3-Input OR Gate with Active Low A Input, Bus Version
- OR3N1S 3-Input OR Gate with Active Low A Input, Single Pin Version
- OR3N2B 3-Input OR Gate with Active Low A and B Inputs, Bus Version
- OR3N2S 3-Input OR Gate with Active Low A and B Inputs, Single Pin Version
- OR3N3B 3-Input OR Gate with Active Low A, B and C Inputs, Bus Version
- OR3N3S 3-Input OR Gate with Active Low A, B and C Inputs, Single Pin Version
- OR3S 3-Input OR Gate, Single Pin Version
- OR4B 4-Input OR Gate, Bus Version
- OR4DB 4-Input OR/NOR Gate, Bus Version
- OR4DS 4-Input OR/NOR Gate, Single Pin Version
- OR4N1B 4-Input OR Gate with Active Low A Input, Bus Version
- OR4N1S 4-Input OR Gate with Active Low A Input, Single Pin Version
- OR4N2B 4-Input OR Gate with Active Low A and B Inputs, Bus Version
- OR4N2S 4-Input OR Gate with Active Low A and B Inputs, Single Pin Version
- OR4N3B 4-Input OR Gate with Active Low A, B and C Inputs, Bus Version
- OR4N3S 4-Input OR Gate with Active Low A, B and C Inputs, Single Pin Version
- OR4N4B 4-Input OR Gate with Active Low A, B, C and D Inputs, Bus Version
- OR4N4S 4-Input OR Gate with Active Low A, B, C and D Inputs, Single Pin Version
- OR4S 4-Input OR Gate, Single Pin Version
- OR5B 5-Input OR Gate, Bus Version
- OR5N1B 5-Input OR Gate with Active Low A Input, Bus Version
- OR5N1S 5-Input OR Gate with Active Low A Input, Single Pin Version
- OR5N2B 5-Input OR Gate with Active Low A and B Inputs, Bus Version
- OR5N2S 5-Input OR Gate with Active Low A and B Inputs, Single Pin Version
- OR5N3B 5-Input OR Gate with Active Low A, B and C Inputs, Bus Version
- OR5N3S 5-Input OR Gate with Active Low A, B and C Inputs, Single Pin Version
- OR5N4B 5-Input OR Gate with Active Low A, B, C and D Inputs, Bus Version

- OR5N4S 5-Input OR Gate with Active Low A, B, C and D Inputs, Single Pin Version
- OR5N5B 5-Input OR Gate with Active Low A, B, C, D and E Inputs, Bus Version
- OR5N5S 5-Input OR Gate with Active Low A, B, C, D and E Inputs, Single Pin Version
- OR5S 5-Input OR Gate, Single Pin Version
- OR6B 6-Input OR Gate, Bus Version
- OR6S 6-Input OR Gate, Single Pin Version
- OR7B 7-Input OR Gate, Bus Version
- OR7S 7-Input OR Gate, Single Pin Version
- OR8B 8-Input OR Gate, Bus Version
- OR8S 8-Input OR Gate, Single Pin Version
- OR9B 9-Input OR Gate, Bus Version
- OR9S 9-Input OR Gate, Single Pin Version
- OR12B 12-Input OR Gate, Bus Version
- OR12S 12-Input OR Gate, Single Pin Version
- OR16B 16-Input OR Gate, Bus Version
- OR16S 16-Input OR Gate, Single Pin Version
- OR32B 32-Input OR Gate, Bus Version

Sum of Product

Sum of product components are available as follows:

- SOP2_2B Sum of Product, two 2-inputs AND-OR-INVERT Gates Combination, Bus Version
- SOP2_2S Sum of Product, two 2-inputs AND-OR-INVERT Gates Combination, Single Pin Version
- SOP2_3B Sum of Product, two 3-inputs AND-OR-INVERT Gates Combination, Bus Version
- SOP2_3S Sum of Product, two 3-inputs AND-OR-INVERT Gates Combination, Single Pin Version
- SOP2_4B Sum of Product, two 4-Inputs AND-OR-INVERT Gates Combination, Bus Version
- SOP2_4S Sum of Product, two 4-Inputs AND-OR-INVERT Gates Combination, Single Pin Version
- SOP4_2B Sum of Product, four 2-inputs AND-OR-INVERT Gates Combination, Bus Version
- SOP4_2S Sum of Product, four 2-inputs AND-OR-INVERT Gates Combination, Single Pin Version

True/Complement

True/Complement

- TCZO — True/Complement, Zero/One Element

XNOR Gates

Various input XNOR Gates are available as follows:

- XNOR2B — 2-Input Exclusive-NOR Gate, Bus Version
- XNOR2N1B — 2-Input Exclusive-NOR Gate with Active Low A Input, Bus Version
- XNOR2N1S — 2-Input Exclusive-NOR Gate with Active Low A Input, Single Pin Version
- XNOR2N2B — 2-Input Exclusive-NOR Gate with Active Low A and B Inputs, Bus Version
- XNOR2N2S — 2-Input Exclusive-NOR Gate with Active Low A and B Inputs, Single Pin Version
- XNOR2S — 2-Input Exclusive-NOR Gate, Single Pin Version
- XNOR3B — 3-Input Exclusive-NOR Gate, Bus Version
- XNOR3N1B — 3-Input Exclusive-NOR Gate with Active Low A Input, Bus Version
- XNOR3N1S — 3-Input Exclusive-NOR Gate with Active Low A Input, Single Pin Version
- XNOR3N2B — 3-Input Exclusive-NOR Gate with Active Low A and B Inputs, Bus Version
- XNOR3N2S — 3-Input Exclusive-NOR Gate with Active Low A and B Inputs, Single Pin Version
- XNOR3N3B — 3-Input Exclusive-NOR Gate with Active Low A, B and C Inputs, Bus Version
- XNOR3N3S — 3-Input Exclusive-NOR Gate with Active Low A, B and C Inputs, Single Pin Version
- XNOR3S — 3-Input Exclusive-NOR Gate, Single Pin Version
- XNOR4B — 4-Input Exclusive-NOR Gate, Bus Version
- XNOR4N1B — 4-Input Exclusive-NOR Gate with Active Low A Input, Bus Version
- XNOR4N1S — 4-Input Exclusive-NOR Gate with Active Low A Input, Single Pin Version
- XNOR4N2B — 4-Input Exclusive-NOR Gate with Active Low A and B Inputs, Bus Version
- XNOR4N2S — 4-Input Exclusive-NOR Gate with Active Low A and B Inputs, Single Pin Version
- XNOR4N3B — 4-Input Exclusive-NOR Gate with Active Low A, B and C Inputs, Bus Version
- XNOR4N3S — 4-Input Exclusive-NOR Gate with Active Low A, B and C Inputs, Single Pin Version

- XNOR4N4B 4-Input Exclusive-NOR Gate with Active Low A, B, C and D Inputs, Bus Version
- XNOR4N4S 4-Input Exclusive-NOR Gate with Active Low A, B, C and D Inputs, Single Pin Version
- XNOR4S 4-Input Exclusive-NOR Gate, Single Pin Version
- XNOR5B 5-Input Exclusive-NOR Gate, Bus Version
- XNOR5N1B 5-Input Exclusive-NOR Gate with Active Low A Input, Bus Version
- XNOR5N1S 5-Input Exclusive-NOR Gate with Active Low A Input, Single Pin Version
- XNOR5N2B 5-Input Exclusive-NOR Gate with Active Low A and B Inputs, Bus Version
- XNOR5N2S 5-Input Exclusive-NOR Gate with Active Low A and B Inputs, Single Pin Version
- XNOR5N3B 5-Input Exclusive-NOR Gate with Active Low A, B and C Inputs, Bus Version
- XNOR5N3S 5-Input Exclusive-NOR Gate with Active Low A, B and C Inputs, Single Pin Version
- XNOR5N4B 5-Input Exclusive-NOR Gate with Active Low A, B, C and D Inputs, Bus Version
- XNOR5N4S 5-Input Exclusive-NOR Gate with Active Low A, B, C and D Inputs, Single Pin Version
- XNOR5N5B 5-Input Exclusive-NOR Gate with Active Low A, B, C, D and E Inputs, Bus Version
- XNOR5N5S 5-Input Exclusive-NOR Gate with Active Low A, B, C, D and E Inputs, Single Pin Version
- XNOR5S 5-Input Exclusive-NOR Gate, Single Pin Version
- XNOR6B 6-Input Exclusive-NOR Gate, Bus Version
- XNOR6S 6-Input Exclusive-NOR Gate, Single Pin Version
- XNOR7B 7-Input Exclusive-NOR Gate, Bus Version
- XNOR7S 7-Input Exclusive-NOR Gate, Single Pin Version
- XNOR8B 8-Input Exclusive-NOR Gate, Bus Version
- XNOR8S 8-Input Exclusive-NOR Gate, Single Pin Version
- XNOR9B 9-Input Exclusive-NOR Gate, Bus Version
- XNOR9S 9-Input Exclusive-NOR Gate, Single Pin Version
- XNOR12B 12-Input Exclusive-NOR Gate, Bus Version
- XNOR12S 12-Input Exclusive-NOR Gate, Single Pin Version
- XNOR16B 16-Input Exclusive-NOR Gate, Bus Version
- XNOR16S 16-Input Exclusive-NOR Gate, Single Pin Version
- XNOR32B 32-Input Exclusive-NOR Gate, Bus Version

XOR Gates

Various input XOR Gates are available as follows:

- XOR2B 2-Input Exclusive-OR Gate, Bus Version
- XOR2N1B 2-Input Exclusive-OR Gate with Active Low A Input, Bus Version
- XOR2N1S 2-Input Exclusive-OR Gate with Active Low A Input, Single Pin Version
- XOR2N2B 2-Input Exclusive-OR Gate with Active Low A and B Inputs, Bus Version
- XOR2N2S 2-Input Exclusive-OR Gate with Active Low A and B Inputs, Single Pin Version
- XOR2S 2-Input Exclusive-OR Gate, Single Pin Version
- XOR3B 3-Input Exclusive-OR Gate, Bus Version
- XOR3N1B 3-Input Exclusive-OR Gate with Active Low A Input, Bus Version
- XOR3N1S 3-Input Exclusive-OR Gate with Active Low A Input, Single Pin Version
- XOR3N2B 3-Input Exclusive-OR Gate with Active Low A and B Inputs, Bus Version
- XOR3N2S 3-Input Exclusive-OR Gate with Active Low A and B Inputs, Single Pin Version
- XOR3N3B 3-Input Exclusive-OR Gate with Active Low A, B and C Inputs, Bus Version
- XOR3N3S 3-Input Exclusive-OR Gate with Active Low A, B and C Inputs, Single Pin Version
- XOR3S 3-Input Exclusive-OR Gate, Single Pin Version
- XOR4B 4-Input Exclusive-OR Gate, Bus Version
- XOR4N1B 4-Input Exclusive-OR Gate with Active Low A Input, Bus Version
- XOR4N1S 4-Input Exclusive-OR Gate with Active Low A Input, Single Pin Version
- XOR4N2B 4-Input Exclusive-OR Gate with Active Low A and B Inputs, Bus Version
- XOR4N2S 4-Input Exclusive-OR Gate with Active Low A and B Inputs, Single Pin Version
- XOR4N3B 4-Input Exclusive-OR Gate with Active Low A, B and C Inputs, Bus Version
- XOR4N3S 4-Input Exclusive-OR Gate with Active Low A, B and C Inputs, Single Pin Version
- XOR4N4B 4-Input Exclusive-OR Gate with Active Low A, B, C and D Inputs, Bus Version
- XOR4N4S 4-Input Exclusive-OR Gate with Active Low A, B, C and D Inputs, Single Pin Version
- XOR4S 4-Input Exclusive-OR Gate, Single Pin Version
- XOR5B 5-Input Exclusive-OR Gate, Bus Version
- XOR5N1B 5-Input Exclusive-OR Gate with Active Low A Input, Bus Version
- XOR5N1S 5-Input Exclusive-OR Gate with Active Low A Input, Single Pin Version
- XOR5N2B 5-Input Exclusive-OR Gate with Active Low A and B Inputs, Bus Version

- XOR5N2S 5-Input Exclusive-OR Gate with Active Low A and B Inputs, Single Pin Version
- XOR5N3B 5-Input Exclusive-OR Gate with Active Low A, B and C Inputs, Bus Version
- XOR5N3S 5-Input Exclusive-OR Gate with Active Low A, B and C Inputs, Single Pin Version
- XOR5N4B 5-Input Exclusive-OR Gate with Active Low A, B, C and D Inputs, Bus Version
- XOR5N4S 5-Input Exclusive-OR Gate with Active Low A, B, C and D Inputs, Single Pin Version
- XOR5N5B 5-Input Exclusive-OR Gate with Active Low A, B, C, D and E Inputs, Bus Version
- XOR5N5S 5-Input Exclusive-OR Gate with Active Low A, B, C, D and E Inputs, Single Pin Version
- XOR5S 5-Input Exclusive-OR Gate, Single Pin Version
- XOR6B 6-Input Exclusive-OR Gate, Bus Version
- XOR6S 6-Input Exclusive-OR Gate, Single Pin Version
- XOR7B 7-Input Exclusive-OR Gate, Bus Version
- XOR7S 7-Input Exclusive-OR Gate, Single Pin Version
- XOR8B 8-Input Exclusive-OR Gate, Bus Version
- XOR8S 8-Input Exclusive-OR Gate, Single Pin Version
- XOR9B 9-Input Exclusive-OR Gate, Bus Version
- XOR9S 9-Input Exclusive-OR Gate, Single Pin Version
- XOR12B 12-Input Exclusive-OR Gate, Bus Version
- XOR12S 12-Input Exclusive-OR Gate, Single Pin Version
- XOR16B 16-Input Exclusive-OR Gate, Bus Version
- XOR16S 16-Input Exclusive-OR Gate, Single Pin Version
- XOR32B 32-Input Exclusive-OR Gate, Bus Version

Memory

Various memory components are available as follows:

- RAMD — Dual Port Random Access Memory
- RAMDB — Dual Port Random Access Memory, Byte Write Enable
- RAMDE — Dual Port Random Access Memory With Enable
- RAMDEB — Dual Port Random Access Memory With Enable, Byte Write Enable
- RAMDR — Dual Port Random Access Memory with Reset
- RAMDRB — Dual Port Random Access Memory with Reset, Byte Write Enable
- RAMDRE — Dual Port Random Access Memory With Enable and Reset
- RAMDREB — Dual Port Random Access Memory With Enable and Reset, Byte Write Enable
- RAMS — Single Port Random Access Memory
- RAMSB — Single Port Random Access Memory, Byte Write Enable
- RAMSE — Single Port Random Access Memory With Enable
- RAMSEB — Single Port Random Access Memory With Enable, Byte Write Enable
- RAMSR — Single Port Random Access Memory with Reset
- RAMSRB — Single Port Random Access Memory with Reset, Byte Write Enable
- RAMSRE — Single Port Random Access Memory With Enable and Reset
- RAMSREB — Single Port Random Access Memory With Enable and Reset, Byte Write Enable
- ROMD — Dual Port Read Only Memory
- ROMDE — Dual Port Read Only Memory With Enable
- ROMDR — Dual Port Read Only Memory with Reset
- ROMDRE — Dual Port Read Only Memory With Enable and Reset
- ROMS — Single Port Read Only Memory
- ROMSE — Single Port Read Only Memory With Enable
- ROMSR — Single Port Read Only Memory with Reset
- ROMSRE — Single Port Read Only Memory With Enable and Reset

Multiplexer

Various Multiplexers and De-Multiplexers are available as follows:

- M1_B2S1 1x2-Bit Bus to 1x1-Single Wire Multiplexer
- M1_B2S1E 1x2-Bit Bus to 1x1-Single Wire Multiplexer With Enable
- M1_B4S1 1x4-Bit Bus to 1x1-Single Wire Multiplexer
- M1_B4S1_SB 1x4-Bit Bus to 1x1-Single Wire Multiplexer With Bus Version Select
- M1_B4S1E 1x4-Bit Bus to 1x1-Single Wire Multiplexer With Enable
- M1_B4S1E_SB 1x4-Bit Bus to 1x1-Single Wire Multiplexer With Enable With Bus Version Select
- M1_B8S1 1x8-Bit Bus to 1x1-Single Wire Multiplexer
- M1_B8S1_SB 1x8-Bit Bus to 1x1-Single Wire Multiplexer With Bus Version Select
- M1_B8S1E 1x8-Bit Bus to 1x1-Single Wire Multiplexer With Enable
- M1_B8S1E_SB 1x8-Bit Bus to 1x1-Single Wire Multiplexer With Enable With Bus Version Select
- M1_B16S1 1x16-Bit Bus to 1x1-Single Wire Multiplexer
- M1_B16S1_SB 1x16-Bit Bus to 1x1-Single Wire Multiplexer With Bus Version Select
- M1_B16S1E 1x16-Bit Bus to 1x1-Single Wire Multiplexer With Enable
- M1_B16S1E_SB 1x16-Bit Bus to 1x1-Single Wire Multiplexer With Enable and Bus Version Select
- M1_S1B2 1x1-Single Wire to 1x2-Bit Bus Multiplexer (Demultiplex)
- M1_S1B2E 1x1-Single Wire to 1x2-Bit Bus Multiplexer (Demultiplex) With Enable
- M1_S1B4 1x1-Single Wire to 1x4-Bit Bus Multiplexer (Demultiplex)
- M1_S1B4_SB 1x1-Single Wire to 1x4-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M1_S1B4E 1x1-Single Wire to 1x4-Bit Bus Multiplexer (Demultiplex) With Enable
- M1_S1B4E_SB 1x1-Single Wire to 1x4-Bit Bus Multiplexer (Demultiplex) With Enable With Bus Version Select
- M1_S1B8 1x1-Single Wire to 1x8-Bit Bus Multiplexer (Demultiplex)
- M1_S1B8_SB 1x1-Single Wire to 1x8-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M1_S1B8E 1x1-Single Wire to 1x8-Bit Bus Multiplexer (Demultiplex) With Enable
- M1_S1B8E_SB 1x1-Single Wire to 1x8-Bit Bus Multiplexer (Demultiplex) With Enable With Bus Version Select
- M1_S1B16 1x1-Single Wire to 1x16-Bit Bus Multiplexer (Demultiplex)
- M1_S1B16_SB 1x1-Single Wire to 1x16-Bit Bus Multiplexer (Demultiplex) With Bus Version Select

- M1_S1B16E 1x1-Single Wire to 1x16-Bit Bus Multiplexer (Demultiplex) With Enable
- M1_S1B16E_SB 1x1-Single Wire to 1x16-Bit Bus Multiplexer (Demultiplex) With Enable and Bus Version Select
- M1_S1S2 1x1-Single Wire to 2x1-Single Wire Multiplexer (Demultiplex)
- M1_S1S2E 1x1-Single Wire to 2x1-Single Wire Multiplexer (Demultiplex) With Enable
- M1_S1S4 1x1-Single Wire to 4x1-Single Wire Multiplexer (Demultiplex)
- M1_S1S4_SB 1x1-Single Wire to 4x1-Single Wire Multiplexer (Demultiplex) With Bus Version Select
- M1_S1S4E 1x1-Single Wire to 4x1-Single Wire Multiplexer (Demultiplex) With Enable
- M1_S1S4E_SB 1x1-Single Wire to 4x1-Single Wire Multiplexer (Demultiplex) With Enable With Bus Version Select
- M1_S1S8 1x1-Single Wire to 8x1-Single Wire Multiplexer (Demultiplex)
- M1_S1S8_SB 1x1-Single Wire to 8x1-Single Wire Multiplexer (Demultiplex) With Bus Version Select
- M1_S1S8E 1x1-Single Wire to 8x1-Single Wire Multiplexer (Demultiplex) With Enable
- M1_S1S8E_SB 1x1-Single Wire to 8x1-Single Wire Multiplexer (Demultiplex) With Enable With Bus Version Select
- M1_S1S16 1x1-Single Wire to 16x1-Single Wire Multiplexer (Demultiplex)
- M1_S1S16_SB 1x1-Single Wire to 16x1-Single Wire Multiplexer (Demultiplex) With Bus Version Select
- M1_S1S16E 1x1-Single Wire to 16x1-Single Wire Multiplexer (Demultiplex) With Enable
- M1_S1S16E_SB 1x1-Single Wire to 16x1-Single Wire Multiplexer (Demultiplex) With Enable and Bus Version Select
- M1_S2S1 2x1-Single Wire to 1x1-Single Wire Multiplexer
- M1_S2S1E 2x1-Single Wire to 1x1-Single Wire Multiplexer With Enable
- M1_S4S1 4x1-Single Wire to 1x1-Single Wire Multiplexer
- M1_S4S1_SB 4x1-Single Wire to 1x1-Single Wire Multiplexer With Bus Version Select
- M1_S4S1E 4x1-Single Wire to 1x1-Single Wire Multiplexer With Enable
- M1_S4S1E_SB 4x1-Single Wire to 1x1-Single Wire Multiplexer With Enable With Bus Version Select
- M1_S8S1 8x1-Single Wire to 1x1-Single Wire Multiplexer
- M1_S8S1_SB 8x1-Single Wire to 1x1-Single Wire Multiplexer With Bus Version Select
- M1_S8S1E 8x1-Single Wire to 1x1-Single Wire Multiplexer With Enable
- M1_S8S1E_SB 8x1-Single Wire to 1x1-Single Wire Multiplexer With Enable With Bus Version Select
- M1_S16S1 16x1-Single Wire to 1x1-Single Wire Multiplexer
- M1_S16S1_SB 16x1-Single Wire to 1x1-Single Wire Multiplexer With Bus Version Select
- M1_S16S1E 16x1-Single Wire to 1x1-Single Wire Multiplexer With Enable

- M1_S16S1E_SB 16x1-Single Wire to 1x1-Single Wire Multiplexer With Enable and Bus Version Select
- M2_B1B2 1x2-Bit Bus to 2x2-Bit Bus Multiplexer (Demultiplex)
- M2_B1B2E 1x2-Bit Bus to 2x2-Bit Bus Multiplexer (Demultiplex) With Enable
- M2_B1B4 1x2-Bit Bus to 4x2-Bit Bus Multiplexer (Demultiplex)
- M2_B1B4_SB 1x2-Bit Bus to 4x2-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M2_B1B4E 1x2-Bit Bus to 4x2-Bit Bus Multiplexer (Demultiplex) With Enable
- M2_B1B4E_SB 1x2-Bit Bus to 4x2-Bit Bus Multiplexer (Demultiplex) With Enable With Bus Version Select
- M2_B1B8 1x2-Bit Bus to 8x2-Bit Bus Multiplexer (Demultiplex)
- M2_B1B8_SB 1x2-Bit Bus to 8x2-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M2_B1B8E 1x2-Bit Bus to 8x2-Bit Bus Multiplexer (Demultiplex) With Enable
- M2_B1B8E_SB 1x2-Bit Bus to 8x2-Bit Bus Multiplexer (Demultiplex) With Enable With Bus Version Select
- M2_B1B16 1x2-Bit Bus to 16x2-Bit Bus Multiplexer (Demultiplex)
- M2_B1B16_SB 1x2-Bit Bus to 16x2-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M2_B1B16E 1x2-Bit Bus to 16x2-Bit Bus Multiplexer (Demultiplex) With Enable
- M2_B1B16E_SB 1x2-Bit Bus to 16x2-Bit Bus Multiplexer (Demultiplex) With Enable and Bus Version Select
- M2_B2B1 2x2-Bit Bus to 1x2-Bit Bus Multiplexer
- M2_B2B1E 2x2-Bit Bus to 1x2-Bit Bus Multiplexer With Enable
- M2_B4B1 4x2-Bit Bus to 1x2-Bit Bus Multiplexer
- M2_B4B1_SB 4x2-Bit Bus to 1x2-Bit Bus Multiplexer With Bus Version Select
- M2_B4B1E 4x2-Bit Bus to 1x2-Bit Bus Multiplexer With Enable
- M2_B4B1E_SB 4x2-Bit Bus to 1x2-Bit Bus Multiplexer With Enable With Bus Version Select
- M2_B8B1 8x2-Bit Bus to 1x2-Bit Bus Multiplexer
- M2_B8B1_SB 8x2-Bit Bus to 1x2-Bit Bus Multiplexer With Bus Version Select
- M2_B8B1E 8x2-Bit Bus to 1x2-Bit Bus Multiplexer With Enable
- M2_B8B1E_SB 8x2-Bit Bus to 1x2-Bit Bus Multiplexer With Enable With Bus Version Select
- M2_B16B1 16x2-Bit Bus to 1x2-Bit Bus Multiplexer
- M2_B16B1_SB 16x2-Bit Bus to 1x2-Bit Bus Multiplexer With Bus Version Select
- M2_B16B1E 16x2-Bit Bus to 1x2-Bit Bus Multiplexer With Enable
- M2_B16B1E_SB 16x2-Bit Bus to 1x2-Bit Bus Multiplexer With Enable and Bus Version Select
- M2_S1S2 1x2-Single Wire to 2x2-Single Wire Multiplexer (Demultiplex)
- M2_S1S2E 1x2-Single Wire to 2x2-Single Wire Multiplexer (Demultiplex) With Enable

- M2_S1S4 1x2-Single Wire to 4x2-Single Wire Multiplexer (Demultiplex)
- M2_S1S4_SB 1x2-Single Wire to 4x2-Single Wire Multiplexer (Demultiplex) With Bus Version Select
- M2_S1S4E 1x2-Single Wire to 4x2-Single Wire Multiplexer (Demultiplex) With Enable
- M2_S1S4E_SB 1x2-Single Wire to 4x2-Single Wire Multiplexer (Demultiplex) With Enable With Bus Version Select
- M2_S1S8 1x2-Single Wire to 8x2-Single Wire Multiplexer (Demultiplex)
- M2_S1S8_SB 1x2-Single Wire to 8x2-Single Wire Multiplexer (Demultiplex) With Bus Version Select
- M2_S1S8E 1x2-Single Wire to 8x2-Single Wire Multiplexer (Demultiplex) With Enable
- M2_S1S8E_SB 1x2-Single Wire to 8x2-Single Wire Multiplexer (Demultiplex) With Enable With Bus Version Select
- M2_S2S1 2x2-Single Wire to 1x2-Single Wire Multiplexer
- M2_S2S1E 2x2-Single Wire to 1x2-Single Wire Multiplexer With Enable
- M2_S4S1 4x2-Single Wire to 1x2-Single Wire Multiplexer
- M2_S4S1_SB 4x2-Single Wire to 1x2-Single Wire Multiplexer With Bus Version Select
- M2_S4S1E 4x2-Single Wire to 1x2-Single Wire Multiplexer With Enable
- M2_S4S1E_SB 4x2-Single Wire to 1x2-Single Wire Multiplexer With Enable With Bus Version Select
- M2_S8S1 8x2-Single Wire to 1x2-Single Wire Multiplexer
- M2_S8S1_SB 8x2-Single Wire to 1x2-Single Wire Multiplexer With Bus Version Select
- M2_S8S1E 8x2-Single Wire to 1x2-Single Wire Multiplexer With Enable
- M2_S8S1E_SB 8x2-Single Wire to 1x2-Single Wire Multiplexer With Enable With Bus Version Select
- M3_B1B2 1x3-Bit Bus to 2x3-Bit Bus Multiplexer (Demultiplex)
- M3_B1B2E 1x3-Bit Bus to 2x3-Bit Bus Multiplexer (Demultiplex) With Enable
- M3_B1B4 1x3-Bit Bus to 4x3-Bit Bus Multiplexer (Demultiplex)
- M3_B1B4_SB 1x3-Bit Bus to 4x3-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M3_B1B4E 1x3-Bit Bus to 4x3-Bit Bus Multiplexer (Demultiplex) With Enable
- M3_B1B4E_SB 1x3-Bit Bus to 4x3-Bit Bus Multiplexer (Demultiplex) With Enable With Bus Version Select
- M3_B1B8 1x3-Bit Bus to 8x3-Bit Bus Multiplexer (Demultiplex)
- M3_B1B8_SB 1x3-Bit Bus to 8x3-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M3_B1B8E 1x3-Bit Bus to 8x3-Bit Bus Multiplexer (Demultiplex) With Enable
- M3_B1B8E_SB 1x3-Bit Bus to 8x3-Bit Bus Multiplexer (Demultiplex) With Enable With Bus Version Select

- M3_B1B16 1x3-Bit Bus to 16x3-Bit Bus Multiplexer (Demultiplex)
- M3_B1B16_SB 1x3-Bit Bus to 16x3-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M3_B1B16E 1x3-Bit Bus to 16x3-Bit Bus Multiplexer (Demultiplex) With Enable
- M3_B1B16E_SB 1x3-Bit Bus to 16x3-Bit Bus Multiplexer (Demultiplex) With Enable and Bus Version Select
- M3_B2B1 2x3-Bit Bus to 1x3-Bit Bus Multiplexer
- M3_B2B1E 2x3-Bit Bus to 1x3-Bit Bus Multiplexer With Enable
- M3_B4B1 4x3-Bit Bus to 1x3-Bit Bus Multiplexer
- M3_B4B1_SB 4x3-Bit Bus to 1x3-Bit Bus Multiplexer With Bus Version Select
- M3_B4B1E 4x3-Bit Bus to 1x3-Bit Bus Multiplexer With Enable
- M3_B4B1E_SB 4x3-Bit Bus to 1x3-Bit Bus Multiplexer With Enable With Bus Version Select
- M3_B8B1 8x3-Bit Bus to 1x3-Bit Bus Multiplexer
- M3_B8B1_SB 8x3-Bit Bus to 1x3-Bit Bus Multiplexer With Bus Version Select
- M3_B8B1E 8x3-Bit Bus to 1x3-Bit Bus Multiplexer With Enable
- M3_B8B1E_SB 8x3-Bit Bus to 1x3-Bit Bus Multiplexer With Enable With Bus Version Select
- M3_B16B1 16x3-Bit Bus to 1x3-Bit Bus Multiplexer
- M3_B16B1_SB 16x3-Bit Bus to 1x3-Bit Bus Multiplexer With Bus Version Select
- M3_B16B1E 16x3-Bit Bus to 1x3-Bit Bus Multiplexer With Enable
- M3_B16B1E_SB 16x3-Bit Bus to 1x3-Bit Bus Multiplexer With Enable and Bus Version Select
- M3_S1S2 1x3-Single Wire to 2x3-Single Wire Multiplexer (Demultiplex)
- M3_S1S2E 1x3-Single Wire to 1x3-Single Wire Multiplexer (Demultiplex) With Enable
- M3_S1S4 1x3-Single Wire to 4x3-Single Wire Multiplexer (Demultiplex)
- M3_S1S4_SB 1x3-Single Wire to 4x3-Single Wire Multiplexer (Demultiplex) With Bus Version Select
- M3_S1S4E 1x3-Single Wire to 1x3-Single Wire Multiplexer (Demultiplex) With Enable
- M3_S1S4E_SB 1x3-Single Wire to 1x3-Single Wire Multiplexer (Demultiplex) With Enable With Bus Version Select
- M3_S2S1 2x3-Single Wire to 1x3-Single Wire Multiplexer
- M3_S2S1E 2x3-Single Wire to 2x3-Single Wire Multiplexer With Enable
- M3_S4S1 4x3-Single Wire to 1x3-Single Wire Multiplexer
- M3_S4S1_SB 4x3-Single Wire to 1x3-Single Wire Multiplexer With Bus Version Select
- M3_S4S1E 4x3-Single Wire to 4x3-Single Wire Multiplexer With Enable
- M3_S4S1E_SB 4x3-Single Wire to 4x3-Single Wire Multiplexer With Enable With Bus Version Select
- M4_B1B2 1x4-Bit Bus to 2x4-Bit Bus Multiplexer (Demultiplex)
- M4_B1B2E 1x4-Bit Bus to 2x4-Bit Bus Multiplexer (Demultiplex) With Enable

- M4_B1B4 1x4-Bit Bus to 4x4-Bit Bus Multiplexer (Demultiplex)
- M4_B1B4_SB 1x4-Bit Bus to 4x4-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M4_B1B4E 1x4-Bit Bus to 4x4-Bit Bus Multiplexer (Demultiplex) With Enable
- M4_B1B4E_SB 1x4-Bit Bus to 4x4-Bit Bus Multiplexer (Demultiplex) With Enable With Bus Version Select
- M4_B1B8 1x4-Bit Bus to 8x4-Bit Bus Multiplexer (Demultiplex)
- M4_B1B8_SB 1x4-Bit Bus to 8x4-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M4_B1B8E 1x4-Bit Bus to 8x4-Bit Bus Multiplexer (Demultiplex) With Enable
- M4_B1B8E_SB 1x4-Bit Bus to 8x4-Bit Bus Multiplexer (Demultiplex) With Enable With Bus Version Select
- M4_B1B16 1x4-Bit Bus to 16x4-Bit Bus Multiplexer (Demultiplex)
- M4_B1B16_SB 1x4-Bit Bus to 16x4-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M4_B1B16E 1x4-Bit Bus to 16x4-Bit Bus Multiplexer (Demultiplex) With Enable
- M4_B1B16E_SB 1x4-Bit Bus to 16x4-Bit Bus Multiplexer (Demultiplex) With Enable and Bus Version Select
- M4_B2B1 2x4-Bit Bus to 1x4-Bit Bus Multiplexer
- M4_B2B1E 2x4-Bit Bus to 1x4-Bit Bus Multiplexer With Enable
- M4_B4B1 4x4-Bit Bus to 1x4-Bit Bus Multiplexer
- M4_B4B1_SB 4x4-Bit Bus to 1x4-Bit Bus Multiplexer With Bus Version Select
- M4_B4B1E 4x4-Bit Bus to 1x4-Bit Bus Multiplexer With Enable
- M4_B4B1E_SB 4x4-Bit Bus to 1x4-Bit Bus Multiplexer With Enable With Bus Version Select
- M4_B8B1 8x4-Bit Bus to 1x4-Bit Bus Multiplexer
- M4_B8B1_SB 8x4-Bit Bus to 1x4-Bit Bus Multiplexer With Bus Version Select
- M4_B8B1E 8x4-Bit Bus to 1x4-Bit Bus Multiplexer With Enable
- M4_B8B1E_SB 8x4-Bit Bus to 1x4-Bit Bus Multiplexer With Enable With Bus Version Select
- M4_B16B1 16x4-Bit Bus to 1x4-Bit Bus Multiplexer
- M4_B16B1_SB 16x4-Bit Bus to 1x4-Bit Bus Multiplexer With Bus Version Select
- M4_B16B1E 16x4-Bit Bus to 1x4-Bit Bus Multiplexer With Enable
- M4_B16B1E_SB 16x4-Bit Bus to 1x4-Bit Bus Multiplexer With Enable and Bus Version Select
- M4_S1S2 1x4-Single Wire to 2x4-Single Wire Multiplexer (Demultiplex)
- M4_S1S2E 1x4-Single Wire to 2x4-Single Wire Multiplexer (Demultiplex) With Enable
- M4_S1S4 1x4-Single Wire to 4x4-Single Wire Multiplexer (Demultiplex)
- M4_S1S4_SB 1x4-Single Wire to 4x4-Single Wire Multiplexer (Demultiplex) With Bus Version Select
- M4_S1S4E 1x4-Single Wire to 4x4-Single Wire Multiplexer (Demultiplex) With Enable

- M4_S1S4E_SB 1x4-Single Wire to 4x4-Single Wire Multiplexer (Demultiplex) With Enable With Bus Version Select
- M4_S2S1 2x4-Single Wire to 1x4-Single Wire Multiplexer
- M4_S2S1E 2x4-Single Wire to 1x4-Single Wire Multiplexer With Enable
- M4_S4S1 4x4-Single Wire to 1x4-Single Wire Multiplexer
- M4_S4S1_SB 4x4-Single Wire to 1x4-Single Wire Multiplexer With Bus Version Select
- M4_S4S1E 4x4-Single Wire to 1x4-Single Wire Multiplexer With Enable
- M4_S4S1E_SB 4x4-Single Wire to 1x4-Single Wire Multiplexer With Enable With Bus Version Select
- M5_B1B2 1x5-Bit Bus to 2x5-Bit Bus Multiplexer (Demultiplex)
- M5_B1B2E 1x5-Bit Bus to 2x5-Bit Bus Multiplexer (Demultiplex) With Enable
- M5_B1B4 1x5-Bit Bus to 4x5-Bit Bus Multiplexer (Demultiplex)
- M5_B1B4_SB 1x5-Bit Bus to 4x5-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M5_B1B4E 1x5-Bit Bus to 4x5-Bit Bus Multiplexer (Demultiplex) With Enable
- M5_B1B4E_SB 1x5-Bit Bus to 4x5-Bit Bus Multiplexer (Demultiplex) With Enable With Bus Version Select
- M5_B1B8 1x5-Bit Bus to 8x5-Bit Bus Multiplexer (Demultiplex)
- M5_B1B8_SB 1x5-Bit Bus to 8x5-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M5_B1B8E 1x5-Bit Bus to 8x5-Bit Bus Multiplexer (Demultiplex) With Enable
- M5_B1B8E_SB 1x5-Bit Bus to 8x5-Bit Bus Multiplexer (Demultiplex) With Enable With Bus Version Select
- M5_B1B16 1x5-Bit Bus to 16x5-Bit Bus Multiplexer (Demultiplex)
- M5_B1B16_SB 1x5-Bit Bus to 16x5-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M5_B1B16E 1x5-Bit Bus to 16x5-Bit Bus Multiplexer (Demultiplex) With Enable
- M5_B1B16E_SB 1x5-Bit Bus to 16x5-Bit Bus Multiplexer (Demultiplex) With Enable and Bus Version Select
- M5_B2B1 2x5-Bit Bus to 1x5-Bit Bus Multiplexer
- M5_B2B1E 2x5-Bit Bus to 1x5-Bit Bus Multiplexer With Enable
- M5_B4B1 4x5-Bit Bus to 1x5-Bit Bus Multiplexer
- M5_B4B1_SB 4x5-Bit Bus to 1x5-Bit Bus Multiplexer With Bus Version Select
- M5_B4B1E 4x5-Bit Bus to 1x5-Bit Bus Multiplexer With Enable
- M5_B4B1E_SB 4x5-Bit Bus to 1x5-Bit Bus Multiplexer With Enable With Bus Version Select
- M5_B8B1 8x5-Bit Bus to 1x5-Bit Bus Multiplexer
- M5_B8B1_SB 8x5-Bit Bus to 1x5-Bit Bus Multiplexer With Bus Version Select
- M5_B8B1E 8x5-Bit Bus to 1x5-Bit Bus Multiplexer With Enable

- M5_B8B1E_SB 8x5-Bit Bus to 1x5-Bit Bus Multiplexer With Enable With Bus Version Select
- M5_B16B1 16x5-Bit Bus to 1x5-Bit Bus Multiplexer
- M5_B16B1_SB 16x5-Bit Bus to 1x5-Bit Bus Multiplexer With Bus Version Select
- M5_B16B1E 16x5-Bit Bus to 1x5-Bit Bus Multiplexer With Enable
- M5_B16B1E_SB 16x5-Bit Bus to 1x5-Bit Bus Multiplexer With Enable and Bus Version Select
- M5_S1S2 1x5-Single Wire to 1x5-Single Wire Multiplexer (Demultiplex)
- M5_S1S2E 1x5-Single Wire to 1x5-Single Wire Multiplexer (Demultiplex) With Enable
- M5_S2S1 2x5-Single Wire to 2x5-Single Wire Multiplexer
- M5_S2S1E 2x5-Single Wire to 2x5-Single Wire Multiplexer With Enable
- M6_B1B2 1x6-Bit Bus to 2x6-Bit Bus Multiplexer (Demultiplex)
- M6_B1B2E 1x6-Bit Bus to 2x6-Bit Bus Multiplexer (Demultiplex) With Enable
- M6_B1B4 1x6-Bit Bus to 4x6-Bit Bus Multiplexer (Demultiplex)
- M6_B1B4_SB 1x6-Bit Bus to 4x6-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M6_B1B4E 1x6-Bit Bus to 4x6-Bit Bus Multiplexer (Demultiplex) With Enable
- M6_B1B4E_SB 1x6-Bit Bus to 4x6-Bit Bus Multiplexer (Demultiplex) With Enable With Bus Version Select
- M6_B1B8 1x6-Bit Bus to 8x6-Bit Bus Multiplexer (Demultiplex)
- M6_B1B8_SB 1x6-Bit Bus to 8x6-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M6_B1B8E 1x6-Bit Bus to 8x6-Bit Bus Multiplexer (Demultiplex) With Enable
- M6_B1B8E_SB 1x6-Bit Bus to 8x6-Bit Bus Multiplexer (Demultiplex) With Enable With Bus Version Select
- M6_B1B16 1x6-Bit Bus to 16x6-Bit Bus Multiplexer (Demultiplex)
- M6_B1B16_SB 1x6-Bit Bus to 16x6-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M6_B1B16E 1x6-Bit Bus to 16x6-Bit Bus Multiplexer (Demultiplex) With Enable
- M6_B1B16E_SB 1x6-Bit Bus to 16x6-Bit Bus Multiplexer (Demultiplex) With Enable and Bus Version Select
- M6_B2B1 2x6-Bit Bus to 1x6-Bit Bus Multiplexer
- M6_B2B1E 2x6-Bit Bus to 1x6-Bit Bus Multiplexer With Enable
- M6_B4B1 4x6-Bit Bus to 1x6-Bit Bus Multiplexer
- M6_B4B1_SB 4x6-Bit Bus to 1x6-Bit Bus Multiplexer With Bus Version Select
- M6_B4B1E 4x6-Bit Bus to 1x6-Bit Bus Multiplexer With Enable
- M6_B4B1E_SB 4x6-Bit Bus to 1x6-Bit Bus Multiplexer With Enable With Bus Version Select
- M6_B8B1 8x6-Bit Bus to 1x6-Bit Bus Multiplexer
- M6_B8B1_SB 8x6-Bit Bus to 1x6-Bit Bus Multiplexer With Bus Version Select

- M6_B8B1E 8x6-Bit Bus to 1x6-Bit Bus Multiplexer With Enable
- M6_B8B1E_SB 8x6-Bit Bus to 1x6-Bit Bus Multiplexer With Enable With Bus Version Select
- M6_B16B1 16x6-Bit Bus to 1x6-Bit Bus Multiplexer
- M6_B16B1_SB 16x6-Bit Bus to 1x6-Bit Bus Multiplexer With Bus Version Select
- M6_B16B1E 16x6-Bit Bus to 1x6-Bit Bus Multiplexer With Enable
- M6_B16B1E_SB 16x6-Bit Bus to 1x6-Bit Bus Multiplexer With Enable and Bus Version Select
- M6_S1S2 1x6-Single Wire to 1x6-Single Wire Multiplexer (Demultiplex)
- M6_S1S2E 1x6-Single Wire to 1x6-Single Wire Multiplexer (Demultiplex) With Enable
- M6_S2S1 2x6-Single Wire to 2x6-Single Wire Multiplexer
- M6_S2S1E 2x6-Single Wire to 2x6-Single Wire Multiplexer With Enable
- M7_B1B2 1x7-Bit Bus to 2x7-Bit Bus Multiplexer (Demultiplex)
- M7_B1B2E 1x7-Bit Bus to 2x7-Bit Bus Multiplexer (Demultiplex) With Enable
- M7_B1B4 1x7-Bit Bus to 4x7-Bit Bus Multiplexer (Demultiplex)
- M7_B1B4_SB 1x7-Bit Bus to 4x7-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M7_B1B4E 1x7-Bit Bus to 4x7-Bit Bus Multiplexer (Demultiplex) With Enable
- M7_B1B4E_SB 1x7-Bit Bus to 4x7-Bit Bus Multiplexer (Demultiplex) With Enable With Bus Version Select
- M7_B1B8 1x7-Bit Bus to 8x7-Bit Bus Multiplexer (Demultiplex)
- M7_B1B8_SB 1x7-Bit Bus to 8x7-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M7_B1B8E 1x7-Bit Bus to 8x7-Bit Bus Multiplexer (Demultiplex) With Enable
- M7_B1B8E_SB 1x7-Bit Bus to 8x7-Bit Bus Multiplexer (Demultiplex) With Enable With Bus Version Select
- M7_B1B16 1x7-Bit Bus to 16x7-Bit Bus Multiplexer (Demultiplex)
- M7_B1B16_SB 1x7-Bit Bus to 16x7-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M7_B1B16E 1x7-Bit Bus to 16x7-Bit Bus Multiplexer (Demultiplex) With Enable
- M7_B1B16E_SB 1x7-Bit Bus to 16x7-Bit Bus Multiplexer (Demultiplex) With Enable and Bus Version Select
- M7_B2B1 2x7-Bit Bus to 1x7-Bit Bus Multiplexer
- M7_B2B1E 2x7-Bit Bus to 1x7-Bit Bus Multiplexer With Enable
- M7_B4B1 4x7-Bit Bus to 1x7-Bit Bus Multiplexer
- M7_B4B1_SB 4x7-Bit Bus to 1x7-Bit Bus Multiplexer With Bus Version Select
- M7_B4B1E 4x7-Bit Bus to 1x7-Bit Bus Multiplexer With Enable
- M7_B4B1E_SB 4x7-Bit Bus to 1x7-Bit Bus Multiplexer With Enable With Bus Version Select
- M7_B8B1 8x7-Bit Bus to 1x7-Bit Bus Multiplexer

- M7_B8B1_SB 8x7-Bit Bus to 1x7-Bit Bus Multiplexer With Bus Version Select
- M7_B8B1E 8x7-Bit Bus to 1x7-Bit Bus Multiplexer With Enable
- M7_B8B1E_SB 8x7-Bit Bus to 1x7-Bit Bus Multiplexer With Enable With Bus Version Select
- M7_B16B1 16x7-Bit Bus to 1x7-Bit Bus Multiplexer
- M7_B16B1_SB 16x7-Bit Bus to 1x7-Bit Bus Multiplexer With Bus Version Select
- M7_B16B1E 16x7-Bit Bus to 1x7-Bit Bus Multiplexer With Enable
- M7_B16B1E_SB 16x7-Bit Bus to 1x7-Bit Bus Multiplexer With Enable and Bus Version Select
- M7_S1S2 1x7-Single Wire to 1x7-Single Wire Multiplexer (Demultiplex)
- M7_S1S2E 1x7-Single Wire to 1x7-Single Wire Multiplexer (Demultiplex) With Enable
- M7_S2S1 2x7-Single Wire to 2x7-Single Wire Multiplexer
- M7_S2S1E 2x7-Single Wire to 2x7-Single Wire Multiplexer With Enable
- M8_B1B2 1x8-Bit Bus to 2x8-Bit Bus Multiplexer (Demultiplex)
- M8_B1B2E 1x8-Bit Bus to 2x8-Bit Bus Multiplexer (Demultiplex) With Enable
- M8_B1B4 1x8-Bit Bus to 4x8-Bit Bus Multiplexer (Demultiplex)
- M8_B1B4_SB 1x8-Bit Bus to 4x8-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M8_B1B4E 1x8-Bit Bus to 4x8-Bit Bus Multiplexer (Demultiplex) With Enable
- M8_B1B4E_SB 1x8-Bit Bus to 4x8-Bit Bus Multiplexer (Demultiplex) With Enable With Bus Version Select
- M8_B1B8 1x8-Bit Bus to 8x8-Bit Bus Multiplexer (Demultiplex)
- M8_B1B8_SB 1x8-Bit Bus to 8x8-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M8_B1B8E 1x8-Bit Bus to 8x8-Bit Bus Multiplexer (Demultiplex) With Enable
- M8_B1B8E_SB 1x8-Bit Bus to 8x8-Bit Bus Multiplexer (Demultiplex) With Enable With Bus Version Select
- M8_B1B16 1x8-Bit Bus to 16x8-Bit Bus Multiplexer (Demultiplex)
- M8_B1B16_SB 1x8-Bit Bus to 16x8-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M8_B1B16E 1x8-Bit Bus to 16x8-Bit Bus Multiplexer (Demultiplex) With Enable
- M8_B1B16E_SB 1x8-Bit Bus to 16x8-Bit Bus Multiplexer (Demultiplex) With Enable and Bus Version Select
- M8_B2B1 2x8-Bit Bus to 1x8-Bit Bus Multiplexer
- M8_B2B1E 2x8-Bit Bus to 1x8-Bit Bus Multiplexer With Enable
- M8_B4B1 4x8-Bit Bus to 1x8-Bit Bus Multiplexer
- M8_B4B1_SB 4x8-Bit Bus to 1x8-Bit Bus Multiplexer With Bus Version Select
- M8_B4B1E 4x8-Bit Bus to 1x8-Bit Bus Multiplexer With Enable
- M8_B4B1E_SB 4x8-Bit Bus to 1x8-Bit Bus Multiplexer With Enable With Bus Version Select

- M8_B8B1 8x8-Bit Bus to 1x8-Bit Bus Multiplexer
- M8_B8B1_SB 8x8-Bit Bus to 1x8-Bit Bus Multiplexer With Bus Version Select
- M8_B8B1E 8x8-Bit Bus to 1x8-Bit Bus Multiplexer With Enable
- M8_B8B1E_SB 8x8-Bit Bus to 1x8-Bit Bus Multiplexer With Enable With Bus Version Select
- M8_B16B1 16x8-Bit Bus to 1x8-Bit Bus Multiplexer
- M8_B16B1_SB 16x8-Bit Bus to 1x8-Bit Bus Multiplexer With Bus Version Select
- M8_B16B1E 16x8-Bit Bus to 1x8-Bit Bus Multiplexer With Enable
- M8_B16B1E_SB 16x8-Bit Bus to 1x8-Bit Bus Multiplexer With Enable and Bus Version Select
- M8_S1S2 1x8-Single Wire to 2x8-Single Wire Multiplexer (Demultiplex)
- M8_S1S2E 1x8-Single Wire to 2x8-Single Wire Multiplexer (Demultiplex) With Enable
- M8_S2S1 2x8-Single Wire to 1x8-Single Wire Multiplexer
- M8_S2S1E 2x8-Single Wire to 1x8-Single Wire Multiplexer With Enable
- M9_B1B2 1x9-Bit Bus to 2x9-Bit Bus Multiplexer (Demultiplex)
- M9_B1B2E 1x9-Bit Bus to 2x9-Bit Bus Multiplexer (Demultiplex) With Enable
- M9_B1B4 1x9-Bit Bus to 4x9-Bit Bus Multiplexer (Demultiplex)
- M9_B1B4_SB 1x9-Bit Bus to 4x9-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M9_B1B4E 1x9-Bit Bus to 4x9-Bit Bus Multiplexer (Demultiplex) With Enable
- M9_B1B4E_SB 1x9-Bit Bus to 4x9-Bit Bus Multiplexer (Demultiplex) With Enable With Bus Version Select
- M9_B1B8 1x9-Bit Bus to 8x9-Bit Bus Multiplexer (Demultiplex)
- M9_B1B8_SB 1x9-Bit Bus to 8x9-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M9_B1B8E 1x9-Bit Bus to 8x9-Bit Bus Multiplexer (Demultiplex) With Enable
- M9_B1B8E_SB 1x9-Bit Bus to 8x9-Bit Bus Multiplexer (Demultiplex) With Enable With Bus Version Select
- M9_B1B16 1x9-Bit Bus to 16x9-Bit Bus Multiplexer (Demultiplex)
- M9_B1B16_SB 1x9-Bit Bus to 16x9-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M9_B1B16E 1x9-Bit Bus to 16x9-Bit Bus Multiplexer (Demultiplex) With Enable
- M9_B1B16E_SB 1x9-Bit Bus to 16x9-Bit Bus Multiplexer (Demultiplex) With Enable and Bus Version Select
- M9_B2B1 2x9-Bit Bus to 1x9-Bit Bus Multiplexer
- M9_B2B1E 2x9-Bit Bus to 1x9-Bit Bus Multiplexer With Enable
- M9_B4B1 4x9-Bit Bus to 1x9-Bit Bus Multiplexer
- M9_B4B1_SB 4x9-Bit Bus to 1x9-Bit Bus Multiplexer With Bus Version Select
- M9_B4B1E 4x9-Bit Bus to 1x9-Bit Bus Multiplexer With Enable

- M9_B4B1E_SB 4x9-Bit Bus to 1x9-Bit Bus Multiplexer With Enable With Bus Version Select
- M9_B8B1 8x9-Bit Bus to 1x9-Bit Bus Multiplexer
- M9_B8B1_SB 8x9-Bit Bus to 1x9-Bit Bus Multiplexer With Bus Version Select
- M9_B8B1E 8x9-Bit Bus to 1x9-Bit Bus Multiplexer With Enable
- M9_B8B1E_SB 8x9-Bit Bus to 1x9-Bit Bus Multiplexer With Enable With Bus Version Select
- M9_B16B1 16x9-Bit Bus to 1x9-Bit Bus Multiplexer
- M9_B16B1_SB 16x9-Bit Bus to 1x9-Bit Bus Multiplexer With Bus Version Select
- M9_B16B1E 16x9-Bit Bus to 1x9-Bit Bus Multiplexer With Enable
- M9_B16B1E_SB 16x9-Bit Bus to 1x9-Bit Bus Multiplexer With Enable and Bus Version Select
- M10_B1B2 1x10-Bit Bus to 2x10-Bit Bus Multiplexer (Demultiplex)
- M10_B1B2E 1x10-Bit Bus to 2x10-Bit Bus Multiplexer (Demultiplex) With Enable
- M10_B1B4 1x10-Bit Bus to 4x10-Bit Bus Multiplexer (Demultiplex)
- M10_B1B4_SB 1x10-Bit Bus to 4x10-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M10_B1B4E 1x10-Bit Bus to 4x10-Bit Bus Multiplexer (Demultiplex) With Enable
- M10_B1B4E_SB 1x10-Bit Bus to 4x10-Bit Bus Multiplexer (Demultiplex) With Enable With Bus Version Select
- M10_B1B8 1x10-Bit Bus to 8x10-Bit Bus Multiplexer (Demultiplex)
- M10_B1B8_SB 1x10-Bit Bus to 8x10-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M10_B1B8E 1x10-Bit Bus to 8x10-Bit Bus Multiplexer (Demultiplex) With Enable
- M10_B1B8E_SB 1x10-Bit Bus to 8x10-Bit Bus Multiplexer (Demultiplex) With Enable With Bus Version Select
- M10_B1B16 1x10-Bit Bus to 16x10-Bit Bus Multiplexer (Demultiplex)
- M10_B1B16_SB 1x10-Bit Bus to 16x10-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M10_B1B16E 1x10-Bit Bus to 16x10-Bit Bus Multiplexer (Demultiplex) With Enable
- M10_B1B16E_SB 1x10-Bit Bus to 16x10-Bit Bus Multiplexer (Demultiplex) With Enable and Bus Version Select
- M10_B2B1 2x10-Bit Bus to 1x10-Bit Bus Multiplexer
- M10_B2B1E 2x10-Bit Bus to 1x10-Bit Bus Multiplexer With Enable
- M10_B4B1 4x10-Bit Bus to 1x10-Bit Bus Multiplexer
- M10_B4B1_SB 4x10-Bit Bus to 1x10-Bit Bus Multiplexer With Bus Version Select
- M10_B4B1E 4x10-Bit Bus to 1x10-Bit Bus Multiplexer With Enable
- M10_B4B1E_SB 4x10-Bit Bus to 1x10-Bit Bus Multiplexer With Enable With Bus Version Select
- M10_B8B1 8x10-Bit Bus to 1x10-Bit Bus Multiplexer
- M10_B8B1_SB 8x10-Bit Bus to 1x10-Bit Bus Multiplexer With Bus Version Select

- M10_B8B1E 8x10-Bit Bus to 1x10-Bit Bus Multiplexer With Enable
- M10_B8B1E_SB 8x10-Bit Bus to 1x10-Bit Bus Multiplexer With Enable With Bus Version Select
- M10_B16B1 16x10-Bit Bus to 1x10-Bit Bus Multiplexer
- M10_B16B1_SB 16x10-Bit Bus to 1x10-Bit Bus Multiplexer With Bus Version Select
- M10_B16B1E 16x10-Bit Bus to 1x10-Bit Bus Multiplexer With Enable
- M10_B16B1E_SB 16x10-Bit Bus to 1x10-Bit Bus Multiplexer With Enable and Bus Version Select
- M12_B1B2 1x12-Bit Bus to 2x12-Bit Bus Multiplexer (Demultiplex)
- M12_B1B2E 1x12-Bit Bus to 2x12-Bit Bus Multiplexer (Demultiplex) With Enable
- M12_B1B4 1x12-Bit Bus to 4x12-Bit Bus Multiplexer (Demultiplex)
- M12_B1B4_SB 1x12-Bit Bus to 4x12-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M12_B1B4E 1x12-Bit Bus to 4x12-Bit Bus Multiplexer (Demultiplex) With Enable
- M12_B1B4E_SB 1x12-Bit Bus to 4x12-Bit Bus Multiplexer (Demultiplex) With Enable With Bus Version Select
- M12_B1B8 1x12-Bit Bus to 8x12-Bit Bus Multiplexer (Demultiplex)
- M12_B1B8_SB 1x12-Bit Bus to 8x12-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M12_B1B8E 1x12-Bit Bus to 8x12-Bit Bus Multiplexer (Demultiplex) With Enable
- M12_B1B8E_SB 1x12-Bit Bus to 8x12-Bit Bus Multiplexer (Demultiplex) With Enable With Bus Version Select
- M12_B1B16 1x12-Bit Bus to 16x12-Bit Bus Multiplexer (Demultiplex)
- M12_B1B16_SB 1x12-Bit Bus to 16x12-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M12_B1B16E 1x12-Bit Bus to 16x12-Bit Bus Multiplexer (Demultiplex) With Enable
- M12_B1B16E_SB 1x12-Bit Bus to 16x12-Bit Bus Multiplexer (Demultiplex) With Enable and Bus Version Select
- M12_B2B1 2x12-Bit Bus to 1x12-Bit Bus Multiplexer
- M12_B2B1E 2x12-Bit Bus to 1x12-Bit Bus Multiplexer With Enable
- M12_B4B1 4x12-Bit Bus to 1x12-Bit Bus Multiplexer
- M12_B4B1_SB 4x12-Bit Bus to 1x12-Bit Bus Multiplexer With Bus Version Select
- M12_B4B1E 4x12-Bit Bus to 1x12-Bit Bus Multiplexer With Enable
- M12_B4B1E_SB 4x12-Bit Bus to 1x12-Bit Bus Multiplexer With Enable With Bus Version Select
- M12_B8B1 8x12-Bit Bus to 1x12-Bit Bus Multiplexer
- M12_B8B1_SB 8x12-Bit Bus to 1x12-Bit Bus Multiplexer With Bus Version Select
- M12_B8B1E 8x12-Bit Bus to 1x12-Bit Bus Multiplexer With Enable

- M12_B8B1E_SB 8x12-Bit Bus to 1x12-Bit Bus Multiplexer With Enable With Bus Version Select
- M12_B16B1 16x12-Bit Bus to 1x12-Bit Bus Multiplexer
- M12_B16B1_SB 16x12-Bit Bus to 1x12-Bit Bus Multiplexer With Bus Version Enable
- M12_B16B1E 16x12-Bit Bus to 1x12-Bit Bus Multiplexer With Enable
- M12_B16B1E_SB 16x12-Bit Bus to 1x12-Bit Bus Multiplexer With Enable and Bus Version Select
- M16_B1B2 1x16-Bit Bus to 2x16-Bit Bus Multiplexer (Demultiplex)
- M16_B1B2E 1x16-Bit Bus to 2x16-Bit Bus Multiplexer (Demultiplex) With Enable
- M16_B1B4 1x16-Bit Bus to 4x16-Bit Bus Multiplexer (Demultiplex)
- M16_B1B4_SB 1x16-Bit Bus to 4x16-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M16_B1B4E 1x16-Bit Bus to 4x16-Bit Bus Multiplexer (Demultiplex) With Enable
- M16_B1B4E_SB 1x16-Bit Bus to 4x16-Bit Bus Multiplexer (Demultiplex) With Enable With Bus Version Select
- M16_B1B8 1x16-Bit Bus to 8x16-Bit Bus Multiplexer (Demultiplex)
- M16_B1B8_SB 1x16-Bit Bus to 8x16-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M16_B1B8E 1x16-Bit Bus to 8x16-Bit Bus Multiplexer (Demultiplex) With Enable
- M16_B1B8E_SB 1x16-Bit Bus to 8x16-Bit Bus Multiplexer (Demultiplex) With Enable With Bus Version Select
- M16_B1B16 1x16-Bit Bus to 16x16-Bit Bus Multiplexer (Demultiplex)
- M16_B1B16_SB 1x16-Bit Bus to 16x16-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M16_B1B16E 1x16-Bit Bus to 16x16-Bit Bus Multiplexer (Demultiplex) With Enable
- M16_B1B16E_SB 1x16-Bit Bus to 16x16-Bit Bus Multiplexer (Demultiplex) With Enable and Bus Version Select
- M16_B2B1 2x16-Bit Bus to 1x16-Bit Bus Multiplexer
- M16_B2B1E 2x16-Bit Bus to 1x16-Bit Bus Multiplexer With Enable
- M16_B4B1 4x16-Bit Bus to 1x16-Bit Bus Multiplexer
- M16_B4B1_SB 4x16-Bit Bus to 1x16-Bit Bus Multiplexer With Bus Version Select
- M16_B4B1E 4x16-Bit Bus to 1x16-Bit Bus Multiplexer With Enable
- M16_B4B1E_SB 4x16-Bit Bus to 1x16-Bit Bus Multiplexer With Enable With Bus Version Select
- M16_B8B1 8x16-Bit Bus to 1x16-Bit Bus Multiplexer
- M16_B8B1_SB 8x16-Bit Bus to 1x16-Bit Bus Multiplexer With Bus Version Select
- M16_B8B1E 8x16-Bit Bus to 1x16-Bit Bus Multiplexer With Enable

- M16_B8B1E_SB 8x16-Bit Bus to 1x16-Bit Bus Multiplexer With Enable With Bus Version Select
- M16_B16B1 16x16-Bit Bus to 1x16-Bit Bus Multiplexer
- M16_B16B1_SB 16x16-Bit Bus to 1x16-Bit Bus Multiplexer With Bus Version Select
- M16_B16B1E 16x16-Bit Bus to 1x16-Bit Bus Multiplexer With Enable
- M16_B16B1E_SB 16x16-Bit Bus to 1x16-Bit Bus Multiplexer With Enable and Bus Version Select
- M32_B1B2 1x32-Bit Bus to 2x32-Bit Bus Multiplexer (Demultiplex)
- M32_B1B2E 1x32-Bit Bus to 2x32-Bit Bus Multiplexer (Demultiplex) With Enable
- M32_B1B4 1x32-Bit Bus to 4x32-Bit Bus Multiplexer (Demultiplex)
- M32_B1B4_SB 1x32-Bit Bus to 4x32-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M32_B1B4E 1x32-Bit Bus to 4x32-Bit Bus Multiplexer (Demultiplex) With Enable
- M32_B1B4E_SB 1x32-Bit Bus to 4x32-Bit Bus Multiplexer (Demultiplex) With Enable With Bus Version Select
- M32_B1B8 1x32-Bit Bus to 8x32-Bit Bus Multiplexer (Demultiplex)
- M32_B1B8_SB 1x32-Bit Bus to 8x32-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M32_B1B8E 1x32-Bit Bus to 8x32-Bit Bus Multiplexer (Demultiplex) With Enable
- M32_B1B8E_SB 1x32-Bit Bus to 8x32-Bit Bus Multiplexer (Demultiplex) With Enable With Bus Version Select
- M32_B1B16 1x32-Bit Bus to 16x32-Bit Bus Multiplexer (Demultiplex)
- M32_B1B16_SB 1x32-Bit Bus to 16x32-Bit Bus Multiplexer (Demultiplex) With Bus Version Select
- M32_B1B16E 1x32-Bit Bus to 16x32-Bit Bus Multiplexer (Demultiplex) With Enable
- M32_B1B16E_SB 1x32-Bit Bus to 16x32-Bit Bus Multiplexer (Demultiplex) With Enable and Bus Version Select
- M32_B2B1 2x32-Bit Bus to 1x32-Bit Bus Multiplexer
- M32_B2B1E 2x32-Bit Bus to 1x32-Bit Bus Multiplexer With Enable
- M32_B4B1 4x32-Bit Bus to 1x32-Bit Bus Multiplexer
- M32_B4B1_SB 4x32-Bit Bus to 1x32-Bit Bus Multiplexer With Bus Version Select
- M32_B4B1E 4x32-Bit Bus to 1x32-Bit Bus Multiplexer With Enable
- M32_B4B1E_SB 4x32-Bit Bus to 1x32-Bit Bus Multiplexer With Enable With Bus Version Select
- M32_B8B1 8x32-Bit Bus to 1x32-Bit Bus Multiplexer
- M32_B8B1_SB 8x32-Bit Bus to 1x32-Bit Bus Multiplexer With Bus Version Select
- M32_B8B1E 8x32-Bit Bus to 1x32-Bit Bus Multiplexer With Enable

- M32_B8B1E_SB 8x32-Bit Bus to 1x32-Bit Bus Multiplexer With Enable With Bus Version Select
- M32_B16B1 16x32-Bit Bus to 1x32-Bit Bus Multiplexer
- M32_B16B1_SB 16x32-Bit Bus to 1x32-Bit Bus Multiplexer With Bus Version Select
- M32_B16B1E 16x32-Bit Bus to 1x32-Bit Bus Multiplexer With Enable
- M32_B16B1E_SB 16x32-Bit Bus to 1x32-Bit Bus Multiplexer With Enable and Bus Version Select

Numeric Connector

Binary numeric connectors are available as follows:

- NUM0 Number Connector 0
- NUM1 Number Connector 1
- NUM2 Number Connector 2
- NUM3 Number Connector 3
- NUM4 Number Connector 4
- NUM5 Number Connector 5
- NUM6 Number Connector 6
- NUM7 Number Connector 7
- NUM8 Number Connector 8
- NUM9 Number Connector 9
- NUMA Number Connector A
- NUMB Number Connector B
- NUMC Number Connector C
- NUMD Number Connector D
- NUME Number Connector E
- NUMF Number Connector F

Shift Register

Multiple capability shift registers are available as follows:

- SR4CEB — 4-Bit Serial-In Parallel-Out Shift Registers with Clock Enable and Asynchronous Clear, Bus Version
- SR4CES — 4-Bit Serial-In Parallel-Out Shift Registers with Clock Enable and Asynchronous Clear, Single Pin Version
- SR4CLEB — 4-Bit Loadable Serial/Parallel-In Parallel-Out Shift Registers with Clock Enable and Asynchronous Clear, Bus Version
- SR4CLEDB — 4-Bit Loadable Serial/Parallel-In Parallel-Out Bidirectional Shift Registers with Clock Enable and Asynchronous Clear, Bus Version
- SR4CLEDS — 4-Bit Loadable Serial/Parallel-In Parallel-Out Bidirectional Shift Registers with Clock Enable and Asynchronous Clear, Single Pin Version
- SR4CLES — 4-Bit Loadable Serial/Parallel-In Parallel-Out Shift Registers with Clock Enable and Asynchronous Clear, Single Pin Version
- SR4REB — 4-Bit Serial-In Parallel-Out Shift Registers with Clock Enable and Synchronous Reset, Bus Version
- SR4RES — 4-Bit Serial-In Parallel-Out Shift Registers with Clock Enable and Synchronous Reset, Single Pin Version
- SR4RLEB — 4-Bit Loadable Serial/Parallel-In Parallel-Out Shift Registers with Clock Enable and Synchronous Reset, Bus Version
- SR4RLEDB — 4-Bit Loadable Serial/Parallel-In Parallel-Out Bidirectional Shift Registers with Clock Enable and Synchronous Reset, Bus Version
- SR4RLEDS — 4-Bit Loadable Serial/Parallel-In Parallel-Out Bidirectional Shift Registers with Clock Enable and Synchronous Reset, Single Pin Version
- SR4RLES — 4-Bit Loadable Serial/Parallel-In Parallel-Out Shift Registers with Clock Enable and Synchronous Reset, Single Pin Version
- SR8CEB — 8-Bit Serial-In Parallel-Out Shift Registers with Clock Enable and Asynchronous Clear, Bus Version
- SR8CES — 8-Bit Serial-In Parallel-Out Shift Registers with Clock Enable and Asynchronous Clear, Single Pin Version
- SR8CLEB — 8-Bit Loadable Serial/Parallel-In Parallel-Out Shift Registers with Clock Enable and Asynchronous Clear, Bus Version
- SR8CLEDB — 8-Bit Loadable Serial/Parallel-In Parallel-Out Bidirectional Shift Registers with Clock Enable and Asynchronous Clear, Bus Version
- SR8CLEDS — 8-Bit Loadable Serial/Parallel-In Parallel-Out Bidirectional Shift Registers with Clock Enable and Asynchronous Clear, Single Pin Version
- SR8CLES — 8-Bit Loadable Serial/Parallel-In Parallel-Out Shift Registers with Clock Enable and Asynchronous Clear, Single Pin Version

- SR8REB — 8-Bit Serial-In Parallel-Out Shift Registers with Clock Enable and Synchronous Reset, Bus Version
- SR8RES — 8-Bit Serial-In Parallel-Out Shift Registers with Clock Enable and Synchronous Reset, Single Pin Version
- SR8RLEB — 8-Bit Loadable Serial/Parallel-In Parallel-Out Shift Registers with Clock Enable and Synchronous Reset, Bus Version
- SR8RLEDB — 8-Bit Loadable Serial/Parallel-In Parallel-Out Bidirectional Shift Registers with Clock Enable and Synchronous Reset, Bus Version
- SR8RLEDS — 8-Bit Loadable Serial/Parallel-In Parallel-Out Bidirectional Shift Registers with Clock Enable and Synchronous Reset, Single Pin Version
- SR8RLES — 8-Bit Loadable Serial/Parallel-In Parallel-Out Shift Registers with Clock Enable and Synchronous Reset, Single Pin Version
- SR16CEB — 16-Bit Serial-In Parallel-Out Shift Registers with Clock Enable and Asynchronous Clear, Bus Version
- SR16CES — 16-Bit Serial-In Parallel-Out Shift Registers with Clock Enable and Asynchronous Clear, Single Pin Version
- SR16CLEB — 16-Bit Loadable Serial/Parallel-In Parallel-Out Shift Registers with Clock Enable and Asynchronous Clear, Bus Version
- SR16CLEDB — 16-Bit Loadable Serial/Parallel-In Parallel-Out Bidirectional Shift Registers with Clock Enable and Asynchronous Clear, Bus Version
- SR16CLEDS — 16-Bit Loadable Serial/Parallel-In Parallel-Out Bidirectional Shift Registers with Clock Enable and Asynchronous Clear, Single Pin Version
- SR16CLES — 16-Bit Loadable Serial/Parallel-In Parallel-Out Shift Registers with Clock Enable and Asynchronous Clear, Single Pin Version
- SR16REB — 16-Bit Serial-In Parallel-Out Shift Registers with Clock Enable and Synchronous Reset, Bus Version
- SR16RES — 16-Bit Serial-In Parallel-Out Shift Registers with Clock Enable and Synchronous Reset, Single Pin Version
- SR16RLEB — 16-Bit Loadable Serial/Parallel-In Parallel-Out Shift Registers with Clock Enable and Synchronous Reset, Bus Version
- SR16RLEDB — 16-Bit Loadable Serial/Parallel-In Parallel-Out Bidirectional Shift Registers with Clock Enable and Synchronous Reset, Bus Version
- SR16RLEDS — 16-Bit Loadable Serial/Parallel-In Parallel-Out Bidirectional Shift Registers with Clock Enable and Synchronous Reset, Single Pin Version
- SR16RLES — 16-Bit Loadable Serial/Parallel-In Parallel-Out Shift Registers with Clock Enable and Synchronous Reset, Single Pin Version
- SR32CEB — 32-Bit Serial-In Parallel-Out Shift Registers with Clock Enable and Asynchronous Clear, Bus Version
- SR32CLEB — 32-Bit Loadable Serial/Parallel-In Parallel-Out Shift Registers with Clock Enable and Asynchronous Clear, Bus Version

- SR32CLEDB 32-Bit Loadable Serial/Parallel-In Parallel-Out Bidirectional Shift Registers with Clock Enable and Asynchronous Clear, Bus Version
- SR32REB 32-Bit Serial-In Parallel-Out Shift Registers with Clock Enable and Synchronous Reset, Bus Version
- SR32RLEB 32-Bit Loadable Serial/Parallel-In Parallel-Out Shift Registers with Clock Enable and Synchronous Reset, Bus Version
- SR32RLEDB 32-Bit Loadable Serial/Parallel-In Parallel-Out Bidirectional Shift Registers with Clock Enable and Synchronous Reset, Bus Version

Shifter

Barrel shifters are available as follows:

- BRLSHFT4B 4-Bit Barrel Shifter, Bus Version
- BRLSHFT4S 4-Bit Barrel Shifter, Single Pin Version
- BRLSHFT8B 8-Bit Barrel Shifter, Bus Version
- BRLSHFT8S 8-Bit Barrel Shifter, Single Pin Version
- BRLSHFT16B 16-Bit Barrel Shifter, Bus Version
- BRLSHFT32B 32-Bit Barrel Shifter, Bus Version
- BRLSHFTM4B 4-Bit Fill Mode Bi-Directional Barrel Shifter, Bus Version
- BRLSHFTM4S 4-Bit Fill Mode Bi-Directional Barrel Shifter, Single pin Version
- BRLSHFTM8B 8-Bit Fill Mode Bi-Directional Barrel Shifter, Bus Version
- BRLSHFTM8S 8-Bit Fill Mode Bi-Directional Barrel Shifter, Single pin Version
- BRLSHFTM16B 16-Bit Fill Mode Bi-Directional Barrel Shifter, Bus Version
- BRLSHFTM32B 32-Bit Fill Mode Bi-Directional Barrel Shifter, Bus Version

Wired Function

Wired functions available are as follows:

- PULLDOWN Level Low
- PULLDOWN4B 4-Bit Level Low Bus
- PULLDOWN4S 4-Bit Level Low
- PULLDOWN8B 8-Bit Level Low Bus
- PULLDOWN8S 8-Bit Level Low
- PULLDOWN12B 12-Bit Level Low Bus
- PULLDOWN12S 12-Bit Level Low
- PULLDOWN16B 16-Bit Level Low Bus
- PULLDOWN16S 16-Bit Level Low
- PULLDOWN32B 32-Bit Level Low Bus
- PULLUP Level High
- PULLUP4B 4-Bit Level High Bus
- PULLUP4S 4-Bit Level High
- PULLUP8B 8-Bit Level High Bus
- PULLUP8S 8-Bit Level High
- PULLUP12B 12-Bit Level High Bus
- PULLUP12S 12-Bit Level High
- PULLUP16B 16-Bit Level High Bus
- PULLUP16S 16-Bit Level High
- PULLUP32B 32-Bit Level High Bus

Design Components

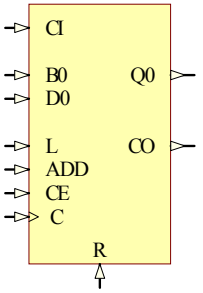
This section contains a complete description of each library component in the FPGA Generic Library. The component list is arranged alphanumerically, with all numeric suffixes in ascending order. Descriptions of the same component type are presented together: These groupings are indicated in the component title. *Example: ADD2, 4, 8, 16.* The designator for the component version, ‘B’ or ‘S’ is omitted from the component title.

The following information is provided for each component, where applicable

- Component(s) Title
- Functional Description
- Schematic Symbol
- Truth Table or equation
- Additional notes (if any)

ACC1

1-Bit Cascadable Loadable Accumulator



ACC1

ACC1 is a 1-Bit cascadable loadable accumulator. It can add or subtract data to or from the contents of a 1-bit data register and store the result back into the register. The register can be loaded with a 1-bit word.

The synchronous reset (R) has highest priority over all other inputs. When R is High, all other inputs are ignored and the outputs are reset to Low during the Low-to-High clock (C) transition.

The Load (L) input is the second highest priority input after R. When L is High, all other inputs are ignored and the data input D0 is loaded into the 1-bit register during the Low-to-High clock transition.

When R and L are Low, accumulation takes place when CE is High. The accumulation method depends on the input ADD. When ADD is High, data on inputs B0 and CI are added with the contents of the data register. When ADD is Low, inputs B0 and CI are subtracted from the contents of the data register. The accumulation result is then stored to the register during the Low-to-High clock transition. Output Q0 always reflects the value in the data register.

CI is a carry-in input and CO is a carry-out output. Both are active High in adding mode and active Low in subtraction mode.

CO is always active one step before the data output (Q) exceeds the 1-bit binary range since CO is not registered synchronously with data output. CO always reflects the accumulation of input B0 and the contents of the register, which allows cascading of ACC1s by connecting CO of one stage to CI of the next stage.

Inputs						Outputs
R	**L**	**CE**	**ADD**	**D0**	**C**	**Q0**
1	x	x	x	x	↑	0
0	1	x	x	d	↑	d
0	0	1	1	x	↑	q0+b+CI
0	0	1	0	x	↑	q0-b-CI
0	0	0	x	x	↑	No Chg

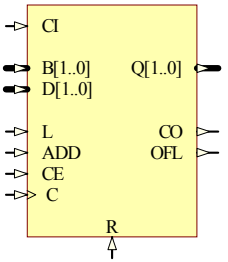
q0 is the previous value of Q (ie. in the register too)

b is the value of data input B0

CI is value of input CI

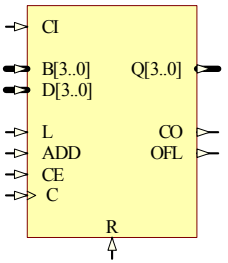
ACC2, 4, 8, 16, 32

Loadable Cascadable Accumulators with Signed and Unsigned Operations



ACC2B

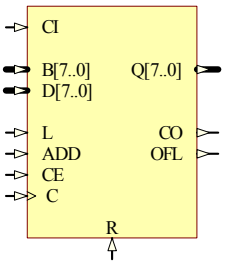
ACC2, ACC4, ACC8, ACC16 and ACC32 are, respectively 2-, 4-, 8-, 16- and 32-Bit loadable cascadable accumulators with signed (twos-complement) and unsigned binary operations. They can add or subtract 2-, 4-, 8-, 16-, 32-bit unsigned binary, respectively or two's complement number to or from the contents of a 2-, 4-, 8-, 16-, 32-bit data register and store the results in the register. The register can be loaded with 2-, 4-, 8-, 16-, 32-bit number.



ACC4B

Unsigned Binary and Two's complement (Signed Binary) operation

The accumulators can operate on signed (two's complement) or unsigned binary numbering formats depending on the interpretation of data input and data output. If the inputs are interpreted as unsigned binary, the result should be interpreted as unsigned binary. If the inputs are interpreted as two's complement, the output should be interpreted as twos-complement. When the data is interpreted as unsigned binary, output CO should be to determine overflow. When the data is interpreted as two's complement, output OFL should be used to determine the overflow. When cascading accumulators, CO is used as carry-out or borrow-out for both numbering format modes.



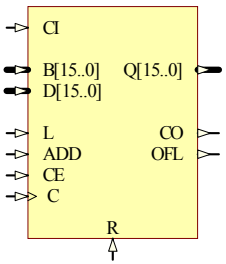
ACC8B

Accumulator Function

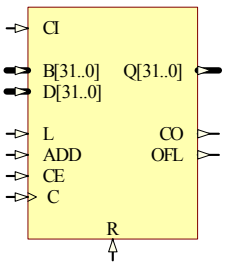
The synchronous reset (R) has highest priority over all other inputs. When R is High, all other inputs are ignored and the outputs are reset to Low during the Low-to-High clock transition.

The Load (L) input is the second highest priority input after R. When L is High, all other inputs are ignored and the data input D is loaded into the register during the Low-to-High clock (C) transition.

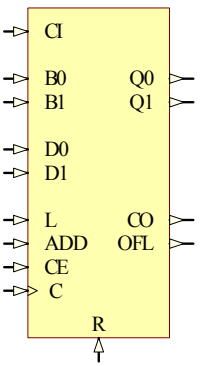
When R and L are Low, accumulation takes place when CE is High. The accumulation method depends on the input ADD. When ADD is High, data on inputs B and CI are added with the contents of the data register. When ADD is Low, inputs B and CI are subtracted from the contents of the data register. The accumulation result is then stored to the register during the Low-to-High clock transition. CI is active High for adding and active Low for subtraction. Output Q always reflects the value in the data register.



ACC16B



ACC32B



ACC2S

Inputs						Outputs
R	L	CE	ADD	D	C	Q
1	x	x	x	x	↑	0
0	1	x	x	d	↑	d
0	0	1	1	x	↑	q0+b+CI
0	0	1	0	x	↑	q0-b-CI
0	0	0	x	x	↑	No Chg

q0 is the previous value of Q (i.e. data in the register)

b is the value of data input B

CI is value of input CI

Overflow detection

CO and OFL are used to determine overflow for unsigned and signed accumulation respectively. They are not registered synchronously with data output. Thus, CO and OFL always active one step before the register or data output value (Q) actually goes overflow.

In unsigned binary operation, CO goes High when accumulation result (S) is going to exceed the unsigned binary boundary in the next accumulation. CO is active High in add mode and active Low in subtract mode, thus CO is Low when overflow occurs in subtract mode. OFL is ignored in unsigned operation.

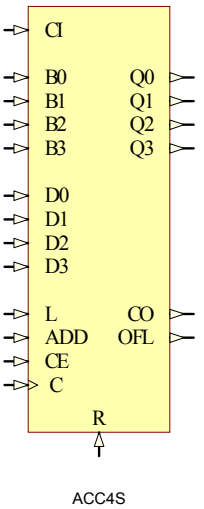
The unsigned binary ranges of the available ACC are:

ACC Type	Numbering System	Number Range
ACC2	2-bit unsigned binary	0 to 3
ACC4	4-bit unsigned binary	0 to 15
ACC8	8-bit unsigned binary	0 to 255
ACC16	16-bit unsigned binary	0 to 65535
ACC32	32-bit unsigned binary	0 to 4294967295

For two's complement operation, OFL is used as overflow detection. If the accumulation or de-accumulation result is going to exceed the two's complement range in the next accumulation step, OFL output goes High. OFL is active High in both add or subtract mode.

The twos-complement ranges of the available ACC are:

ACC Type	Numbering System	Number Range
ACC2	2-bit twos-complement	-2 to +1
ACC4	4-bit twos-complement	-8 to + 7
ACC8	8-bit twos-complement	-128 to +127
ACC16	16-bit twos-complement	-32768 to +32767

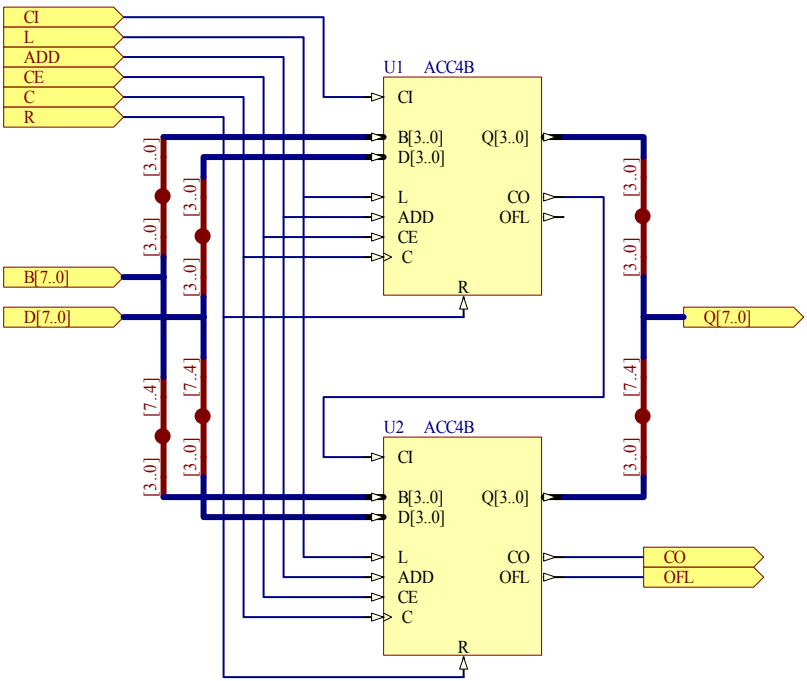


ACC4S

ACC32	32-bit twos-complement	-2147483648 to +2147483647

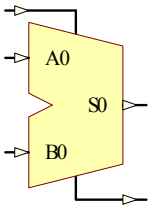
“Carry-out”,”Borrow-out” and Cascading

For cascading purpose, CO is used as a “carry-out” or “borrow-out” irrespective of the numbering format used. When cascading two or more ACC components together to create a larger device; CO from the upper level of ACC is connected to CI of the next level. OFL from the upper level is ignored, but the last OFL can still be used as overflows for two’s complement operation. The following example demonstrates how to create an 8-bit accumulator from 2 ACC4 components:



ADD1

1-Bit Cascadable Full Adder



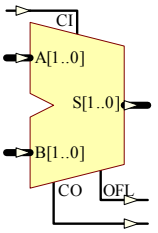
ADD1

ADD1 is a 1-Bit full adder. The device adds two 1-bit words (A0, B0) and a carry-in (CI), producing a binary sum (S0) output and a carry-out (CO).

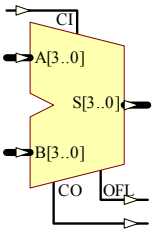
Inputs			Outputs	
CI	**A0**	**B0**	**S0**	**CO**
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

ADD2, 4, 8, 16, 32

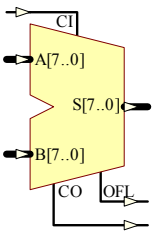
Cascadable Full Adders with Signed and Unsigned Operations



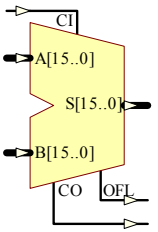
ADD2B



ADD4B



ADD8B



ADD16B

ADD2, ADD4, ADD8, ADD16 and ADD32 are, respectively 2-, 4-, 8-, 16- and 32-Bit cascadable full adders with signed (twos-complement) and unsigned binary operation. These adders add two input words (A, B) and a carry-in (CI) producing a sum output (S), which can be interpreted as either unsigned binary or two's complement format, and carry-out (CO) and overflow (OFL) outputs.

Unsigned Binary and Two's complement (Signed Binary) operation

ADD2, ADD4, ADD8, ADD16 and ADD32 can operate on either, 2-, 4-, 8-, 16- and 32-bit unsigned binary numbers or 2-, 4-, 8-, 16- and 32-bit two's complement numbers respectively.

If the inputs are interpreted as unsigned binary, the result should be interpreted as unsigned binary and the CO output should be used.

If the inputs are interpreted as two's complement, the output should be interpreted as twos-complement and the OFL output should be used.

The CO output is used as a carry-out in both numbering formats when cascading.

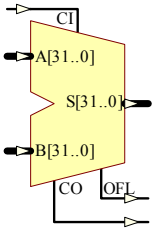
Overflow detection

For unsigned binary operation, CO is used for overflow detection. CO goes High when the sum result (S) goes beyond the unsigned binary boundary. For example, if component ADD4 is used to add 8 (1000) and 9 (1001) together, the resulting sum will be 17 (1 0001), which is out of the 4-bit unsigned binary range, thus CO will be 1. OFL is ignored in unsigned binary operations.

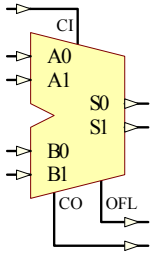
The following shows the unsigned binary range for the different ADD types:

ADD Type	Numbering System	Number Range
ADD2	2-bit unsigned binary	0 to 3
ADD4	4-bit unsigned binary	0 to 15
ADD8	8-bit unsigned binary	0 to 255
ADD16	16-bit unsigned binary	0 to 65535
ADD32	32-bit unsigned binary	0 to 4294967295

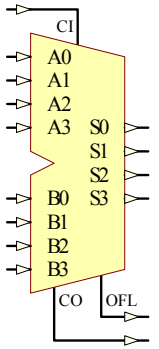
For two's complement operation, OFL is used for overflow detection. When the sum result goes beyond the two's complement boundary, the OFL output goes High. For example, if component ADD4 is used to add 4 (0100) and 5 (0101) together, the resulting sum will be 9 (1001), which is out of the 4-bit twos-complement range because the binary value of 1001 is interpreted as -7 in the 4-bit twos-complement



ADD32B



ADD2S



ADD4S

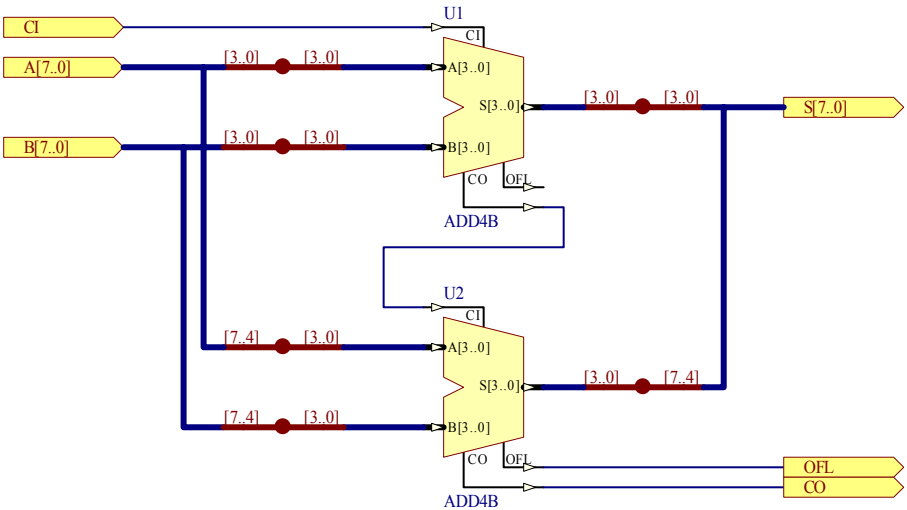
system and thus OFL = 1. CO is ignored in two's complement operation.

The following shows the twos-complement range for the different ADD types:

ADD Type	Numbering System	Number Range
ADD2	2-bit twos-complement	-2 to +1
ADD4	4-bit twos-complement	-8 to + 7
ADD8	8-bit twos-complement	-128 to +127
ADD16	16-bit twos-complement	-32768 to +32767
ADD32	32-bit twos-complement	-2147483648 to +2147483647

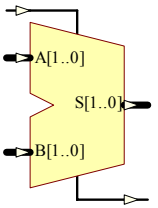
"Carry-out" and Cascading

For cascading purposes, CO is used as a "carry-out" irrespective of the numbering format used. When cascading two or more adders together to create a larger component, the CO output from the upper level adder is connected to CI of the next level. OFL from the upper level is ignored, but the last OFL can still be used as overflows for two's complement operation. The following example demonstrates how to create an 8-bit adder from 2 ADD4 components:



ADDF2, 4, 8, 16, 32

Cascadable Unsigned Binary Full Adder

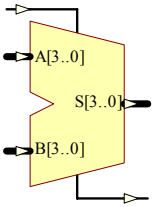


ADDF2B

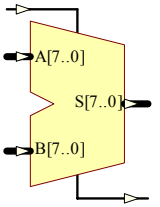
ADDF2, ADDF4, ADDF8, ADDF16 and ADDF32 are, respectively 2-, 4-, 8-, 16- and 32- bit cascadable unsigned binary full adders.

ADDF2, ADDF4, ADDF8, ADDF16 and ADDF32 add two 2-, 4-, 8-, 16-, and 32-bit words (A, B) together respectively and a carry-in (CI) producing 2-, 4-, 8-, 16-, 32-bit binary sum output (S) and carry out (CO). All inputs and outputs of the adders are represented in unsigned binary format.

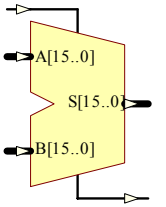
Larger binary adders can be created by connecting CO from the first adder to the CI of the next one.



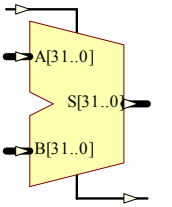
ADDF4B



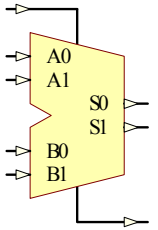
ADDF8B



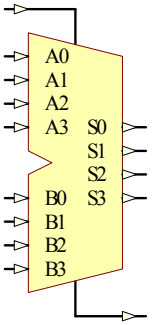
ADDF16B



ADDF32B



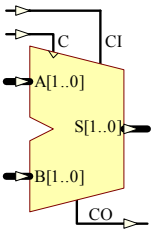
ADDF2S



ADDF4S

ADDFR2, 4, 8, 16, 32

Cascadable Unsigned Binary Registered Full Adder

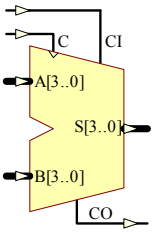


ADDFR2B

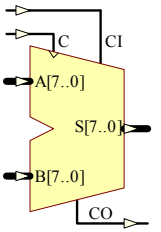
ADDFR2, ADDFR4, ADDFR8, ADDFR16 and ADDFR32 are, respectively 2-, 4-, 8-, 16- and 32- bit cascadable unsigned binary registered full adders.

ADDFR2, ADDFR4, ADDFR8, ADDFR16 and ADDFR32 add two 2-, 4-, 8-, 16-, and 32-bit words (A, B) together respectively and a carry-in (CI) on the rising-edge of clock input (C) producing 2-, 4-, 8-, 16-, 32-bit binary sum output (S) and carry out (CO). All inputs and outputs of the adders are represented in unsigned binary format.

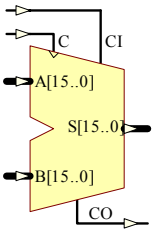
Larger binary adders can be created by connecting CO from the first adder to the CI of the next one.



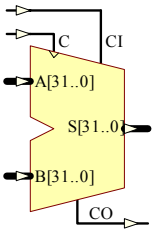
ADDFR4B



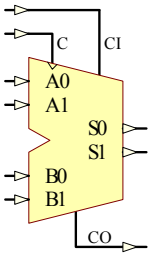
ADDFR8B



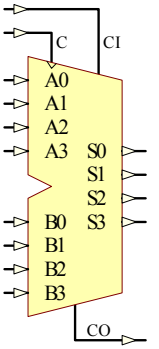
ADDFR16B



ADDFR32B



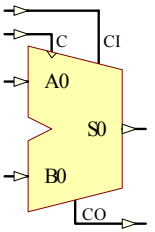
ADDFR2S



ADDFR4S

ADDR1

1-Bit Cascadable Registered Full Adder



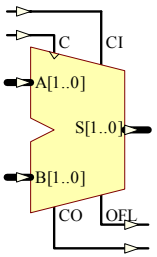
ADDR1

ADDR1 is a 1-Bit registered full adder. The device adds two 1-bit words (A0, B0) and a carry-in (CI) on the rising-edge of the clock input (C), producing a binary sum (S0) output and a carry-out (CO).

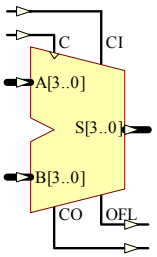
Inputs				Outputs	
C	**CI**	**A0**	**B0**	**S0**	**CO**
↑	0	0	0	0	0
↑	0	0	1	1	0
↑	0	1	0	1	0
↑	0	1	1	0	1
↑	1	0	0	1	0
↑	1	0	1	0	1
↑	1	1	0	0	1
↑	1	1	1	1	1

ADDR2, 4, 8, 16, 32

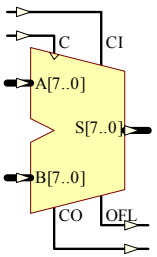
Cascadable Registered Full Adders with Signed and Unsigned Operations



ADDR2B



ADDR4B



ADDR8B

ADDR2, ADDR4, ADDR8, ADDR16 and ADDR32 are, respectively 2-, 4-, 8-, 16- and 32-Bit cascadable registered full adders with signed (twos-complement) and unsigned binary operations. These adders add two input words (A, B) and a carry-in (CI) on the rising-edge of the clock input (C) producing a sum output (S), which can be interpreted as either unsigned binary or two's complement format, and carry-out (CO) and overflow (OFL) outputs.

Unsigned Binary and Two's complement (Signed Binary) operation

ADDR2, ADDR4, ADDR8, ADDR16 and ADDR32 can operate on either, 2-, 4-, 8-, 16- and 32-bit unsigned binary numbers or 2-, 4-, 8-, 16- and 32-bit two's complement numbers respectively.

If the inputs are interpreted as unsigned binary, the result should be interpreted as unsigned binary and the CO output should be used.

If the inputs are interpreted as two's complement, the output should be interpreted as twos-complement and the OFL output should be used.

The CO output is used as a carry-out in both numbering formats when cascading.

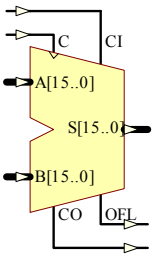
Overflow detection

For unsigned binary operation, CO is used to determine. CO goes High when the sum result (S) goes beyond the unsigned binary boundary. For example, if component ADDR4 is used to add 8 (1000) and 9 (1001) together, the resulting sum will be 17 (1 0001), which is out of the 4-bit unsigned binary range, thus CO will be 1. OFL is ignored in unsigned binary operations.

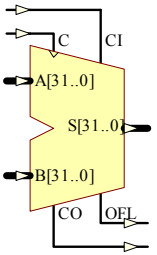
The following shows the unsigned binary range for the different ADDR types:

ADDR Type	Numbering System	Number Range
ADDR2	2-bit unsigned binary	0 to 3
ADDR4	4-bit unsigned binary	0 to 15
ADDR8	8-bit unsigned binary	0 to 255
ADDR16	16-bit unsigned binary	0 to 65535
ADDR32	32-bit unsigned binary	0 to 4294967295

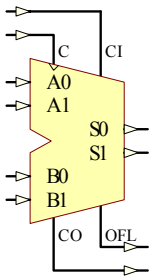
For two's complement operation, OFL is used for overflow detection. When the sum result goes beyond the two's complement boundary, OFL goes High. For example, if component ADDR4 is used to add 4 (0100) and 5 (0101) together, the resulting sum



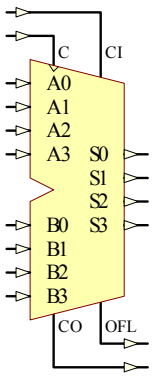
ADDR16B



ADDR32B



ADDR2S



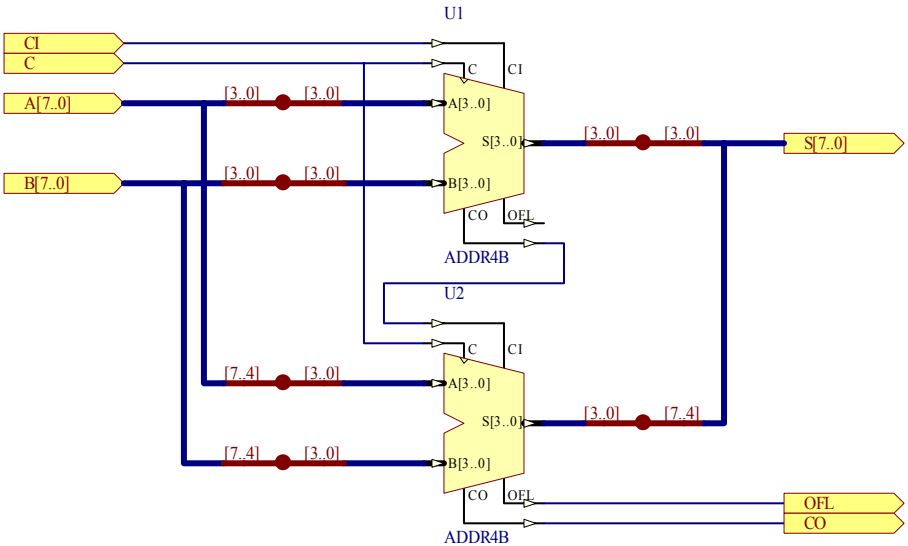
will be 9 (1001), which is out of the 4-bit twos-complement range because the binary value of 1001 is interpreted as -7 in the 4-bit twos-complement system and thus OFL = 1. CO is ignored in two's complement operation.

The following shows the twos-complement range for the different ADDR types:

ADDR Type	Numbering System	Number Range
ADDR2	2-bit twos-complement	-2 to +1
ADDR4	4-bit twos-complement	-8 to + 7
ADDR8	8-bit twos-complement	-128 to +127
ADDR16	16-bit twos-complement	-32768 to +32767
ADDR32	32-bit twos-complement	-2147483648 to +2147483647

"Carry-out" and Cascading

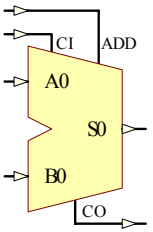
For cascading purposes, CO is used as a "carry-out" irrespective of the numbering format used. When cascading two or more adders together to create a larger component, the CO output from the upper level adder is connected to CI of the next level. OFL from the upper level is ignored, but the last OFL can still be used as overflows for two's complement operation. The following example demonstrates how to create an 8-bit adder from 2 ADDR4 components:



ADDR4S

ADSU1

1-Bit Cascadable Full Adder/Subtracter



ADSU1

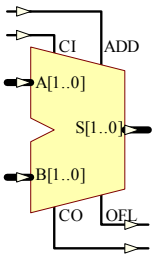
ADSU1 is a 1-Bit cascadable full adder/subtracter. It adds or subtracts two input bits (A0, B0) producing a result (S0) and a carry-out (CO).

When ADD is High, it operates as an adder. When ADD is Low, it operates as a subtracter. CI and CO are active High in add mode and active Low in subtract mode.

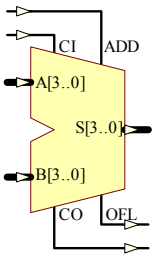
ADD = 1	ADD = 0
$S0 = A0 + B0 + CI$	$S0 = A0 - B0 - \overline{CI}$
CI, CO active HIGH	CI, CO active LOW

ADSU2, 4, 8, 16, 32

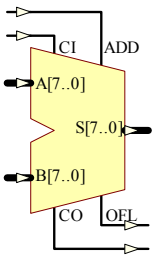
Cascadable Full Adder/Subtracter with Signed and Unsigned Operations



ADSU2B



ADSU4B



ADSU8B

ADSU2, ADSU4, ADSU8, ADSU16 and ADSU32 are, respectively 2-, 4-, 8-, 16 and 32-Bit cascadable full adders and full subtracters with signed (twos-complement) and unsigned binary operations.

Add and Subtract Mode

When ADD = 1, two words (A and B) are added with a carry-in (CI), producing a sum output (S), carry-out (CO) and overflow (OFL). CI and CO are active-High in add mode.

When ADD = 0, B and CI are subtracted from A, producing a result (S), a borrow-out (CO) and an overflow (OFL). CI and CO are active-Low in subtract mode and act as Borrows.

OFL is active High in both add and subtract mode for overflow detection in two's complement numbering format.

ADD = 1	ADD = 0
$S = A + B + CI$	$S = A - B - \overline{CI}$
CI, CO active HIGH	CI, CO active LOW
OFL active HIGH	

Unsigned Binary and Two's complement (Signed Binary) operation

ADSU2, ADSU4, ADSU8, ADSU16 and ADSU32 can operate on either, 2-, 4-, 8-, 16- and 32-bit unsigned binary numbers or 2-, 4-, 8-, 16- and 32-bit two's complement numbers respectively.

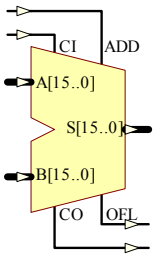
If the inputs are interpreted as unsigned binary, the result should be interpreted as unsigned binary and the CO output should be used.

If the inputs are interpreted as two's complement, the output should be interpreted as twos-complement and the OFL output should be used.

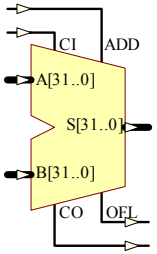
The CO output is used as a carry-out in both numbering formats when cascading.

Overflow detection

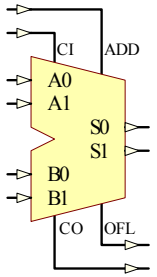
In unsigned binary operation, CO is used to determine overflow. CO goes High when the sum result (S) goes beyond the unsigned binary boundary. For example, if component ADSU4 is used to add 8 (1000) and 9 (1001) together, the resulting sum will be 17 (1 0001), which is out of the 4-bit unsigned binary



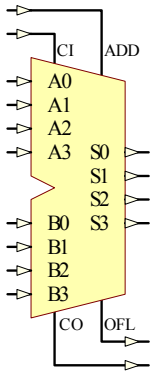
ADSU16B



ADSU32B



ADSU2S



range, thus CO will be 1. Again, CO is active High in add mode and active Low in subtract mode. Also OFL is ignored in unsigned binary operations.

The unsigned binary ranges of the available ADSU's are:

ADSU Type	Numbering System	Number Range
ADSU2	2-bit unsigned binary	0 to 3
ADSU4	4-bit unsigned binary	0 to 15
ADSU8	8-bit unsigned binary	0 to 255
ADSU16	16-bit unsigned binary	0 to 65535
ADSU32	32-bit unsigned binary	0 to 4294967295

For two's complement operation, OFL is used as overflow detection. If an adding or subtraction operation result exceeds the Two's complement range, OFL output goes High. For example, if component ADSU4 is used to add 4 (0100) and 5 (0101) together, the resulting sum will be 9 (1001), which is out of the 4-bit twos-complement range because the binary value of 1001 is interpreted as -7 in the 4-bit twos-complement system and thus OFL = 1. OFL is active High in both add or subtract mode.

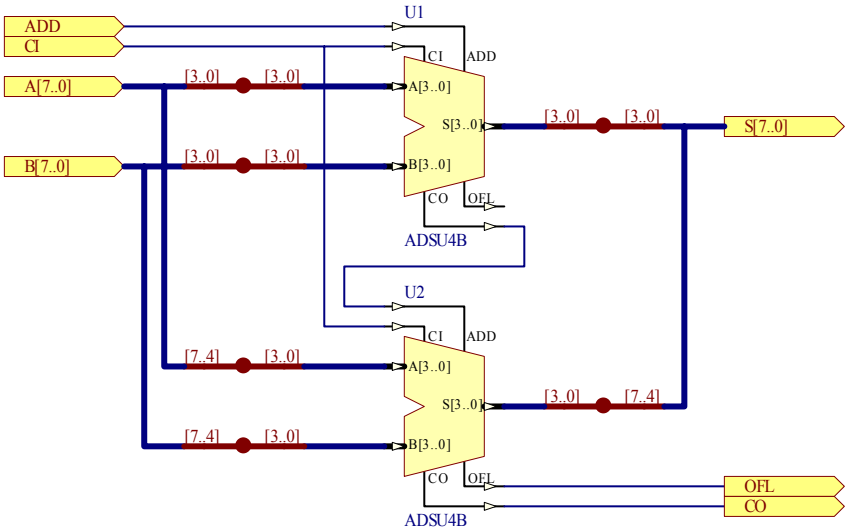
The twos-complement ranges of the available ADSU are:

ADSU Type	Numbering System	Number Range
ADSU2	2-bit twos-complement	-2 to +1
ADSU4	4-bit twos-complement	-8 to + 7
ADSU8	8-bit twos-complement	-128 to +127
ADSU16	16-bit twos-complement	-32768 to +32767
ADSU32	32-bit twos-complement	-2147483648 to +2147483647

"Carry-out","Borrow-out" and Cascading

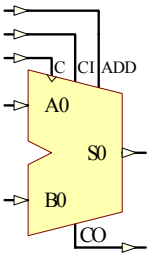
For cascading purposes, CO is used as a "carry-out" or "borrow-out" irrespective of the numbering format used. When cascading two or more ADSU's together to create a larger device; CO from the upper ADSU device is connected to CI of the next device. OFL from the upper level is ignored, but the last OFL can still be used as overflows for two's complement operation. The following example demonstrates how to create an 8-bit adder from 2 ADSU components:

ADSU4S


ADD
CI
A[7..0]
B[7..0]
U1
CI
ADD
A[3..0]
S[3..0]
B[3..0]
CO
OFL
ADSU4B
U2
CI
ADD
A[3..0]
S[3..0]
B[3..0]
CO
OFL
ADSU4B
[3..0]
[3..0]
[7..4]
S[7..0]
OFL
CO

ADSUR1

1-Bit Cascadable Registered Full Adder/Subtracter



ADSUR1

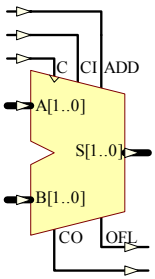
ADSUR1 is a 1-Bit cascadable registered full adder/subtracter. It adds or subtracts two input bits (A0, B0) with a carry-in (CI) during the Low-to-High clock (C) transition, producing a result (S0) and a carry-out (CO).

When ADD is High, it operates as an adder. When ADD is Low, it operates as a subtracter. CI and CO are active High in add mode and active Low in subtract mode.

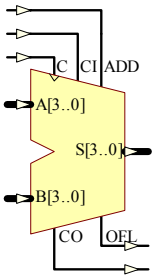
ADD = 1	ADD = 0
$S0 = A0 + B0 + CI$	$S0 = A0 - B0 - \overline{CI}$
CI, CO active HIGH	CI, CO active LOW

ADSUR2, 4, 8, 16, 32

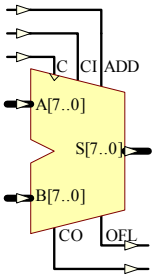
Cascadable Registered Full Adder/Subtracter with Signed and Unsigned Operations



ADSUR2B



ADSUR4B



ADSUR8B

ADSUR2, ADSUR4, ADSUR8, ADSUR16 and ADD32 are, respectively 2-, 4-, 8-, 16 and 32-Bit cascadable registered full adders and full subtracters with signed (twos-complement) and unsigned binary operations.

Add and Subtract Mode

This device is synchronous with clock input (C); calculation occurs during the Low-to-High clock transition.

When ADD = 1, two words (A and B) are added with a carry-in (CI), producing a sum output (S), carry-out (CO) and overflow (OFL). CI and CO are active-High in add mode.

When ADD = 0, B and CI are subtracted from A producing a result (S), borrow (CO) and overflow (OFL). CI and CO are active-Low in subtract mode and act as Borrows.

OFL is active High in both add and subtract mode for overflow detection in two's complement numbering format.

ADD = 1.	ADD = 0.
$S = A + B + CI$	$S = A - B - \overline{CI}$
CI, CO active HIGH	CI, CO active LOW
OFL active HIGH	

Unsigned Binary and Two's complement (Signed Binary) operation

ADSUR2, ADSUR4, ADSUR8, ADSUR16 and ADD32 can operate on either, 2-, 4-, 8-, 16- and 32-bit unsigned binary numbers or 2-, 4-, 8-, 16- and 32-bit two's complement numbers respectively.

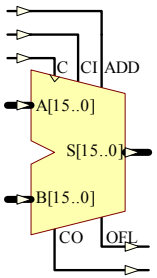
If the inputs are interpreted as unsigned binary, the result should be interpreted as unsigned binary and the CO output should be used.

If the inputs are interpreted as two's complement, the output should be interpreted as twos-complement and the OFL output should be used.

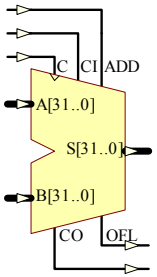
The CO output is used as a carry-out in both numbering formats when cascading.

Overflow detection

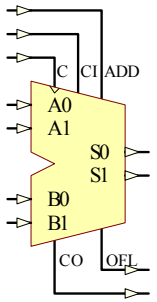
In unsigned binary operation, CO is used to determine overflow. CO goes High when the sum result (S) goes beyond the unsigned binary boundary. For



ADSUR16B



ADSUR32B



ADSUR2S

example, if component ADSUR4 is used to add 8 (1000) and 9 (1001) together, the resulting sum will be 17 (1 0001), which is out of the 4-bit unsigned binary range, thus CO will be 1. Again, CO is active High in add mode and active Low in subtract mode. Also OFL is ignored in unsigned binary operations.

The unsigned binary ranges of the available ADSUR's are:

ADSUR Type	Numbering System	Number Range
ADSUR2	2-bit unsigned binary	0 to 3
ADSUR4	4-bit unsigned binary	0 to 15
ADSUR8	8-bit unsigned binary	0 to 255
ADSUR16	16-bit unsigned binary	0 to 65535
ADSUR32	32-bit unsigned binary	0 to 4294967295

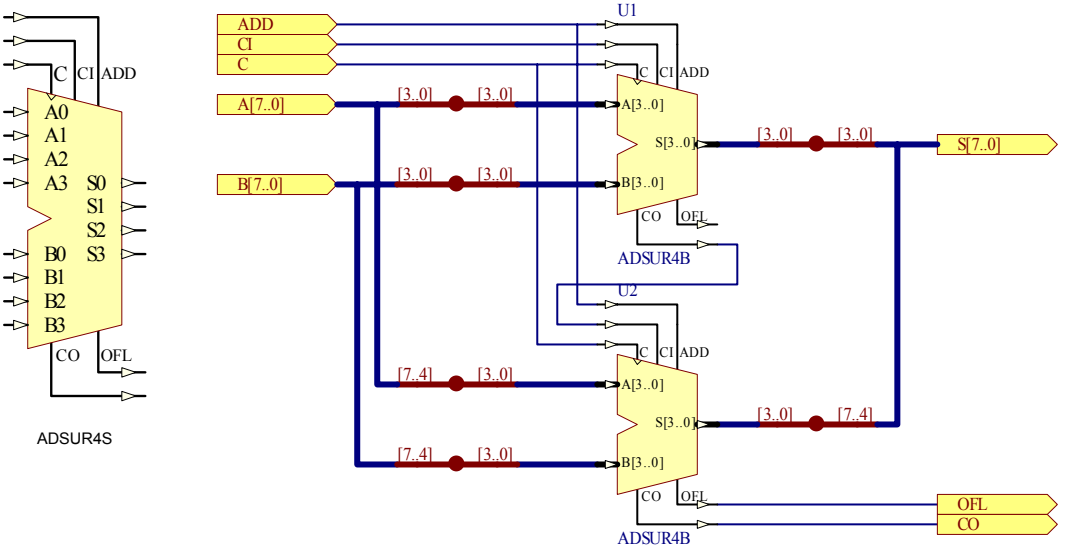
For two's complement operation, OFL is used as overflow detection. If an adding or subtraction operation result exceeds the Two's complement range, OFL output goes High. For example, if component ADSUR4 is used to add 4 (0100) and 5 (0101) together, the resulting sum will be 9 (1001), which is out of the 4-bit twos-complement range because the binary value of 1001 is interpreted as -7 in the 4-bit twos-complement system and thus OFL = 1. OFL is active High in both add or subtract mode.

The twos-complement ranges of the available ADSUR are:

ADSUR Type	Numbering System	Number Range
ADSUR2	2-bit twos-complement	-2 to +1
ADSUR4	4-bit twos-complement	-8 to + 7
ADSUR8	8-bit twos-complement	-128 to +127
ADSUR16	16-bit twos-complement	-32768 to +32767
ADSUR32	32-bit twos-complement	-2147483648 to +2147483647

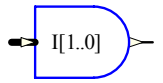
"Carry-out", "Borrow-out" and Cascading

For cascading purposes, CO is used as a "carry-out" or "borrow-out" irrespective of the numbering format used. When cascading two or more ADSUR's together to create a larger device; CO from the upper ADSUR device is connected to CI of the next device. OFL from the upper level is ignored, but the last OFL can still be used as overflows for two's complement operation. The following example demonstrates how to create an 8-bit adder from 2 ADSUR components:

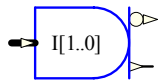

C
CI
ADD
A0
A1
A2
A3
S0
S1
S2
S3
B0
B1
B2
B3
CO
OFL
ADSUR4S
ADD
CI
C
A[7..0]
B[7..0]
U1
[3..0]
[3..0]
A[3..0]
S[3..0]
B[3..0]
CO
OFL
ADSUR4B
S[7..0]
U2
[7..4]
[3..0]
[7..4]
[3..0]
OFL
CO

AND2 – 32

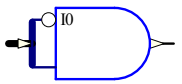
AND Gates



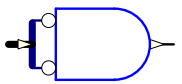
AND2B



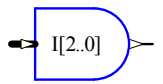
AND2DB



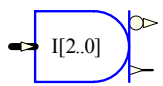
AND2N1B



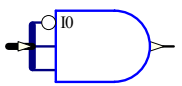
AND2N2B



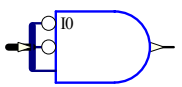
AND3B



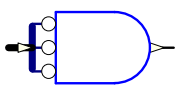
AND3DB



AND3N1B



AND3N2B



AND3N3B

AND Gates provide a variety of AND functions, ranging from 2 to 32 inverted or non-inverted Inputs with Single or Dual output.

AND*n* - Non-Inverted input AND Gates

n is input bit length, *n* = 2, 3, 4, 5, 6, 7, 8, 9, 12, 16, 32

Inputs			Output
I0	**…**	**I*n*-1**	**O**
1	1	1	1
0	x	x	0
x	0	x	0
x	x	0	0

AND*n*N*m* - Inverted input AND Gates

n is input bit length, *m* is number of inverted input.

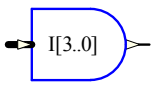
n, m = 2, 3, 4, 5, *m* <= *n*.

Inputs						Output
I0	**…**	**I*m-1***	**I*m***	**…**	**I*n-1***	**O**
0	0	0	1	1	1	1
1	x	x	x	x	x	0
x	1	x	x	x	x	0
x	x	1	x	x	x	0
x	x	x	0	x	x	0
x	x	x	x	0	x	0
x	x	x	x	x	0	0

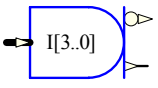
AND*n*D - Dual Output AND Gates

n is input bit length, *n* = 2, 3, 4

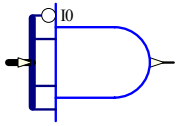
Inputs			Output	
I0	**…**	**I*n-1***	**Y**	**YN**
1	1	1	1	0
0	x	x	0	1
x	0	x	0	1
x	x	0	0	1



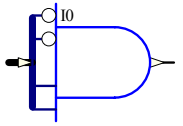
AND4B



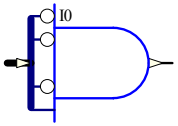
AND4DB



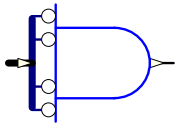
AND4N1B



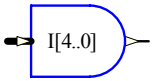
AND4N2B



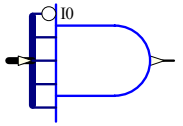
AND4N3B



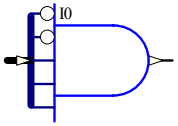
AND4N4B



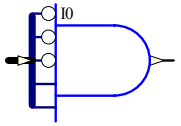
AND5B



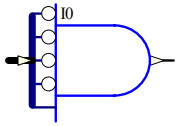
AND5N1B



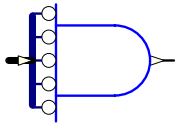
AND5N2B



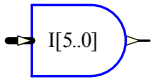
AND5N3B



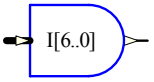
AND5N4B



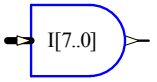
AND5N5B



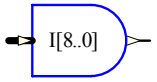
AND6B



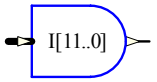
AND7B



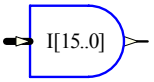
AND8B



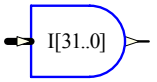
AND9B



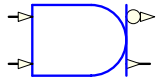
AND12B



AND16B



AND32B



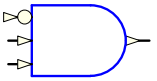
AND2DS

AND2N1S

AND2N2S

AND2S

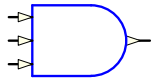
AND3DS



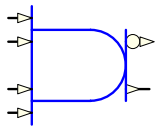
AND3N1S

AND3N2S

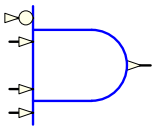
AND3N3S



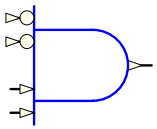
AND3S



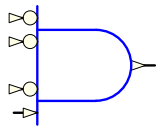
AND4DS



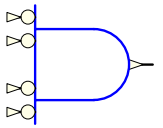
AND4N1S



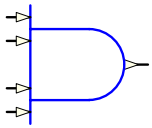
AND4N2S



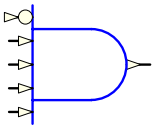
AND4N3S



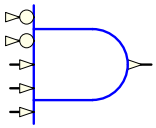
AND4N4S



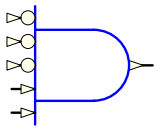
AND4S



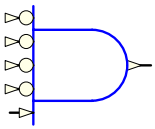
AND5N1S



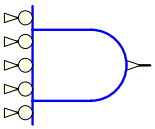
AND5N2S



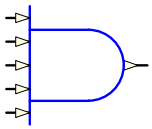
AND5N3S



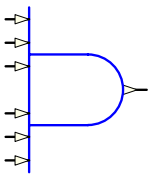
AND5N4S



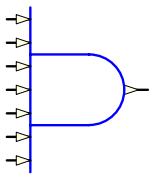
AND5N5S



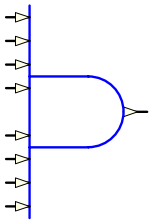
AND5S



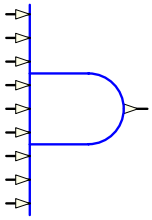
AND6S



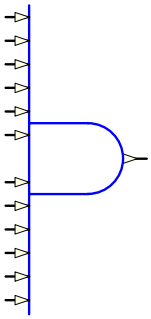
AND7S



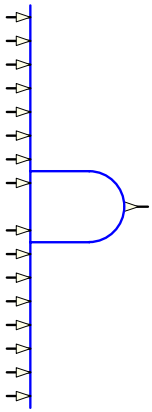
AND8S



AND9S



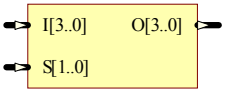
AND12S



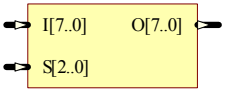
AND16S

BRLSHFT4, 8, 16, 32

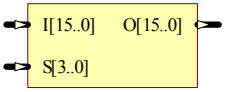
Barrel Shifter



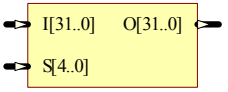
BRLSHFT4B



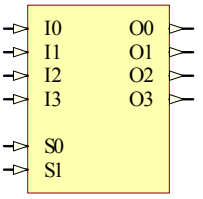
BRLSHFT8B



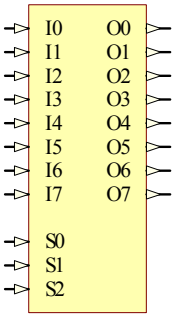
BRLSHFT16B



BRLSHFT32B



BRLSHFT4S



BRLSHFT8S

BRLSHFT4 - 4-bit barrel shifters

Rotate four inputs (I3 – I0) up to four places. The control inputs (S1 and S0) determine the number of positions, from one to four, that the data is rotated. The four outputs (O3 – O0) reflect the shifted data inputs.

BRLSHFT8 - 8-bit barrel shifters

Rotate the eight inputs (I7 – I0) up to eight places. The control inputs (S2 – S0) determine the number of positions, from one to eight, that the data is rotated. The eight outputs (O7 – O0) reflect the shifted data inputs.

BRLSHFT16 - 16-bit barrel shifters

Rotate the sixteen inputs (I15 – I0) up to sixteen places. The control inputs (S3 – S0) determine the number of positions, from one to sixteen, that the data is rotated. The sixteen outputs (O15 – O0) reflect the shifted data inputs.

BRLSHFT32 - 32-bit barrel shifters

Rotate the thirty-two inputs (I31 – I0) up to thirty-two places. The control inputs (S4 – S0) determine the number of positions, from one to thirty-two, that the data is rotated. The thirty-two outputs (O31 – O0) reflect the shifted data inputs.

BRLSHFT4 - 4-bit barrel shifters

Inputs		Outputs			
S1	**S0**	**O3**	**O2**	**O1**	**O0**
0	0	I3	I2	I1	I0
0	1	I0	I3	I2	I1
1	0	I1	I0	I3	I2
1	1	I2	I1	I0	I3

BRLSHFT8 - 8-bit barrel shifters

Inputs			Outputs							
S2	**S1**	**S0**	**O7**	**O6**	**O5**	**O4**	**O3**	**O2**	**O1**	**O0**
0	0	0	I7	I6	I5	I4	I3	I2	I1	I0
0	0	1	I0	I7	I6	I5	I4	I3	I2	I1
0	1	0	I1	I0	I7	I6	I5	I4	I3	I2
0	1	1	I2	I1	I0	I7	I6	I5	I4	I3
1	0	0	I3	I2	I1	I0	I7	I6	I5	I4
1	0	1	I4	I3	I2	I1	I0	I7	I6	I5
1	1	0	I5	I4	I3	I2	I1	I0	I7	I6
1	1	1	I6	I5	I4	I3	I2	I1	I0	I7

BRLSHFT16 - 16-bit barrel shifters

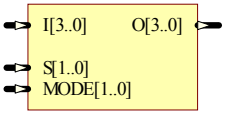
Inputs				Outputs									
S3	**S2**	**S1**	**S0**	**O15**	**O14**	**O13**	**…**	**O8**	**O7**	**…**	**O2**	**O1**	**O0**
0	0	0	0	I15	I14	I13	…	I8	I7	…	I2	I1	I0
0	0	0	1	I0	I15	I14	…	I9	I8	…	I3	I2	I1
0	0	1	0	I1	I0	I15	…	I10	I9	…	I4	I3	I2
0	0	1	1	I2	I1	I0	…	I11	I10	…	I5	I4	I3
0	1	0	0	I3	I2	I1	…	I12	I11	…	I6	I5	I4
0	1	0	1	I4	I3	I2	…	I13	I12	…	I7	I6	I5
0	1	1	0	I5	I4	I3	…	I14	I13	…	I8	I7	I6
0	1	1	1	I6	I5	I4	…	I15	I14	…	I9	I8	I7
1	0	0	0	I7	I6	I5	…	I0	I15	…	I10	I9	I8
1	0	0	1	I8	I7	I6	…	I1	I0	…	I11	I10	I9
1	0	1	0	I9	I8	I7	…	I2	I1	…	I12	I11	I10
1	0	1	1	I10	I9	I8	…	I3	I2	…	I13	I12	I11
1	1	0	0	I11	I10	I9	…	I4	I3	…	I14	I13	I12
1	1	0	1	I12	I11	I10	…	I5	I4	…	I15	I14	I13
1	1	1	0	I13	I12	I11	…	I6	I5	…	I0	I15	I14
1	1	1	1	I14	I13	I12	…	I7	I6	…	I1	I0	I15

BRLSHFT32 - 32-bit barrel shifters

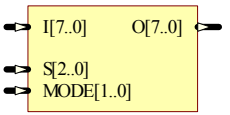
Inputs					Outputs									
S4	**S3**	**S2**	**S1**	**S0**	**O31**	**O30**	**O29**	**…**	**O16**	**O15**	**…**	**O2**	**O1**	**O0**
0	0	0	0	0	I31	I30	I29	…	I16	I15	…	I2	I1	I0
0	0	0	0	1	I0	I31	I30	…	I17	I16	…	I3	I2	I1
0	0	0	1	0	I1	I0	I31	…	I18	I17	…	I4	I3	I2
0	0	0	1	1	I2	I1	I0	…	I19	I18	…	I5	I4	I3
0	0	1	0	0	I3	I2	I1	…	I20	I19	…	I6	I5	I4
0	0	1	0	1	I4	I3	I2	…	I21	I20	…	I7	I6	I5
0	0	1	1	0	I5	I4	I3	…	I22	I21	…	I8	I7	I6
0	0	1	1	1	I6	I5	I4	…	I23	I22	…	I9	I8	I7
0	1	0	0	0	I7	I6	I5	…	I24	I23	…	I10	I9	I8
0	1	0	0	1	I8	I7	I6	…	I25	I24	…	I11	I10	I9
0	1	0	1	0	I9	I8	I7	…	I26	I25	…	I12	I11	I10
0	1	0	1	1	I10	I9	I8	…	I27	I26	…	I13	I12	I11
0	1	1	0	0	I11	I10	I9	…	I28	I27	…	I14	I13	I12
0	1	1	0	1	I12	I11	I10	…	I29	I28	…	I15	I14	I13
0	1	1	1	0	I13	I12	I11	…	I30	I29	…	I16	I15	I14
0	1	1	1	1	I14	I13	I12	…	I31	I30	…	I17	I16	I15
1	0	0	0	0	I15	I14	I13	…	I0	I31	…	I18	I17	I16
1	0	0	0	1	I16	I15	I14	…	I1	I0	…	I19	I18	I17
1	0	0	1	0	I17	I16	I15	…	I2	I1	…	I20	I19	I18
1	0	0	1	1	I18	I17	I16	…	I3	I2	…	I21	I20	I19
1	0	1	0	0	I19	I18	I17	…	I4	I3	…	I22	I21	I20
1	0	1	0	1	I20	I19	I18	…	I5	I4	…	I23	I22	I21
1	0	1	1	0	I21	I20	I19	…	I6	I5	…	I24	I23	I22
1	0	1	1	1	I22	I21	I20	…	I7	I6	…	I25	I24	I23
1	1	0	0	0	I23	I22	I21	…	I8	I7	…	I26	I25	I24
1	1	0	0	1	I24	I23	I22	…	I9	I8	…	I27	I26	I25
1	1	0	1	0	I25	I24	I23	…	I10	I9	…	I28	I27	I26
1	1	0	1	1	I26	I25	I24	…	I11	I10	…	I29	I28	I27
1	1	1	0	0	I27	I26	I25	…	I12	I11	…	I30	I29	I28
1	1	1	0	1	I28	I27	I26	…	I13	I12	…	I31	I30	I29
1	1	1	1	0	I29	I28	I27	…	I14	I13	…	I0	I31	I30
1	1	1	1	1	I30	I29	I28	…	I15	I14	…	I1	I0	I31

BRLSHFTM4, 8, 16, 32

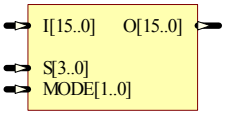
Fill Mode Bi-Directional Barrel Shifter



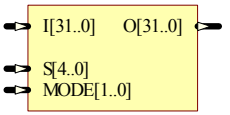
BRLSHFTM4B



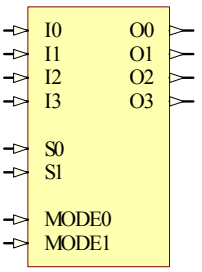
BRLSHFTM8B



BRLSHFTM16B



BRLSHFTM32B



BRLSHFTM4S

BRLSHFTM are 4, 8, 16 and 32 bit fill mode bi-directional barrel shifters. Direction and fill mode is chosen by using the shift mode (MODE) input.

When MODE input is set to “01”, data from (I) input slice is shifted to the left (O) output slice controlled by select (S) inputs. The trailing output (O) slice is filled with 0.

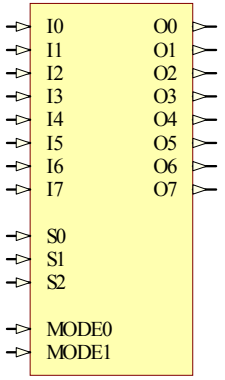
When MODE input is set to “10”, data from (I) input slice is shifted to the right (O) output slice controlled by select (S) inputs. The trailing output (O) slice is filled with 0.

When MODE input is set to “11”, data from (I) input slice is shifted to the right (O) output slice controlled by select (S) inputs. The trailing output (O) slice is filled with 1.

The following truth table describes the behavior of 4-Bit fill mode bi-directional barrel shifter.

BRLSHFTM4 - 4-bit Fill Mode Bi-Directional Barrel Shifter

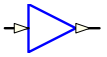
Functions	Inputs							Outputs			
	MODE	S1	S0	I0	I1	I2	I3	O0	O1	O2	O3
Shift Nothing	00	x	x	x	x	x	x	0	0	0	0
Unsigned Left Shift	01	0	0	a	b	c	d	a	b	c	d
		0	1	a	b	c	d	b	c	d	0
		1	0	a	b	c	d	c	d	0	0
		1	1	a	b	c	d	d	0	0	0
Unsigned Right Shift	10	0	0	a	b	c	d	a	b	c	d
		0	1	a	b	c	d	0	a	b	c
		1	0	a	b	c	d	0	0	a	b
		1	1	a	b	c	d	0	0	0	a
Signed Right Shift	11	0	0	a	b	c	d	a	b	c	d
		0	1	a	b	c	d	1	a	b	c
		1	0	a	b	c	d	1	1	a	b
		1	1	a	b	c	d	1	1	1	a



BRLSHFTM8S

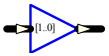
BUF – BUF32

General Purpose (Non-Inverting) Buffer

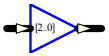


BUF

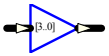
Single or multiple, general purpose, non-inverting buffer, where the output is always equal the input.



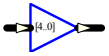
BUF2B



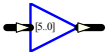
BUF3B



BUF4B



BUF5B



BUF6B

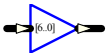
BUF

Input	Output
I	O
1	1
0	0

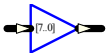
BUF2-32

Inputs			Outputs		
I0	…	In-1	O0	…	On-1
1	…	1	1	…	1
1	…	0	1	…	0
0	…	1	0	…	1
0	…	0	0	…	0

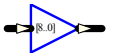
n = 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, 32



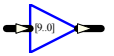
BUF7B



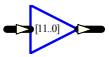
BUF8B



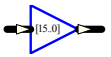
BUF9B



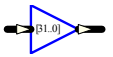
BUF10B



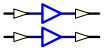
BUF12B



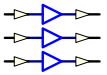
BUF16B



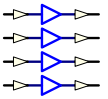
BUF32B



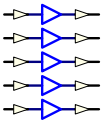
BUF2S



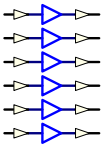
BUF3S



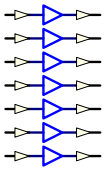
BUF4S



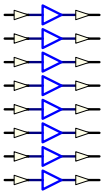
BUF5S



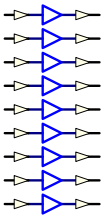
BUF6S



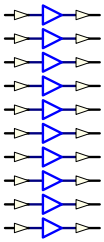
BUF7S



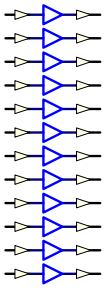
BUF8S



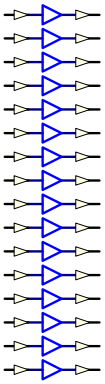
BUF9S



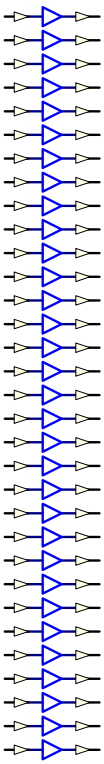
BUF10S



BUF12S



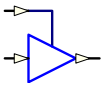
BUF16S



BUF32S

BUFE – BUFE32

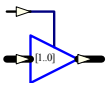
3-State Buffers with Active High Enable



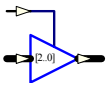
BUFE

Single or multiple 3-state Buffers with common active High Enable (E).

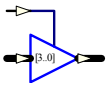
When E = 0, the output goes into the High-impendence (Z) state. When E = 1, output is same as the input.



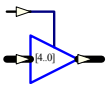
BUFE2B



BUFE3B



BUFE4B



BUFE5B

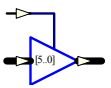
BUFE

Inputs		Output
E	**I**	**O**
0	x	Z
1	1	1
1	0	0

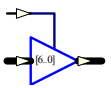
BUFE2 - 32

Inputs				Outputs		
E	**I0**	**…**	**In-1**	**O0**	**…**	**On-1**
0	x	…	x	Z	…	Z
1	1	…	1	1	…	1
1	1	…	0	1	…	0
1	0	…	1	0	…	1
1	0	…	0	0	…	0

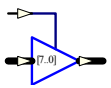
n = 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, 32



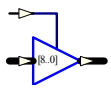
BUFE6B



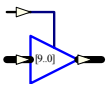
BUFE7B



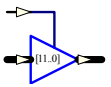
BUFE8B



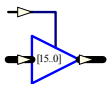
BUFE9B



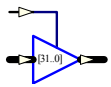
BUFE10B



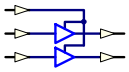
BUFE12B



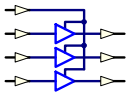
BUFE16B



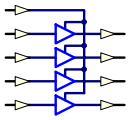
BUFE32B



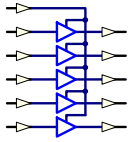
BUFE2S



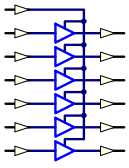
BUFE3S



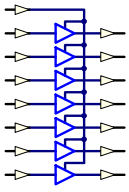
BUFE4S



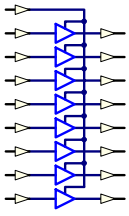
BUFE5S



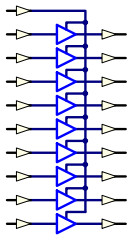
BUFE6S



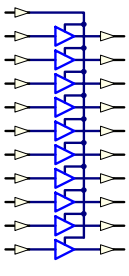
BUFE7S



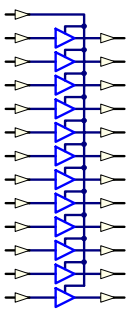
BUFE8S



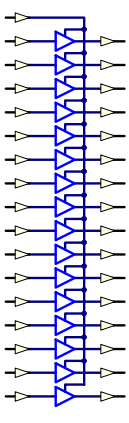
BUFE9S



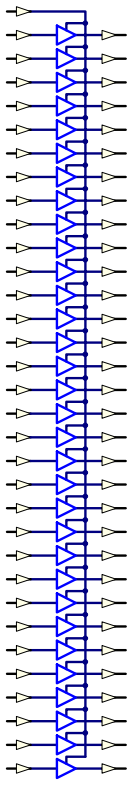
BUFE10S



BUFE12S



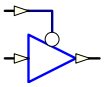
BUFE16S



BUFE32S

BUFT – BUFT32

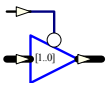
3-State Buffers with Active Low Enable



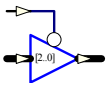
BUFT

Single or multiple 3-state Buffers with active Low Enable (T).

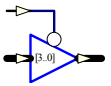
When T = 1, output goes into the High-impendence (Z) state. When T = 0, output is the same as the input.



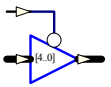
BUFT2B



BUFT3B



BUFT4B



BUFT5B

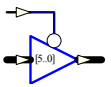
BUFT

Inputs		Output
T	**I**	**O**
1	x	Z
0	1	1
0	0	0

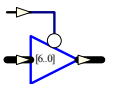
BUFT2-32

Inputs				Outputs		
T	**I0**	**…**	**In-1**	**O0**	**…**	**On-1**
1	x	…	x	Z	…	Z
0	1	…	1	1	…	1
0	1	…	0	1	…	0
0	0	…	1	0	…	1
0	0	…	0	0	…	0

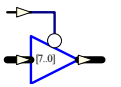
n = 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, 32



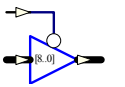
BUFT6B



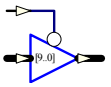
BUFT7B



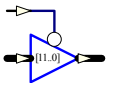
BUFT8B



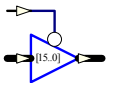
BUFT9B



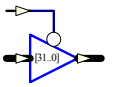
BUFT10B



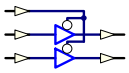
BUFT12B



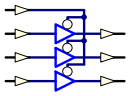
BUFT16B



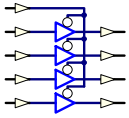
BUFT32B



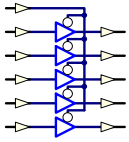
BUFT2S



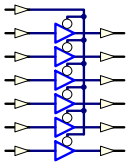
BUFT3S



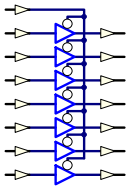
BUFT4S



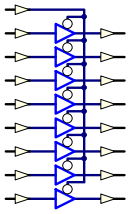
BUFT5S



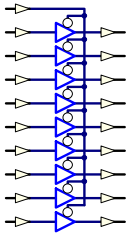
BUFT6S



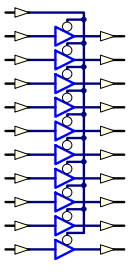
BUFT7S



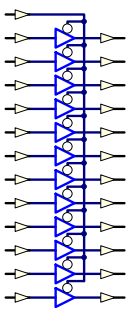
BUFT8S



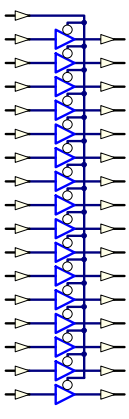
BUFT9S



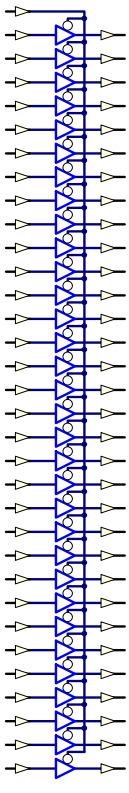
BUFT10S



BUFT12S



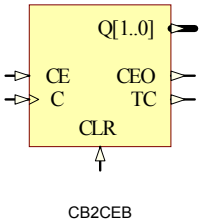
BUFT16S



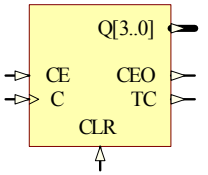
BUFT32S

CB2CE, CB4CE, CB8CE, CB16CE, CB32CE

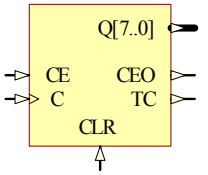
Cascadable Binary Counters with Clock Enable and Asynchronous Clear



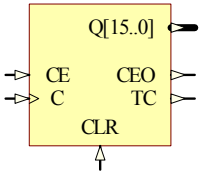
CB2CEB



CB4CEB



CB8CEB



CB16CEB

CB2CE, CB4CE, CB8CE, CB16CE and CB32CE are, respectively 2-, 4-, 8-,16-,32-Bit Cascadable Binary Counters with Clock Enable and Asynchronous Clear.

The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored and all outputs go Low independent of the clock (C) transitions.

The Q outputs increment when clock enable (CE) is High during the Low-to-High clock transition. When CE is Low, clock transitions are ignored and outputs remain unchanged from the previous state.

The terminal count (TC) output is High when all Q outputs are High. The clock enable output (CEO) is High when TC and CE are both High.

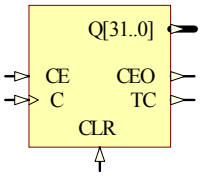
Larger counters can be created by connecting the CEO output of the first stage to the CE input of the next stage and connecting C and CLR inputs in parallel. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

Inputs			Outputs		
CLR	**CE**	**C**	**Qz-Q0**	**TC**	**CEO**
1	x	x	0	0	0
0	0	x	No Chg	No Chg	0
0	1	↑	Inc	TC	CEO

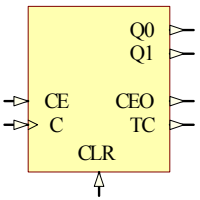
z = 1 for CB2CE; z = 3 for CB4CE; z = 7 for CB8CE; z = 15 for CB16CE; z = 31 for CB32CE

TC = Qz•Q(z-1)•Q(z-2)•...•Q0

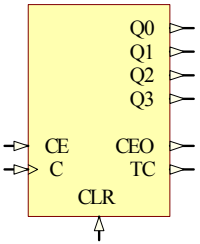
CEO = TC•CE



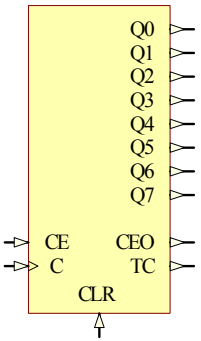
CB32CEB



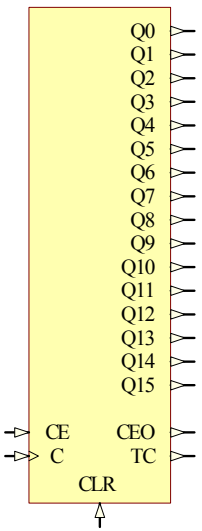
CB2CES



CB4CES



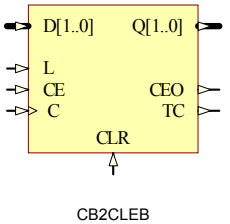
CB8CES



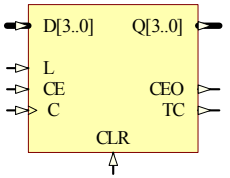
CB16CES

CB2CLE, CB4CLE, CB8CLE, CB16CLE, CB32CLE

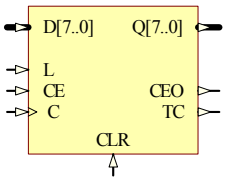
Loadable Cascadable Binary Counters with Clock Enable and Asynchronous Clear



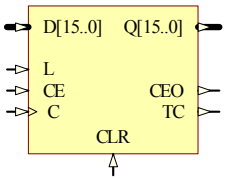
CB2CLEB



CB4CLEB



CB8CLEB



CB16CLEB

CB2CLE, CB4CLE, CB8CLE, CB16CLE, CB32CLE are, respectively 2-, 4-, 8-,16-, 32-Bit Loadable Cascadable Binary Counters with Clock Enable and Asynchronous Clear.

The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored and all outputs go Low independent of the clock (C) transitions.

The data on the D input is loaded into the counter when the load enable input (L) is High during the Low-to-High clock transition, independent of the state of clock enable (CE).

The Q outputs increment when clock enable (CE) is High during the Low-to-High clock transition. When CE is Low, clock transitions are ignored and outputs remain unchanged from the previous state.

The terminal count (TC) output is High when all Q outputs are High. The clock enable output (CEO) is High when TC and CE are both High.

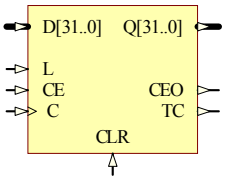
Larger counters can be created by connecting the CEO output of the first stage to the CE input of the next stage and connecting C, L and CLR inputs in parallel. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

Inputs					Outputs		
CLR	**L**	**CE**	**C**	**Dz – D0**	**Qz – Q0**	**TC**	**CEO**
1	x	x	x	x	0	0	0
0	1	x	↑	Dn	Dn	TC	CEO
0	0	0	x	x	No Chg	No Chg	0
0	0	1	↑	x	Inc	TC	CEO

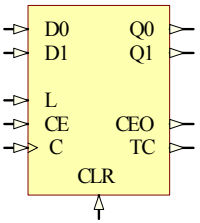
z= 1 for CB2CLE; z = 3 for CB4CLE; z = 7 for CB8CLE; z = 15 for CB16CLE; z = 31 for CB32CLE

TC = Qz•Q(z-1)•Q(z-2)•...•Q0

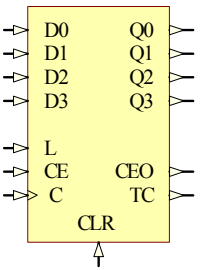
CEO = TC•CE



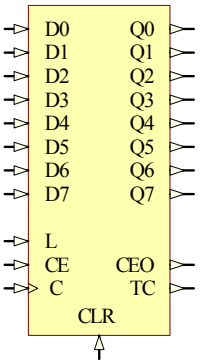
CB32CLEB



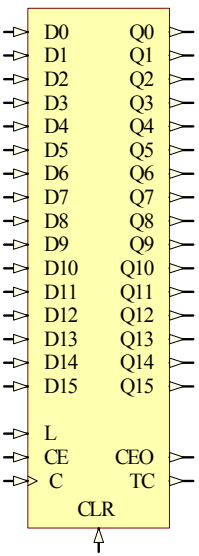
CB2CLES



CB4CLES



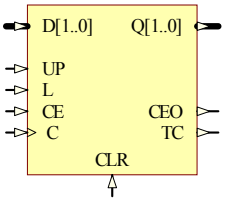
CB8CLES



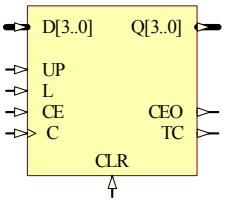
CB16CLES

CB2CLED, CB4CLED, CB8CLED, CB16CLED, CB32CLED

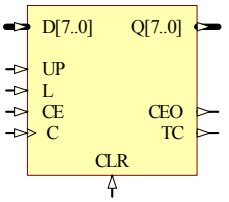
Loadable Cascadable Bidirectional Binary Counters with Clock Enable and Asynchronous Clear



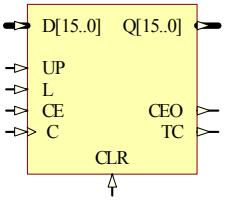
CB2CLEDB



CB4CLEDB



CB8CLEDB



CB16CLEDB

CB2CLED, CB4CLED, CB8CLED, CB16CLED, CB32CLED are, respectively 2-, 4-, 8-, 16-, 32-Bit Loadable Cascadable Bidirectional Binary Counters with Clock Enable and Asynchronous Clear.

The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored and all outputs go Low independent of the clock (C) transitions.

The data on the D input is loaded into the counter when the load enable input (L) is High during the Low-to-High clock transition, independent of the state of clock enable (CE).

The Q outputs decrement when CE is High and UP is Low during the Low-to-High clock transition. The Q outputs increment when CE and UP are both High. When CE is Low, clock transitions are ignored and outputs remain unchanged from the previous state.

For counting up, the terminal count (TC) output is High when all Q outputs are High. For counting down, the TC output is High when all Q outputs and UP are Low.

To cascade counters, the clock enable output (CEO) of each counter is connected to the CE pin of the next stage. The C, UP, L and CLR inputs are connected in parallel. The CEO output is High when TC and CE is High.

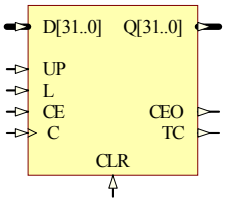
When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

Inputs						Outputs		
CLR	**L**	**CE**	**C**	**UP**	**Dz – D0**	**Qz – Q0**	**TC**	**CEO**
1	x	x	x	x	x	0	0	0
0	1	x	↑	x	Dn	Dn	TC	CEO
0	0	0	x	x	x	No Chg	No Chg	0
0	0	1	↑	1	x	Inc	TC	CEO
0	0	1	↑	0	x	Dec	TC	CEO

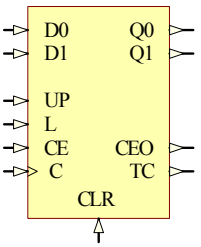
z = 1 for CB2CLED; z = 3 for CB4CLED; z = 7 for CB8CLED; z = 15 for CB16CLED; z = 31 for CB32CLED

TC = (Qz.Q(z-1)....Q0.UP) + (not(Qz).not(Q(z-1))...not(Q0.UP))

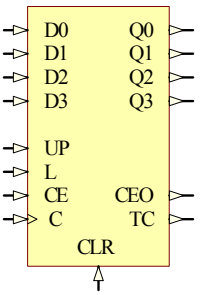
CEO = TC•CE



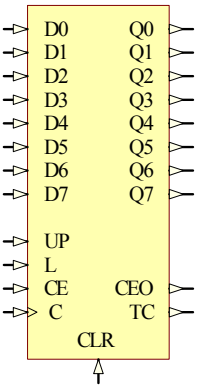
CB32CLEDB



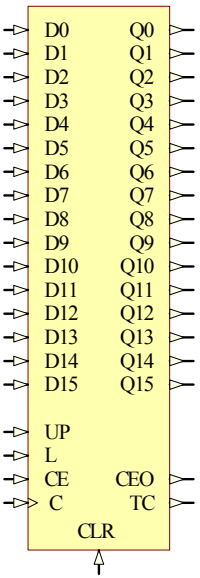
CB2CLEDS



CB4CLEDS



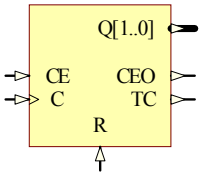
CB8CLEDS



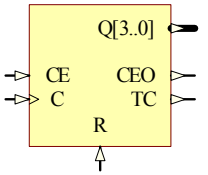
CB16CLEDS

CB2RE, CB4RE, CB8RE, CB16RE, CB32RE

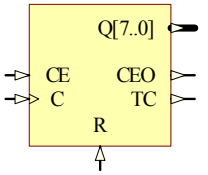
Cascadable Binary Counters with Clock Enable and Synchronous Reset



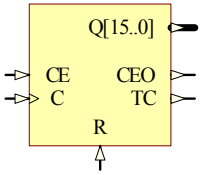
CB2REB



CB4REB



CB8REB



CB16REB

CB2RE, CB4RE, CB8RE, CB16RE, CB32RE are respectively 2-, 4-, 8-, 16-, 32-Bit Cascadable Binary Counters with Clock Enable and Synchronous Reset.

The synchronous reset (R) is the highest priority input. When R is High, all other inputs are ignored and all outputs go Low during the Low-to-High clock (C) transitions.

The Q outputs increment when clock enable (CE) is High during the Low-to-High clock transition. When CE is Low, clock transitions are ignored and outputs remain unchanged from the previous state.

The terminal count (TC) output is High when all Q outputs are High. The clock enable output (CEO) is High when TC and CE are both High.

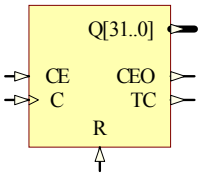
Larger counters can be created by connecting the CEO output of the first stage to the CE input of the next stage and connecting C and CLR inputs in parallel. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

Inputs			Outputs		
R	**CE**	**C**	**Qz – Q0**	**TC**	**CEO**
1	x	↑	0	0	0
0	0	x	No Chg	No Chg	0
0	1	↑	Inc	TC	CEO

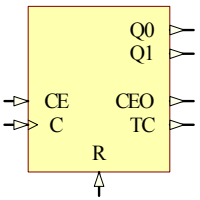
z = 1 for CB2RE; z = 3 for CB4RE; z = 7 for CB8RE; z = 15 for CB16RE; z = 31 for CB32RE

TC = Qz•Q(z-1)•Q(z-2)•...•Q0)

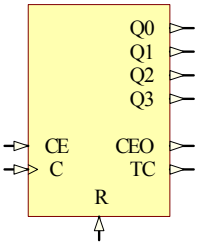
CEO = TC•CE



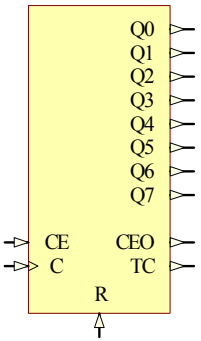
CB32REB



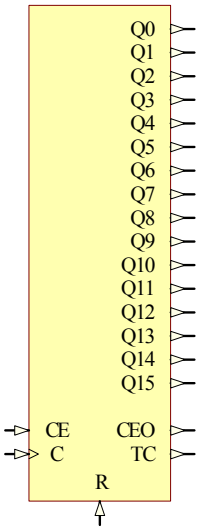
CB2RES



CB4RES



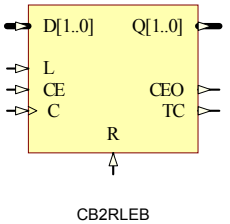
CB8RES



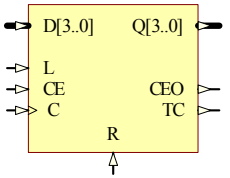
CB16RES

CB2RLE, CB4RLE, CB8RLE, CB16RLE, CB32RLE

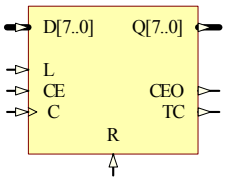
Loadable Cascadable Binary Counters with Clock Enable and Synchronous Reset



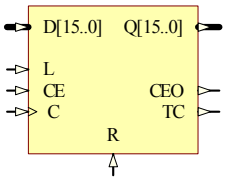
CB2RLEB



CB4RLEB



CB8RLEB



CB16RLEB

CB2RLE, CB4RLE, CB8RLE, CB16RLE, CB32RLE are respectively 2-. 4-. 8-, 16- and 32-Bit Loadable Cascadable Binary Counters with Clock Enable and Synchronous Reset.

The synchronous reset (R) is the highest priority input. When R is High, all other inputs are ignored and all outputs go Low during the Low-to-High clock (C) transition.

The data on the D input is loaded into the counter when the load enable input (L) is High during the Low-to-High clock transition, independent of the state of clock enable (CE).

The Q outputs increment when clock enable (CE) is High during the Low-to-High clock transition. When CE is Low, clock transitions are ignored and outputs remain unchanged from the previous state.

The terminal count (TC) output is High when all Q outputs are High. The clock enable output (CEO) is High when TC and CE are both High.

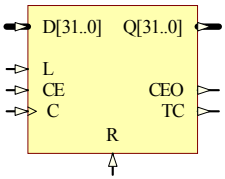
Larger counters can be created by connecting the CEO output of the first stage to the CE input of the next stage and connecting C, L and CLR inputs in parallel. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

Inputs					Outputs		
R	**L**	**CE**	**C**	**Dz – D0**	**Qz – Q0**	**TC**	**CEO**
1	x	x	↑	x	0	0	0
0	1	x	↑	Dn	Dn	TC	CEO
0	0	0	x	x	No Chg	No Chg	0
0	0	1	↑	x	Inc	TC	CEO

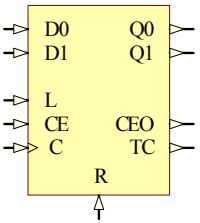
z = 1 for CB2RLE; z = 3 for CB4RLE; z = 7 for CB8RLE; z = 15 for CB16RLE; z = 31 for CB32RLE

TC = Qz•Q(z-1)•Q(z-2)•...•Q0

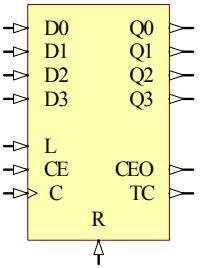
CEO = TC•CE



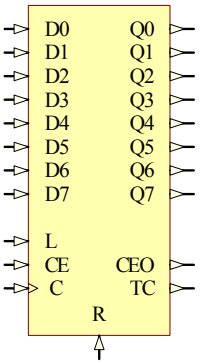
CB32RLEB



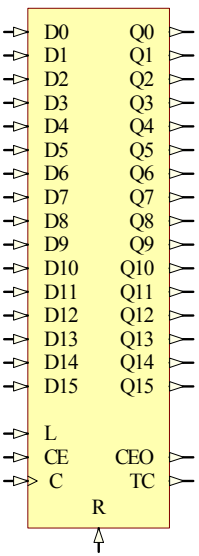
CB2RLES



CB4RLES



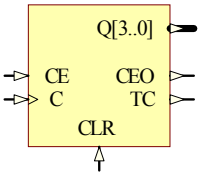
CB8RLES



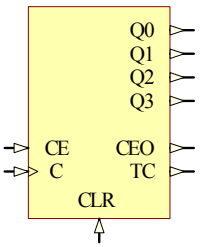
CB16RLES

CD4CE

Cascadable BCD Counter with Clock Enable and Asynchronous Clear



CD4CEB



CD4CES

CD4CE is a cascadable binary-coded-decimal (BCD) counter with clock enable and asynchronous clear.

The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored and all outputs go Low independent of the clock (C) transitions.

The Q outputs increment when clock enable (CE) is High during the Low-to-High clock transition. When CE is Low, clock transitions are ignored and outputs remain unchanged from the previous state.

The terminal count (TC) output is High when Q3 and Q0 are High and Q2 and Q1 are Low. The clock enable output (CEO) is High when TC and CE are both High.

Larger counters can be created by connecting the CEO output of the first stage to the CE input of the next stage and connecting C and CLR inputs in parallel. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

As it is a BCD counter, it counts from decimal 0 to 9 usually, if there is an illegal count (ie. 10, 11, 12, 13, 14, 15) happen, it returns to 0 immediately in the next count, eg. 11 -> 0.

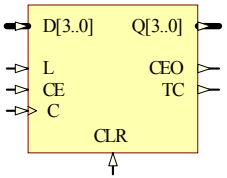
Inputs			Outputs					
CLR	**CE**	**C**	**Q3**	**Q2**	**Q1**	**Q0**	**TC**	**CEO**
1	x	x	0	0	0	0	0	0
0	1	↑	Inc	Inc	Inc	Inc	TC	CEO
0	0	x	No Chg	No Chg	No Chg	No Chg	TC	0
0	1	x	1	0	0	1	1	1

TC = Q3•!Q2•!Q1•Q0

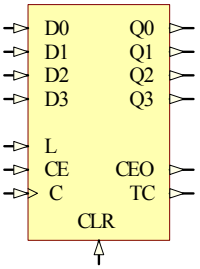
CEO = TC•CE

CD4CLE

Loadable Cascadable BCD Counter with Clock Enable and Asynchronous Clear



CD4CLEB



CD4CLES

CD4CLE is a loadable, cascadable binary-coded-decimal (BCD) counter with clock enable and asynchronous Clear.

The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored and all outputs go Low independent of the clock (C) transitions.

The data on the D input is loaded into the counter when the load enable input (L) is High during the Low-to-High clock transition, independent of the state of clock enable (CE).

The Q outputs increment when clock enable (CE) is High during the Low-to-High clock transition. When CE is Low, clock transitions are ignored and outputs remain unchanged from the previous state.

The terminal count (TC) output is High when Q3 and Q0 are High and Q2 and Q1 are Low. The clock enable output (CEO) is High when TC and CE are both High.

Larger counters can be created by connecting the CEO output of the first stage to the CE input of the next stage and connecting C, L and CLR inputs in parallel. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

As it is a BCD counter, it counts from decimal 0 to 9 usually, if there is an illegal count (ie. 10, 11, 12, 13, 14, 15) happen, it returns to 0 immediately in the next count, eg. 11 -> 0.

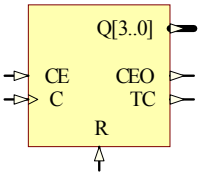
Inputs					Outputs					
CLR	**L**	**CE**	**D3 – D0**	**C**	**Q3**	**Q2**	**Q1**	**Q0**	**TC**	**CEO**
1	x	x	x	x	0	0	0	0	0	0
0	1	x	D3 – D0	↑	D3	D2	D1	D0	TC	CEO
0	0	1	x	↑	Inc	Inc	Inc	Inc	TC	CEO
0	0	0	x	x	No Chg	No Chg	No Chg	No Chg	TC	0
0	0	1	x	x	1	0	0	1	1	1

TC = Q3•!Q2•!Q1•Q0

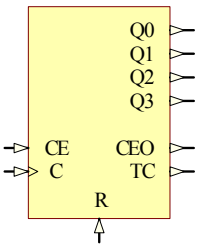
CEO = TC•CE

CD4RE

Cascadable BCD Counter with Clock Enable and Synchronous Reset



CD4REB



CD4RES

CD4RE is a cascadable binary-coded-decimal (BCD) counter with clock enable and synchronous reset.

The synchronous reset (R) is the highest priority input. When R is High, all other inputs are ignored and all outputs go Low during the Low-to-High clock (C) transitions.

The Q outputs increment when clock enable (CE) is High during the Low-to-High clock transition. When CE is Low, clock transitions are ignored and outputs remain unchanged from the previous state.

The terminal count (TC) output is High when Q3 and Q0 are High and Q2 and Q1 are Low. The clock enable output (CEO) is High when TC and CE are both High.

Larger counters can be created by connecting the CEO output of the first stage to the CE input of the next stage and connecting C and CLR inputs in parallel. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

As it is a BCD counter, it counts from decimal 0 to 9 usually, if there is an illegal count (ie. 10, 11, 12, 13, 14, 15) happen, it returns to 0 immediately in the next count, eg. 11 -> 0.

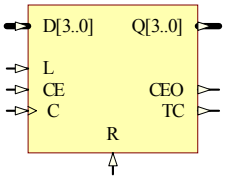
Inputs			Outputs					
R	**CE**	**C**	**Q3**	**Q2**	**Q1**	**Q0**	**TC**	**CEO**
1	x	↑	0	0	0	0	0	0
0	1	↑	Inc	Inc	Inc	Inc	TC	CEO
0	0	x	No Chg	No Chg	No Chg	No Chg	TC	0
0	1	x	1	0	0	1	1	1

TC = Q3•!Q2•!Q1•Q0

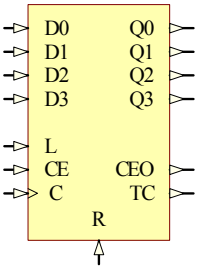
CEO = TC•CE

CD4RLE

Loadable Cascadable BCD Counter with Clock Enable and Synchronous Reset



CD4RLEB



CD4RLES

CD4RLE is a loadable cascadable binary-coded-decimal (BCD) counter with clock enable and synchronous reset

The synchronous reset (R) is the highest priority input. When R is High, all other inputs are ignored and all outputs go Low during the Low-to-High clock (C) transition.

The data on the D input is loaded into the counter when the load enable input (L) is High during the Low-to-High clock transition, independent of the state of clock enable (CE).

The Q outputs increment when clock enable (CE) is High during the Low-to-High clock transition. When CE is Low, clock transitions are ignored and outputs remain unchanged from the previous state.

The terminal count (TC) output is High when Q3 and Q0 are High and Q2 and Q1 are Low. The clock enable output (CEO) is High when TC and CE are both High.

Larger counters can be created by connecting the CEO output of the first stage to the CE input of the next stage and connecting C, L and CLR inputs in parallel. When cascading counters, use the CEO output if the counter uses the CE input; use the TC output if it does not.

As it is a BCD counter, it counts from decimal 0 to 9 usually, if there is an illegal count (ie. 10, 11, 12, 13, 14, 15) happen, it returns to 0 immediately in the next count, eg. 11 -> 0.

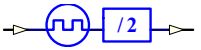
Inputs					Outputs					
R	**L**	**CE**	**D3 – D0**	**C**	**Q3**	**Q2**	**Q1**	**Q0**	**TC**	**CEO**
1	x	x	x	↑	0	0	0	0	0	0
0	1	x	D3 – D0	↑	D3	D	D	D0	TC	CEO
0	0	1	x	↑	Inc	Inc	Inc	Inc	TC	CEO
0	0	0	x	x	No Chg	No Chg	No Chg	No Chg	TC	0
0	0	1	x	x	1	0	0	1	1	1

TC = Q3•!Q2•!Q1•Q0

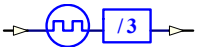
CEO = TC•CE

CDIV2 - 256

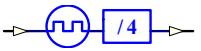
Clock Dividers



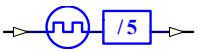
CDIV2



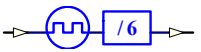
CDIV3



CDIV4



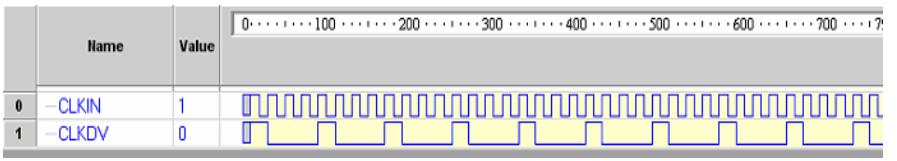
CDIV5

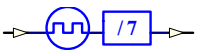


CDIV6

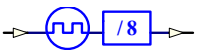
The CDIVn components are clock pulse dividers that can divide the clock cycle to produce the fixed value n pulse. The divide-by n values available are 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, 20, 24, 32, 64, 128 and 256. The duty cycle of the output (CLKDV) clock is 1/n.

The waveform below shows the CDIV4 component where the incoming clock (CLKIN) is divided by 4, therefore the outgoing clock is 4 clock cycles slower than the incoming clock with duty cycle of 25%.

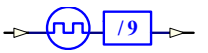




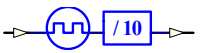
CDIV7



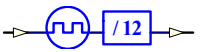
CDIV8



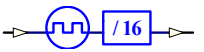
CDIV9



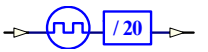
CDIV10



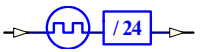
CDIV12



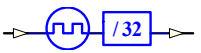
CDIV16



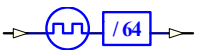
CDIV20



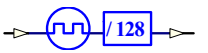
CDIV24



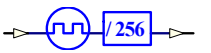
CDIV32



CDIV64



CDIV128



CDIV256

CDIV2DC50 – CDIV256DC50

Clock Dividers with 50% Duty Cycle Output

CDIV2DC50

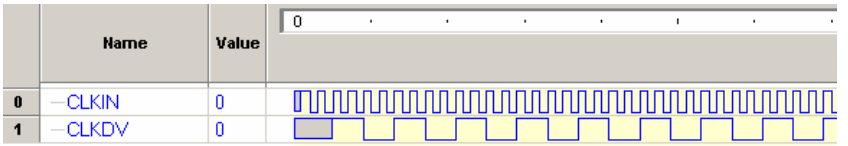
CDIV4DC50

CDIV6DC50

CDIV8DC50

The CDIVn50DC components are clock dividers that can produce clock division output of 50% duty cycle using only even division numbers. The divide-by n values available are 2, 4, 6, 8, 10, 12, 16, 20, 24, 32, 64, 128 and 256.

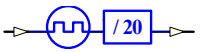
The waveform below shows the CDIV4 component where the output (CLKDV) is produced with duty cycle of 50%.



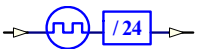
CDIV10DC50

CDIV12DC50

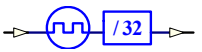
CDIV16DC50



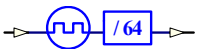
CDIV20DC50



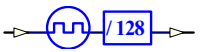
CDIV24DC50



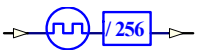
CDIV32DC50



CDIV64DC50



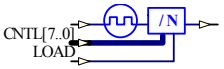
CDIV128DC50



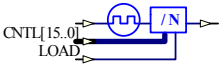
CDIV256DC50

CDIVN_8, CDIVN_16, CDIVN_32

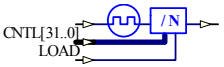
Programmable Clock Divider



CDIVN_8



CDIVN_16



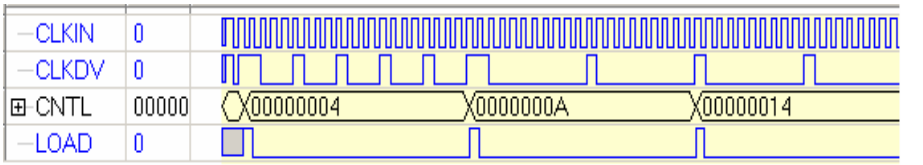
CDIVN_32

These are programmable clock dividers that can divide the incoming clock by user-programmed value present at the control input (CNTL). The bus length of the control input (CNTL) is available in 8-, 16- and 32-bit for CDIVN_8, CDIVN _16, and CDIVN _32 components respectively.

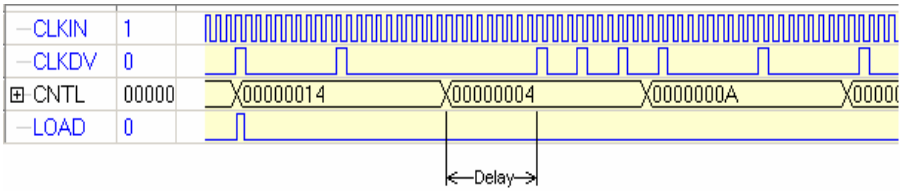
When devisor (CNTL) input is set to 0 the output (CLKDV) takes precedence over the internal counter output and becomes equal to the clock input (CLKIN). When Load input is High internal counter can be forced to load. When Load input is Low and a change in CNTL input occurs, a delay due to last value in the internal counter can be expected.

The clock output (CLKDV) duty cycle is 1/n where n is the devisor value from the control (CNTL) input.

The following waveform shows the expected behavior using the Load inputs:



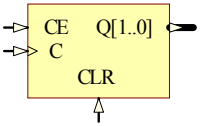
Expected output when Load is set to High and used to force the counter with new CNTL input.



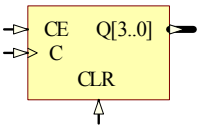
Expected output when Load is held Low and new CNTL input is loaded. A delay due to the internal counter can be expected between the next CLKDV output transitions.

CJ2CE, CJ4CE, CJ5CE, CJ8CE, CJ16CE, CJ32CE

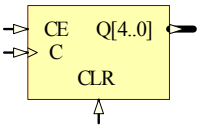
Johnson Counters with Clock Enable and Asynchronous Clear



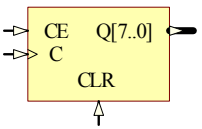
CJ2CEB



CJ4CEB



CJ5CEB



CJ8CEB

CJ2CE, CJ4CE, CJ5CE, CJ8CE, CJ16CE, and CJ32CE are, respectively 2-, 4-, 5-, 8-, 16-, and 32-Bit Johnson/ shift counters with clock enable and asynchronous clear.

The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored and all outputs go Low independent of the clock (C) transitions.

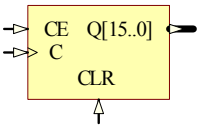
The counter increments when clock enable (CE) input is High during the Low-to-High clock transition. When CE is Low, clock transitions are ignored and outputs remain unchanged from the previous state.

When the Johnson/shift counter increment, the output data is shifted along one place, i.e. from Q0 to Q1, Q1 to Q2 and so forth.

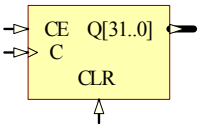
Inputs			Outputs				
CLR	**CE**	**C**	**Q0**	**Q1**	**…**	**Qz-1**	**Qz**
1	x	x	0	0	0	0	0
0	0	x	No Chg	No Chg	No Chg	No Chg	No Chg
0	1	↑	$\overline{qz}$	q0	…	qz-2	qz-1

q = state of referenced output one setup time prior to active clock transition

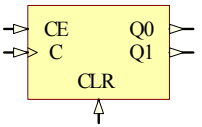
z = 1 for CJ2CE; z = 3 for CJ4CE; z = 4 for CJ5CE; z = 7 for CJ8CE; z = 15 for CJ16CE; z = 31 for CJ32CE



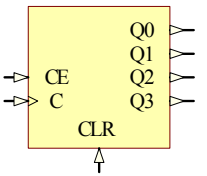
CJ16CEB



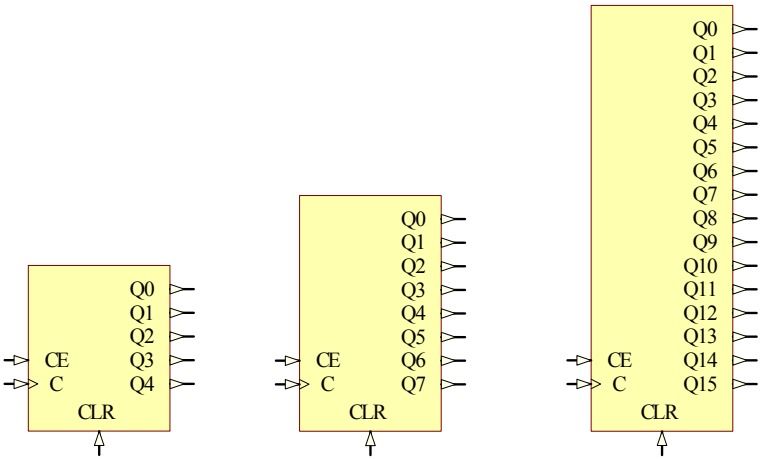
CJ32CEB



CJ2CES



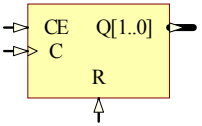
CJ4CES



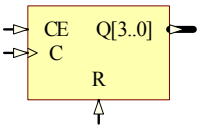
CJ5CES CJ8CES CJ16CES

CJ2RE, CJ4RE, CJ5RE, CJ8RE, CJ16RE, CJ32RE

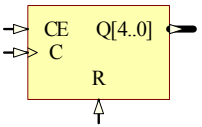
Johnson Counters with Clock Enable and Synchronous Reset



CJ2REB



CJ4REB



CJ5REB

CJ2RE, CJ4RE, CJ5RE, CJ8RE, CJ16RE, CJ32RE are, respectively 2-, 4-, 5-, 8-, 16-, 32-Bit Johnson/ shift counters with clock enable and synchronous reset.

The synchronous clear (R) is the highest priority input. When R is High, all other inputs are ignored and all outputs go Low during the Low-to-High clock (C) transitions.

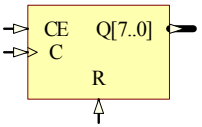
The counter increments when the clock enable input (CE) is High during the Low-to-High clock transition. When CE is Low, clock transitions are ignored and the outputs remain unchanged from the previous state.

When the Johnson/shift counter increment, the output data is shifted along one place, i.e. from Q0 to Q1, Q1 to Q2 and so forth.

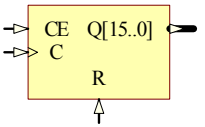
Inputs			Outputs				
R	**CE**	**C**	**Q0**	**Q1**	**…**	**Qz-1**	**Qz**
1	x	↑	0	0	0	0	0
0	0	x	No Chg	No Chg	No Chg	No Chg	No Chg
0	1	↑	$\overline{qz}$	q0	…	qz-2	qz-1

q = state of referenced output one setup time prior to active clock transition

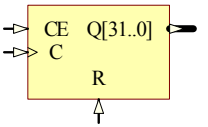
z = 1 for CJ2RE; z = 3 for CJ4RE; z = 4 for CJ5RE; z = 7 for CJ8RE; z = 15 for CJ16RE; z = 31 for CJ32RE



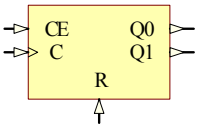
CJ8REB



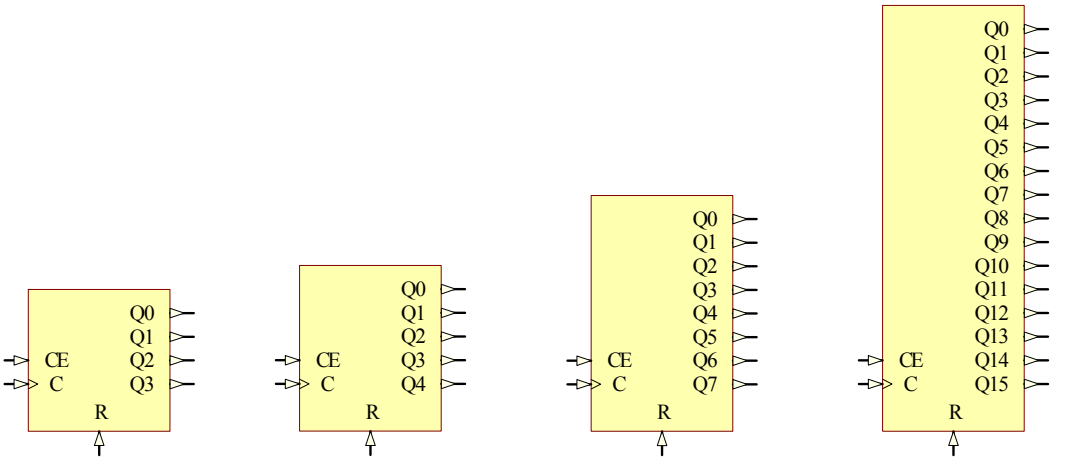
CJ16REB



CJ32REB



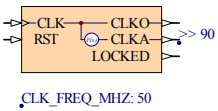
CJ2RES



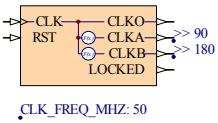
CJ4RES CJ5RES CJ8RES CJ16RES

CLKMAN_1, 2, 3, 4

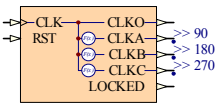
n Operational Output Digital Clock Manager



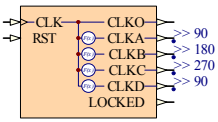
CLKMAN_1



CLKMAN_2



CLKMAN_3



CLKMAN_4

CLKMAN_1, CLKMAN_2, CLKMAN_3 and CLKMAN_4 are single, dual and multiple operational generic digital clock managers. These components provide a means to generate a wide variety of clocks depending on the users need at design time. CLKO is the exact same period as the input clock CLK however it is synchronized by the clock manager. The clock output (CLKA, CLKB, CLKC, and CLKD) of these components provides functions such as divide, multiply and phase shift of clock input CLKI. These outputs are also synchronized relative to the CLKO pin which serves as a reference clock for these outputs.

The CLKMAN_n components are automatically linked with the Altium core generator engine. Once an FPGA design containing this component is synthesized, the FPGA device clock manager or phase lock loop type primitives are automatically inferred in the design output before place and route occurs.

The number of CLKMAN_n components used per FPGA design is determined by the number of clock manager primitives allowed by the particular FPGA. Please refer to the FPGA device vendor's data sheet for the number of actual inferred primitive (see table below) supported.

The following table lists the supported FPGA devices and its primitive inferred.

FPGA Vendor	Device	Inferred Primitive
Xilinx	Spartan-II	CLKDLL
	Virtex	CLKDLL
	Spartan-IIE	CLKDLLE
	Virtex-E	CLKDLLE
	Virtex-II	DCM
	Virtex-II Pro	DCM
	Spartan3	DCM
	Virtex-4	DCM
Altera	Cyclone	ALTPLL
	Stratix	ALTPLL
	StratixII	ALTPLL
	StratixGX	ALTPLL
Actel	ProAsic Plus	PLLCORE

The desired clock output operation derived from the clock input is achieved by using the following configurable parameters found on the components properties:

CLK_FREQ_MHZ – This specifies the clock input (CLK) frequency. The default value is set to 50 MHz. The frequency range is dependent on the FPGA device and its speed grade. This parameter is essential for Altera but not used for Xilinx clock managers.

CLK*n*_OPERATION – Where *n* represents A, B, C and D output ports. The parameter defines the desired operation of CLK*n* output. The default operation is set to phase shift with angles set to 90°, 180° and 270°.

Phase Shifting Operation

Phase shift operation is performed by setting the relevant operational output port's CLKn_Operation parameter value to **>> <phase_shift>**. Where phase_shift is the actual value in degrees this clock needs to be phase shifted compared to the reference clock CLKO. The set of allowable values here depend on the particular device.

Divide operation

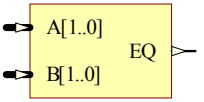
Divide operation is performed by setting the relevant operational output port's CLKn_Operation parameter value to **/<divison_number>**. Where division_number is the value CLKn is divided by compared to the reference clock CLKO. The set of values permitted depends on the particular device.

Multiply Operation

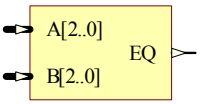
Multiply operation is performed by setting the relevant operational output ports CLKn_Operation parameter value to **x<multiply_number>**. Where multiply_number is the number of times this clock is multiplied by when compared to the reference clock.

COMP2 – COMP32

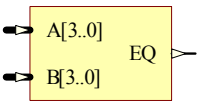
Identity Comparator



COMP2B



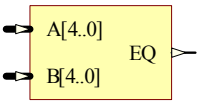
COMP3B



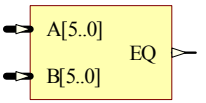
COMP4B

COMP2, COMP3, COMP4, COMP5, COMP6, COMP7, COMP8, COMP9, COMP10, COMP12, COMP16 and COMP32 are, respectively, 2-, 3-, 4-, 5-, 6-, 7-, 8-, 9-, 10-, 12-, 16- and 32-bit identity comparators. The equal output (EQ) of the identity comparator is High when the two words An – A0 and Bn – B0 are equal. When any two of the corresponding bits from each word are not the same, the EQ output is Low.

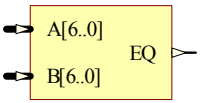
Inputs				Outputs
An, Bn	**…**	**A1, B1**	**A0, B0**	**EQ**
An ≠ Bn	…	A1 ≠ B1	A0 ≠ B0	0
An ≠ Bn	…	A1 ≠ B1	A0 = B0	0
An ≠ Bn	…	A1 = B1	A0 = B0	0
An = Bn	…	A1 = B1	A0 = B0	1



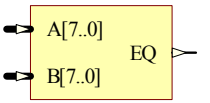
COMP5B



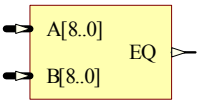
COMP6B



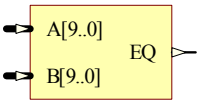
COMP7B



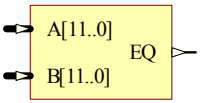
COMP8B



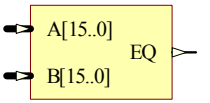
COMP9B



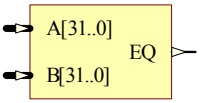
COMP10B



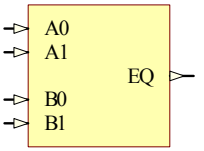
COMP12B



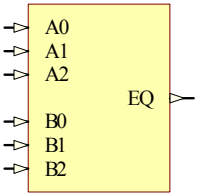
COMP16B



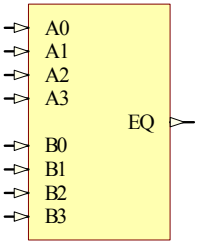
COMP32B



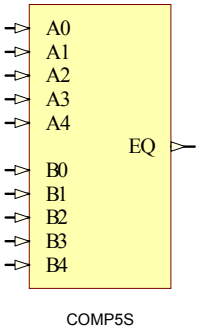
COMP2S



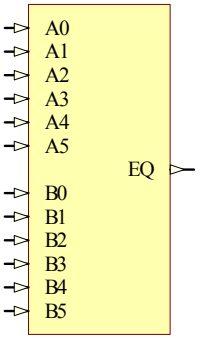
COMP3S



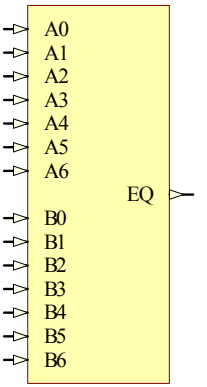
COMP4S



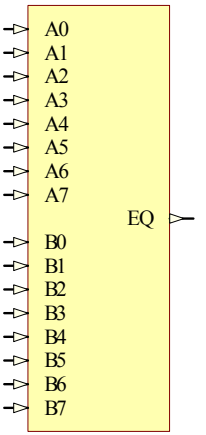
COMP5S



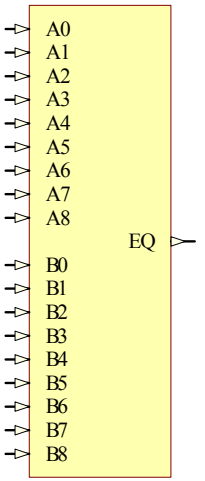
COMP6S



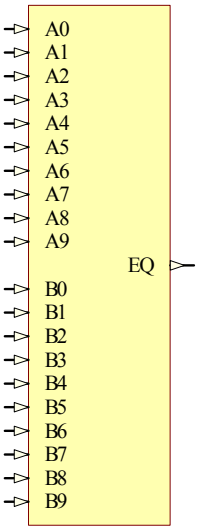
COMP7S



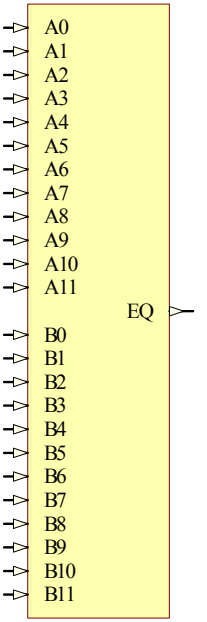
COMP8S



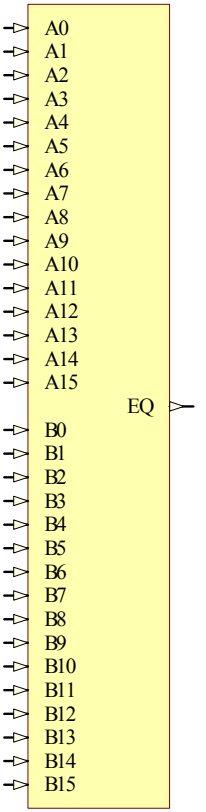
COMP9S



COMP10S



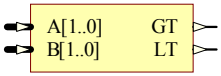
COMP12S



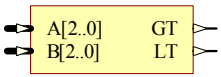
COMP16S

COMPM2 – COMPM32

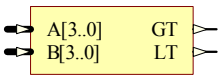
Magnitude Comparator



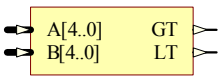
COMPM2B



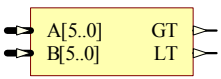
COMPM3B



COMPM4B



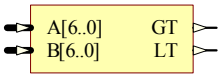
COMPM5B



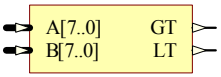
COMPM6B

COMPM2, COMPM3, COMPM4, COMPM5, COMPM6, COMPM7, COMPM8, COMPM9, COMPM10, COMPM12, COMPM16, and COMPM32 are, respectively, 2-, 3-, 4-, 5-, 6-, 7-, 8-, 9-, 10-, 12-, 16- and 32-bit magnitude comparators that compare two positive binary-weighted words. The greater-than output (GT) is High when A>B, and the less-than output (LT) is High when A<B. When the two words are equal, both GT and LT are Low. Equality can be measured with this macro by comparing both outputs with a NOR gate.

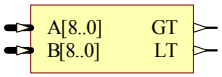
Inputs				Outputs	
A*n*, B*n*	**...**	**A1, B1**	**A0, B0**	**GT**	**LT**
A*n*>B*n*	...	X	X	1	0
A*n*=B*n*	...	A1>B1	X		
A*n*=B*n*	...	A1=B1	A0>B0		
A*n*<B*n*	...	X	X	0	1
A*n*=B*n*	...	A1<B1	X		
A*n*=B*n*	...	A1=B1	A0<B0		
A*n*=B*n*	...	A1=B1	A0=B0	0	0



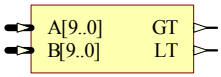
COMPM7B



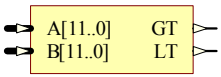
COMPM8B



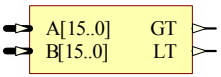
COMPM9B



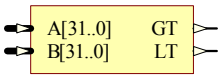
COMPM10B



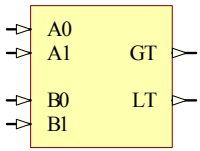
COMPM12B



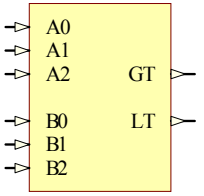
COMPM16B



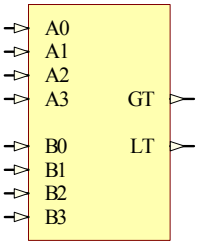
COMPM32B



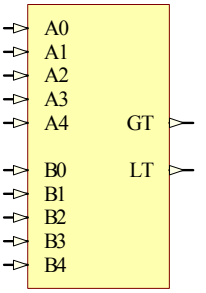
COMPM2S



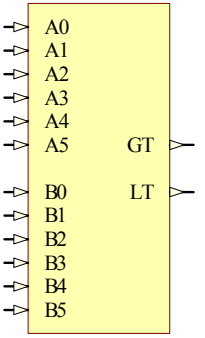
COMPM3S



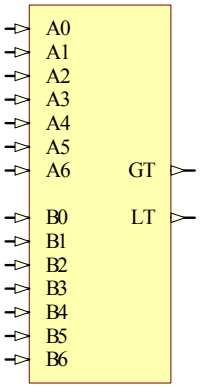
COMPM4S



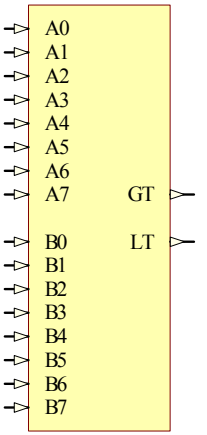
COMPM5S



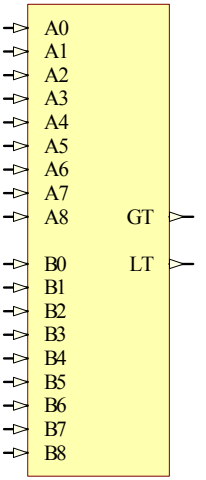
COMPM6S



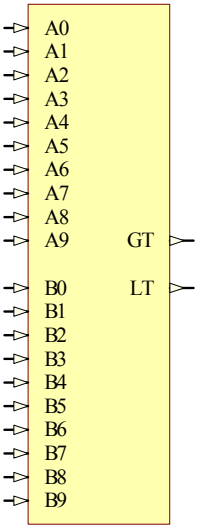
COMPM7S



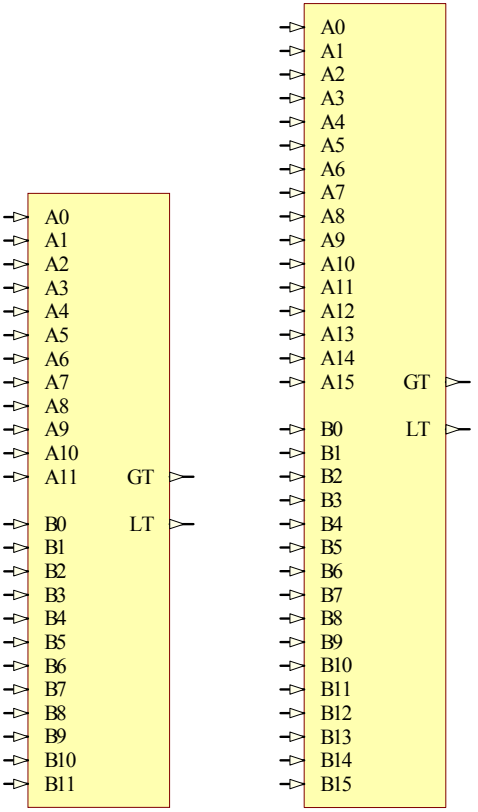
COMPM8S



COMPM9S

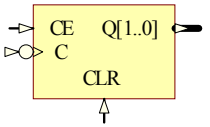


COMPM10S

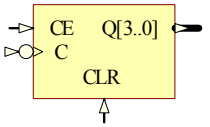

A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
GT
B0
LT
B1
B2
B3
B4
B5
B6
B7
B8
B9
B10
B11
COMPM12S
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
GT
B0
LT
B1
B2
B3
B4
B5
B6
B7
B8
B9
B10
B11
B12
B13
B14
B15
COMPM16S

CR2CE, CR4CE, CR8CE, CR16CE, CR32CE

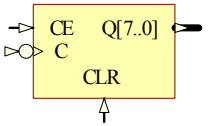
Negative-Edge Binary Ripple Counters with Clock Enable and Asynchronous Clear



CR2CEB



CR4CEB



CR8CEB

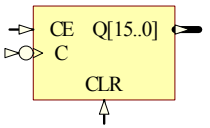
CR2CE, CR4CE, CR8CE, CR16CE, CR32CE are respectively 2-, 4-, 8-, 16-, 32-bit negative-edge binary ripple counters with clock enable and asynchronous clear.

The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored and all outputs go to Low independent of the clock (C) transitions.

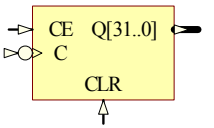
The counter increments when clock enable input (CE) is High during the High-to-Low clock transition. When CE is Low, clock transitions are ignored and the outputs remain in the same state as the previous clock cycle.

Inputs			Outputs
CLR	**CE**	**C**	**Qz – Q0**
1	x	x	0
0	0	x	No Chg
0	1	↓	Inc

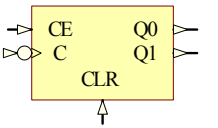
z=1 for CR2CE; Z=3 for CR4CE; z = 7 for CR8CE; z = 15 for CR16CE; z = 31 for CR32CE.



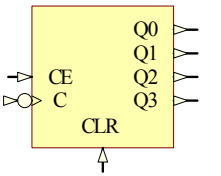
CR16CEB



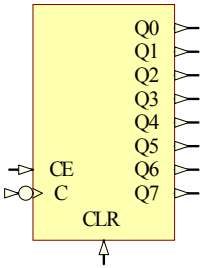
CR32CEB



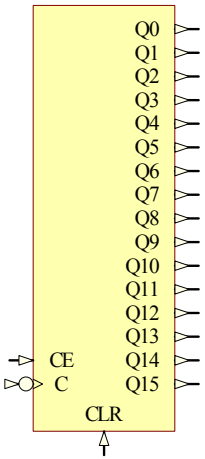
CR2CES



CR4CES



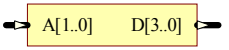
CR8CES



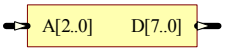
CR16CES

D2_4, D3_8, D4_16, D5_32

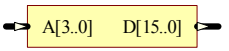
m- to *n*-Line Decoder



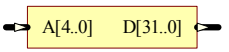
D2_4B



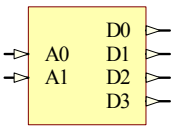
D3_8B



D4_16B



D5_32B



D2_4S

D2_4, D3_8, D4_16 and D5_32 are respectively 2- to 4-Line, 3- to 8-Line, 4- to 16-Line, and 5- to 32-Line Decoders that select one active-High output (Dn – D0) based on the value of the binary address (An – A0) input. The non-selected outputs are Low.

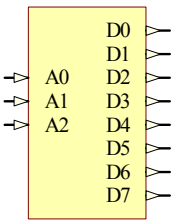
Decimal value of Input A	Inputs				Outputs					
	A*m*	…	A1	A0	D*n*	…	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	0	0	1	0
2	0	0	1	0	0	0	0	1	0	0
3	0	0	1	1	0	0	1	0	0	0
…	…	…	…	…	…	…	…	…	…	…
n	1	1	1	1	1	0	0	0	0	0

m = 1, n = 3 for D2_4

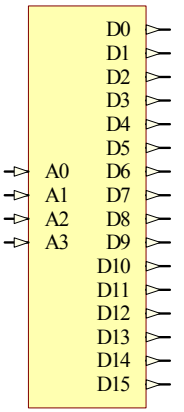
m = 2, n = 7 for D3_8

m = 3, n = 15 for D4_16

m = 4, n = 31 for D5_32



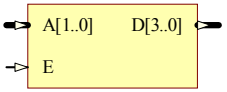
D3_8S



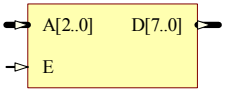
D4_16S

D2_4E, D3_8E, D4_16E, D5_32E

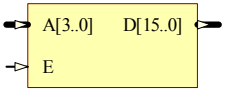
m- to *n*-Line Decoder with Enable



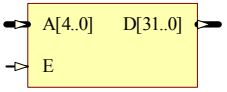
D2_4EB



D3_8EB



D4_16EB



D5_32EB

D2_4E, D3_8E, D4_16E and D5_32E are respectively 2-to 4-Line, 3-to 8-Line, 4-to 16-Line and 5-to 32-Line Decoders. When the enable (E) input is High, one of the active-High outputs (Dn – D0) is selected based on the value of the binary address (An – A0) input. The non-selected outputs are Low. Also, when the E input is Low, all outputs are Low.

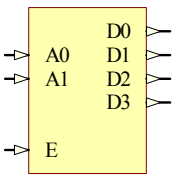
Decimal value of Input A	Inputs					Outputs					
	E	A*m*	…	A1	A0	D*n*	…	D3	D2	D1	D0
x	0	x	x	x	x	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	1
1	1	0	0	0	1	0	0	0	0	1	0
2	1	0	0	1	0	0	0	0	1	0	0
3	1	0	0	1	1	0	0	1	0	0	0
…	1	…	…	…	…	…	…	…	…	…	…
n	1	1	1	1	1	1	0	0	0	0	0

m = 1, n = 3 for D2_4E

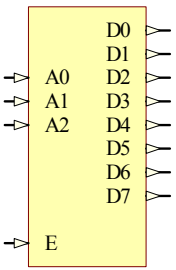
m = 2, n = 7 for D3_8E

m = 3, n = 15 for D4_16E

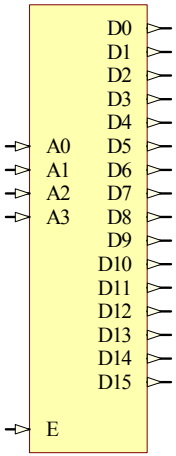
m = 4, n = 31 for D5_32E



D2_4ES



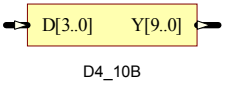
D3_8ES



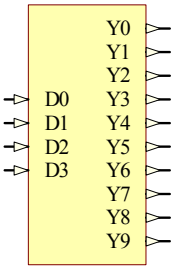
D4_16ES

D4_10

BCD-to-Decimal Decoder/Driver



D4_10B



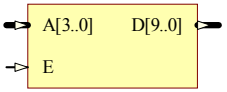
D4_10S

D4_10 is an inverted one-hot decoder. It decodes valid BCD input logic ensuring that all outputs remain off for all invalid binary input conditions. The BCD logic is connected to inputs D[3..0], with the resulting inverted one-hot decoded logic appearing on outputs Y[9..0].

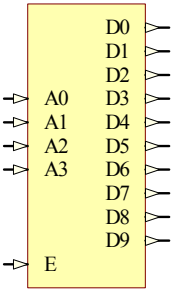
Inputs				Outputs									
D3	**D2**	**D1**	**D0**	**Y9**	**Y8**	**Y7**	**Y6**	**Y5**	**Y4**	**Y3**	**Y2**	**Y1**	**Y0**
0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	1	0	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	0	1	0	0	0	0	0	0
0	1	1	1	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	0
1	1	x	x	0	0	0	0	0	0	0	0	0	0

D4_10E

BCD-to-Decimal Decoder/Driver with Enable



D4_10EB



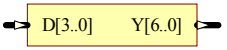
D4_10ES

D4_10E is an inverted one-hot decoder. When the Enable input is High, the device decodes valid BCD input logic ensuring that all outputs remain off for all invalid binary input conditions. The BCD logic is connected to inputs A[3..0], with the resulting inverted one-hot decoded logic appearing on outputs D[9..0].

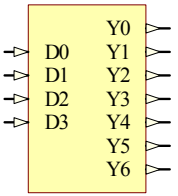
Inputs					Outputs									
E	**D3**	**D2**	**D1**	**D0**	**Y9**	**Y8**	**Y7**	**Y6**	**Y5**	**Y4**	**Y3**	**Y2**	**Y1**	**Y0**
0	x	x	x	x	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	1	0	0
1	0	0	1	1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	0	0	1	0	0	0	0
1	0	1	0	1	0	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	0	1	0	0	0	0	0	0
1	0	1	1	1	0	0	1	0	0	0	0	0	0	0
1	1	0	0	0	0	1	0	0	0	0	0	0	0	0
1	1	0	0	1	1	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	1	0	0	0	0	0	0	0	0	0	0
1	1	1	x	x	0	0	0	0	0	0	0	0	0	0

D7SEG

7-Segment Display Decoder for Common-Cathode LED



D7SEGB



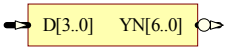
D7SEGS

D7SEG decodes a 4-bit binary-coded-decimal (BCD) input for 7-Segment Common-Cathode LED Display. Outputs of D7SEG are Active High.

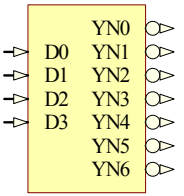
Inputs				Outputs						
D3	**D2**	**D1**	**D0**	**YN6**	**YN5**	**YN4**	**YN3**	**YN2**	**YN1**	**YN0**
0	0	0	0	0	1	1	1	1	1	1
0	0	0	1	0	0	0	0	1	1	0
0	0	1	0	1	0	1	1	0	1	1
0	0	1	1	1	0	0	1	1	1	1
0	1	0	0	1	1	0	0	1	1	0
0	1	0	1	1	1	0	1	1	0	1
0	1	1	0	1	1	1	1	1	0	0
0	1	1	1	0	0	0	0	1	1	1
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	0	0	1	1	1
1	0	1	0	1	1	1	0	1	1	1
1	0	1	1	1	1	1	1	1	0	0
1	1	0	0	0	1	1	1	0	0	1
1	1	0	1	1	0	1	1	1	1	0
1	1	1	0	1	1	1	1	0	0	1
1	1	1	0	1	1	1	0	0	0	1

D7SEGN

7-Segment Display Decoder for Common-Anode LED



D7SEGNB



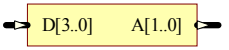
D7SEGNS

D7SEGN decodes a 4-bit binary-coded-decimal (BCD) input for 7-Segment Common-Anode LED Display. Outputs of D7SEGN are Active Low.

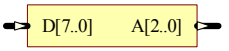
Inputs				Outputs						
D3	**D2**	**D1**	**D0**	**Y6**	**Y5**	**Y4**	**Y3**	**Y2**	**Y1**	**Y0**
0	0	0	0	1	0	0	0	0	0	0
0	0	0	1	1	1	1	1	0	0	1
0	0	1	0	0	1	0	0	1	0	0
0	0	1	1	0	1	1	0	0	0	0
0	1	0	0	0	0	1	1	0	0	1
0	1	0	1	0	0	1	0	0	1	0
0	1	1	0	0	0	0	0	0	1	1
0	1	1	1	1	1	1	1	0	0	0
1	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	1	1	0	0	0
1	0	1	0	0	0	0	1	0	0	0
1	0	1	1	0	0	0	0	0	1	1
1	1	0	0	1	0	0	0	1	1	0
1	1	0	1	0	1	0	0	0	0	1
1	1	1	0	0	0	0	0	1	1	0
1	1	1	0	0	0	0	1	1	1	0

E4_2, E8_3, E16_4, E32_5

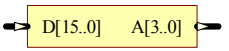
m-Line to *n*-Line Priority Encoder



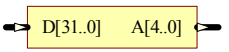
E4_2B



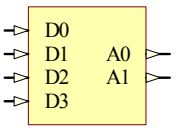
E8_3B



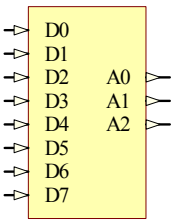
E16_4B



E32_5B



E4_2S



E8_3S

E4_2, E8_3, E16_4 and E32_5 are respectively 4-to 2-Line, 8-to 3-Line, 16-to 4-Line and 32-to 5-Line Priority Encoders. It accepts data from D0 - D*m* inputs and provides binary representation on the outputs A0 - A*n*. A priority is assigned to each input so that when two or more inputs are simultaneously active, the input with the highest priority is represented on the output. Input lines D3, D7, D15, D31 are the highest priority in each of the types of the encoders.

When all data inputs are Low or the lowest priority line (D0) is High and all other inputs are Low, all outputs (A0, A1, A2, An) are forced to Low state.

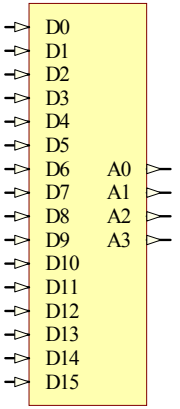
Inputs						Outputs				Decimal value of Output A
D*n*	…	D3	D2	D1	D0	A*m*	…	A1	A0	
0	0	0	0	0	1/0	0	0	0	0	0
0	0	0	0	1	x	0	0	0	1	1
0	0	0	1	x	x	0	0	1	0	2
0	0	1	x	x	x	0	0	1	1	3
…	…	…	…	…	…	…	…	…	…	…
1	x	x	x	x	x	1	1	1	1	*n*

n = 3, m = 1 for E4_2E

n = 7, m = 2 for E8_3E

n = 15, m = 3 for E16_4E

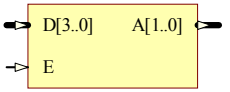
n = 31, m = 4 for E32_5E



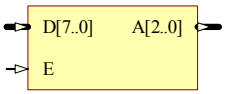
E16_4S

E4_2E, E8_3E, E16_4E, E32_5E

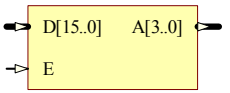
m-Line to *n*-Line Priority Encoder with Enable



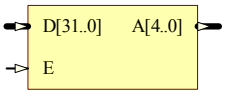
E4_2EB



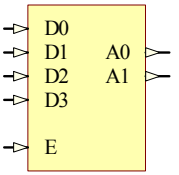
E8_3EB



E16_4EB



E32_5EB



E4_2ES

E4_2, E8_3, E16_4 and E32_5E are respectively 4-to 2-Line, 8-to 3-Line, 16-to 4-Line and 32-to 5-Line Priority Encoders with Enable. It accepts data from D0 - D*m* inputs and provides binary representation on the outputs A0 - A*n* when enable (E) is High. A priority is assigned to each input so that when two or more inputs are simultaneously active, the input with the highest priority is represented on the output. Input lines D3, D7, D15, D31 are the highest priority in each of the types of the encoders.

When all data inputs are Low or the lowest priority line (D0) is High and all other inputs are Low, and enable (E) is High all outputs (A0, A1, A2, An) are forced to Low state. When enable (E) is Low all data inputs as overridden and outputs (A0, A1, A2, An) are forced to Low state.

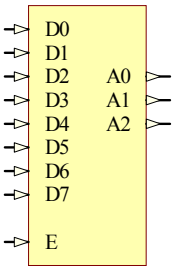
Inputs							Outputs				Decimal value of Output A
E	**D*n***	**…**	**D3**	**D2**	**D1**	**D0**	**A*m***	**…**	**A1**	**A0**	
0	x	x	x	x	x	x	0	0	0	0	-
1	0	0	0	0	0	1/0	0	0	0	0	0
1	0	0	0	0	1	x	0	0	0	1	1
1	0	0	0	1	x	x	0	0	1	0	2
1	0	0	1	x	x	x	0	0	1	1	3
1	…	…	…	…	…	…	…	…	…	…	…
1	1	x	x	x	x	x	1	1	1	1	*n*

n = 3, m = 1 for E4_2

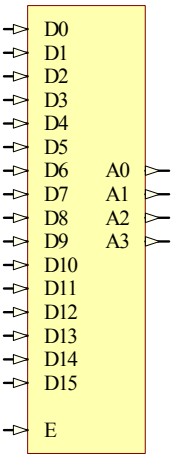
n = 7, m = 2 for E8_3

n = 15, m = 3 for E16_4

n = 31, m = 4 for E32_5



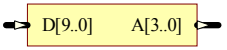
E8_3ES



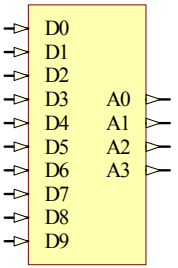
E16_4ES

E10_4

Decimal-to-BCD Decoder/Driver



E10_4B



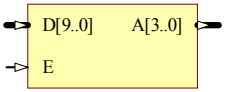
E10_4S

This device decodes data from Decimal logic to BCD logic ensuring that all outputs remain high for all invalid binary input conditions. The binary logic is connected to inputs D[9..0], with the resulting BCD decoded logic appearing on outputs A[3..0].

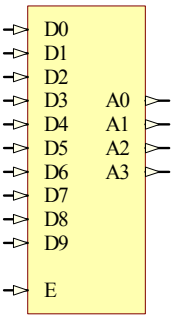
D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	A3	A2	A1	A0
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	0	1	0	1
0	0	0	1	0	0	0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	1	0	0	1
All other combinations										1	1	1	1

E10_4E

Decimal-to-BCD Decoder/Driver with Enable



E10_4EB



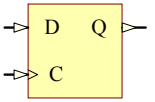
E10_4ES

When the Enable input is High, this device decodes data from Decimal logic to BCD logic ensuring that all outputs remain high for all invalid binary input conditions. The binary logic is connected to inputs D[9..0], with the resulting inverted BCD decoded logic appearing on outputs A[3..0].

Inputs											Outputs			
E	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**	**A3**	**A2**	**A1**	**A0**
0	x	x	x	x	x	x	x	x	x	x	1	1	1	1
1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	0	0	1	0	0	0	0	1
1	0	0	0	0	0	0	0	1	0	0	0	0	1	0
1	0	0	0	0	0	0	1	0	0	0	0	0	1	1
1	0	0	0	0	0	1	0	0	0	0	0	1	0	0
1	0	0	0	0	1	0	0	0	0	0	0	1	0	1
1	0	0	0	1	0	0	0	0	0	0	0	1	1	0
1	0	0	1	0	0	0	0	0	0	0	0	1	1	1
1	0	1	0	0	0	0	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	0	0	0	0	1	0	0	1
1	All other combinations										1	1	1	1

FD, FD2, FD4, FD8, FD16, FD32

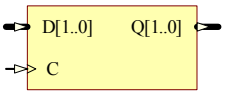
D-Type Flip-Flop



FD

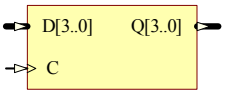
FD, FD2, FD4, FD8, FD16, FD32 are, respectively 1-, 2-, 4-, 8-, 16-, 32-Bit D-type positive edge trigger flip-flop.

Input data (D) is loaded into the output (Q) during Low-to-High clock (C) transition.

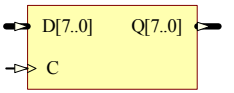


FD2B

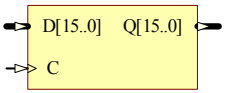
Inputs		Output
C	D	Q
↑	d	d



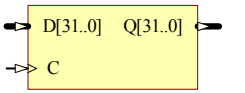
FD4B



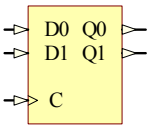
FD8B



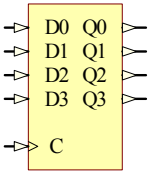
FD16B



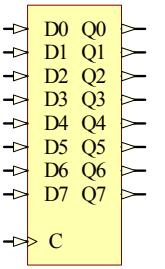
FD32B



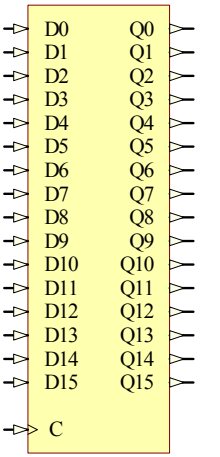
FD2S



FD4S



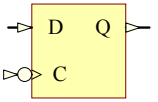
FD8S



FD16S

FD_1

D-Type Negative Edge Flip-Flop



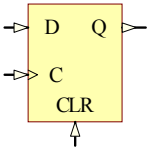
FD_1

FD_1 is a D-type negative edge trigger flip-flop. Input data (D) is loaded into the output (Q) during High-to-Low clock (C) transition.

Inputs		Output
C	D	Q
↓	d	d

FDC, FD2C, FD4C, FD8C, FD16C, FD32C

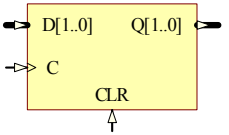
D-Type Flip-Flop with Asynchronous Clear



FDC

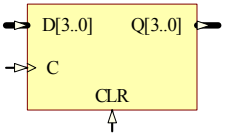
FDC, FD2C, FD4C, FD8C, FD16C, FD32C are, respectively 1-, 2-, 4-, 8-, 16-, 32-Bit D type positive edge flip-flops with asynchronous clear (CLR).

When clear (CLR) is High, all other inputs are ignored and output (Q) is set to Low. Input data (D) is loaded into the output (Q) when clear (CLR) is Low on the Low-to-High clock (C) transition.

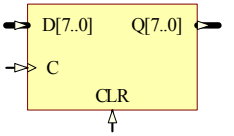


FD2CB

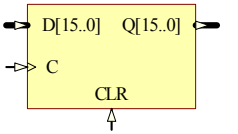
Inputs			Outputs
CLR	C	D	Q
1	x	x	0
0	↑	d	d



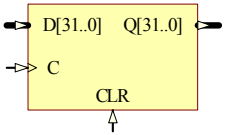
FD4CB



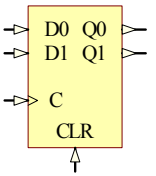
FD8CB



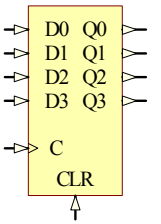
FD16CB



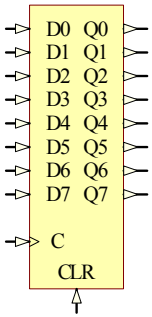
FD32CB



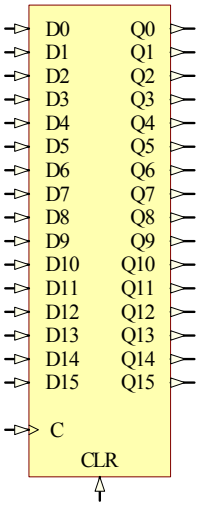
FD2CS



FD4CS



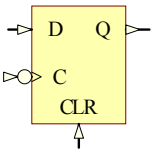
FD8CS



FD16CS

FDC_1

D-Type Negative Edge Flip-Flop with Asynchronous Clear



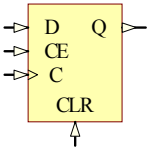
FDC_1

FDC_1 is a D type negative edge flip-flop with asynchronous clear (CLR). When clear (CLR) is High, all other inputs are ignored and output (Q) is set to Low. Input data (D) is loaded into the output (Q) when clear (CLR) is Low on the High-to-Low clock (C) transition.

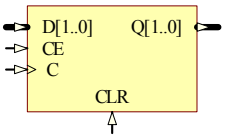
Inputs			Output
CLR	**C**	**D**	**Q**
1	x	x	0
0	↓	d	d

FDCE, FD2CE, FD4CE, FD8CE, FD16CE, FD32CE

D-Type Flip-Flop with Clock Enable and Asynchronous Clear



FDCE

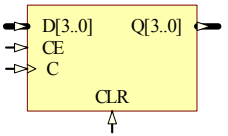


FD2CEB

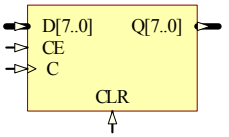
FDCE, FD2CE, FD4CE, FD8CE, FD16CE and FD32CE are, respectively 1-, 2-, 4-, 8-, 16-, 32-Bit D type positive edge flip-flops with clock enable (CE) and asynchronous clear (CLR).

When clear (CLR) is High, all other inputs are ignored and output (Q) is set to Low. When clear (CLR) is Low and clock enable (CE) is High, input data (D) is loaded into the output (Q) on the Low-to-High clock (C) transition. When clock enable (CE) is Low and clear (CLR) is Low, clock transitions are ignored and output (Q) does not change state.

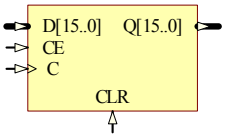
Inputs				Output
CLR	**CE**	**C**	**D**	**Q**
1	x	x	x	0
0	0	x	x	Q_0
0	1	↑	d	d



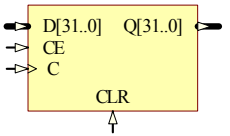
FD4CEB



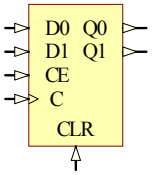
FD8CEB



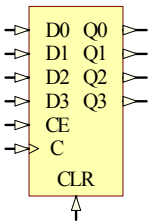
FD16CEB



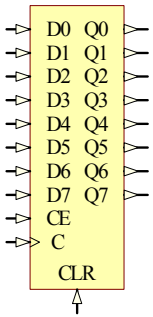
FD32CEB



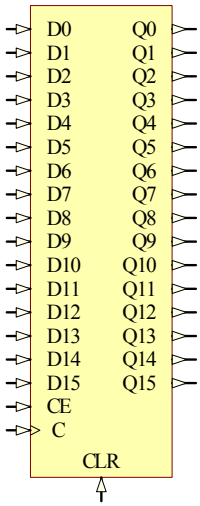
FD2CES



FD4CES



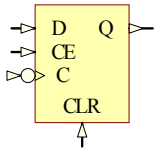
FD8CES



FD16CES

FDCE_1

D-Type Negative Edge Flip-Flop with Clock Enable and Asynchronous Clear



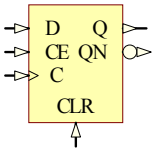
FDCE_1

FDCE_1 is a D type negative edge flip-flop with clock enable (CE) and asynchronous clear (CLR). When clear (CLR) is High, all other inputs are ignored and output (Q) is set to Low. When clear (CLR) is Low and clock enable (CE) is High, input data (D) is loaded into the output (Q) on the High-to-Low clock (C) transition. When clock enable (CE) is Low and clear (CLR) is Low, clock transitions are ignored and output (Q) does not change state.

Inputs				Output
CLR	**CE**	**C**	**D**	**Q**
1	x	x	x	0
0	0	x	x	Q_0
0	1	↓	d	d

FDCEN

D-Type Flip-Flop with Clock Enable and Asynchronous Clear and Inverted and Non-Inverted Outputs



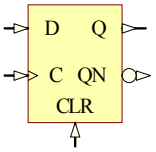
FDCEN

FDCEN is a D type positive edge flip-flop with clock enable (CE) and asynchronous clear (CLR) and inverted (QN) and non-inverted (Q) outputs. When clear (CLR) is High, all other inputs are ignored and inverted (QN) and non-inverted (Q) outputs are set to High and Low respectively. When clear (CLR) is Low and clock enable (CE) is High, input data (D) is loaded into the outputs Q and QN on the Low-to-High clock (C) transition. When clock enable (CE) is Low and clear (CLR) is Low, clock transitions are ignored and outputs do not change state.

Inputs				Outputs	
CLR	**CE**	**C**	**D**	**Q**	**QN**
1	x	x	x	0	1
0	0	x	x	Q_0	QN_0
0	1	↑	d	d	$\overline{d}$

FDCN

D-Type Flip-Flop with Asynchronous Clear and Inverted and Non-Inverted Outputs



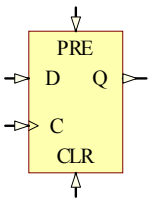
FDCN

FDCN is a D type positive edge flip-flop with asynchronous clear (CLR) and inverted (QN) and non-inverted (Q) outputs. When clear (CLR) is High, all other inputs are ignored and inverted (QN) and non-inverted (Q) outputs are set to High and Low respectively. Input data (D) is loaded into the outputs when clear (CLR) is Low on the Low-to-High clock (C) transition.

Inputs			Outputs	
CLR	C	D	Q	QN
1	x	x	0	1
0	↑	d	d	$\overline{d}$

FDCP, FD2CP, FD4CP, FD8CP, FD16CP, FD32CP

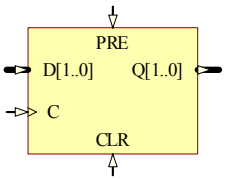
D-Type Flip-Flop with Asynchronous Preset and Clear



FDCP

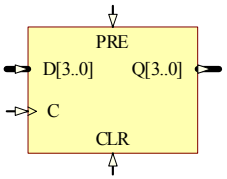
FDCP, FD2CP, FD4CP, FD8CP, FD16CP, FD32CP are, respectively 1-, 2-, 4-, 8-, 16-, 32-Bit D type positive edge flip-flops with asynchronous preset (PRE) and clear (CLR).

When clear (CLR) is High, all inputs are cleared and output (Q) is set to Low. When clear (CLR) is Low and preset (PRE) is High, clock (C) transition and input data (D) is ignored and output (Q) is set to High. When clear (CLR) and preset (PRE) is Low, input data (D) is transferred to output (Q) on the Low-to-High clock (C) transition.

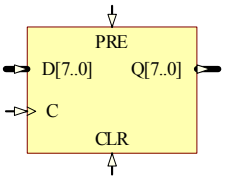


FD2CPB

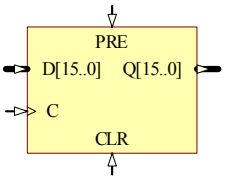
Inputs				Output
CLR	PRE	C	D	Q
1	x	x	x	0
0	1	x	x	1
0	0	↑	d	d



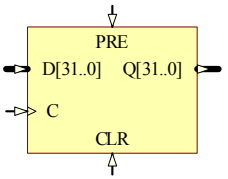
FD4CPB



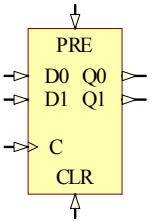
FD8CPB



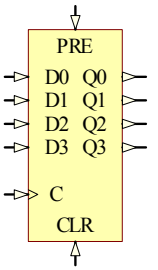
FD16CPB



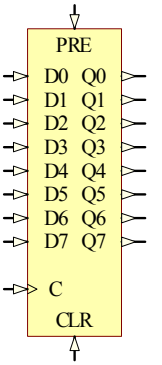
FD32CPB



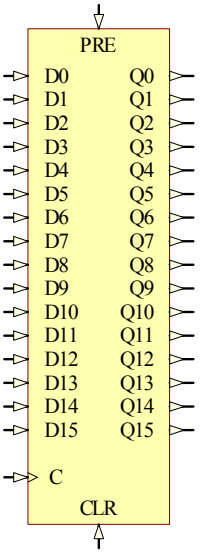
FD2CPS



FD4CPS



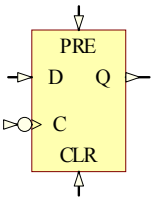
FD8CPS



FD16CPS

FDCP_1

D-Type Negative Edge Flip-Flop with Asynchronous Preset and Clear



FDCP_1

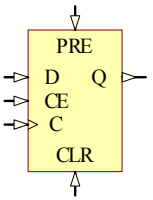
FDCP_1 is a D type negative edge flip-flop with asynchronous preset (PRE) and clear (CLR).

When clear (CLR) is High, all inputs other inputs are ignored and output (Q) is set to Low. When clear (CLR) is Low and preset (PRE) is High, clock (C) transition and input data (D) is ignored and output (Q) is set to High. When clear (CLR) and preset (PRE) are Low, input data (D) is transferred to output (Q) on the High-to-Low clock (C) transition.

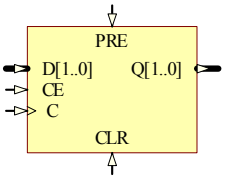
Inputs				Output
CLR	**PRE**	**C**	**D**	**Q**
1	x	x	x	0
0	1	x	x	1
0	0	↓	d	d

FDCPE, FD2CPE, FD4CPE, FD8CPE, FD16CPE, FD32CPE

D-Type Flip-Flop with Clock Enable and Asynchronous Preset and Clear



FDCPE

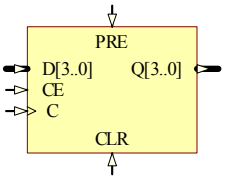


FD2CPEB

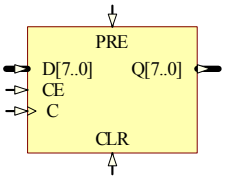
FDCPE, FD2CPE, FD4CPE, FD8CPE, FD16CPE, FD32CPE are, respectively 1-, 2-, 4-, 8-, 16-, 32-Bit D type positive edge flip-flops with clock enable (CE) and asynchronous preset (PRE) and clear (CLR).

When clear (CLR) is High, all other inputs are ignored and output (Q) is set to Low. When clear (CLR) is Low and preset (PRE) is High, all other inputs are ignored and output (Q) is set to High. When clear (CLR) and preset (PRE) are Low and clock enable is High, input data (D) is transferred to output (Q) on the Low-to-High clock (C) transition. When clear (CLR), preset (PRE) and clock enable (CE) are Low, clock (C) transition and input data (D) is ignored and output (Q) does not change state.

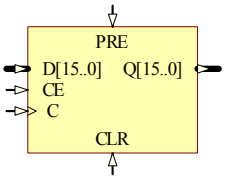
Inputs					Output
CLR	PRE	CE	C	D	Q
1	x	x	x	x	0
0	1	x	x	x	1
0	0	0	x	x	Q_0
0	0	1	↑	d	d



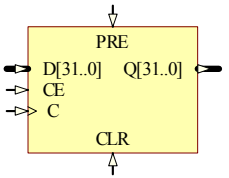
FD4CPEB



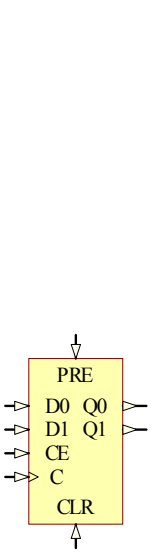
FD8CPEB



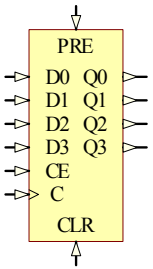
FD16CPEB



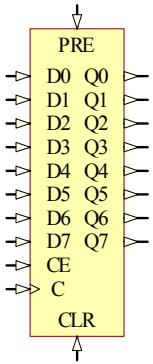
FD32CPEB



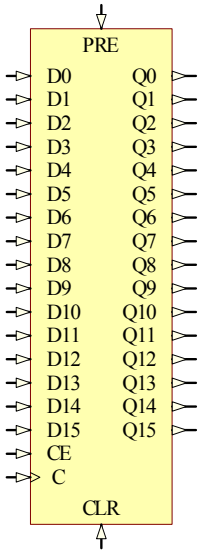
FD2CPES



FD4CPES



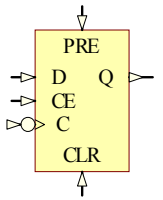
FD8CPES



FD16CPES

FDCPE_1

D-Type Negative Edge Flip-Flop with Clock Enable and Asynchronous Preset and Clear



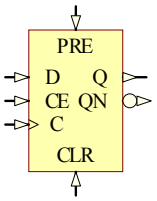
FDCPE_1

FDCPE_1 is a D type negative edge flip-flop with clock enable (CE) and asynchronous preset (PRE) and clear (CLR). When clear (CLR) is High, all other inputs are ignored and output (Q) is set to Low. When clear (CLR) is Low and preset (PRE) is High, all other inputs are ignored and output (Q) is set to High. When clear (CLR) and preset (PRE) are Low and clock enable is High, input data (D) is transferred to output (Q) on the High-to-Low clock (C) transition. When clear (CLR), preset (PRE) and clock enable (CE) are Low, clock (C) transition and input data (D) is ignored and output (Q) does not change state.

Inputs					Output
CLR	**PRE**	**CE**	**C**	**D**	**Q**
1	x	x	x	x	0
0	1	x	x	x	1
0	0	0	x	x	Q_0
0	0	1	↓	d	d

FDCPEN

D-Type Flip-Flop with Clock Enable and Asynchronous Preset and Clear and Inverted and Non-Inverted Outputs



FDCPEN

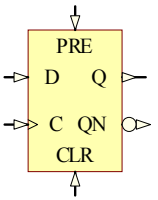
FDCPEN is a D type positive edge flip-flop with clock enable (CE) and asynchronous preset (PRE) and clear (CLR) and inverted (QN) and non-inverted (Q) outputs.

When clear (CLR) is High, all other inputs are ignored and outputs Q and QN are set to Low and High respectively. When clear (CLR) is Low and preset (PRE) is High, all other inputs are ignored and outputs Q and QN are set to High and Low respectively. When clear (CLR) and preset (PRE) is Low and clock enable is High, input data (D) is transferred to the outputs on the Low-to-High clock (C) transition. When clear (CLR), preset (PRE) and clock enable (CE) are Low, clock (C) transition and input data (D) is ignored and outputs do not change states.

Inputs					Outputs	
CLR	**PRE**	**CE**	**C**	**D**	**Q**	**QN**
1	x	x	x	x	0	1
0	1	x	x	x	1	0
0	0	0	x	x	Q_0	QN_0
0	0	1	↑	d	d	$\overline{d}$

FDCPN

D-Type Flip-Flop with Asynchronous Preset and Clear and Inverted and Non-Inverted Outputs



FDCPN

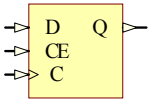
FDCPN is a D type positive edge flip-flop with asynchronous preset (PRE) and clear (CLR) and inverted (QN) and non-inverted (Q) outputs.

When clear (CLR) is High, all other inputs are ignored and outputs Q and QN are set to Low and High respectively. When clear (CLR) is Low and preset (PRE) is High, clock (C) transition and input data (D) is ignored and outputs Q and QN are set to High and Low respectively. When clear (CLR) and preset (PRE) are Low, input data (D) is transferred to outputs on the Low-to-High clock (C) transition.

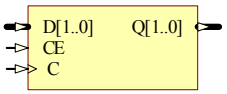
Inputs				Outputs	
CLR	**PRE**	**C**	**D**	**Q**	**QN**
1	x	x	x	0	1
0	1	x	x	1	0
0	0	↑	d	d	$\overline{d}$

FDE, FD2E, FD4E, FD8E, FD16E, FD32E

D-Type Flip-Flop with Clock Enable



FDE

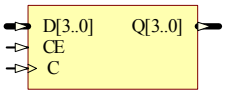


FD2EB

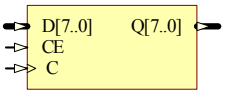
FDES, FD2E, FD4E, FD8E, FD16E and FD32E are, respectively 1-, 2-, 4-, 8-, 16-, 32-Bit D-type positive edge flip-flops with clock enable (CE).

When clock enable (CE) is High, input data (D) is transferred to the output (D) during Low-to-High clock transition. When clock enable is Low, the output does not change from its previous state.

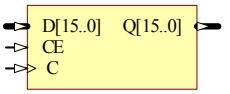
Inputs			Output
CE	D	C	Q
1	D	↑	D
0	x	x	Qo



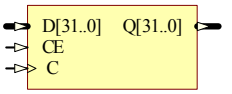
FD4EB



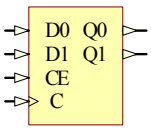
FD8EB



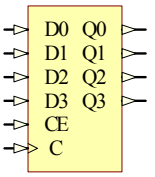
FD16EB



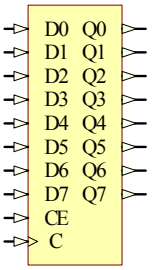
FD32EB



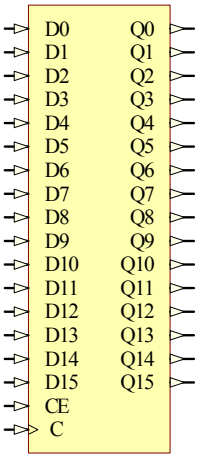
FD2ES



FD4ES



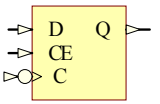
FD8ES



FD16ES

FDE_1

D-Type Negative Edge Flip-Flop with Clock Enable



FDE_1

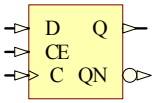
FDE_1 is a D-type negative edge flip-flop with clock enable (CE).

When clock enable (CE) is High, input data (D) is transferred to the output (D) during High-to-Low clock transition. When clock enable is Low, the output does not change from its previous state.

Inputs			Output
CE	**D**	**C**	**Q**
1	D	↓	D
0	x	x	Qo

FDEN

D Flip-Flop with Clock Enable and Inverted and Non-Inverted Outputs



FDEN

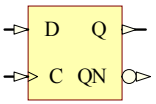
FDEN is D-type flip-flop with clock enable and inverted and non-Inverted outputs.

When clock enable (CE) is High, input data (D) is transferred to the outputs Q and QN during the Low-to-High clock transition. When clock enable is Low, the output does not change from its previous state.

Inputs			Output	
CE	**D**	**C**	**Q**	**QN**
1	D	↑	D	$\overline{D}$
0	x	x	Qo	QN_0

FDN

D-Type Flip-Flop with Inverted and Non-Inverted Outputs



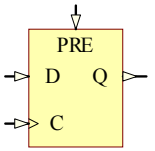
FDN

FDN is a D-type positive edge trigger flip-flop with inverted (QN) and non-inverted (Q) output. Input data (D) is loaded into the non inverted (Q) and inverted (QN) outputs during the Low-to-High clock (C) transition.

Inputs		Outputs	
C	**D**	**Q**	**QN**
↑	d	d	$\bar{d}$

FDP, FD2P, FD4P, FD8P, FD16P, FD32P

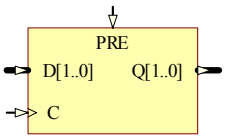
D-Type Flip-Flop with Asynchronous Preset



FDP

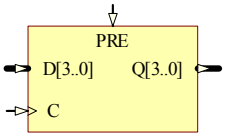
FDP, FD2P, FD4P, FD8P, FD16P, FD32P are, respectively 1-, 2-, 4-, 8-, 16-, 32-Bit D-type positive edge flip-flop with asynchronous preset (PRE).

When preset (PRE) is High, all other inputs are ignored and output (Q) is set to High. When preset (PRE) is Low, Input data (D) is loaded into the output (Q) during Low-to-High clock (C) transition.

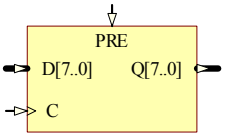


FD2PB

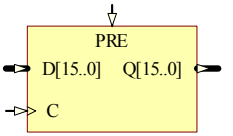
Inputs			Output
PRE	C	D	Q
1	x	x	1
0	↑	d	d



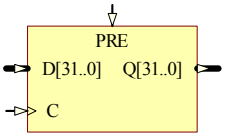
FD4PB



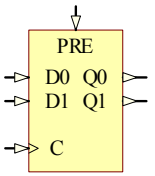
FD8PB



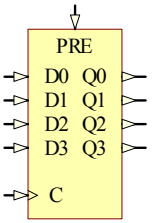
FD16PB



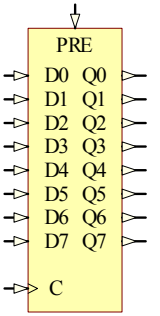
FD32PB



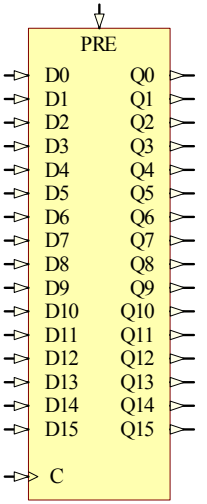
FD2PS



FD4PS



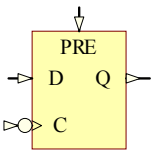
FD8PS



FD16PS

FDP_1

D-Type Negative Edge Flip-Flop with Asynchronous Preset



FDP_1

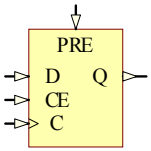
FDP_1 is a D-type negative edge flip-flop with asynchronous preset (PRE).

When preset (PRE) is High, all other inputs are ignored and output (Q) is set to High. When preset (PRE) is Low, Input data (D) is loaded into the output (Q) during High-to-Low clock (C) transition.

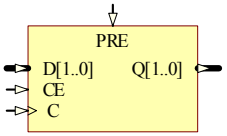
Inputs			Output
PRE	**C**	**D**	**Q**
1	x	x	1
0	↓	d	d

FDPE, FD2PE, FD4PE, FD8PE, FD16PE, FD32PE

D-Type Flip-Flop with Clock Enable and Asynchronous Preset



FDPE

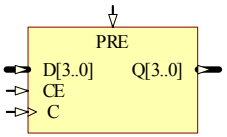


FD2PEB

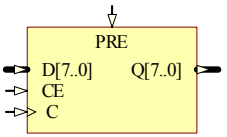
FDPE, FD2PE, FD4PE, FD8PE, FD16PE and FD32PE are, respectively 1-, 2-, 4-, 8-, 16-, 32-Bit D type positive edge flip-flops with clock enable (CE) and asynchronous preset (PRE).

When preset (PRE) is High, all other inputs are ignored and output (Q) is set to High. When preset (PRE) and clock enable (CE) is High, input data (D) is transferred to the output (Q) during Low-to-High clock (C) transition. When preset (PRE) and clock enable (CE) are Low, input data (D) and clock transition are ignored and output (Q) does not change state.

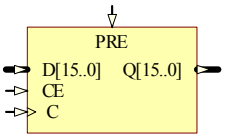
Inputs				Output
PRE	CE	C	D	Q
1	x	x	x	1
0	0	x	x	Q_0
0	1	↑	d	d



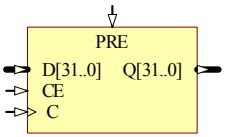
FD4PEB



FD8PEB

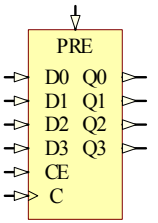


FD16PEB

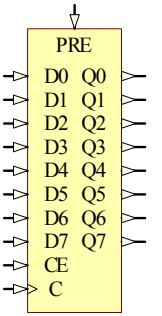


FD32PEB

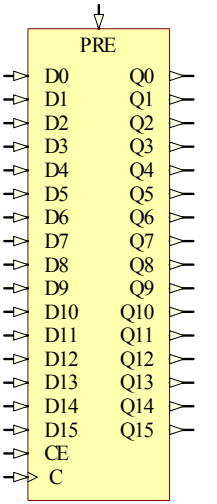
FD2PES



FD4PES



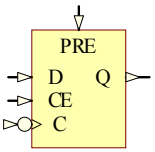
FD8PES



FD16PES

FDPE_1

D-Type Negative Edge Flip-Flop with Clock Enable and Asynchronous Preset



FDPE_1

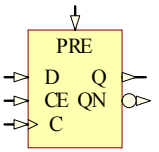
FDPE_1 is a D type negative edge flip-flop with clock enable (CE) and asynchronous preset (PRE).

When preset (PRE) is High, all other inputs are ignored and output (Q) is set to High. When preset (PRE) is Low and clock enable (CE) is High, input data (D) is transferred to the output (Q) during High-to-Low clock (C) transition. When preset (PRE) and clock enable (CE) are Low, input data (D) and clock transition are ignored and output (Q) does not change state.

Inputs				Output
PRE	**CE**	**C**	**D**	**Q**
1	x	x	x	1
0	0	x	x	Q_0
0	1	↓	d	d

FDPEN

D-Type Flip-Flop with Clock Enable and Asynchronous Preset and Inverted and Non-Inverted Outputs



FDPEN

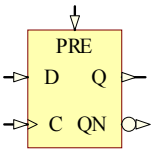
FDPEN is a D type positive edge flip-flop with clock enable (CE) and asynchronous preset (PRE).

When preset (PRE) is High, all other inputs are ignored and outputs Q and QN are set to High and Low respectively. When preset (PRE) is Low and clock enable (CE) is High, input data (D) is transferred to the outputs during Low-to-High clock (C) transition. When preset (PRE) and clock enable (CE) are Low, input data (D) and clock transition are ignored and the outputs do not change state.

Inputs				Outputs	
PRE	**CE**	**C**	**D**	**Q**	**QN**
1	x	x	x	1	0
0	0	x	x	Q_0	QN_0
0	1	↑	d	d	$\overline{d}$

FDPN

D-Type Flip-Flop with Asynchronous Preset and Inverted and Non-Inverted Outputs



FDPN

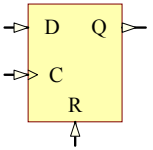
FDPN is a D-type positive edge flip-flop with asynchronous preset (PRE) and inverted (QN) and non-inverted (Q) outputs. They are available in 1, 2, 4 or 8 bit bus or single pin versions.

When preset (PRE) is High, all other inputs are ignored and output Q and QN are set to High and Low respectively. When preset (PRE) is Low, Input data (D) is loaded into the outputs during Low-to-High clock (C) transition.

Inputs			Outputs	
PRE	**C**	**D**	**Q**	**QN**
1	x	x	1	0
0	↑	d	d	$\overline{d}$

FDR, FD2R, FD4R, FD8R, FD16R, FD32R

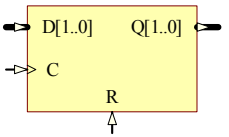
D-Type Flip-Flop with Synchronous Reset



FDR

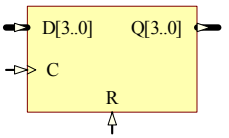
FDR, FD2R, FD4R, FD8R, FD16R and FD32R are, respectively 1-, 2-, 4-, 8-, 16-, 32-Bit D-type positive edge flip-flops with synchronous reset (R).

When reset (R) is High, input data (D) is ignored and output (Q) is set to Low on the Low-to-High clock (C) transition. When reset (R) is Low, input data (D) is transferred to the output (Q) on the Low-to-High clock (C) transition.

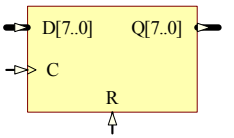


FD2RB

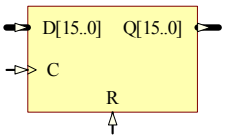
Inputs			Output
R	C	D	Q
1	↑	x	0
0	↑	d	d



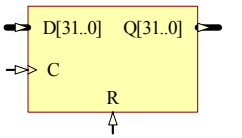
FD4RB



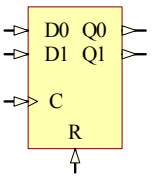
FD8RB



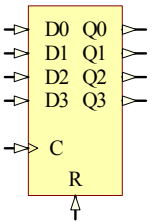
FD16RB



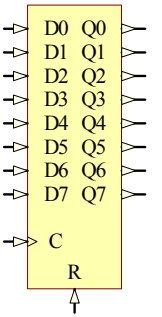
FD32RB



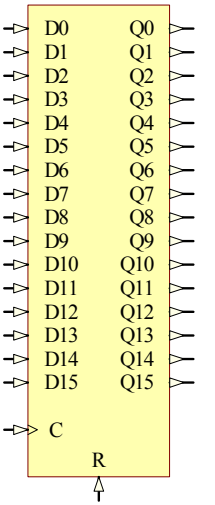
FD2RS



FD4RS



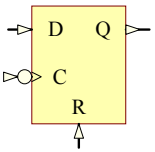
FD8RS



FD16RS

FDR_1

D-Type Negative Edge Flip-Flop with Synchronous Reset



FDR_1

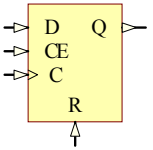
FDR_1 is a D-type negative edge flip-flop with synchronous reset (R). They are available in 1, 2, 4 or 8 bit bus or single pin versions.

When reset (R) is High, input data (D) is ignored and output (Q) is set to Low on the High-to-Low clock (C) transition. When reset (R) is Low, input data (D) is transferred to the output (Q) on the High-to-Low clock (C) transition.

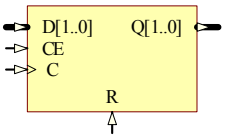
Inputs			Output
R	**C**	**D**	**Q**
1	↓	x	0
0	↓	d	d

FDRE, FD2RE, FD4RE, FD8RE, FD16RE, FD32RE

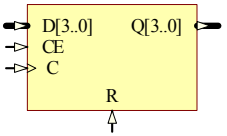
D-Type Flip-Flop with Clock Enable and Synchronous Reset



FDRE



FD2REB

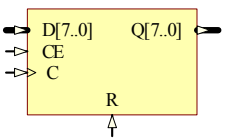


FD4REB

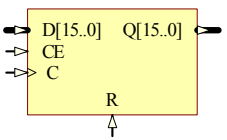
FDRE, FD2RE, FD4RE, FD8RE, FD16RE, FD32RE are, respectively 1-, 2-, 4-, 8-, 16-, 32-Bit D-type positive edge flip-flops with clock enable (CE) and synchronous reset (R).

When reset (R) is High, input data (D) and clock enable (CE) are ignored and output (Q) is set to Low on the Low-to-High clock (C) transition. When reset (R) is Low and clock enable (CE) is High, input data (D) is transferred to the output (Q) on Low-to-High clock (C) transition. When reset (R) and clock enable (CE) are Low, all other inputs are ignored and output (Q) does not change state.

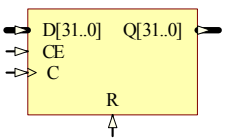
Inputs				Output
R	CE	C	D	Q
1	x	↑	x	0
0	0	x	x	Q_0
0	1	↑	d	d



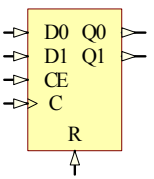
FD8REB



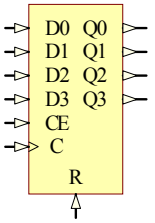
FD16REB



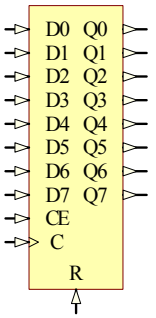
FD32REB



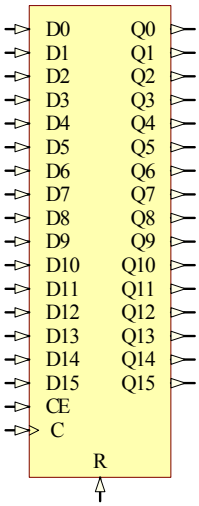
FD2RES



FD4RES



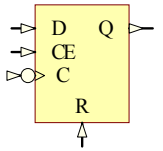
FD8RES



FD16RES

FDRE_1

D-Type Negative Edge Flip-Flop with Clock Enable and Synchronous Reset



FDRE_1

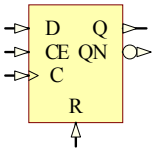
FDRE_1 is a D-type negative edge flip-flop with clock enable (CE) and synchronous reset (R).

When reset (R) is High, input data (D) and clock enable (CE) are ignored and output (Q) is set to Low on the High-to-Low clock (C) transition. When reset (R) is Low and clock enable (CE) is High, input data (D) is transferred to the output (Q) on High-to-Low clock (C) transition. When reset (R) and clock enable (CE) are Low, all other inputs are ignored and output (Q) does not change state.

Inputs				Output
R	**CE**	**C**	**D**	**Q**
1	x	↓	x	0
0	0	x	x	Q_0
0	1	↓	d	d

FDREN

D-Type Flip-Flop with Clock Enable and Synchronous Reset and Inverted and Non-Inverted Outputs



FDREN

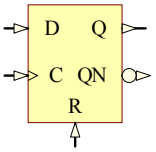
FDREN is a D-type positive edge flip-flop with clock enable (CE) and synchronous reset (R) and inverted (QN) and non-inverted (Q) outputs.

When reset (R) is High, input data (D) and clock enable (CE) are ignored and outputs Q and QN are set to Low and High, respectively, on the Low-to-High clock (C) transition. When reset (R) is Low and clock enable (CE) is High, input data (D) is transferred to the outputs on the Low-to-High clock (C) transition. When reset (R) and clock enable (CE) are Low, all other inputs are ignored and the outputs do not change state.

Inputs				Outputs	
R	**CE**	**C**	**D**	**Q**	**QN**
1	x	↑	x	0	1
0	0	x	x	Q_0	QN_0
0	1	↑	d	d	$\overline{d}$

FDRN

D-Type Flip-Flop with Synchronous Reset with Inverted and Non-Inverted Outputs



FDRN

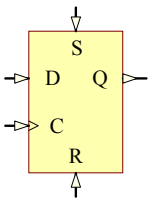
FDRN is a D-type positive edge flip-flop with synchronous reset (R) and inverted (QN) and non-inverted (QN) outputs.

When reset (R) is High, input data (D) is ignored and outputs Q and QN are set to Low and High respectively on the Low-to-High clock (C) transition. When reset (R) is Low, input data (D) is transferred to the outputs on the Low-to-High clock (C) transition.

Inputs			Outputs	
R	**C**	**D**	**Q**	**QN**
1	↑	x	0	1
0	↑	d	d	$\overline{d}$

FDRS, FD2RS, FD4RS, FD8RS, FD16RS, FD32RS

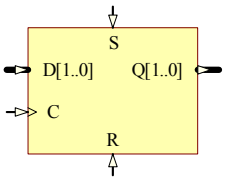
D-Type Flip-Flop with Synchronous Reset and Set



FDRS

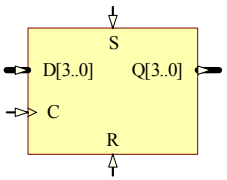
FDRS, FD2RS, FD4RS, FD8RS, FD16RS, FD32RS are, respectively 1-, 2-, 4-, 8-, 16-, 32-Bit D-type positive edge flip-flop with synchronous reset (R) and set (S).

When reset (R) is High, input data (D) and set (S) are ignored and output (Q) is set to Low on the Low-to-High clock (C) transition. When reset (R) is Low and set (S) is High, input data (D) is ignored and output (Q) is set to High on the Low-to-High clock (C) transition. When reset (R) and set (S) are Low, input data (D) is transferred to the output (Q) during Low-to-High clock (C) transition.

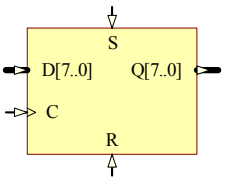


FD2RSB

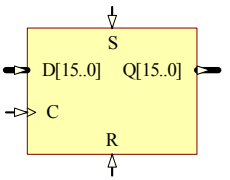
Inputs				Output
R	**S**	**C**	**D**	**Q**
1	x	↑	x	0
0	1	↑	x	1
0	0	↑	d	d



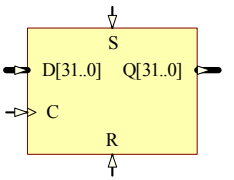
FD4RSB



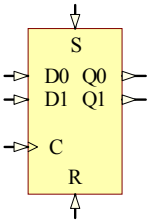
FD8RSB



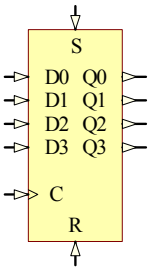
FD16RSB



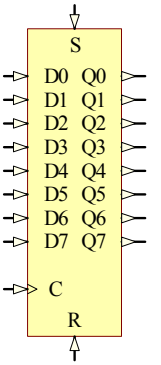
FD32RSB



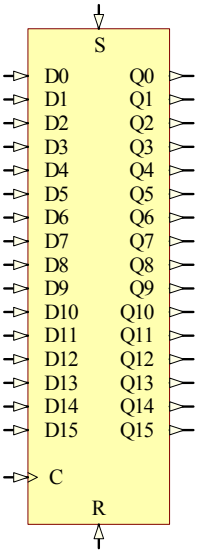
FD2RSS



FD4RSS



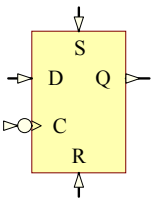
FD8RSS



FD16RSS

FDRS_1

D-Type Negative Edge Flip-Flop with Synchronous Reset and Set



FDRS_1

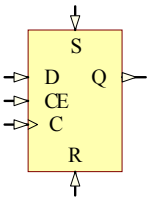
FDRS_1 is a D-type negative edge flip-flop with synchronous reset (R) and set (S).

When reset (R) is High, input data (D) and set (S) are ignored and output (Q) is set to Low on the High-to-Low clock (C) transition. When reset (R) is Low and set (S) is High, input data (D) is ignored and output (Q) is set to High on the High-to-Low clock (C) transition. When reset (R) and set (S) are Low, input data (D) is transferred to the output (Q) during High-to-Low clock (C) transition.

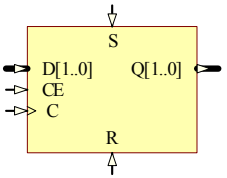
Inputs				Output
R	**S**	**C**	**D**	**Q**
1	x	↓	x	0
0	1	↓	x	1
0	0	↓	d	d

FDRSE, FD2RSE, FD4RSE, FD8RSE, FD16RSE, FD32RSE

D-Type Flip-Flop with Synchronous Reset and Set and Clock Enable



FDRSE

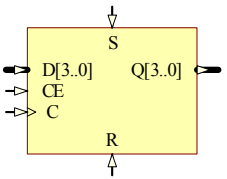


FD2RSEB

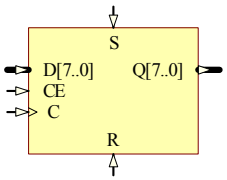
FDRSE, FD2RSE, FD4RSE, FD8RSE, FD16RSE, FD32RSE are, respectively 1-, 2-, 4-, 8-, 16-, 32-Bit D-type positive edge flip-flops with synchronous reset (R) and set (S) and clock enable (CE).

When reset (R) is High, input data (D), clock enable (CE) and set (S) are ignored and output (Q) is set to Low on the Low-to-High clock (C) transition. When reset (R) is Low and set (S) is High, input data (D) and clock enable (CE) is ignored and output (Q) is set to High on the Low-to-High clock (C) transition. When reset (R), and set (S) are Low and clock enable (CE) is High, input data (D) is transferred to the output (Q) during Low-to-High clock (C) transition. When reset (R), set (S) and clock enable (CE) Low, input data (D) and clock (C) transition is ignored and output (Q) does not change state.

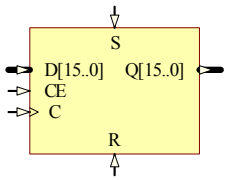
Inputs					Output
R	**S**	**CE**	**C**	**D**	**Q**
1	x	x	↑	x	0
0	1	x	↑	x	1
0	0	0	x	x	Q_0
0	0	1	↑	d	d



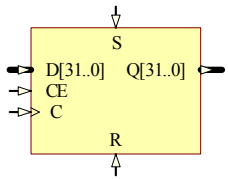
FD4RSEB



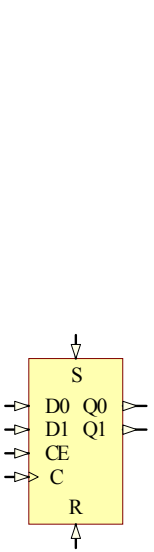
FD8RSEB



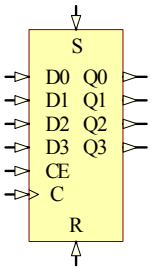
FD16RSEB



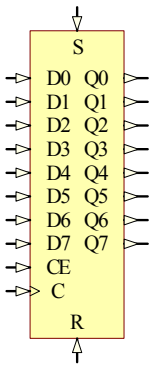
FD32RSEB



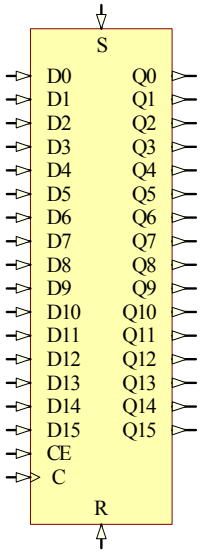
FD2RSES



FD4RSES



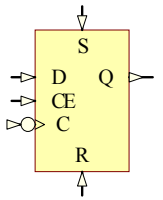
FD8RSES



FD16RSES

FDRSE_1

D-Type Negative Edge Flip-Flop with Synchronous Reset and Set and Clock Enable



FDRSE_1

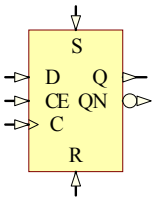
FDRSE_1 is a D-type negative edge flip-flop with synchronous reset (R), set (S) and clock enable (CE). They are available in 1, 2, 4 or 8 bit bus or single pin versions.

When reset (R) is High, input data (D), clock enable (CE) and set (S) are ignored and output (Q) is set to Low on the High-to-Low clock (C) transition. When reset (R) is Low and set (S) is High, input data (D) and clock enable (CE) are ignored and output (Q) is set to High on the High-to-Low clock (C) transition. When reset (R), and set (S) are Low and clock enable (CE) is High, input data (D) is transferred to the output (Q) during High-to-Low clock (C) transition. When reset (R), set (S) and clock enable (CE) Low, input data (D) and clock (C) transition is ignored and output (Q) does not change state.

Inputs					Output
R	**S**	**CE**	**C**	**D**	**Q**
1	x	x	↓	x	0
0	1	x	↓	x	1
0	0	0	x	x	Q_0
0	0	1	↓	d	d

FDRSEN

D-Type Flip-Flop with Synchronous Reset and Set and Clock Enable and Inverted and Non-Inverted Outputs



FDRSEN

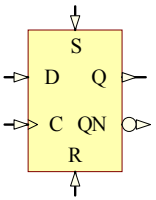
FDRSEN is a D-type positive edge flip-flop with synchronous reset (R), set (S) and clock enable (CE) and inverted (QN) and non-inverted (Q) outputs. They are available in 1, 2, 4 or 8 bit bus or single pin versions.

When reset (R) is High, input data (D), clock enable (CE) and set (S) are ignored and outputs Q and QN are set to Low and High respectively on the Low-to-High clock (C) transition. When reset (R) is Low and set (S) is High, input data (D) and clock enable (CE) is ignored and output Q and QN are set to High and Low respectively on the Low-to-High clock (C) transition. . When reset (R), and set (S) are Low and clock enable (CE) is High, input data (D) is transferred to the output (Q) during Low-to-High clock (C) transition. When reset (R), set (S) and clock enable (CE) Low, input data (D) and clock (C) transition is ignored and output (Q) does not change state.

Inputs					Outputs	
R	**S**	**CE**	**C**	**D**	**Q**	**QN**
1	x	x	↑	x	0	1
0	1	x	↑	x	1	0
0	0	0	x	x	Q_0	QN_0
0	0	1	↑	d	d	$\bar{d}$

FDRSN

D-Type Flip-Flop with Synchronous Reset and Set and Inverted and Non-Inverted Outputs



FDRSN

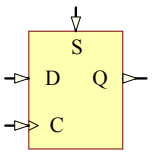
FDRSN is a D-type positive edge flip-flop with synchronous reset (R) and set (S) and inverted (QN) and non-inverted (Q) outputs.

When reset (R) is High, input data (D) and set (S) are ignored and outputs Q and QN are set to Low and High respectively on the Low-to-High clock (C) transition. When reset (R) is Low and set (S) is High, input data (D) is ignored and outputs Q and QN are set to High and Low respectively on the Low-to-High clock (C) transition. When reset (R) and set (S) are Low, input data (D) is transferred to the outputs during the Low-to-High clock (C) transition.

Inputs				Outputs	
R	**S**	**C**	**D**	**Q**	**QN**
1	x	↑	x	0	1
0	1	↑	x	1	0
0	0	↑	d	d	$\overline{d}$

FDS, FD2S, FD4S, FD8S, FD16S, FD32S

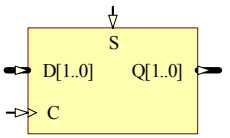
D-Type Flip-Flop with Synchronous Set



FDS

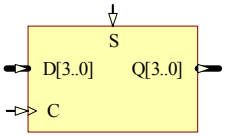
FDS, FD2S, FD4S, FD8S, FD16S, FD32S are, respectively 1-, 2-, 4-, 8-, 16-, 32-Bit D-type positive edge flip-flops with synchronous set (S).

When set (S) is High, input data is ignored and output (Q) is set to High on the Low-to-High clock (C) transition. When set (S) is Low, input data (D) is transferred to the output (Q) during Low-to-High clock (C) transition.

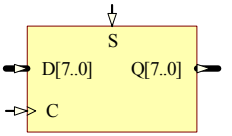


FD2SB

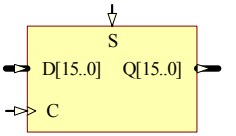
Inputs			Output
S	**C**	**D**	**Q**
1	↑	x	1
0	↑	d	d



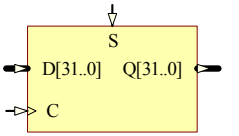
FD4SB



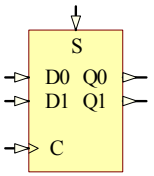
FD8SB



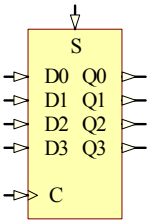
FD16SB



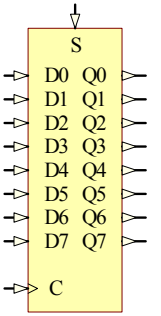
FD32SB



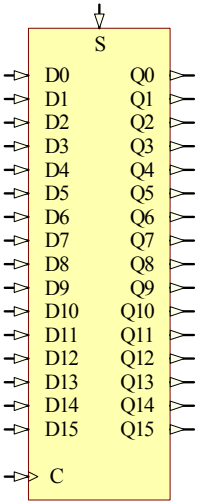
FD2SS



FD4SS



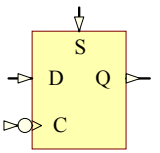
FD8SS



FD16SS

FDS_1

D-Type Negative Edge Flip-Flop with Synchronous Set



FDS_1

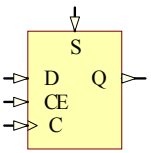
FDS_1 is a D-type negative edge flip-flop with synchronous set (S).

When set (S) is High, input data is ignored and output (Q) is set to High on the High-to-Low clock (C) transition. When set (S) is Low, input data (D) is transferred to the output (Q) during High-to-Low clock (C) transition.

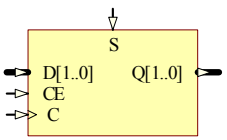
Inputs			Outputs
S	**C**	**D**	**Q**
1	↓	x	1
0	↓	d	d

FDSE, FD2SE, FD4SE, FD8SE, FD16SE, FD32SE

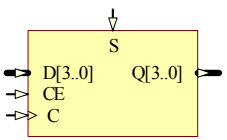
D-Type Flip-Flop with Clock Enable and Synchronous Set



FDSE



FD2SEB

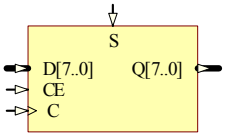


FD4SEB

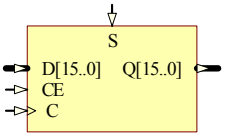
FDSE, FD2SE, FD4SE, FD8SE, FD16SE, FD32SE are, respectively 1-, 2-, 4-, 8-, 16-, 32-Bit D-type positive edge flip-flops with clock enable (CE) and synchronous set (S).

When set (S) is High, clock enable (CE) and input data (D) are ignored and output (Q) is set to High on the Low-to-High clock (C) transition. When set (S) is Low and clock enable (CE) is High, input data (D) is transferred to the output (Q) during Low-to-High clock (C) transition. When set (S) and clock enable (CE) are Low, input data (D) and clock (C) transition are ignored and output (Q) does not change state.

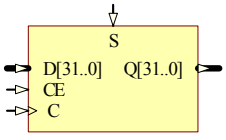
Inputs				Output
S	**CE**	**C**	**D**	**Q**
1	x	↑	x	1
0	0	x	x	Q_0
0	1	↑	d	d



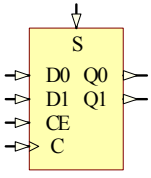
FD8SEB



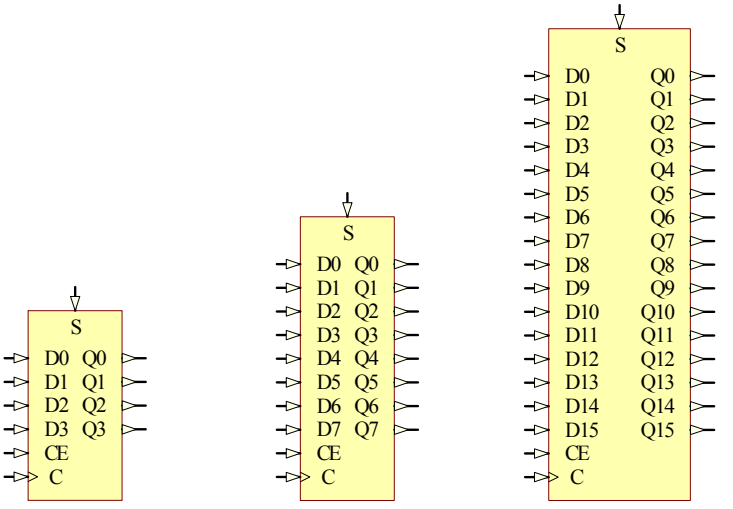
FD16SEB



FD32SEB

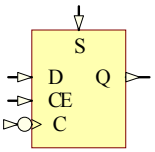


FD2SES


S
D0 Q0
D1 Q1
D2 Q2
D3 Q3
CE
C
FD4SES
S
D0 Q0
D1 Q1
D2 Q2
D3 Q3
D4 Q4
D5 Q5
D6 Q6
D7 Q7
CE
C
FD8SES
S
D0 Q0
D1 Q1
D2 Q2
D3 Q3
D4 Q4
D5 Q5
D6 Q6
D7 Q7
D8 Q8
D9 Q9
D10 Q10
D11 Q11
D12 Q12
D13 Q13
D14 Q14
D15 Q15
CE
C
FD16SES

FDSE_1

D-Type Negative Edge Flip-Flop with Clock Enable and Synchronous Set



FDSE_1

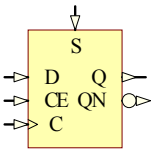
FDSE_1 is a D-type negative edge flip-flop with clock enable (CE) and synchronous set (S).

When set (S) is High, clock enable (CE) and input data (D) are ignored and output (Q) is set to High on the High- to-Low clock (C) transition. When set (S) is Low and clock enable (CE) is High, input data (D) is transferred to the output (Q) during High-to-Low clock (C) transition. When set (S) and clock enable (CE) are Low, input data (D) and clock (C) transition are ignored and output (Q) does not change state.

Inputs				Output
S	**CE**	**C**	**D**	**Q**
1	x	↓	x	1
0	0	x	x	Q_0
0	1	↓	d	d

FDSEN

D-Type Flip-Flop with Clock Enable and Synchronous Set and Inverted and Non-Inverted Outputs



FDSEN

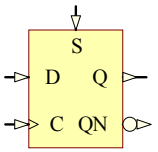
FDSEN is a D-type positive edge flip-flop with clock enable (CE) and synchronous set (S) and inverted (QN) and non-inverted (Q) outputs.

When set (S) is High, clock enable (CE) and input data (D) are ignored and outputs Q and QN are set to High and Low respectively on the Low-to-High clock (C) transition. When set (S) is Low and clock enable (CE) is High, input data (D) is transferred to the outputs during Low-to-High clock (C) transition. When set (S) and clock enable (CE) are Low, input data (D) and clock (C) transition are ignored and the outputs do not change state.

Inputs				Outputs	
S	**CE**	**C**	**D**	**Q**	**QN**
1	x	↑	x	1	0
0	0	x	x	Q_0	QN_0
0	1	↑	d	d	$\overline{d}$

FDSN

D-Type Flip-Flop with Synchronous Set and Inverted and Non-Inverted Outputs



FDSN

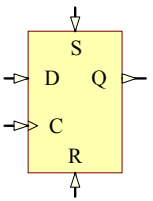
FDSN is a D-type positive edge flip-flop with synchronous set (S) and inverted (QN) and non-inverted (Q) outputs.

When set (S) is High, input data is ignored and outputs Q and QN are set to High and Low respectively on the Low-to-High clock (C) transition. When set (S) is Low, input data (D) is transferred to the outputs during Low-to-High clock (C) transition.

Inputs			Outputs	
S	**C**	**D**	**Q**	**QN**
1	↑	x	1	0
0	↑	d	d	$\overline{d}$

FDSR, FD2SR, FD4SR, FD8SR, FD16SR, FD32SR

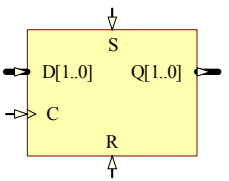
D-Type Flip-Flop with Synchronous Set and Reset



FDSR

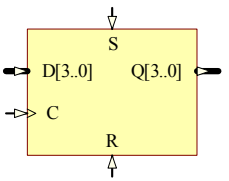
FDSR, FD2SR, FD4SR, FD8SR, FD16SR, FD32SR are, respectively 1-, 2-, 4-, 8-, 16-, 32-Bit D-type positive edge flip-flops with synchronous set (S) and reset (R).

When set (S) is High, input data (D) and reset (R) are ignored and output (Q) is set to High on the Low-to-High clock (C) transition. When set (S) is Low and reset (R) is High, input data (D) is ignored and output (Q) is set to Low on the Low-to-High clock (C) transition. When set (S) and reset (R) are Low, input data (D) is transferred to the output (Q) during Low-to-High clock (C) transition.

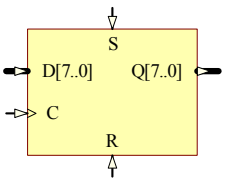


FD2SRB

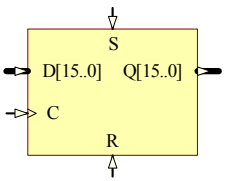
Inputs				Output
S	R	C	D	Q
1	x	↑	x	1
0	1	↑	x	0
0	0	↑	d	d



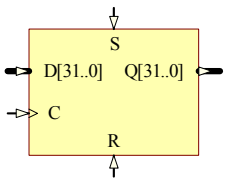
FD4SRB



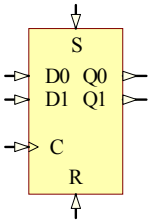
FD8SRB



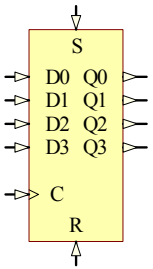
FD16SRB



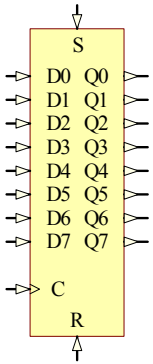
FD32SRB



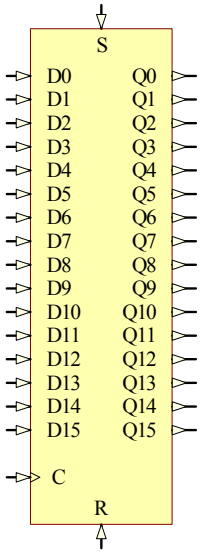
FD2SRS



FD4SRS



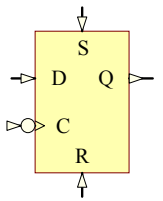
FD8SRS



FD16SRS

FDSR_1

D-Type Negative Edge Flip-Flop with Synchronous Set and Reset



FDSR_1

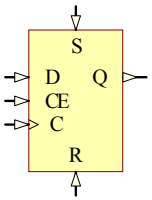
FDSR_1 is a D-type negative edge flip-flop with synchronous set (S) and reset (R).

When set (S) is High, input data (D) and reset (R) are ignored and output (Q) is set to High on the High-to-Low clock (C) transition. When set (S) is Low and reset (R) is High, input data (D) is ignored and output (Q) is set to Low on the High-to-Low clock (C) transition. When set (S) and reset (R) are Low, input data (D) is transferred to the output (Q) during High-to-Low clock (C) transition.

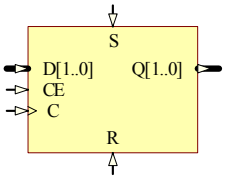
Inputs				Output
S	**R**	**C**	**D**	**Q**
1	x	↓	x	1
0	1	↓	x	0
0	0	↓	d	d

FDSRE, FD2SRE, FD4SRE, FD8SRE, FD16SRE, FD32SRE

D-Type Flip-Flop with Synchronous Set and Reset and Clock Enable



FDSRE

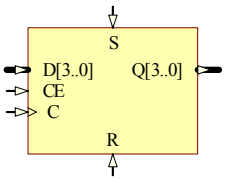


FD2SREB

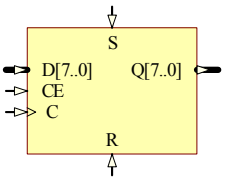
FDSRE, FD2SRE, FD4SRE, FD8SRE, FD16SRE, FD32SRE are, respectively 1-, 2-, 4-, 8-, 16-, 32-Bit D-type positive edge flip-flops with synchronous set (S) and reset (R) and clock enable (CE).

When set (S) is High, input data (D), reset (R) and clock enable (CE) are ignored and output (Q) is set to High on the Low-to-High clock (C) transition. When set (S) is Low and reset (R) is High, input data (D) and clock enable (CE) are ignored and output (Q) is set to Low on the Low-to-High (C) clock transition. When set (S) and reset (R) are Low and clock enable (CE) is High, input data (D) is transferred to the output (Q) during the Low-to-High clock (C) transition. When set (S), reset (R) and clock enable (CE) are Low, input data (D) and clock (C) transition are ignored and output (Q) does not change state.

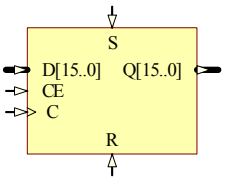
Inputs					Output
S	**R**	**CE**	**C**	**D**	**Q**
1	x	x	↑	x	1
0	1	x	↑	x	0
0	0	0	x	x	Q_0
0	0	1	↑	d	d



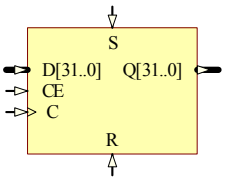
FD4SREB



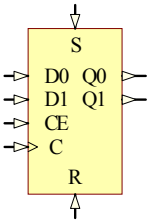
FD8SREB



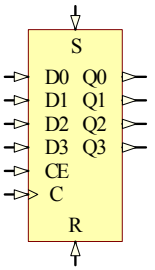
FD16SREB



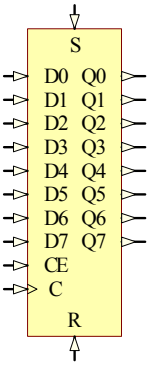
FD32SREB



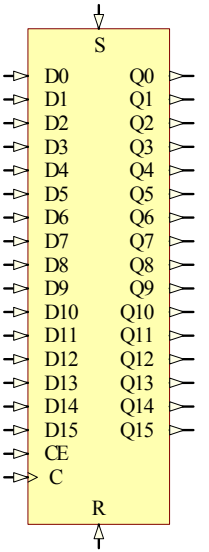
FD2SRES



FD4SRES



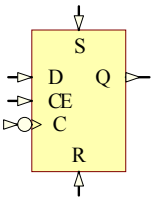
FD8SRES



FD16SRES

FDSRE_1

D-Type Negative Edge Flip-Flop with Synchronous Set and Reset and Clock Enable



FDSRE_1

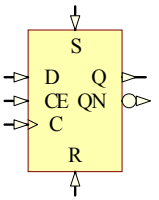
FDSRE_1 is a D-type negative edge flip-flop with synchronous set (S) and reset (R) and clock enable (CE).

When set (S) is High, input data (D), reset (R) and clock enable (CE) are ignored and output (Q) is set to High on the High-to-Low clock (C) transition. When set (S) is Low and reset (R) is High, input data (D) and clock enable (CE) are ignored and output (Q) is set to Low on the High-to-Low (C) clock transition. When set (S) and reset (R) are Low and clock enable (CE) is High, input data (D) is transferred to the output (Q) during the High-to-Low clock (C) transition. When set (S), reset (R) and clock enable (CE) are Low, input data (D) and clock (C) transition are ignored and output (Q) does not change state.

Inputs					Output
S	R	CE	C	D	Q
1	x	x	↓	x	1
0	1	x	↓	x	0
0	0	0	x	x	Q_0
0	0	1	↓	d	d

FDSREN

D-Type Flip-Flop with Synchronous Set and Reset and Clock Enable and Inverted and Non-Inverted Outputs



FDSREN

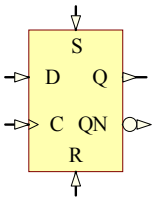
FDSREN is a D-type positive edge flip-flop with synchronous set (S), reset (R) and clock enable (CE) with inverted (QN) and non-inverted (Q) outputs.

When set (S) is High, input data (D), reset (R) and clock enable (CE) are ignored and outputs Q and QN are set to High and Low on the Low-to-High clock (C) transition. When set (S) is Low and reset (R) is High, input data (D) and clock enable (CE) are ignored and outputs Q and QN are set to Low and High on the Low-to-High (C) clock transition. When set (S) and reset (R) are Low and clock enable (CE) are High, input data (D) is transferred to the outputs during the Low-to-High clock (C) transition. When set (S), reset (R) and clock enable (CE) are Low, input data (D) and clock (C) transition is ignored and outputs do not change states.

Inputs					Outputs	
S	**R**	**CE**	**C**	**D**	**Q**	**QN**
1	x	x	↑	x	1	0
0	1	x	↑	x	0	1
0	0	0	x	x	Q_0	QN_0
0	0	1	↑	d	d	$\overline{d}$

FDSRN

D-Type Flip-Flop with Synchronous Set and Reset and Inverted and Non-Inverted Outputs



FDSRN

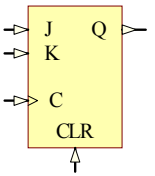
FDSRN is a D-type positive edge flip-flop with synchronous set (S) and reset (R) and inverted and non-inverted outputs.

When set (S) is High, input data (D) and reset (R) are ignored and outputs Q and QN are set to High and Low respectively on the Low-to-High clock (C) transition. When set (S) is Low and reset (R) is High, input data (D) is ignored and outputs Q and QN are set to Low and High respectively on the Low-to-High clock (C) transition. When set (S) and reset (R) are Low, input data (D) is transferred to the outputs during Low-to-High clock (C) transition.

Inputs				Outputs	
S	**R**	**C**	**D**	**Q**	**QN**
1	x	↑	x	1	0
0	1	↑	x	0	1
0	0	↑	d	d	$\overline{d}$

FJKC

J-K Flip-Flop with Asynchronous Clear



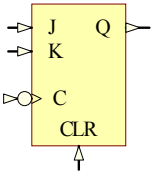
FJKC

FJKC is a single J-K-type flip-flop with J, K, and asynchronous clear (CLR) inputs and data output (Q). When High, the asynchronous clear (CLR) input overrides all other inputs and resets the Q output Low. When CLR is Low, the output responds to the state of the J and K inputs, as shown in the following truth table, during the Low-to-High clock (C) transition.

Inputs				Output
CLR	**J**	**K**	**C**	**Q**
1	X	X	X	0
0	0	0	↑	Q_0
0	0	1	↑	0
0	1	0	↑	1
0	1	1	↑	Toggle

FJKC_1

J-K Negative Edge Flip-Flop with Asynchronous Clear



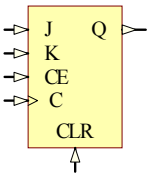
FJKC_1

FJKC_1 is a single J-K-type negative-edge triggered flip-flop with J, K, and asynchronous clear (CLR) inputs and data output (Q). When High, the asynchronous clear (CLR) input overrides all other inputs and resets the Q output Low. When CLR is Low, the output responds to the state of the J and K inputs, as shown in the following truth table, during the High-to-Low clock (C) transition.

Inputs				Output
CLR	**J**	**K**	**C**	**Q**
1	X	X	X	0
0	0	0	↓	Q_0
0	0	1	↓	0
0	1	0	↓	1
0	1	1	↓	Toggle

FJKCE

J-K Flip-Flop with Clock Enable and Asynchronous Clear



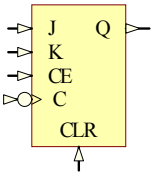
FJKCE

FJKCE is a single J-K-type flip-flop with J, K, clock enable (CE), and asynchronous clear (CLR) inputs and data output (Q). When High, the asynchronous clear (CLR) overrides all other inputs and resets the Q output Low. When CLR is Low and CE is High, Q responds to the state of the J and K inputs, as shown in the following truth table, during the Low-to-High clock transition. When CE is Low, the clock transitions are ignored and the state of Q remains unchanged.

Inputs					Output
CLR	**CE**	**J**	**K**	**C**	**Q**
1	X	X	X	X	0
0	0	X	X	X	Q_0
0	1	0	0	X	Q_0
0	1	0	1	↑	0
0	1	1	0	↑	1
0	1	1	1	↑	Toggle

FJKCE_1

J-K Negative Edge Flip-Flop with Clock Enable and Asynchronous Clear



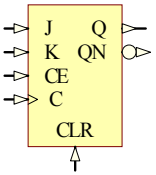
FJKCE_1

FJKCE_1 is a single J-K-type negative-edge triggered flip-flop with J, K, clock enable (CE), and asynchronous clear (CLR) inputs and data output (Q). When High, the asynchronous clear (CLR) overrides all other inputs and resets the Q output Low. When CLR is Low and CE is High, Q responds to the state of the J and K inputs, as shown in the following truth table, during the High-to-Low clock transition. When CE is Low, the clock transitions are ignored and the state of Q remains unchanged.

Inputs					Output
CLR	**CE**	**J**	**K**	**C**	**Q**
1	X	X	X	X	0
0	0	X	X	X	Q_0
0	1	0	0	X	Q_0
0	1	0	1	↓	0
0	1	1	0	↓	1
0	1	1	1	↓	Toggle

FJKCEN

J-K Flip-Flop with Clock Enable, Asynchronous Clear and Inverted and Non-Inverted Outputs



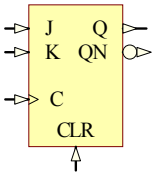
FJKCEN

FJKCEN is a single J-K-type flip-flop with J, K, clock enable (CE), and asynchronous clear (CLR) inputs and inverted (QN) and non-inverted (Q) outputs. When High, the asynchronous clear (CLR) overrides all other inputs and resets the Q output Low and the QN output High. When CLR is Low and CE is High, Q and QN respond to the state of the J and K inputs, as shown in the following truth table, during the Low-to-High clock transition. When CE is Low, the clock transitions are ignored and the state of the outputs remains unchanged.

Inputs					Outputs	
CLR	**CE**	**J**	**K**	**C**	**Q**	**QN**
1	X	X	X	X	0	1
0	0	X	X	X	Q_0	QN_0
0	1	0	0	X	Q_0	QN_0
0	1	0	1	↑	0	1
0	1	1	0	↑	1	0
0	1	1	1	↑	Toggle	Toggle

FJKCN

J-K Flip-Flop with Asynchronous Clear and Inverted and Non-Inverted Outputs



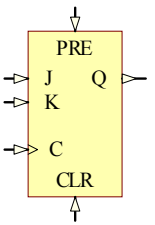
FJKCN

FJKCN is a single J-K-type flip-flop with J, K, and asynchronous clear (CLR) inputs and inverted (QN) and non-inverted (Q) outputs. When High, the asynchronous clear (CLR) input overrides all other inputs and resets the Q output Low and the QN output to High. When CLR is Low, the outputs respond to the state of the J and K inputs, as shown in the following truth table, during the Low-to-High clock (C) transition.

Inputs				Outputs	
CLR	**J**	**K**	**C**	**Q**	**QN**
1	X	X	X	0	1
0	0	0	↑	Q_0	QN_0
0	0	1	↑	0	1
0	1	0	↑	1	0
0	1	1	↑	Toggle	Toggle

FJKCP

J-K Flip-Flop with Asynchronous Clear and Preset



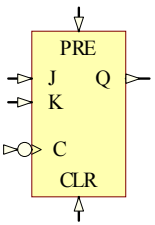
FJKCP

FJKCP is a single J-K-type flip-flop with J, K, asynchronous clear (CLR), and asynchronous preset (PRE) inputs and data output (Q). When High, the asynchronous clear (CLR) input overrides all other inputs and resets the Q output Low. When the asynchronous preset (PRE) is High, and CLR set to Low all other inputs are overridden and the Q output is set High. When CLR and PRE are Low, Q responds to the state of the J and K inputs during the Low-to-High clock transition, as shown in the following truth table.

Inputs					Output
CLR	**PRE**	**J**	**K**	**C**	**Q**
1	X	X	X	X	0
0	1	X	X	X	1
0	0	0	0	X	Q_0
0	0	0	1	↑	0
0	0	1	0	↑	1
0	0	1	1	↑	Toggle

FJKCP_1

J-K Negative Edge Flip-Flop with Asynchronous Clear and Preset



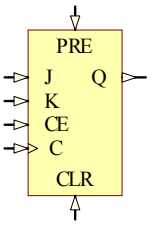
FJKCP_1

FJKCP_1 is a single J-K-type negative-edge triggered flip-flop with J, K, asynchronous clear (CLR), and asynchronous preset (PRE) inputs and data output (Q). When High, the asynchronous clear (CLR) input overrides all other inputs and resets the Q output Low. When the asynchronous preset (PRE) is High, and CLR set to Low all other inputs are overridden and the Q output is set High. When CLR and PRE are Low, Q responds to the state of the J and K inputs during the High-to-Low clock transition, as shown in the following truth table.

Inputs					Output
CLR	**PRE**	**J**	**K**	**C**	**Q**
1	X	X	X	X	0
0	1	X	X	X	1
0	0	0	0	X	Q_0
0	0	0	1	↓	0
0	0	1	0	↓	1
0	0	1	1	↓	Toggle

FJKCPE

J-K Flip-Flop with Asynchronous Clear and Preset and Clock Enable



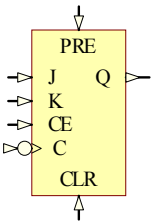
FJKCPE

FJKCPE is a single J-K-type flip-flop with J, K, asynchronous clear (CLR), asynchronous preset (PRE), and clock enable (CE) inputs and data output (Q). When High, the asynchronous clear (CLR) input overrides all other inputs and resets the Q output Low. When the asynchronous preset (PRE) is High and CLR is Low, all other inputs are ignored and Q is set High. When CLR and PRE are Low and CE is High, Q responds to the state of the J and K inputs, as shown in the following truth table, during the Low-to-High clock transition. Clock transitions are ignored and the state of Q remains unchanged when CE is Low.

Inputs						Output
CLR	**PRE**	**CE**	**J**	**K**	**C**	**Q**
1	X	X	X	X	X	0
0	1	X	X	X	X	1
0	0	0	X	X	X	Q_0
0	0	1	0	0	X	Q_0
0	0	1	0	1	↑	0
0	0	1	1	0	↑	1
0	0	1	1	1	↑	Toggle

FJKCPE_1

J-K Negative Edge Flip-Flop with Asynchronous Clear and Preset and Clock Enable



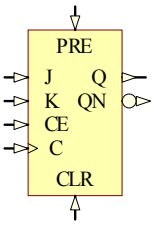
FJKCPE_1

FJKCPE_1 is a single J-K-type negative-edge triggered flip-flop with J, K, asynchronous clear (CLR), asynchronous preset (PRE), and clock enable (CE) inputs and data output (Q). When High, the asynchronous clear (CLR) input overrides all other inputs and resets the Q output Low. When the asynchronous preset (PRE) is High and CLR is Low, all other inputs are ignored and Q is set High. When CLR and PRE are Low and CE is High, Q responds to the state of the J and K inputs, as shown in the following truth table, during the High-to-Low clock transition. Clock transitions are ignored and the state of Q remains unchanged when CE is Low.

Inputs						Output
CLR	**PRE**	**CE**	**J**	**K**	**C**	**Q**
1	X	X	X	X	X	0
0	1	X	X	X	X	1
0	0	0	X	X	X	Q_0
0	0	1	0	0	X	Q_0
0	0	1	0	1	↓	0
0	0	1	1	0	↓	1
0	0	1	1	1	↓	Toggle

FJKCPEN

J-K Flip-Flop with Asynchronous Clear, Preset, Clock Enable and Inverted and Non-Inverted Outputs



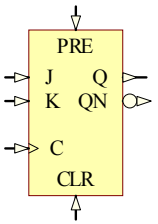
FJKCPEN

FJKCPEN is a single J-K-type flip-flop with J, K, asynchronous clear (CLR), asynchronous preset (PRE), and clock enable (CE) inputs and inverted (QN) and non-inverted (Q) outputs. When High, the asynchronous clear (CLR) input overrides all other inputs and resets the Q and QN output Low and High respectively. When the asynchronous preset (PRE) is High and CLR is Low, all other inputs are ignored and Q is set High whilst QN is set Low. When CLR and PRE are Low and CE is High, Q and QN respond to the state of the J and K inputs, as shown in the following truth table, during the Low-to-High clock transition. Clock transitions are ignored when CE is Low and the state of the outputs remains unchanged.

Inputs						Outputs	
CLR	**PRE**	**CE**	**J**	**K**	**C**	**Q**	**QN**
1	X	X	X	X	X	0	1
0	1	X	X	X	X	1	0
0	0	0	X	X	X	Q_0	QN_0
0	0	1	0	0	X	Q_0	QN_0
0	0	1	0	1	↑	0	1
0	0	1	1	0	↑	1	0
0	0	1	1	1	↑	Toggle	Toggle

FJKCPN

J-K Flip-Flop with Asynchronous Clear, Preset and Inverted and Non-Inverted Outputs



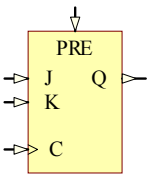
FJKCPN

FJKCPN is a single J-K-type flip-flop with J, K, asynchronous clear (CLR), and asynchronous preset (PRE) inputs and inverted (QN) and non-inverted (Q) outputs. When High, the asynchronous clear (CLR) input overrides all other inputs and resets the Q output Low and the QN output to High. When the asynchronous preset (PRE) is High and CLR is Low, all other inputs are ignored and Q is set High whilst QN is set Low. When CLR and PRE are Low, Q and QN respond to the state of the J and K inputs during the Low-to-High clock transition, as shown in the following truth table.

Inputs					Outputs	
CLR	**PRE**	**J**	**K**	**C**	**Q**	**QN**
1	X	X	X	X	0	1
0	1	X	X	X	1	0
0	0	0	0	X	Q_0	QN_0
0	0	0	1	↑	0	1
0	0	1	0	↑	1	0
0	0	1	1	↑	Toggle	Toggle

FJKP

J-K Flip-Flop with Asynchronous Preset



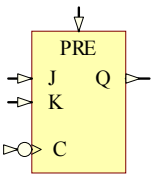
FJKP

FJKP is a single J-K-type flip-flop with J, K, and asynchronous preset (PRE) inputs and data output (Q). When High, the asynchronous preset (PRE) input overrides all other inputs and sets the Q output High. When PRE is Low, the Q output responds to the state of the J and K inputs, as shown in the following truth table, during the Low-to-High clock transition.

Inputs				Output
PRE	**J**	**K**	**C**	**Q**
1	X	X	X	1
0	0	0	X	Q_0
0	0	1	↑	0
0	1	0	↑	1
0	1	1	↑	Toggle

FJKP_1

J-K Negative Edge Flip-Flop with Asynchronous Preset



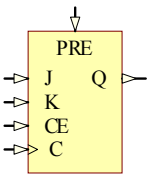
FJKP_1

FJKP_1 is a single J-K-type negative-edge triggered flip-flop with J, K, and asynchronous preset (PRE) inputs and data output (Q). When High, the asynchronous preset (PRE) input overrides all other inputs and sets the Q output High. When PRE is Low, the Q output responds to the state of the J and K inputs, as shown in the following truth table, during the High-to-Low clock transition.

Inputs				Output
PRE	**J**	**K**	**C**	**Q**
1	X	X	X	1
0	0	0	X	Q_0
0	0	1	↓	0
0	1	0	↓	1
0	1	1	↓	Toggle

FJKPE

J-K Flip-Flop with Clock Enable and Asynchronous Preset



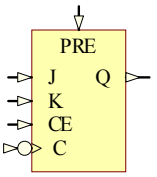
FJKPE

FJKPE is a single J-K-type flip-flop with J, K, clock enable (CE), and asynchronous preset (PRE) inputs and data output (Q). When High, the asynchronous preset (PRE) overrides all other inputs and sets the Q output High. When PRE is Low and CE is High, the Q output responds to the state of the J and K inputs, as shown in the truth table, during the Low-to-High clock (C) transition. Clock transitions are ignored and the state of Q remains unchanged when CE is Low.

Inputs					Output
PRE	**CE**	**J**	**K**	**C**	**Q**
1	X	X	X	X	1
0	0	X	X	X	Q_0
0	1	0	0	X	Q_0
0	1	0	1	↑	0
0	1	1	0	↑	1
0	1	1	1	↑	Toggle

FJKPE_1

J-K Negative Edge Flip-Flop with Clock Enable and Asynchronous Preset



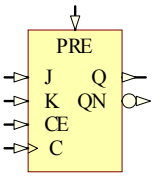
FJKPE_1

FJKPE_1 is a single J-K-type negative-edge triggered flip-flop with J, K, clock enable (CE), and asynchronous preset (PRE) inputs and data output (Q). When High, the asynchronous preset (PRE) overrides all other inputs and sets the Q output High. When PRE is Low and CE is High, the Q output responds to the state of the J and K inputs, as shown in the truth table, during the High-to-Low clock (C) transition. Clock transitions are ignored and the state of Q remains unchanged when CE is Low.

Inputs					Output
PRE	**CE**	**J**	**K**	**C**	**Q**
1	X	X	X	X	1
0	0	X	X	X	Q_0
0	1	0	0	X	Q_0
0	1	0	1	↓	0
0	1	1	0	↓	1
0	1	1	1	↓	Toggle

FJKPEN

J-K Flip-Flop with Clock Enable, Asynchronous Preset and Inverted and Non-Inverted Outputs



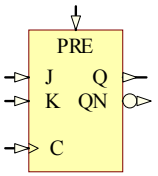
FJKPEN

FJKPEN is a single J-K-type flip-flop with J, K, clock enable (CE), and asynchronous preset (PRE) inputs and inverted (QN) and non-inverted (Q) outputs. When High, the asynchronous preset (PRE) overrides all other inputs and sets outputs Q to High and QN to Low. When PRE is Low and CE is High, the Q and QN outputs respond to the state of the J and K inputs, as shown in the truth table, during the Low-to-High clock (C) transition. Clock transitions are ignored and the state of the outputs remains unchanged when CE is Low.

Inputs					Outputs	
PRE	**CE**	**J**	**K**	**C**	**Q**	**QN**
1	X	X	X	X	1	0
0	0	X	X	X	Q_0	QN_0
0	1	0	0	X	Q_0	QN_0
0	1	0	1	↑	0	1
0	1	1	0	↑	1	0
0	1	1	1	↑	Toggle	Toggle

FJKPN

J-K Flip-Flop with Asynchronous Preset and Inverted and Non-Inverted Outputs



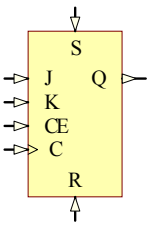
FJKPN

FJKPN is a single J-K-type flip-flop with J, K, and asynchronous preset (PRE) inputs and inverted (QN) and non-inverted (Q) outputs. When High, the asynchronous preset (PRE) input overrides all other inputs and sets the outputs Q to High and QN to Low. When PRE is Low, the Q and QN outputs respond to the state of the J and K inputs, as shown in the following truth table, during the Low-to-High clock transition.

Inputs				Outputs	
PRE	**J**	**K**	**C**	**Q**	**QN**
1	X	X	X	1	0
0	0	0	X	Q_0	QN_0
0	0	1	↑	0	1
0	1	0	↑	1	0
0	1	1	↑	Toggle	Toggle

FJKRSE

J-K Flip-Flop with Clock Enable and Synchronous Reset and Set



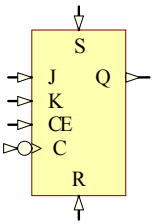
FJKRSE

FJKRSE is a single J-K-type flip-flop with J, K, synchronous reset (R), synchronous set (S), and clock enable (CE) inputs and data output (Q). When synchronous reset (R) is High, all other inputs are ignored and output Q is reset Low. (Reset has precedence over Set.) When synchronous set (S) is High and R is Low, output Q is set High. When R and S are Low and CE is High, output Q responds to the state of the J and K inputs, according to the following truth table, during the Low-to-High clock (C) transition. Clock transitions are ignored and the state of Q remains unchanged when CE is Low.

Inputs						Outputs
R	**S**	**CE**	**J**	**K**	**C**	**Q**
1	X	X	X	X	↑	0
0	1	X	X	X	↑	1
0	0	0	X	X	X	Q_0
0	0	1	0	0	X	Q_0
0	0	1	0	1	↑	0
0	0	1	1	1	↑	Toggle
0	0	1	1	0	↑	1

FJKRSE_1

J-K Negative Edge Flip-Flop with Clock Enable and Synchronous Reset and Set



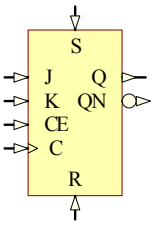
FJKRSE_1

FJKRSE_1 is a single J-K-type negative-edge triggered flip-flop with J, K, synchronous reset (R), synchronous set (S), and clock enable (CE) inputs and data output (Q). When synchronous reset (R) is High, all other inputs are ignored and output Q is reset Low. (Reset has precedence over Set.) When synchronous set (S) is High and R is Low, output Q is set High. When R and S are Low and CE is High, output Q responds to the state of the J and K inputs, according to the following truth table, during the High-to-Low clock (C) transition. Clock transitions are ignored and the state of Q remains unchanged when CE is Low.

Inputs						Output
R	**S**	**CE**	**J**	**K**	**C**	**Q**
1	X	X	X	X	↓	0
0	1	X	X	X	↓	1
0	0	0	X	X	X	Q_0
0	0	1	0	0	X	Q_0
0	0	1	0	1	↓	0
0	0	1	1	1	↓	Toggle
0	0	1	1	0	↓	1

FJKRSEN

J-K Flip-Flop with Clock Enable, Synchronous Reset and Set and Inverted and Non-Inverted Outputs



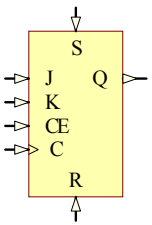
FJKRSEN

FJKRSEN is a single J-K-type flip-flop with J, K, synchronous reset (R), synchronous set (S), and clock enable (CE) inputs and inverted (QN) and non-inverted (Q) outputs. When synchronous reset (R) is High, all other inputs are ignored and outputs Q is reset Low whilst QN is reset High. (Reset has precedence over Set.) When synchronous set (S) is High and R is Low, output Q is set High whilst QN is set Low. When R and S are Low and CE is High, outputs Q and QN respond to the state of the J and K inputs, according to the following truth table, during the Low-to-High clock (C) transition. Clock transitions are ignored and the state of the outputs remains unchanged when CE is Low.

Inputs						Outputs	
R	**S**	**CE**	**J**	**K**	**C**	**Q**	**QN**
1	X	X	X	X	↑	0	1
0	1	X	X	X	↑	1	0
0	0	0	X	X	X	Q_0	QN_0
0	0	1	0	0	X	Q_0	QN_0
0	0	1	0	1	↑	0	1
0	0	1	1	1	↑	Toggle	Toggle
0	0	1	1	0	↑	1	0

FJKSRE

J-K Flip-Flop with Clock Enable and Synchronous Set and Reset



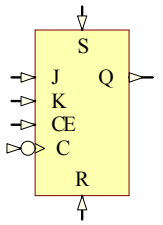
FJKSRE

FJKSRE is a single J-K-type flip-flop with J, K, synchronous set (S), synchronous reset (R), and clock enable (CE) inputs and data output (Q). When synchronous set (S) is High, all other inputs are ignored and output Q is set High. (Set has precedence over Reset.) When synchronous reset (R) is High and S is Low, output Q is reset Low. When S and R are Low and CE is High, output Q responds to the state of the J and K inputs, as shown in the following truth table, during the Low-to-High clock (C) transition. Clock transitions are ignored and the state of Q remains unchanged when CE is Low.

Inputs						Output
S	**R**	**CE**	**J**	**K**	**C**	**Q**
1	X	X	X	X	↑	1
0	1	X	X	X	↑	0
0	0	0	X	X	X	Q_0
0	0	1	0	0	X	Q_0
0	0	1	0	1	↑	0
0	0	1	1	0	↑	1
0	0	1	1	1	↑	Toggle

FJKSRE_1

J-K Negative Edge Flip-Flop with Clock Enable and Synchronous Set and Reset



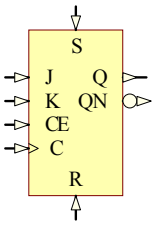
FJKSRE_1

FJKSRE_1 is a single J-K-type negative-edge triggered flip-flop with J, K, synchronous set (S), synchronous reset (R), and clock enable (CE) inputs and data output (Q). When synchronous set (S) is High, all other inputs are ignored and output Q is set High. (Set has precedence over Reset.) When synchronous reset (R) is High and S is Low, output Q is reset Low. When S and R are Low and CE is High, output Q responds to the state of the J and K inputs, as shown in the following truth table, during the High-to-Low clock (C) transition. Clock transitions are ignored and the state of Q remains unchanged when CE is Low.

Inputs						Output
S	R	CE	J	K	C	Q
1	X	X	X	X	↓	1
0	1	X	X	X	↓	0
0	0	0	X	X	X	Q_0
0	0	1	0	0	X	Q_0
0	0	1	0	1	↓	0
0	0	1	1	0	↓	1
0	0	1	1	1	↓	Toggle

FJKSREN

J-K Flip-Flop with Clock Enable, Synchronous Set and Reset and Inverted and Non-Inverted Outputs



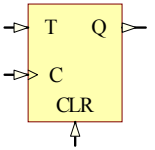
FJKSREN

FJKSREN is a single J-K-type flip-flop with J, K, synchronous set (S), synchronous reset (R), and clock enable (CE) inputs and inverted (QN) and non-inverted (Q) outputs. When synchronous set (S) is High, all other inputs are ignored and output Q is set High whilst QN is set Low. (Set has precedence over Reset.) When synchronous reset (R) is High and S is Low, output Q is reset Low whilst QN is reset High. When S and R are Low and CE is High, outputs Q and QN respond to the state of the J and K inputs, as shown in the following truth table, during the Low-to-High clock (C) transition. Clock transitions are ignored and the state of the outputs remains unchanged when CE is Low.

Inputs						Outputs	
S	**R**	**CE**	**J**	**K**	**C**	**Q**	**QN**
1	X	X	X	X	↑	1	0
0	1	X	X	X	↑	0	1
0	0	0	X	X	X	Q_0	QN_0
0	0	1	0	0	X	Q_0	QN_0
0	0	1	0	1	↑	0	1
0	0	1	1	0	↑	1	0
0	0	1	1	1	↑	Toggle	Toggle

FTC

Toggle Flip-Flop with Toggle Enable and Asynchronous Clear



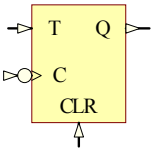
FTC

FTC is a synchronous, resettable toggle flip-flop. When High, the asynchronous clear (CLR) input overrides all other inputs and resets the data output (Q) Low. The Q output toggles, or changes state, when the toggle enable (T) input is High and CLR is Low during the Low-to-High clock transition.

Inputs			Output
CLR	**T**	**C**	**Q**
1	X	X	0
0	0	X	Q_0
0	1	↑	Toggle

FTC_1

Negative Edge Toggle Flip-Flop with Toggle Enable and Asynchronous Clear



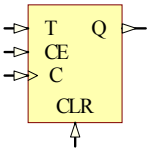
FTC_1

FTC_1 is a synchronous, resettable negative-edge toggle flip-flop. When High, the asynchronous clear (CLR) input, overrides all other inputs and resets the data output (Q) Low. The Q output toggles, or changes state, when the toggle enable (T) input is High and CLR is Low during the High-to-Low clock transition.

Inputs			Output
CLR	**T**	**C**	**Q**
1	X	X	0
0	0	X	Q_0
0	1	↓	Toggle

FTCE

Toggle Flip-Flop with Toggle and Clock Enable and Asynchronous Clear



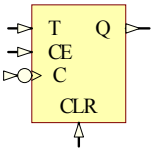
FTCE

FTCE is a toggle flip-flop with toggle and clock enable and asynchronous clear. When the asynchronous clear (CLR) input is High all other inputs are ignored and the data output (Q) is reset Low. When CLR is Low and toggle enable (T) and clock enable (CE) are High, Q output toggles, or changes state, during the Low-to-High clock (C) transition. Clock transitions are ignored and the state of Q remains unchanged when CE is Low.

Inputs				Output
CLR	CE	T	C	Q
1	X	X	X	0
0	0	X	X	Q_0
0	1	0	X	Q_0
0	1	1	↑	Toggle

FTCE_1

Negative Edge Toggle Flip-Flop with Toggle and Clock Enable and Asynchronous Clear



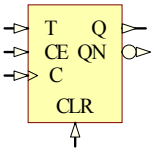
FTCE_1

FTCE_1 is a negative-edge toggle flip-flop with toggle, clock enable and asynchronous clear. When the asynchronous clear (CLR) input is High all other inputs are ignored and the data output (Q) is reset Low. When CLR is Low and toggle enable (T) and clock enable (CE) are High, Q output toggles, or changes state, during the High-to-Low clock (C) transition. Clock transitions are ignored and the state of Q remains unchanged when CE is Low.

Inputs				Output
CLR	**CE**	**T**	**C**	**Q**
1	X	X	X	0
0	0	X	X	Q_0
0	1	0	X	Q_0
0	1	1	↓	Toggle

FTCEN

Toggle Flip-Flop with Toggle, Clock Enable, Asynchronous Clear and Dual Outputs



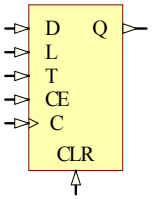
FTCEN

FTCEN is a toggle flip-flop with toggle and clock enable and asynchronous clear. When the asynchronous clear (CLR) input is High all other inputs are ignored and the data outputs Q is reset Low and QN is reset High. When CLR is Low and toggle enable (T) and clock enable (CE) are High, Q and QN outputs both toggle, or change state, during the Low-to-High clock (C) transition. Clock transitions are ignored and the state of the outputs remains unchanged when CE is Low.

Inputs				Outputs	
CLR	**CE**	**T**	**C**	**Q**	**QN**
1	X	X	X	0	1
0	0	X	X	Q_0	QN_0
0	1	0	X	Q_0	QN_0
0	1	1	↑	Toggle	Toggle

FTCLE

Toggle/Loadable Flip-Flop with Toggle and Clock Enable and Asynchronous Clear



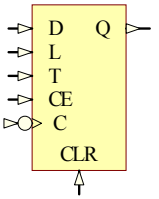
FTCLE

FTCLE is a toggle/loadable flip-flop with toggle and clock enable and asynchronous clear. When the asynchronous clear input (CLR) is High, all other inputs are ignored and output Q is reset Low. When load enable input (L) is High and CLR is Low, clock enable (CE) is overridden and the data on data input (D) is loaded into the flip-flop during the Low-to-High clock (C) transition. When toggle enable (T) and CE are High and L and CLR are Low, output Q toggles, or changes state, during the Low- to-High clock transition. Clock transitions are ignored and the state of Q remains unchanged when CE is Low.

Inputs						Outputs
CLR	L	CE	T	D	C	Q
1	X	X	X	X	X	0
0	1	X	X	1	↑	1
0	1	X	X	0	↑	0
0	0	0	X	X	X	Q_0
0	0	1	0	X	X	Q_0
0	0	1	1	X	↑	Toggle

FTCLE_1

Toggle/Loadable Negative Edge Flip-Flop with Toggle and Clock Enable and Asynchronous Clear



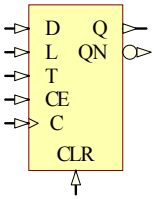
FTCLE_1

FTCLE_1 is a negative-edge toggle/loadable flip-flop with toggle and clock enable and asynchronous clear. When the asynchronous clear input (CLR) is High, all other inputs are ignored and output Q is reset Low. When load enable input (L) is High and CLR is Low, clock enable (CE) is overridden and the data on data input (D) is loaded into the flip-flop during the High-to-Low clock (C) transition. When toggle enable (T) and CE are High and L and CLR are Low, output Q toggles, or changes state, during the Low- to-High clock transition. Clock transitions are ignored and the state of Q remains unchanged when CE is Low.

Inputs						Outputs
CLR	**L**	**CE**	**T**	**D**	***C***	**Q**
1	X	X	X	X	X	0
0	1	X	X	1	↓	1
0	1	X	X	0	↓	0
0	0	0	X	X	X	Q_0
0	0	1	0	X	X	Q_0
0	0	1	1	X	↓	Toggle

FTCLEN

Toggle/Loadable Flip-Flop with Toggle, Clock Enable, Asynchronous Clear and Inverted and Non-Inverted Outputs



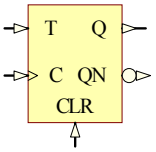
FTCLEN

FTCLEN is a toggle/loadable flip-flop with toggle and clock enable and asynchronous clear. When the asynchronous clear input (CLR) is High, all other inputs are ignored and output Q is reset Low whilst QN is reset High. When load enable input (L) is High and CLR is Low, clock enable (CE) is overridden and the data on data input (D) is loaded into the flip-flop during the Low-to-High clock (C) transition. When toggle enable (T) and CE are High and L and CLR are Low, outputs Q and QN both toggle, or changes state, during the Low-to-High clock transition. Clock transitions are ignored and the state of the outputs remains unchanged when CE is Low.

Inputs						Outputs	
CLR	**L**	**CE**	**T**	**D**	**C**	**Q**	**QN**
1	X	X	X	X	X	0	1
0	1	X	X	1	↑	1	0
0	1	X	X	0	↑	0	1
0	0	0	X	X	X	Q_0	QN_0
0	0	1	0	X	X	Q_0	QN_0
0	0	1	1	X	↑	Toggle	Toggle

FTCN

Toggle Flip-Flop with Toggle Enable, Asynchronous Clear and Inverted and Non-Inverted Outputs



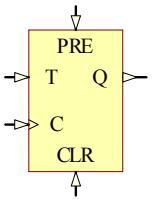
FTCN

FTCN is a synchronous, resettable toggle flip-flop. When High, the asynchronous clear (CLR) input, overrides all other inputs and resets the data outputs Q to Low and QN to High. The Q and QN outputs both toggle, or changes state, when the toggle enable (T) input is High and CLR is Low during the Low-to-High clock transition.

Inputs			Outputs	
CLR	**T**	**C**	**Q**	**QN**
1	X	X	0	1
0	0	X	Q_0	QN_0
0	1	↑	Toggle	Toggle

FTCP

Toggle Flip-Flop with Toggle Enable and Asynchronous Clear and Preset



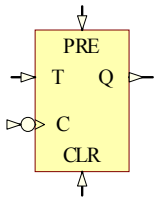
FTCP

FTCP is a toggle flip-flop with toggle enable and asynchronous clear and preset. When the asynchronous clear (CLR) input is High, all other inputs are ignored and the output (Q) is reset Low. When the asynchronous preset (PRE) is High and CLR is Low, all other inputs are ignored and Q is set High. When the toggle enable input (T) is High and CLR and PRE are Low, output Q toggles, or changes state, during the Low-to-High clock (C) transition.

Inputs				Output
CLR	**PRE**	**T**	**C**	**Q**
1	X	X	X	0
0	1	X	X	1
0	0	0	X	Q_0
0	0	1	↑	Toggle

FTCP_1

Negative Edge Toggle Flip-Flop with Toggle Enable and Asynchronous Clear and Preset



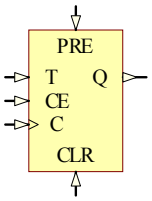
FTCP_1

FTCP_1 is a negative-edge toggle flip-flop with toggle enable and asynchronous clear and preset. When the asynchronous clear (CLR) input is High, all other inputs are ignored and the output (Q) is reset Low. When the asynchronous preset (PRE) is High and CLR is Low, all other inputs are ignored and Q is set High. When the toggle enable input (T) is High and CLR and PRE are Low, output Q toggles, or changes state, during the High-to-Low clock (C) transition.

Inputs				Output
CLR	**PRE**	**T**	**C**	**Q**
1	X	X	X	0
0	1	X	X	1
0	0	0	X	Q_0
0	0	1	↓	Toggle

FTCPE

Toggle Flip-Flop with Toggle and Clock Enable and Asynchronous Clear and Preset



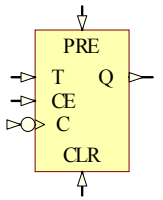
FTCPE

FTCPE is a toggle flip-flop with toggle and clock enable and asynchronous clear and preset. When the asynchronous clear (CLR) input is High, all other inputs are ignored and the output (Q) is reset Low. When the asynchronous preset (PRE) is High and CLR is Low, all other inputs are ignored and Q is set High. When the toggle enable input (T) and the clock enable input (CE) are High and CLR and PRE are Low, output Q toggles, or changes state, during the Low-to-High clock (C) transition. Clock transitions are ignored and the state of Q remains unchanged when CE is Low.

Inputs					Output
CLR	**PRE**	**CE**	**T**	**C**	**Q**
1	X	X	X	X	0
0	1	X	X	X	1
0	0	0	X	X	Q_0
0	0	1	0	X	Q_0
0	0	1	1	↑	Toggle

FTCPE_1

Negative Edge Toggle Flip-Flop with Toggle and Clock Enable and Asynchronous Clear and Preset



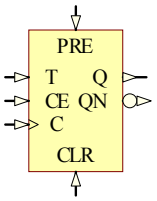
FTCPE_1

FTCPE_1 is a negative-edge toggle flip-flop with toggle and clock enable and asynchronous clear and preset. When the asynchronous clear (CLR) input is High, all other inputs are ignored and the output (Q) is reset Low. When the asynchronous preset (PRE) is High and CLR is Low, all other inputs are ignored and Q is set High. When the toggle enable input (T) and the clock enable input (CE) are High and CLR and PRE are Low, output Q toggles, or changes state, during the High-to-Low clock (C) transition. Clock transitions are ignored and the state of Q remains unchanged when CE is Low.

Inputs					Output
CLR	**PRE**	**CE**	**T**	**C**	**Q**
1	X	X	X	X	0
0	1	X	X	X	1
0	0	0	X	X	Q_0
0	0	1	0	X	Q_0
0	0	1	1	↓	Toggle

FTCPEN

Toggle Flip-Flop with Toggle, Clock Enable, Asynchronous Clear and Preset and Inverted and Non-Inverted Outputs



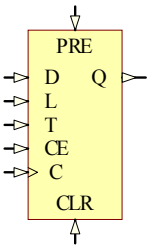
FTCPEN

FTCPEN is a toggle flip-flop with toggle and clock enable and asynchronous clear and preset. When the asynchronous clear (CLR) input is High, all other inputs are ignored and the output Q is reset Low whilst QN is reset High. When the asynchronous preset (PRE) is High and CLR is Low, all other inputs are ignored and Q is set High whilst QN is set Low. When the toggle enable input (T) and the clock enable input (CE) are High and CLR and PRE are Low, output Q and QN both toggle, or changes state, during the Low-to-High clock (C) transition. Clock transitions are ignored and the state of the outputs remains unchanged when CE is Low.

Inputs					Outputs	
CLR	**PRE**	**CE**	**T**	**C**	**Q**	**QN**
1	X	X	X	X	0	1
0	1	X	X	X	1	0
0	0	0	X	X	Q_0	QN_0
0	0	1	0	X	Q_0	QN_0
0	0	1	1	↑	Toggle	Toggle

FTCPLE

Loadable Toggle Flip-Flop with Toggle and Clock Enable and Asynchronous Clear and Preset



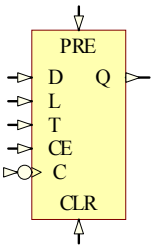
FTCPLE

FTCPLE is a loadable toggle flip-flop with toggle and clock enable and asynchronous clear and preset. When the asynchronous clear (CLR) input is High, all other inputs are ignored and the output (Q) is reset Low. When the asynchronous preset (PRE) is High and CLR is Low, all other inputs are ignored and Q is set High. When the load input (L) is High, the clock enable input (CE) is overridden and data on data input (D) is loaded into the flip-flop during the Low-to-High clock transition. When the toggle enable input (T) and the clock enable input (CE) are High and CLR, PRE, and L are Low, output Q toggles, or changes state, during the Low-to-High clock (C) transition. Clock transitions are ignored and the state of Q remains unchanged when CE is Low.

Inputs							Output
CLR	**PRE**	**L**	**CE**	**T**	**C**	**D**	**Q**
1	X	X	X	X	X	X	0
0	1	X	X	X	X	X	1
0	0	1	X	X	↑	0	0
0	0	1	X	X	↑	1	1
0	0	0	0	X	X	X	Q_0
0	0	0	1	0	X	X	Q_0
0	0	0	1	1	↑	X	Toggle

FTCPLE_1

Loadable Negative Edge Toggle Flip-Flop with Toggle and Clock Enable and Asynchronous Clear and Preset



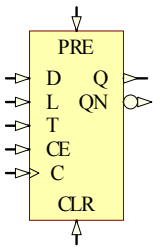
FTCPLE_1

FTCPLE_1 is a loadable negative-edge toggle flip-flop with toggle and clock enable and asynchronous clear and preset. When the asynchronous clear (CLR) input is High, all other inputs are ignored and the output (Q) is reset Low. When the asynchronous preset (PRE) is High and CLR is Low, all other inputs are ignored and Q is set High. When the load input (L) is High, the clock enable input (CE) is overridden and data on data input (D) is loaded into the flip-flop during the High-to-Low clock transition. When the toggle enable input (T) and the clock enable input (CE) are High and CLR, PRE, and L are Low, output Q toggles, or changes state, during the High-to-Low clock (C) transition. Clock transitions are ignored and the state of Q remains unchanged when CE is Low.

Inputs							Output
CLR	**PRE**	**L**	**CE**	**T**	**C**	**D**	**Q**
1	X	X	X	X	X	X	0
0	1	X	X	X	X	X	1
0	0	1	X	X	↓	0	0
0	0	1	X	X	↓	1	1
0	0	0	0	X	X	X	Q_0
0	0	0	1	0	X	X	Q_0
0	0	0	1	1	↓	X	Toggle

FTCPLEN

Loadable Toggle Flip-Flop with Toggle, Clock Enable, Asynchronous Clear and Preset and Inverted and Non-Inverted Outputs



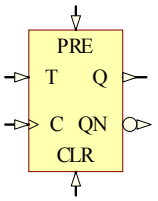
FTCPLEN

FTCPLEN is a loadable toggle flip-flop with toggle and clock enable and asynchronous clear and preset. When the asynchronous clear (CLR) input is High, all other inputs are ignored and output Q is reset Low whilst QN is reset High. When the asynchronous preset (PRE) is High and CLR is Low, all other inputs are ignored and Q is set High whilst QN is set Low. When the load input (L) is High, the clock enable input (CE) is overridden and data on data input (D) is loaded into the flip-flop during the Low-to-High clock transition. When the toggle enable input (T) and the clock enable input (CE) are High and CLR, PRE, and L are Low, outputs Q and QN both toggle, or changes state, during the Low-to-High clock (C) transition. Clock transitions are ignored and the state of the outputs remains unchanged when CE is Low.

Inputs							Outputs	
CLR	**PRE**	**L**	**CE**	**T**	**C**	**D**	**Q**	**QN**
1	X	X	X	X	X	X	0	1
0	1	X	X	X	X	X	1	0
0	0	1	X	X	↑	0	0	1
0	0	1	X	X	↑	1	1	0
0	0	0	0	X	X	X	Q_0	QN_0
0	0	0	1	0	X	X	Q_0	QN_0
0	0	0	1	1	↑	X	Toggle	Toggle

FTCPN

Toggle Flip-Flop with Toggle Enable, Asynchronous Clear, Preset and Inverted and Non-Inverted Outputs



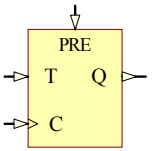
FTCPN

FTCPN is a toggle flip-flop with toggle enable and asynchronous clear and preset. When the asynchronous clear (CLR) input is High, all other inputs are ignored and the output Q is reset Low whilst QN is reset High. When the asynchronous preset (PRE) is High and CLR is Low, all other inputs are ignored and Q is set High whilst QN is set Low. When the toggle enable input (T) is High and CLR and PRE are Low, outputs Q and QN both toggle, or changes state, during the Low-to-High clock (C) transition.

Inputs				Outputs	
CLR	**PRE**	**T**	**C**	**Q**	**QN**
1	X	X	X	0	1
0	1	X	X	1	0
0	0	0	X	Q_0	QN_0
0	0	1	↑	Toggle	Toggle

FTP

Toggle Flip-Flop with Toggle Enable and Asynchronous Preset



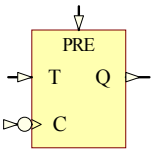
FTP

FTP is a toggle flip-flop with toggle enable and asynchronous preset. When the asynchronous preset (PRE) input is High, all other inputs are ignored and output Q is set High. When toggle-enable input (T) is High and PRE is Low, output Q toggles, or changes state, during the Low-to-High clock (C) transition.

Inputs			Output
PRE	**T**	**C**	**Q**
1	X	X	1
0	0	X	Q_0
0	1	↑	Toggle

FTP_1

Negative Edge Toggle Flip-Flop with Toggle Enable and Asynchronous Preset



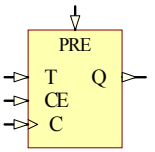
FTP_1

FTP_1 is a negative-edge toggle flip-flop with toggle enable and asynchronous preset. When the asynchronous preset (PRE) input is High, all other inputs are ignored and output Q is set High. When toggle-enable input (T) is High and PRE is Low, output Q toggles, or changes state, during the High-to-Low clock (C) transition.

Inputs			Output
PRE	**T**	**C**	**Q**
1	X	X	1
0	0	X	Q_0
0	1	↓	Toggle

FTPE

Toggle Flip-Flop with Toggle and Clock Enable and Asynchronous Preset



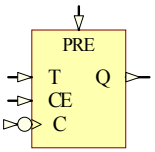
FTPE

FTPE is a toggle flip-flop with toggle and clock enable and asynchronous preset. When the asynchronous preset (PRE) input is High, all other inputs are ignored and output Q is set High. When the toggle enable input (T) is High, clock enable (CE) is High, and PRE is Low, output Q toggles, or changes state, during the Low-to-High clock transition. Clock transitions are ignored and the state of Q remains unchanged when CE is Low.

Inputs				Output
PRE	**CE**	**T**	**C**	**Q**
1	X	X	X	1
0	0	X	X	Q_0
0	1	0	X	Q_0
0	1	1	↑	Toggle

FTPE_1

Negative Edge Toggle Flip-Flop with Toggle and Clock Enable and Asynchronous Preset



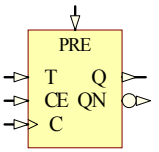
FTPE_1

FTPE_1 is a negative-edge toggle flip-flop with toggle and clock enable and asynchronous preset. When the asynchronous preset (PRE) input is High, all other inputs are ignored and output Q is set High. When the toggle enable input (T) is High, clock enable (CE) is High, and PRE is Low, output Q toggles, or changes state, during the High-to-Low clock transition. Clock transitions are ignored and the state of Q remains unchanged when CE is Low.

Inputs				Output
PRE	**CE**	**T**	**C**	**Q**
1	X	X	X	1
0	0	X	X	Q_0
0	1	0	X	Q_0
0	1	1	↓	Toggle

FTPEN

Toggle Flip-Flop with Toggle, Clock Enable, Asynchronous Preset and Inverted and Non-Inverted Outputs



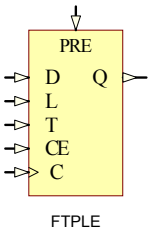
FTPEN

FTPEN is a toggle flip-flop with toggle and clock enable and asynchronous preset. When the asynchronous preset (PRE) input is High, all other inputs are ignored and output Q is set High whilst QN is set Low. When the toggle enable input (T) is High, clock enable (CE) is High, and PRE is Low, outputs Q and QN toggle, or changes state, during the Low-to-High clock transition. Clock transitions are ignored and the state of the outputs remains unchanged when CE is Low.

Inputs				Outputs	
PRE	**CE**	**T**	**C**	**Q**	**QN**
1	X	X	X	1	0
0	0	X	X	Q_0	QN_0
0	1	0	X	Q_0	QN_0
0	1	1	↑	Toggle	Toggle

FTPLE

Toggle/Loadable Flip-Flop with Toggle and Clock Enable and Asynchronous Preset



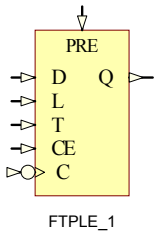
FTPLE

FTPLE is a toggle/loadable flip-flop with toggle and clock enable and asynchronous preset. When the asynchronous preset input (PRE) is High, all other inputs are ignored and output Q is set High. When the load enable input (L) is High and PRE is Low, the clock enable (CE) is overridden and the data (D) is loaded into the flip-flop during the Low-to-High clock transition. When L and PRE are Low and toggle-enable input (T) and CE are High, output Q toggles, or changes state, during the Low-to-High clock transition. Clock transitions are ignored and the state of Q remains unchanged when CE is Low.

Inputs						Output
PRE	**L**	**CE**	**T**	**D**	**C**	**Q**
1	X	X	X	X	X	1
0	1	X	X	1	↑	1
0	1	X	X	0	↑	0
0	0	0	X	X	X	Q_0
0	0	1	0	X	X	Q_0
0	0	1	1	X	↑	Toggle

FTPLE_1

Toggle/Loadable Negative Edge Flip-Flop with Toggle and Clock Enable and Asynchronous Preset



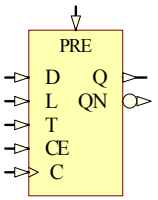
FTPLE_1

FTPLE_1 is a negative-edge toggle/loadable flip-flop with toggle and clock enable and asynchronous preset. When the asynchronous preset input (PRE) is High, all other inputs are ignored and output Q is set High. When the load enable input (L) is High and PRE is Low, the clock enable (CE) is overridden and the data (D) is loaded into the flip-flop during the High-to-Low clock transition. When L and PRE are Low and toggle-enable input (T) and CE are High, output Q toggles, or changes state, during the High-to-Low clock transition. Clock transitions are ignored and the state of Q remains unchanged when CE is Low.

Inputs						Output
PRE	L	CE	T	D	C	Q
1	X	X	X	X	X	1
0	1	X	X	1	↓	1
0	1	X	X	0	↓	0
0	0	0	X	X	X	Q_0
0	0	1	0	X	X	Q_0
0	0	1	1	X	↓	Toggle

FTPLEN

Toggle/Loadable Flip-Flop with Toggle, Clock Enable, Asynchronous Preset and Inverted and Non-inverted Outputs



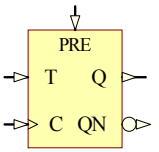
FTPLEN

FTPLEN is a toggle/loadable flip-flop with toggle and clock enable and asynchronous preset. When the asynchronous preset input (PRE) is High, all other inputs are ignored and output Q is set High whilst QN is set Low. When the load enable input (L) is High and PRE is Low, the clock enable (CE) is overridden and the data (D) is loaded into the flip-flop during the Low-to-High clock transition. When L and PRE are Low and toggle-enable input (T) and CE are High, outputs Q and QN both toggle, or changes state, during the Low-to-High clock transition. Clock transitions are ignored and the state of the outputs remains unchanged when CE is Low.

Inputs						Outputs	
PRE	**L**	**CE**	**T**	**D**	**C**	**Q**	**QN**
1	X	X	X	X	X	1	0
0	1	X	X	1	↑	1	0
0	1	X	X	0	↑	0	1
0	0	0	X	X	X	Q_0	QN_0
0	0	1	0	X	X	Q_0	QN_0
0	0	1	1	X	↑	Toggle	Toggle

FTPN

Toggle Flip-Flop with Toggle Enable, Asynchronous Preset and Inverted and Non-Inverted Outputs



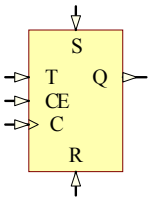
FTPN

FTPN is a toggle flip-flop with toggle enable and asynchronous preset. When the asynchronous preset (PRE) input is High, all other inputs are ignored and output Q is set High whilst QN is set Low. When toggle-enable input (T) is High and PRE is Low, outputs Q and QN both toggle, or changes state, during the Low-to-High clock (C) transition.

Inputs			Outputs	
PRE	**T**	**C**	**Q**	**QN**
1	X	X	1	0
0	0	X	Q_0	QN_0
0	1	↑	Toggle	Toggle

FTRSE

Toggle Flip-Flop with Toggle and Clock Enable and Synchronous Reset and Set



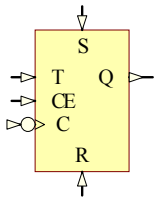
FTRSE

FTRSE is a toggle flip-flop with toggle and clock enable and synchronous reset and set. When the synchronous reset input (R) is High, it overrides all other inputs and the data output (Q) is reset Low. When the synchronous set input (S) is High and R is Low, clock enable input (CE) is overridden and output Q is set High. (Reset has precedence over Set.) When toggle enable input (T) and CE are High and R and S are Low, output Q toggles, or changes state, during the Low-to-High clock transition.

Inputs					Output
R	**S**	**CE**	**T**	**C**	**Q**
1	X	X	X	↑	0
0	1	X	X	↑	1
0	0	0	X	X	Q_0
0	0	1	0	X	Q_0
0	0	1	1	↑	Toggle

FTRSE_1

Negative Edge Toggle Flip-Flop with Toggle and Clock Enable and Synchronous Reset and Set



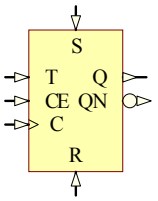
FTRSE_1

FTRSE_1 is a negative-edge toggle flip-flop with toggle and clock enable and synchronous reset and set. When the synchronous reset input (R) is High, it overrides all other inputs and the data output (Q) is reset Low. When the synchronous set input (S) is High and R is Low, clock enable input (CE) is overridden and output Q is set High. (Reset has precedence over Set.) When toggle enable input (T) and CE are High and R and S are Low, output Q toggles, or changes state, during the High-to-Low clock transition.

Inputs					Output
R	**S**	**CE**	**T**	**C**	**Q**
1	X	X	X	↓	0
0	1	X	X	↓	1
0	0	0	X	X	Q_0
0	0	1	0	X	Q_0
0	0	1	1	↓	Toggle

FTRSEN

Toggle Flip-Flop with Toggle, Clock Enable, Synchronous Reset and Set and Inverted and Non-Inverted Outputs



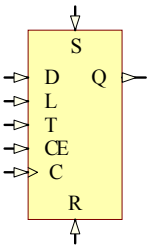
FTRSEN

FTRSEN is a toggle flip-flop with toggle and clock enable and synchronous reset and set. When the synchronous reset input (R) is High, it overrides all other inputs and the data output Q is reset Low whilst QN is reset High. When the synchronous set input (S) is High and R is Low, clock enable input (CE) is overridden and output Q is set High whilst QN is set Low. (Reset has precedence over Set.) When toggle enable input (T) and CE are High and R and S are Low, outputs Q and QN both toggle, or changes state, during the Low-to-High clock transition.

Inputs					Outputs	
R	**S**	**CE**	**T**	**C**	**Q**	**QN**
1	X	X	X	↑	0	1
0	1	X	X	↑	1	0
0	0	0	X	X	Q_0	QN_0
0	0	1	0	X	Q_0	QN_0
0	0	1	1	↑	Toggle	Toggle

FTRSLE

Toggle/Loadable Flip-Flop with Toggle and Clock Enable and Synchronous Reset and Set



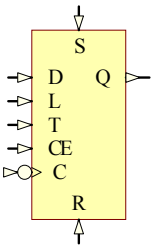
FTRSLE

FTRSLE is a toggle/loadable flip-flop with toggle and clock enable and synchronous reset and set. When High the synchronous reset input (R) overrides all other inputs and resets the data output (Q) Low. (Reset has precedence over Set.) When R is Low and synchronous set input (S) is High, the clock enable input (CE) is overridden and output Q is set High. When R and S are Low and load enable input (L) is High, CE is overridden and data on data input (D) is loaded into the flip-flop during the Low-to-High clock transition. When R, S, and L are Low, CE is High and T is High, output Q toggles, or changes state, during the Low-to-High clock transition. Clock transitions are ignored and the state of Q remains unchanged when CE is Low.

Inputs							Output
R	**S**	**L**	**CE**	**T**	**D**	**C**	**Q**
1	0	X	X	X	X	↑	0
0	1	X	X	X	X	↑	1
0	0	1	X	X	1	↑	1
0	0	1	X	X	0	↑	0
0	0	0	0	X	X	X	Q_0
0	0	0	1	0	X	X	Q_0
0	0	0	1	1	X	↑	Toggle

FTRSLE_1

Toggle/Loadable Negative Edge Flip-Flop with Toggle and Clock Enable and Synchronous Reset and Set



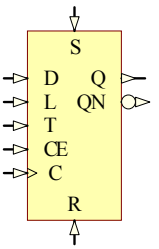
FTRSLE_1

FTRSLE_1 is a negative-edge toggle/loadable flip-flop with toggle and clock enable and synchronous reset and set. When High the synchronous reset input (R) overrides all other inputs and resets the data output (Q) Low. (Reset has precedence over Set.) When R is Low and synchronous set input (S) is High, the clock enable input (CE) is overridden and output Q is set High. When R and S are Low and load enable input (L) is High, CE is overridden and data on data input (D) is loaded into the flip-flop during the High-to-Low clock transition. When R, S, and L are Low, CE is High and T is High, output Q toggles, or changes state, during the High-to-Low clock transition. Clock transitions are ignored and the state of Q remains unchanged when CE is Low.

Inputs							Outputs
R	**S**	**L**	**CE**	**T**	**D**	**C**	**Q**
1	0	X	X	X	X	↓	0
0	1	X	X	X	X	↓	1
0	0	1	X	X	1	↓	1
0	0	1	X	X	0	↓	0
0	0	0	0	X	X	X	Q_0
0	0	0	1	0	X	X	Q_0
0	0	0	1	1	X	↓	Toggle

FTRSLEN

Toggle/Loadable Flip-Flop with Toggle, Clock Enable, Synchronous Reset and Set and Inverted and Non-Inverted Outputs



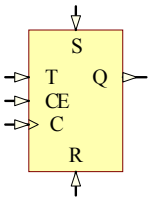
FTRSLEN

FTRSLEN is a toggle/loadable flip-flop with toggle and clock enable and synchronous reset and set. When High the synchronous reset input (R) overrides all other inputs and resets the data output Q to Low and Q to High. (Reset has precedence over Set.) When R is Low and synchronous set input (S) is High, the clock enable input (CE) is overridden and output Q is set High whilst QN is set Low. When R and S are Low and load enable input (L) is High, CE is overridden and data on data input (D) is loaded into the flip-flop during the Low-to-High clock transition. When R, S, and L are Low, CE is High and T is High, outputs Q and QN both toggle, or changes state, during the Low-to-High clock transition. Clock transitions are ignored and the state of the outputs remains unchanged when CE is Low.

Inputs							Outputs	
R	**S**	**L**	**CE**	**T**	**D**	**C**	**Q**	**QN**
1	0	X	X	X	X	↑	0	1
0	1	X	X	X	X	↑	1	0
0	0	1	X	X	1	↑	1	0
0	0	1	X	X	0	↑	0	1
0	0	0	0	X	X	X	Q_0	QN_0
0	0	0	1	0	X	X	Q_0	QN_0
0	0	0	1	1	X	↑	Toggle	Toggle

FTSRE

Toggle Flip-Flop with Toggle and Clock Enable and Synchronous Set and Reset



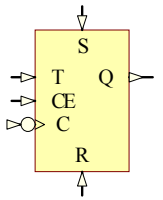
FTSRE

FTSRE is a toggle flip-flop with toggle and clock enable and synchronous set and reset. When High, the synchronous set input overrides all other inputs and sets data output (Q) High. (Set has precedence over Reset.) When synchronous reset input (R) is High and S is Low, clock enable input (CE) is overridden and output Q is reset Low. When toggle enable input (T) and CE are High and S and R are Low, output Q toggles, or changes state, during the Low-to-High clock transition. Clock transitions are ignored and the state of Q remains unchanged when CE is Low.

Inputs					Output
S	**R**	**CE**	**T**	**C**	**Q**
1	X	X	X	↑	1
0	1	X	X	↑	0
0	0	0	X	X	Q_0
0	0	1	0	X	Q_0
0	0	1	1	↑	Toggle

FTSRE_1

Negative Edge Toggle Flip-Flop with Toggle and Clock Enable and Synchronous Set and Reset



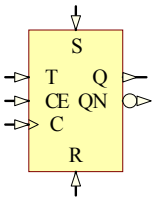
FTSRE_1

FTSRE_1 is a negative-edge toggle flip-flop with toggle and clock enable and synchronous set and reset. When High, the synchronous set input overrides all other inputs and sets data output (Q) High. (Set has precedence over Reset.) When synchronous reset input (R) is High and S is Low, clock enable input (CE) is overridden and output Q is reset Low. When toggle enable input (T) and CE are High and S and R are Low, output Q toggles, or changes state, during the High-to-Low clock transition. Clock transitions are ignored and the state of Q remains unchanged when CE is Low.

Inputs					Output
S	**R**	**CE**	**T**	**C**	**Q**
1	X	X	X	↓	1
0	1	X	X	↓	0
0	0	0	X	X	Q_0
0	0	1	0	X	Q_0
0	0	1	1	↓	Toggle

FTSREN

Toggle Flip-Flop with Toggle, Clock Enable, Synchronous Set and Reset and Inverted and Non-Inverted Outputs



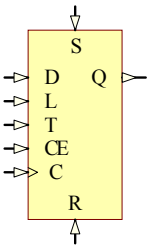
FTSREN

FTSREN is a toggle flip-flop with toggle and clock enable and synchronous set and reset. When High the synchronous set input overrides all other inputs and sets data outputs Q to High and QN to low. (Set has precedence over Reset.) When synchronous reset input (R) is High and S is Low, clock enable input (CE) is overridden and output Q is reset Low whilst QN is reset High. When toggle enable input (T) and CE are High and S and R are Low, outputs Q and QN both toggle, or changes state, during the Low-to-High clock transition. Clock transitions are ignored and the state of the outputs remains unchanged when CE is Low.

Inputs					Outputs	
S	**R**	**CE**	**T**	**C**	**Q**	**QN**
1	X	X	X	↑	1	0
0	1	X	X	↑	0	1
0	0	0	X	X	Q_0	QN_0
0	0	1	0	X	Q_0	QN_0
0	0	1	1	↑	Toggle	Toggle

FTSRLE

Toggle/Loadable Flip-Flop with Toggle and Clock Enable and Synchronous Set and Reset



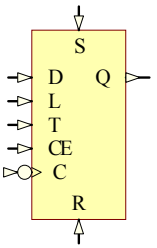
FTSRLE

FTSRLE is a toggle/loadable flip-flop with toggle and clock enable and synchronous set and reset. When High, the synchronous set input (S) overrides all other inputs and sets data output (Q) High. (Set has precedence over Reset.) When synchronous reset (R) is High and S is Low, clock enable input (CE) is overridden and output Q is reset Low. When load enable input (L) is High and S and R are Low, CE is overridden and data on data input (D) is loaded into the flip-flop during the Low-to-High clock transition. When the toggle enable input (T) and CE are High and S, R, and L are Low, output Q toggles, or changes state, during the Low-to- High clock transition. Clock transitions are ignored and the state of Q remains unchanged when CE is Low.

Inputs							Output
S	**R**	**L**	**CE**	**T**	**D**	**C**	**Q**
1	X	X	X	X	X	↑	1
0	1	X	X	X	X	↑	0
0	0	1	X	X	1	↑	1
0	0	1	X	X	0	↑	0
0	0	0	0	X	X	X	Q_0
0	0	0	1	0	X	X	Q_0
0	0	0	1	1	X	↑	Toggle

FTSRLE_1

Toggle/Loadable Negative Edge Flip-Flop with Toggle and Clock Enable and Synchronous Set and Reset



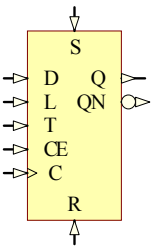
FTSRLE_1

FTSRLE_1 is a negative-edge toggle/loadable flip-flop with toggle and clock enable and synchronous set and reset. When High, the synchronous set input (S) overrides all other inputs and sets data output (Q) High. (Set has precedence over Reset.) When synchronous reset (R) is High and S is Low, clock enable input (CE) is overridden and output Q is reset Low. When load enable input (L) is High and S and R are Low, CE is overridden and data on data input (D) is loaded into the flip-flop during the High-to-Low clock transition. When the toggle enable input (T) and CE are High and S, R, and L are Low, output Q toggles, or changes state, during the High-to-Low clock transition. Clock transitions are ignored and the state of Q remains unchanged when CE is Low.

Inputs							Output
S	**R**	**L**	**CE**	**T**	**D**	**C**	**Q**
1	X	X	X	X	X	↓	1
0	1	X	X	X	X	↓	0
0	0	1	X	X	1	↓	1
0	0	1	X	X	0	↓	0
0	0	0	0	X	X	X	Q_0
0	0	0	1	0	X	X	Q_0
0	0	0	1	1	X	↓	Toggle

FTSRLEN

Toggle/Loadable Flip-Flop with Toggle, Clock Enable, Synchronous Set and Reset and Inverted and Non-Inverted Outputs



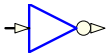
FTSRLEN

FTSRLEN is a toggle/loadable flip-flop with toggle and clock enable and synchronous set and reset. When High, the synchronous set input (S) overrides all other inputs and sets data output Q to High whilst QN is set Low. (Set has precedence over Reset.) When synchronous reset (R) is High and S is Low, clock enable input (CE) is overridden and output Q is reset Low whilst Qn is reset High. When load enable input (L) is High and S and R are Low, CE is overridden and data on data input (D) is loaded into the flip-flop during the Low-to-High clock transition. When the toggle enable input (T) and CE are High and S, R, and L are Low, outputs Q and QN both toggle, or changes state, during the Low-to- High clock transition. Clock transitions are ignored and the state of the outputs remains unchanged when CE is Low.

Inputs							Outputs	
S	**R**	**L**	**CE**	**T**	**D**	**C**	**Q**	**QN**
1	X	X	X	X	X	↑	1	0
0	1	X	X	X	X	↑	0	1
0	0	1	X	X	1	↑	1	0
0	0	1	X	X	0	↑	0	1
0	0	0	0	X	X	X	Q_0	QN_0
0	0	0	1	0	X	X	Q_0	QN_0
0	0	0	1	1	X	↑	Toggle	Toggle

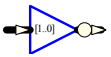
INV – INV32

Inverters

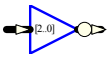


INV

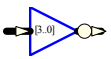
These are 1-, 2-, 3-, 4-, 5-, 6-, 7-, 8-, 9-, 10-, 12-, 16-, 32-bit inverters. They invert all input signals (I) to the outputs (O).



INV2B



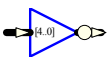
INV3B



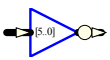
INV4B

Inputs			Outputs		
I0	**…**	**I*n-1***	**O0**	**…**	**On-1**
0	0	0	1	1	1
0	0	1	1	1	0
0	1	1	1	0	0
1	1	1	0	0	0

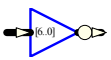
n = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16



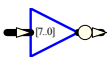
INV5B



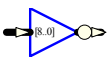
INV6B



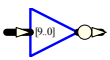
INV7B



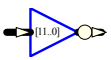
INV8B



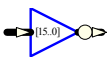
INV9B



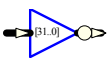
INV10B



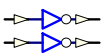
INV12B



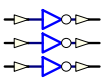
INV16B



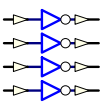
INV32B



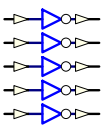
INV2S



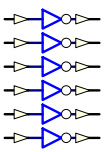
INV3S



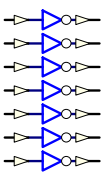
INV4S



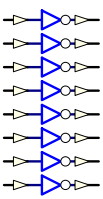
INV5S



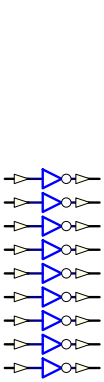
INV6S



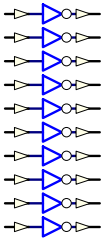
INV7S



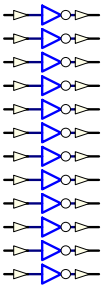
INV8S



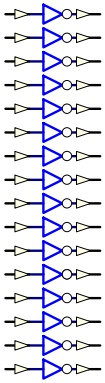
INV9S



INV10S



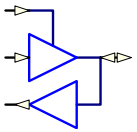
INV12S



INV16S

IOBUF – IOBUF32

Input/Output Buffer



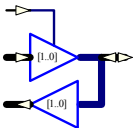
IOBUF

IOBUF, IOBUF 2, IOBUF 3, IOBUF4, IOBUF5, IOBUF6, IOBUF7, IOBUF8, IOBUF9, IOBUF10, IOBUF12, IOBUF16, IOBUF32 are respectively 1-, 2-, 3-, 4, 5-, 6-, 7-, 8-, 9-, 10-, 12-, 16-, 32-bit input/output buffers.

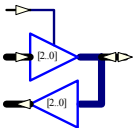
When tri-state control input (T) is High, inputs (I) are ignored and all IO pins are set to High-Impedance state. At this condition a valid input signal (High or Low) can be driven at the IO pins and transferred to the output pins (O).

When tri-state control input (T) is Low, all inputs (I) data is transferred to the IO pins and the output pins (O).

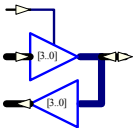
All IOBUFs are control by a common control pin T.



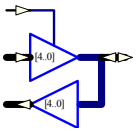
IOBUF2B



IOBUF3B



IOBUF4B



IOBUF5B

IOBUF

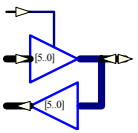
Inputs		In/Out	Output
T	**I**	**IO**	**O**
1	X	Z	X
0	1	1	1
0	0	0	0

IOBUF2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16

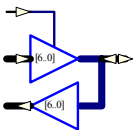
Inputs		In/Out	Output
T	**I(n-1..0)**	**IO(n-1..0)**	**O(n-1..0)**
1	X	Z	X
0	i	i	i

n is the length of data bus, available in 2,3,4,5,6,7,8,9,10,12,16, 32-bits

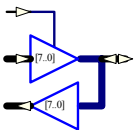
i is the value of inputs bus I



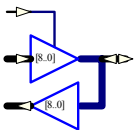
IOBUF6B



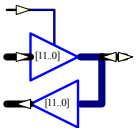
IOBUF7B



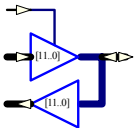
IOBUF8B



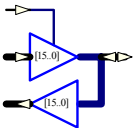
IOBUF9B



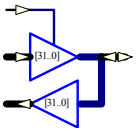
IOBUF10B



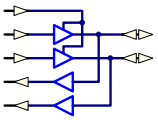
IOBUF12B



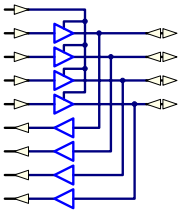
IOBUF16B



IOBUF32B



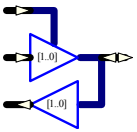
IOBUF2S



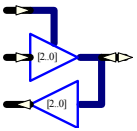
IOBUF4S

IOBUFC2 – IOBUFC32

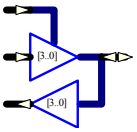
Input/Output Buffer with Separated Control



IOBUFC2B



IOBUFC3B



IOBUFC4B

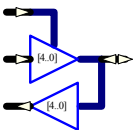
IOBUFC 2, IOBUFC 3, IOBUFC4, IOBUFC 5, IOBUFC 6, IOBUFC 7, IOBUFC8, IOBUFC 9, IOBUFC 10, IOBUFC12, IOBUFC16, IOBUFC32 are respectively, group of 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, and 32 single-bit IOBUF collections.

As they are independent IOBUF, I, IO and O pins of each group are controlled by a separated control pin T and behave like an IOBUF.

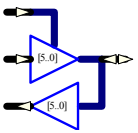
IOBUFCn

Inputs						In/Out			Outputs		
Tn-1	**…**	**T0**	**In-1**	**…**	**I0**	**IOn-1**	**…**	**IO0**	**On-1**	**…**	**O0**
0	0	0	a	b	c	a	b	c	a	b	c
0	0	1	a	b	X	a	b	Z	a	b	X
0	1	0	a	X	c	a	Z	c	a	X	c
0	1	1	a	X	X	a	Z	Z	a	X	X
1	0	0	X	b	c	Z	b	c	X	b	c
1	0	1	X	b	X	Z	b	Z	X	b	X
1	1	0	X	X	c	Z	Z	c	X	X	c
1	1	1	X	X	X	Z	Z	Z	X	X	X

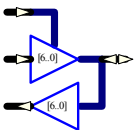
n is the length of data bus, available in 2,3,4,5,6,7,8,9,10,12,16, 32-bits



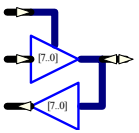
IOBUFC5B



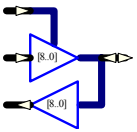
IOBUFC6B



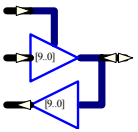
IOBUFC7B



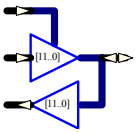
IOBUFC8B



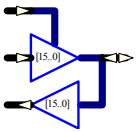
IOBUFC9B



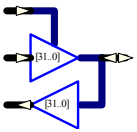
IOBUFC10B



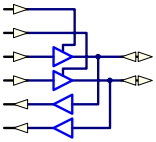
IOBUFC12B



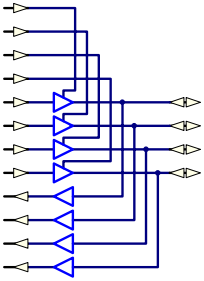
IOBUFC16B



IOBUFC32B



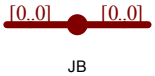
IOBUFC2S



IOBUFC4S

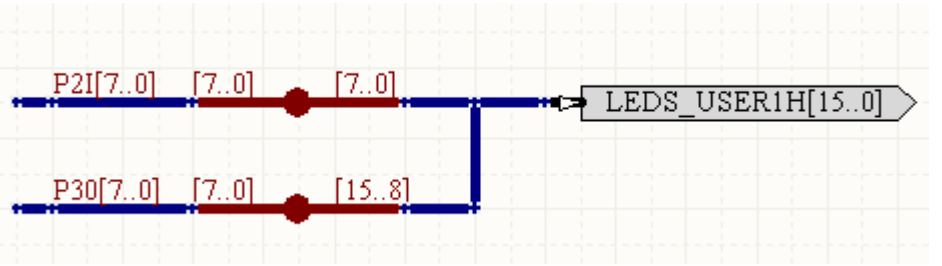
JB

System BUS Joiner



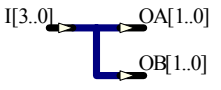
This component allows mapping between two unique buses by using component parameters and pin properties. The parameters IndexA and IndexB are used to define the mapping slice range of both pin sides respectively. The electrical type of the pins can be changed to meet any combination requirements. The default values of IndexA and IndexB is set to [0..0] respectively.

The example below illustrates the mapping of two unique buses (P2I and P3O) onto a single output bus (LEDS_USER1H) by using two instantiations of the JB component.

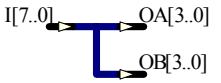


J*m*B_*n*B*p*

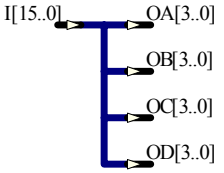
1 x *m*-Bit Input Bus to *p* x *n*-Bit Output Bus



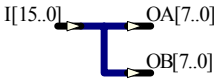
J4B_2B2



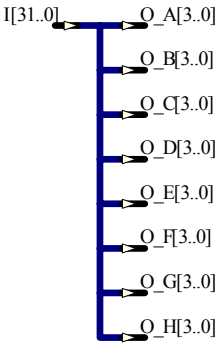
J8B_4B2



J16B_4B4



J16B_8B2



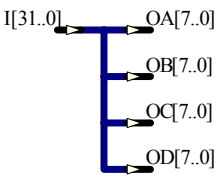
J32B_4B8

This component takes one *m*-bit input bus and maps to *p, n*-bit output buses. The following components are available:

Component	Function
J4B_2B2	1 x 4-bit input bus to 2 x 2-Bit output bus
J8B_4B2	1 x 8-bit input bus to 2 x 4-Bit output bus
J16B_4B4	1 x 16-bit input bus to 4 x 4-Bit output bus
J16B_8B2	1 x 16-bit input bus to 2 x 8-Bit output bus
J32B_4B8	1 x 32-Bit input bus to 8 x 4-Bit output bus
J32B_8B4	1 x 32-Bit input bus to 4 x 8-Bit output bus
J32B_16B2	1 x 32-bit input bus to 2 x 16-Bit output bus

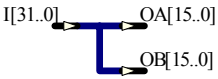
The following table shows how the input pins are mapped to the output pins:

Component	Input → Output
J4B_2B2	I[1..0] → OA[1..0]
	I[3..2] → OB[1..0]
J8B_4B2	I[3..0] → OA[3..0]
	I[7..4] → OB[3..0]
J16B_8B2	I[7..0] → OA[7..0]
	I[15..8] → OB[7..0]
J32B_16B2	I[15..0] → OA[15..0]
	I[31..16] → OB[15..0]
J32B_8B4	I[7..0] → OA[7..0]
	I[15..8] → OB[7..0]
	I[23..16] → OC[7..0]
	I[31..24] → OD[7..0]
J16B_4B4	I[3..0] → OA[3..0]
	I[7..4] → OB[3..0]
	I[11..8] → OC[3..0]
	I[15..12] → OD[3..0]



J32B_8B4

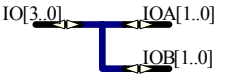
Component	Input → Output
J32B_4B8	I[3..0] → OA[3..0]
	I[7..4] → OB[3..0]
	I[11..8] → OC[3..0]
	I[15..12] → OD[3..0]
	I[19..16] → OE[3..0]
	I[23..20] → OF[3..0]



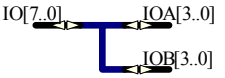
J32B_16B2

J*m*B_*n*B*p*X

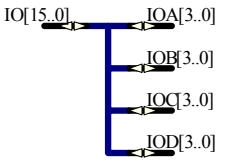
1 x *m*-Bit IO Bus to *p* x *n*-Bit IO Bus



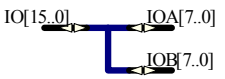
J4B_2B2X



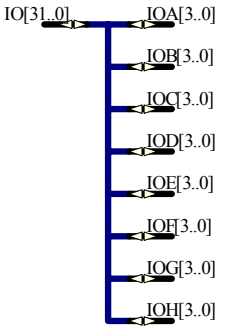
J8B_4B2X



J16B_4B4X



J16B_8B2X



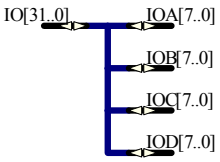
J32B_4B8X

This component takes a single *m*-bit I/O bus and maps to *p*, *n*-bit I/O buses. The following components are available:

Component	Function
J4B_2B2X	1 x 4-bit IO bus to 2 x 2-Bit IO bus
J8B_4B2X	1 x 8-bit IO bus to 2 x 4-Bit IO bus
J16B_4B4X	1 x 16-bit IO bus to 4 x 4-Bit IO bus
J16B_8B2X	1 x 16-bit IO bus to 2 x 8-Bit IO bus
J32B_4B8X	1 x 32-Bit IO bus to 8 x 4-Bit IO bus
J32B_8B4X	1 x 32-Bit IO bus to 4 x 8-Bit IO bus
J32B_16B2X	1 x 32-bit IO bus to 2 x 16-Bit IO bus

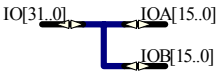
The following table shows how the input pins are mapped to the output pins:

Component	Input → IOutput
J4B_2B2X	IO[1..0] → IOA[1..0]
	IO[3..2] → IOB[1..0]
J8B_4B2X	IO[3..0] → IOA[3..0]
	IO[7..4] → IOB[3..0]
J16B_8B2X	IO[7..0] → IOA[7..0]
	IO[15..8] → IOB[7..0]
J32B_16B2X	IO[15..0] → IOA[15..0]
	IO[31..16] → IOB[15..0]
J32B_8B4X	IO[7..0] → IOA[7..0]
	IO[15..8] → IOB[7..0]
	IO[23..16] → IOC[7..0]
	IO[31..24] → IOD[7..0]
J16B_4B4X	IO[3..0] → IOA[3..0]
	IO[7..4] → IOB[3..0]
	IO[11..8] → IOC[3..0]
	IO[15..12] → IOD[3..0]



J32B_8B4X

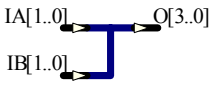
Component	Input → IOutput
J32B_4B8X	IO[3..0] → IOA[3..0]
	IO[7..4] → IOB[3..0]
	IO[11..8] → IOC[3..0]
	IO[15..12] → IOD[3..0]
	IO[19..16] → IOE[3..0]
	IO[23..20] → IOF[3..0]
	IO[27..24] → IOG[3..0]
	IO[31..28] → IOH[3..0]



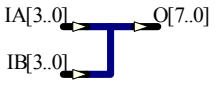
J32B_16B2X

JmBn_pB

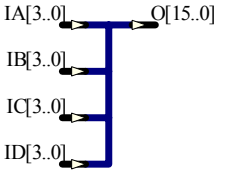
n x *m*-Bit Input Bus to 1 x *p*-Bit Output Bus



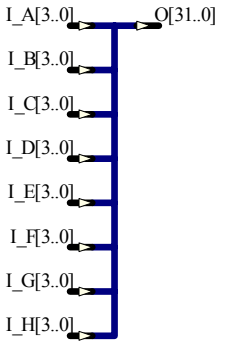
J2B2_4B



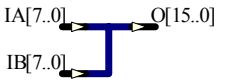
J4B2_8B



J4B4_16B



J4B8_32B



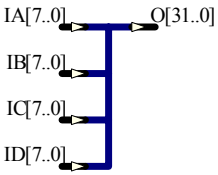
J8B2_16B

This component takes *n*, *m*-bit input buses and maps to one single *p*-bit output bus. The following components are available:

Component	Function
J2B2_4B	2 x 2-Bit input bus to 1 x 4-bit output bus
J4B2_8B	2 x 4-Bit input bus to 1 x 8-bit output bus
J4B4_16B	4 x 4-Bit input bus to 1 x 16-bit output bus
J4B8_32B	8 x 4-Bit input bus to 1 x 32-bit output bus
J8B2_16B	2 x 8-Bit input bus to 1 x 16-bit output bus
J8B4_32B	4 x 8-Bit input bus to 1 x 32-bit output bus
J16B2_32B	2 x 16-Bit input bus to 1 x 32-bit output bus

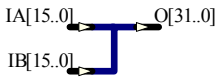
The following table shows how the input pins are mapped to the output pins:

Component	Input → Output
J2B2_4B	IA[1..0] → O[1..0]
	IB[1..0] → O[3..2]
J4B2_8B	IA[3..0] → O[3..0]
	IB[3..0] → O[7..4]
J4B4_16B	IA[3..0] → O[3..0]
	IB[3..0] → O[7..4]
	IC[3..0] → O[11..8]
	ID[3..0] → O[15..12]
J4B8_32B	IA[3..0] → O[3..0]
	IB[3..0] → O[7..4]
	IC[3..0] → O[11..8]
	ID[3..0] → O[15..12]
	IE[3..0] → O[19..16]
	IF[3..0] → O[23..20]
	IG[3..0] → O[27..24]
	IH[3..0] → O[31..28]



J8B4_32B

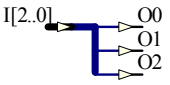
Component	Input → Output
J8B2_16B	IA[7..0] → O[7..0]
	IB[7..0] → O[15..8]
J8B4_32B	IA[7..0] → O[7..0]
	IB[7..0] → O[15..8]
	IC[7..0] → O[23..16]
	ID[7..0] → O[31..24]
J16B2_32B	IA[15..0] → O[15..0]
	IB[15..0] → O[31..16]



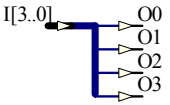
J16B2_32B

JnB_nS

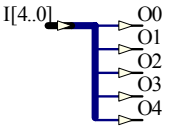
n-Bit Input Bus to n Single Pin Outputs



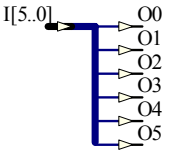
J3B_3S



J4B_4S



J5B_5S



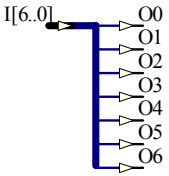
J6B_6S

This component takes an *n*-bit input bus and maps each bit to *n* single output pins. The following components are available:

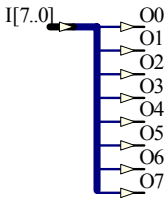
Component	Function
J3B_3S	3-Bit input bus to 3 Single pin outputs
J4B_4S	4-Bit input bus to 4 Single pin outputs
J5B_5S	5-Bit input bus to 5 Single pin outputs
J6B_6S	6-Bit input bus to 6 Single pin outputs
J7B_7S	7-Bit input bus to 7 Single pin outputs
J8B_8S	8-Bit input bus to 8 single pin outputs
J9B_9S	9-Bit input bus to 9 Single pin outputs
J10B_10S	10-Bit input bus to 10 Single pin outputs
J12B_12S	12-Bit input bus to 12 single pin outputs
J16B_16S	16-Bit input bus to 16 single pin outputs

The following table shows how the input pins are mapped to the output pins:

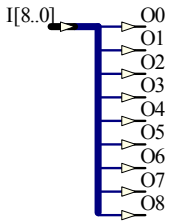
Inputs	Outputs
I(0)	O0
I(1)	O1
I(2)	O2
I(3)	O3
:	:
:	:
I(n)	On



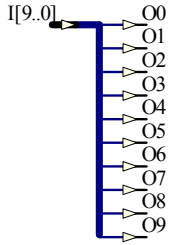
J7B_7S



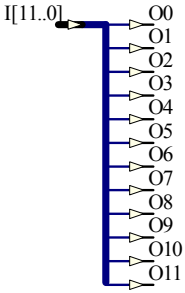
J8B_8S



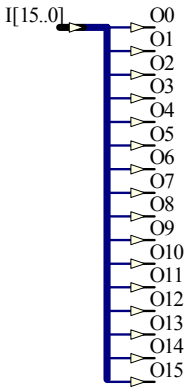
J9B_9S



J10B_10S



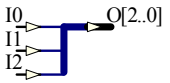
J12B_12S



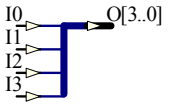
J16B_16S

JnS_nB

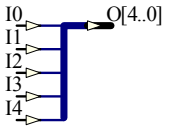
n Single Pin Inputs to Single n-Bit Output Bus



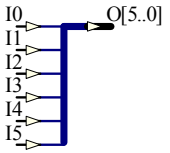
J3S_3B



J4S_4B



J5S_5B



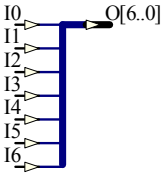
J6S_6B

This component takes *n* single pin inputs and maps each pin to a single *n*-bit output bus. The following components are available:

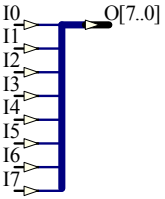
Component	Function
J3S_3B	3 Single pin inputs to single 3-Bit output bus
J4S_4B	4 Single pin inputs to single 4-Bit output bus
J5S_5B	5 Single pin inputs to single 5-Bit output bus
J6S_6B	6 Single pin inputs to single 6-Bit output bus
J7S_7B	7 Single pin inputs to single 7-Bit output bus
J8S_8B	8 Single pin inputs to single 8-Bit output bus
J9S_9B	9 Single pin inputs to single 9-Bit output bus
J10S_10B	10 Single pin inputs to single 10-Bit output bus
J12S_12B	12 Single pin inputs to single 12-Bit output bus
J16S_16B	16 Single pin inputs to single 16-Bit output bus

The following table shows how the input pins are mapped to the output pins:

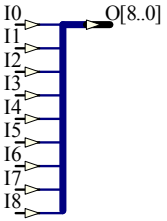
Inputs	Outputs
I0	O(0)
I1	O(1)
I2	O(2)
I3	O(3)
:	:
:	:
In	O(n)



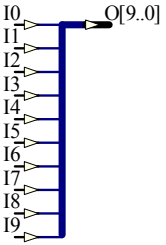
J7S_7B



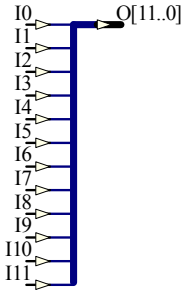
J8S_8B



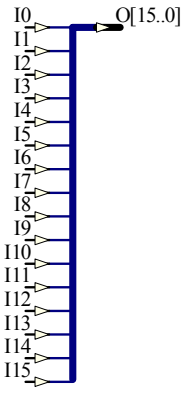
J9S_9B



J10S_10B



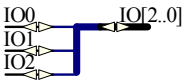
J12S_12B



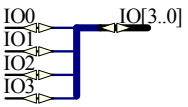
J16S_16B

J*n*S_*n*BX

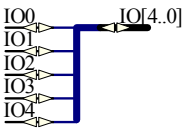
n Single Pin IO to Single *n*-Bit IO Bus



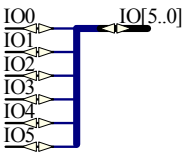
J3S_3BX



J4S_4BX



J5S_5BX



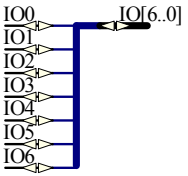
J6S_6BX

This component takes *n* single I/O pins and maps each pin to a single *n*-bit I/O bus. The following components are available:

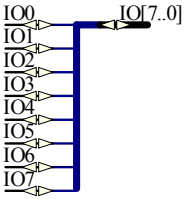
Component	Function
J3S_3BX	3 Single pin IO to single 3-Bit IO bus
J4S_4BX	4 Single pin IO to single 4-Bit IO bus
J5S_5BX	5 Single pin IO to single 5-Bit IO bus
J6S_6BX	6 Single pin IO to single 6-Bit IO bus
J7S_7BX	7 Single pin IO to single 7-Bit IO bus
J8S_8BX	8 Single pin IO to single 8-Bit IO bus
J9S_9BX	9 Single pin IO to single 9-Bit IO bus
J10S_10BX	10 Single pin IO to single 10-Bit IO bus
J12S_12BX	12 single-Bit IO to single 12-Bit IO bus
J16S_16BX	16 Single pin IO to single 16-Bit IO bus

The following table shows how the single pins are mapped to the bus pins:

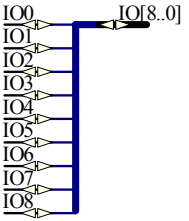
Single	Bus
IO0	IO(0)
IO1	IO(1)
IO2	IO(2)
IO3	IO(3)
:	:
:	:
IOn	IO(n)



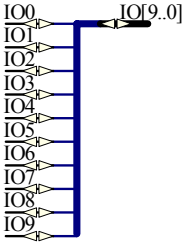
J7S_7BX



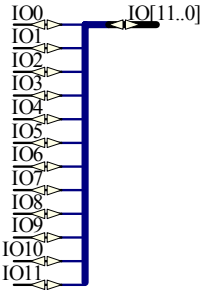
J8S_8BX



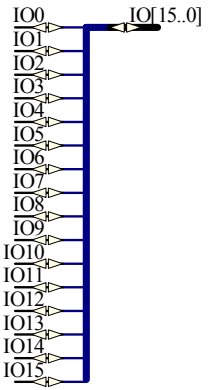
J9S_9BX



J10S_10BX



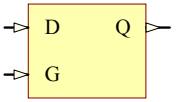
J12S_12BX



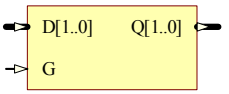
J16S_16BX

LD, 2, 3, 4, 8, 16, 32

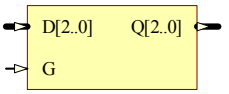
Transparent Data Latches



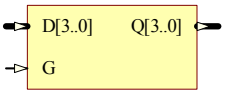
LD



LD2B



LD3B



LD4B

LD, LD2, LD3, LD4, LD8, LD16 and LD32 are, respectively 1-, 2-, 3-, 4-, 8-, 16-, 32-bit transparent data latches.

The data output Q follows the value of the data input D while the gate input (G) is High, otherwise Q remains unchanged.

Inputs		Output
G	**D**	**Q**
0	x	No Chg
1	d	d

For LD2, D = D1, D0; Q = Q1, Q0

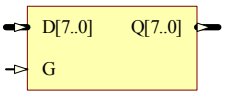
For LD3, D = D2, D1, D0; Q = Q2, Q1, Q0

For LD4, D = D3, D2, D1, D0; Q = Q3, Q2, Q1, Q0

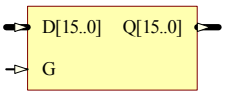
For LD8, D = D7 - D0; Q = Q7 - Q0

For LD16, D = D15 - D0; Q = Q15 - Q0

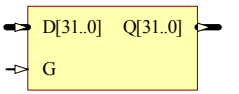
For LD32, D = D31 - D0; Q = Q31 - Q0



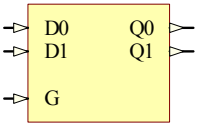
LD8B



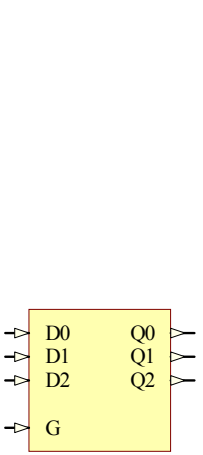
LD16B



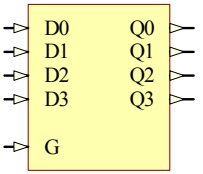
LD32B



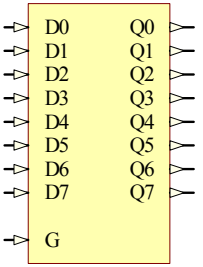
LD2S



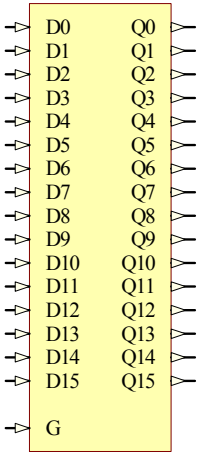
LD3S



LD4S



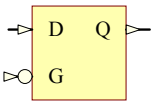
LD8S



LD16S

LD_1

Transparent Data Latch with Inverted Gate



LD_1

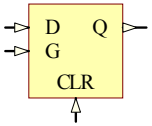
LD_1 is a 1-bit transparent data latch with inverted gate.

The data output Q follows the value of the data input D while the inverted gate input (G) is Low, otherwise Q remains unchanged.

Inputs		Output
G	**D**	**Q**
0	0	0
0	1	1
1	x	No Chg

LDC

Transparent Data Latch with Asynchronous Clear



LDC

LDC is a 1-bit transparent data latch with asynchronous clear.

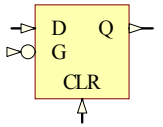
The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored and output Q is cleared to Low.

The data output Q follows the value of the input data D while the gate input (G) is High, otherwise Q remains unchanged.

Inputs			Output
CLR	**G**	**D**	**Q**
1	x	x	0
0	1	0	0
0	1	1	1
0	0	x	No Chg

LDC_1

Transparent Data Latch with Asynchronous Clear and Inverted Gate



LDC_1

LDC_1 is a 1-bit transparent data latch with asynchronous clear and inverted gate.

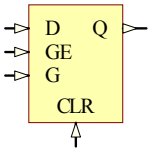
The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored and output Q is cleared to Low.

The data output Q follows the value of the data input D while the inverted gate input (G) is Low, otherwise Q remains unchanged.

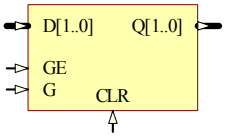
Inputs			Output
CLR	G	D	Q
1	x	x	0
0	0	0	0
0	0	1	1
0	1	x	No Chg

LDCE, LD2CE, LD4CE, LD8CE, LD16CE, LD32CE

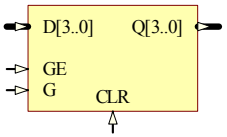
Transparent Data Latches with Asynchronous Clear and Gate Enable



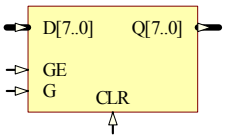
LDCE



LD2CEB



LD4CEB



LD8CEB

LDCE, LD4CE, LD8CE, LD16CE and LD32CE are respectively 1-, 2-, 4-, 8-, 16-, 32-bit transparent data latches with asynchronous clear and gate enable.

The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored and the outputs are cleared to Low.

The gate enable (GE) is the second highest priority input after CLR. GE is used to disable the gate input (G). When GE is low, G is ignored and data outputs Q remain unchanged.

The data output Q follows the value of the input data D while the gate input (G) is High, otherwise Q remains unchanged.

Inputs				Output
CLR	**GE**	**G**	**Dn**	**Qn**
1	x	x	x	0
0	0	x	x	No Chg
0	1	1	1	1
0	1	1	0	0
0	1	0	x	No Chg

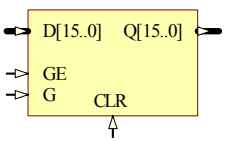
For LD2CE, Dn = D1, D0; Qn = Q1, Q0

For LD4CE, Dn = D3, D2, D1, D0; Qn = Q3, Q2, Q1, Q0

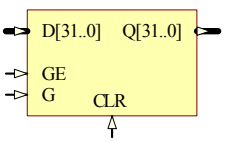
For LD8CE, Dn = D7 - D0; Qn = Q7 - Q0

For LD16CE, Dn = D15 - D0; Qn = Q15 - Q0

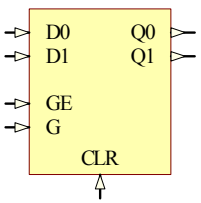
For LD32CE, Dn = D31 - D0; Qn = Q31 - Q0



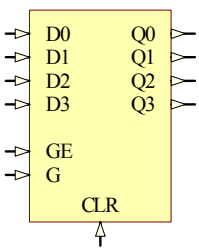
LD16CEB



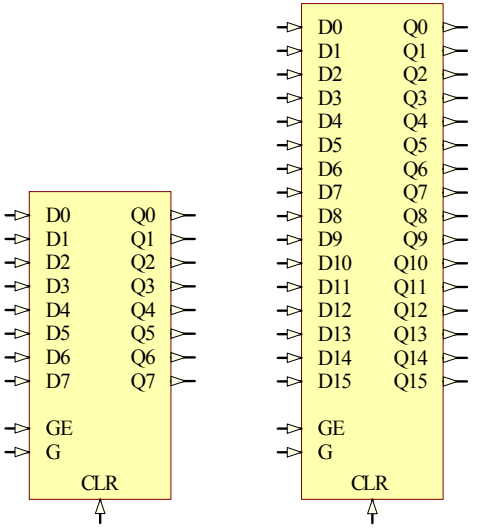
LD32CEB



LD2CES



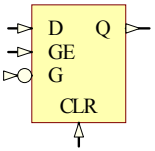
LD4CES



LD8CES LD16CES

LDCE_1

Transparent Data Latch with Asynchronous Clear, Gate Enable, and Inverted Gate



LDCE_1

LDCE_1 is a 1-bit Transparent Data Latch with Asynchronous Clear, Gate Enable, and Inverted Gate.

The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored and output Q is cleared to Low.

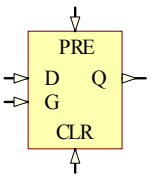
The gate enable (GE) is the second highest priority input after CLR. GE is used to disable the gate input (G). When GE is low, G is ignored and data output Q remains unchanged.

The data output Q follows the value of the data input D while the inverted gate input (G) is Low, otherwise Q remains unchanged.

Inputs				Output
CLR	**GE**	**G**	**D**	**Q**
1	x	x	x	0
0	0	x	x	No Chg
0	1	0	0	0
0	1	0	1	1
0	1	1	x	No Chg

LDCP

Transparent Data Latch with Asynchronous Clear and Preset



LDCP

LDCP is a 1-bit transparent data latch with asynchronous clear and preset.

The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored and output Q is cleared to Low.

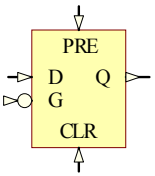
The asynchronous preset (PRE) is the second highest priority input after CLR. When PRE is High and CLR is Low, all other inputs are ignored and output Q is set to High.

The data output Q follows the value of the input data D while the gate input (G) is High, otherwise Q remains unchanged.

Inputs				Output
CLR	**PRE**	**G**	**D**	**Q**
1	x	x	x	0
0	1	x	x	1
0	0	1	1	1
0	0	1	0	0
0	0	0	x	No Chg

LDCP_1

Transparent Data Latch with Asynchronous Clear and Preset and Inverted Gate



LDCP_1

LDCP_1 is a 1-bit transparent data latch with asynchronous clear and preset and inverted gate.

The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored and output Q is cleared to Low.

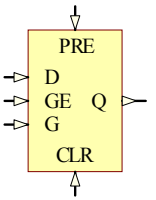
The asynchronous preset (PRE) is the second highest priority input after CLR. When PRE is High and CLR is Low, all other inputs are ignored and output Q is set to High.

The data output Q follows the value of the data input D while the inverted gate input (G) is Low, otherwise Q remains unchanged.

Inputs				Output
CLR	**PRE**	**G**	**D**	**Q**
1	x	x	x	0
0	1	x	x	1
0	0	0	1	1
0	0	0	0	0
0	0	1	x	No Chg

LDCPE

Transparent Data Latch with Asynchronous Clear and Preset and Gate Enable



LDCPE

LDCPE is a 1-bit transparent data latch with asynchronous clear and preset and gate enable.

The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored and output Q is cleared to Low.

The asynchronous preset (PRE) is the second highest priority input after CLR. When PRE is High and CLR is Low, all other inputs are ignored and output Q is set to High.

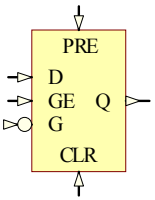
The gate enable (GE) is the third highest priority input after CLR and PRE. GE is used to disable the gate input (G). When GE is Low, G is ignored and output Q remains unchanged.

The data output Q follows the value of the input data D while the gate input (G) is High, otherwise Q remains unchanged.

Inputs					Output
CLR	PRE	GE	G	D	Q
1	x	x	x	x	0
0	1	x	x	x	1
0	0	0	x	x	No Chg
0	0	1	1	0	0
0	0	1	1	1	1
0	0	1	0	x	No Chg

LDCPE_1

Transparent Data Latch with Asynchronous Clear and Preset, Gate Enable, and Inverted Gate



LDCPE_1

LDCPE_1 is a 1-bit transparent data latch with asynchronous clear and preset and inverted gate enable.

The asynchronous clear (CLR) is the highest priority input. When CLR is High, all other inputs are ignored and output Q is cleared to Low.

The asynchronous preset (PRE) is the second highest priority input after CLR. When PRE is High and CLR is Low, all other inputs are ignored and output Q is set to High.

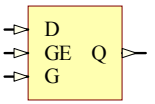
The gate enable (GE) is the third highest priority input after CLR and PRE. GE is used to disable the gate input (G). When GE is Low, G is ignored and output Q remains unchanged.

The data output Q follows the value of the data input D while the inverted gate input (G) is Low otherwise Q remains unchanged.

Inputs					Output
CLR	**PRE**	**GE**	**G**	**D**	**Q**
1	x	x	x	x	0
0	1	x	x	x	1
0	0	0	x	x	No Chg
0	0	1	0	0	0
0	0	1	0	1	1
0	0	1	1	x	No Chg

LDE

Transparent Data Latch with Gate Enable



LDE

LDE is a 1-bit transparent data latch with gate enable.

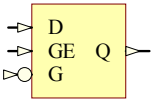
The gate enable (GE) is used to disable the gate input (G), when GE is Low, G is ignored and output Q remains unchanged.

The data output Q follows the value of the input data D while the gate input (G) is High, otherwise Q remains unchanged.

Inputs			Output
GE	G	D	Q
0	x	x	No Chg
1	1	0	0
1	1	1	1
1	0	x	No Chg

LDE_1

Transparent Data Latch with Gate Enable and Inverted Gate



LDE_1

LDE_1 is a 1-bit transparent data latch with gate enable and inverted gate.

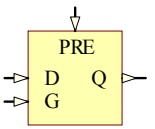
The gate enable (GE) is used to disable the gate input (G), when GE is Low, G is ignored and output Q remains unchanged.

The data output Q follows the value of the data input D while the inverted gate input (G) is Low otherwise Q remains unchanged.

Inputs			Output
GE	**G**	**D**	**Q**
0	x	x	No Chg
1	0	0	0
1	0	1	1
1	1	x	No Chg

LDP

Transparent Data Latch with Asynchronous Preset



LDP

LDP is a 1-bit transparent latch with asynchronous preset.

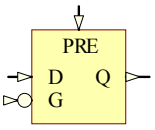
The asynchronous preset (PRE) is the highest priority input. When PRE is High, all other inputs are ignored and the output Q is set to High.

The data output Q follows the value of the input data D while the gate input (G) is High, otherwise Q remains unchanged.

Inputs			Output
PRE	G	D	Q
1	x	x	1
0	1	0	0
0	1	1	1
0	0	x	No Chg

LDP_1

Transparent Data Latch with Asynchronous Preset and Inverted Gate



LDP_1

LDP_1 is a 1-bit transparent latch with asynchronous preset.

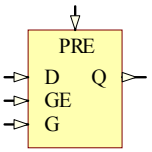
The asynchronous preset (PRE) is the highest priority input. When PRE is High, all other inputs are ignored and the output Q is set to High.

The data output Q follows the value of the data input D while the inverted gate input (G) is Low otherwise Q remains unchanged.

Inputs			Output
PRE	**G**	**D**	**Q**
1	x	x	1
0	0	0	0
0	0	1	1
0	1	x	No Chg

LDPE

Transparent Data Latch with Asynchronous Preset and Gate Enable



LDPE

LDPE is a 1-bit transparent data latch with asynchronous preset and gate enable.

The asynchronous preset (PRE) is the highest priority input. When PRE is High, all other inputs are ignored and output Q is set to High.

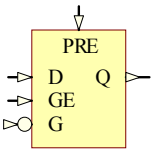
The gate enable (GE) is the second highest priority input after PRE. GE is used to disable the gate input (G). When GE is Low, G is ignored and output Q remains unchanged.

The data output Q follows the value of the input data D while the gate input (G) is High, otherwise remains Q unchanged.

Inputs				Output
PRE	**GE**	**G**	**D**	**Q**
1	x	x	x	1
0	0	x	x	No Chg
0	1	1	0	0
0	1	1	1	1
0	1	0	x	No Chg

LDPE_1

Transparent Data Latch with Asynchronous Preset, Gate Enable, and Inverted Gate



LDPE_1

LDPE_1 is a 1-bit transparent data latch with asynchronous preset and gate enable and inverted gate.

The asynchronous preset (PRE) is the highest priority input. When PRE is High, all other inputs are ignored and output Q is set to High.

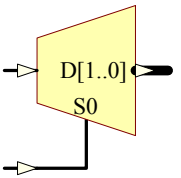
The gate enable (GE) is the second highest priority input after PRE. GE is used to disable the gate input (G). When GE is Low, G is ignored and output Q remains unchanged.

The data output Q follows the value of the data input D while the inverted gate input (G) is Low otherwise Q remains unchanged.

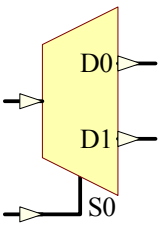
Inputs				Output
PRE	**GE**	**G**	**D**	**Q**
1	x	x	x	1
0	0	x	x	No Chg
0	1	0	0	0
0	1	0	1	1
0	1	1	x	No Chg

M*n*_B1B2, M*n*_S1S2, M1_S1B2

1-to-2 Demultiplexers



M1_S1B2



M1_S1S2

Mn_B1B2, Mn_S1S2, M1_S1B2 are various *n*-bit data width 1-to-2 demultiplexers, available in bus-to-bus, pin-to-pin and pin-to-bus versions.

M*n*_B1B2 are bus-to-bus version of 1-to-2 demultiplexers, which switch 1 x *n*-bit bus to 2 x *n*-bit bus according to the select input. The width of the data bus, *n* is available in 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, and 32-bit.

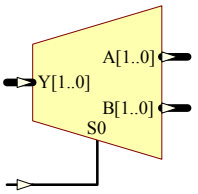
M*n*_S1S2 are pin-to-pin version of 1-to-2 demultiplexers, which switch 1 x *n*-single pins to 2 x *n*-single pins according to the select input. The number of single pin, *n* is available in 1, 2, 3, 4, 5, 6, 7, and 8.

M1_S1B2 is bus-to-pin version of 1-to-2 demultiplexer, which switches a single pin to 1-bit of the 2-bit bus according to the select input.

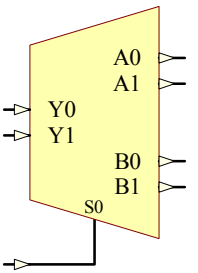
Select	Input	Outputs	
S0	O	D1	D0
	Y	B	A
0	d	0	d
1	d	d	0

For M1 follow O, D0, D1

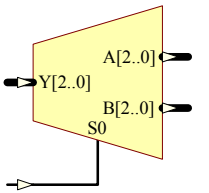
For Mn follow Y, A, B



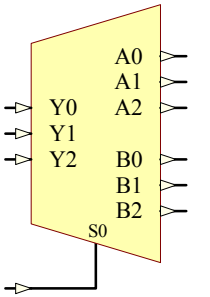
M2_B1B2



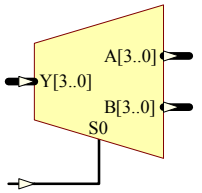
M2_S1S2



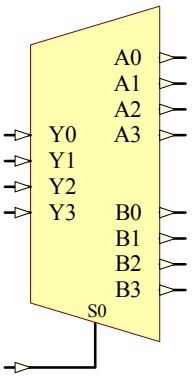
M3_B1B2



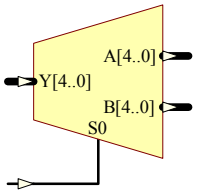
M3_S1S2



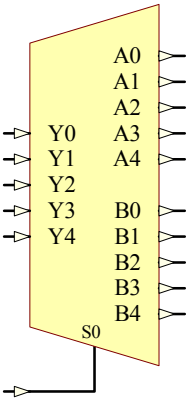
M4_B1B2



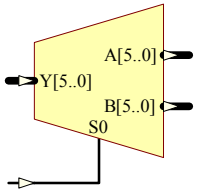
M4_S1S2



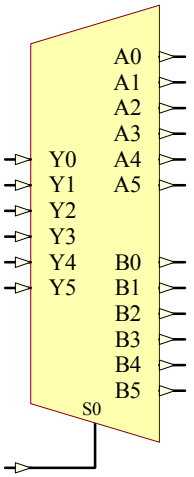
M5_B1B2



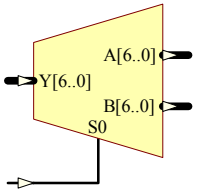
M5_S1S2



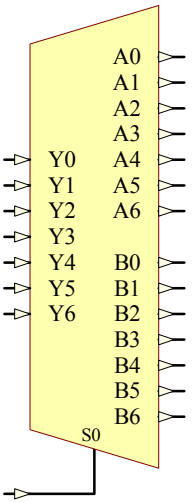
M6_B1B2



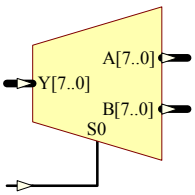
M6_S1S2



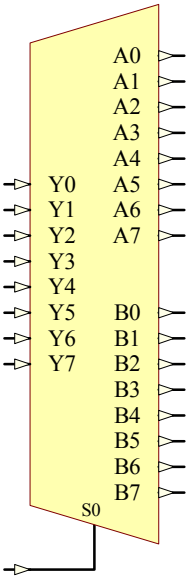
M7_B1B2



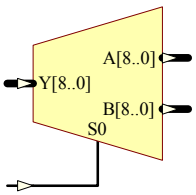
M7_S1S2



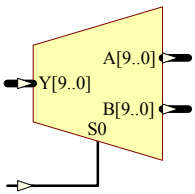
M8_B1B2



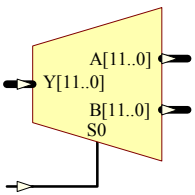
M8_S1S2



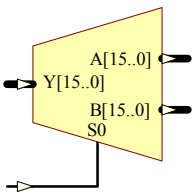
M9_B1B2



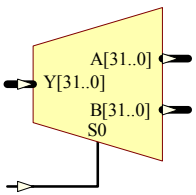
M10_B1B2



M12_B1B2



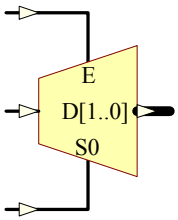
M16_B1B2



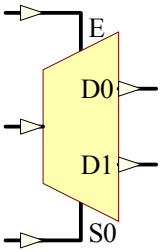
M32_B1B2

M*n*_B1B2E, M*n*_S1S2E, M1_S1B2E

1-to-2 Demultiplexers with Enable



M1_S1B2E



M1_S1S2E

Mn_B1B2E, Mn_S1S2E, M1_S1B2E are various *n*-bit data width 1-to-2 demultiplexers with enable, available in bus-to-bus, pin-to-pin and pin-to-bus versions.

M*n*_B1B2E are bus-to-bus version of 1-to-2 demultiplexers, which switch 1 x *n*-bit bus to 2 x *n*-bit bus according to the select input. The width of the data bus, *n* is available in 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, and 32-bit.

M*n*_S1S2E are pin-to-pin version of 1-to-2 demultiplexers, which switch 1 x *n*-single pins to 2 x *n*-single pins according to the select input. The number of single pin, *n* is available in 1, 2, 3, 4, 5, 6, 7, and 8.

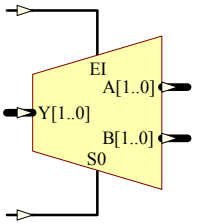
M1_S1B2E is bus-to-pin version of 1-to-2 demultiplexer, which switches a single pin to 1-bit of the 2-bit bus according to the select input.

Enable is the highest priority input, when enable is Low, all inputs are ignored, and outputs remain Low. When enable is High, the demultiplexers switch the data from inputs to output.

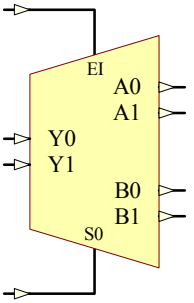
Enable	Select	Input	Outputs	
E	S0	O	D1	D0
EI		Y	B	A
0	x	x	0	0
1	0	d	0	d
1	1	d	d	0

For M1 follow E, O, D0, D1

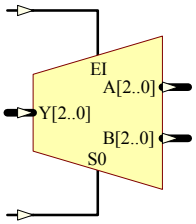
For Mn follow EI, Y, A, B



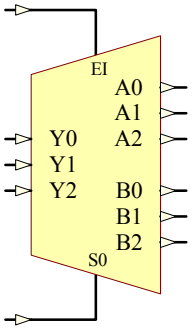
M2_B1B2E



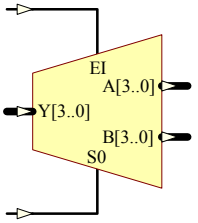
M2_S1S2E



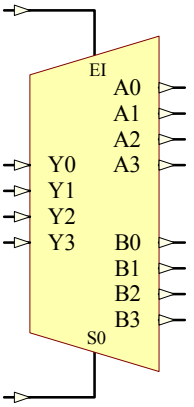
M3_B1B2E



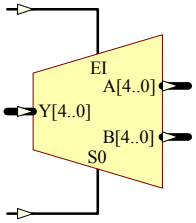
M3_S1S2E



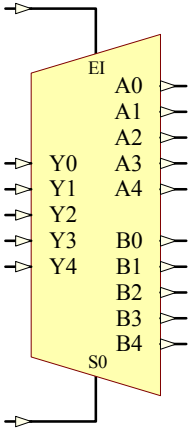
M4_B1B2E



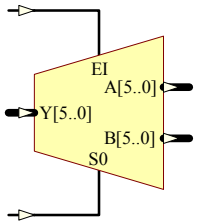
M4_S1S2E



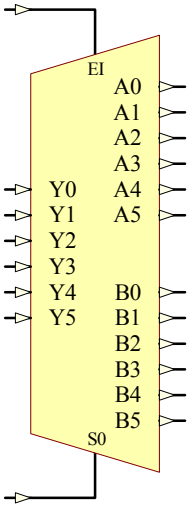
M5_B1B2E



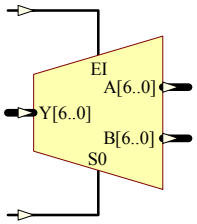
M5_S1S2E



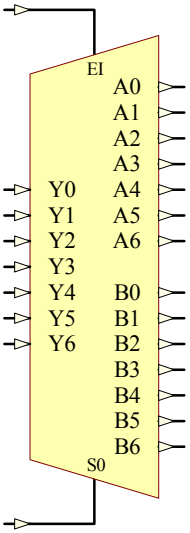
M6_B1B2E



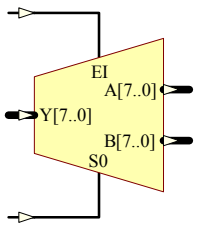
M6_S1S2E



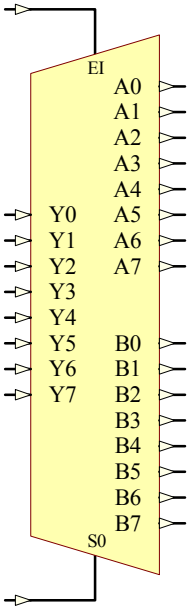
M7_B1B2E



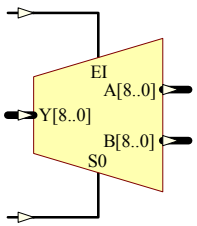
M7_S1S2E



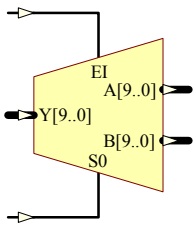
M8_B1B2E



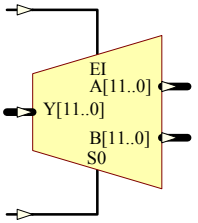
M8_S1S2E



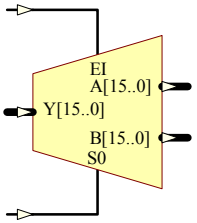
M9_B1B2E



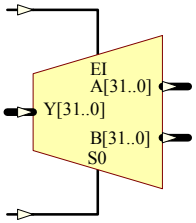
M10_B1B2E



M12_B1B2E



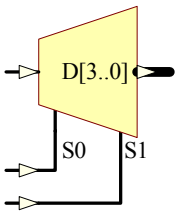
M16_B1B2E



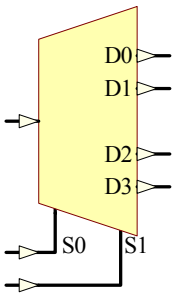
M32_B1B2E

M*n*_B1B4, M*n*_S1S4, M1_S1B4, M*n*_B1B4_SB, M*n*_S1S4_SB, M1_S1B4_SB

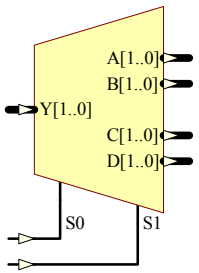
1-to-4 Demultiplexers



M1_S1B4



M1_S1S4



M2_B1B4

M*n*_B1B4, M*n*_S1S4, M1_S1B4, Mn_B1B4_SB, Mn_S1S4_SB and M1_S1B4_SB are various *n*-bit data width 1-to-4 demultiplexers, available in bus-to-bus, pin-to-pin and pin-to-bus versions.

M*n*_B1B4 and M*n*_B1B4_SB are bus-to-bus versions of the 1-to-4 demultiplexers, which switch a 1 x *n*-bit bus to a 4 x *n*-bit bus according to the select inputs. The width of the data bus, *n,* is available in 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, and 32-bit.

M*n*_S1S4 and M*n*_S1S4_SB are pin-to-pin versions of the 1-to-4 demultiplexers, which switch 1 x *n*-single pins to 4 x *n*-single pins according to the select inputs. The number of single pin, *n* is available in 1, 2, 3, and 4.

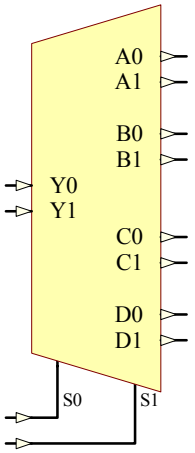
M1_S1B4 and M1_S1B4_SB are pin-to-bus versions of the 1-to-4 demultiplexer, which switches a single pin to 1-bit of the 4-bit bus according to the select inputs.

The demultiplexers with the "_SB" suffix in the name have the Select input pins (S1-S0) grouped into a single bus pin (S[1..0]), whereas those demultiplexers without this suffix leave the Select input pins ungrouped.

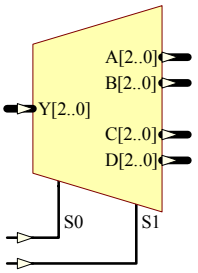
Select Inputs		Input	Outputs			
S1	S0	O	D3	D2	D1	D0
		Y	D	C	B	A
0	0	d	0	0	0	d
0	1	d	0	0	d	0
1	0	d	0	d	0	0
1	1	d	d	0	0	0

For M1 follow O, D0, D1, D3

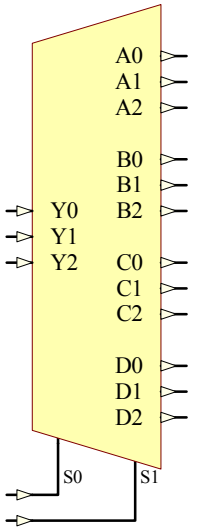
For Mn follow Y, A, B, C, D



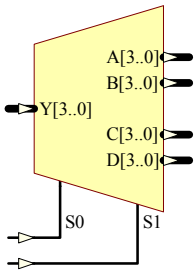
M2_S1S4



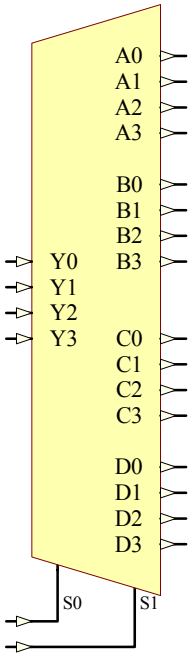
M3_B1B4



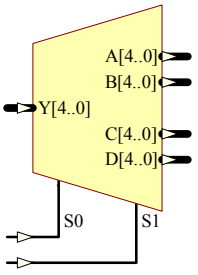
M3_S1S4



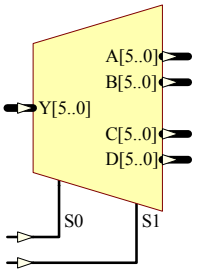
M4_B1B4



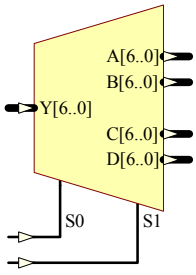
M4_S1S4



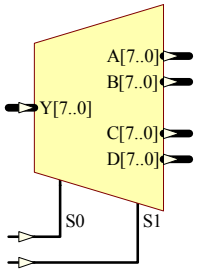
M5_B1B4



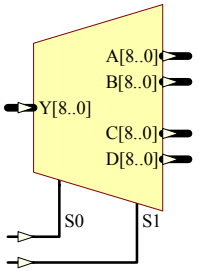
M6_B1B4



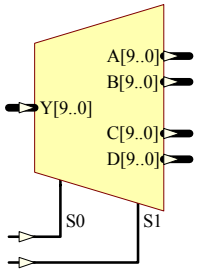
M7_B1B4



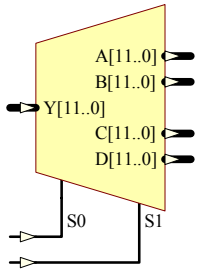
M8_B1B4



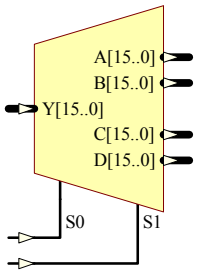
M9_B1B4



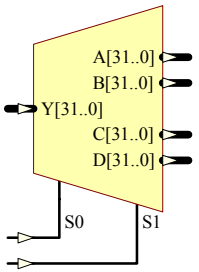
M10_B1B4



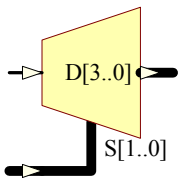
M12_B1B4



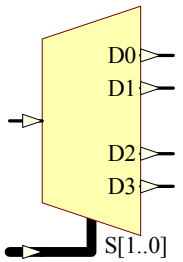
M16_B1B4



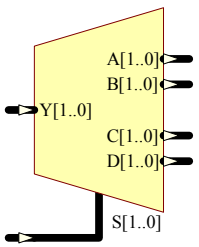
M32_B1B4



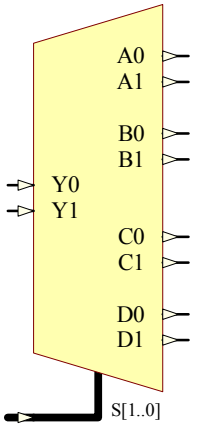
M1_S1B4_SB



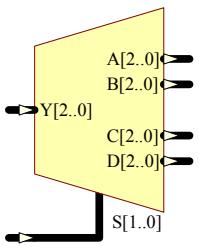
M1_S1S4_SB



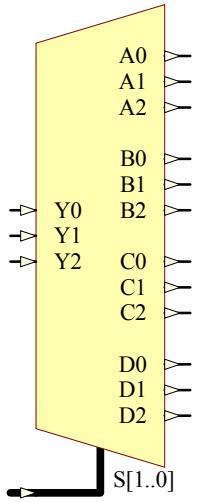
M2_B1B4_SB



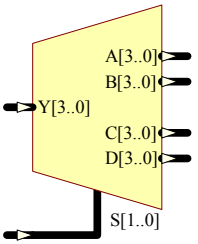
M2_S1S4_SB



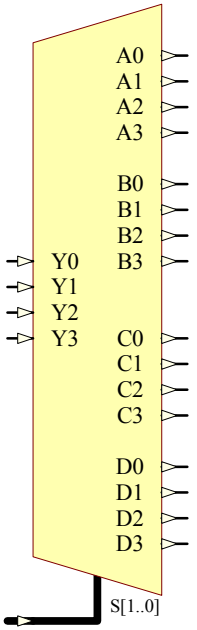
M3_B1B4_SB



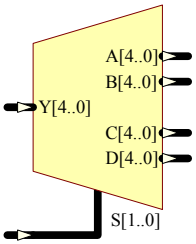
M3_S1S4_SB



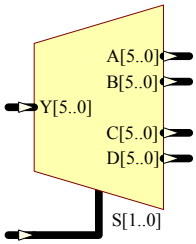
M4_B1B4_SB



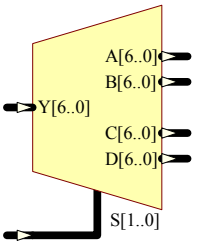
M4_S1S4_SB



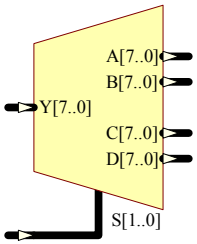
M5_B1B4_SB



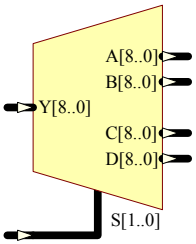
M6_B1B4_SB



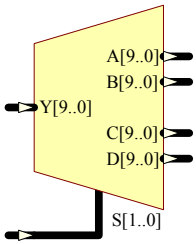
M7_B1B4_SB



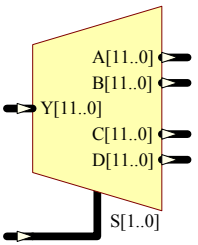
M8_B1B4_SB



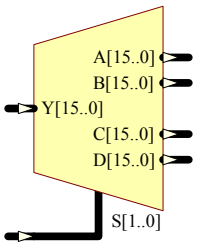
M9_B1B4_SB



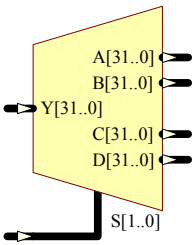
M10_B1B4_SB



M12_B1B4_SB



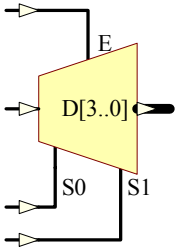
M16_B1B4_SB



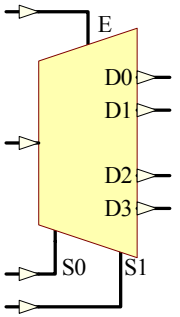
M32_B1B4_SB

M*n*_B1B4E, M*n*_S1S4E, M1_S1B4E, M*n*_B1B4E_SB, M*n*_S1S4E_SB, M1_S1B4E_SB

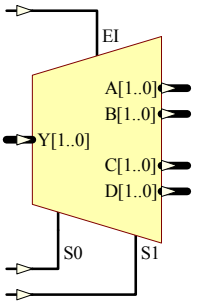
1-to-4 Demultiplexers with Enable



M1_S1B4E



M1_S1S4E



M2_B1B4E

M*n*_B1B4E, M*n*_S1S4E, M1_S1B4E, M*n*_B1B4E_SB, M*n*_S1S4E_SB and M1_S1B4E_SB are various *n*-bit data width 1-to-4 demultiplexers, available in bus-to-bus, pin-to-pin and pin-to-bus versions.

M*n*_B1B4E and M*n*_B1B4E_SB are bus-to-bus versions of the 1-to-4 demultiplexers, which switch a 1 x *n*-bit bus to a 4 x *n*-bit bus according to the select inputs. The width of the data bus, *n,* is available in 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, and 32-bit.

M*n*_S1S4E and M*n*_S1S4E_SB are pin-to-pin versions of the 1-to-4 demultiplexers, which switch 1 x *n*-single pins to 4 x *n*-single pins according to the select inputs. The number of single pin, *n,* is available in 1, 2, 3, and 4.

M1_S1B4E and M1_S1B4E_SB are pin-to-bus versions of the 1-to-4 demultiplexer, which switches a single pin to 1-bit of the 4-bit bus according to the select inputs.

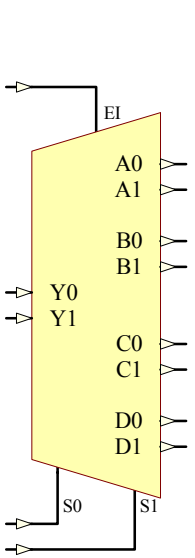
Enable is the highest priority input, when enable is Low, all inputs are ignored, and outputs remain Low. When enable is High, the demultiplexers switch the data from inputs to output.

The demultiplexers with the "_SB" suffix in the name have the Select input pins (S1-S0) grouped into a single bus pin (S[1..0]), whereas those demultiplexers without this suffix leave the Select input pins ungrouped.

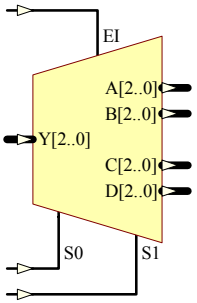
Enable	Select Inputs		Input	Outputs			
E EI	S1	S0	O Y	D3 D	D2 C	D1 B	D0 A
0	x	x	x	0	0	0	0
1	0	0	d	0	0	0	d
1	0	1	d	0	0	d	0
1	1	0	d	0	d	0	0
1	1	1	d	d	0	0	0

For M1 follow E, O, D0, D1, D3

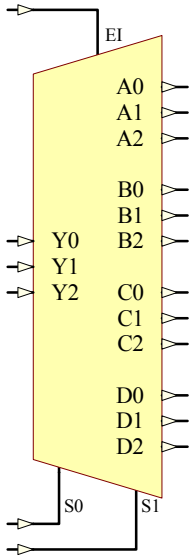
For Mn follow EI, Y, A, B, C, D



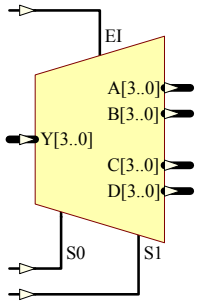
M2_S1S4E



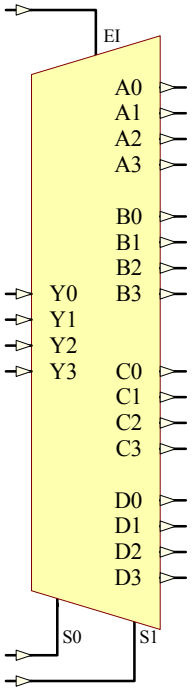
M3_B1B4E



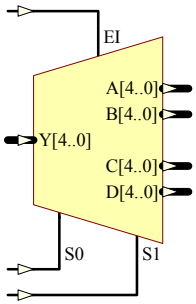
M3_S1S4E



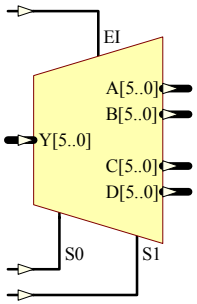
M4_B1B4E



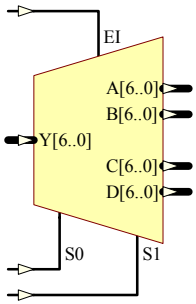
M4_S1S4E



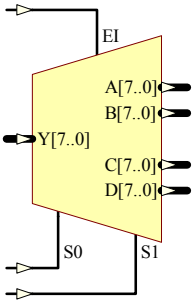
M5_B1B4E



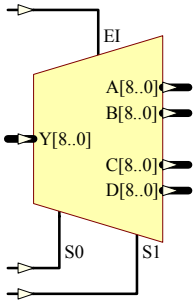
M6_B1B4E



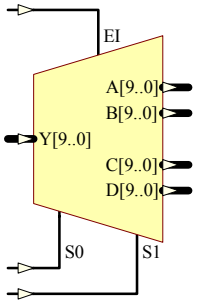
M7_B1B4E



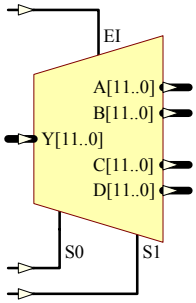
M8_B1B4E



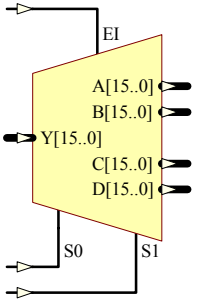
M9_B1B4E



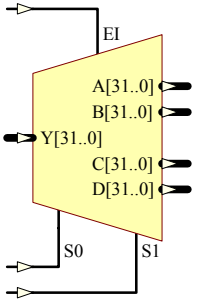
M10_B1B4E



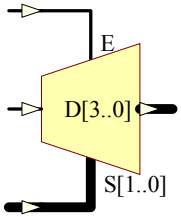
M12_B1B4E



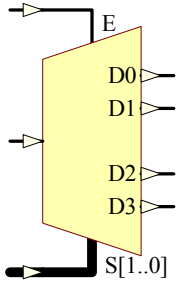
M16_B1B4E



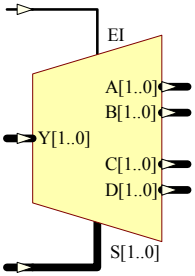
M32_B1B4E



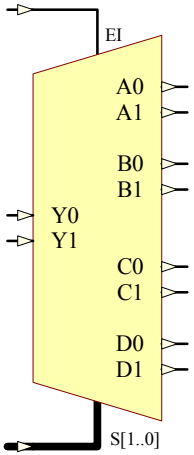
M1_S1B4E_SB



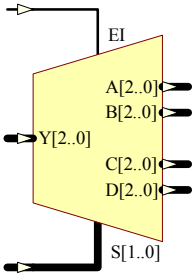
M1_S1S4E_SB



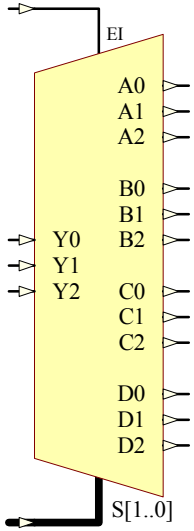
M2_B1B4E_SB



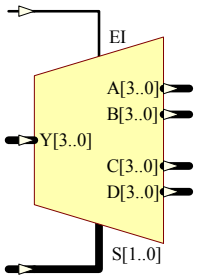
M2_S1S4E_SB



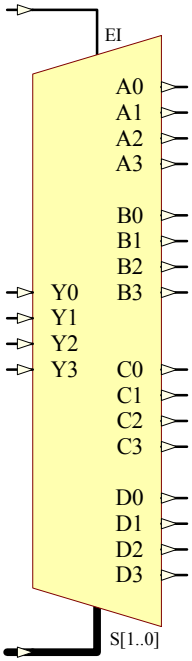
M3_B1B4E_SB



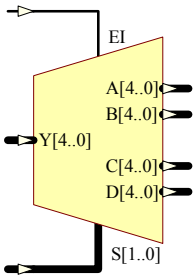
M3_S1S4E_SB



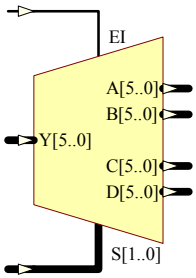
M4_B1B4E_SB



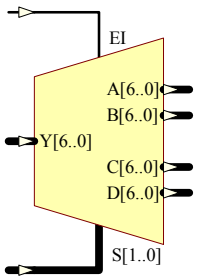
M4_S1S4E_SB



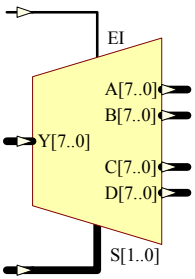
M5_B1B4E_SB



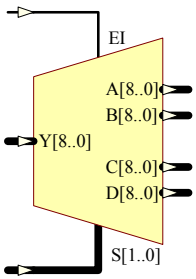
M6_B1B4E_SB



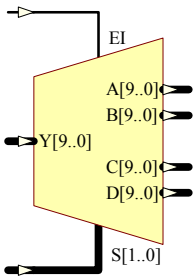
M7_B1B4E_SB



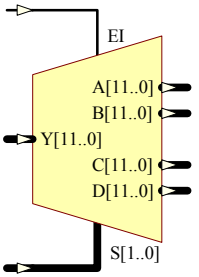
M8_B1B4E_SB



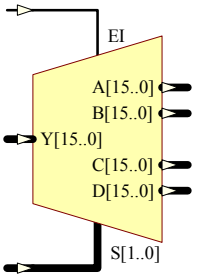
M9_B1B4E_SB



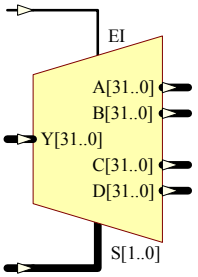
M10_B1B4E_SB



M12_B1B4E_SB



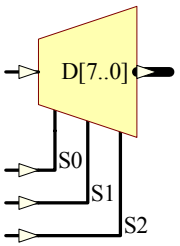
M16_B1B4E_SB



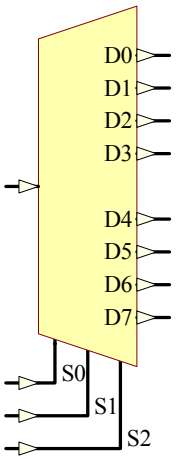
M32_B1B4E_SB

M*n*_B1B8, M*n*_S1S8, M1_S1B8, M*n*_B1B8_SB, M*n*_S1S8_SB, M1_S1B8_SB

1-to-8 Demultiplexers



M1_S1B8



M1_S1S8

M*n*_B1B8, M*n*_S1S8, M1_S1B8, M*n*_B1B8_SB, M*n*_S1S8_SB and M1_S1B8_SB are various *n*-bit data width 1-to-8 demultiplexers, available in bus-to-bus, pin-to-pin and pin-to-bus versions.

M*n*_B1B8 and M*n*_B1B8_SB are bus-to-bus versions of the 1-to-8 demultiplexers, which switch a 1 x *n*-bit bus to an 8 x *n*-bit bus according to the select inputs. The width of the data bus, *n,* is available in 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, and 32-bit.

M*n*_S1S8 and M*n*_S1S8_SB are pin-to-pin versions of the 1-to-8 demultiplexers, which switch 1 x *n*-single pins to 8 x *n*-single pins according to the select inputs. The number of single pins, *n,* is available in 1, and 2.

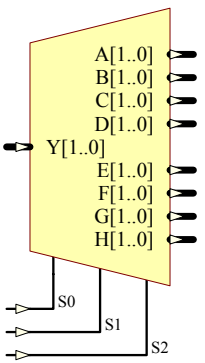
M1_S1B8 and M1_S1B8_SB are bus-to-pin versions of the 1-to-8 demultiplexer, which switches a single pin to 1-bit of the 8-bit bus according to the select inputs.

The demultiplexers with the "_SB" suffix in the name have the Select input pins (S2-S0) grouped into a single bus pin (S[2..0]), whereas those demultiplexers without this suffix leave the Select input pins ungrouped.

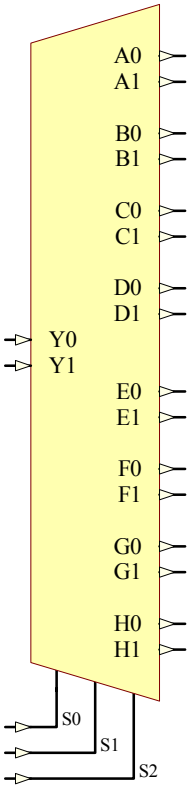
Select Inputs			Input	Outputs							
S2	S1	S0	O	D0	D1	D2	D3	D4	D5	D6	D7
			Y	A	B	C	D	E	F	G	H
0	0	0	d	d	0	0	0	0	0	0	0
0	0	1	d	0	d	0	0	0	0	0	0
0	1	0	d	0	0	d	0	0	0	0	0
0	1	1	d	0	0	0	d	0	0	0	0
1	0	0	d	0	0	0	0	d	0	0	0
1	0	1	d	0	0	0	0	0	d	0	0
1	1	0	d	0	0	0	0	0	0	d	0
1	1	1	d	0	0	0	0	0	0	0	d

For M1 follow O, D0, D1, D3, D4, D5, D6, D7

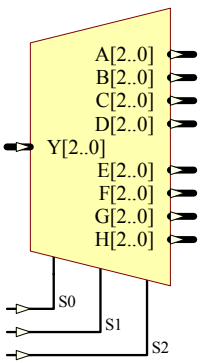
For Mn follow Y, A, B, C, D, E, F, G, H



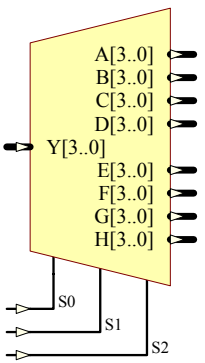
M2_B1B8



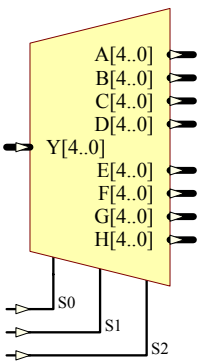
M2_S1S8



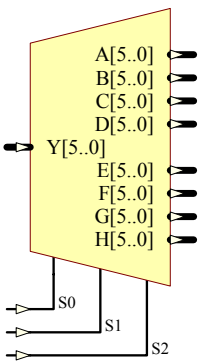
M3_B1B8



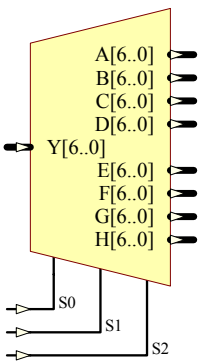
M4_B1B8



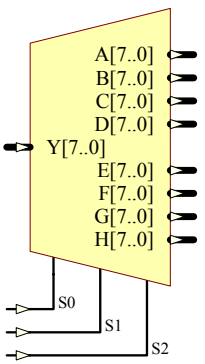
M5_B1B8



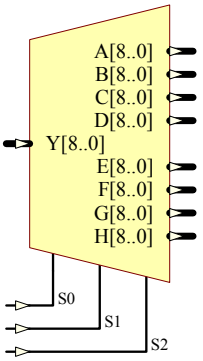
M6_B1B8



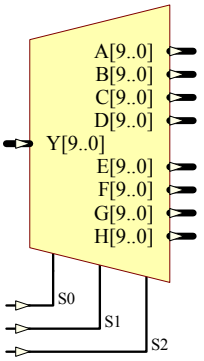
M7_B1B8



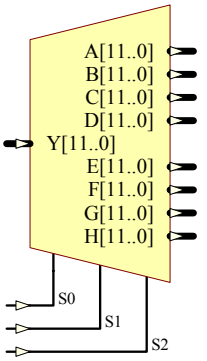
M8_B1B8



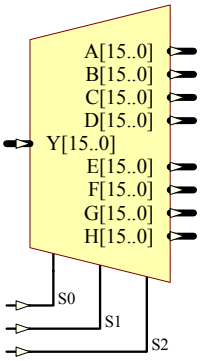
M9_B1B8



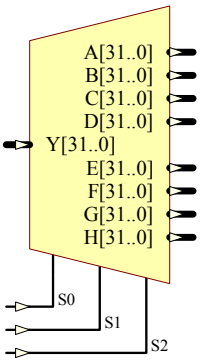
M10_B1B8



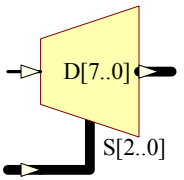
M12_B1B8



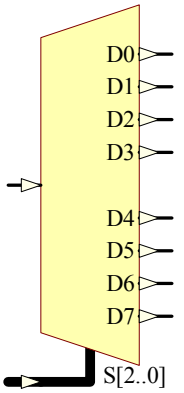
M16_B1B8



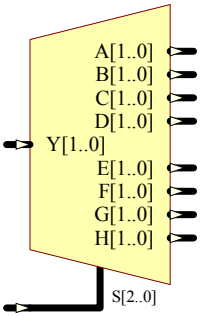
M32_B1B8



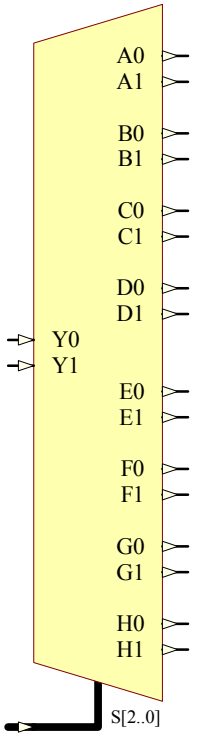
M1_S1B8_SB



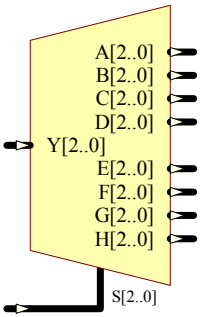
M1_S1S8_SB



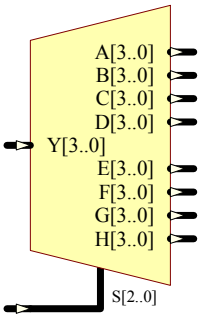
M2_B1B8_SB



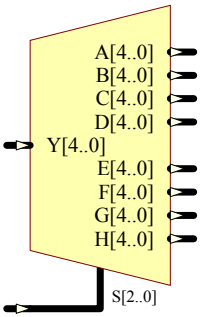
M2_S1S8_SB



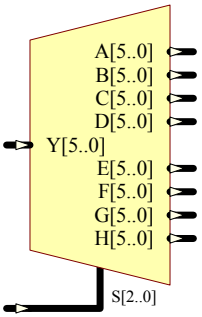
M3_B1B8_SB



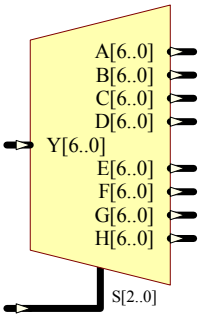
M4_B1B8_SB



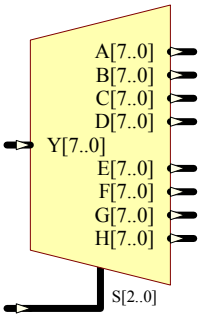
M5_B1B8_SB



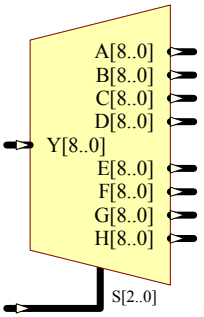
M6_B1B8_SB



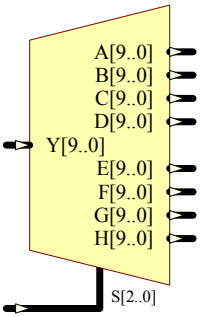
M7_B1B8_SB



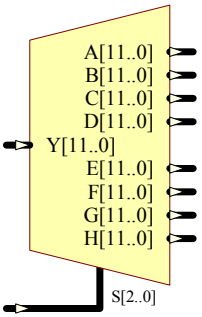
M8_B1B8_SB



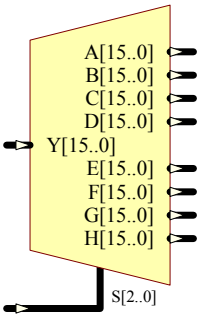
M9_B1B8_SB



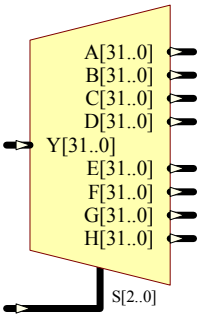
M10_B1B8_SB



M12_B1B8_SB



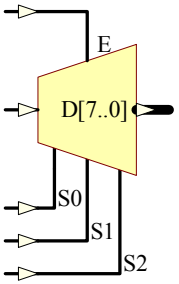
M16_B1B8_SB



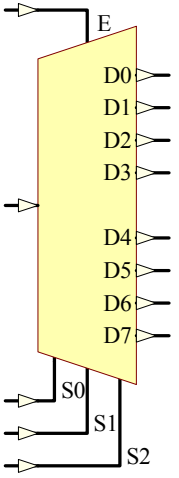
M32_B1B8_SB

M*n*_B1B8E, M*n*_S1S8E, M1_S1B8E, M*n*_B1B8E_SB, M*n*_S1S8E_SB, M1_S1B8E_SB

1-to-8 Demultiplexers with Enable



M1_S1B8E



M1_S1S8E

M*n*_B1B8E, M*n*_S1S8E, M1_S1B8E, M*n*_B1B8E_SB, M*n*_S1S8E_SB and M1_S1B8E_SB are various *n*-bit data width 1-to-8 demultiplexers with enable, available in bus-to-bus, pin-to-pin and pin-to-bus versions.

M*n*_B1B8E and M*n*_B1B8E_SB are bus-to-bus versions of the 1-to-8 demultiplexers, which switch a 1 x *n*-bit bus to an 8 x *n*-bit bus according to the select inputs. The width of the data bus, *n,* is available in 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, and 32-bit.

M*n*_S1S8E and M*n*_S1S8E_SB are pin-to-pin versions of 1-to-8 demultiplexers, which switch 1 x *n*-single pins to 8 x *n*-single pins according to the select inputs. The number of single pin, *n,* is available in 1, and 2.

M1_S1B8E and M1_S1B8E_SB are bus-to-pin versions of the 1-to-8 demultiplexer, which switches a single pin to 1-bit of the 8-bit bus according to the select inputs.

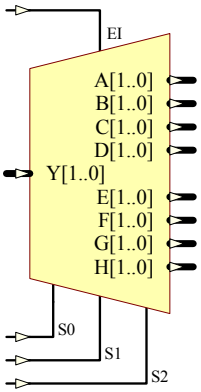
Enable is the highest priority input, when enable is Low, all inputs are ignored, and outputs remain Low. When enable is High, the demultiplexers switch the data from inputs to output.

The demultiplexers with the "_SB" suffix in the name have the Select input pins (S2-S0) grouped into a single bus pin (S[2..0]), whereas those demultiplexers without this suffix leave the Select input pins ungrouped.

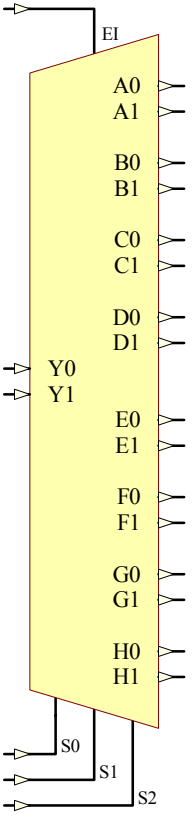
Enable	Select Inputs			Input	Outputs							
E	S2	S1	S0	O	D0	D1	D2	D3	D4	D5	D6	D7
EI				Y	A	B	C	D	E	F	G	H
0	x	x	x	x	0	0	0	0	0	0	0	0
1	0	0	0	d	d	0	0	0	0	0	0	0
1	0	0	1	d	0	d	0	0	0	0	0	0
1	0	1	0	d	0	0	d	0	0	0	0	0
1	0	1	1	d	0	0	0	d	0	0	0	0
1	1	0	0	d	0	0	0	0	d	0	0	0
1	1	0	1	d	0	0	0	0	0	d	0	0
1	1	1	0	d	0	0	0	0	0	0	d	0
1	1	1	1	d	0	0	0	0	0	0	0	d

For M1 follow E, O, D0, D1, D3, D4, D5, D6, D7

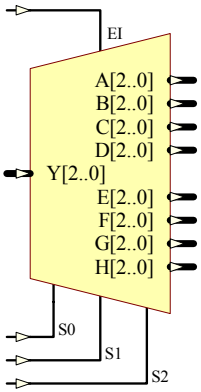
For Mn follow EI, Y, A, B, C, D, E, F, G, H



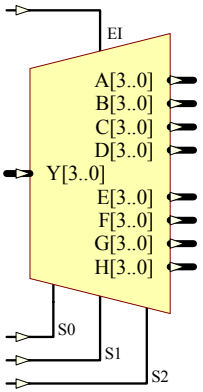
M2_B1B8E



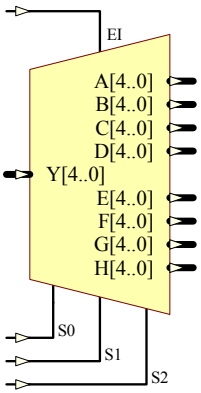
M2_S1S8E



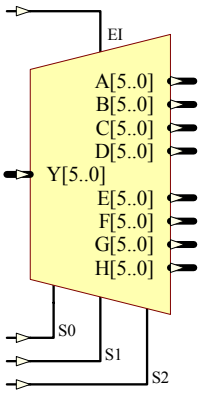
M3_B1B8E



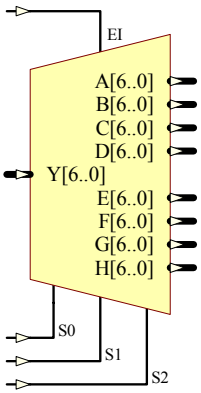
M4_B1B8E



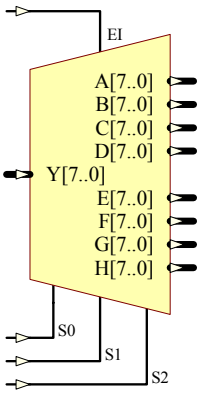
M5_B1B8E



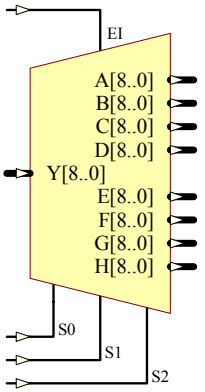
M6_B1B8E



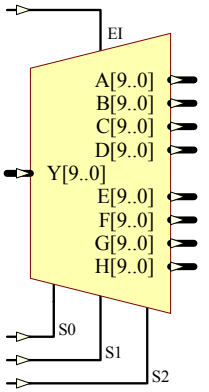
M7_B1B8E



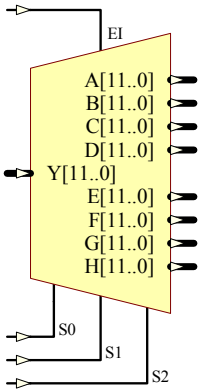
M8_B1B8E



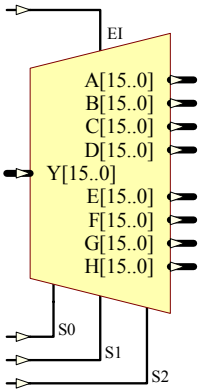
M9_B1B8E



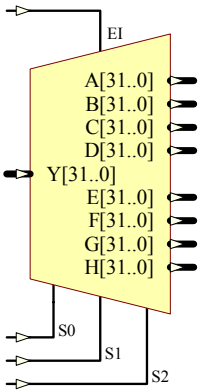
M10_B1B8E



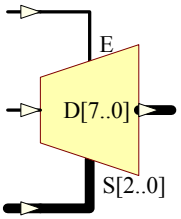
M12_B1B8E



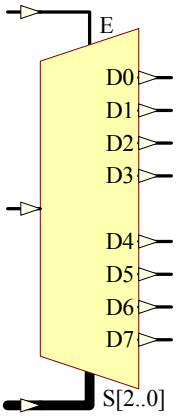
M16_B1B8E



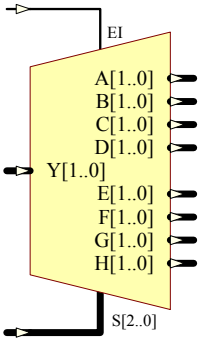
M32_B1B8E



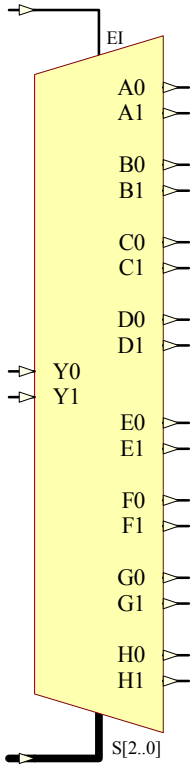
M1_S1B8E_SB



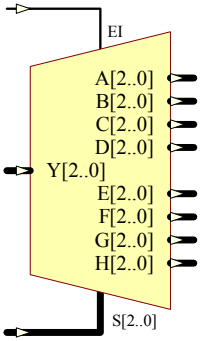
M1_S1S8E_SB



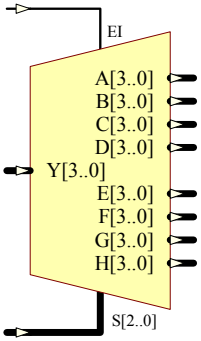
M2_B1B8E_SB



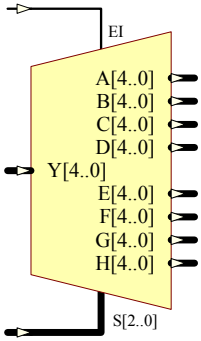
M2_S1S8E_SB



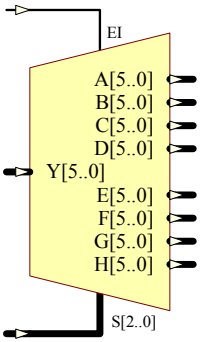
M3_B1B8E_SB



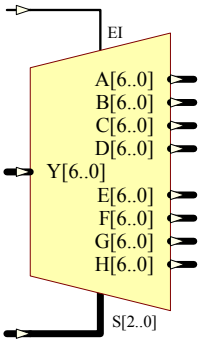
M4_B1B8E_SB



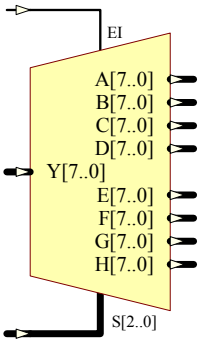
M5_B1B8E_SB



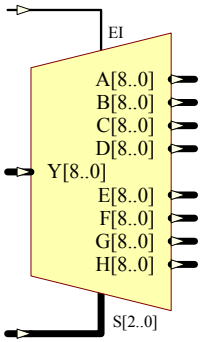
M6_B1B8E_SB



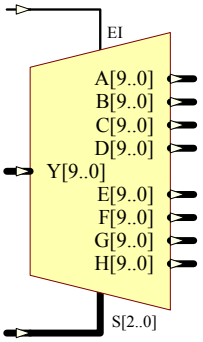
M7_B1B8E_SB



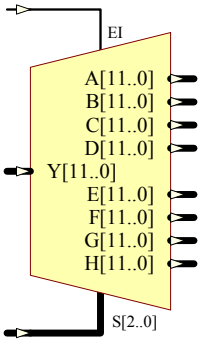
M8_B1B8E_SB



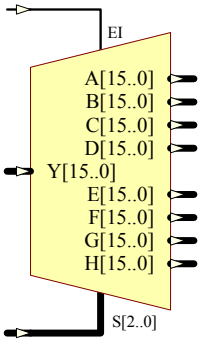
M9_B1B8E_SB



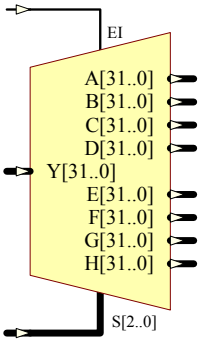
M10_B1B8E_SB



M12_B1B8E_SB



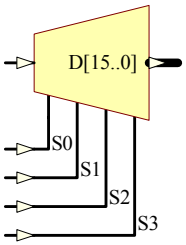
M16_B1B8E_SB



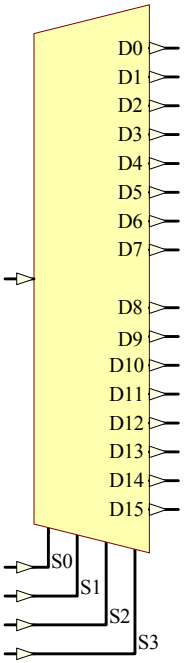
M32_B1B8E_SB

M*n*_B1B16, M1_S1S16, M1_S1B16, M*n*_B1B16_SB, M1_S1S16_SB, M1_S1B16_SB

1-to-16 Demultiplexers



M1_S1B16



M1_S1S16

Mn_B1B16, M1_S1S16, M1_S1B16, Mn_B1B16_SB, M1_S1S16_SB and M1_S1B16_SB are various *n*-bit data width 1-to-16 demultiplexers, available in bus-to-bus, pin-to-pin and pin-to-bus versions.

M*n*_B1B16 and M*n*_B1B16_SB are bus-to-bus version of 1-to-16 demultiplexers, which switch 1 x *n*-bit bus to 16 x *n*-bit bus according to the select inputs. The width of the data bus, *n* is available in 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, and 32-bit.

M1_S1S16 and M1_S1S16 _SB are a pin-to-pin version of 1-to-16 demultiplexers, which switches 1 single pin to 16 single pins according to the select inputs.

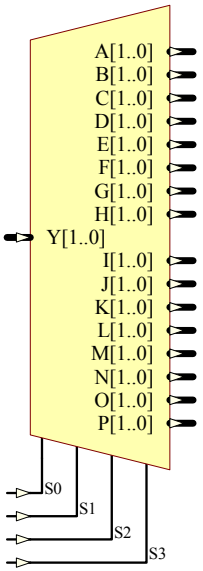
M1_S1B16 and M1_S1B16_SB are a pin-to-bus version of 1-to-16 demultiplexer, which switches a single pin to 1-bit of the 16-bit bus according to the select inputs.

Selects (S3-S0) are grouped in a bus (S[3..0]) for demultiplexer with "_SB" suffix in the name, otherwise are separated pins.

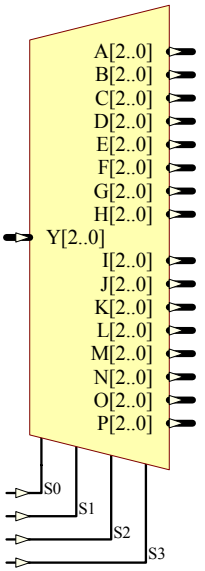
Select Inputs				Input	Outputs															
S3	S2	S1	S0	O	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15
				Y	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
0	0	0	0	d	d	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	d	0	d	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	d	0	0	d	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	d	0	0	0	d	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	d	0	0	0	0	d	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	d	0	0	0	0	0	d	0	0	0	0	0	0	0	0	0	0
0	1	1	0	d	0	0	0	0	0	0	d	0	0	0	0	0	0	0	0	0
0	1	1	1	d	0	0	0	0	0	0	0	d	0	0	0	0	0	0	0	0
1	0	0	0	d	0	0	0	0	0	0	0	0	d	0	0	0	0	0	0	0
1	0	0	1	d	0	0	0	0	0	0	0	0	0	d	0	0	0	0	0	0
1	0	1	0	d	0	0	0	0	0	0	0	0	0	0	d	0	0	0	0	0
1	0	1	1	d	0	0	0	0	0	0	0	0	0	0	0	d	0	0	0	0
1	1	0	0	d	0	0	0	0	0	0	0	0	0	0	0	0	d	0	0	0
1	1	0	1	d	0	0	0	0	0	0	0	0	0	0	0	0	0	d	0	0
1	1	1	0	d	0	0	0	0	0	0	0	0	0	0	0	0	0	0	d	0
1	1	1	1	d	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	d

For M1 follow O, D0, D1, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12, D13, D14, D15

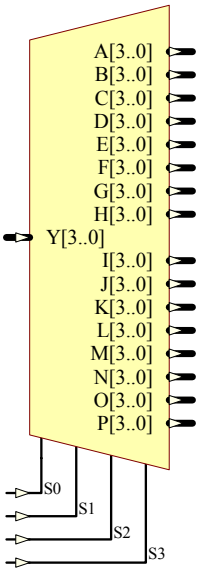
For Mn follow Y, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P



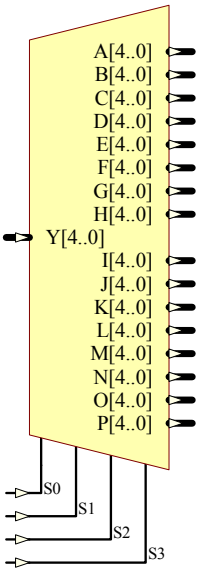
M2_B1B16



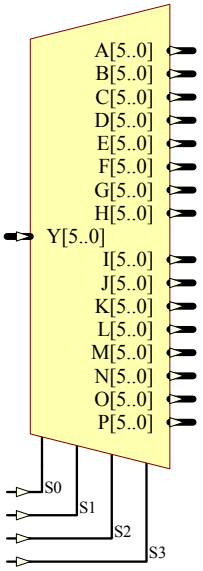
M3_B1B16



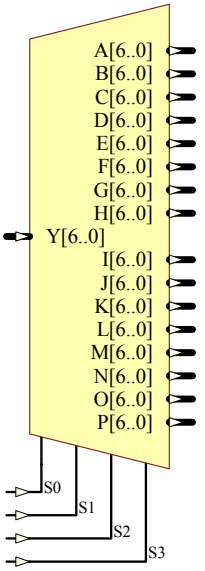
M4_B1B16



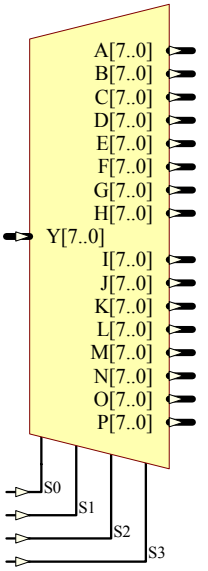
M5_B1B16



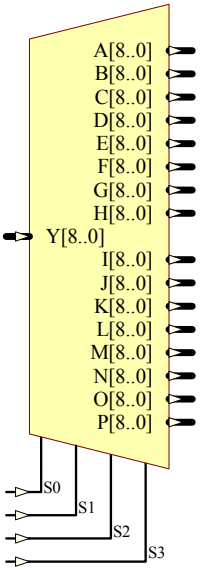
M6_B1B16



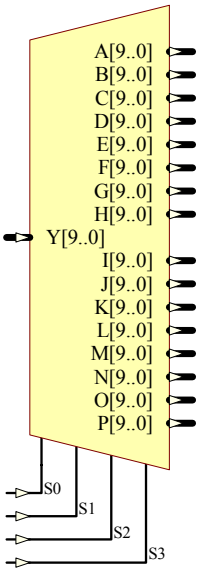
M7_B1B16



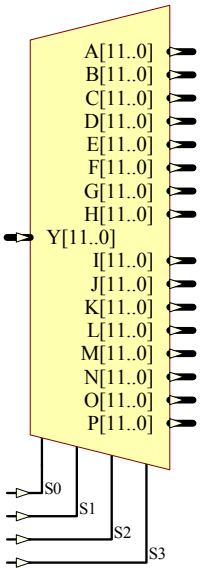
M8_B1B16



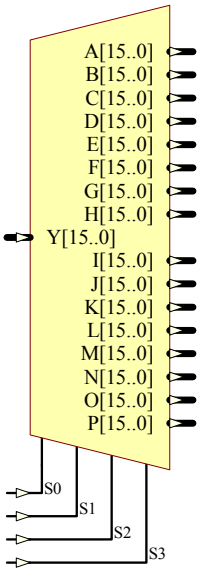
M9_B1B16



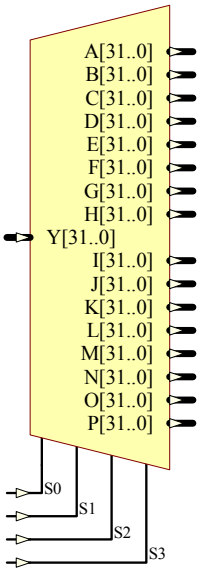
M10_B1B16



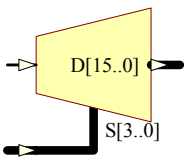
M12_B1B16



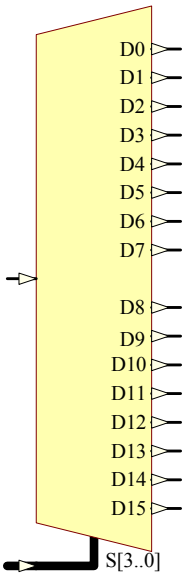
M16_B1B16



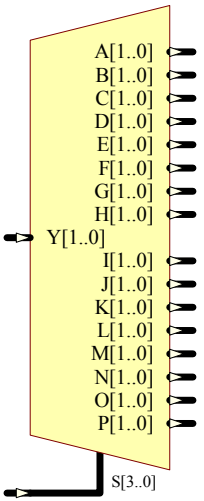
M32_B1B16



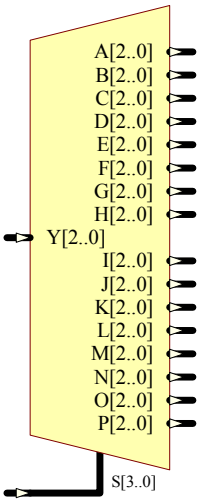
M1_S1B16_SB



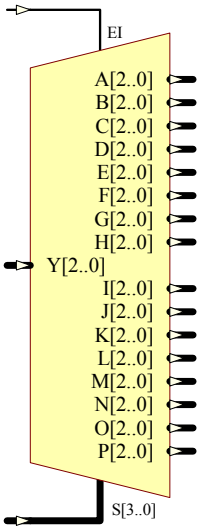
M1_S1S16_SB



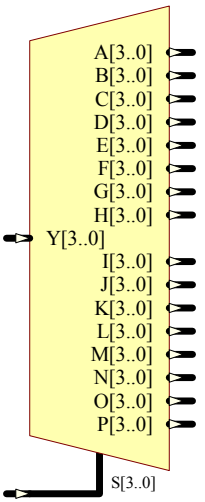
M2_B1B16_SB



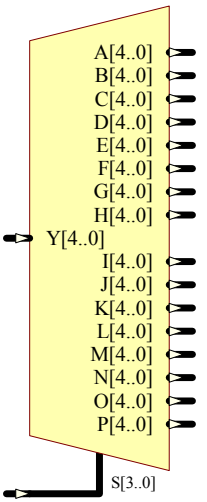
M3_B1B16_SB



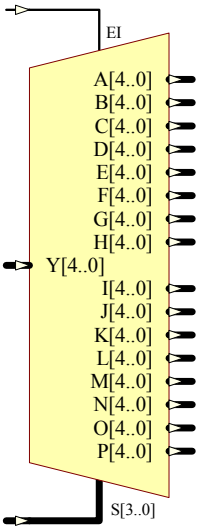
M3_B1B16E_SB



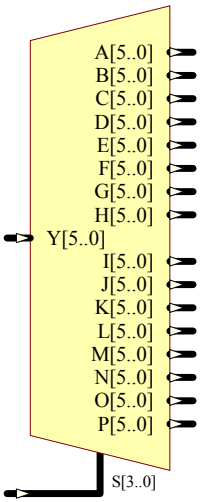
M4_B1B16_SB



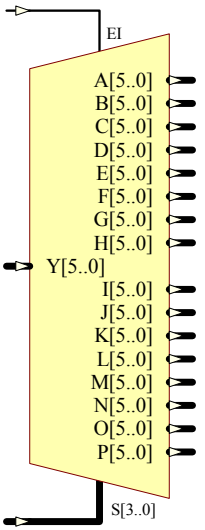
M5_B1B16_SB



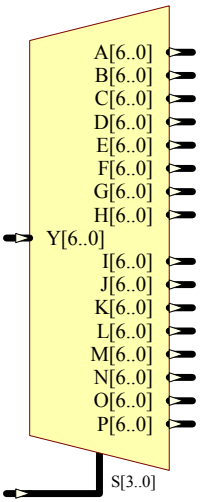
M5_B1B16E_SB



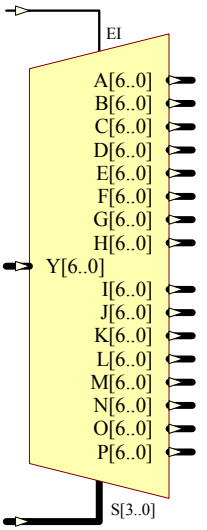
M6_B1B16_SB



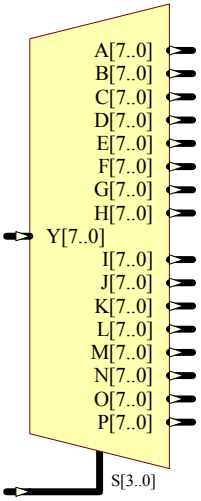
M6_B1B16E_SB



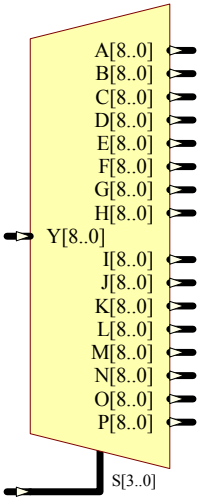
M7_B1B16_SB



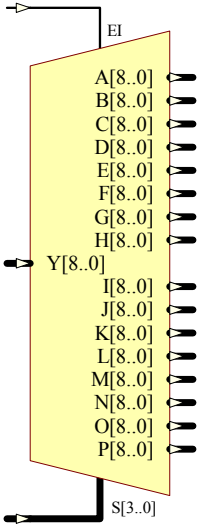
M7_B1B16E_SB



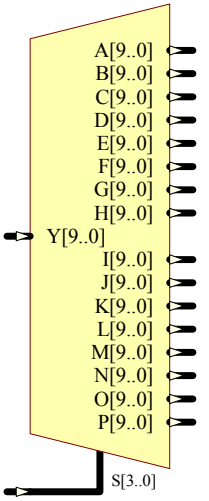
M8_B1B16_SB



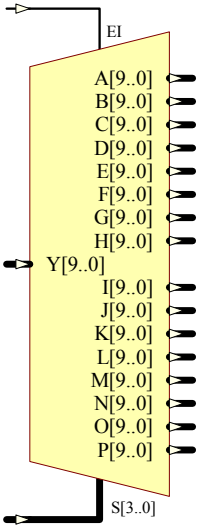
M9_B1B16_SB



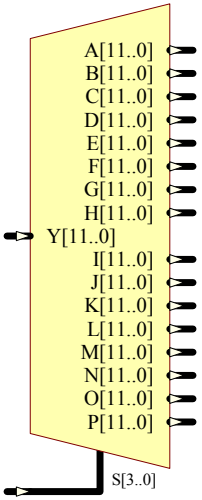
M9_B1B16E_SB



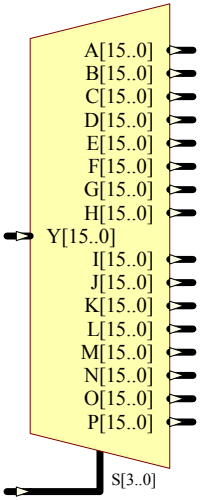
M10_B1B16_SB



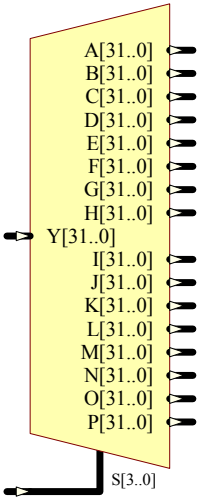
M10_B1B16E_SB



M12_B1B16_SB



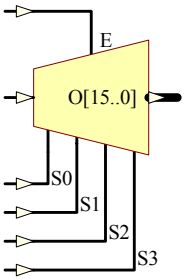
M16_B1B16_SB



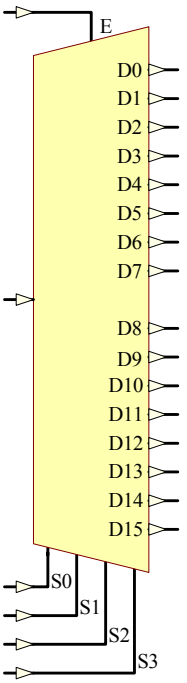
M32_B1B16_SB

M*n*_B1B16E, M1_S1S16E, M1_S1B16E, M*n*_B1B16E_SB, M1_S1S16E_SB, M1_S1B16E_SB

1-to-16 Demultiplexers with Enable



M1_S1B16E



M1_S1S16E

Mn_B1B16E, M1_S1S16E, M1_S1B16E, Mn_B1B16E_SB, M1_S1S16E_SB and M1_S1B16E_SB are various *n*-bit data width 1-to-16 demultiplexers with enable, available in bus-to-bus, pin-to-pin and pin-to-bus versions.

M*n*_B1B16E and Mn_B1B16E_SB are bus-to-bus version of 1-to-16 demultiplexers, which switch 1 x *n*-bit bus to 16 x *n*-bit bus according to the select inputs. The width of the data bus, *n* is available in 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, and 32-bit.

M1_S1S16E and M1_S1S16E_SB are a pin-to-pin version of 1-to-16 demultiplexers, which switches 1 single pin to 16 single pins according to the select inputs.

M1_S1B16E and M1_S1B16E_SB are a pin-to-bus version of 1-to-16 demultiplexer, which switches a single pin to 1-bit of the 16-bit bus according to the select inputs.

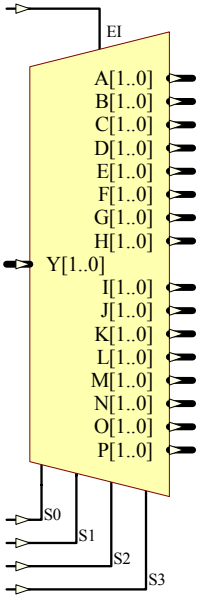
Enable is the highest priority input, when enable is Low, all inputs are ignored, and outputs remain Low. When enable is High, the demultiplexers switch the data from inputs to output.

Selects (S3-S0) are grouped in a bus (S[3..0]) for demultiplexers with "_SB" suffix in the name, otherwise are separated pins.

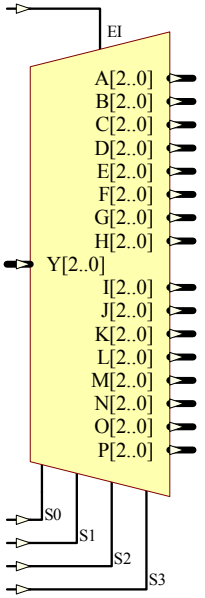
Enable	Select Inputs				Input	Outputs														
E	S3	S2	S1	S0	O	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15
EI					Y	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
0	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	d	d	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	d	0	d	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	d	0	0	d	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	1	d	0	0	0	d	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	d	0	0	0	0	d	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	1	d	0	0	0	0	0	d	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	d	0	0	0	0	0	0	d	0	0	0	0	0	0	0	0	0
1	0	1	1	1	d	0	0	0	0	0	0	0	d	0	0	0	0	0	0	0	0
1	1	0	0	0	d	0	0	0	0	0	0	0	0	d	0	0	0	0	0	0	0
1	1	0	0	1	d	0	0	0	0	0	0	0	0	0	d	0	0	0	0	0	0
1	1	0	1	0	d	0	0	0	0	0	0	0	0	0	0	d	0	0	0	0	0
1	1	0	1	1	d	0	0	0	0	0	0	0	0	0	0	0	d	0	0	0	0
1	1	1	0	0	d	0	0	0	0	0	0	0	0	0	0	0	0	d	0	0	0
1	1	1	0	1	d	0	0	0	0	0	0	0	0	0	0	0	0	0	d	0	0
1	1	1	1	0	d	0	0	0	0	0	0	0	0	0	0	0	0	0	0	d	0
1	1	1	1	1	d	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	d

For M1 follow E, O, D0, D1, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12, D13, D14, D15

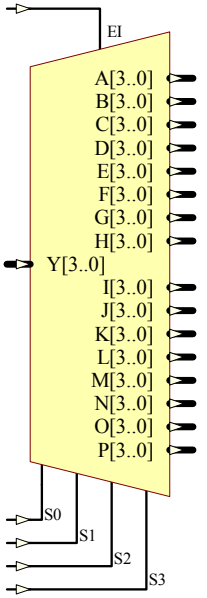
For Mn follow EI, Y, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P



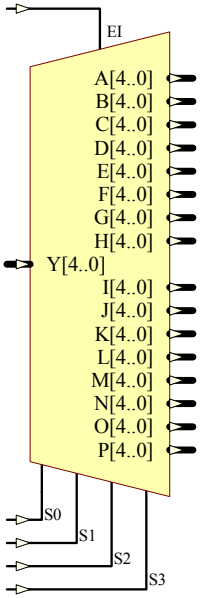
M2_B1B16E



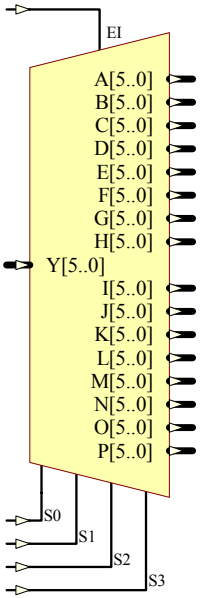
M3_B1B16E



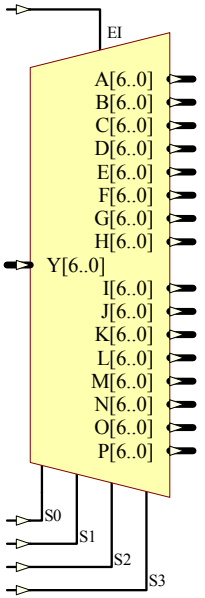
M4_B1B16E



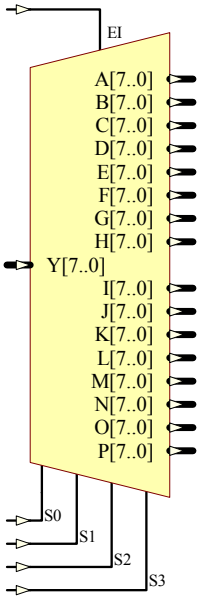
M5_B1B16E



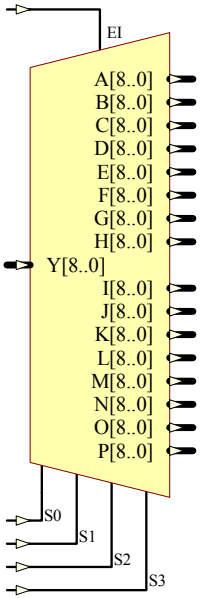
M6_B1B16E



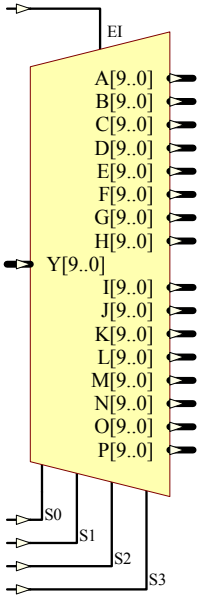
M7_B1B16E



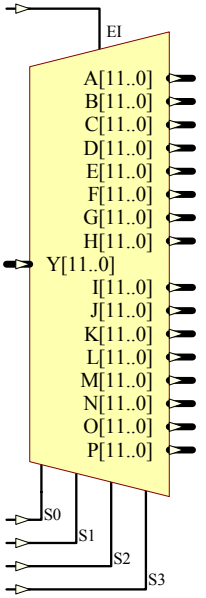
M8_B1B16E



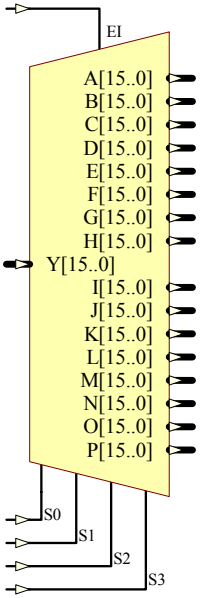
M9_B1B16E



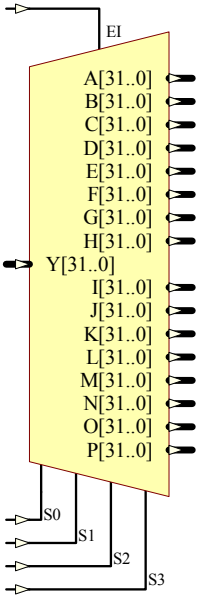
M10_B1B16E



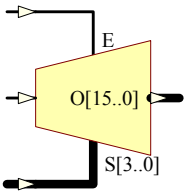
M12_B1B16E



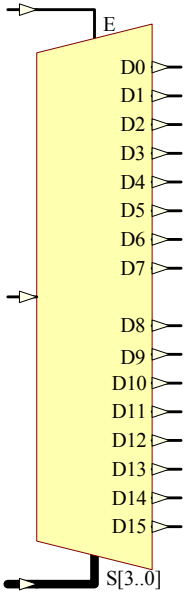
M16_B1B16E



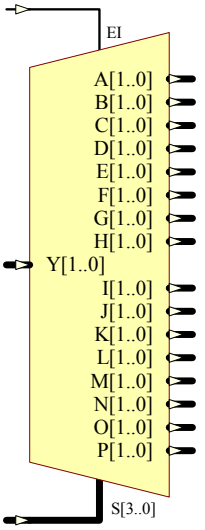
M32_B1B16E



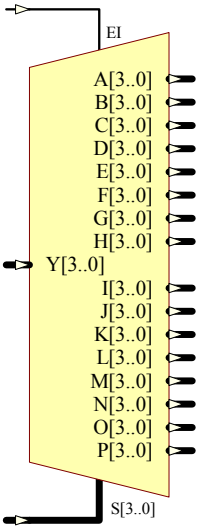
M1_S1B16E_SB



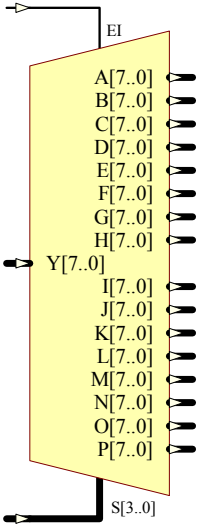
M1_S1S16E_SB



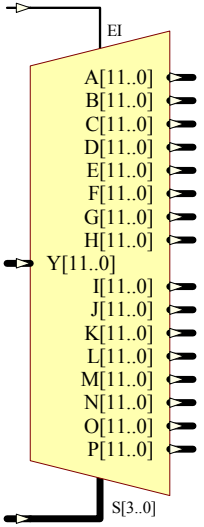
M2_B1B16E_SB



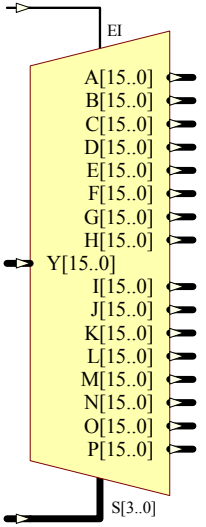
M4_B1B16E_SB



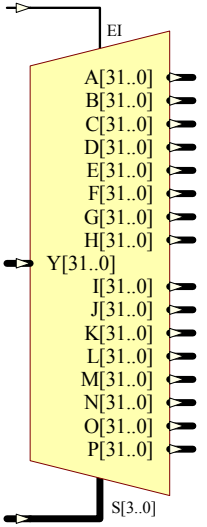
M8_B1B16E_SB



M12_B1B16E_SB



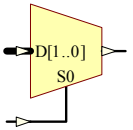
M16_B1B16E_SB



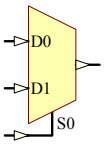
M32_B1B16E_SB

M*n*_B2B1, M*n*_S2S1, M1_B2S1

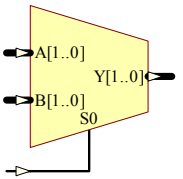
2-to-1 Multiplexers



M1_B2S1



M1_S2S1



M2_B2B1

M*n*_B2B1, M*n*_S2S1, M1_B2S1 are various *n*-bit data width 2-to-1 multiplexers, available in bus-to-bus, pin-to-pin and bus-to-pin versions.

M*n*_B2B1 are bus-to-bus version of 2-to-1 multiplexers, which switch 2 x *n*-bit bus to 1 x *n*-bit bus according to the select input. The width of the data bus, *n* is available in 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, and 32-bit.

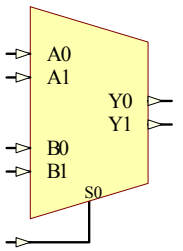
M*n*_S2S1 are pin-to-pin version of 2-to-1 multiplexers, which switch 2 x *n*-single pins to 1 x *n*-single pins according to the select input. The number of single pin, *n* is available in 1, 2, 3, 4, 5, 6, 7, and 8.

M1_B2S1 is bus-to-pin version of 2-to-1 multiplexer, which switches 1-bit of the 2-bit bus to 1-single pin according to the select input.

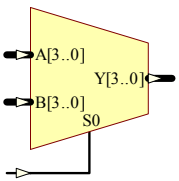
Select	Data Inputs		Outputs
S0	D1	D0	O
	B	A	Y
0	x	d0	d0
1	d1	x	d1

For M1 follow D0, D1, O

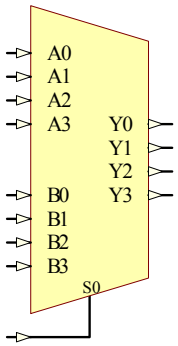
For Mn follow A, B, Y



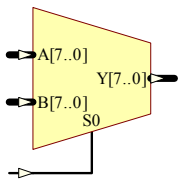
M2_S2S1



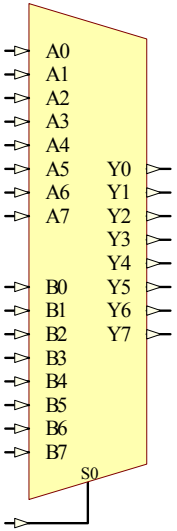
M4_B2B1



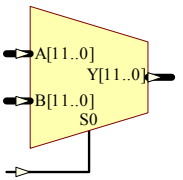
M4_S2S1



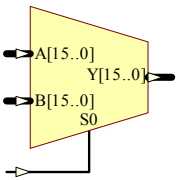
M8_B2B1



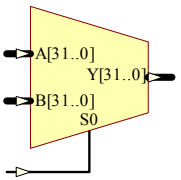
M8_S2S1



M12_B2B1



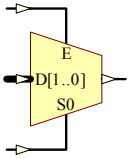
M16_B2B1



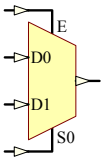
M32_B2B1

M*n*_B2B1E, M*n*_S2S1E, M1_B2S1E

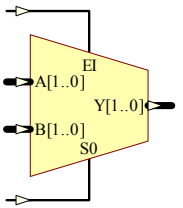
2-to-1 Multiplexers with Enable



M1_B2S1E



M1_S2S1E



M2_B2B1E

M*n*_B2B1E, M*n*_S2S1E, M1_B2S1E are various *n*-bit data width 2-to-1 multiplexers with enable, available in bus-to-bus, pin-to-pin and bus-to-pin versions.

M*n*_B2B1E are bus-to-bus version of 2-to-1 multiplexers, which switch 2 x *n*-bit bus to 1 x *n*-bit bus according to the select input. The width of the data bus, *n* is available in 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, and 32-bit.

M*n*_S2S1E are pin-to-pin version of 2-to-1 multiplexers, which switch 2 x *n*-single pins to 1 x *n*-single pins according to the select input. The number of single pin, *n* is available in 1, 2, 3, 4, 5, 6, 7, and 8.

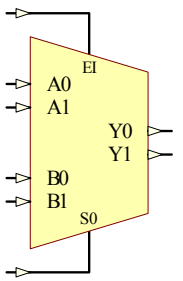
M1_B2S1 is bus-to-pin version of 2-to-1 multiplexer, which switches 1-bit of the 2-bit bus to 1-single pin according to the select input.

Enable is the highest priority input, when enable is Low, all inputs are ignored, and outputs remain Low. When enable is High, the multiplexers switch the data from inputs to output.

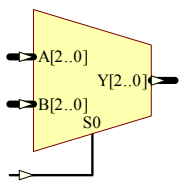
Enable	Select	Data Inputs		Outputs
E	S0	D1	D0	O
EI		B	A	Y
0	x	x	x	0
1	0	x	d0	d0
1	1	d1	x	d1

For M1 follow E, D0, D1, O

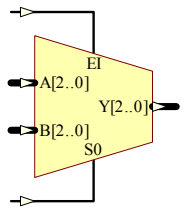
For Mn follow EI, A, B, Y



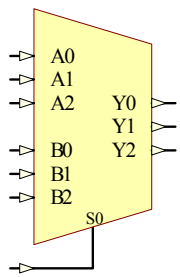
M2_S2S1E



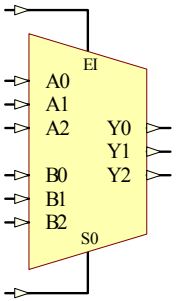
M3_B2B1



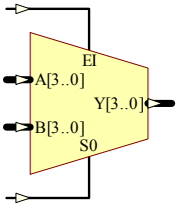
M3_B2B1E



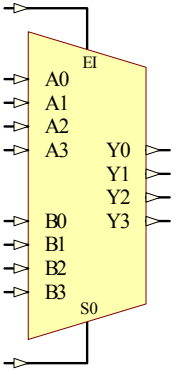
M3_S2S1



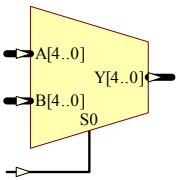
M3_S2S1E



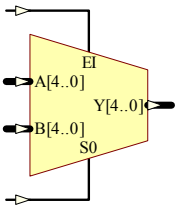
M4_B2B1E



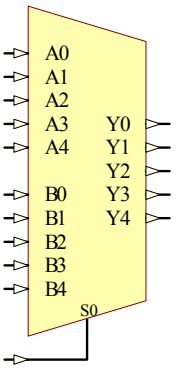
M4_S2S1E



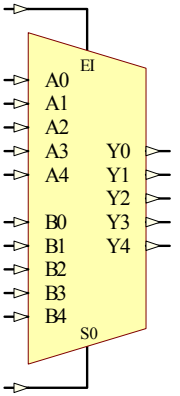
M5_B2B1



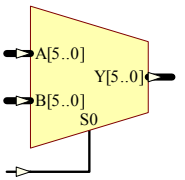
M5_B2B1E



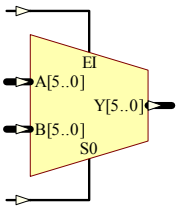
M5_S2S1



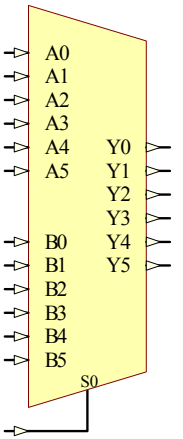
M5_S2S1E



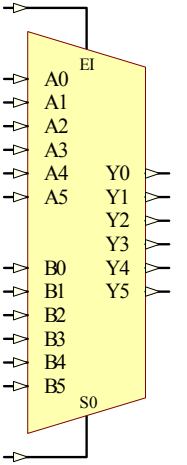
M6_B2B1



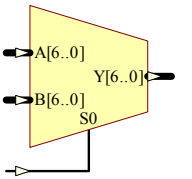
M6_B2B1E



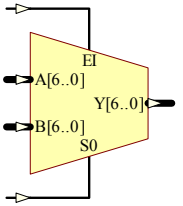
M6_S2S1



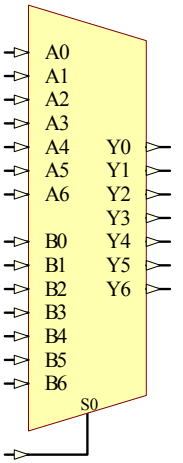
M6_S2S1E



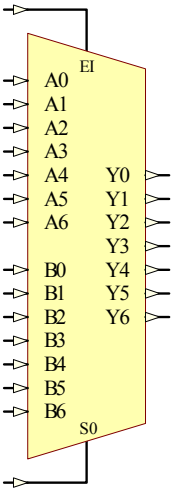
M7_B2B1



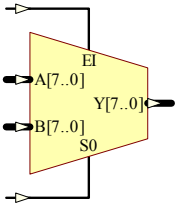
M7_B2B1E



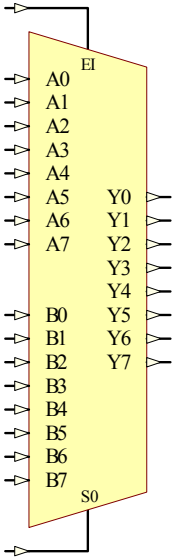
M7_S2S1



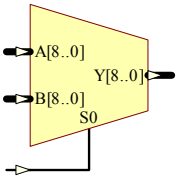
M7_S2S1E



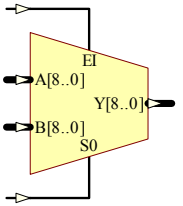
M8_B2B1E



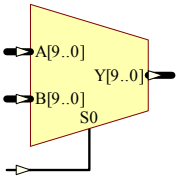
M8_S2S1E



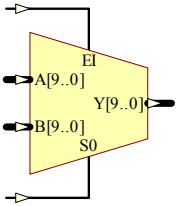
M9_B2B1



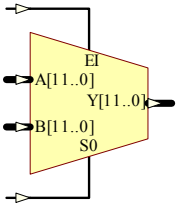
M9_B2B1E



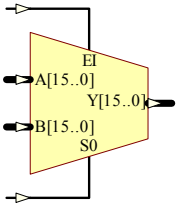
M10_B2B1



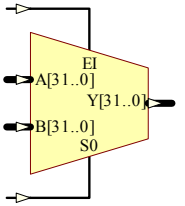
M10_B2B1E



M12_B2B1E



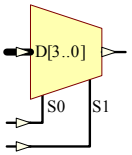
M16_B2B1E



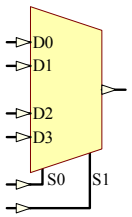
M32_B2B1E

M*n*_B4B1, M*n*_S4S1, M1_B4S1, M*n*_B4B1_SB, M*n*_S4S1_SB, M1_B4S1_SB

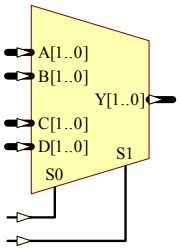
4-to-1 Multiplexers



M1_B4S1



M1_S4S1



M2_B4B1

M*n*_B4B1, M*n*_S4S1, M1_B4S1, M*n*_B4B1_SB, M*n*_S4S1_SB and M1_B4S1_SB are various *n*-bit data width 4-to-1 multiplexers, available in bus-to-bus, pin-to-pin and bus-to-pin versions.

M*n*_B4B1 and M*n*_B4B1_SB are bus-to-bus versions of the 4-to-1 multiplexers, which switch a 4 x *n*-bit bus to a 1 x *n*-bit bus according to the select inputs. The width of the data bus, *n,* is available in 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, and 32-bit.

M*n*_S4S1 and M*n*_S4S1_SB are pin-to-pin versions of the 4-to-1 multiplexers, which switch 4 x *n*-single pins to 1 x *n*-single pins according to the select inputs. The number of single pin, *n,* is available in 1, 2, 3, and 4.

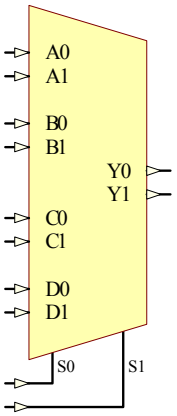
M1_B4S1 and M1_B4S1_SB are bus-to-pin versions of the 4-to-1 multiplexer, which switches 1-bit of the 4-bit bus to a 1-single pin according to the select inputs.

The multiplexers with the "_SB" suffix in the name have the Select input pins (S1-S0) grouped into a single bus pin (S[1..0]), whereas those multiplexers without this suffix leave the Select input pins ungrouped.

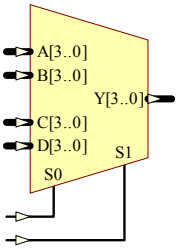
Select Inputs		Data Inputs				Outputs
S1	S0	D3	D2	D1	D0	O
		D	C	B	A	Y
0	0	x	x	x	d0	d0
0	1	x	x	d1	x	d1
1	0	x	d2	x	x	d2
1	1	d3	x	x	x	d3

For M1 follow D0, D1, D3, O

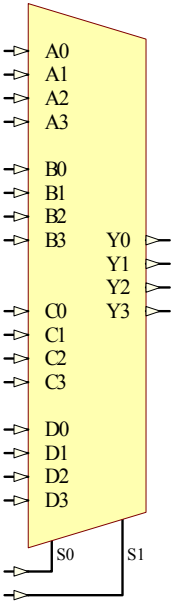
For Mn follow A, B, C, D, Y



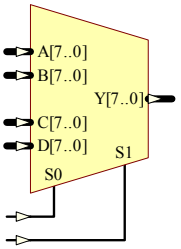
M2_S4S1



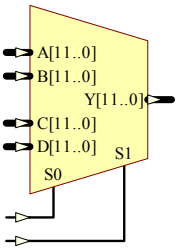
M4_B4B1



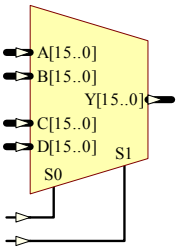
M4_S4S1



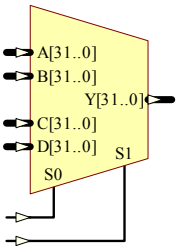
M8_B4B1



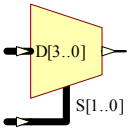
M12_B4B1



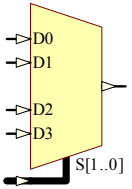
M16_B4B1



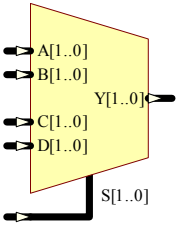
M32_B4B1



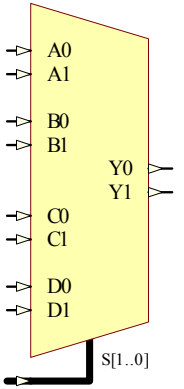
M1_B4S1_SB



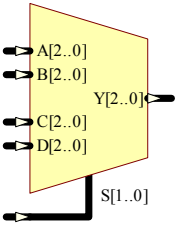
M1_S4S1_SB



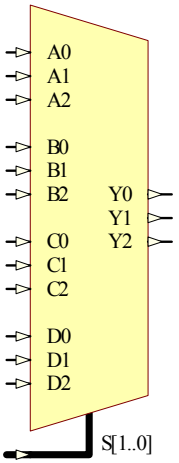
M2_B4B1_SB



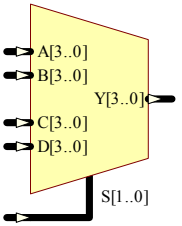
M2_S4S1_SB



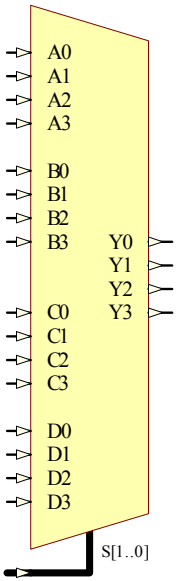
M3_B4B1_SB



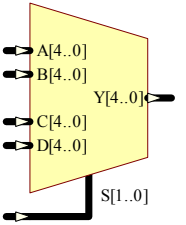
M3_S4S1_SB



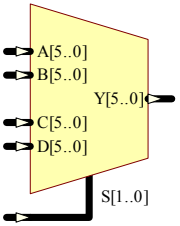
M4_B4B1_SB



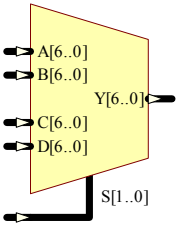
M4_S4S1_SB



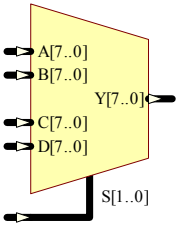
M5_B4B1_SB



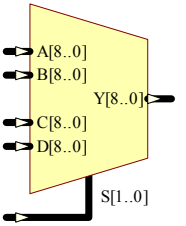
M6_B4B1_SB



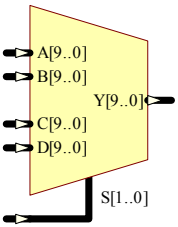
M7_B4B1_SB



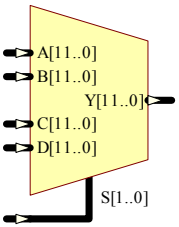
M8_B4B1_SB



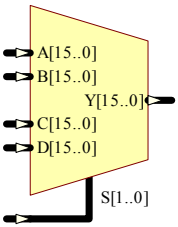
M9_B4B1_SB



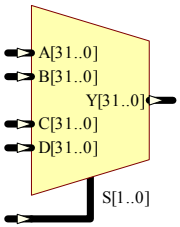
M10_B4B1_SB



M12_B4B1_SB



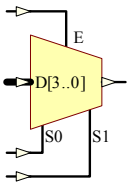
M16_B4B1_SB



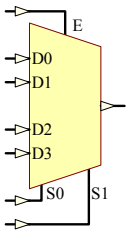
M32_B4B1_SB

M*n*_B4B1E, M*n*_S4S1E, M1_B4S1E, M*n*_B4B1E_SB, M*n*_S4S1E_SB, M1_B4S1E_SB

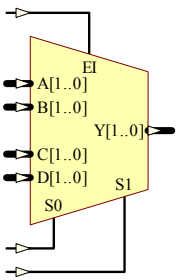
4-to-1 Multiplexers with Enable



M1_B4S1E



M1_S4S1E



M2_B4B1E

M*n*_B4B1E, M*n*_S4S1E, M1_B4S1E, M*n*_B4B1E_SB, M*n*_S4S1E_SB and M1_B4S1E_SB are various *n*-bit data width 4-to-1 multiplexers with enable, available in bus-to-bus, pin-to-pin and bus-to-pin versions.

M*n*_B4B1E and M*n*_B4B1E_SB are bus-to-bus versions of the 4-to-1 multiplexers, which switch a 4 x *n*-bit bus to a 1 x *n*-bit bus. The width of the data bus, *n,* is available in 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, and 32-bit.

M*n*_S4S1E and M*n*_S4S1E_SB are pin-to-pin versions of the 4-to-1 multiplexers, which switch 4 x *n*-single pins to 1 x *n*-single pins. The number of single pin, *n,* is available in 1, 2, 3, and 4.

M1_B4S1E and M1_B4S1E_SB are bus-to-pin versions of the 4-to-1 multiplexer, which switches 1-bit of the 4-bit bus to 1-single pin.

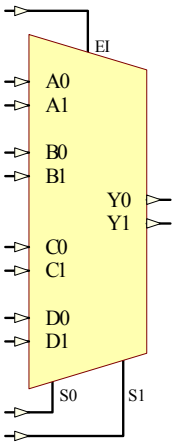
Enable is the highest priority input, when enable is Low, all inputs are ignored, and outputs remain Low. When enable is High, the multiplexers switch the data from inputs to output.

The multiplexers with the "_SB" suffix in the name have the Select input pins (S1-S0) grouped into a single bus pin (S[1..0]), whereas those multiplexers without this suffix leave the Select input pins ungrouped.

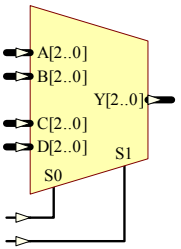
Enable	Select Inputs		Data Inputs				Outputs
E	S1	S0	D3	D2	D1	D0	O
EI			D	C	B	A	Y
0	x	x	x	x	x	x	0
1	0	0	x	x	x	a	a
1	0	1	x	x	b	x	b
1	1	0	x	c	x	x	c
1	1	1	d	x	x	x	d

For M1 follow E, D0, D1, D3, O

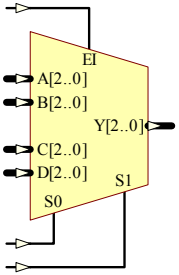
For Mn follow EI, A, B, C, D, Y



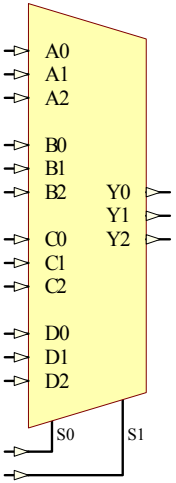
M2_S4S1E



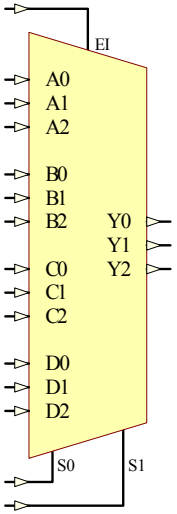
M3_B4B1



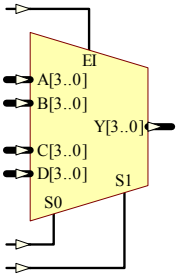
M3_B4B1E



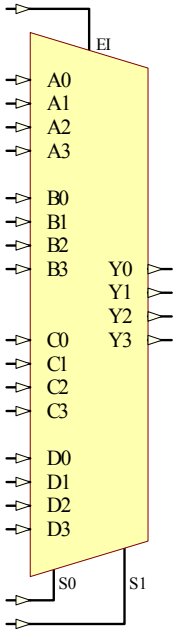
M3_S4S1



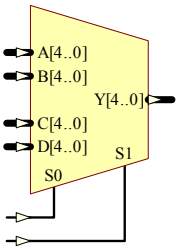
M3_S4S1E



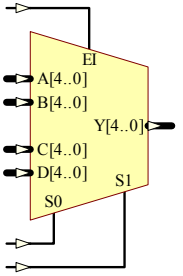
M4_B4B1E



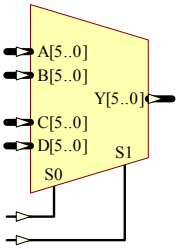
M4_S4S1E



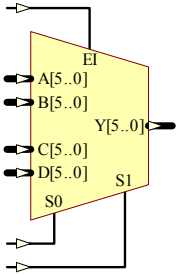
M5_B4B1



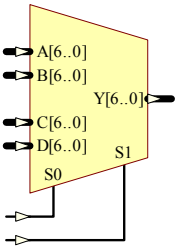
M5_B4B1E



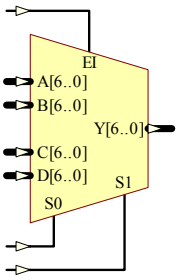
M6_B4B1



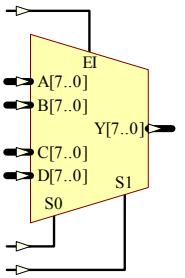
M6_B4B1E



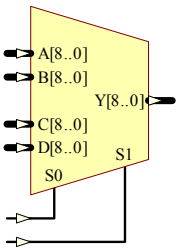
M7_B4B1



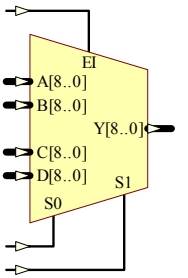
M7_B4B1E



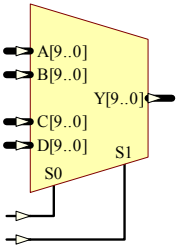
M8_B4B1E



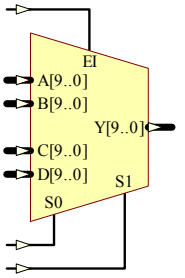
M9_B4B1



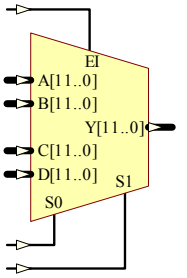
M9_B4B1E



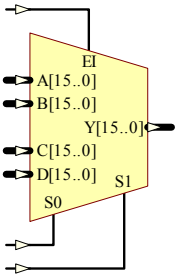
M10_B4B1



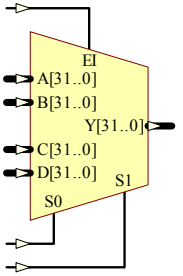
M10_B4B1E



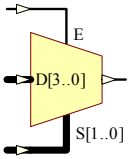
M12_B4B1E



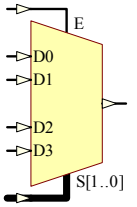
M16_B4B1E



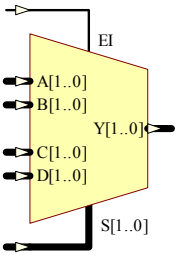
M32_B4B1E



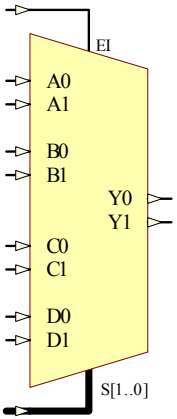
M1_B4S1E_SB



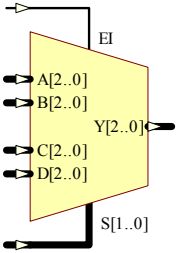
M1_S4S1E_SB



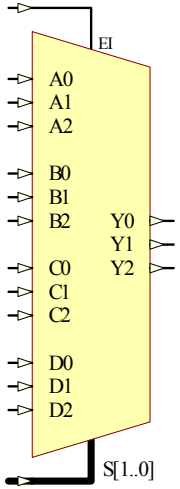
M2_B4B1E_SB



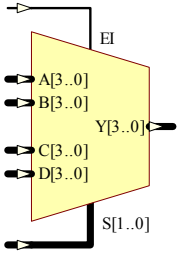
M2_S4S1E_SB



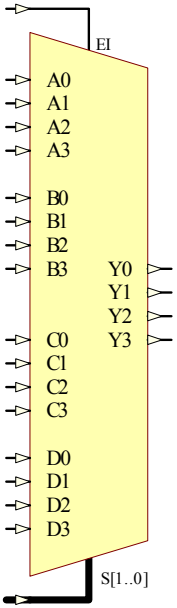
M3_B4B1E_SB



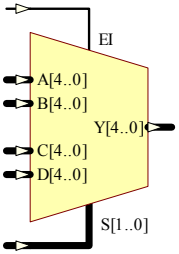
M3_S4S1E_SB



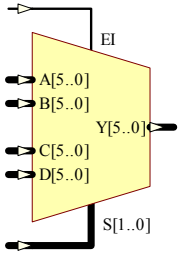
M4_B4B1E_SB



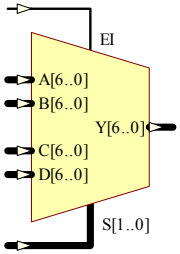
M4_S4S1E_SB



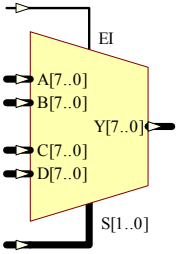
M5_B4B1E_SB



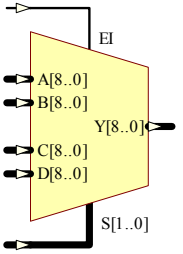
M6_B4B1E_SB



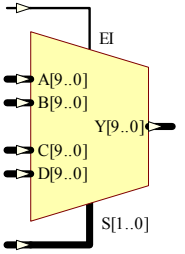
M7_B4B1E_SB



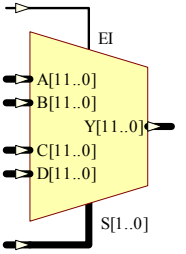
M8_B4B1E_SB



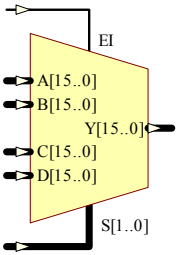
M9_B4B1E_SB



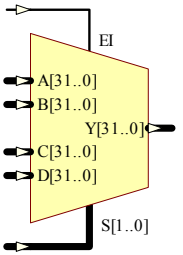
M10_B4B1E_SB



M12_B4B1E_SB



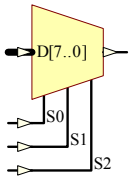
M16_B4B1E_SB



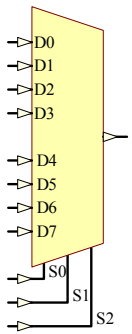
M32_B4B1E_SB

M*n*_B8B1, M*n*_S8S1, M1_B8S1, M*n*_B8B1_SB, M*n*_S8S1_SB, M1_B8S1_SB

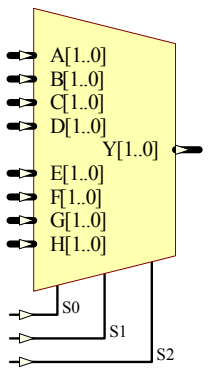
8-to-1 Multiplexers



M1_B8S1



M1_S8S1



M2_B8B1

M*n*_B8B1, M*n*_S8S1, M1_B8S1, M*n*_B8B1_SB, M*n*_S8S1_SB and M1_B8S1_SB are various *n*-bit data width 8-to-1 multiplexers, available in bus-to-bus, pin-to-pin and bus-to-pin versions.

M*n*_B8B1 and M*n*_B8B1_SB are bus-to-bus versions of the 8-to-1 multiplexers, which switch an 8 x *n*-bit bus to a 1 x *n*-bit bus according to the select inputs. The width of the data bus, *n,* is available in 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, and 32-bit.

M*n*_S8S1 and M*n*_S8S1_SB are pin-to-pin versions of the 8-to-1 multiplexers, which switch 8 x *n*-single pins to 1 x *n*-single pins according to the select inputs. The number of single pin, *n,* is available in 1, and 2.

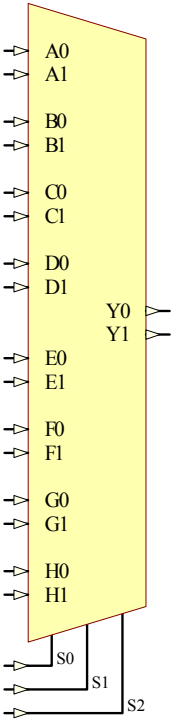
M1_B8S1 and M1_B8S1_SB are bus-to-pin versions of the 8-to-1 multiplexer, which switches 1-bit of the 8-bit bus to 1-single pin according to the select inputs.

The multiplexers with the "_SB" suffix in the name have the Select input pins (S2-S0) grouped into a single bus pin (S[2..0]), whereas those multiplexers without this suffix leave the Select input pins ungrouped.

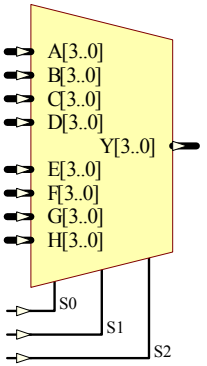
Select Inputs			Data Inputs								Outputs
S2	S1	S0	D0	D1	D2	D3	D4	D5	D6	D7	O
			A	B	C	D	E	F	G	H	Y
0	0	0	d0	x	x	x	x	x	x	x	d0
0	0	1	x	d1	x	x	x	x	x	x	d1
0	1	0	x	x	d2	x	x	x	x	x	d2
0	1	1	x	x	x	d3	x	x	x	x	d3
1	0	0	x	x	x	x	d4	x	x	x	d4
1	0	1	x	x	x	x	x	d5	x	x	d5
1	1	0	x	x	x	x	x	x	d6	x	d6
1	1	1	x	x	x	x	x	x	x	d7	d7

For M1 follow D0, D1, D3, D4, D5, D6, D7, O

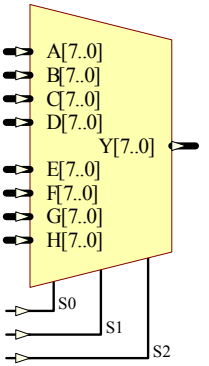
For Mn follow A, B, C, D, E, F, G, H, Y



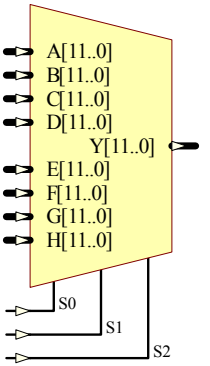
M2_S8S1



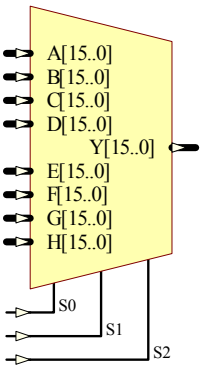
M4_B8B1



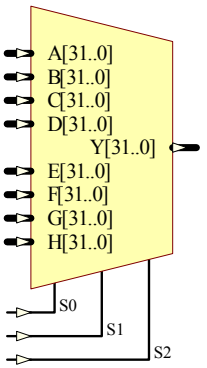
M8_B8B1



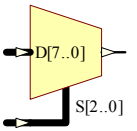
M12_B8B1



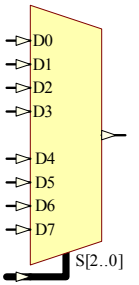
M16_B8B1



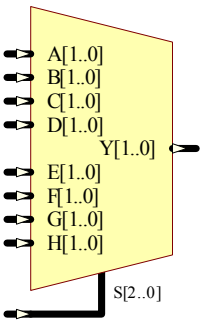
M32_B8B1



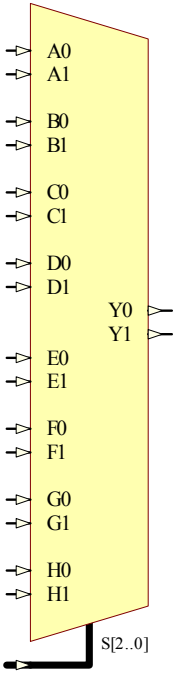
M1_B8S1_SB



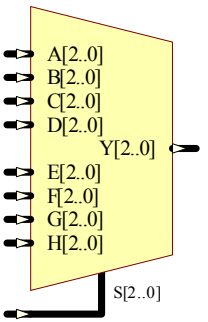
M1_S8S1_SB



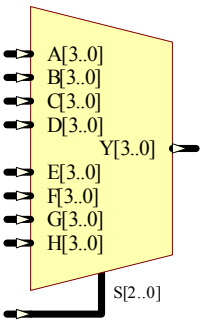
M2_B8B1_SB



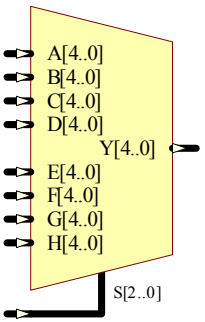
M2_S8S1_SB



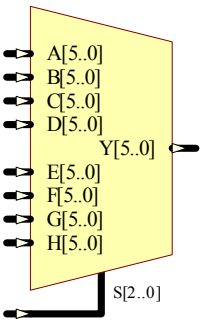
M3_B8B1_SB



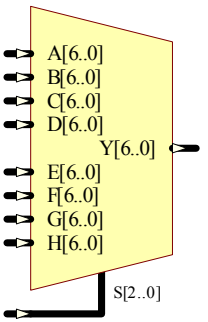
M4_B8B1_SB



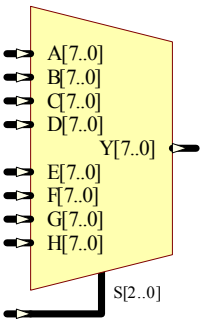
M5_B8B1_SB



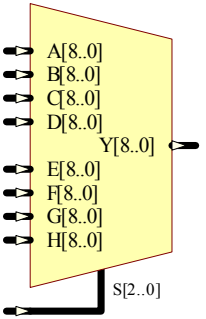
M6_B8B1_SB



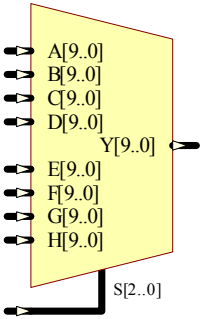
M7_B8B1_SB



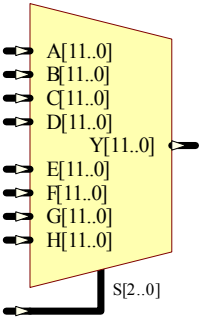
M8_B8B1_SB



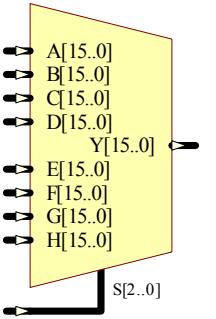
M9_B8B1_SB



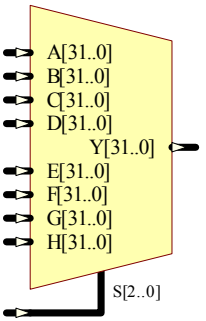
M10_B8B1_SB



M12_B8B1_SB



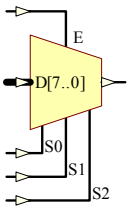
M16_B8B1_SB



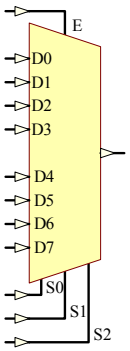
M32_B8B1_SB

M*n*_B8B1E, M*n*_S8S1E, M1_B8S1E, M*n*_B8B1E_SB, M*n*_S8S1E_SB, M1_B8S1E_SB

8-to-1 Multiplexers with Enable



M1_B8S1E



M1_S8S1E

M*n*_B8B1E, M*n*_S8S1E, M1_B8S1E, M*n*_B8B1E_SB, M*n*_S8S1E_SB, and M1_B8S1E_SB are various *n*-bit data width 8-to-1 multiplexers with enable, available in bus-to-bus, pin-to-pin and bus-to-pin versions.

M*n*_B8B1E and M*n*_B8B1E_SB are bus-to-bus versions of the 8-to-1 multiplexers, which switch an 8 x *n*-bit bus to a 1 x *n*-bit bus. The width of the data bus, *n,* is available in 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, and 32-bit.

M*n*_S8S1E and M*n*_S8S1E_SB are pin-to-pin versions of the 8-to-1 multiplexers, which switch 8 x *n*-single pins to 1 x *n*-single pins. The number of single pin, *n,* is available in 1, and 2.

M1_B8S1E and M1_B8S1E_SB are bus-to-pin versions of the 8-to-1 multiplexer, which switches 1-bit of the 8-bit bus to 1-single pin.

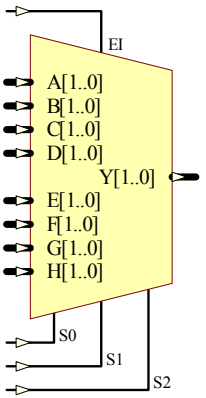
When enable is High, the multiplexers switch the data from inputs to output, when enable is Low output of the multiplexers will remain Low.

The multiplexers with the "_SB" suffix in the name have the Select input pins (S2-S0) grouped into a single bus pin (S[2..0]), whereas those multiplexers without this suffix leave the Select input pins ungrouped.

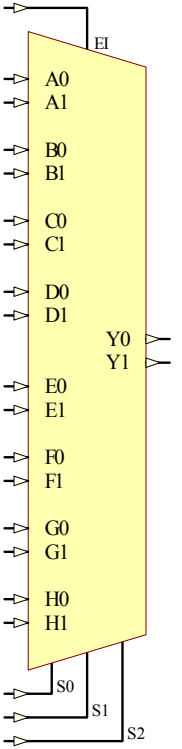
Enable	Select Inputs			Data Inputs								Outputs
E	S2	S1	S0	D0	D1	D2	D3	D4	D5	D6	D7	O
EI				A	B	C	D	E	F	G	H	Y
0	x	x	x	x	x	x	x	x	x	x	x	0
1	0	0	0	a	x	x	x	x	x	x	x	a
1	0	0	1	x	b	x	x	x	x	x	x	b
1	0	1	0	x	x	c	x	x	x	x	x	c
1	0	1	1	x	x	x	d	x	x	x	x	d
1	1	0	0	x	x	x	x	e	x	x	x	e
1	1	0	1	x	x	x	x	x	f	x	x	f
1	1	1	0	x	x	x	x	x	x	g	x	g
1	1	1	1	x	x	x	x	x	x	x	h	h

For M1 follow E, D0, D1, D3, D4, D5, D6, D7, O

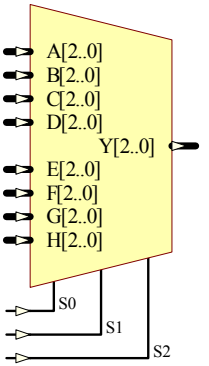
For Mn follow EI, A, B, C, D, E, F, G, H, Y



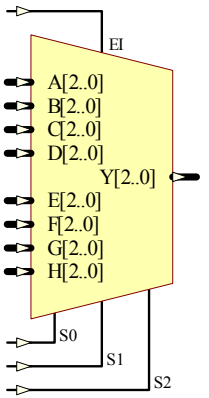
M2_B8B1E



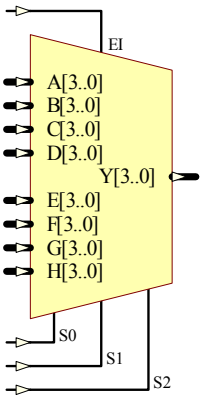
M2_S8S1E



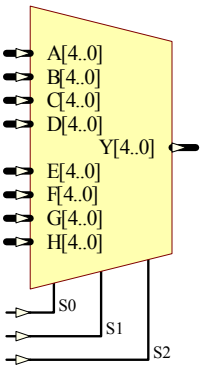
M3_B8B1



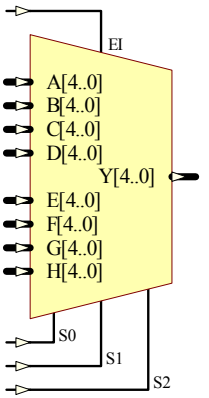
M3_B8B1E



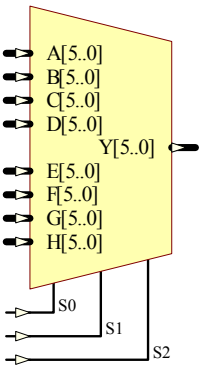
M4_B8B1E



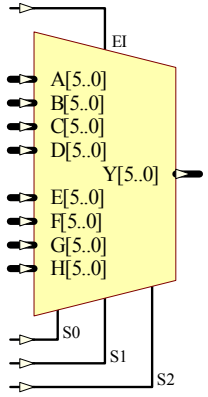
M5_B8B1



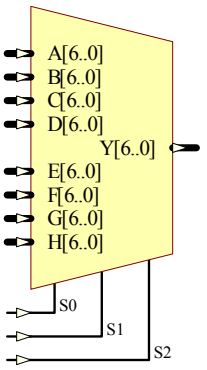
M5_B8B1E



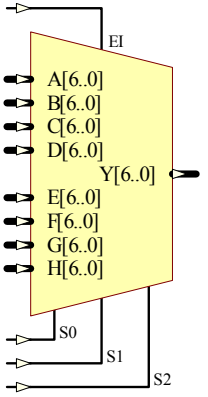
M6_B8B1



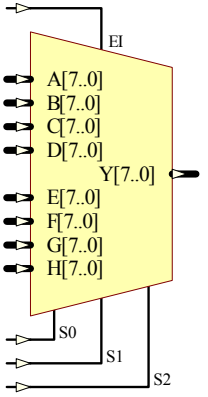
M6_B8B1E



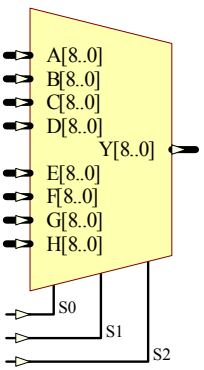
M7_B8B1



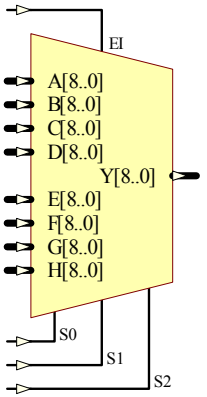
M7_B8B1E



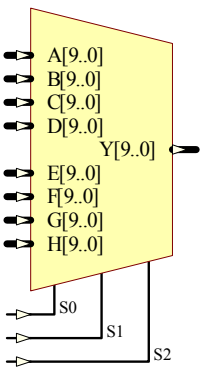
M8_B8B1E



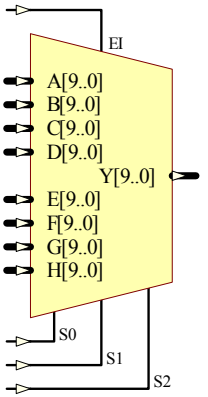
M9_B8B1



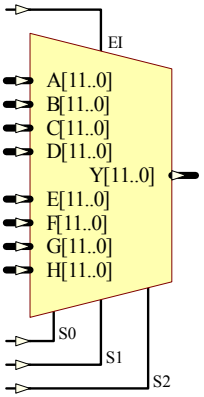
M9_B8B1E



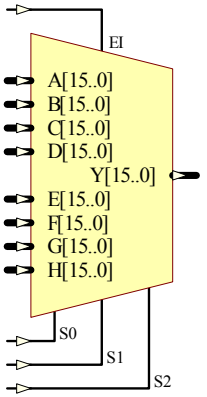
M10_B8B1



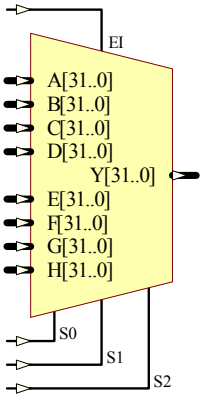
M10_B8B1E



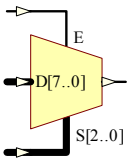
M12_B8B1E



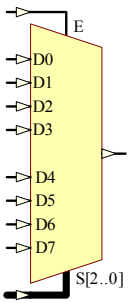
M16_B8B1E



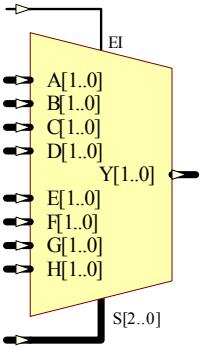
M32_B8B1E



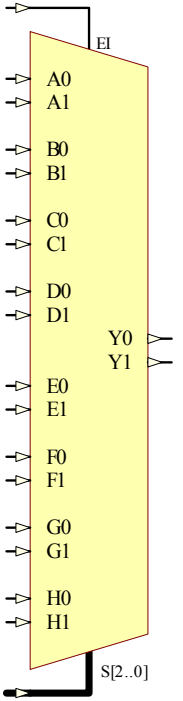
M1_B8S1E_SB



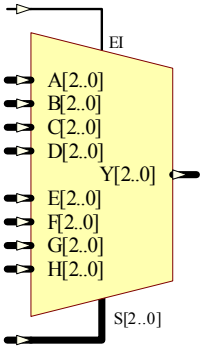
M1_S8S1E_SB



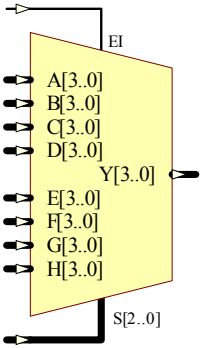
M2_B8B1E_SB



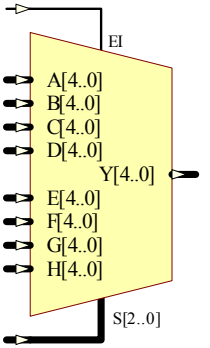
M2_S8S1E_SB



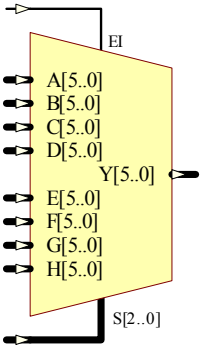
M3_B8B1E_SB



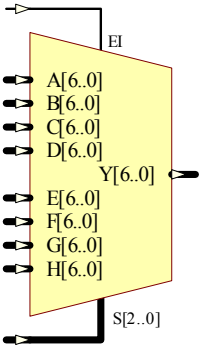
M4_B8B1E_SB



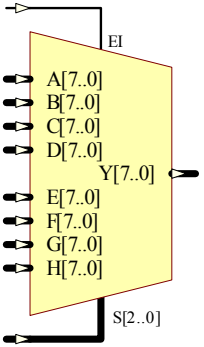
M5_B8B1E_SB



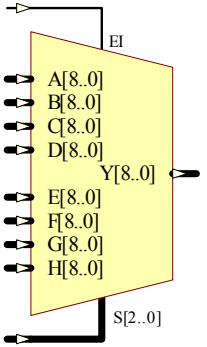
M6_B8B1E_SB



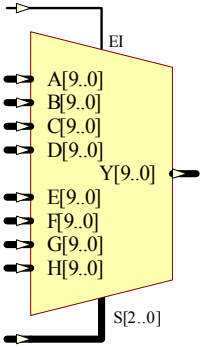
M7_B8B1E_SB



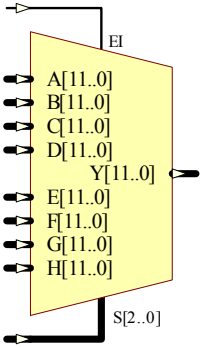
M8_B8B1E_SB



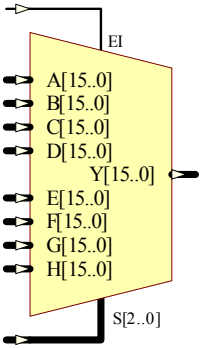
M9_B8B1E_SB



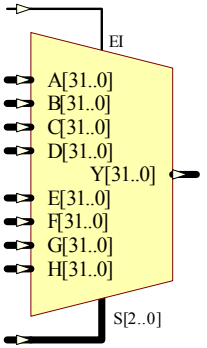
M10_B8B1E_SB



M12_B8B1E_SB



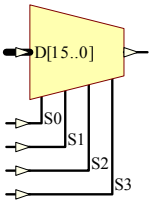
M16_B8B1E_SB



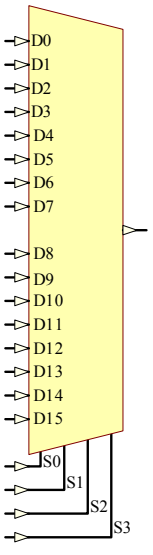
M32_B8B1E_SB

M*n*_B16B1, M1_S16S1, M1_B16S1, M*n*_B16B1_SB, M1_S16S1_SB, M1_B16S1_SB

16-to-1 Multiplexers



M1_B16S1



M1_S16S1

Mn_B16B1, M1_S16S1, M1_B16S1, Mn_B16B1_SB, M1_S16S1_SB and M1_B16S1_SB are various *n*-bit data width 16-to-1 multiplexers, available in bus-to-bus, pin-to-pin and bus-to-pin versions.

M*n*_B16B1 and M*n*_B16B1 _SB are bus-to-bus version of 16-to-1 multiplexers, which switch 16 x *n*-bit bus to 1 x *n*-bit bus. The width of the data bus, *n* is available in 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, and 32-bit.

M1_S16S1 and M1_S16S1_SB are a pin-to-pin version of 16-to-1 multiplexers, which switches 16 single pins to 1 single pin.

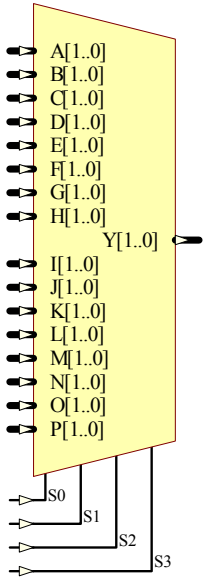
M1_B16S1 and M1_B16S1_SB are a bus-to-pin version of 16-to-1 multiplexer, which switches 1-bit of the 16-bit bus to 1 single pin.

Selects (S3-S0) are grouped in a bus (S[3..0]) for multiplexer with "_SB" suffix in the name, otherwise are separated pins.

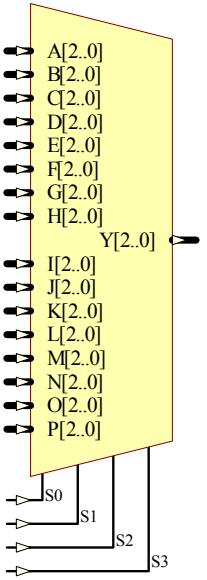
Select Inputs				Data Inputs															Output
S3	S2	S1	S0	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	O
				A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Y
0	0	0	0	a	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	a
0	0	0	1	x	b	x	x	x	x	x	x	x	x	x	x	x	x	x	x	b
0	0	1	0	x	x	c	x	x	x	x	x	x	x	x	x	x	x	x	x	c
0	0	1	1	x	x	x	d	x	x	x	x	x	x	x	x	x	x	x	x	d
0	1	0	0	x	x	x	x	e	x	x	x	x	x	x	x	x	x	x	x	e
0	1	0	1	x	x	x	x	x	f	x	x	x	x	x	x	x	x	x	x	f
0	1	1	0	x	x	x	x	x	x	g	x	x	x	x	x	x	x	x	x	g
0	1	1	1	x	x	x	x	x	x	x	h	x	x	x	x	x	x	x	x	h
1	0	0	0	x	x	x	x	x	x	x	x	i	x	x	x	x	x	x	x	i
1	0	0	1	x	x	x	x	x	x	x	x	x	j	x	x	x	x	x	x	j
1	0	1	0	x	x	x	x	x	x	x	x	x	x	k	x	x	x	x	x	k
1	0	1	1	x	x	x	x	x	x	x	x	x	x	x	l	x	x	x	x	l
1	1	0	0	x	x	x	x	x	x	x	x	x	x	x	x	m	x	x	x	m
1	1	0	1	x	x	x	x	x	x	x	x	x	x	x	x	x	n	x	x	n
1	1	1	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	o	x	o
1	1	1	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	p	p

For M1 follow D0, D1, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12, D13, D14, D15, O

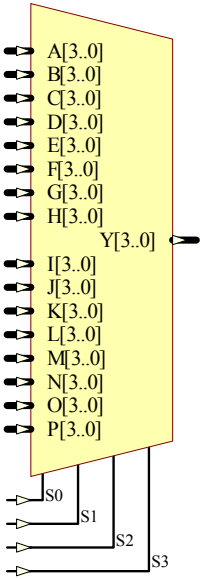
For Mn follow A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Y



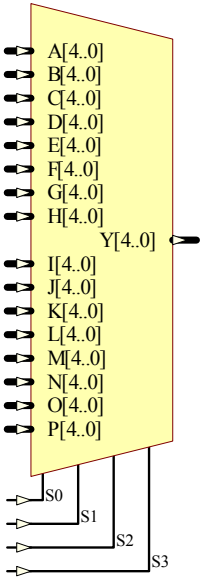
M2_B16B1



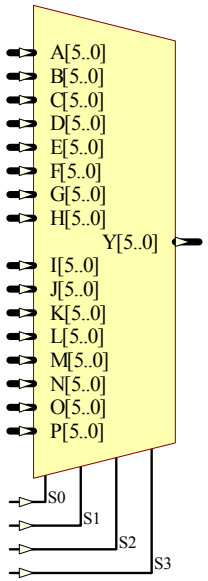
M3_B16B1



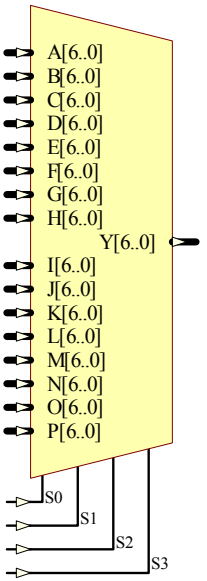
M4_B16B1



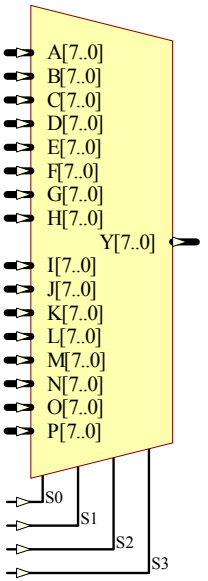
M5_B16B1



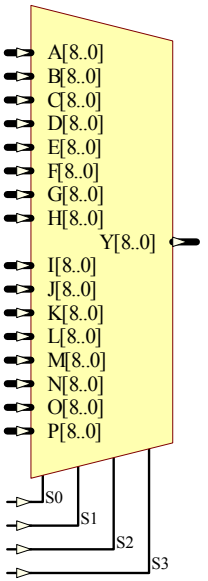
M6_B16B1



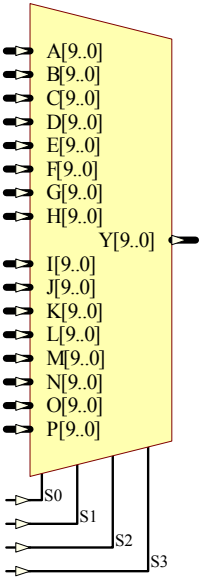
M7_B16B1



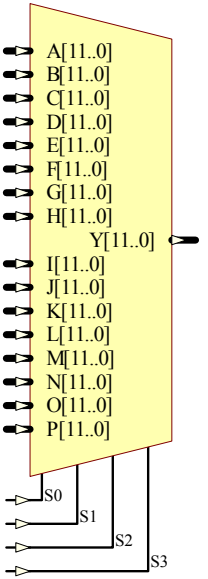
M8_B16B1



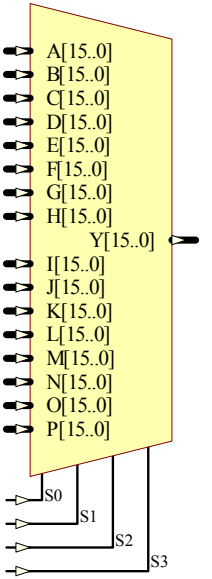
M9_B16B1



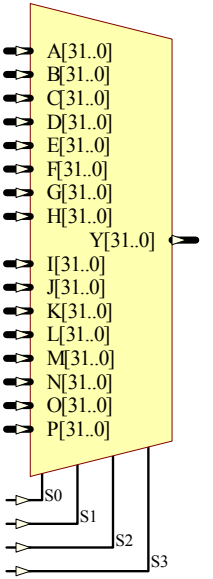
M10_B16B1



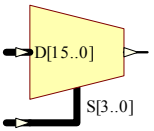
M12_B16B1



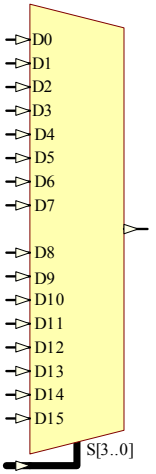
M16_B16B1



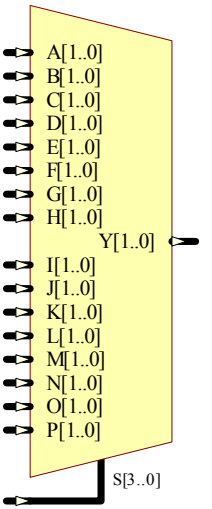
M32_B16B1



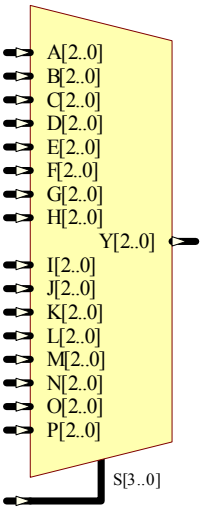
M1_B16S1_SB



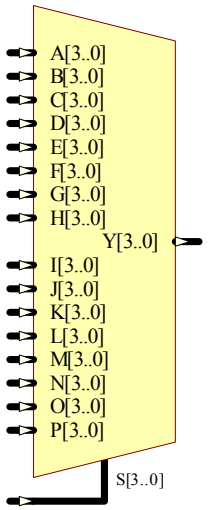
M1_S16S1_SB



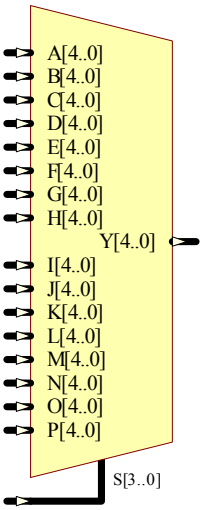
M2_B16B1_SB



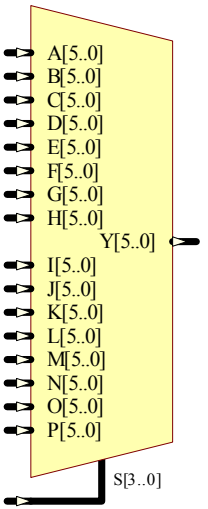
M3_B16B1_SB



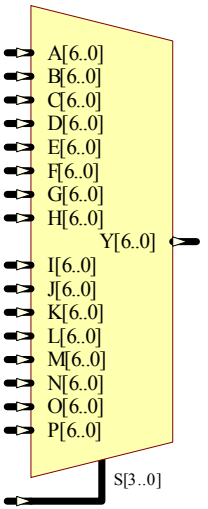
M4_B16B1_SB



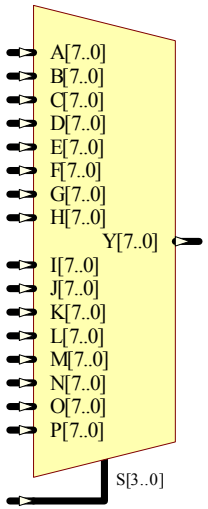
M5_B16B1_SB



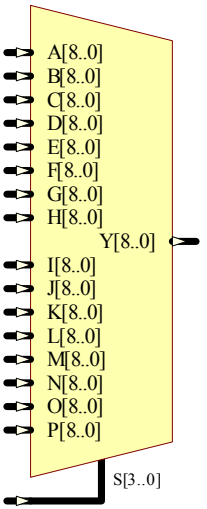
M6_B16B1_SB



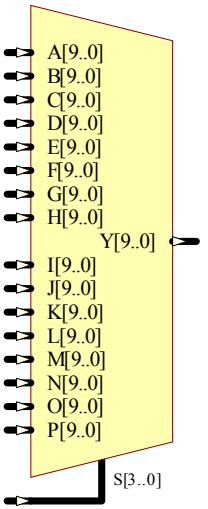
M7_B16B1_SB



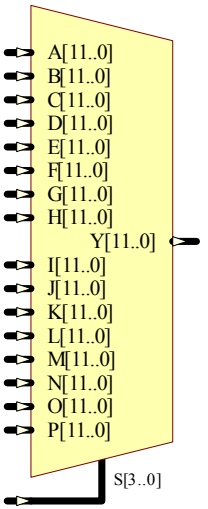
M8_B16B1_SB



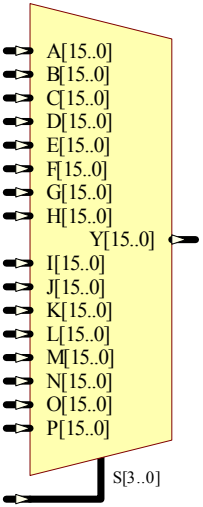
M9_B16B1_SB



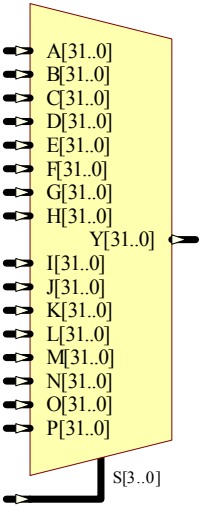
M10_B16B1_SB



M12_B16B1_SB



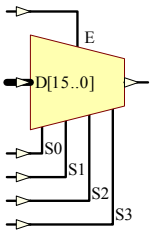
M16_B16B1_SB



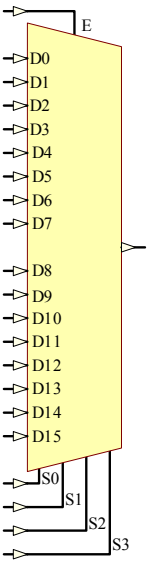
M32_B16B1_SB

M*n*_B16B1E, M1_S16S1E, M1_B16S1E, M*n*_B16B1E_SB, M1_S16S1E_SB, M1_B16S1E_SB

16-to-1 Multiplexers with Enable



M1_B16S1E



M1_S16S1E

Mn_B16B1E, M1_S16S1E, M1_B16S1E, Mn_B16B1E_SB, M1_S16S1E_SB and M1_B16S1E_SB are various *n*-bit data width 16-to-1 multiplexers with enable, available in bus-to-bus, pin-to-pin and bus-to-pin versions.

M*n*_B16B1E and M*n*_B16B1E _SB are bus-to-bus version of 16-to-1 multiplexers, which switch 16 x *n*-bit bus to 1 x *n*-bit bus. The width of the data bus, *n* is available in 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, and 32-bit.

M1_S16S1E and M1_S16S1E_SB are a pin-to-pin version of 16-to-1 multiplexers, which switches 16 single pins to 1 single pin.

M1_B16S1E and M1_B16S1E_SB are a bus-to-pin version of 16-to-1 multiplexer, which switches 1-bit of the 16-bit bus to 1 single pin.

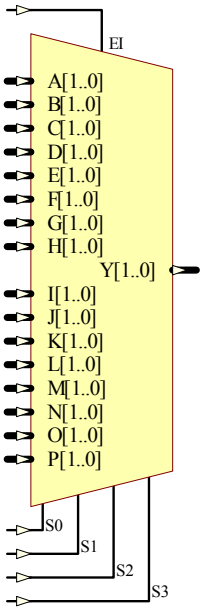
When enable is High, the multiplexers switch the data from inputs to output, when enable is Low output of the multiplexers will remain Low.

Selects (S3-S0) are grouped in a bus (S[3..0]) for multiplexer with "_SB" suffix in the name, otherwise are separated pins.

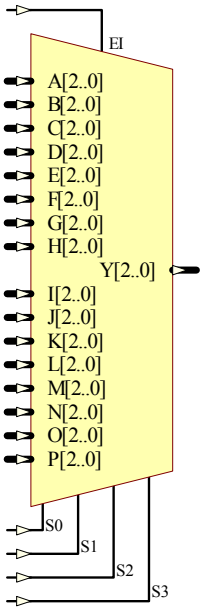
Enable	Select Inputs				Data Inputs																Output
E	S3	S2	S1	S0	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	O
EI					A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Y
0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0
1	0	0	0	0	a	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	a
1	0	0	0	1	x	b	x	x	x	x	x	x	x	x	x	x	x	x	x	x	b
1	0	0	1	0	x	x	c	x	x	x	x	x	x	x	x	x	x	x	x	x	c
1	0	0	1	1	x	x	x	d	x	x	x	x	x	x	x	x	x	x	x	x	d
1	0	1	0	0	x	x	x	x	e	x	x	x	x	x	x	x	x	x	x	x	e
1	0	1	0	1	x	x	x	x	x	f	x	x	x	x	x	x	x	x	x	x	f
1	0	1	1	0	x	x	x	x	x	x	g	x	x	x	x	x	x	x	x	x	g
1	0	1	1	1	x	x	x	x	x	x	x	h	x	x	x	x	x	x	x	x	h
1	1	0	0	0	x	x	x	x	x	x	x	x	i	x	x	x	x	x	x	x	i
1	1	0	0	1	x	x	x	x	x	x	x	x	x	j	x	x	x	x	x	x	j
1	1	0	1	0	x	x	x	x	x	x	x	x	x	x	k	x	x	x	x	x	k
1	1	0	1	1	x	x	x	x	x	x	x	x	x	x	x	l	x	x	x	x	l
1	1	1	0	0	x	x	x	x	x	x	x	x	x	x	x	x	m	x	x	x	m
1	1	1	0	1	x	x	x	x	x	x	x	x	x	x	x	x	x	n	x	x	n
1	1	1	1	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	o	x	o
1	1	1	1	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	p	p

For M1 follow E, D0, D1, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12, D13, D14, D15, O

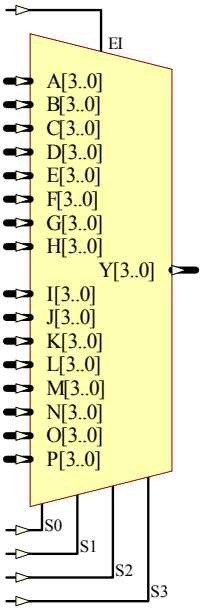
For Mn follow EI, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Y



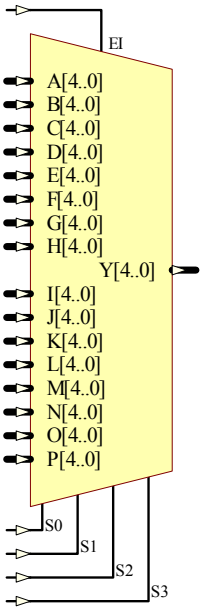
M2_B16B1E



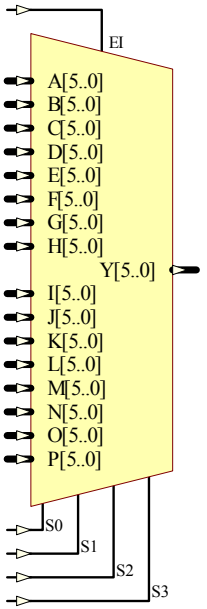
M3_B16B1E



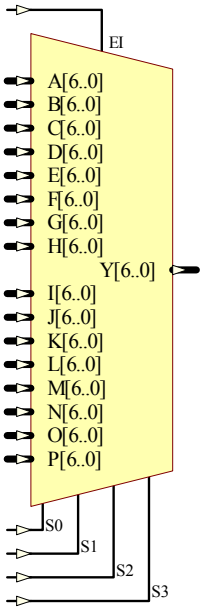
M4_B16B1E



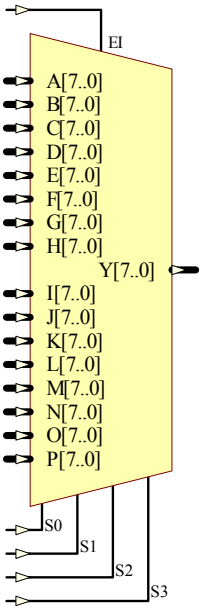
M5_B16B1E



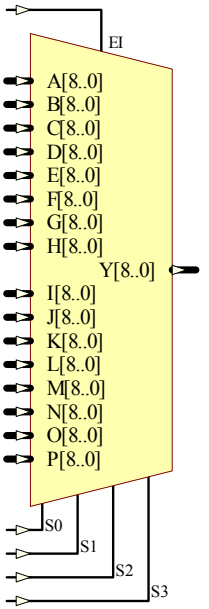
M6_B16B1E



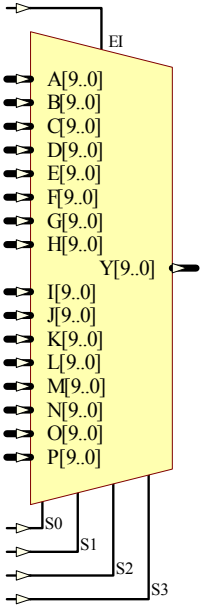
M7_B16B1E



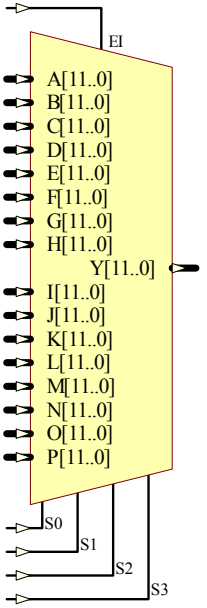
M8_B16B1E



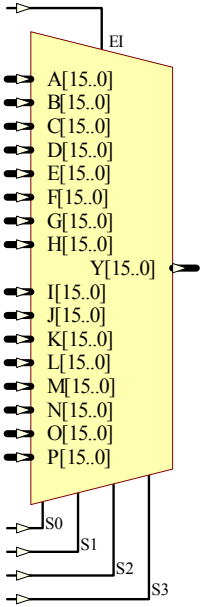
M9_B16B1E



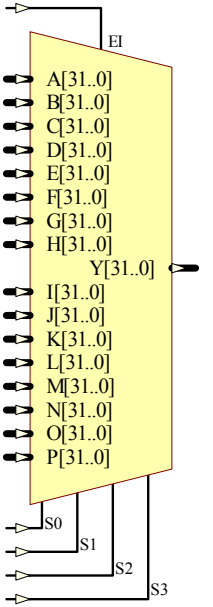
M10_B16B1E



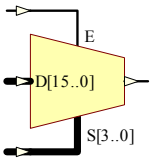
M12_B16B1E



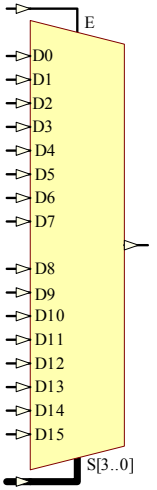
M16_B16B1E



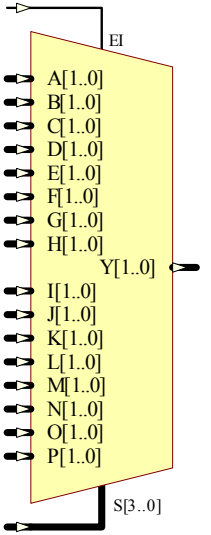
M32_B16B1E



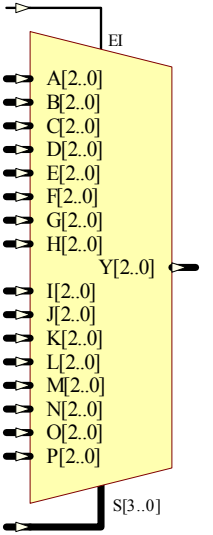
M1_B16S1E_SB



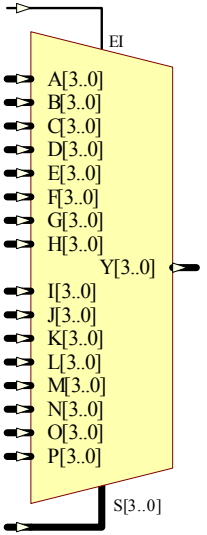
M1_S16S1E_SB



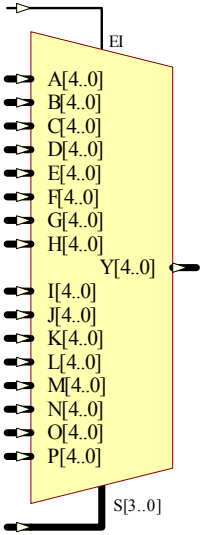
M2_B16B1E_SB



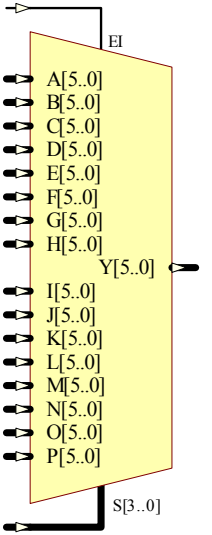
M3_B16B1E_SB



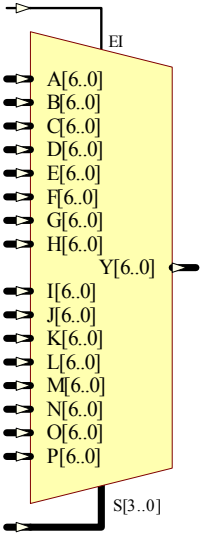
M4_B16B1E_SB



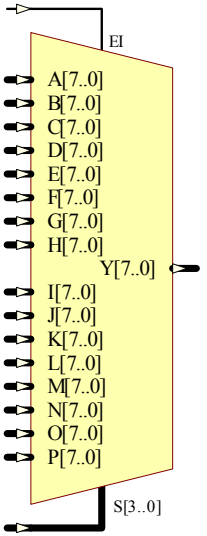
M5_B16B1E_SB



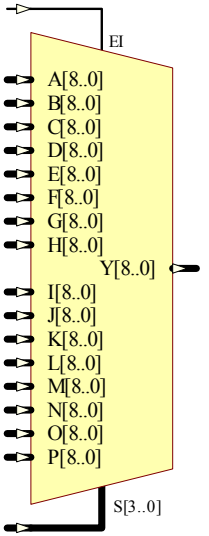
M6_B16B1E_SB



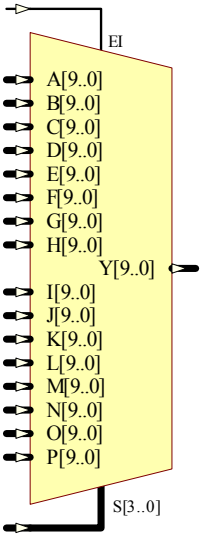
M7_B16B1E_SB



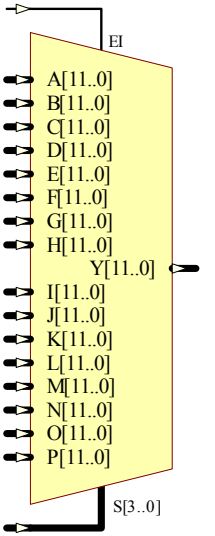
M8_B16B1E_SB



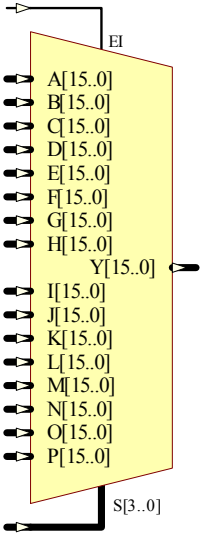
M9_B16B1E_SB



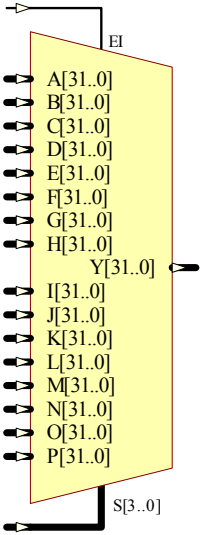
M10_B16B1E_SB



M12_B16B1E_SB



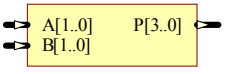
M16_B16B1E_SB



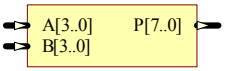
M32_B16B1E_SB

MULT2, 4, 8, 16, 18, 32

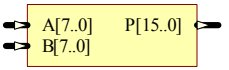
Signed Multiplier



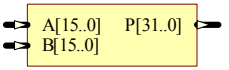
MULT2B



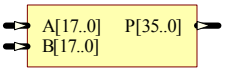
MULT4B



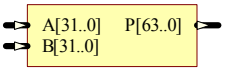
MULT8B



MULT16B



MULT18B



MULT32B

MULT2, MULT4, MULT8, MULT16, MULT18, MULT32 are respectively 2x2, 4x4, 8x8, 16x16, 18x18, 32x32 Signed Multipliers. They perform multiplication of two signed values from the two inputs (A and B) and produce a product (P).

Input A and B are 2-, 4-, 8-, 16-, 18, 32-Bit length and output P is 4-, 8-, 16-, 32-, 36, 64-Bit length for MULT2, MULT4, MULT8, MULT16, MULT18, MULT32 respectively. All input and output values are represented in two-complement format.

Input		Output
A	B	P
a	b	a x b

For MULT2, A=A1-A0, B=B1-B0, P=P3-P0

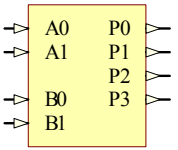
For MULT4, A = A3-A0, B= B3-B0, P=P7-P0

For MULT8, A = A7-A0, B= B7-B0, P=P15-P0

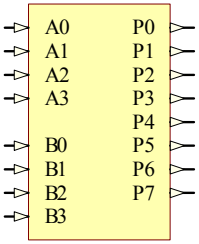
For MULT18, A = A17-A0, B= B17-B0, P=P35-P0

For MULT16, A = A15-A0, B= B15-B0, P=P31-P0

For MULT32, A = A31-A0, B= B31-B0, P=P63-P0



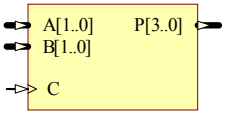
MULT2S



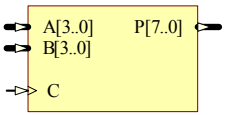
MULT4S

MULTR2, 4, 8, 16, 18, 32

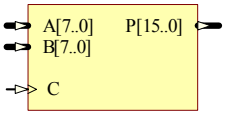
Registered Signed Multiplier



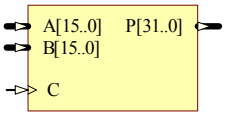
MULTR2B



MULTR4B



MULTR8B



MULTR16B

MULTR2, MULTR4, MULTR8, MULTR16, MULTR18, MULTR32 are respectively 2x2, 4x4, 8x8, 16x16, 18x18, 32x32 registered signed multipliers. They perform multiplication of two signed values from the two inputs (A and B) on the rising-edge of the clock input (C) and produce a product (P).

Input A and B are 2-, 4-, 8-, 16-, 18, 32-Bit length and output P is 4-, 8-, 16-, 32-, 36, 64-Bit length for MULTR2, MULTR4, MULTR8, MULTR16, MULTR18, MULTR32 respectively. All input and output values are represented in two's-complement format.

Input			Output
C	A	B	P
↑	a	b	a x b

For MULTR2, A=A1-A0, B=B1-B0, P=P3-P0

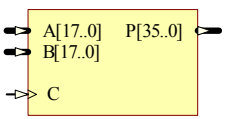
For MULTR4, A = A3-A0, B= B3-B0, P=P7-P0

For MULTR8, A = A7-A0, B= B7-B0, P=P15-P0

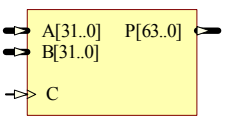
For MULTR18, A = A17-A0, B= B17-B0, P=P35-P0

For MULTR16, A = A15-A0, B= B15-B0, P=P31-P0

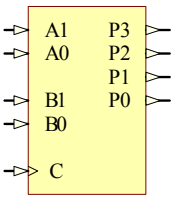
For MULTR32, A = A31-A0, B= B31-B0, P=P63-P0



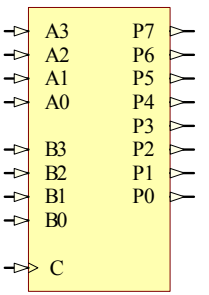
MULTR18B



MULTR32B



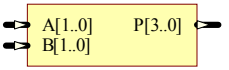
MULTR2S



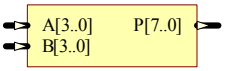
MULTR4S

MULTU2, 4, 8, 16, 18, 32

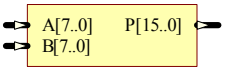
Unsigned Multiplier



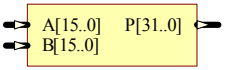
MULTU2B



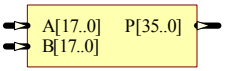
MULTU4B



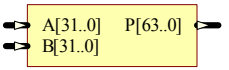
MULTU8B



MULTU16B



MULTU18B



MULTU32B

MULTU2, MULTU4, MULTU8, MULTU16, MULTU18, MULTU32 are respectively 2x2, 4x4, 8x8, 16x16, 18x18, 32x32Unsigned Multipliers. They perform multiplication of two signed values from the two inputs (A and B) and produce a product (P).

Input A and B are 2-, 4-, 8-, 16-, 18, 32-Bit length and output P is 4-, 8-, 16-, 32-, 36, 64-Bit length for MULTU2, MULTU4, MULTU8, MULTU16, MULTU18, MULTU32 respectively. All input and output values are represented in unsigned binary format.

Input		Output
A	B	P
a	b	a x b

For MULTU2, A=A1-A0, B=B1-B0, P=P3-P0

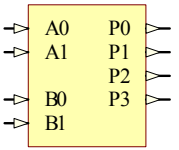
For MULTU4, A = A3-A0, B= B3-B0, P=P7-P0

For MULTU8, A = A7-A0, B= B7-B0, P=P15-P0

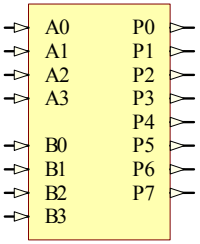
For MULTU18, A = A17-A0, B= B17-B0, P=P35-P0

For MULTU16, A = A15-A0, B= B15-B0, P=P31-P0

For MULTU32, A = A31-A0, B= B31-B0, P=P63-P0



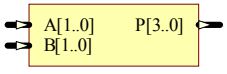
MULTU2S



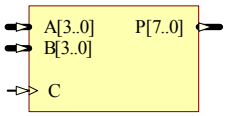
MULTU4S

MULTUR2, 4, 8, 16, 18, 32

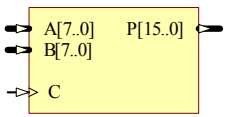
Registered Unsigned Multiplier



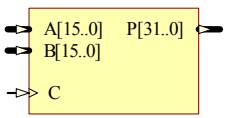
MULTUR2B



MULTUR4B



MULTUR8B



MULTUR16B

MULTUR2, MULTUR4, MULTUR8, MULTUR16, MULTUR18, MULTUR32 are respectively 2x2, 4x4, 8x8, 16x16, 18x18, 32x32 registered unsigned multipliers. They perform multiplication of two signed values from the two inputs (A and B) on the rising-edge of the clock input (C) and produce a product (P).

Input A and B are 2-, 4-, 8-, 16-, 18, 32-Bit length and output P is 4-, 8-, 16-, 32-, 36, 64-Bit length for MULTUR2, MULTUR4, MULTUR8, MULTUR16, MULTUR18, MULTUR32 respectively. All input and output values are represented in unsigned binary format.

Input			Output
C	A	B	P
↑	a	b	a x b

For MULTUR2, A=A1-A0, B=B1-B0, P=P3-P0

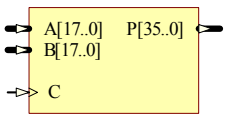
For MULTUR4, A = A3-A0, B= B3-B0, P=P7-P0

For MULTUR8, A = A7-A0, B= B7-B0, P=P15-P0

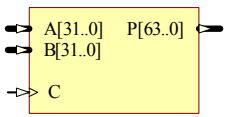
For MULTUR18, A = A17-A0, B= B17-B0, P=P35-P0

For MULTUR16, A = A15-A0, B= B15-B0, P=P31-P0

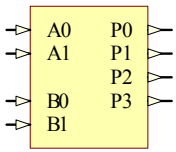
For MULTUR32, A = A31-A0, B= B31-B0, P=P63-P0



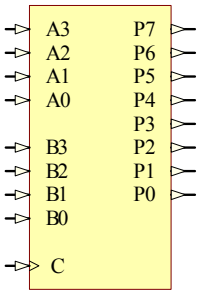
MULTUR18B



MULTUR32B



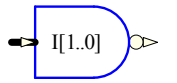
MULTUR2S



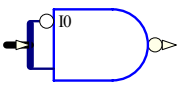
MULTUR4S

NAND2 – 32

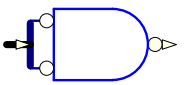
NAND Gates



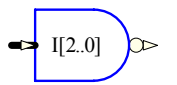
NAND2B



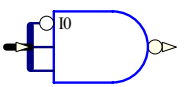
NAND2N1B



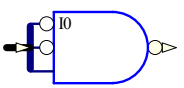
NAND2N2B



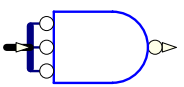
NAND3B



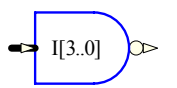
NAND3N1B



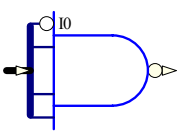
NAND3N2B



NAND3N3B



NAND4B



NAND4N1B

NAND Gates provide a variety of NAND functions, range from 2 to 32 inverted or non-inverted Inputs.

NAND*n* - Non-Inverted input NAND Gates

n is input bit length, *n* = 2, 3, 4, 5, 6, 7, 8, 9, 12, 13 16

Input			Output
I0	**…**	**I*n*-1**	**O**
1	1	1	0
0	x	x	1
x	0	x	1
x	x	0	1

NAND*n*N*m* - Inverted input NAND Gates

n is input bit length, *m* is number of inverted input.

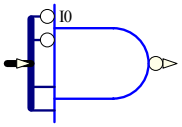
m, n = 2, 3, 4, 5, m <= n.

Input						Output
I0	**…**	**I*m-1***	**I*m***	**…**	**I*n-1***	**O**
0	0	0	1	1	1	0
1	x	x	x	x	x	1
x	1	x	x	x	x	1
x	x	1	x	x	x	1
x	x	x	0	x	x	1
x	x	x	x	0	x	1
x	x	x	x	x	0	1

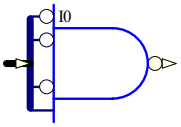
NAND*n*T – 3-state Output NAND Gates

n is input bit length, *n* = 12

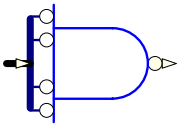
Input				Output
EN	**I0**	**…**	**I*n*-1**	**O**
0	1	1	1	0
0	0	x	x	1
0	x	0	x	1
0	x	x	0	1
1	x	x	x	Z



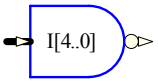
NAND4N2B



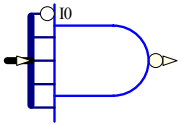
NAND4N3B



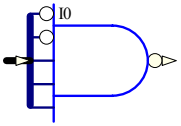
NAND4N4B



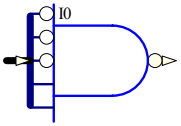
NAND5B



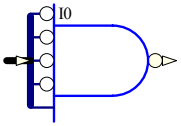
NAND5N1B



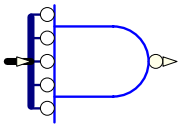
NAND5N2B



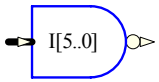
NAND5N3B



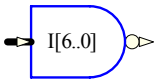
NAND5N4B



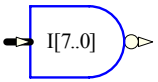
NAND5N5B



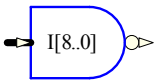
NAND6B



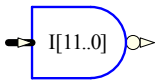
NAND7B



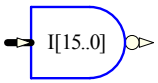
NAND8B



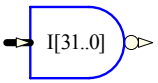
NAND9B



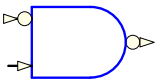
NAND12B



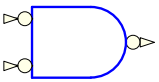
NAND16B



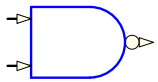
NAND32B



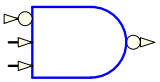
NAND2N1S



NAND2N2S



NAND2S

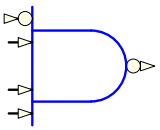


NAND3N1S

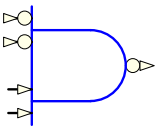
NAND3N2S

NAND3N3S

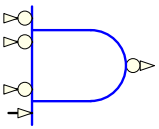
NAND3S



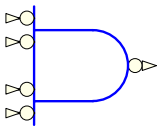
NAND4N1S



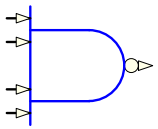
NAND4N2S



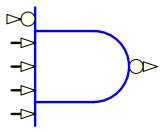
NAND4N3S



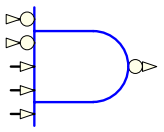
NAND4N4S



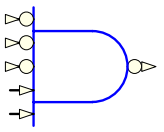
NAND4S



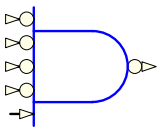
NAND5N1S



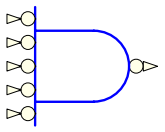
NAND5N2S



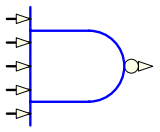
NAND5N3S



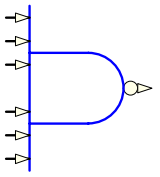
NAND5N4S



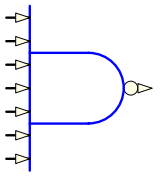
NAND5N5S



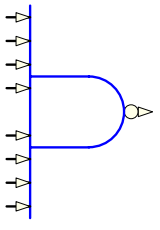
NAND5S



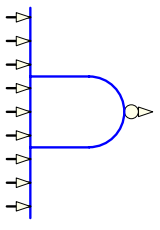
NAND6S



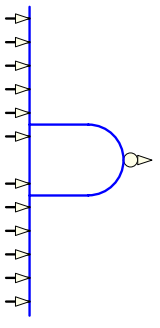
NAND7S



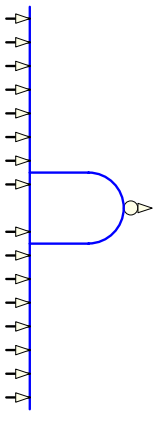
NAND8S



NAND9S



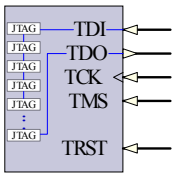
NAND12S



NAND16S

NEXUS_JTAG_PORT

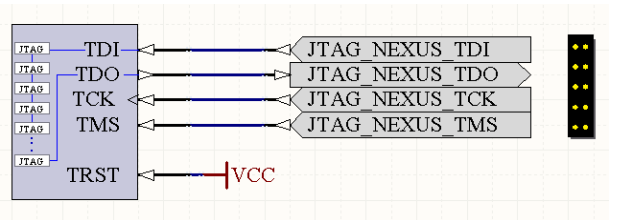
Soft Nexus-Chain Connector



NEXUS_JTAG_PORT

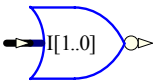
This component is required when using any components from the “FPGA Instruments.IntLib” integrated library or using on chip debug (OCD) system type processors from the “FPGA Processors.IntLib” integrated library.

NEXUS_JTAG_PORT essentially forms the soft JTAG ports of all debug systems cores and instruments. It allows all JTAG ports to be chained together in one single component. The example below show a typical connection with NEXUS_JTAG_CONNECTOR port component from the “FPGA NanoBoard Port-Plugin.IntLib” integrated library.

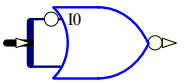


NOR2 – 32

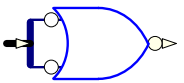
NOR Gates



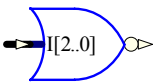
NOR2B



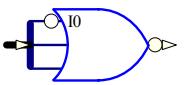
NOR2N1B



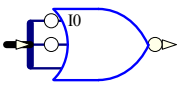
NOR2N2B



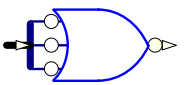
NOR3B



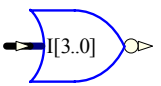
NOR3N1B



NOR3N2B



NOR3N3B



NOR4B

NOR Gates provide a variety of NOR functions, range from 2 to 32 inverted or non-inverted Inputs and with or without strobe.

NOR*n* - Non-Inverted input NOR Gates

n is input bit length, *n* = 2, 3, 4, 5, 6, 7, 8, 9, 12, 16, 32

Input			Output
I0	**…**	**I*n*-1**	**O**
0	0	0	1
1	x	x	0
x	1	x	0
x	x	1	0

NOR*n*N*m* - Inverted input NOR Gates

n is input bit length, *m* is number of inverted input.

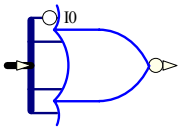
m, n = 2, 3, 4, 5, *m* <= *n.*

Input						Output
I0	**…**	**I*m-1***	**I*m***	**…**	**I*n-1***	**O**
1	1	1	0	0	0	1
0	x	x	x	x	x	0
x	0	x	x	x	x	0
x	x	0	x	x	x	0
x	x	x	1	x	x	0
x	x	x	x	1	x	0
x	x	x	x	x	1	0

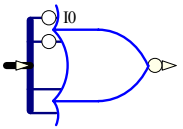
NOR*n*G – NOR Gates with strobes

n is input bit length, *n* = 4

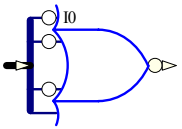
Input				Output
G	**I0**	**…**	**I*n*-1**	**O**
1	0	0	0	1
1	1	x	x	0
1	x	1	x	0
1	x	x	1	0
0	x	x	x	0



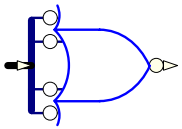
NOR4N1B



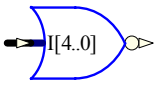
NOR4N2B



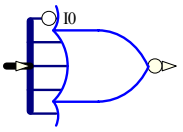
NOR4N3B



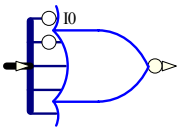
NOR4N4B



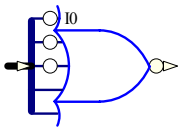
NOR5B



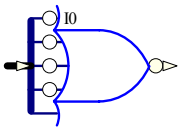
NOR5N1B



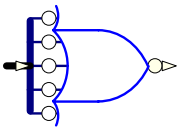
NOR5N2B



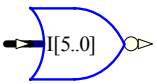
NOR5N3B



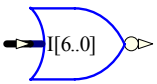
NOR5N4B



NOR5N5B

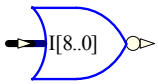


NOR6B

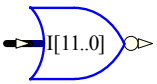


NOR7B

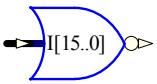
NOR8B



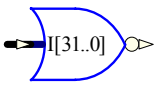
NOR9B



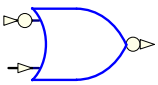
NOR12B



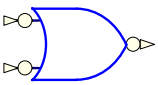
NOR16B



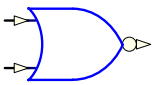
NOR32B



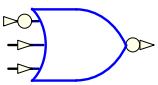
NOR2N1S



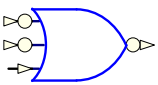
NOR2N2S



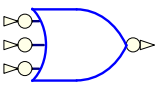
NOR2S



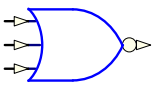
NOR3N1S



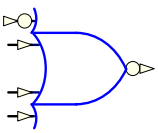
NOR3N2S



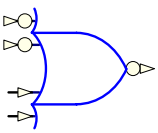
NOR3N3S



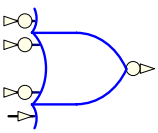
NOR3S



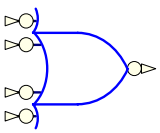
NOR4N1S



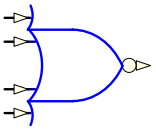
NOR4N2S



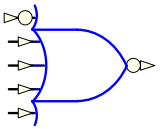
NOR4N3S



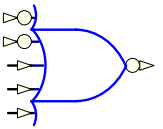
NOR4N4S



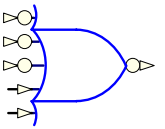
NOR4S



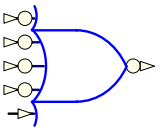
NOR5N1S



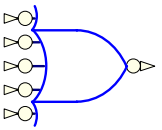
NOR5N2S



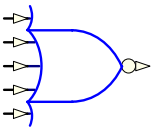
NOR5N3S



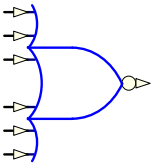
NOR5N4S



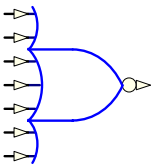
NOR5N5S



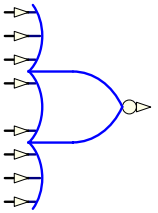
NOR5S



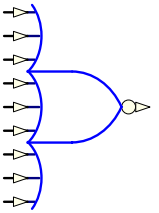
NOR6S



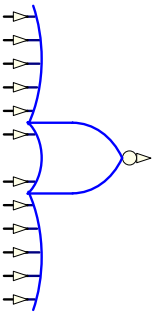
NOR7S



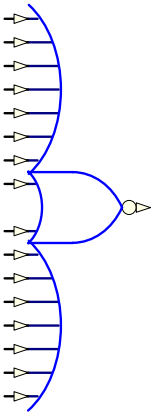
NOR8S



NOR9S



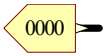
NOR12S



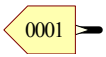
NOR16S

NUM0 – NUMF

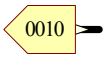
Hex Number Connector of Value 0 – F



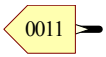
NUM0



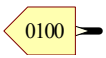
NUM1



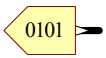
NUM2



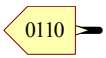
NUM3



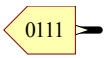
NUM4



NUM5



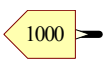
NUM6



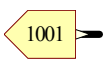
NUM7

NUM*n* is used to provide a permanent HEX value *n* at the output (O3-O0). The following truth table shows the output of all number connector.

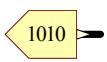
NUMn	Hex Value	Outputs			
		O3	O2	O1	O0
NUM0	0	0	0	0	0
NUM1	1	0	0	0	1
NUM2	2	0	0	1	0
NUM3	3	0	0	1	1
NUM4	4	0	1	0	0
NUM5	5	0	1	0	1
NUM6	6	0	1	1	0
NUM7	7	0	1	1	1
NUM8	8	1	0	0	0
NUM9	9	1	0	0	1
NUMA	A	1	0	1	0
NUMB	B	1	0	1	1
NUMC	C	1	1	0	0
NUMD	D	1	1	0	1
NUME	E	1	1	1	0
NUMF	F	1	1	1	1



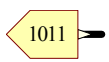
NUM8



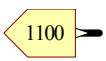
NUM9



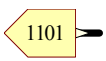
NUMA



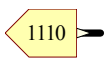
NUMB



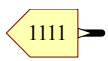
NUMC



NUMD



NUME

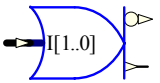


NUMF

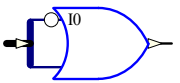
OR2 – 32

OR Gates

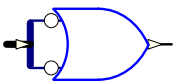
OR2B



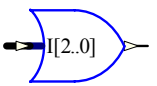
OR2DB



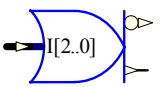
OR2N1B



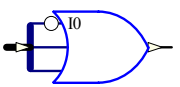
OR2N2B



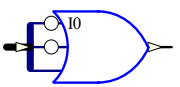
OR3B



OR3DB



OR3N1B



OR3N2B

OR Gates provide a variety of OR functions, range from 2 to 32 inverted or non-inverted Inputs and Single or Dual output.

OR*n* - Non-Inverted input OR Gates

n is input bit length, *n* = 2, 3, 4, 5, 6, 7, 8, 9, 12, 16, 32

Input			Output
I0	**…**	**I*n*-1**	**O**
0	0	0	0
1	x	x	1
x	1	x	1
x	x	1	1

OR*n*N*m* – Inverted input OR Gates

n is input bit length, *m* is number of inverted input.

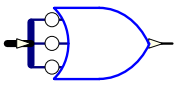
m, n = 2, 3, 4, 5, *m* <= *n.*

Input						Output
I0	**…**	**I*m-1***	**I*m***	**…**	**I*n-1***	**O**
1	1	1	0	0	0	0
0	x	x	x	x	x	1
x	0	x	x	x	x	1
x	x	0	x	x	x	1
x	x	x	1	x	x	1
x	x	x	x	1	x	1
x	x	x	x	x	1	1

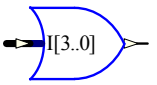
OR*n*D - Dual Output OR Gates

n is input bit length, *n* = 2, 3, 4, 8

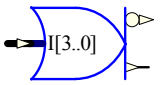
Input			Output	
I0	**…**	**I*n-1***	**Y**	**YN**
0	0	0	0	1
1	x	x	1	0
x	1	x	1	0
x	x	1	1	0



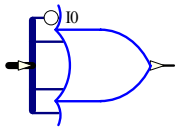
OR3N3B



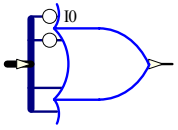
OR4B



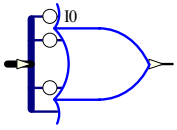
OR4DB



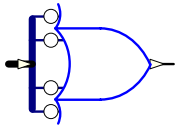
OR4N1B



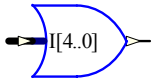
OR4N2B



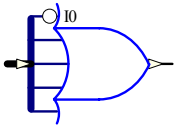
OR4N3B



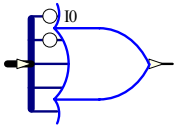
OR4N4B



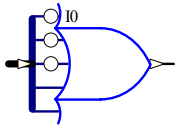
OR5B



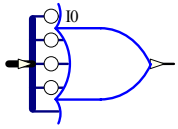
OR5N1B



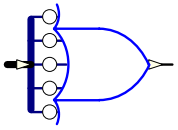
OR5N2B



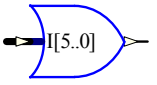
OR5N3B



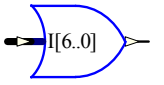
OR5N4B



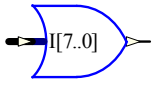
OR5N5B



OR6B

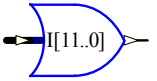


OR7B

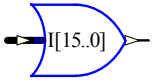


OR8B

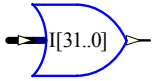
OR9B



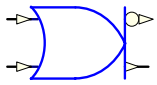
OR12B



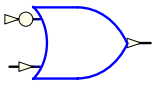
OR16B



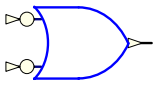
OR32B



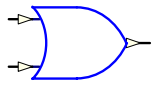
OR2DS



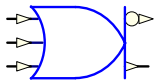
OR2N1S



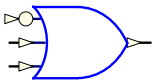
OR2N2S



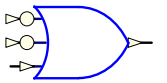
OR2S



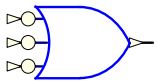
OR3DS



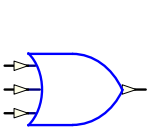
OR3N1S



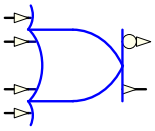
OR3N2S



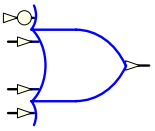
OR3N3S



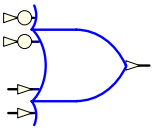
OR3S



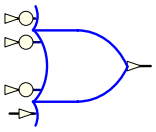
OR4DS



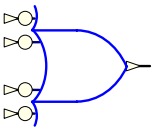
OR4N1S



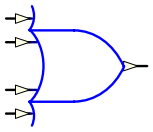
OR4N2S



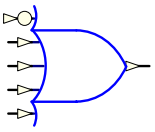
OR4N3S



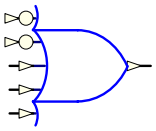
OR4N4S



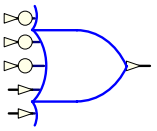
OR4S



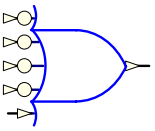
OR5N1S



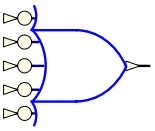
OR5N2S



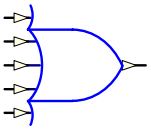
OR5N3S



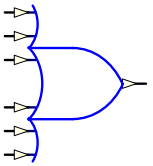
OR5N4S



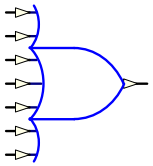
OR5N5S



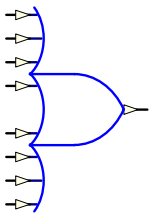
OR5S



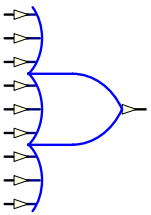
OR6S



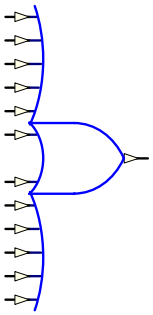
OR7S



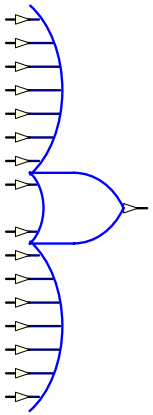
OR8S



OR9S



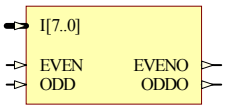
OR12S



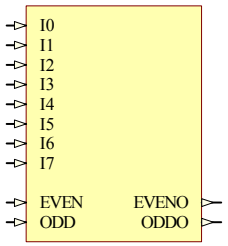
OR16S

PAR9

9-Bit Odd/Even Parity Generators/Checkers



PAR9B



PAR9S

PAR9 is a universal, monolithic, 9-bit (8 data bits plus 1 parity bit) parity generators/checkers feature odd/even outputs (EVENO, ODDO) and control inputs to facilitate operation in either odd- or even-parity application. Depending on whether even or odd parity is being generated or checked, then EVEN or ODD inputs can be utilized as the parity or the 9th-bit input.

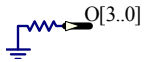
Inputs			Outputs	
I7-I0	**EVEN**	**ODD**	**EVENO**	**ODDO**
even number of '1's	1	0	1	0
odd number of '1's	1	0	0	1
even number of '1's	0	1	0	1
odd number of '1's	0	1	1	0
x	1	1	0	0
x	0	0	1	1

PULLDOWN, 4, 8, 12, 16, 32

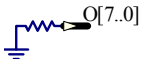
Pull Down Resistors

PULLDOWN

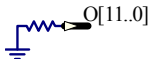
PULLDOWN, PULLDOWN4, PULLDOWN8, PULLDOWN12, PULLDOWN16, PULLDOWN32 are, respectively 1-bit, 4-bit, 12-bit, 16-bit, 32-bit pull-down resistors.



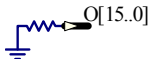
PULLDOWN4B



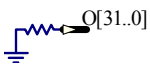
PULLDOWN8B



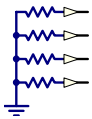
PULLDOWN12B



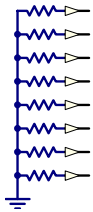
PULLDOWN16B



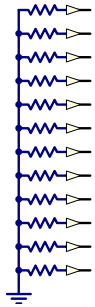
PULLDOWN32B



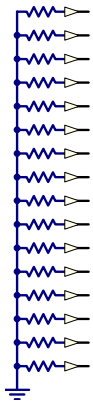
PULLDOWN4S



PULLDOWN8S



PULLDOWN12S



PULLDOWN16S

PULLUP, 4, 8, 12, 16, 32

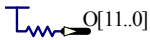
Pull Up Resistors

PULLUP

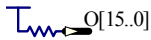
PULLUP, PULLUP4, PULLUP8, PULLUP12, PULLUP16, PULLUP32 are, respectively 1-bit, 4-bit, 12-bit, 16-bit and 32-bit pull-up resistors.

PULLUP4B

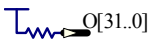
PULLUP8B



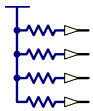
PULLUP12B



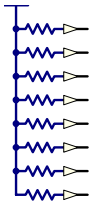
PULLUP16B



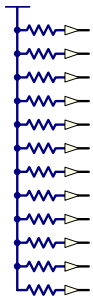
PULLUP32B



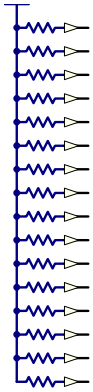
PULLUP4S



PULLUP8S



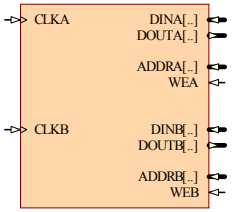
PULLUP12S



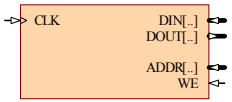
PULLUP16S

RAMS, RAMD

Single/Dual Port Random Access Memory



RAMD



RAMS

The RAMS and RAMD are single and dual port random access memory.

When write enable (WE) is High, data (DIN) input is transferred to the memory on the configured clock trigger, addressed by the input (ADDR) address bus.

Data output (DOUT) is always active on the defined clock trigger and valid address (ADDR) input.

Data lengths and clock trigger is configurable by editing the following parameters found on the component properties:

Memory_Depth - This defines the depth of memory. It is set to DefinedBy=ADDR by default. This allows automatic configuration of ram depth size depending on the size of the address bus connected on the address (ADDR) input port.

Memory_Width - This defines the data port size. It is set to DefinedBy=DIN. This allows automatic configuration of ram data port size depending on the size of data bus connected to the data (DIN) input port.

Memory_ClockEdge - This parameter can be set to Rising or Falling depending on your desired clock trigger. It is set to Rising by default.

Memory_ContentFile – This enables the ram content to be initialized using a Hex (Intel format) file.

Inputs				Outputs	
WE	**CLK**	**ADDR**	**DIN**	**DOUT**	**RAM Contents**
0	clk	addr	X	RAM(addr)	No Chg
1	clk	addr	data	data	RAM(addr) => data

addr=RAM address

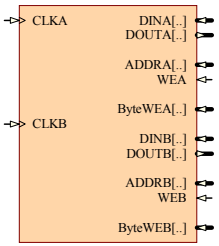
RAM(addr)=RAM contents at address ADDR

data=RAM input data

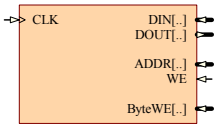
clk=Clock edge defined by Memory_ClockEdge parameter

RAMSB, RAMDB

Single/Dual Port Random Access Memory With Byte Write Enable



RAMDB



RAMSB

The RAMSB and RAMDB are single and dual port random access memory with byte write enable. When the memory width spans multiple bytes, individual bytes of data can be accessed during the Read or Write cycles.

When write enable (WE) is high, depending on the value of byte write enable (ByteWE), corresponding bytes of data (DIN) input are transferred to the memory on the configured clock trigger, addressed by the input (ADDR) address bus.

Data output (DOUT) is always active, depending on the value of byte write enable (ByteWE), on the defined clock trigger and valid address (ADDR) input.

Data lengths and clock trigger is configurable by editing the following parameters found on the component properties:

Memory_Depth - This defines the depth of memory. It is set to DefinedBy=ADDR by default. This allows automatic configuration of ram depth size depending on the size of the address bus connected on the address (ADDR) input port.

Memory_Width - This defines the data port size. It is set to DefinedBy=DIN. This allows automatic configuration of ram data port size depending on the size of data bus connected to the data (DIN) input port.

Memory_ClockEdge - This parameter can be set to Rising or Falling depending on your desired clock trigger. It is set to Rising by default.

Memory_ContentFile – This enables the ram content to be initialized using a Hex (Intel format) file.

Inputs				Outputs	
WE	**CLK**	**ADDR**	**DIN**	**DOUT**	**RAM Contents**
0	clk	addr	X	RAM(ByteWE(addr))	No Chg
1	clk	addr	data	data	RAM(ByteWE(addr)) => data

addr=RAM address

RAM(addr)=RAM contents at address ADDR

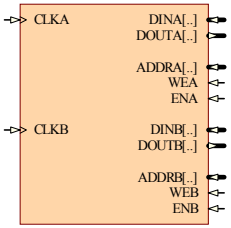
data=RAM input data

clk=Clock edge defined by Memory_ClockEdge parameter

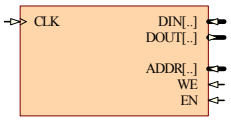
ByteWE=Select signal for individual bytes

RAMSE, RAMDE

Single/Dual Port Random Access Memory With Enable



RAMDE



RAMSE

The RAMSE and RAMDE are single and dual port random access memory with enable.

When enable (EN) is High, data transfer is disabled and no change occurs at data output (DOUT).

When enable (EN) is Low and write enable (WE) is High, data (DIN) input is transferred to the memory on the configured clock trigger, addressed by the input (ADDR) address bus.

When enable (EN) is Low, data output (DOUT) is always active on the defined clock trigger and valid address (ADDR) input.

Data lengths and clock trigger is configurable by editing the following parameters found on the component properties:

Memory_Depth - This defines the depth of memory. It is set to DefinedBy=ADDR by default. This allows automatic configuration of ram depth size depending on the size of the address bus connected on the address (ADDR) input port.

Memory_Width - This defines the data port size. It is set to DefinedBy=DIN. This allows automatic configuration of ram data port size depending on the size of data bus connected to the data (DIN) input port.

Memory_ClockEdge - This parameter can be set to Rising or Falling depending on your desired clock trigger. It is set to Rising by default.

Memory_ContentFile – This enables the ram content to be initialized using a Hex (Intel format) file.

Inputs					Outputs	
EN	**WE**	**CLK**	**ADDR**	**DIN**	**DOUT**	**RAM Contents**
1	X	X	X	X	No Chg	No Chg
0	0	clk	addr	X	RAM(addr)	No Chg
0	1	clk	addr	data	data	RAM(addr) => data

addr=RAM address

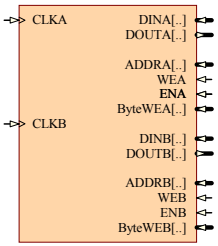
RAM(addr)=RAM contents at address ADDR

data=RAM input data

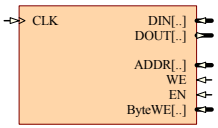
clk=Clock edge defined by Memory_ClockEdge parameter

RAMSEB, RAMDEB

Single/Dual Port Random Access Memory With Enable And Byte Write Enable



RAMDEB



RAMSEB

The RAMSEB and RAMDEB are single and dual port random access memory with enable and byte write enable. When the memory width spans multiple bytes, individual bytes of data can be accessed during the Read or Write cycles.

When enable (EN) is High, data transfer is disabled and no change occurs at data output (DOUT). When enable (EN) is Low and write enable (WE) is High, depending on the value of byte write enable (ByteWE), corresponding bytes of data (DIN) input are transferred to the memory on the configured clock trigger, addressed by the input (ADDR) address bus.

When enable (EN) is Low, data output (DOUT) is always active, depending on the value of byte write enable (ByteWE), on the defined clock trigger and valid address (ADDR) input.

Data lengths and clock trigger is configurable by editing the following parameters found on the component properties:

Memory_Depth - This defines the depth of memory. It is set to DefinedBy=ADDR by default. This allows automatic configuration of ram depth size depending on the size of the address bus connected on the address (ADDR) input port.

Memory_Width - This defines the data port size. It is set to DefinedBy=DIN. This allows automatic configuration of ram data port size depending on the size of data bus connected to the data (DIN) input port.

Memory_ClockEdge - This parameter can be set to Rising or Falling depending on your desired clock trigger. It is set to Rising by default.

Memory_ContentFile – This enables the ram content to be initialized using a Hex (Intel format) file.

Inputs					Outputs	
EN	**WE**	**CLK**	**ADDR**	**DIN**	**DOUT**	**RAM Contents**
1	X	X	X	X	No Chg	No Chg
0	0	clk	addr	X	RAM(ByteWE(addr))	No Chg
0	1	clk	addr	data	data	RAM(ByteWE(addr)) => data

addr=RAM address

RAM(addr)=RAM contents at address ADDR

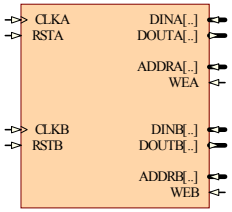
data=RAM input data

clk=Clock edge defined by Memory_ClockEdge parameter

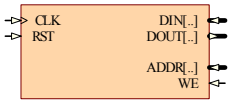
ByteWE=Select signal for individual bytes

RAMSR, RAMDR

Single/Dual Port Random Access Memory With Reset



RAMDR



RAMSR

The RAMSR and RAMDR are single and dual port random access memory with reset.

When reset (RST) is High, data output (DOUT) is cleared on the defined clock trigger.

When write enable (WE) is High, data input (DIN) can be transferred to memory on the defined clock trigger while reset is High or Low.

When reset (RST) and write enable (WE) is Low, data output (DOUT) is active on the defined clock trigger and valid address (ADDR) input.

Data lengths and clock trigger is configurable by editing the following parameters found on the component properties:

Memory_Depth - This defines the depth of memory. It is set to DefinedBy=ADDR by default. This allows automatic configuration of ram depth size depending on the size of the address bus connected on the address (ADDR) input port.

Memory_Width - This defines the data port size. It is set to DefinedBy=DIN. This allows automatic configuration of ram data port size depending on the size of data bus connected to the data (DIN) input port.

Memory_ClockEdge - This parameter can be set to Rising or Falling depending on your desired clock trigger. It is set to Rising by default.

Memory_ContentFile – This enables the ram content to be initialized using a Hex (Intel format) file.

Inputs					Outputs	
RST	**WE**	**CLK**	**ADDR**	**DIN**	**DOUT**	**RAM Contents**
1	0	clk	X	X	0	No Chg
1	1	clk	addr	data	0	RAM(addr) => data
0	0	clk	addr	X	RAM(addr)	No Chg
0	1	clk	addr	data	data	RAM(addr) => data

addr=RAM address

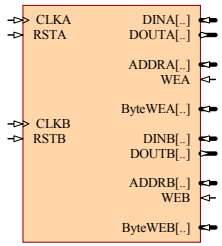
RAM(addr)=RAM contents at address ADDR

data=RAM input data

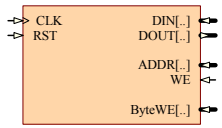
clk=Clock edge defined by Memory_ClockEdge parameter

RAMSRB, RAMDRB

Single/Dual Port Random Access Memory With Reset And Byte Write Enable



RAMDRB



RAMSRB

The RAMSRB and RAMDRB are single and dual port random access memory with reset and byte write enable. When the memory width spans multiple bytes, individual bytes of data can be accessed during the Read or Write cycles.

When reset (RST) is High, data output (DOUT) is cleared on the defined clock trigger.

When write enable (WE) is High, depending on the value of byte write enable (ByteWE), corresponding bytes of data input (DIN) can be transferred to memory on the defined clock trigger while reset is High or Low.

When reset (RST) and write enable (WE) is Low, data output (DOUT) is active, depending on the value of byte write enable (ByteWE), on the defined clock trigger and valid address (ADDR) input.

Data lengths and clock trigger is configurable by editing the following parameters found on the component properties:

Memory_Depth - This defines the depth of memory. It is set to DefinedBy=ADDR by default. This allows automatic configuration of ram depth size depending on the size of the address bus connected on the address (ADDR) input port.

Memory_Width - This defines the data port size. It is set to DefinedBy=DIN. This allows automatic configuration of ram data port size depending on the size of data bus connected to the data (DIN) input port.

Memory_ClockEdge - This parameter can be set to Rising or Falling depending on your desired clock trigger. It is set to Rising by default.

Memory_ContentFile – This enables the ram content to be initialized using a Hex (Intel format) file.

Inputs					Outputs	
RST	**WE**	**CLK**	**ADDR**	**DIN**	**DOUT**	**RAM Contents**
1	0	clk	X	X	0	No Chg
1	1	clk	addr	data	0	RAM(ByteWE(addr)) => data
0	0	clk	addr	X	RAM(ByteWE(addr))	No Chg
0	1	clk	addr	data	data	RAM(ByteWE(addr)) => data

addr=RAM address

RAM(addr)=RAM contents at address ADDR

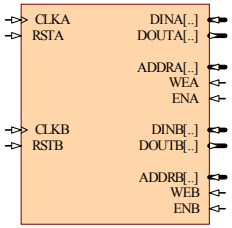
data=RAM input data

clk=Clock edge defined by Memory_ClockEdge parameter

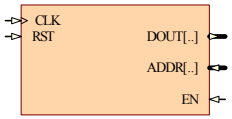
ByteWE=Select signal for individual bytes

RAMSRE, RAMDRE

Single/Dual Port Random Access Memory With Enable and Reset



RAMDRE



RAMSRE

The RAMSRE and RAMDRE are single and dual port random access memory with enable and Reset.

When enable (EN) is High, all control inputs are overridden. Data transfer is disabled and no change occurs at data output (DOUT).

When enable (EN) is Low and reset (RST) is High, data output (DOUT) is cleared on the defined clock trigger.

When write enable (WE) is High, data input (DIN) can be transferred to memory on the defined clock trigger while reset is High or Low.

When enable (EN), reset (RST) and write enable (WE) is Low, data output (DOUT) is active on the defined clock trigger and valid address (ADDR) input.

Data lengths and clock trigger is configurable by editing the following parameters found on the component properties:

Memory_Depth - This defines the depth of memory. It is set to DefinedBy=ADDR by default. This allows automatic configuration of ram depth size depending on the size of the address bus connected on the address (ADDR) input port.

Memory_Width - This defines the data port size. It is set to DefinedBy=DIN. This allows automatic configuration of ram data port size depending on the size of data bus connected to the data (DIN) input port.

Memory_ClockEdge - This parameter can be set to Rising or Falling depending on your desired clock trigger. It is set to Rising by default.

Memory_ContentFile – This enables the ram content to be initialized using a Hex (Intel format) file.

Inputs						Outputs	
EN	**RST**	**WE**	**CLK**	**ADDR**	**DIN**	**DOUT**	**RAM Contents**
1	X	X	X	X	X	No Chg	No Chg
0	1	0	clk	X	X	0	No Chg
0	1	1	clk	addr	data	0	RAM(addr) => data
0	0	0	clk	addr	X	RAM(addr)	No Chg
0	0	1	clk	addr	data	data	RAM(addr) => data

addr=RAM address

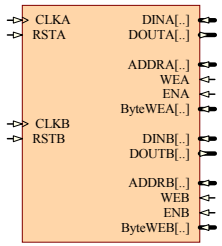
RAM(addr)=RAM contents at address ADDR

data=RAM input data

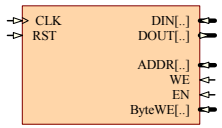
clk=Clock edge defined by Memory_ClockEdge parameter

RAMSREB, RAMDREB

Single/Dual Port Random Access Memory With Enable, Reset And Byte Write Enable



RAMDREB



RAMSREB

The RAMSREB and RAMDREB are single and dual port random access memory with enable, reset and byte write enable. When the memory width spans multiple bytes, individual bytes of data can be accessed during the Read or Write cycles.

When enable (EN) is High, all control inputs are overridden. Data transfer is disabled and no change occurs at data output (DOUT). When enable (EN) is Low and reset (RST) is High, data output (DOUT) is cleared on the defined clock trigger.

When write enable (WE) is High, depending on the value of byte write enable (ByteWE), corresponding bytes of data input (DIN) can be transferred to memory on the defined clock trigger while reset is High or Low.

When enable (EN), reset (RST) and write enable (WE) is Low, data output (DOUT) is active, depending on the value of byte write enable (ByteWE), on the defined clock trigger and valid address (ADDR) input.

Data lengths and clock trigger is configurable by editing the following parameters found on the component properties:

Memory_Depth - This defines the depth of memory. It is set to DefinedBy=ADDR by default. This allows automatic configuration of ram depth size depending on the size of the address bus connected on the address (ADDR) input port.

Memory_Width - This defines the data port size. It is set to DefinedBy=DIN. This allows automatic configuration of ram data port size depending on the size of data bus connected to the data (DIN) input port.

Memory_ClockEdge - This parameter can be set to Rising or Falling depending on your desired clock trigger. It is set to Rising by default.

Memory_ContentFile – This enables the ram content to be initialized using a Hex (Intel format) file.

Inputs						Outputs	
EN	RST	WE	CLK	ADDR	DIN	DOUT	RAM Contents
1	X	X	X	X	X	No Chg	No Chg
0	1	0	clk	X	X	0	No Chg
0	1	1	clk	addr	data	0	RAM(ByteWE(addr)) => data
0	0	0	clk	addr	X	RAM(ByteWE(addr))	No Chg
0	0	1	clk	addr	data	data	RAM(ByteWE(addr)) => data

addr=RAM address

RAM(addr)=RAM contents at address ADDR

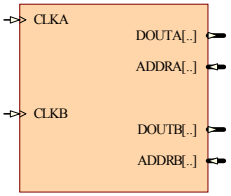
data=RAM input data

clk=Clock edge defined by Memory_ClockEdge parameter

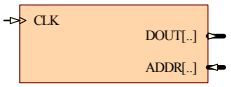
ByteWE=Select signal for individual bytes

ROMS, ROMD

Single/Dual Port Read Only Memory



ROMD



ROMS

ROMS and ROMD are single and dual port read only memory. Data output (DOUT) is always active on the defined clock trigger and valid address (ADDR) input.

Memory initialization, data lengths and clock trigger is configurable by editing the following parameter found on the component properties:

Memory_Depth - This defines the depth of memory. It is set to DefinedBy=ADDR by default. This allows automatic configuration of ram depth size depending on the size of the address bus connected on the address (ADDR) input port.

Memory_Width - This defines the data port size. It is set to DefinedBy=DOUT. This allows automatic configuration of ram data port size depending on the size of data bus connected to the data (DOUT) output port.

Memory_ClockEdge - This parameter can be set to Rising or Falling depending on your desired clock trigger. It is set to Rising by default.

Memory_ContentFile – This enables the rom content to be initialized using a Hex (Intel format) file.

Inputs		Outputs	
CLK	**ADDR**	**DOUT**	**ROM Contents**
clk	addr	ROM(addr)	No Chg

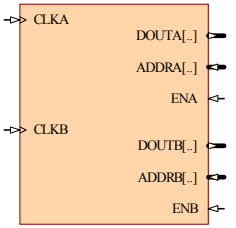
addr=ROM address

ROM(addr)=ROM contents at address ADDR

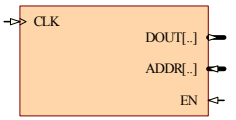
clk=Clock edge defined by Memory_ClockEdge parameter

ROMSE, ROMDE

Single/Dual Port Read Only Memory With Enable



ROMDE



ROMSE

ROMSE and ROMDE are single and dual port read only memory with enable. When enable (EN) is High, data transfer is disabled and no change occurs at data output (DOUT). When enable (EN) is Low, data output (DOUT) is always active on the defined clock trigger and valid address (ADDR) input.

Memory initialization, data lengths and clock trigger is configurable by editing the following parameter found on the component properties:

Memory_Depth - This defines the depth of memory. It is set to DefinedBy=ADDR by default. This allows automatic configuration of ram depth size depending on the size of the address bus connected on the address (ADDR) input port.

Memory_Width - This defines the data port size. It is set to DefinedBy=DOUT. This allows automatic configuration of ram data port size depending on the size of data bus connected to the data (DOUT) output port.

Memory_ClockEdge - This parameter can be set to Rising or Falling depending on your desired clock trigger. It is set to Rising by default.

Memory_ContentFile – This enables the rom content to be initialized using a Hex (Intel format) file.

Inputs			Outputs	
EN	**CLK**	**ADDR**	**DOUT**	**ROM Contents**
1	X	X	No Chg	No Chg
0	clk	addr	ROM(addr)	No Chg

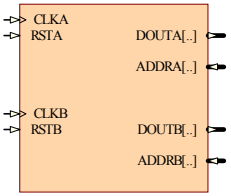
addr=ROM address

ROM(addr)=ROM contents at address ADDR

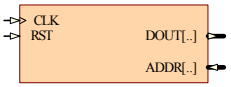
clk=Clock edge defined by Memory_ClockEdge parameter

ROMSR, ROMDR

Single/Dual Port Read Only Memory With Reset



ROMDR



ROMSR

ROMSR and ROMDR are single and dual port read only memory with reset. When reset (RST) is High, data output (DOUT) is cleared on the defined clock trigger. When reset (RST) is Low, data output (DOUT) is always active on the defined clock trigger and valid address (ADDR) input.

Memory initialization, data lengths and clock trigger is configurable by editing the following parameter found on the component properties:

Memory_Depth - This defines the depth of memory. It is set to DefinedBy=ADDR by default. This allows automatic configuration of ram depth size depending on the size of the address bus connected on the address (ADDR) input port.

Memory_Width - This defines the data port size. It is set to DefinedBy=DOUT. This allows automatic configuration of ram data port size depending on the size of data bus connected to the data (DOUT) output port.

Memory_ClockEdge - This parameter can be set to Rising or Falling depending on your desired clock trigger. It is set to Rising by default.

Memory_ContentFile – This enables the rom content to be initialized using a Hex (Intel format) file.

Inputs			Outputs	
RST	**CLK**	**ADDR**	**DOUT**	**ROM Contents**
1	clk	X	0	No Chg
0	clk	addr	ROM(addr)	No Chg

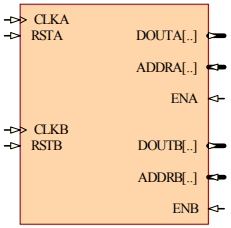
addr=ROM address

ROM(addr)=ROM contents at address ADDR

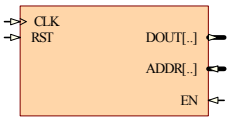
clk=Clock edge defined by Memory_ClockEdge parameter

ROMSRE, ROMDRE

Single/Dual Port Read Only Memory With Enable and Reset



ROMDRE



ROMSRE

ROMSE and ROMDE are single and dual port read only memory with enable. When enable (EN) is High, all control inputs are overridden. Data transfer is disabled and no change occurs at data output (DOUT). When enable is Low and reset (RST) is High, data output (DOUT) is cleared on the defined clock trigger.

When enable (EN) and reset (RST) is Low, data output (DOUT) is always active on the defined clock trigger and valid address (ADDR) input.

Memory initialization, data lengths and clock trigger is configurable by editing the following parameter found on the component properties:

Memory_Depth - This defines the depth of memory. It is set to DefinedBy=ADDR by default. This allows automatic configuration of ram depth size depending on the size of the address bus connected on the address (ADDR) input port.

Memory_Width - This defines the data port size. It is set to DefinedBy=DOUT. This allows automatic configuration of ram data port size depending on the size of data bus connected to the data (DOUT) output port.

Memory_ClockEdge - This parameter can be set to Rising or Falling depending on your desired clock trigger. It is set to Rising by default.

Memory_ContentFile – This enables the rom content to be initialized using a Hex (Intel format) file.

Inputs				Outputs	
EN	**RST**	**CLK**	**ADDR**	**DOUT**	**ROM Contents**
1	X	X	X	No Chg	No Chg
0	1	clk	X	0	No Chg
0	0	clk	addr	ROM(addr)	No Chg

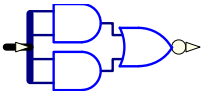
addr=ROM address

ROM(addr)=ROM contents at address ADDR

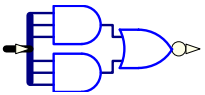
clk=Clock edge defined by Memory_ClockEdge parameter

SOP2_2, SOP2_3, SOP2_4, SOP4_2

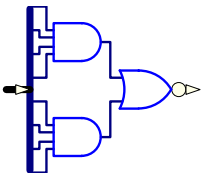
Sum of Products



SOP2_2B



SOP2_3B



SOP2_4B

Sum Of Products (SOP) provide common logic functions by NOR-gating the outputs of two or four AND functions. The function of each SOPs are described by the following equations:

SOP2_2

$$Y = \overline{(I0 \bullet I1) + (I2 \bullet I3)}$$

SOP2_3

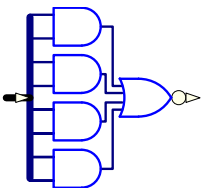
$$Y = \overline{(I0 \bullet I1 \bullet I2) + (I3 \bullet I4 \bullet I5)}$$

SOP2_4

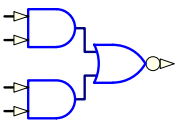
$$Y = \overline{(I0 \bullet I1 \bullet I2 \bullet I3) + (I4 \bullet I5 \bullet I6 \bullet I7)}$$

SOP4_2

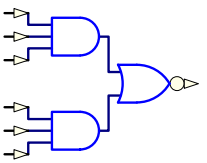
$$Y = \overline{(I0 \bullet I1) + (I2 \bullet I3) + (I4 \bullet I5) + (I6 \bullet I7)}$$



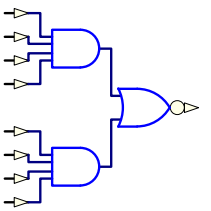
SOP4_2B



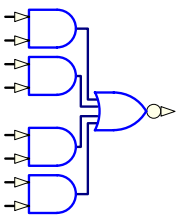
SOP2_2S



SOP2_3S



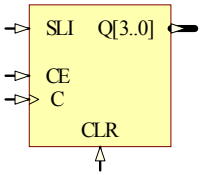
SOP2_4S



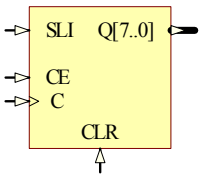
SOP4_2S

SR4CE, SR8CE, SR16CE, SR32CE

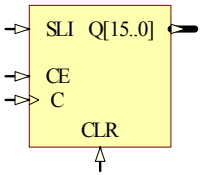
Serial-In Parallel-Out Shift Registers with Clock Enable and Asynchronous Clear



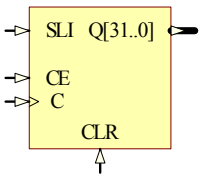
SR4CEB



SR8CEB



SR16CEB



SR32CEB

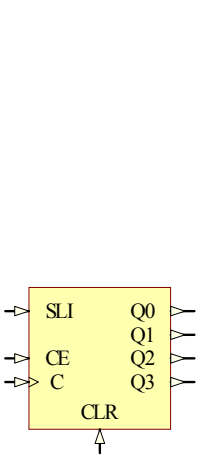
SR4CE, SR8CE, SR16CE and SR32CE are respectively 4-bit, 8-bit, 16-bit and 32-bit shift registers, with a shift-left serial input (SLI), parallel outputs (Q), clock enable (CE) and asynchronous clear (CLR) inputs. When High, the CLR input overrides all other inputs and resets the data outputs (Q) to logic level zero. When CE is High and CLR is Low, the data on the SLI input is loaded into the first bit of the shift register during the Low-to-High clock (C) transition and appears on the Q0 output. During subsequent Low-to-High clock transitions, when CE is High and CLR is Low, data is shifted to the next highest bit position as new data is loaded into Q0 (That is, SLI → Q0, Q0 → Q1, Q1 → Q2, and so forth). The register ignores clock transitions when CE is Low.

Registers can be cascaded by connecting the last Q output (Q3 for SR4CE, Q7 for SR8CE, Q15 for SR16CE, or Q31 for SR32CE) of one stage to the SLI input of the next stage and connecting clock, CE, and CLR in parallel.

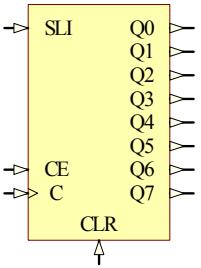
Inputs				Outputs	
CLR	**CE**	**SLI**	**C**	**Q0**	**Qz – Q1**
1	X	X	X	0	0
0	0	X	X	No Chg	No Chg
0	1	1	↑	1	qn-1
0	1	0	↑	0	qn-1

z = 3 for SR4CE; z = 7for SR8CE; z = 15 for SR16CE; z = 31 for SR32CE

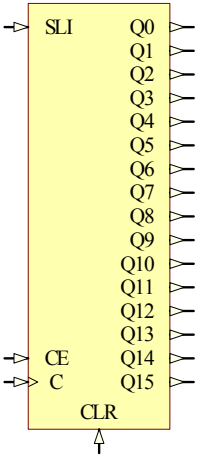
qn-1 = state of referenced output one setup time prior to active clock transition



SR4CES



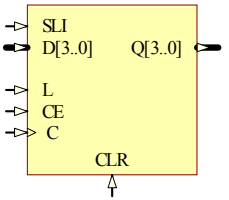
SR8CES



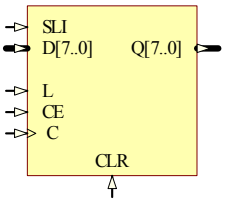
SR16CES

SR4CLE, SR8CLE, SR16CLE, SR32CLE

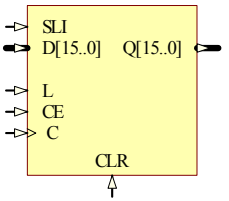
Loadable Serial/Parallel-In Parallel-Out Shift Registers with Clock Enable and Asynchronous Clear



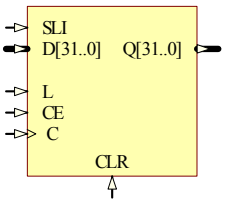
SR4CLEB



SR8CLEB



SR16CLEB



SR32CLEB

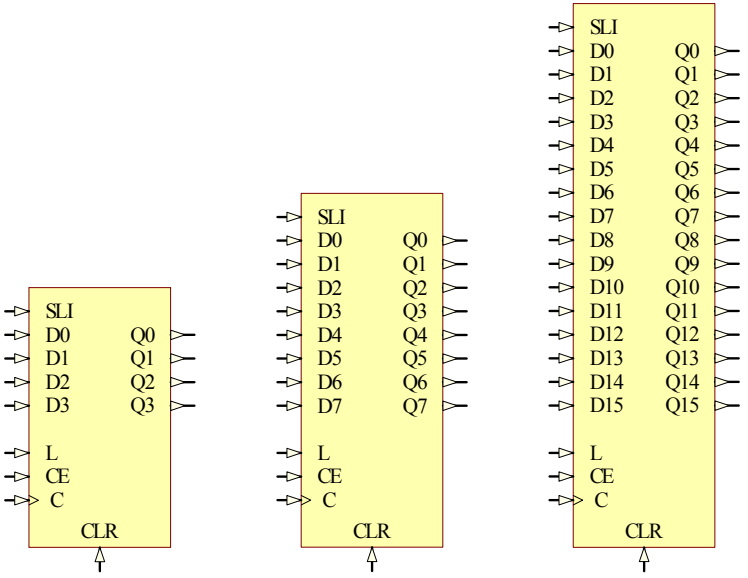
SR4CLE, SR8CLE, SR16CLE and SR32CLE are respectively 4-bit, 8-bit, 16-bit and 32-bit shift registers with a shift-left serial input (SLI), parallel inputs (D), parallel outputs (Q), and three control inputs: clock enable (CE), load enable (L), and asynchronous clear (CLR). The register ignores clock transitions when L and CE are Low. When High, the asynchronous CLR pin overrides all other inputs and resets the data outputs (Q) to logic level zero. When L is High and CLR is Low, data on the Dn – D0 inputs is loaded into the corresponding Qn – Q0 bits of the register. When CE is High and L and CLR are Low, data on the SLI input is loaded into the first bit of the shift register during the Low-to-High clock (C) transition and appears on the Q0 output. During subsequent clock transitions, when CE is High and L and CLR are Low, the data is shifted to the next highest bit position as new data is loaded into Q0 (That is, SLI → Q0, Q0 → Q1, Q1 → Q2, and so forth).

Registers can be cascaded by connecting the last Q output (Q3 for SR4CLE, Q7 for SR8CLE, Q15 for SR16CLE, or Q31 for SR32CLE) of one stage to the SLI input of the next stage and connecting clock, CE, L, and CLR inputs in parallel.

Inputs						Outputs	
CLR	**L**	**CE**	**SLI**	**Dn – D0**	**C**	**Q0**	**Qz – Q1**
1	X	X	X	X	X	0	0
0	1	X	X	Dn – D0	↑	D0	Dn
0	0	1	SLI	X	↑	SLI	qn-1
0	0	0	X	X	X	No Chg	No Chg

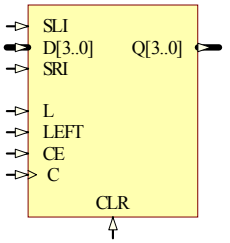
z = 3 for SR4CLE; z = 7 for SR8CLE; z = 15 for SR16CLE; z = 31 for SR32CLE

qn-1 = state of referenced output one setup time prior to active clock transition

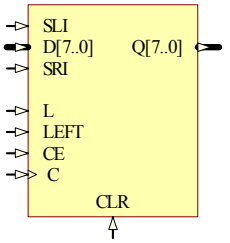

SLI
D0
D1
D2
D3
Q0
Q1
Q2
Q3
L
CE
C
CLR
SR4CLES
SLI
D0
D1
D2
D3
D4
D5
D6
D7
Q0
Q1
Q2
Q3
Q4
Q5
Q6
Q7
L
CE
C
CLR
SR8CLES
SLI
D0
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
D14
D15
Q0
Q1
Q2
Q3
Q4
Q5
Q6
Q7
Q8
Q9
Q10
Q11
Q12
Q13
Q14
Q15
L
CE
C
CLR
SR16CLES

SR4CLED, SR8CLED, SR16CLED, SR32CLED

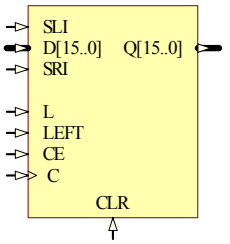
Loadable Serial/Parallel-In Parallel-Out Bidirectional Shift Registers with Clock Enable and Asynchronous Clear



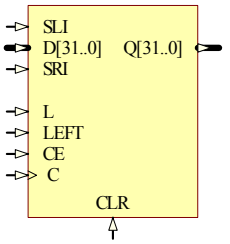
SR4CLEDB



SR8CLEDB



SR16CLEDB



SR32CLEDB

SR4CLED, SR8CLED, SR16CLED and SR32CLED are respectively 4-bit, 8-bit, 16- and 32-bit shift registers with shift-left (SLI) and shift-right (SRI) serial inputs, parallel inputs (D), parallel outputs (Q), and four control inputs: clock enable (CE), load enable (L), shift left/right (LEFT), and asynchronous clear (CLR). The register ignores clock transitions when CE and L are Low. When High, the asynchronous clear overrides all other inputs and resets the data outputs (Qn) to logic level zero. When L is High and CLR is Low, the data on the D inputs is loaded into the corresponding Q bits of the register. When CE is High and L and CLR are Low, data is shifted right or left, depending on the state of the LEFT input. If LEFT is High, data on the SLI is loaded into Q0 during the Low-to-High clock transition and shifted left (to Q1, Q2, and so forth) during subsequent clock transitions. If LEFT is Low, data on the SRI is loaded into the last Q output (Q3 for SR4CLED, Q7 for SR8CLED, Q15 for SR16CLED, or Q31 for SR32CLED) during the Low-to-High clock transition and shifted right (to Q2, Q1,... for SR4CLED; to Q6, Q5,... for SR8CLED; to Q14, Q13,... for SR16CLED and to Q30, Q29,... for SR32CLED) during subsequent clock transitions.

SR4CLED Truth Table

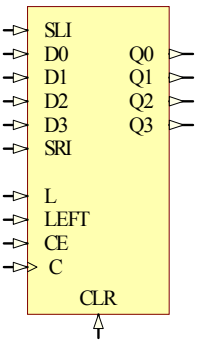
Inputs									Outputs	
CLR	L	CE	LEFT	SLI	SRI	D3 – D0	C	Q0	Q3	Q2 – Q1
1	X	X	X	X	X	X	X	0	0	0
0	1	X	X	X	X	D3– D0	↑	D0	D3	Dn
0	0	0	X	X	X	X	X	No Chg	No Chg	No Chg
0	0	1	1	SLI	X	X	↑	SLI	q2	qn-1
0	0	1	0	X	SRI	X	↑	q1	SRI	qn+1

qn-1 and qn+1 = state of referenced output one setup time prior to active clock transition

SR8CLED Truth Table

Inputs								Outputs		
CLR	L	CE	LEFT	SLI	SRI	D7 – D0	C	Q0	Q7	Q6 – Q1
1	X	X	X	X	X	X	X	0	0	0
0	1	X	X	X	X	D7 – D0	↑	D0	D7	Dn
0	0	0	X	X	X	X	X	No Chg	No Chg	No Chg
0	0	1	1	SLI	X	X	↑	SLI	q6	qn-1
0	0	1	0	X	SRI	X	↑	q1	SRI	qn+1

qn-1 or qn+1 = state of referenced output one setup time prior to active clock



SR4CLEDS

transition

SR16CLED Truth Table

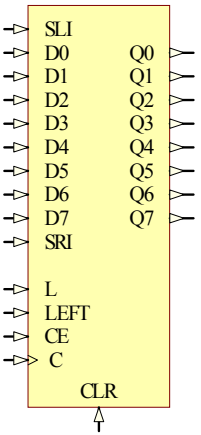
Inputs								Outputs		
CLR	**L**	**CE**	**LEFT**	**SLI**	**SRI**	**D15 – D0**	**C**	**Q0**	**Q15**	**Q14 – Q1**
1	X	X	X	X	X	X	X	0	0	0
0	1	X	X	X	X	D15 – D0	↑	D0	D15	Dn
0	0	0	X	X	X	X	X	No Chg	No Chg	No Chg
0	0	1	1	SLI	X	X	↑	SLI	q14	qn-1
0	0	1	0	X	SRI	X	↑	q1	SRI	qn+1

qn-1 or qn+1 = state of referenced output one setup time prior to active clock transition

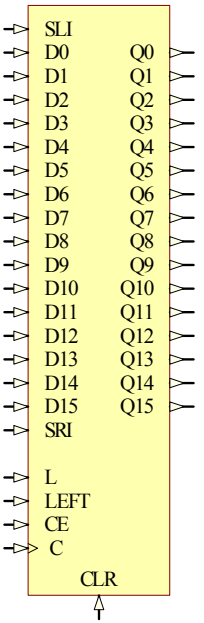
SR32CLED Truth Table

Inputs								Outputs		
CLR	**L**	**CE**	**LEFT**	**SLI**	**SRI**	**D31 – D0**	**C**	**Q0**	**Q31**	**Q30 – Q1**
1	X	X	X	X	X	X	X	0	0	0
0	1	X	X	X	X	D31 – D0	↑	D0	D31	Dn
0	0	0	X	X	X	X	X	No Chg	No Chg	No Chg
0	0	1	1	SLI	X	X	↑	SLI	q30	qn-1
0	0	1	0	X	SRI	X	↑	q1	SRI	qn+1

qn-1 or qn+1 = state of referenced output one setup time prior to active clock transition



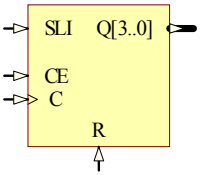
SR8CLEDS



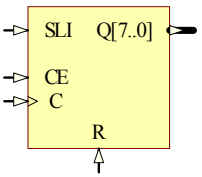
SR16CLEDS

SR4RE, SR8RE, SR16RE, SR32RE

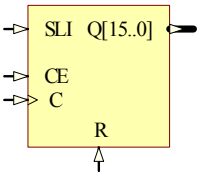
Serial-In Parallel-Out Shift Registers with Clock Enable and Synchronous Reset



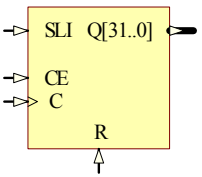
SR4REB



SR8REB



SR16REB



SR32REB

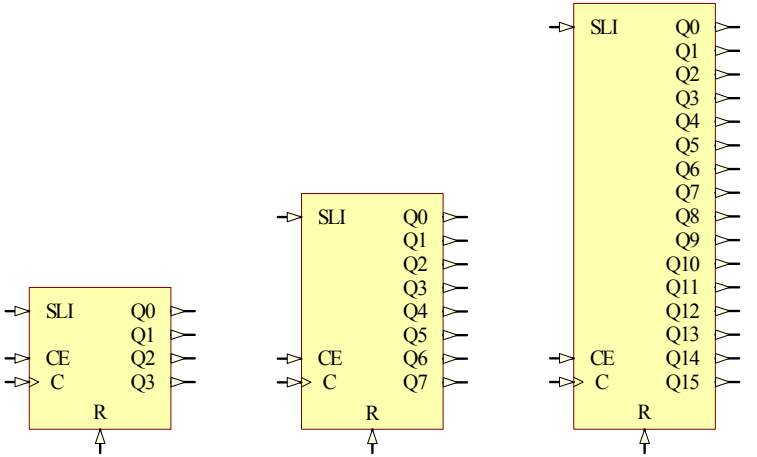
SR4RE, SR8RE, SR16RE and SR32RE are respectively 4-bit, 8-bit, 16-bit and 32-bit shift registers with shift-left serial input (SLI), parallel outputs (Qn), clock enable (CE), and synchronous reset (R) inputs. The R input, when High, overrides all other inputs during the Low-to-High clock (C) transition and resets the data outputs (Q) to logic level zero. When CE is High and R is Low, the data on the SLI is loaded into the first bit of the shift register during the Low-to-High clock (C) transition and appears on the Q0 output. During subsequent Low-to-High clock transitions, when CE is High and R is Low, data is shifted to the next highest bit position as new data is loaded into Q0 (That is, SL → Q0, Q → Q1, Q → Q2, and so forth). The register ignores clock transitions when CE is Low.

Registers can be cascaded by connecting the last Q output (Q3 for SR4RE, Q7 for SR8RE, Q15 for SR16RE or Q31 for SR32RE) of one stage to the SLI input of the next stage and connecting clock, CE, and R in parallel.

Inputs				Outputs	
R	**CE**	**SLI**	**C**	**Q0**	**Qz – Q1**
1	X	X	↑	0	0
0	0	X	X	No Chg	No Chg
0	1	1	↑	1	qn-1
0	1	0	↑	0	qn-1

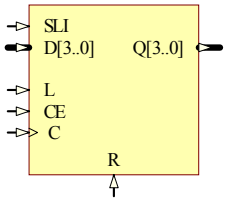
z = 3 for SR4RE; z = 7 for SR8RE; z = 15 for SR16RE; z = 31 for SR32RE

qn-1 = state of referenced output one setup time prior to active clock transition

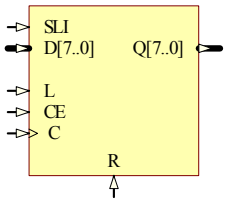

SLI
Q0
Q1
CE
Q2
C
Q3
R
SR4RES
SLI
Q0
Q1
Q2
Q3
Q4
Q5
CE
Q6
C
Q7
R
SR8RES
SLI
Q0
Q1
Q2
Q3
Q4
Q5
Q6
Q7
Q8
Q9
Q10
Q11
Q12
Q13
CE
Q14
C
Q15
R
SR16RES

SR4RLE, SR8RLE, SR16RLE, SR32RLE

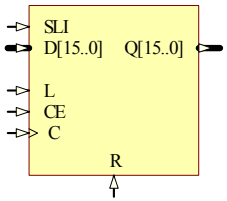
Loadable Serial/Parallel-In Parallel-Out Shift Registers with Clock Enable and Synchronous Reset



SR4RLEB



SR8RLEB



SR16RLEB

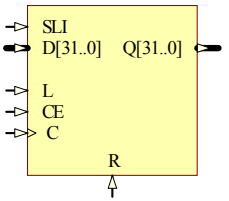
SR4RLE, SR8RLE, SR16RLE and SR32RLE are respectively 4-bit, 8-bit, 16-bit and 32-bit shift registers with shift-left serial input (SLI), parallel inputs (D), parallel outputs (Q), and three control inputs: clock enable (CE), load enable (L), and synchronous reset (R). The register ignores clock transitions when L and CE are Low. When High the synchronous reset R overrides all other inputs during the Low-to-High clock (C) transition and resets the data outputs (Q) to logic level zero. When L is High and R is Low during the Low-to-High clock transition, data on the D inputs is loaded into the corresponding Q bits of the register. When CE is High and L and R are Low, data on the SLI input is loaded into the first bit of the shift register during the Low-to-High clock (C) transition and appears on the Q0 output. During subsequent clock transitions, when CE is High and L and R are Low, the data is shifted to the next highest bit position as new data is loaded into Q0 (That is, SL → Q0, Q → Q1, Q → Q2, and so forth).

Registers can be cascaded by connecting the last Q output (Q3 for SR4RLE, Q7 for SR8RLE, 15 for SR16RLE or 31 for SR32RLE) of one stage to the SLI input of the next stage and connecting clock, CE, L, and R inputs in parallel.

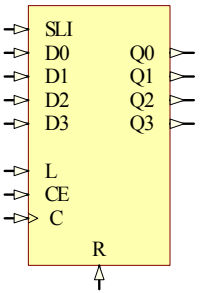
Inputs						Outputs	
R	**L**	**CE**	**SLI**	**Dz – D0**	**C**	**Q0**	**Qz – Q1**
1	X	X	X	X	↑	0	0
0	1	X	X	Dz – D0	↑	D0	Dn
0	0	1	SLI	X	↑	SLI	qn-1
0	0	0	X	X	X	No Chg	No Chg

z = 3 for SR4RLE; z = 7 for SR8RLE; z = 15 for SR16RLE; z = 31 for SR32RLE

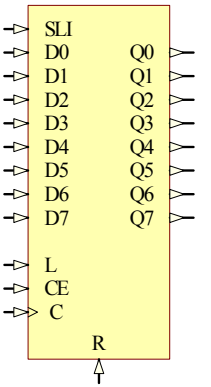
qn-1 = state of referenced output one setup time prior to active clock transition



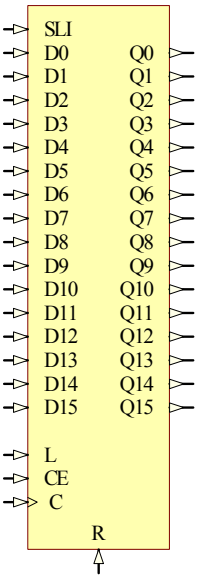
SR32RLEB



SR4RLES



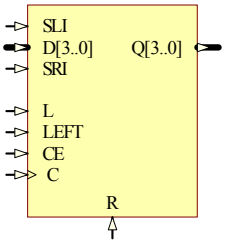
SR8RLES



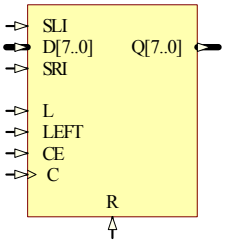
SR16RLES

SR4RLED, SR8RLED, SR16RLED, SR32RLED

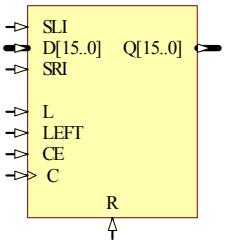
Loadable Serial/Parallel-In Parallel-Out Bidirectional Shift Registers with Clock Enable and Synchronous Reset



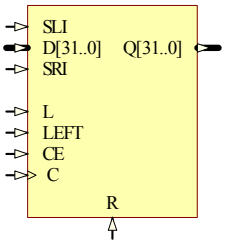
SR4RLEDB



SR8RLEDB



SR16RLEDB



SR32RLEDB

SR4RLED, SR8RLED SR16RLED and SR32RLED are respectively 4-bit, 8-bit, 16-bit and 32-bit shift registers with shift-left (SLI) and shift-right (SRI) serial inputs, parallel inputs (D), parallel outputs (Q) and four control inputs - clock enable (CE), load enable (L), shift left/right (LEFT), and synchronous reset (R). The register ignores clock transitions when CE and L are Low. When High, the synchronous reset R overrides all other inputs during the Low-to-High clock (C) transition and resets the data outputs (Q) to logic level zero. When L is High and R is Low during the Low-to-High clock transition, the data on the D inputs is loaded into the corresponding Q bits of the register. When CE is High and L and R are Low, data is shifted right or left, depending on the state of the LEFT input. If LEFT is High, data on SLI is loaded into Q0 during the Low-to-High clock transition and shifted left (to Q1, Q2, and so forth) during subsequent clock transitions. If LEFT is Low, data on the SRI is loaded into the last Q output (Q3 for SR4RLED, Q7 for SR8RLED, Q15 for SR16RLED or Q31 for SR32RLED) during the Low-to-High clock transition and shifted right (to Q2, Q1,... for SR4RLED; to Q6, Q5,... for SR8RLED; to Q14, Q13,... for SR16RLED or to Q30, Q29,... for SR32RLED) during subsequent clock transitions.

SR4RLED

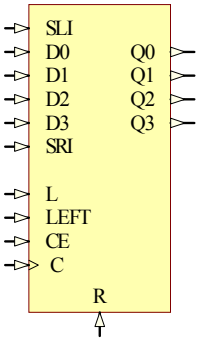
Inputs								Outputs		
R	L	CE	LEFT	SLI	SRI	D3 – D0	C	Q0	Q3	Q2 – Q1
1	X	X	X	X	X	X	↑	0	0	0
0	1	X	X	X	X	D3 – D0	↑	D0	D3	Dn
0	0	0	X	X	X	X	X	No Chg	No Chg	No Chg
0	0	1	1	SLI	X	X	↑	SLI	q2	qn-1
0	0	1	0	X	SRI	X	↑	q1	SRI	qn+1

qn-1 or qn+1 = state of referenced output one setup time prior to active clock transition

SR8RLED

Inputs								Outputs		
R	L	CE	LEFT	SLI	SRI	D7– D0	C	Q0	Q7	Q6 – Q1
1	X	X	X	X	X	X	↑	0	0	0
0	1	X	X	X	X	D7 – D0	↑	D0	D7	Dn
0	0	0	X	X	X	X	X	No Chg	No Chg	No Chg
0	0	1	1	SLI	X	X	↑	SLI	q6	qn-1
0	0	1	0	X	SRI	X	↑	q1	SRI	qn+1

qn-1 or qn+1 = state of referenced output one setup time prior to active clock



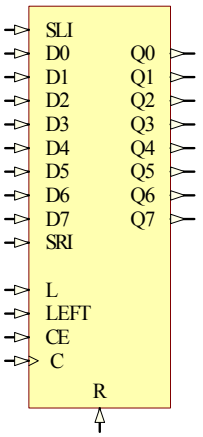
SR4RLEDS

transition

SR16RLED

Inputs								Outputs		
R	**L**	**CE**	**LEFT**	**SLI**	**SRI**	**D15 – D0**	**C**	**Q0**	**Q15**	**Q14 – Q1**
1	X	X	X	X	X	X	↑	0	0	0
0	1	X	X	X	X	D15 – D0	↑	D0	D15	Dn
0	0	0	X	X	X	X	X	No Chg	No Chg	No Chg
0	0	1	1	SLI	X	X	↑	SLI	q14	qn-1
0	0	1	0	X	SRI	X	↑	q1	SRI	qn+1

qn-1 or qn+1 = state of referenced output one setup time prior to active clock transition

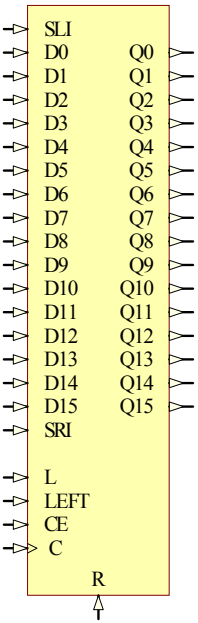


SR8RLEDS

SR32RLED

Inputs								Outputs		
R	**L**	**CE**	**LEFT**	**SLI**	**SRI**	**D31 – D0**	**C**	**Q0**	**Q31**	**Q30 – Q1**
1	X	X	X	X	X	X	↑	0	0	0
0	1	X	X	X	X	D31 – D0	↑	D0	D31	Dn
0	0	0	X	X	X	X	X	No Chg	No Chg	No Chg
0	0	1	1	SLI	X	X	↑	SLI	q30	qn-1
0	0	1	0	X	SRI	X	↑	q1	SRI	qn+1

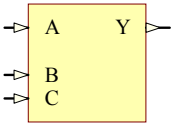
qn-1 or qn+1 = state of referenced output one setup time prior to active clock transition


SLI
D0
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
D14
D15
SRI
L
LEFT
CE
C
R
Q0
Q1
Q2
Q3
Q4
Q5
Q6
Q7
Q8
Q9
Q10
Q11
Q12
Q13
Q14
Q15

SR16RLEDS

TCZO

True/Complement, Zero/One Element



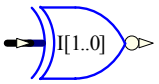
TCZO

TCZO can output a true signal of A (A), a complement of A ($\overline{A}$), a zero ('0') or a one ('1') to Y depending on the state of B and C.

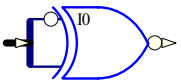
Inputs		Output
B	**C**	**Y**
0	0	$\overline{A}$
0	1	A
1	0	1
1	1	0

XNOR2 – 32

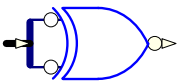
XNOR (Exclusive-NOR) Gates



XNOR2B

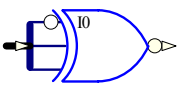


XNOR2N1B



XNOR2N2B

XNOR3B



XNOR3N1B

XNOR Gates provide a variety of XNOR functions, range from 2 to 32 inverted or non-inverted Inputs.

XNOR*n* – *n*-bit non-inverted input XNOR gates

If there is odd number of Low in the inputs, the output Y will be High, otherwise set to Low.

The truth table can be derived form the following equation:

$$O = \overline{I0 \oplus I1 ... \oplus In-1}$$

n = 2, 3, 4, 5, 6, 7, 8, 9, 12, 16, 32

XOR*n*N*m* – *n*-bit input, *m*-bit inverted XNOR gates

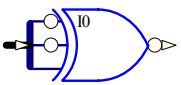
The truth table can be derived from the following equation:

$$O = \overline{\overline{I0} \oplus \overline{I1} \oplus ... \oplus \overline{Im-1} \oplus Im \oplus ... \oplus In-1}$$

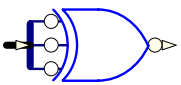
m, *n* = 2, 3, 4, 5; *m* <= *n*.

XNOR*n* – *n*-bit non-inverted input XNOR gates

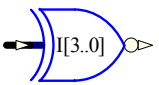
Input	Output
I0 … I*n-1*	**O**
odd number of 1	0
even number of 1	1



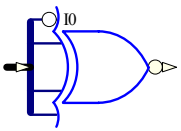
XNOR3N2B



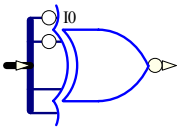
XNOR3N3B



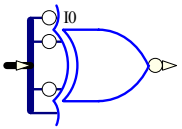
XNOR4B



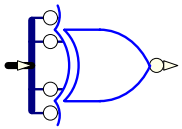
XNOR4N1B



XNOR4N2B

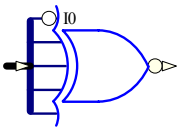


XNOR4N3B

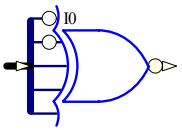


XNOR4N4B

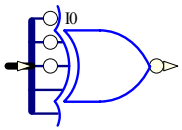
XNOR5B



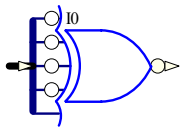
XNOR5N1B



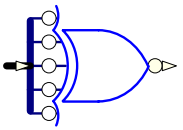
XNOR5N2B



XNOR5N3B



XNOR5N4B



XNOR5N5B

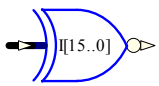
XNOR6B

XNOR7B

XNOR8B

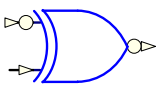
XNOR9B

XNOR12B

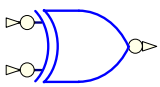


XNOR16B

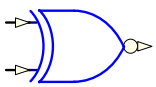
XNOR32B



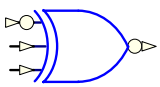
XNOR2N1S



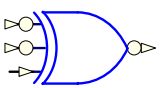
XNOR2N2S



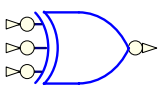
XNOR2S



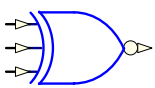
XNOR3N1S



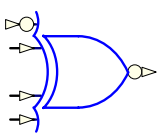
XNOR3N2S



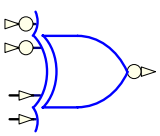
XNOR3N3S



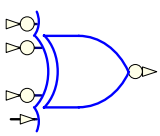
XNOR3S



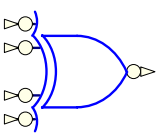
XNOR4N1S



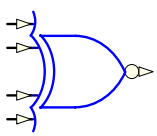
XNOR4N2S



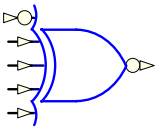
XNOR4N3S



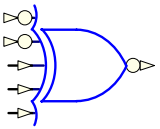
XNOR4N4S



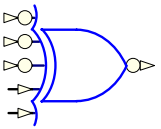
XNOR4S



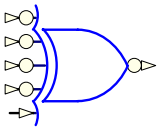
XNOR5N1S



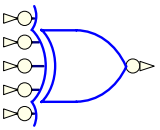
XNOR5N2S



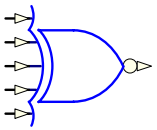
XNOR5N3S



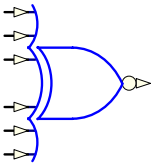
XNOR5N4S



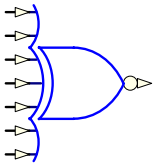
XNOR5N5S



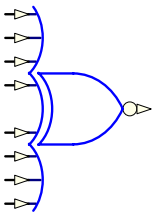
XNOR5S



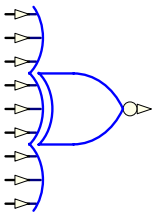
XNOR6S



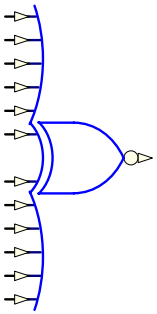
XNOR7S



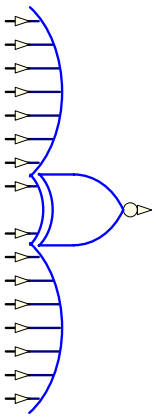
XNOR8S



XNOR9S



XNOR12S



XNOR16S

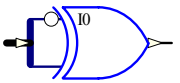
XOR2 – 32

XOR (Exclusive-OR) Gates

XOR2B

XOR Gates provide a variety of XOR functions, range from 2 to 32 inverted or non-inverted Inputs.

XOR*n* – *n*-bit non-inverted inputs XOR gates

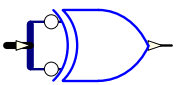


XOR2N1B

If there is odd number of High in the inputs, the output (Y) will be High, otherwise set to Low

The truth table can be derived from the following equation:

$$O = I0 \oplus I1 \ldots \oplus In - 1$$



XOR2N2B

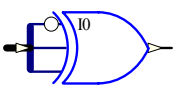
n = 2, 3, 4, 5, 6, 7, 8, 9, 12, 16, 32

XOR3B

XOR*n*N*m* – *n*-bit input, *m*-bit inverted XOR Gates

The truth table can be derived from the following equation:

$$O = \overline{I0} \oplus \overline{I1} \oplus \ldots \oplus \overline{Im - 1} \oplus Im \oplus \ldots \oplus In - 1$$

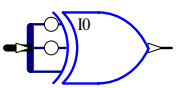


XOR3N1B

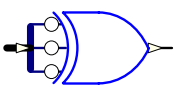
m, *n* = 2, 3, 4, 5, *m* <= *n*.

XOR*n* – *n*-bit non-inverted inputs XOR gates

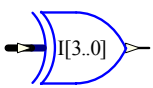
Input	Output
I0 … I*n-1*	**O**
odd number of 1	1
even number of 1	0



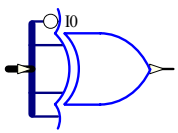
XOR3N2B



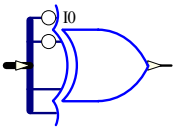
XOR3N3B



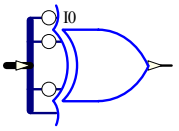
XOR4B



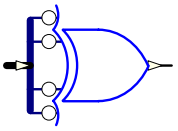
XOR4N1B



XOR4N2B

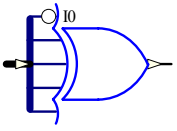


XOR4N3B

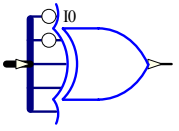


XOR4N4B

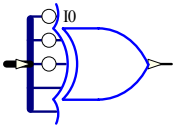
XOR5B



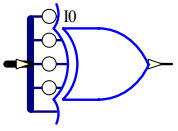
XOR5N1B



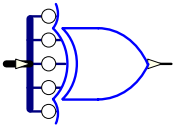
XOR5N2B



XOR5N3B



XOR5N4B



XOR5N5B

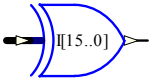
XOR6B

XOR7B

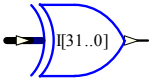
XOR8B

XOR9B

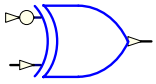
XOR12B



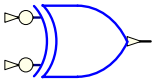
XOR16B



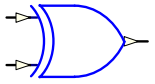
XOR32B



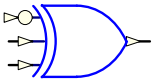
XOR2N1S



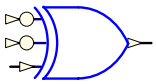
XOR2N2S



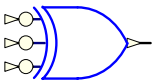
XOR2S



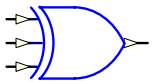
XOR3N1S



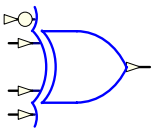
XOR3N2S



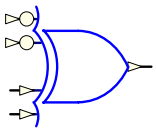
XOR3N3S



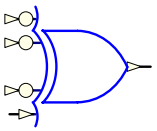
XOR3S



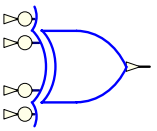
XOR4N1S



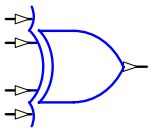
XOR4N2S



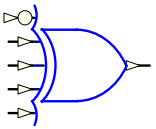
XOR4N3S



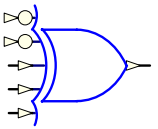
XOR4N4S



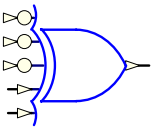
XOR4S



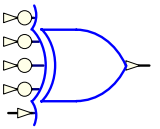
XOR5N1S



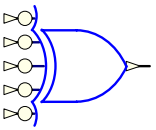
XOR5N2S



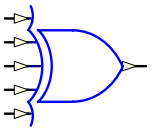
XOR5N3S



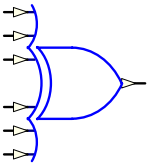
XOR5N4S



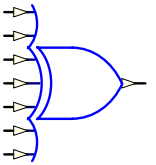
XOR5N5S



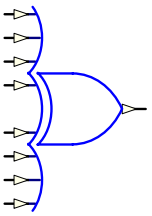
XOR5S



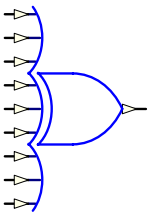
XOR6S



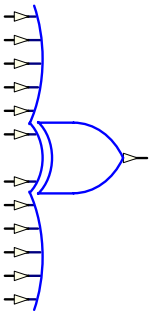
XOR7S



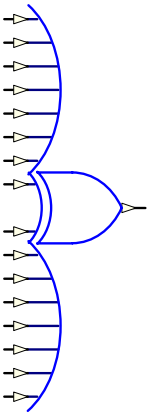
XOR8S



XOR9S



XOR12S



XOR16S

Revision History

Date	Version No.	Revision
25-Jan-2004	1.0	New product release
6-May-2004	2.0	Service pack 1 release • Details new components for 32-bit versions plus additions to Arithmetic Function, Decoder, Encoder, Memory, Multiplexer and Shifter. • Various naming conventions, page titles, descriptions, truth-tables and symbols revised.
9-Dec-2004	2.2	Service pack 2 release New components added: CLKMAN_1, 2, 3, 4
29-Apr-2005	2.201	Demultiplexer output states changed from 'Don't Care' to 'Low'
6-Jun-2005	2.202	Service pack 4 release Byte addressable RAMs added Demultiplexer symbols changed