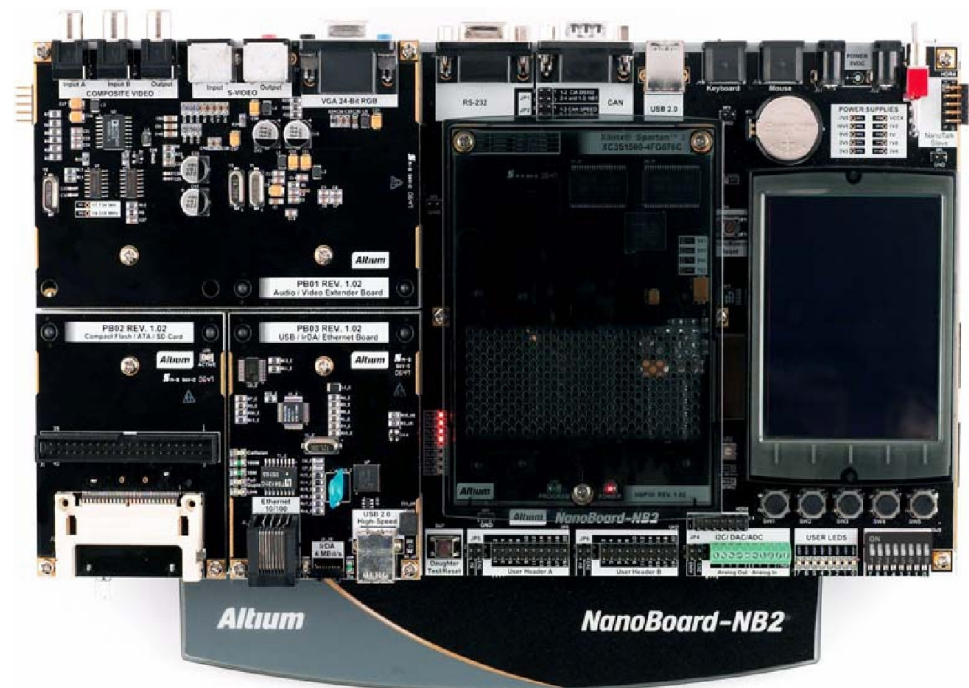


Projet IMA3- Filières Systèmes communicants

Tutoriel:

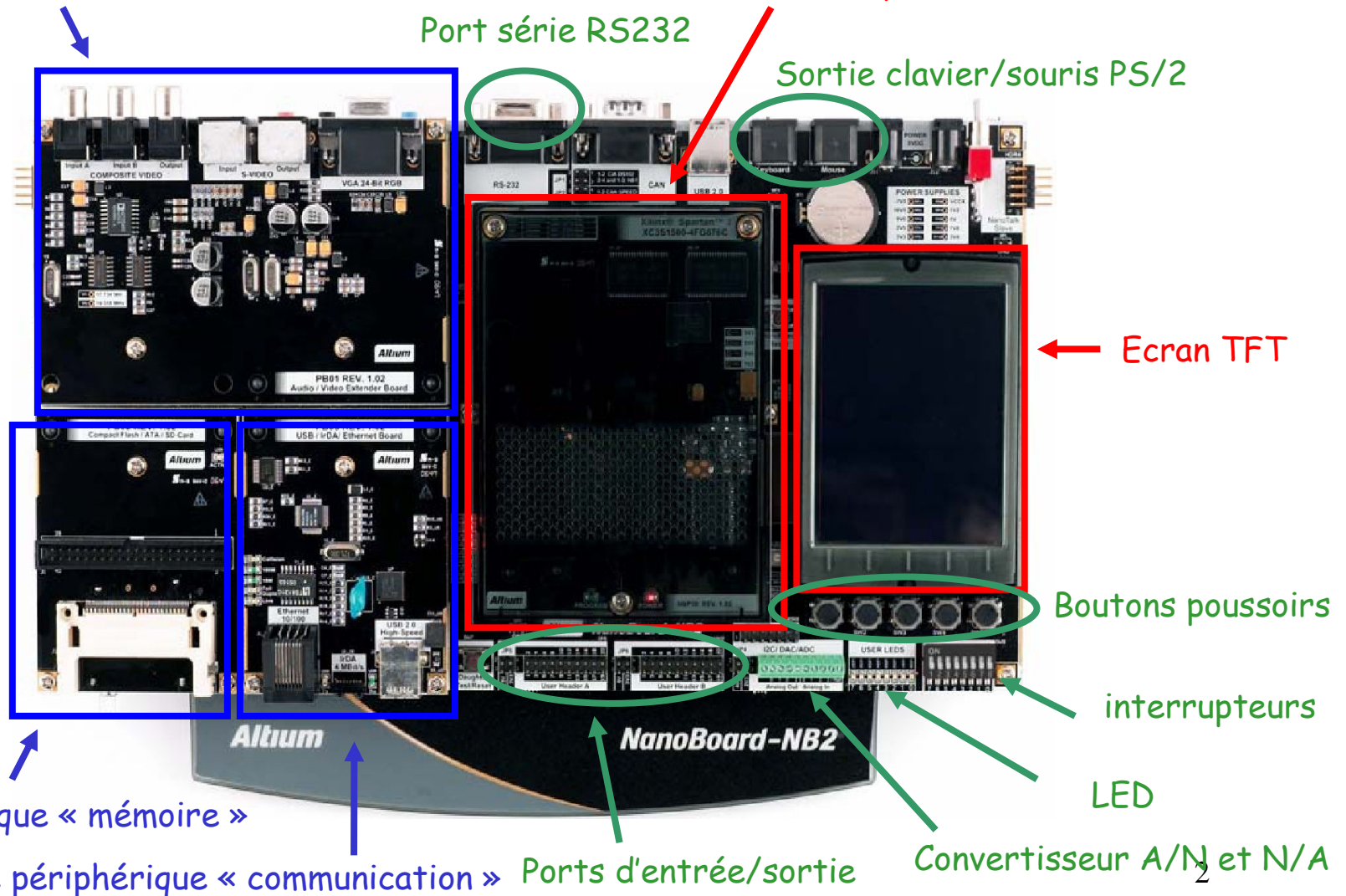
programmation FPGA sur la plateforme « Nanoboard » d'Altium



Plateforme de prototypage d'Altium: NanoBoard-NB2

Carte périphérique « audio-video »

Carte fille possédant le FPGA



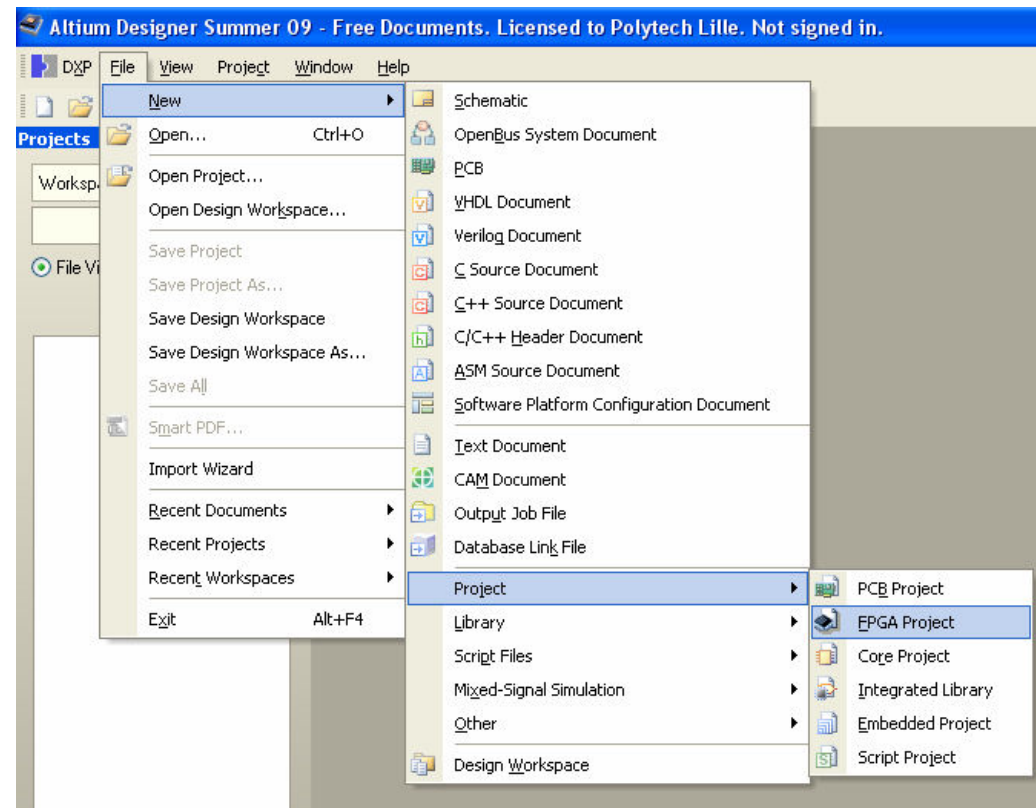
*Un FPGA est un composant numérique reconfigurable par programmation composé de différents circuits logiques que l'on peut interconnecter.

*La sélection des circuits logiques utilisés ainsi que leur interconnexion est faite à l'aide du logiciel de conception, *Altium Designer*

*Après conception du système, le « programme » est implémenté dans le FPGA puis testé à l'aide des différents périphériques inclus sur la *nanoboard*.

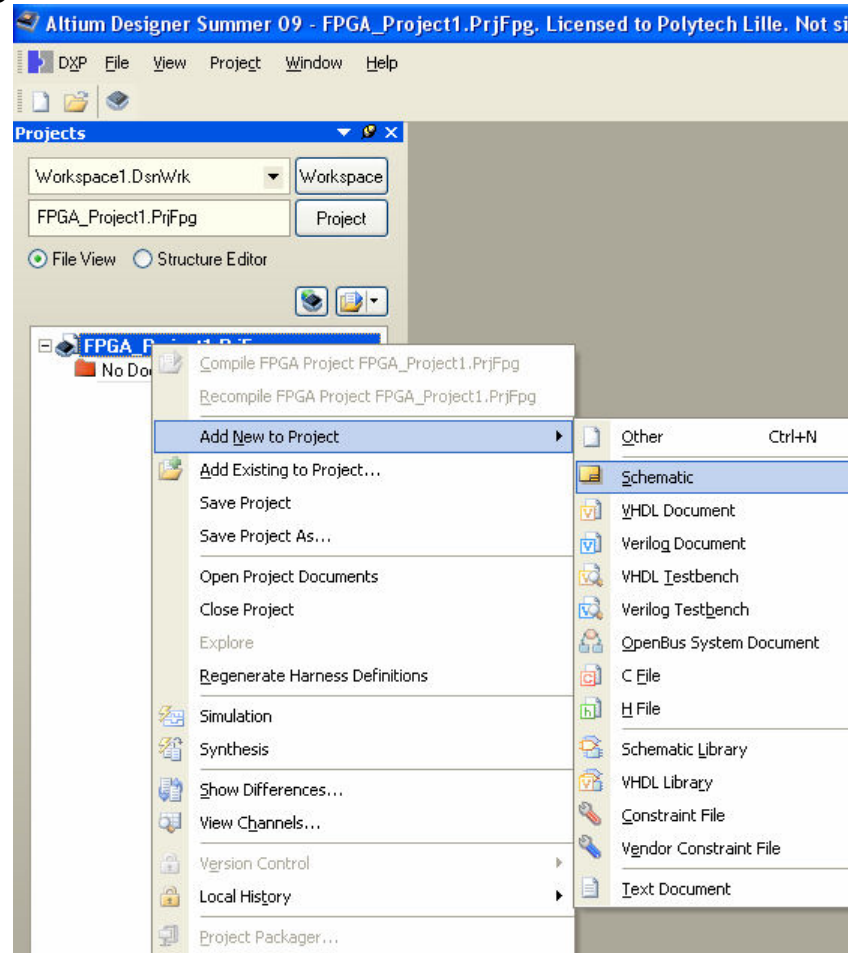
*L'objectif de ce tutoriel est de vous familiariser avec ce logiciel de conception et d'implémenter une fonction logique sur la *nanoboard*.

- Démarrage du logiciel:
Démarrer → Altium Designer Summer 09 → Altium Designer Summer 09
- Création d'un nouveau projet:
file → new → project → FPGA project



- Sauvegarder votre projet dans votre répertoire de travail:
file → save project as

- Ajout d'une source « schematic » au projet pour la saisie du schéma:
sélectionner le projet dans l'arborescence + click bouton-droit:



- Sauvegarde cette source dans le répertoire de travail :
sélectionner la source dans l'arborescence + click bouton-droit: « save as »
nommer ce fichier avec l'extension « .schdoc »

• vous obtenez l'écran suivant:

The screenshot shows the Altium Designer Summer 09 interface. The main workspace is a grid-based schematic editor. On the left, there is a 'Projects' panel showing a tree view of the project files. At the top, there is a menu bar and a toolbar. At the bottom, there is a 'Messages' panel with a table for error messages.

Annotations with arrows pointing to specific parts of the interface:

- Fils d'interconnexion**: Points to the top toolbar.
- Bus d'interconnexion**: Points to a horizontal bus line in the schematic.
- '0' logique**: Points to a logic '0' symbol in the schematic.
- '1' logique**: Points to a logic '1' symbol in the schematic.
- Bus de '1' logique**: Points to a horizontal bus line labeled '1' in the schematic.
- Bus de '0' logique**: Points to a horizontal bus line labeled '0' in the schematic.
- Arborescence de votre projet**: Points to the project tree in the 'Projects' panel.
- Feuille de saisie de votre schéma**: Points to the main schematic grid.
- Librairies des composants logiques à utiliser**: Points to the 'Libraries' panel on the right side.
- Zone de message d'erreur**: Points to the 'Messages' panel at the bottom.

Class	Document	Source	Message	Time	Date	No.

• En cliquant sur l'onglet « bibliothèques », vous accédez aux différentes bibliothèques de composants logiques.

Nom de la bibliothèque

Composants logiques de la bibliothèque « FPGA generic »

Format du composant sélectionné

Component Name	Library	Description
ACC1	FPGA Generic	1-Bit Loadable Cascadable
ACC2B	FPGA Generic	2-Bit Loadable Cascadable
ACC2S	FPGA Generic	2-Bit Loadable Cascadable
ACC4	FPGA Generic	4-Bit Loadable Cascadable
ACC4B	FPGA Generic	4-Bit Loadable Cascadable
ACC4S	FPGA Generic	4-Bit Loadable Cascadable
ACC8	FPGA Generic	8-Bit Loadable Cascadable
ACC8B	FPGA Generic	8-Bit Loadable Cascadable
ACC16	FPGA Generic	16-Bit Loadable Cascadable
ACC16B	FPGA Generic	16-Bit Loadable Cascadable
ACC32B	FPGA Generic	32-Bit Loadable Cascadable
ADD1	FPGA Generic	1-Bit Cascadable Full Adder
ADD2B	FPGA Generic	2-Bit Cascadable Full Adder

1896 components

U?

CI	
B0	Q0
D0	
L	C0
ADD	
CE	

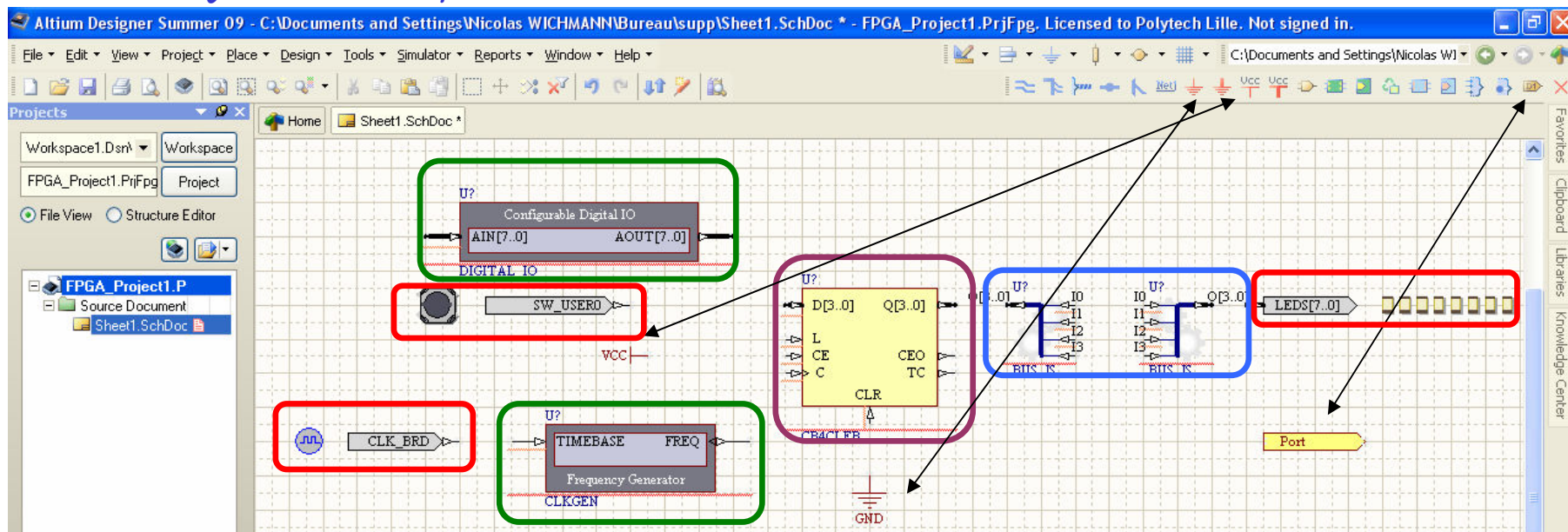
• Les bibliothèques à utiliser pour vos projets sont:

- **FPGA GENERIC**
composants logiques standards (bascules (D,JK,RS,...), comparateur, additionneur, compteur, multiplexeur,...)
- **FPGA CONFIGURABLE GENERIC**
composants logiques standards configurables manuellement
- **FPGA NB2DSK01 port-plugin**
ports d'entrées-sorties situés sur la nanoboard (horloge interne, LED, RS232, VGA, boutons-poussoirs,...)
- **FPGA instruments**
instruments virtuels disponibles (générateur d'horloge, générateur de signaux internes,...)

Nous allons réaliser un compteur 4bits avec chargement initial:

- le chargement sera réalisé grâce à un instrument virtuel.
- la validation du chargement se fera au moyen d'un bouton poussoir de la nanoboard.
- la fréquence d'incrémentation du compteur sera réglée à l'aide d'un générateur d'horloge.
- la valeur du compteur (4bits) sera envoyée sur 4 LEDS de la nanoboard.
- l'horloge du compteur sera envoyée sur une broche externe de la nanoboard pour la visualiser à l'analyseur logique (appeler l'enseignant)

Ajouter les composants suivants sur votre feuille « schematic »:



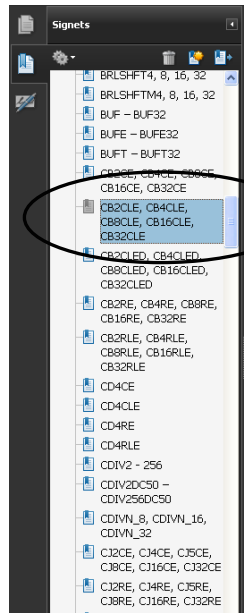
Bibliothèques:

- :FPGA NB2DSK01 port-plug-in
- :FPGA configurable generic
- :FPGA generic
- :FPGA instruments

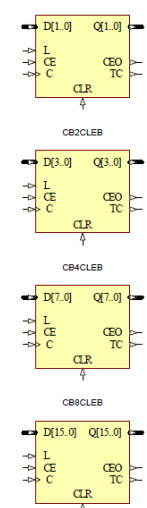
Raccourci clavier:

- Rotation des composants → barre espace
- Zoom → page up/page down
- Miroir horizontal → click gauche sur composant + touche X
- Miroir vertical → click gauche sur composant + touche Y

- **CB4CLEB**: compteur (click bouton droit sur composant → *references* → *help*)



CB2CLE, CB4CLE, CB8CLE, CB16CLE, CB32CLE
Loadable Cascadable Binary Counters with Clock Enable and Asynchronous Clear



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

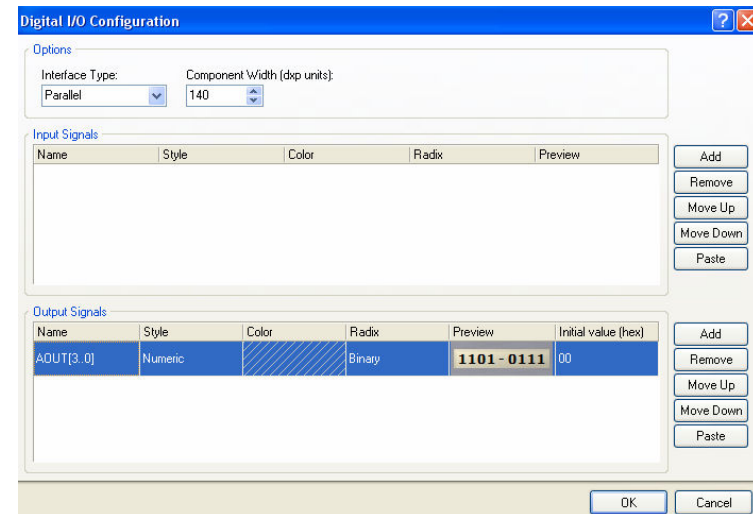
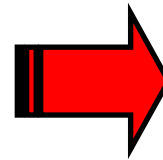
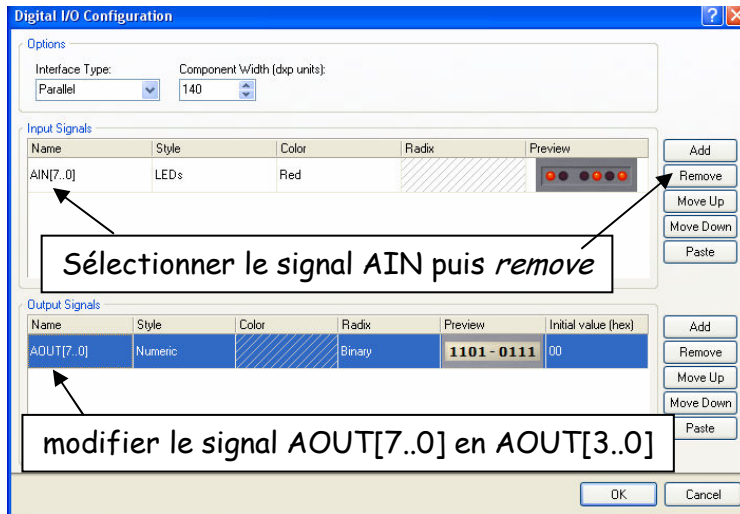
z= 1 for CB2CLE; z = 3 for CB4CLE; z = 7 for CB8CLE; z = 15 for CB16CLE; z = 31 for CB32CLE

- **SW_USER0**: bouton poussoir de la nanoboard → *état '1' au repos*
- **CLK_BRD**: horloge interne de la nanoboard fixée à 50MHz
- **LED[7..0]**: bus de LED de la nanoboard (8bits) → *LED(7): LED de poids fort; LED(0): LED de poids faible*
- **PORT**: servira à envoyer l'horloge du compteur vers une broche externe de la nanoboard
- **BUS_JS**: élément permettant de dégroupier les bits d'un bus (*splitter*) ou de regrouper les bits en bus (*joiner*)
- **CLKGEN**: générateur d'horloge réglable (*sortie FREQ*) à partir d'un horloge de référence (*entrée TIMEBASE*)
- **DIGITAL_IO**: générateur de signaux

1) Configuration du composant DIGITAL_IO:

ce composant doit être configuré afin de pouvoir générer un bus de 4 bits

Double-click sur le composant puis *configure*



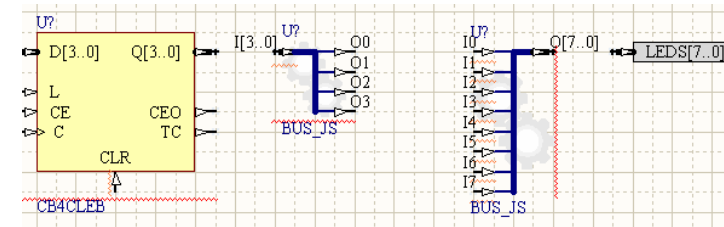
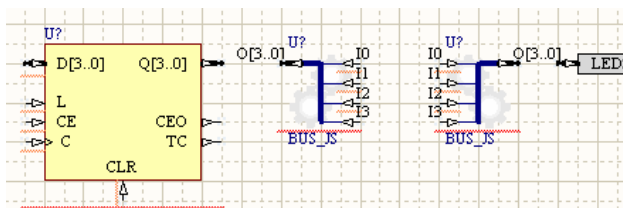
2) Configuration des 2 composants BUS_JS:

le premier doit être configuré en tant que splitter 4bits (extraction des 4 bits du compteur):

Double-click sur le composant puis *configure* → type: *splitter*

Le second doit être configuré en tant que joiner 8bits:

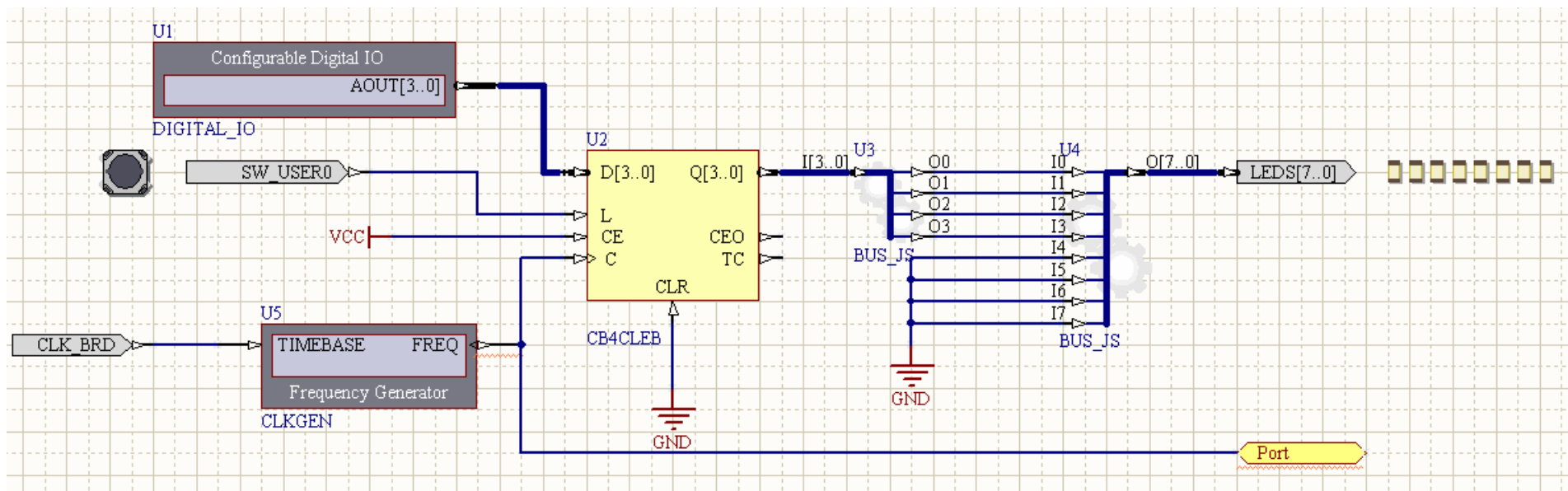
Double-click sur le composant puis *configure* → type: *joiner* + ajouter les bits nécessaires afin d'avoir 8bits



1) Relier les composants entre eux avec:



2) Effectuer l'annotation automatique des composants
menu *tools* → *annotate schematics quietly*




2) Affecter le port « PORT » à une broche externe de la nanoboard:

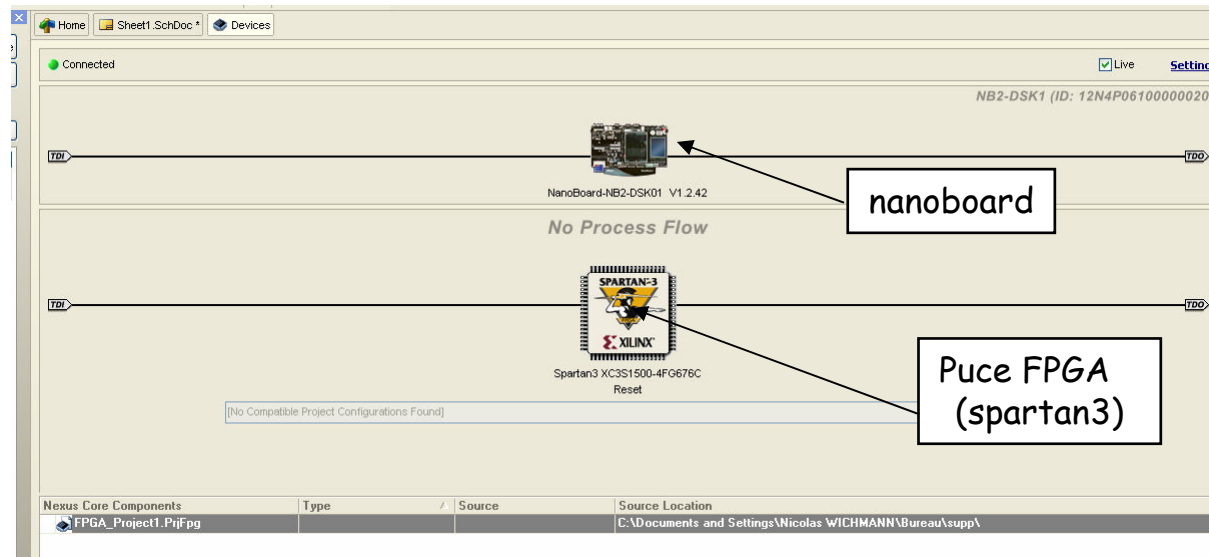
double click sur le composant « port »

modifier le nom du port → *name: HA2* (HA2 correspond à la broche 2 du USER HEADER A)

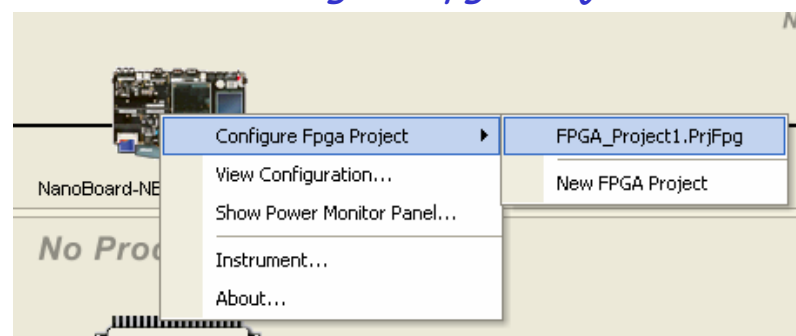


modifier son type → *IO type: output*

- Nous allons maintenant implémenter ce système numérique sur la carte FPGA de la nanoboard
Cliquez sur l'icône  de votre barre de menu pour accéder à la nanoboard:



Cliquez bouton-droit sur la nanoboard → « configure fpga Project » → cliquez sur le nom de votre projet:



Cliquez sur OK pour valider la configuration qui a été créée

Remarque: ces étapes sont à faire qu'une seule fois par projet

*Lancer, l'une après l'autre, les étapes: « compile », « synthesize », « build » et « program FPGA »
(un point vert valide chaque étape)*

Nexus Core Components	Type	Source	Source Location
FPGA_Project1.PriFpg			C:\Documents and Settings\Nicolas WICHMANN\Bureau\supp\
U1	DIGITAL_IO	Sheet1.SchDoc	C:\Documents and Settings\Nicolas WICHMANN\Bureau\supp\
U5	CLKGEN	Sheet1.SchDoc	C:\Documents and Settings\Nicolas WICHMANN\Bureau\supp\

Vous pouvez voir apparaître votre chaîne d'instrument virtuel (générateur d'horloge, générateur de signaux)

Votre FPGA est programmé

*Vous pouvez maintenant ouvrir les 2 instruments virtuels afin de les régler.
double-click sur les 2 instruments virtuels*



Régler la fréquence d'horloge du compteur à 2Hz à l'aide du générateur de fréquence

*Imposer une valeur initial du compteur à l'aide du générateur de signaux AOUT[3..0]
(click sur les bits de AOUT pour modifier leur valeur)*

Vous pouvez constater qu'en laissant votre doigt appuyé sur le bouton poussoir SW0, votre compteur compte à partir de la valeur fixée par le bus AOUT

En relâchant votre doigt, le compteur est en mode chargement.