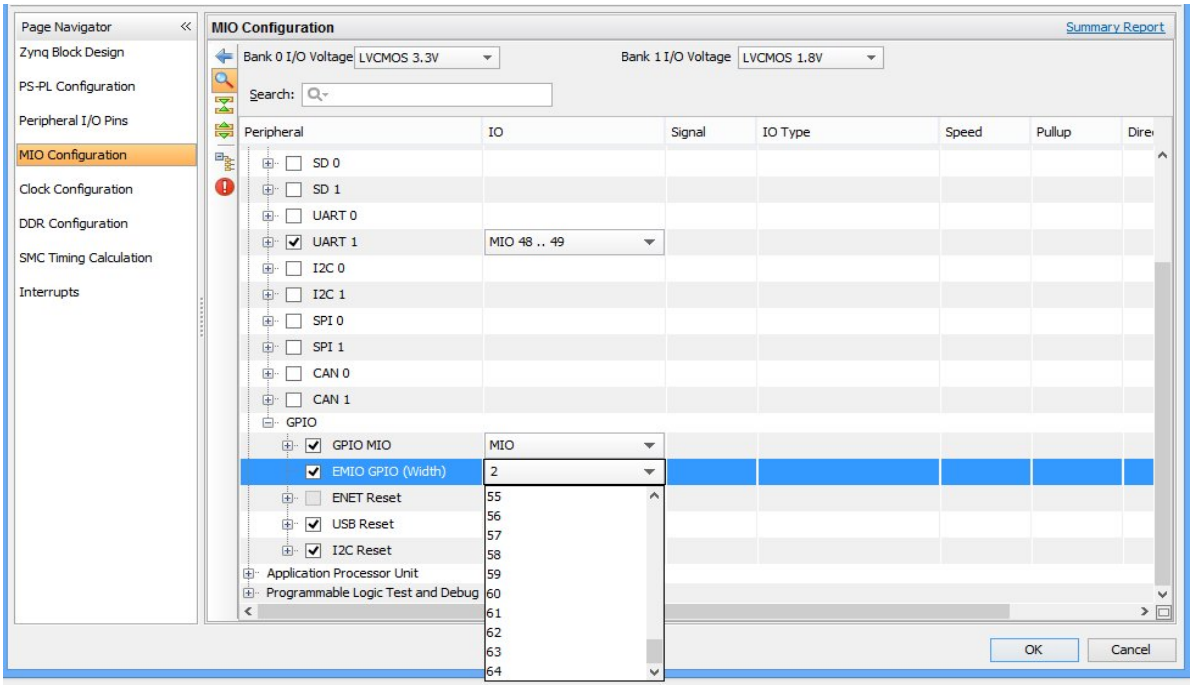
# Exploitation du BUS EMIO
# Sur la plateforme ZYNQ

## 1) Problématique

Le port EMIO permet de connecter la partie traitement PS du ZYNQ vers la partie programmable PL du ZYNQ. Le port EMIO évite l'utilisation du BUS AXI4 beaucoup plus compliqué à mettre en œuvre. Le port EMIO sera composé de deux lignes d'entrées sorties, une ligne de consigne vers l'interrupteur SW0 et une ligne de pilotage vers la LED LD0. Une des broches du MIO sera connectée vers la LED LD4.

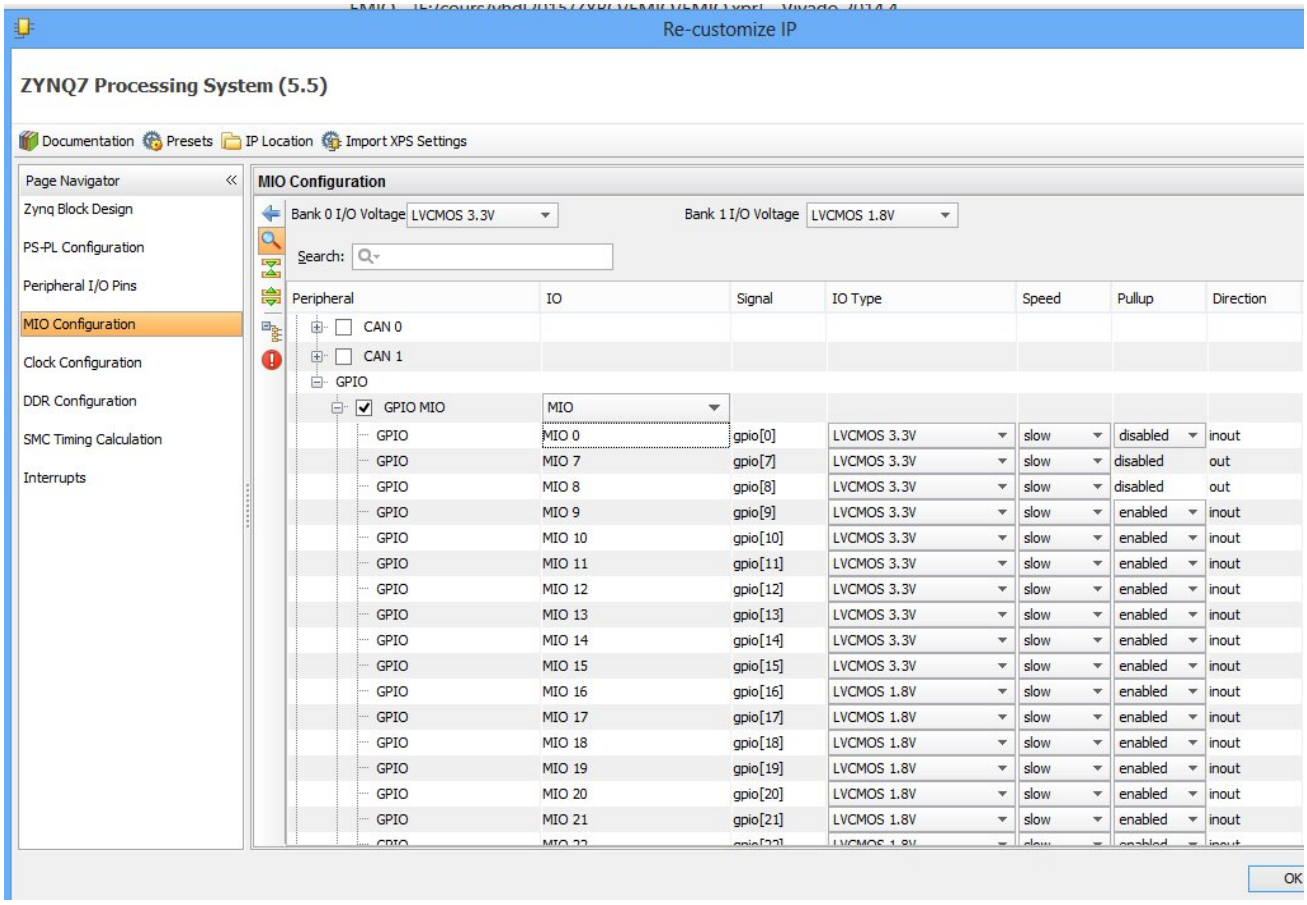
## 2) Mise en œuvre matériel du SOC ZYNQ

Comme précédemment, lancez VIVADO et créez un projet incluant le fichier de contrainte de la carte ZYBO.
Lancez la création d'un bloc design et intégrez le PS ZYNQ. Automatisez la connexion du ZYNQ vers la DDR.
On doit configurer le PS afin de mettre en place le port MIO et EMIO.
N'oubliez pas d'inclure le fichier de définition du PS
Sélectionnez l'UART1, le port MOI et EMIO




Le port EMIO est composé de 64 lignes ce que l'on peut remarquer sur la figure précédente. On choisira une dimension de 2 bits

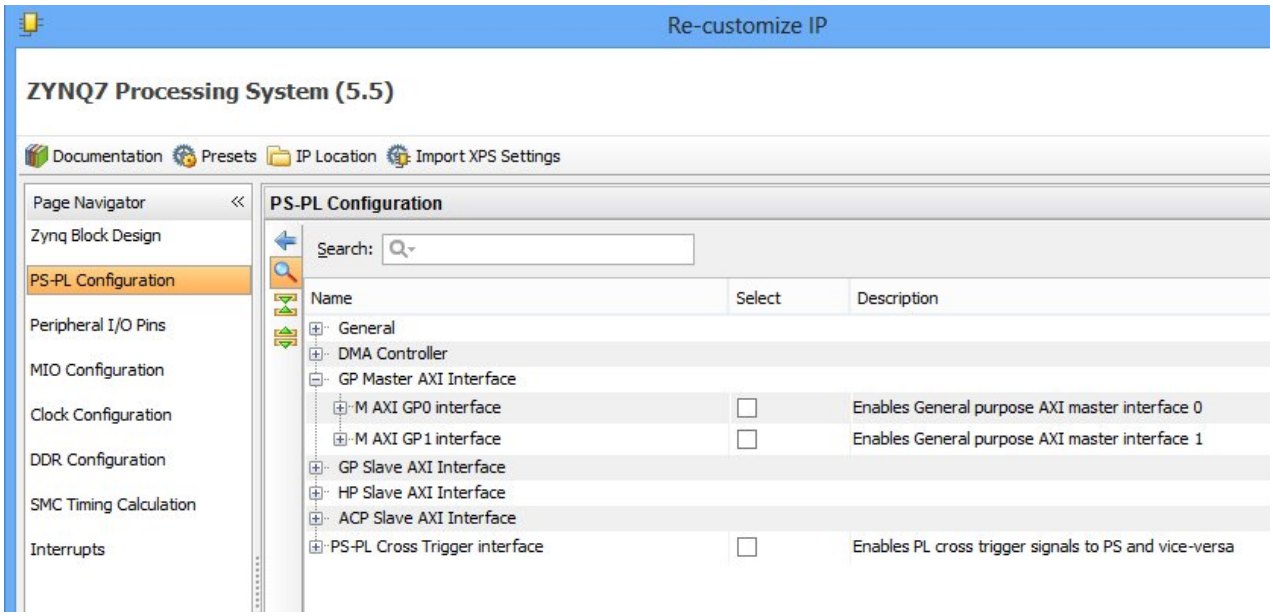
Le port MIO est composé de 54 bits et permet de connecter les périphériques du PS vers les broches externes du ZYNQ.



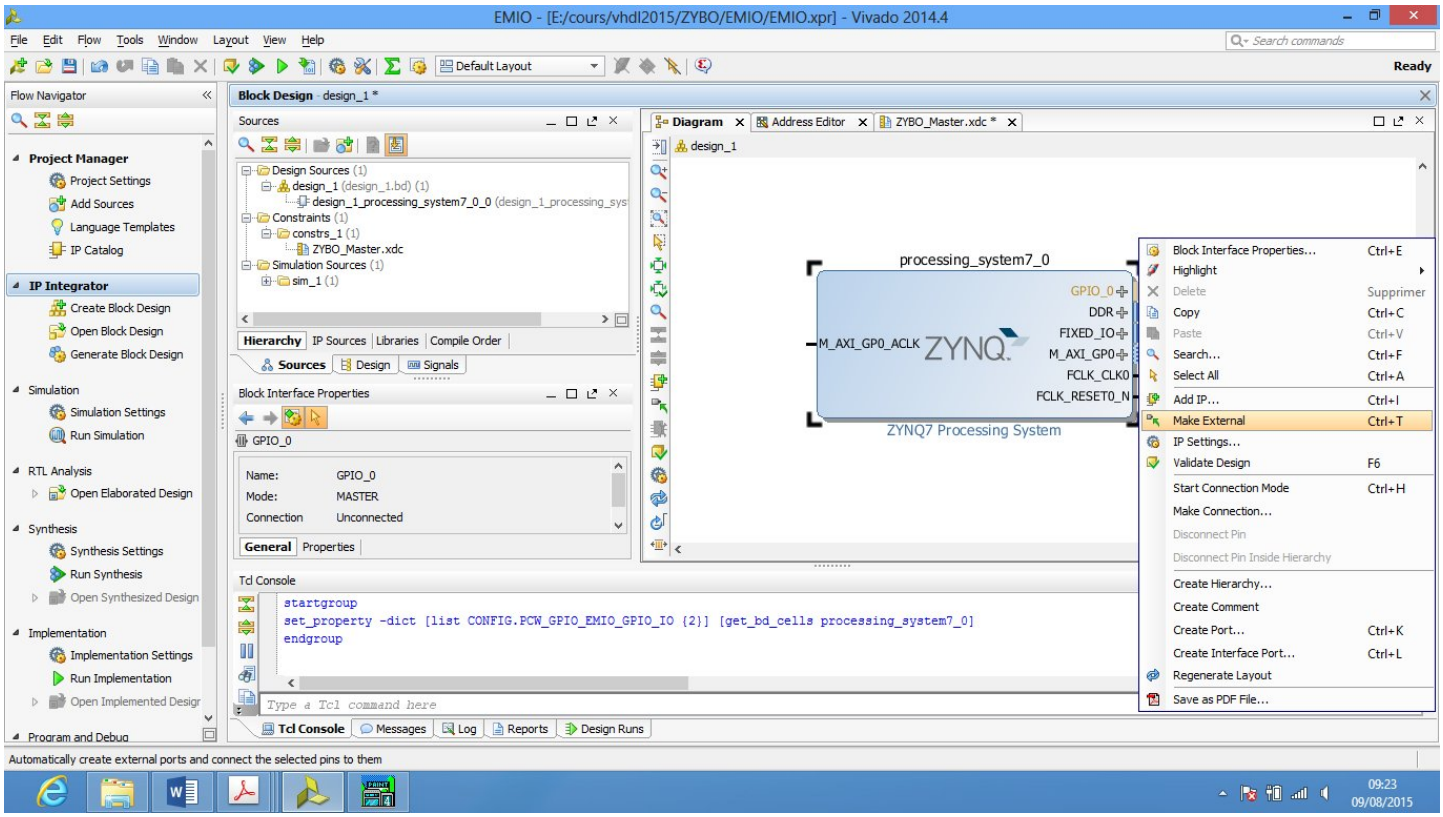

Le fichier de configuration ZYBO_zynq_def.xml permet de figer la connexion des périphériques vers le MOI. La documentation de la carte ZYBO nous donne ce câblage par défaut. On peut constater que la broche MIO 7 est connectée sur la LED N°4, ce qui nous sera utile par la suite

| MIO 500 3.3 V | Peripherals | Peripherals | Peripherals |
|---|---|---|---|
| Pin | Pmod | SPI Flash | GPIO |
| 0 | JF7 | | |
| 1 | | CS | |
| 2 | | DQ0 | |
| 3 | | DQ1 | |
| 4 | | DQ2 | |
| 5 | | DQ3 | |
| 6 | | SCLK | |
| 7 | | | LED4 |
| 8 | | SLCK FB | |
| 9 | JF8 | | |
| 10 | JF2 | | |
| 11 | JF3 | | |
| 12 | JF4 | | |
| 13 | JF1 | | |
| 14 | JF9 | | |
| 15 | JF10 | | |

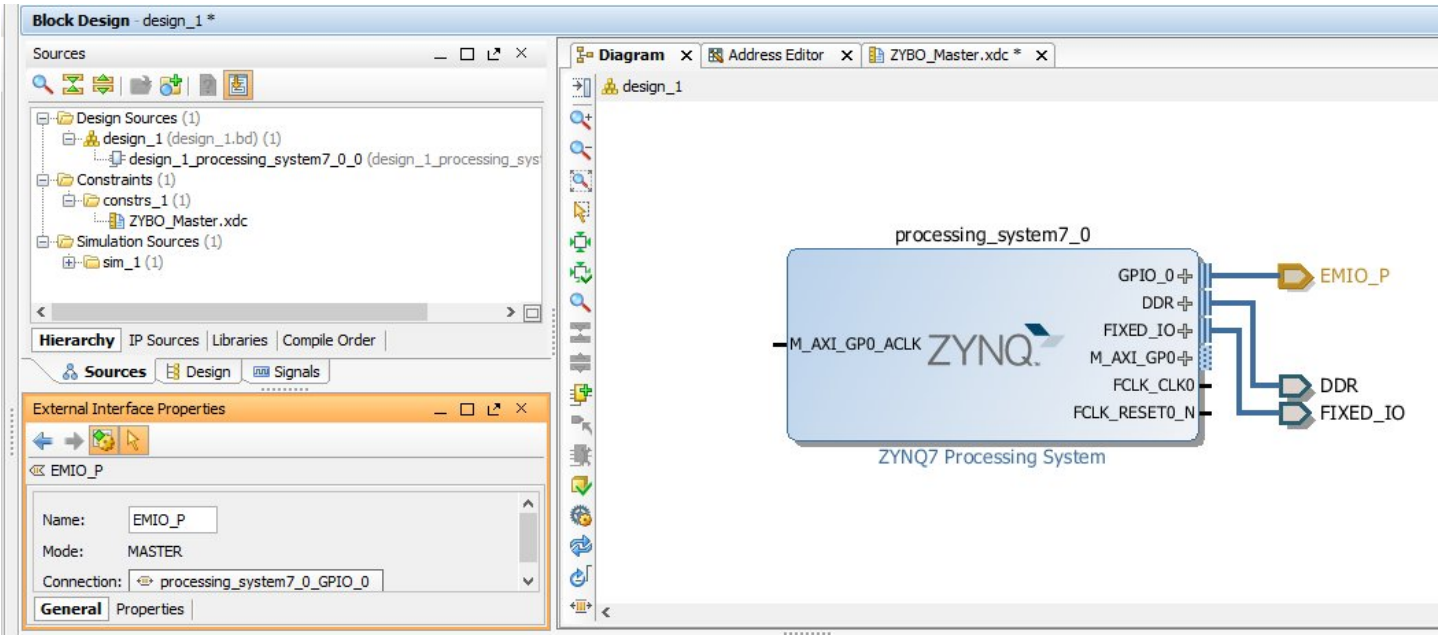
Noubliez pas de déconnecter l'interface M AXI GP0 interface

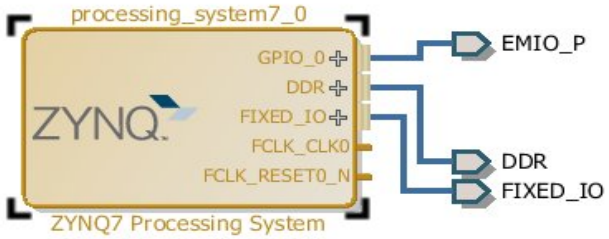



Créez une connexion externe du GPIO_0, cliquez droit GPIO_0 et make external

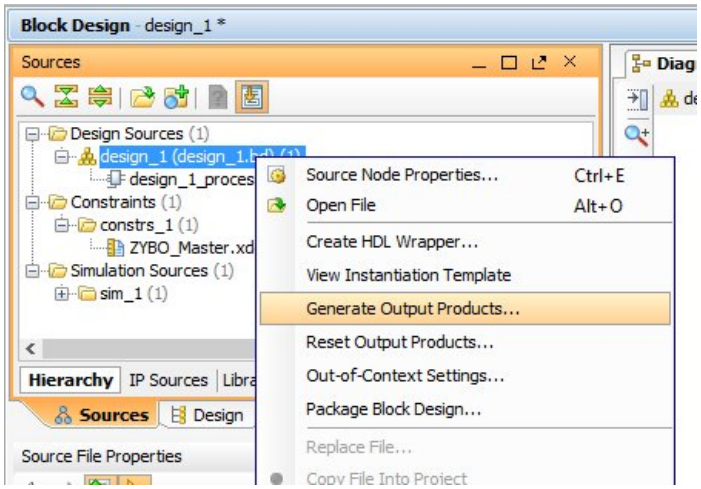


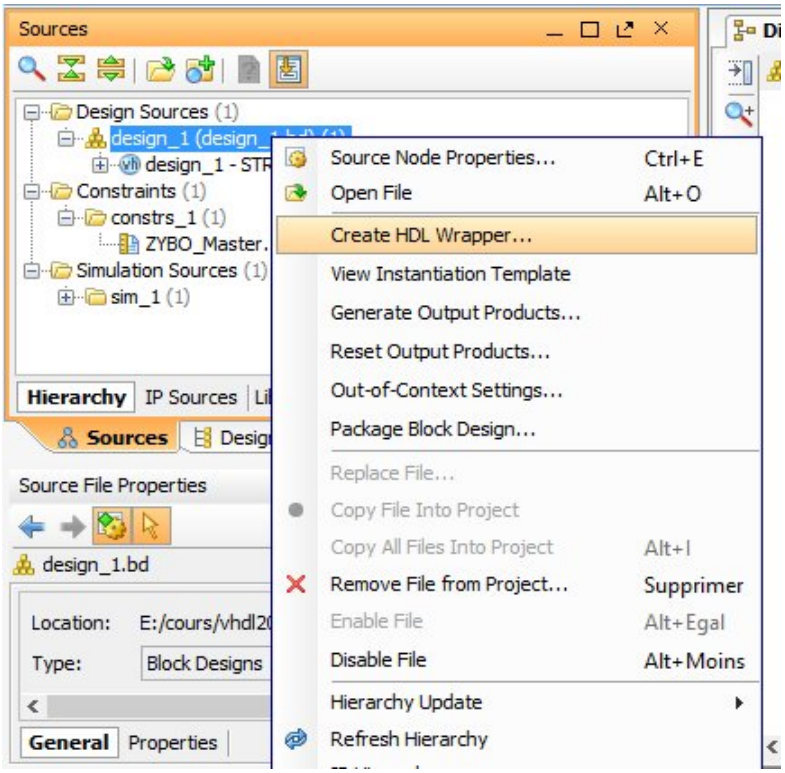
Dans la fenêtre external interface properties changez le nom en EMIO_P



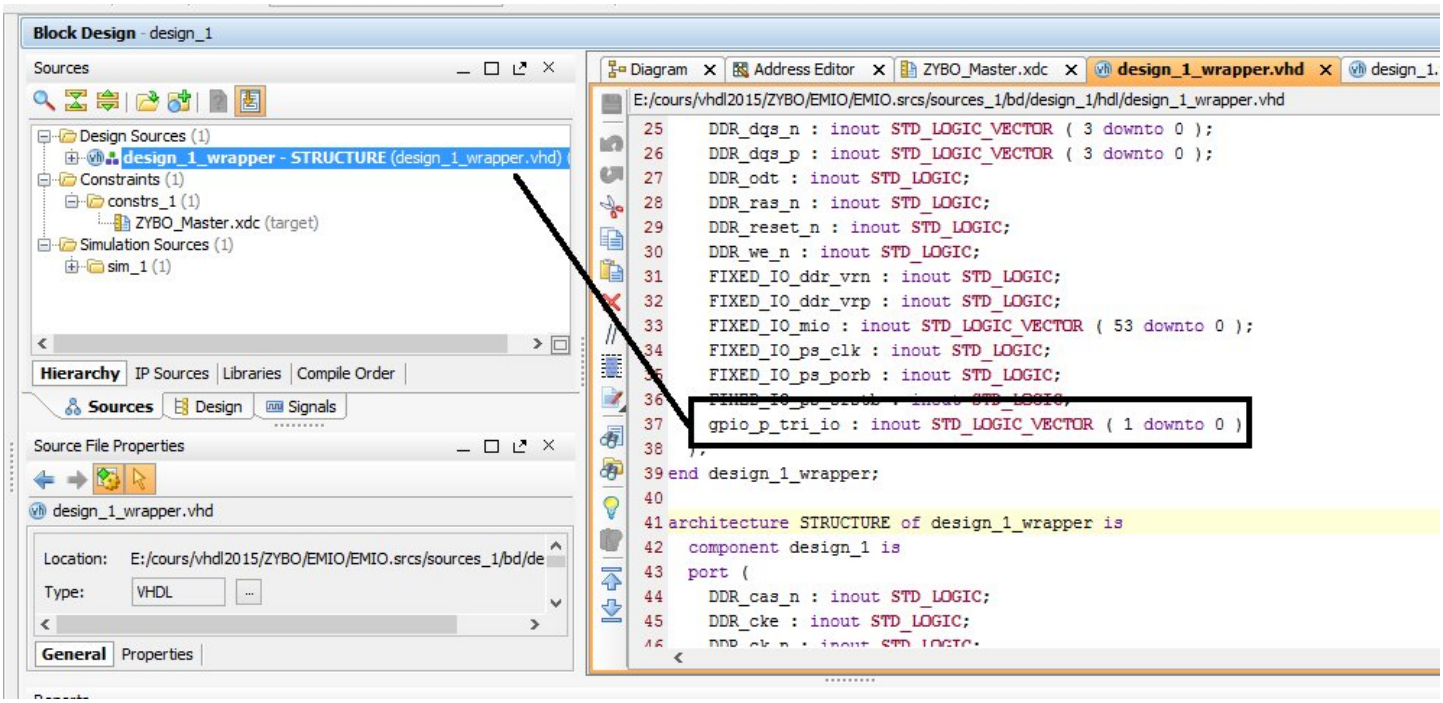




Générez le fichier de configuration du PS pour l'exportation vers la suite SDK

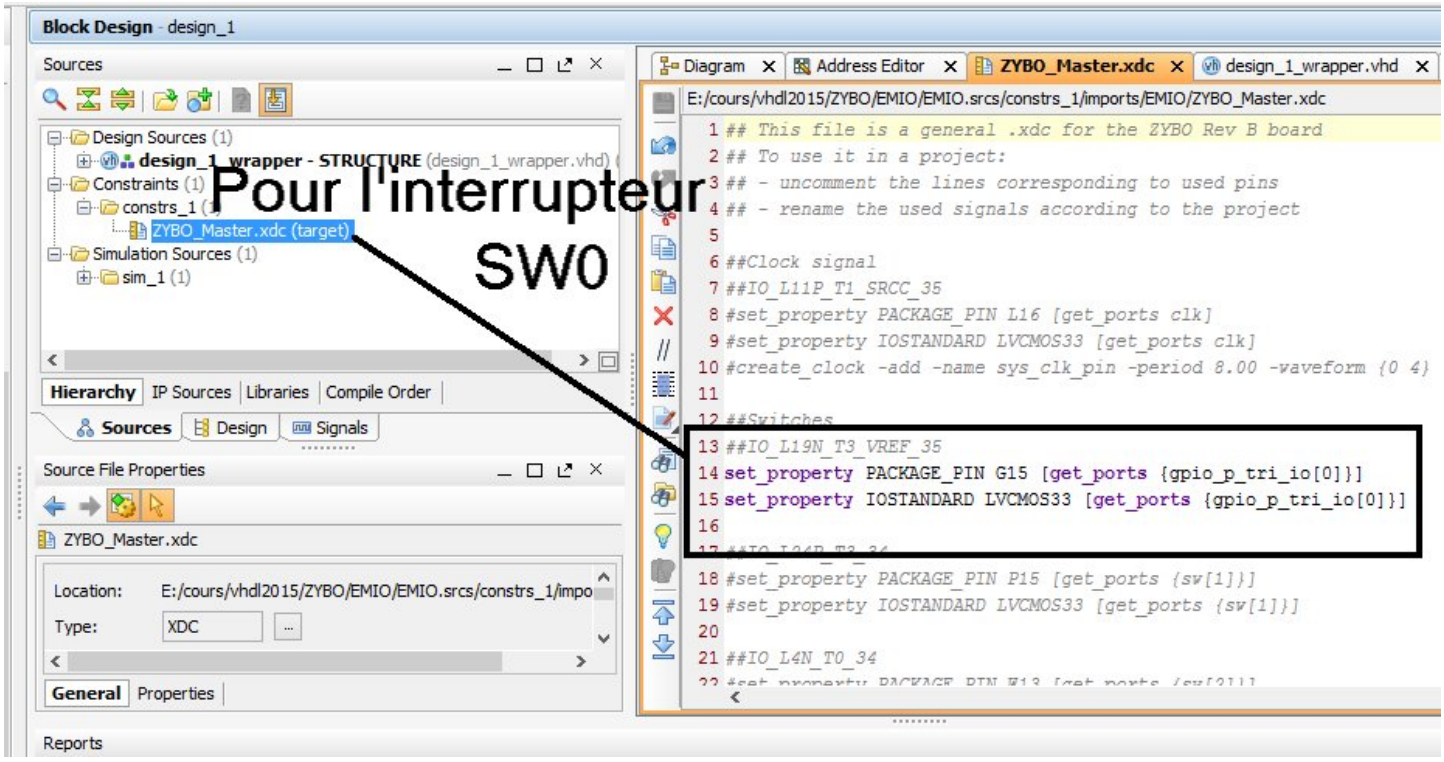




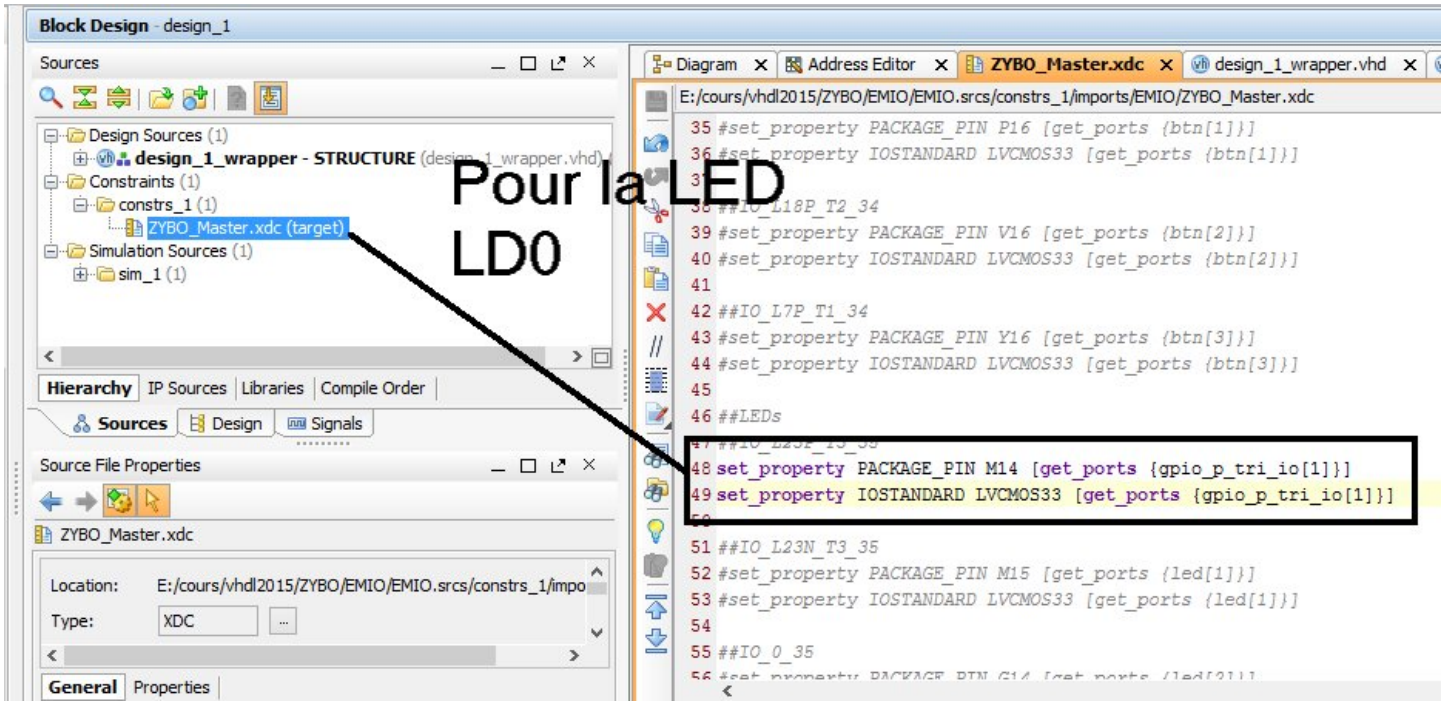

Dans le fichier VHDL de TOP level



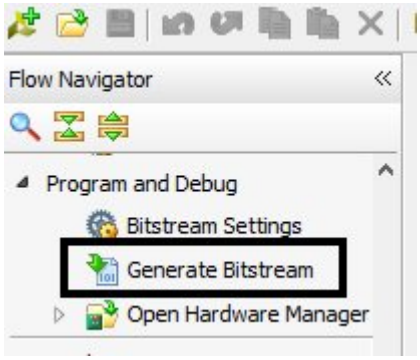
Dans le fichier de contrainte déclarez ces deux lignes pour connecter la ligne 0 du port EMIO vers l'interrupteur SW0



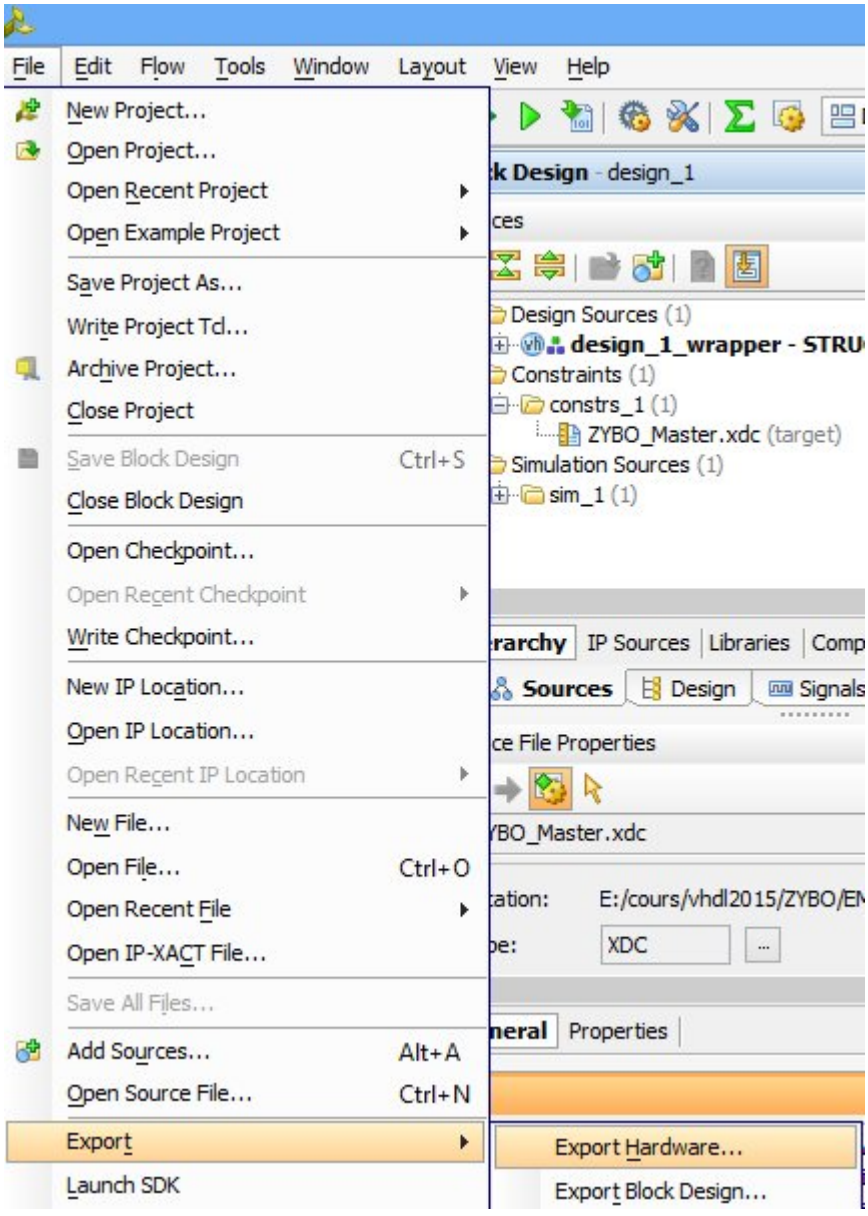

Dans le fichier de contrainte déclarez ces deux lignes pour connecter la ligne 1 du port EMIO vers la LED4



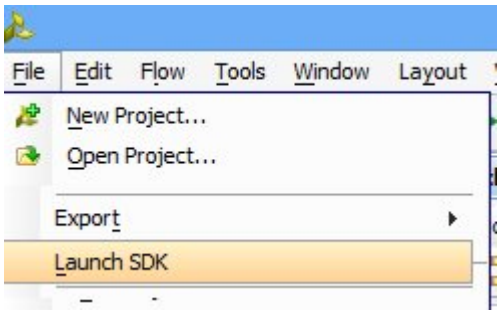
Générez le bitstream




Exportez le matériel vers SDK



Lancez SDK




Créez une nouvelle application, complétez votre programme et testez