

TRAVAUX PRATIQUES

SoPC

TP2 : INITIATION À LA CONCEPTION POUR ZYNQ AVEC LA CARTE PYNQ

Les instructions de ce TP sont sur la façon d'utiliser la carte Pynq-Z2.

Ce TP nous montrera comment créer une nouvelle conception matérielle Vivado pour PYNQ. Ce tutoriel est basé sur l'image PYNQ v2.4 et utilisera Vivado 2018.2.

La carte PYNQ-Z2 a été utilisée pour tester cette conception.

Si vous utilisez le PYNQ-Z2, assurez-vous d'abord que les fichiers de la carte ont été installés. Téléchargez les fichiers de la carte PYNQ-Z1 365 ou les fichiers de la carte PYNQ-Z2 432, extrayez et copiez sur votre installation Vivado, dans le répertoire ./data/boards/board_files.
« Voir Annexe TP »

1. Ajouter la carte Pynq-Z2 à Vivado

Les instructions sur la façon d'ajouter la carte Pynq-Z2 à Vivado.

Dans l'assistant de création de projet Vivado, il est possible de lancer votre conception à partir d'une définition de carte.

Vous n'avez pas besoin de savoir quel est le FPGA exact ni quel matériel est disponible.

2. Obtenir les fichiers « board file »

Les fichiers du board sur pync.io ne sont pas disponibles. Vous les trouverez ici <https://www.tulembedded.com/FPGA/ProductsPYNQ-Z2.html>

Downloads

- PYNQ-Z2 User Manual (PDF)
- PYNQ-Z2 Boot Image
 - 1. V2.4
 - 2. V2.5
 - 3. V2.6
 - 4. V3.0.1
- PYNQ-Z2 Board File (for Pmod IP support please refer here)
- Master XDC
- Protective Acrylic Case (PDF)
- Zynq Datasheet (PDF)
- Zynq Manual (PDF)
- Schematics (PDF)

3. Installer et enregistrer la carte dans Vivado

Pour se faire, il y a deux options :

1. (Obsolète) Placez directement les fichiers dans le dossier des cartes Vivado :

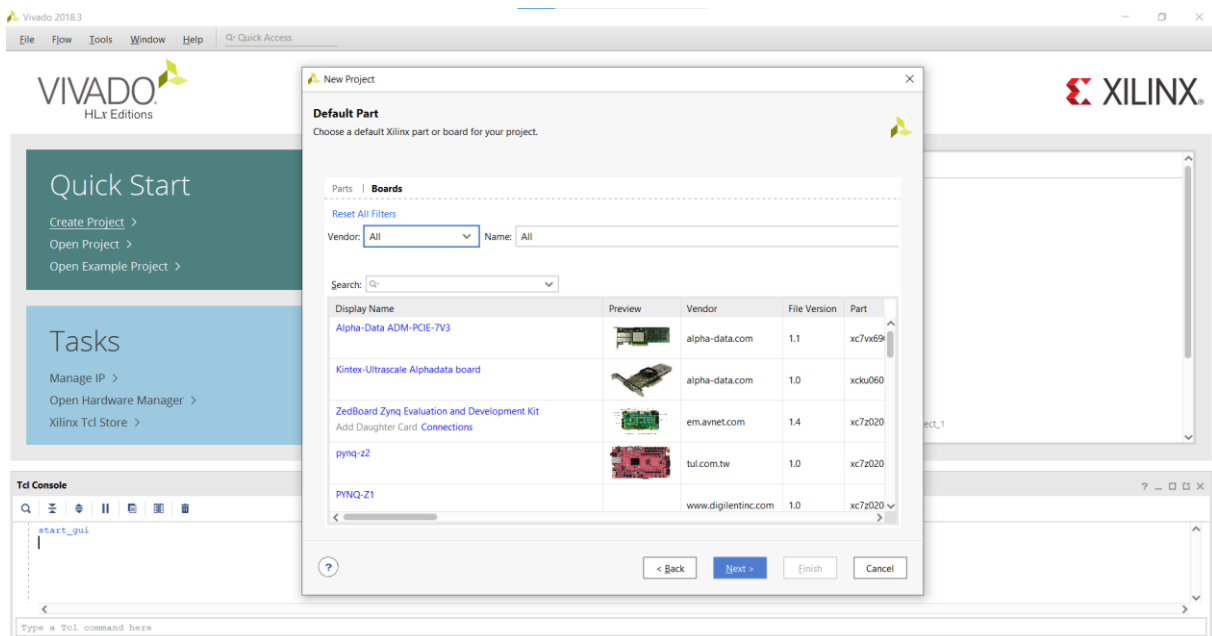
<Xilinx installation directory>\Vivado\<version>\data\boards\board_files.

2. Ou placez-les à l'emplacement de votre choix et ajoutez une ligne au fichier d'initialisation de Vivado, %APPDATA%\Xilinx\Vivado\init.tcl (create it if it does not exist)

set_param board.repoPaths [list "D:/Xilinx/pynq-z2/A.0"]

J'ai utilisé la 1ère option. Vous devez redémarrer l'éditeur.

Le résultat est que vous pouvez désormais créer des projets dotés des bonnes informations FPGA pour le Pynq-Z2 : Vous n'avez pas besoin de savoir quel est le FPGA exact ni quel matériel est disponible.

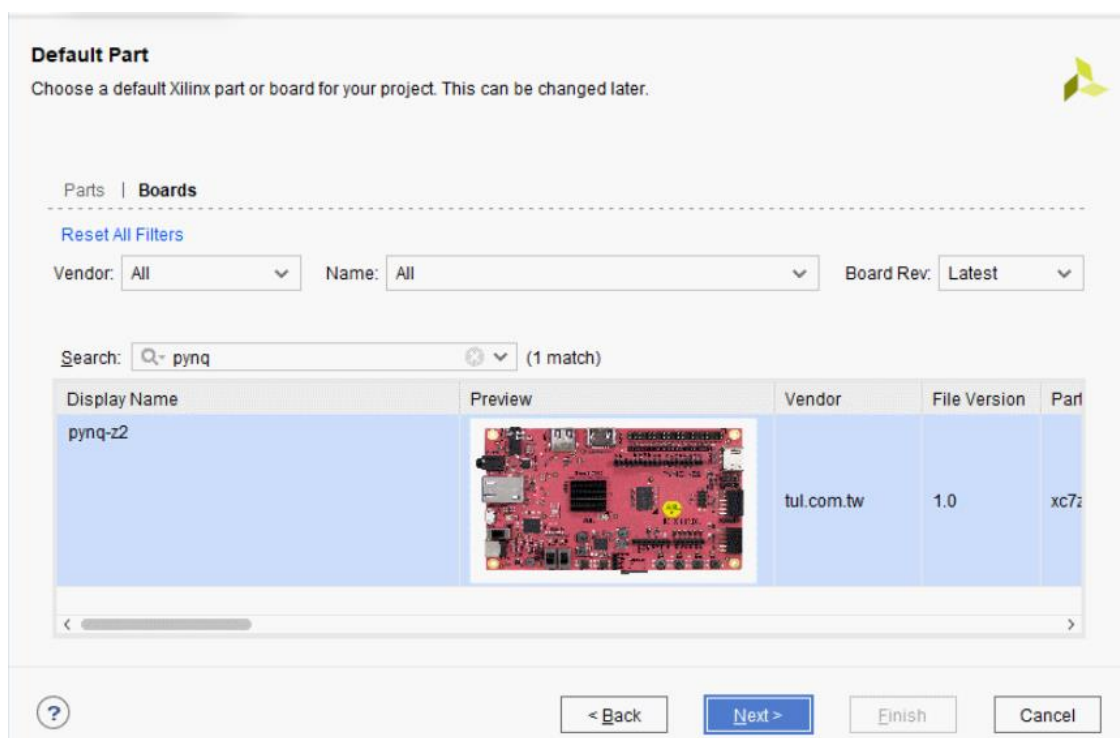
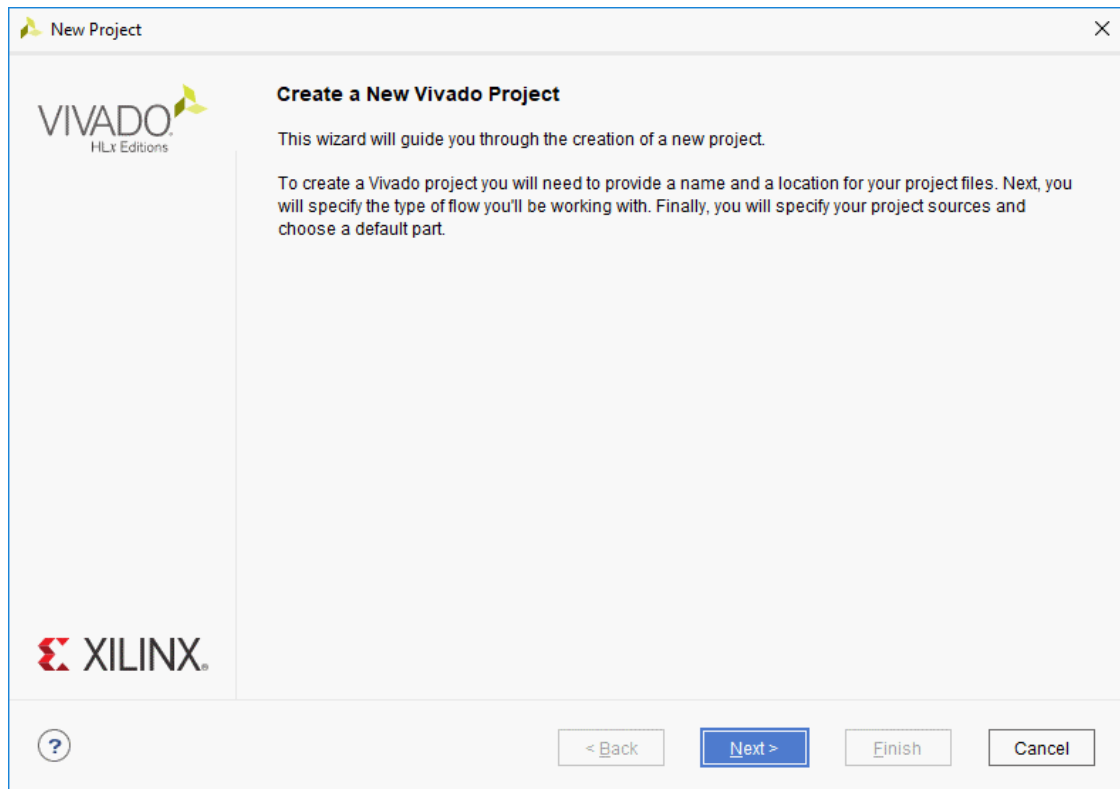


4. CRÉER UNE CONCEPTION MATÉRIELLE POUR PYNQ


4.1 Créer un nouveau projet Vivado

Ce didacticiel créera une conception pour la carte PYNQ-Z2 (Zynq). Vous devriez pouvoir suivre les instructions pour créer une conception similaire pour d'autres cartes Zynq ou Zynq Ultrascale+.

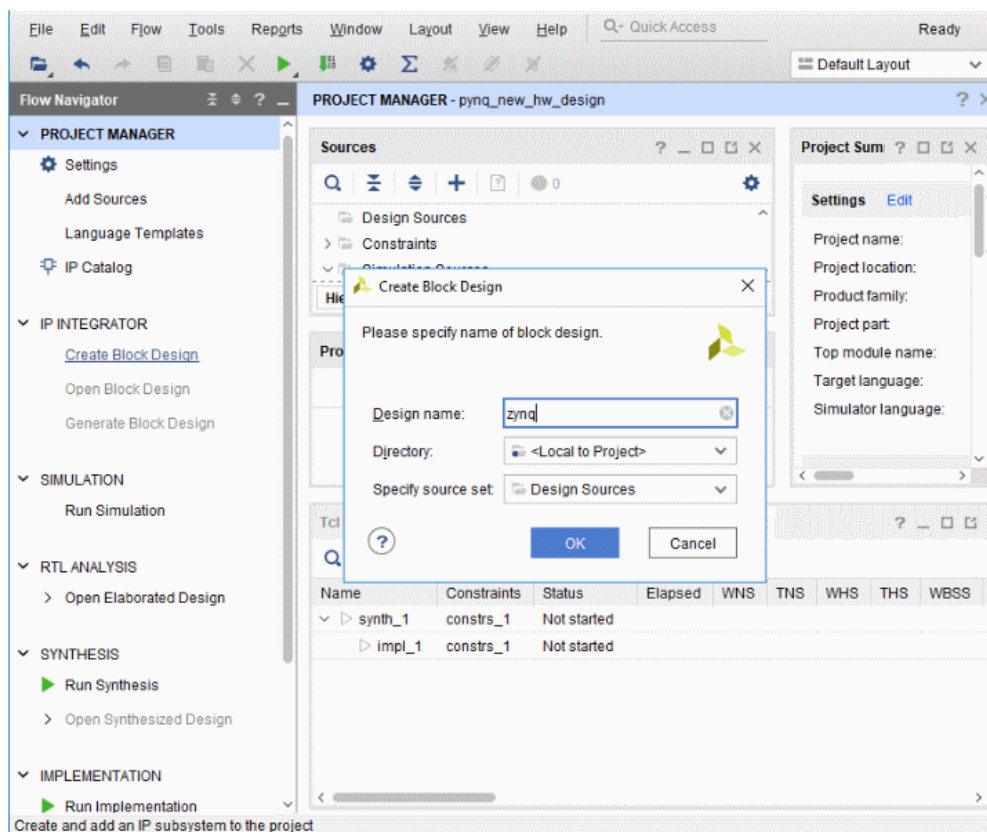
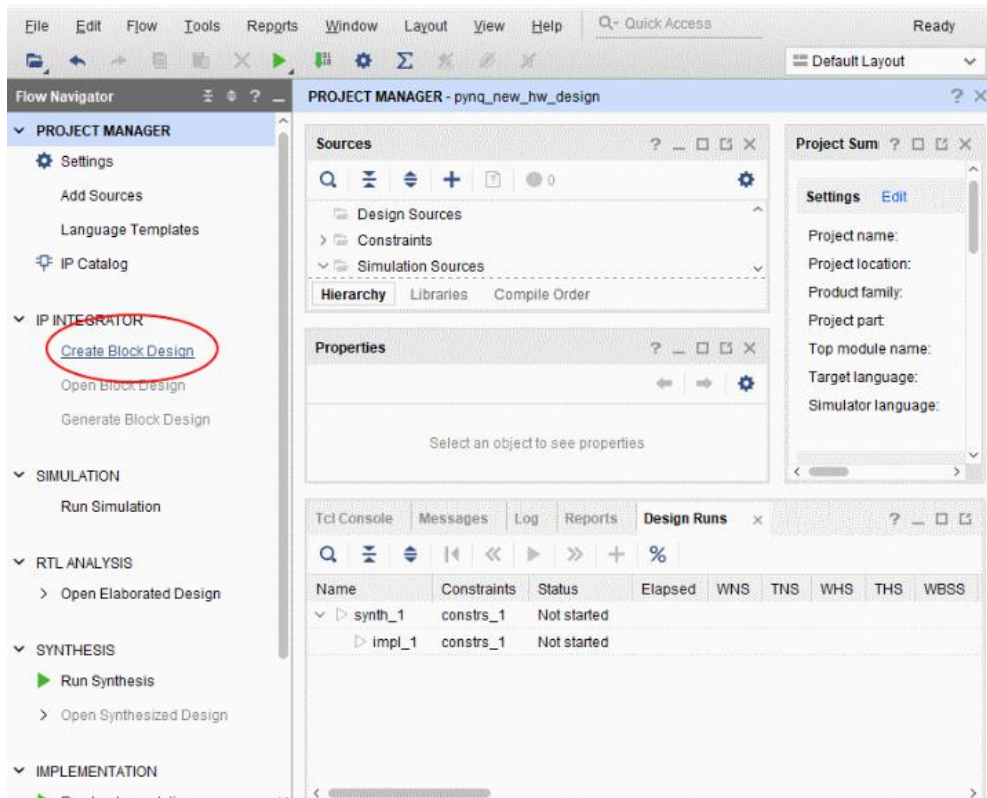
- Ouvrez Vivado et créez un nouveau projet.
- Choisissez un nom de projet et sélectionnez votre tableau Zynq comme cible
Dans cet exemple, le PYNQ-Z2 est sélectionné.
- Sélectionnez RTL comme type de projet et cochez la case Ne pas spécifier de sources pour le moment

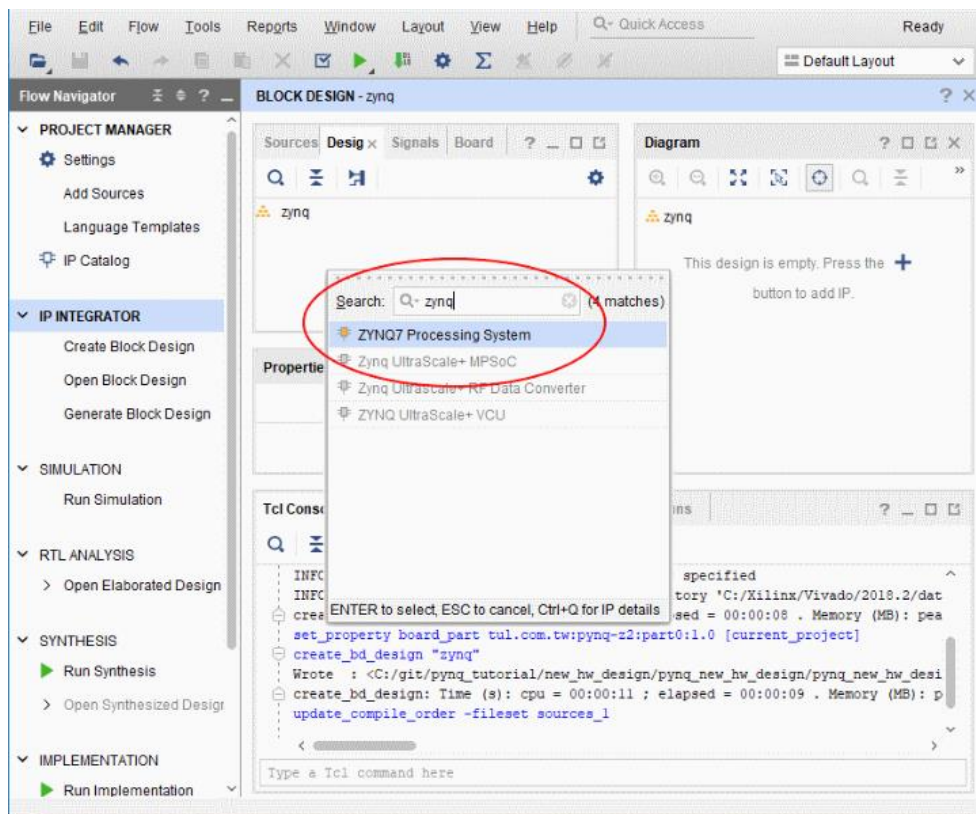
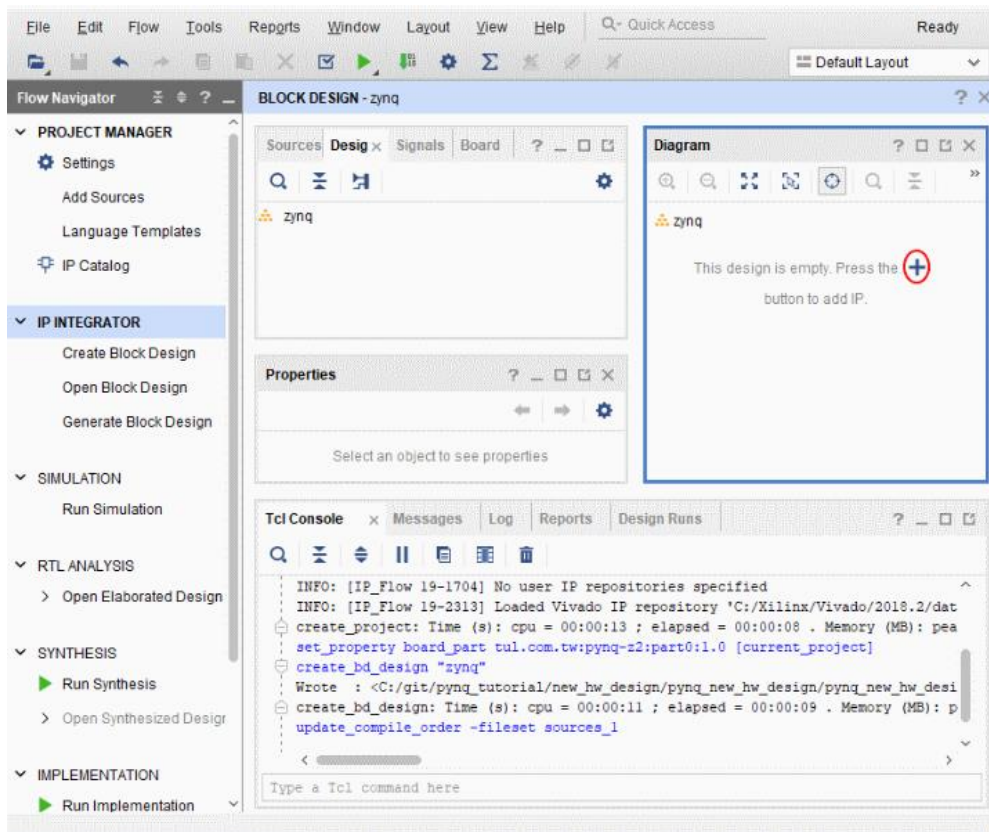


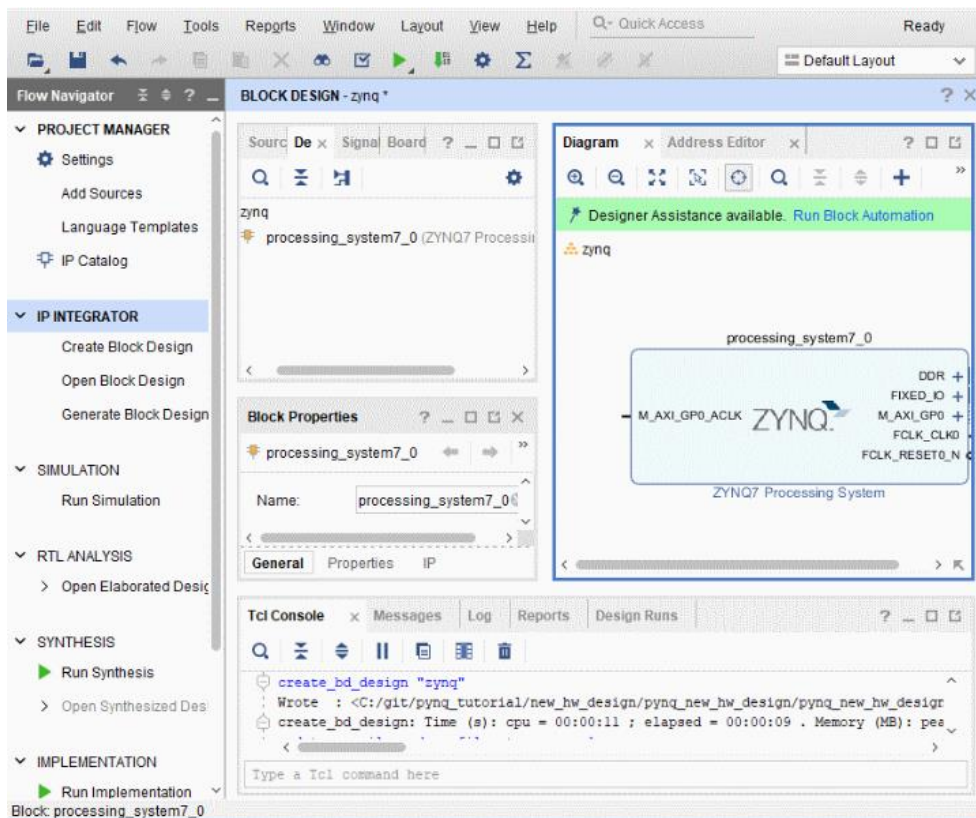
Le projet Vivado va ouvrir.

- Sous IP INTEGRATOR, cliquez sur Créer une conception de bloc
- Dans la barre de menu du diagramme, cliquez sur , recherchez ZYNQ7 Processing System (PS) et double-cliquez pour l'ajouter à la conception.

Notez que vous ne verrez que les IP pouvant être utilisées avec le circuit que vous avez sélectionné.







Zynq est un système sur puce. Il comprend un processeur ARM, une logique FPGA, ainsi que des contrôleurs de mémoire et des périphériques, notamment USB, Ethernet et carte SD.

Le Zynq PS est configuré au moment du démarrage.

Dans une conception Vivado, les paramètres du Zynq PS peuvent être configurés. Vivado est utilisé pour générer la conception logique programmable (bitstream). Il peut également être utilisé pour générer une image de démarrage comprenant tout le code permettant de démarrer le processeur ainsi que le bitstream.

Cependant, l'image de la carte SD PYNQ est utilisée pour démarrer une carte. Les OVERLAYS PYNQ peuvent être chargées après le démarrage du système. Cela signifie que lorsque nous créons une conception Zynq dans Vivado, nous devons uniquement créer la conception logique programmable et pouvons ignorer les paramètres de configuration Zynq PS. Cependant, il est recommandé d'appliquer les paramètres de carte appropriés à votre carte.

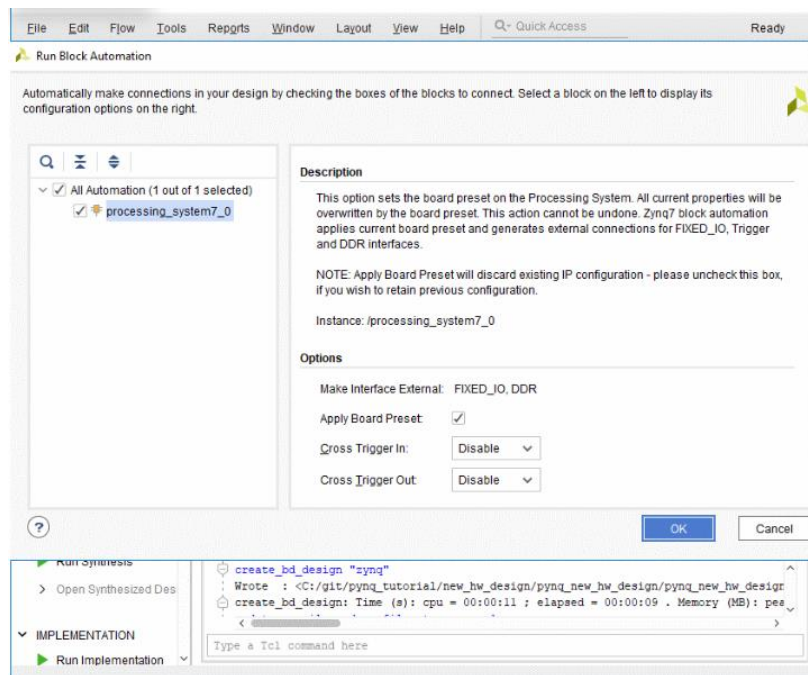
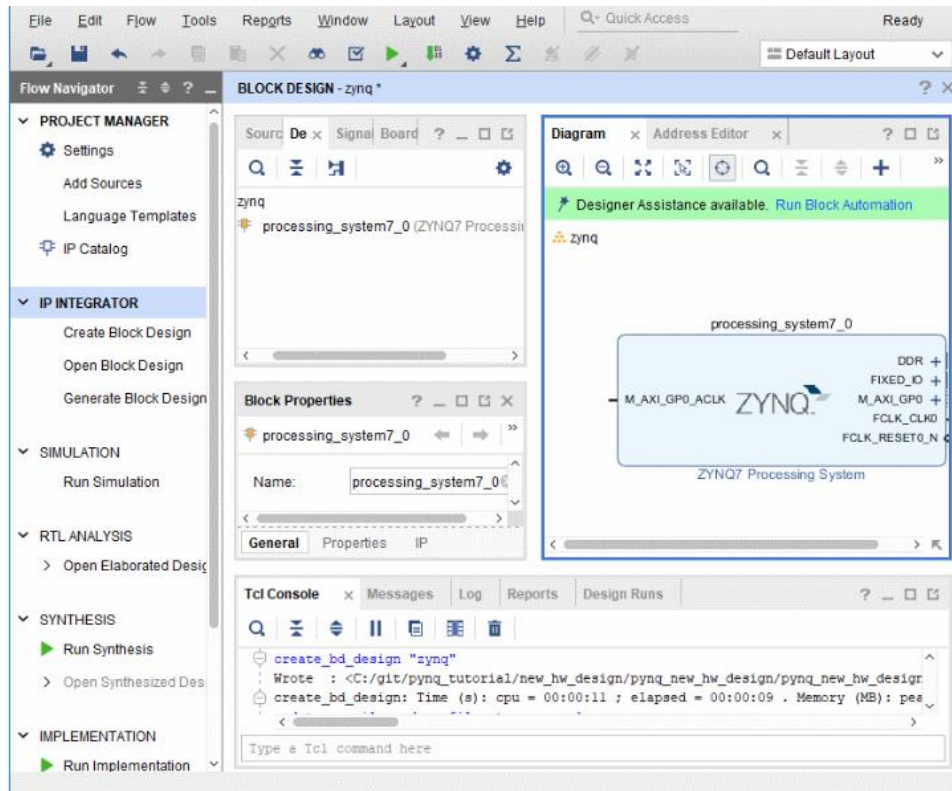
Vous devrez activer les ports et les interfaces sur le bloc PS afin de pouvoir les connecter à votre conception personnalisée.

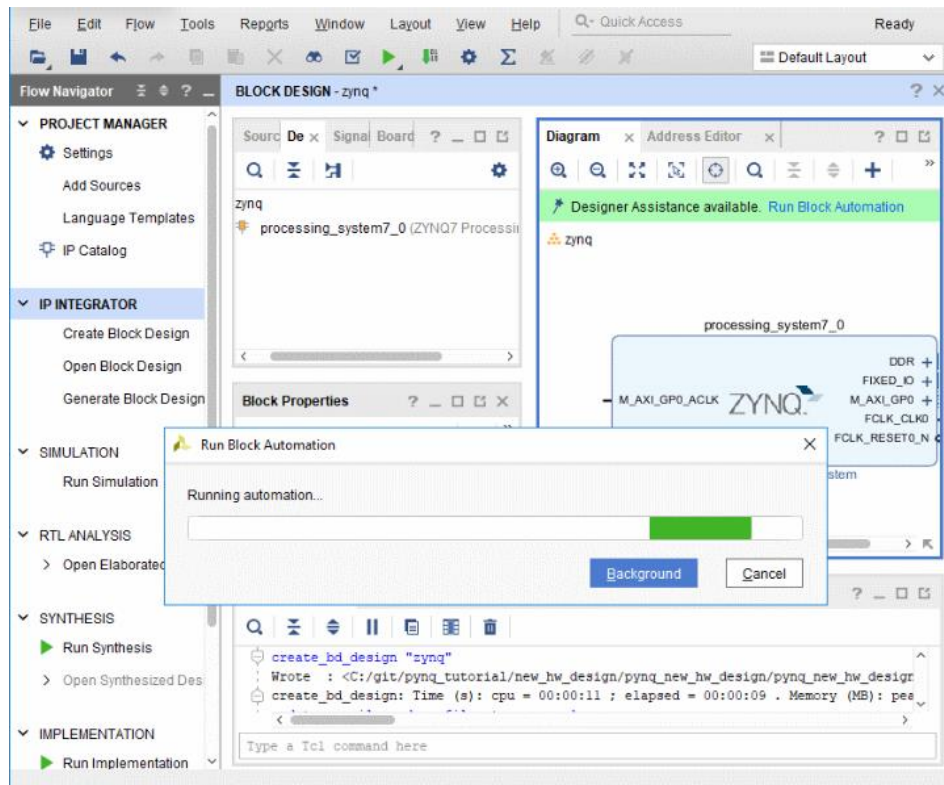
4.2 Block Automation

Lors de l'ajout du bloc Zynq PS, un message devrait apparaître donnant la possibilité de *Run Block Automation*.

- Cliquez sur *Run Block Automation du bloc* pour exécuter ce processus.

Cela applique une configuration par défaut pour le PS de votre carte.





Vous êtes maintenant prêt à commencer à créer votre design.

4.3 Ajouter des blocs à votre conception

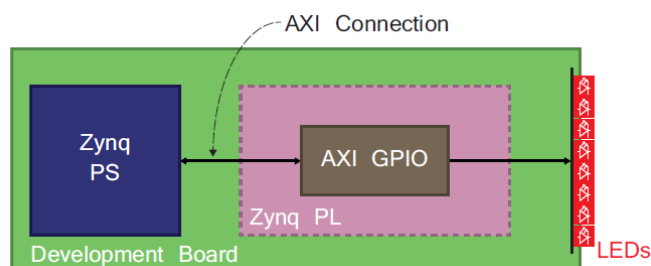
À ce stade, vous pouvez commencer à ajouter des blocs à votre conception. Vous pouvez ajouter une IP à partir du catalogue Vivado ou ajouter votre propre IP personnalisée.

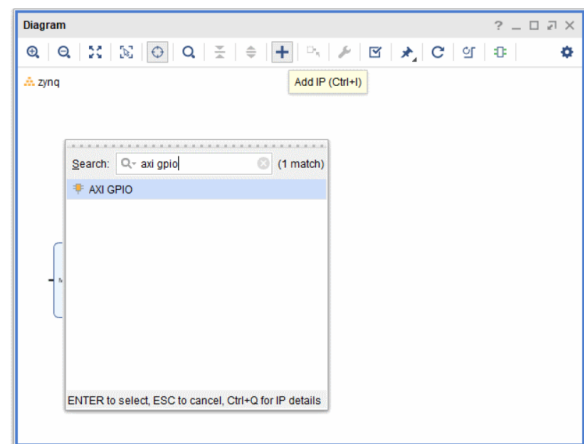
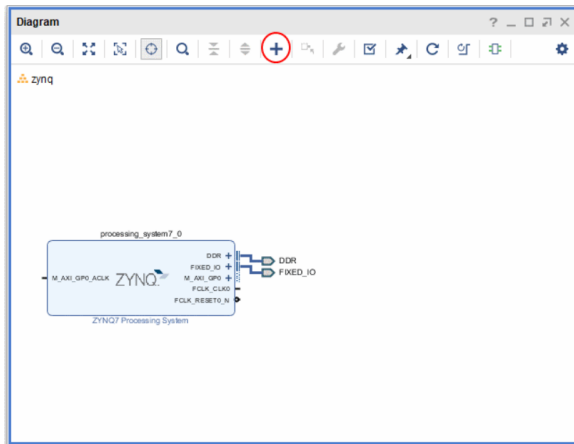
4.3.1 Aperçu de la conception matérielle

Cette étape consiste à présenter la disposition pour implémenter un contrôleur GPIO dans le Zynq PL.

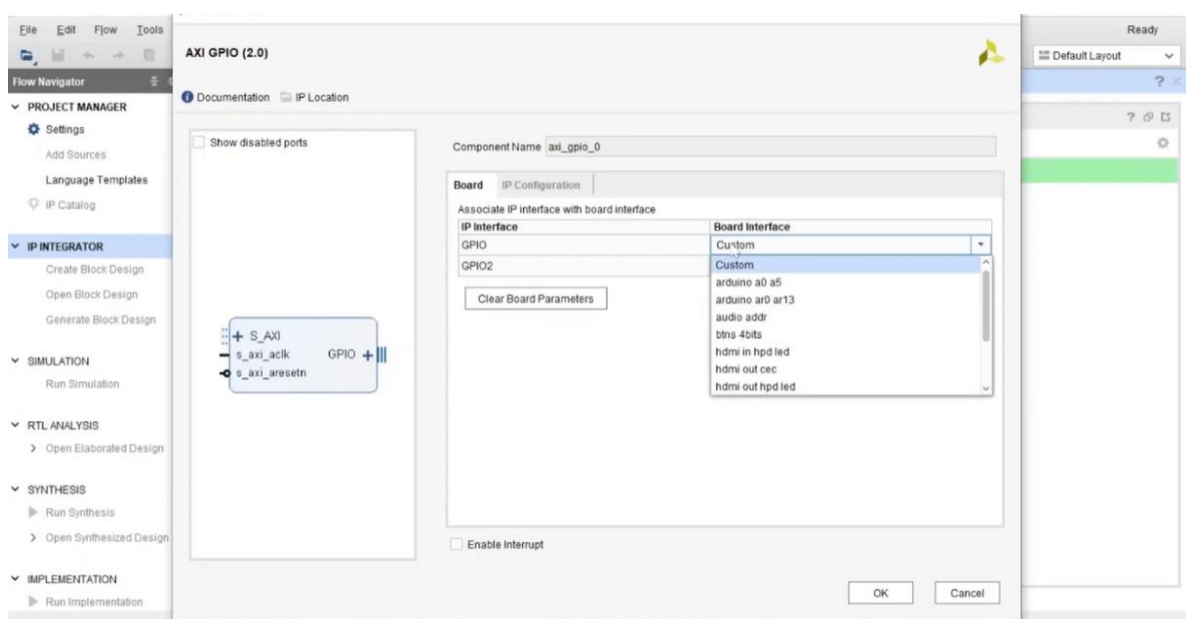
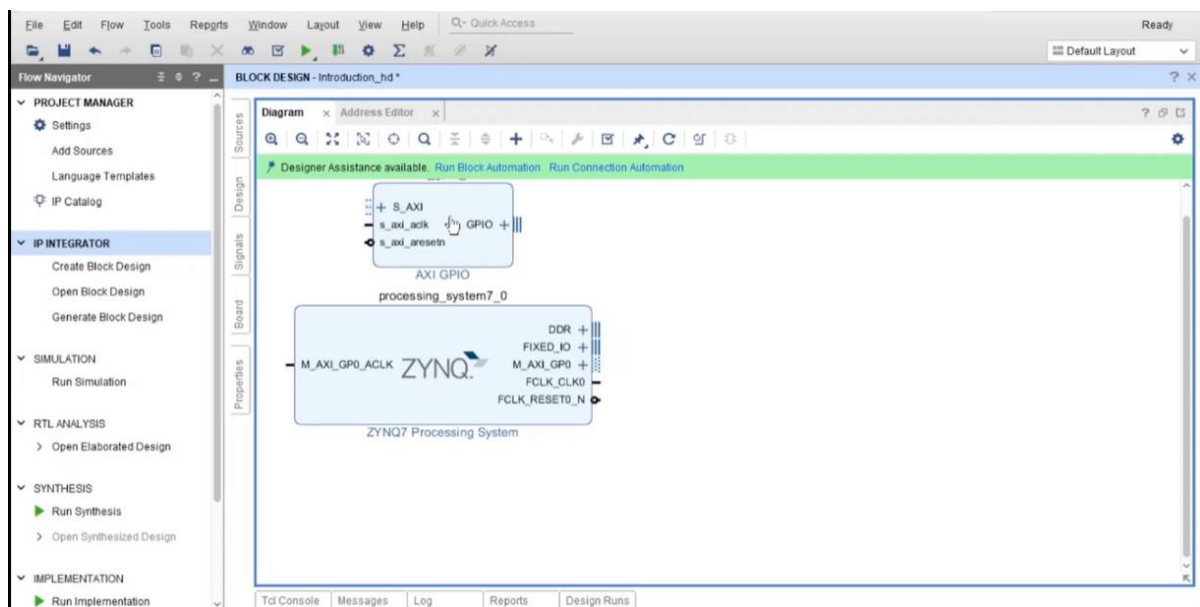
L'outil Vivado IP Integrator sera utilisé pour créer cette interface à l'aide de l'environnement graphique fourni, et les outils Designer Assistance fournis seront utilisés pour la configuration et la connexion automatique d'une interconnexion AXI qui connecte le Zynq PS au module IP du PL. Le contrôleur GPIO sera connecté aux LEDs disponibles sur le Pynq Board.

Un aperçu de la conception matérielle qui sera créée dans cet exercice est fourni dans la figure :

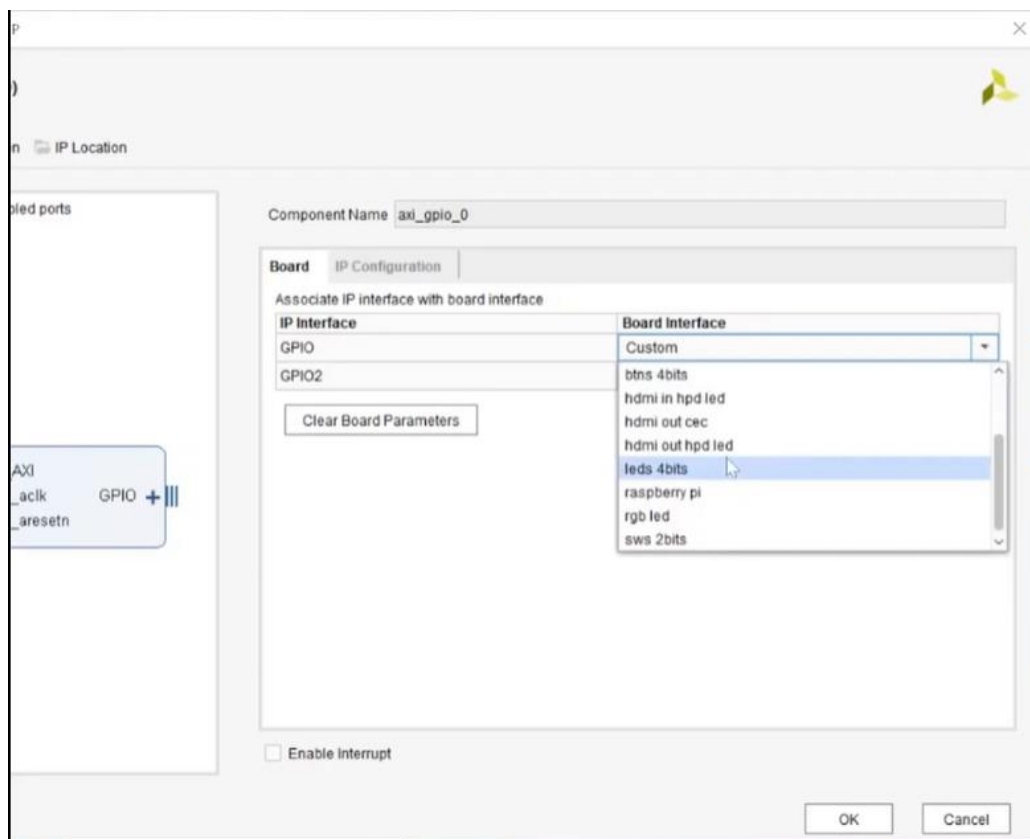




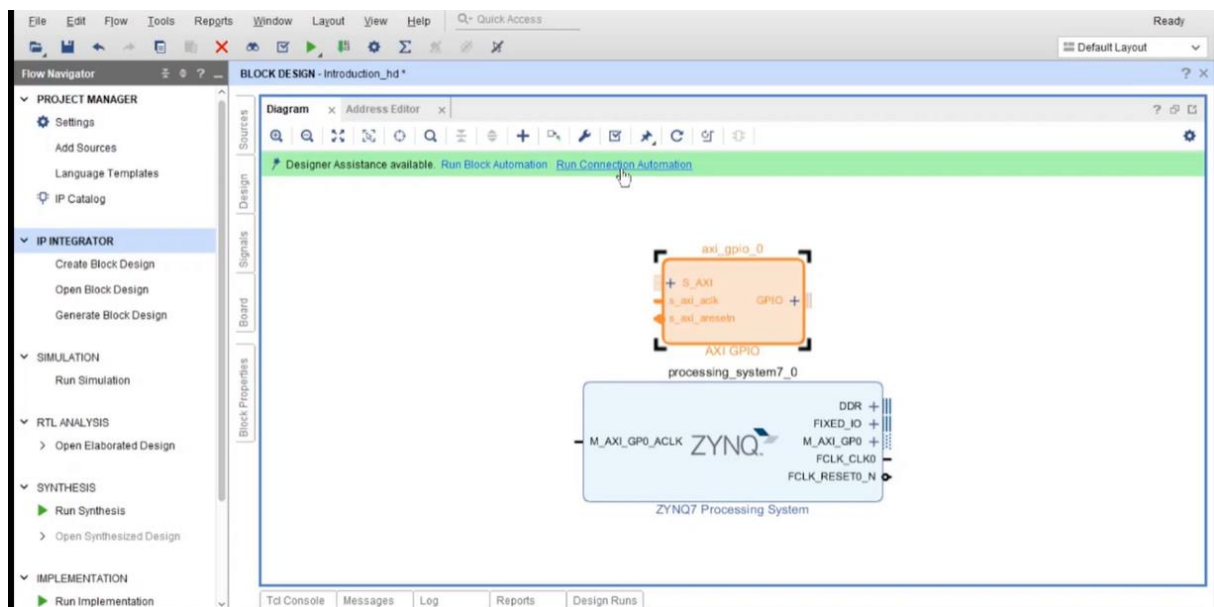
➤ Double cliquer sur le composant AXI GPIO



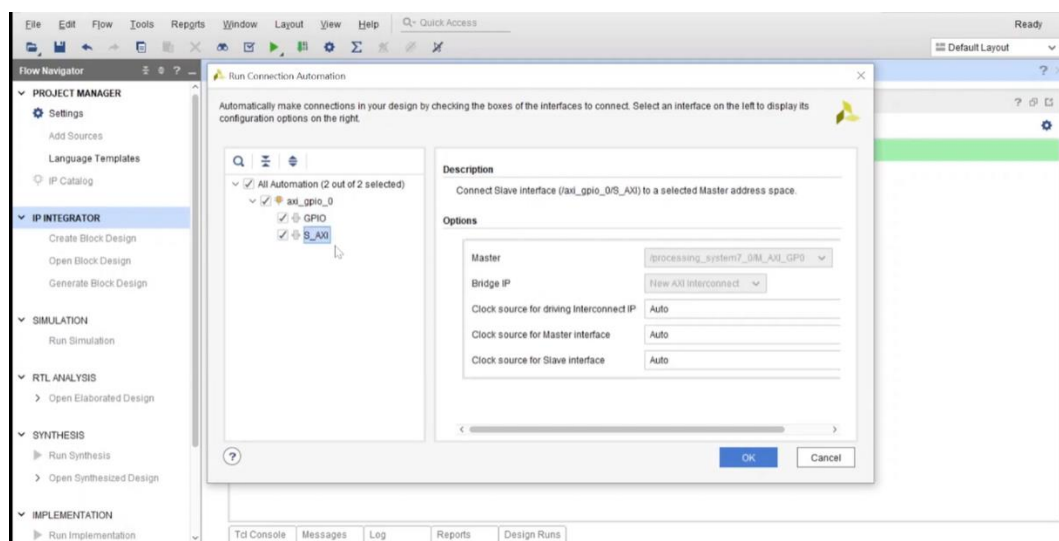
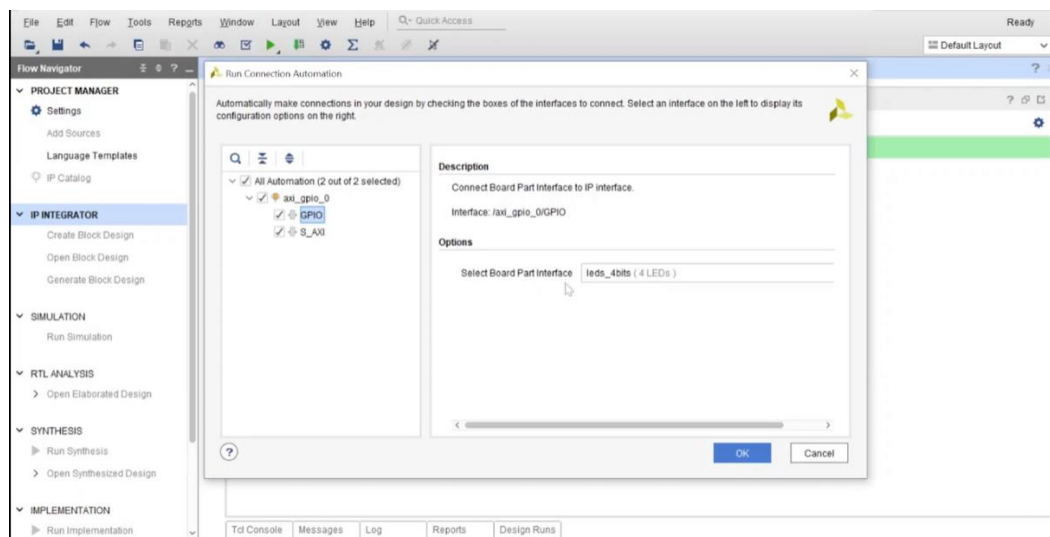
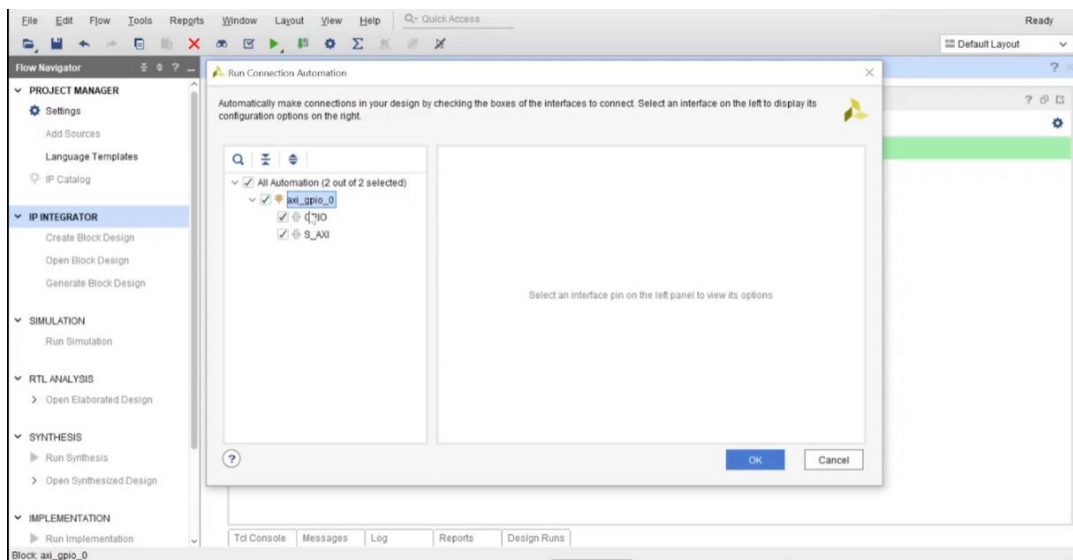
- Sélectionner les connexions dAXI GPIO : Leds_4bits



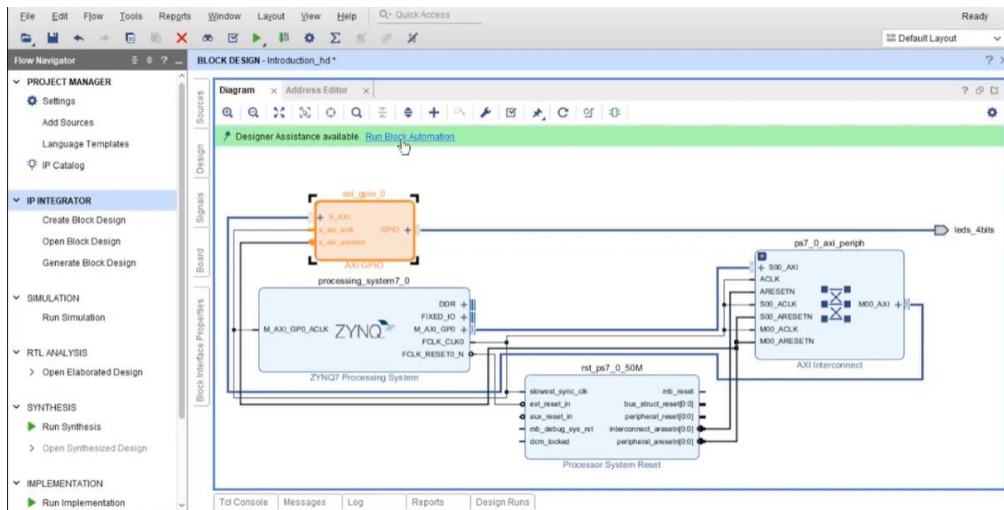
- Appuyer sur **Run Connection Automation**



➤ Cocher les différentes connexions

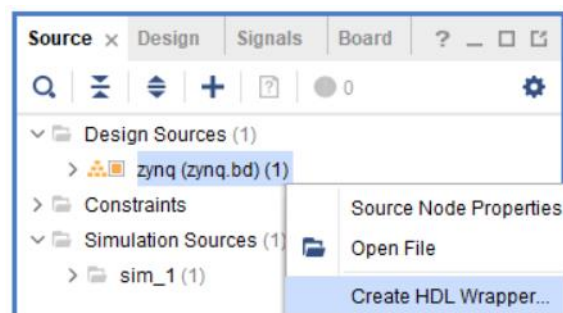


Le résultat des connexions est comme suit :



La conception est maintenant terminée. Nous avons besoin d'un fichier top level HDL pour le projet, qui peut être créé automatiquement.

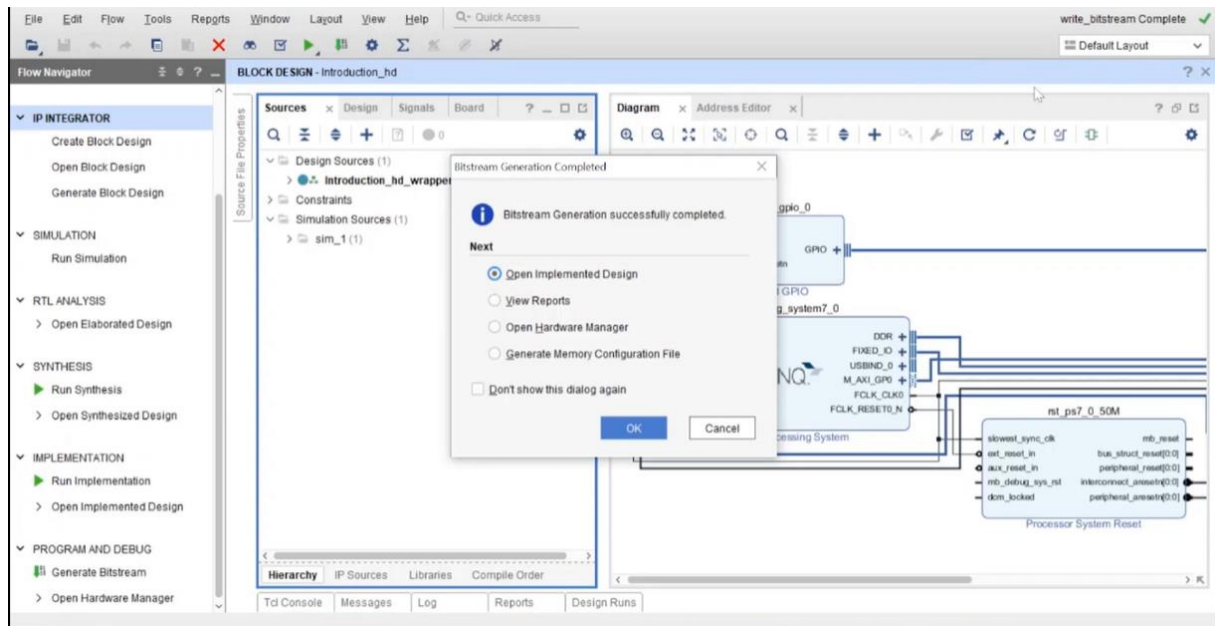
- Dans l'onglet Source, cliquez avec le bouton droit sur zynq.bd (fichier de diagramme) et sélectionnez **Créer un wrapper HDL**.



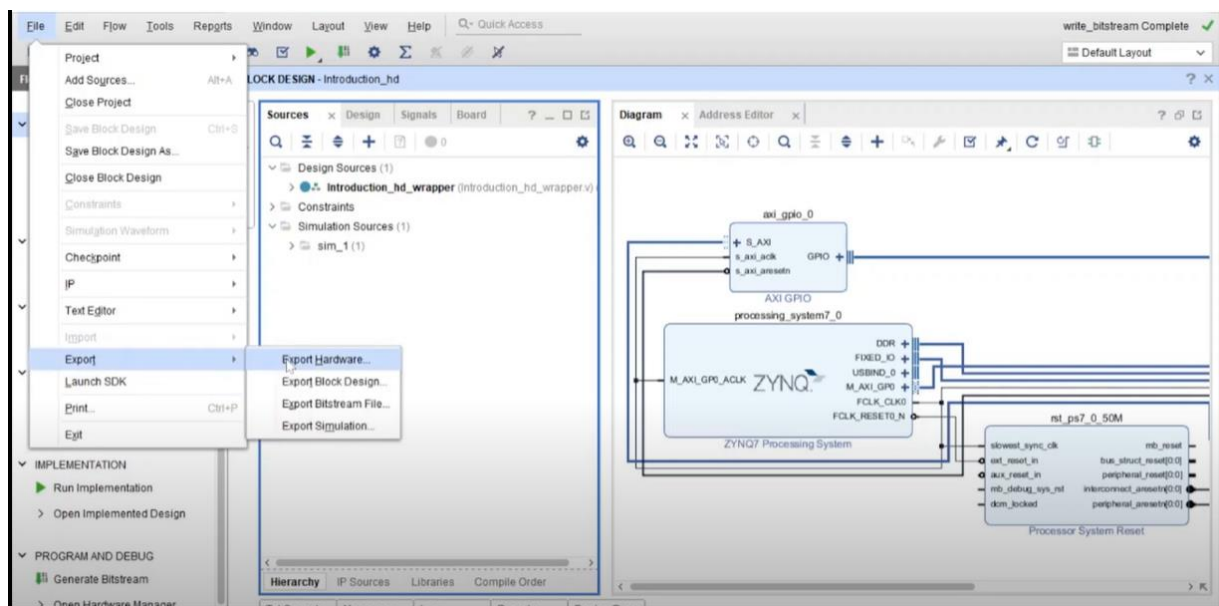
Notez qu'un wrapper VHDL ou Verilog peut être créé, en fonction des paramètres du projet. (Si vous avez une préférence, vous pouvez également définir Verilog ou VHDL comme valeur par défaut dans les paramètres du projet.)

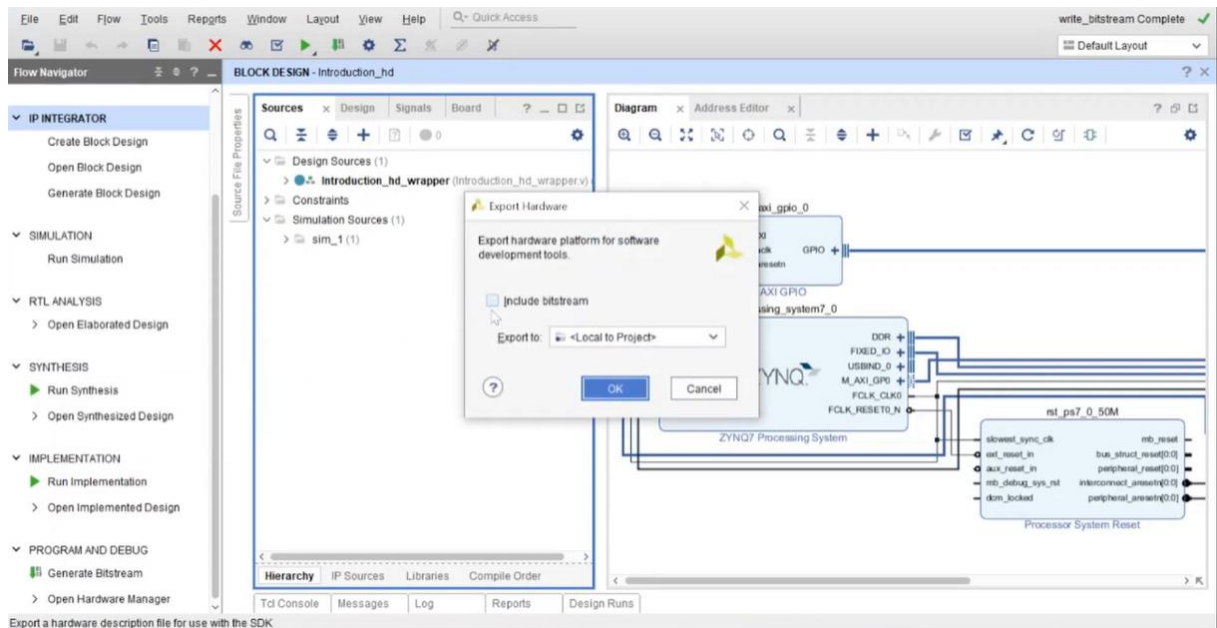
- Dans Vivado, cliquez sur **Generate Bitstream** (sous Program and Debug) en bas du *Flow Navigator*.

Vous remarquerez peut-être un certain nombre d'informations, d'avertissements et de messages dans le (même si, espérons-le, il n'y aura pas de messages d'erreur pour cette conception !). Ces messages peuvent être ignorés pour l'instant.



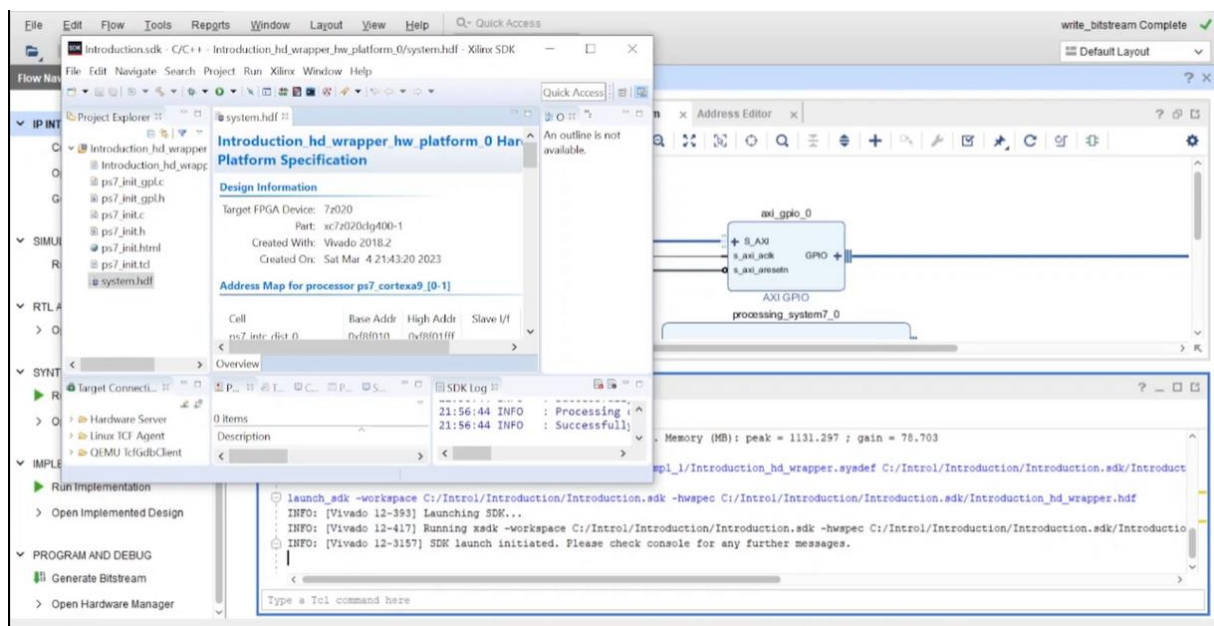
- Une fois le bitstream créé, générez le Tcl correspondant pour la conception en sélectionnant **File > Export > Export Block Design** . (Assurez-vous que la conception du bloc est ouverte lors de cette opération.)



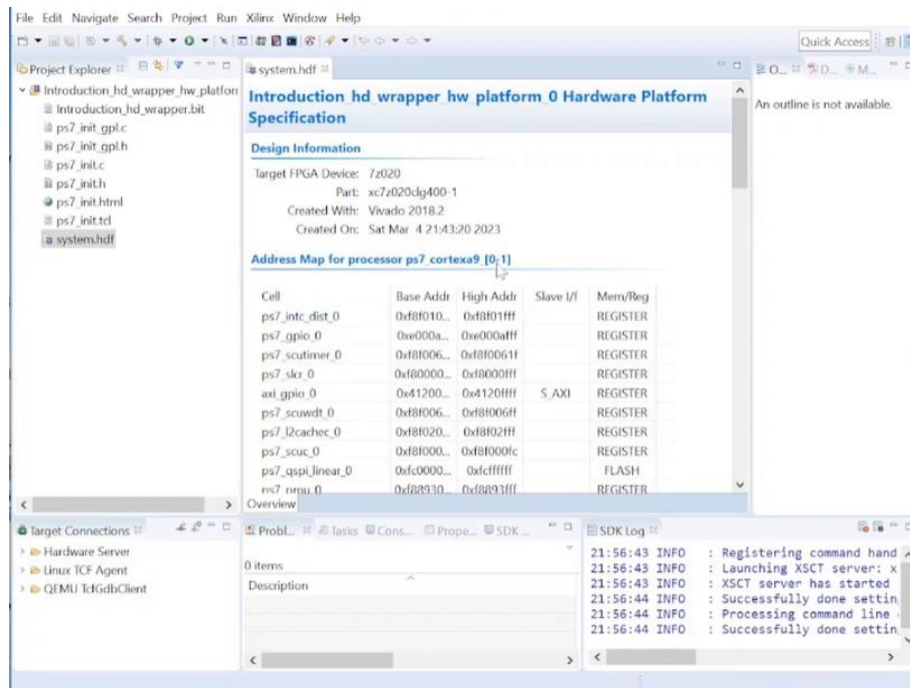


5. CRÉER UNE CONCEPTION LOGICIELLE POUR PYNQ

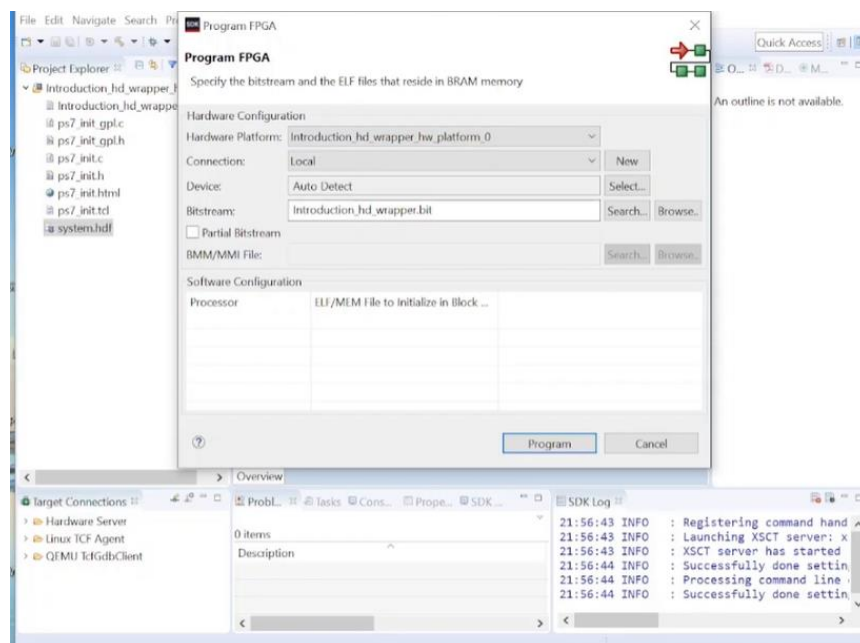
Après avoir exporter le design, une interface de développement logiciel se lance :



Sur cette interface, on peut visualiser les différents registres du processeur.

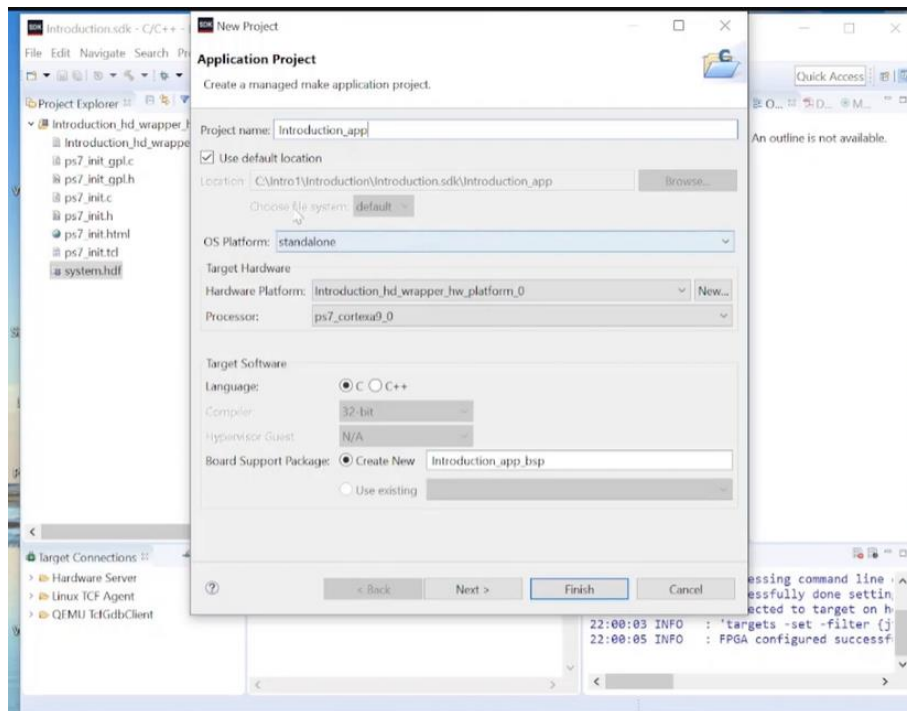


➤ Programmation du FPGA :

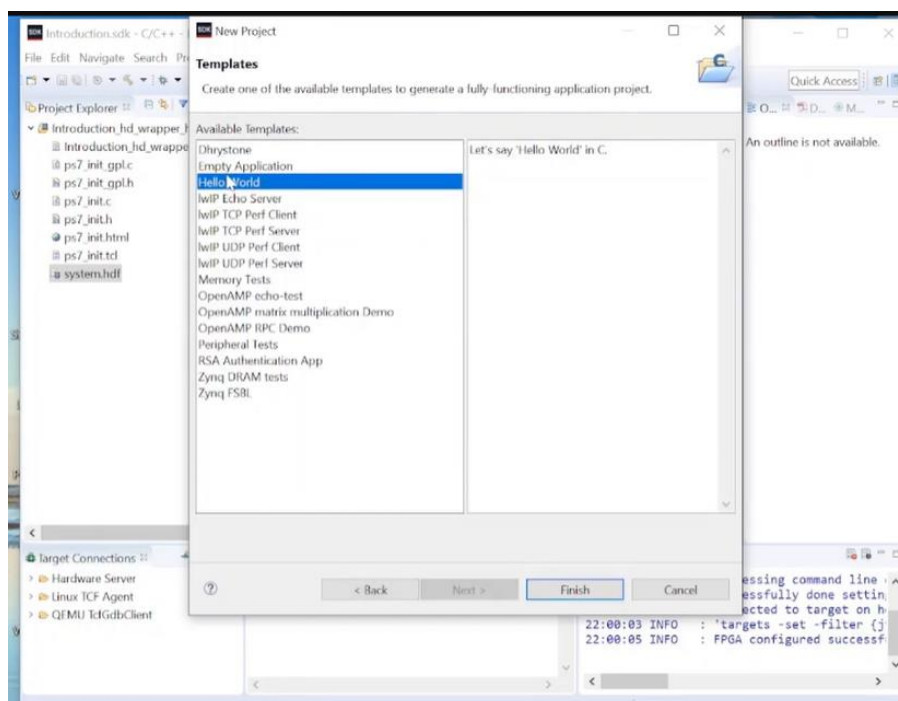


5.1 Création d'une nouvelle application

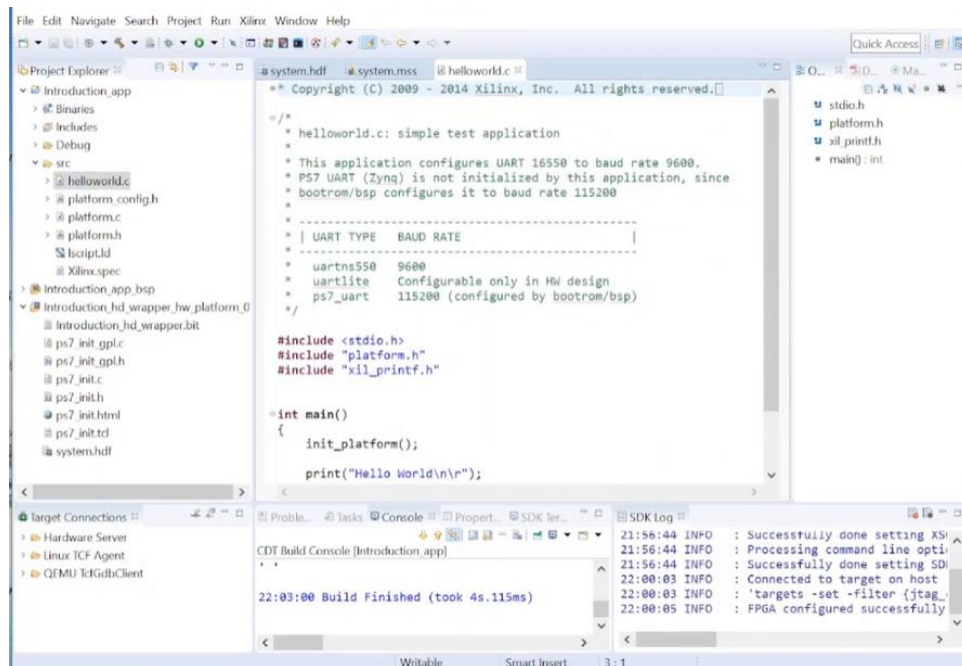
Pour la création d'une nouvelle application, appuyer s **File → New → Application Project**



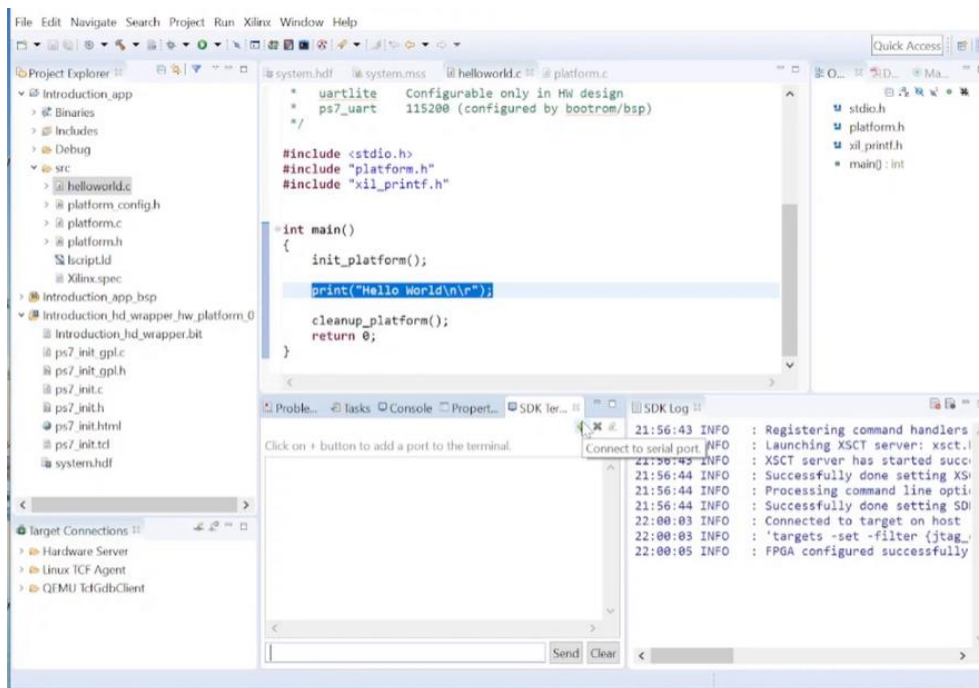
- Appuyer sur **Next** pour choisir un Template de programme.



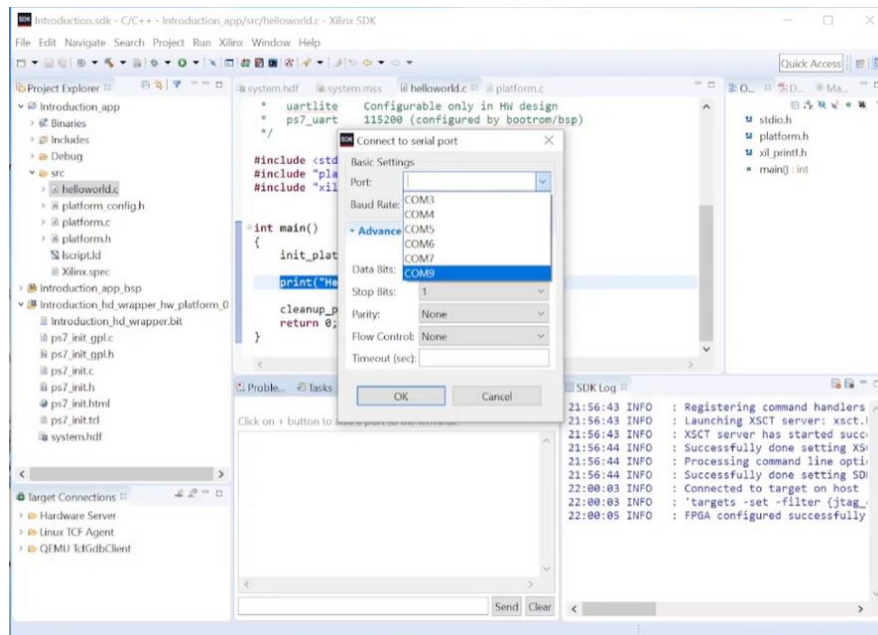
- Choisir un Template **Hello World**



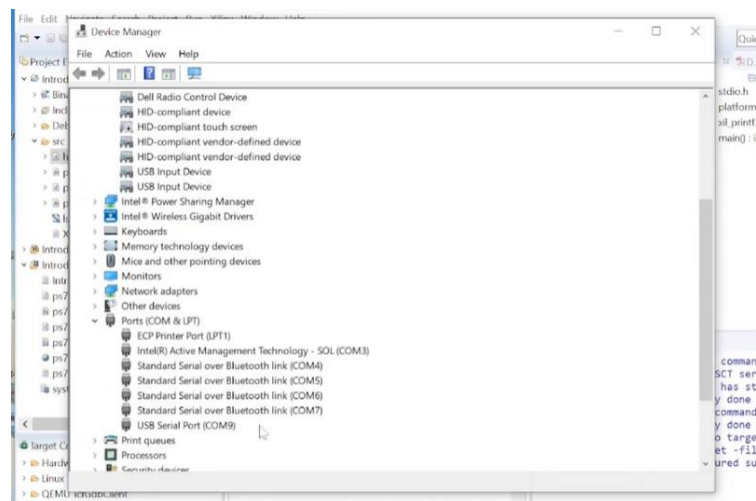
- Pour tester le programme, appuyer sur **Connect to serial port** dans **SDK Terminal**



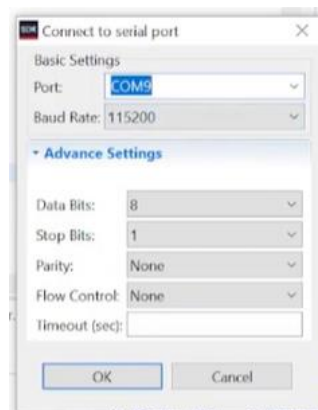
- Choisir le port où s'est branché la carte :



Pour savoir à quel port, la carte est branchée, vérifier sur **Device Manager**



➤ Choisir la configuration suivante :

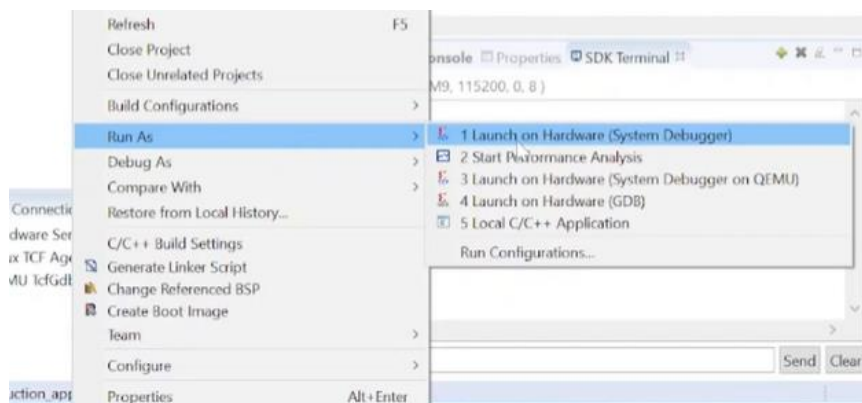
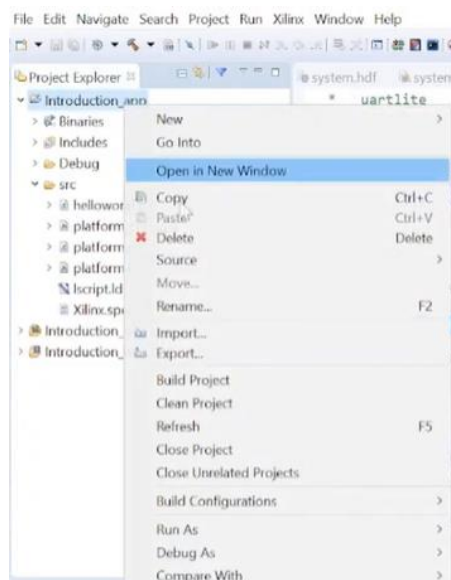


Le résultat de la connexion sera :

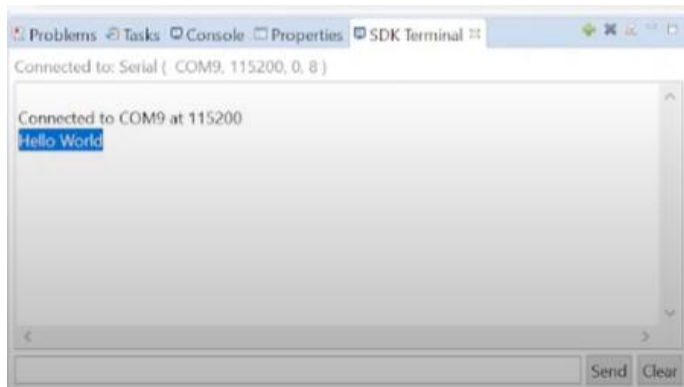


5.2 Exécution des programmes

1. Programmer la FPGA en appuyant sur l'icône **Program FPGA**
2. **Build** le code
3. **Lancer** le programme



Le résultat de l'exécution est comme suit :



5.3 Programme d'utilisation des Leds de la carte Pynq

Modifier le programme Hello World comme suit :

```
//-----  
// LOOK AT THIS CODE FROM THE ZYNQ PS PERSPECTIVE  
// As if we were the Zynq chip, who receives this code as  
// a task to carry out.  
//-----  
/* Include Files */  
#include "platform.h"  
#include "xparameters.h"  
#include "xgpio.h"  
#include "xstatus.h"  
#include "xil_printf.h"  
#include "sleep.h"  
  
//Interfaces of GPIO 0:  
#define LED_CHANNEL 1 // GPIO port/interface for LEDs  
  
//Instance of the GPIO Device Driver.  
  
XGpio Gpio0;  
  
void ToggleLeds(int ledData)  
{  
    XGpio_DiscreteWrite(&Gpio0, LED_CHANNEL, ledData);  
    usleep(500*1000);  
    return;  
}  
  
/* Main function. */  
int main(void){  
  
    int led= 0b00001111;
```

```

// Variable to store the address of Gpio(s) predefined configuration
XGpio_Config *cfg_ptr0;

// Initialize xilinx platform
xil_printf("initialization of the platform\n\r");
init_platform();

//-----
// INITIALIZE and CONFIGURE (or SETUP) the GPIO(s):
//-----
//1. INTIALIZATION STEP:
    //1.1 Look up for the predefined configuration of the GPIO 0:
    cfg_ptr0 = XGpio_LookupConfig(XPAR_AXI_GPIO_0_DEVICE_ID); // GPIO 0
    cfg_ptr1 = XGpio_LookupConfig(XPAR_AXI_GPIO_1_DEVICE_ID); // GPIO 1

    //1.2 Initialize GPIO 0:
    XGpio_CfgInitialize(&Gpio0, cfg_ptr0, cfg_ptr0->BaseAddress); // GPIO 0

//2. CONFIGURE STEP
// Configure the interfaces of GPIO 0 in software
// to be compatible with the hardware design of those Gpio(s):
//On GPIO 0:
    XGpio_SetDataDirection(&Gpio0, LED_CHANNEL, 0x00); // channel 1 (leds),
output mode

//-----
// USING THE GPIO(s).
//-----

ToggleLeds(led);

// Cleanup the xilinx platform
xil_printf("cleanup\n\r");
cleanup_platform();

return 0;
}

```

Exécuter le code et Vérifier sur la carte Pynq

6. Terminologie

- **AXI - Advanced eXtensible Interface** - une interconnexion qui fait partie de la norme ARM AMBA. La plupart des IP Xilinx prennent en charge cette norme et elle est utilisée dans Vivado pour connecter les IP entre elles.
- **GPIO General Purpose Input Output.** - Entrée-sortie à usage général.
- **GP Zynq General Purpose AXI port(s)** - Port(s) AXI à usage général Zynq. Utilisé pour connecter des périphériques qui seront contrôleurs du système de traitement Zynq.
- **HP Zynq High Performance AXI port(s)** - Port(s) AXI hautes performances Zynq. Utilisé pour connecter IP directement au contrôleur de mémoire Zynq. Cela permet à IP d'accéder directement au système de mémoire Zynq.
- **(Vivado) IP Integrator** - L'environnement graphique Vivado pour créer des conceptions matérielles (conceptions de blocs) pour Zynq.
- **BRAM Block RAM (Memory) on Xilinx devices** - Bloquer la RAM (mémoire) sur les circuits Xilinx.
- **PL - Programmable Logic** - la partie FPGA de la puce Zynq.
- **PS - Processing System** - le processeur ARM, les périphériques et le contrôleur de mémoire à l'intérieur d'une puce Zynq.

Bon courage