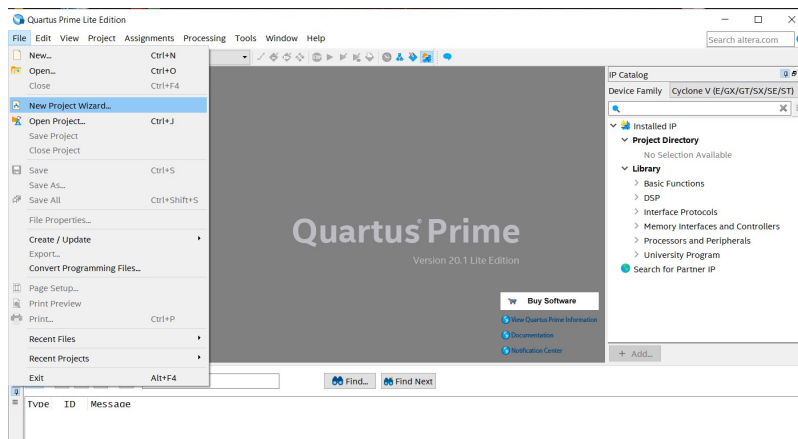
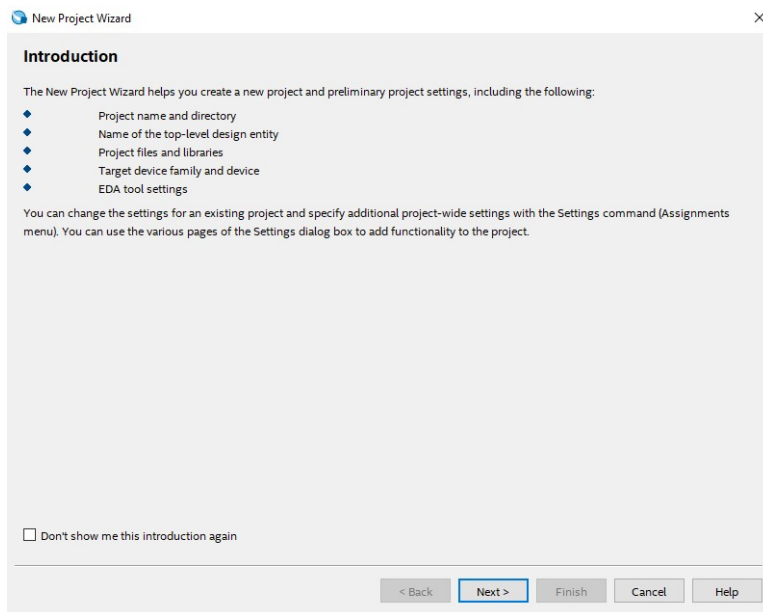
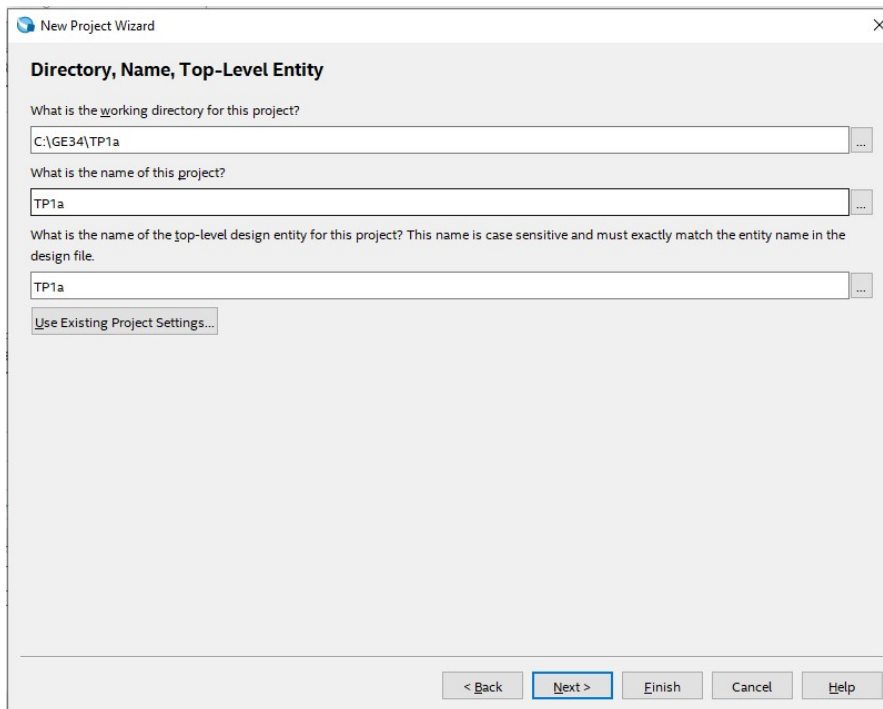


**TP1 M34****Partie a)****Introduction au Quartus version 2020  
Quartus (Quartus Prime 20.1) Lite Edition****1) Ouvrir Quartus 20.1 et choisir New Project Wizard****2) Une fenêtre s'ouvre, cliquer Next****3) Remplir les champs comme est indiqué dans la figure suivante.**



**New Project Wizard**

**Directory, Name, Top-Level Entity**

What is the working directory for this project?

C:\GE34\TP1a

What is the name of this project?

TP1a

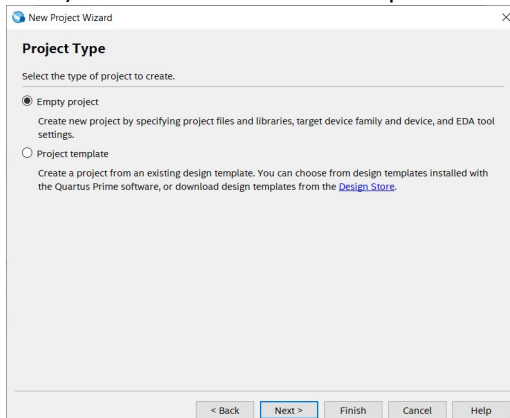
What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.

TP1a

Use Existing Project Settings...

< Back   **Next >**   Finish   Cancel   Help

#### 4) Une fenêtre s'ouvre. Cliquer Next



**New Project Wizard**

**Project Type**

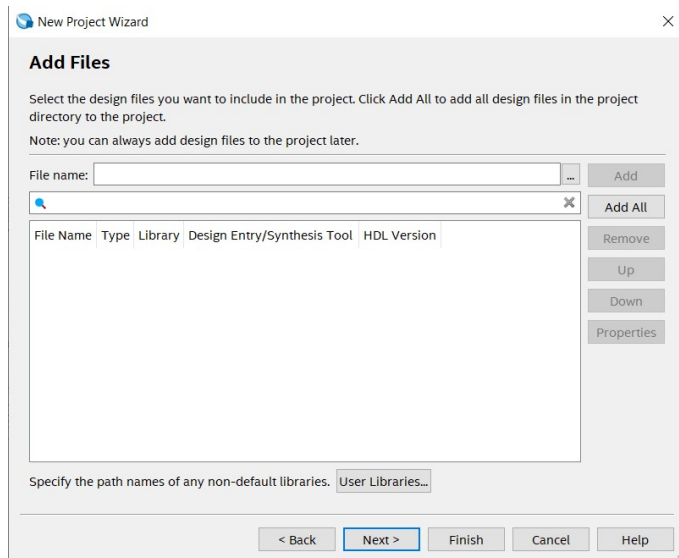
Select the type of project to create.

☒ Empty project  
Create new project by specifying project files and libraries, target device family and device, and EDA tool settings.

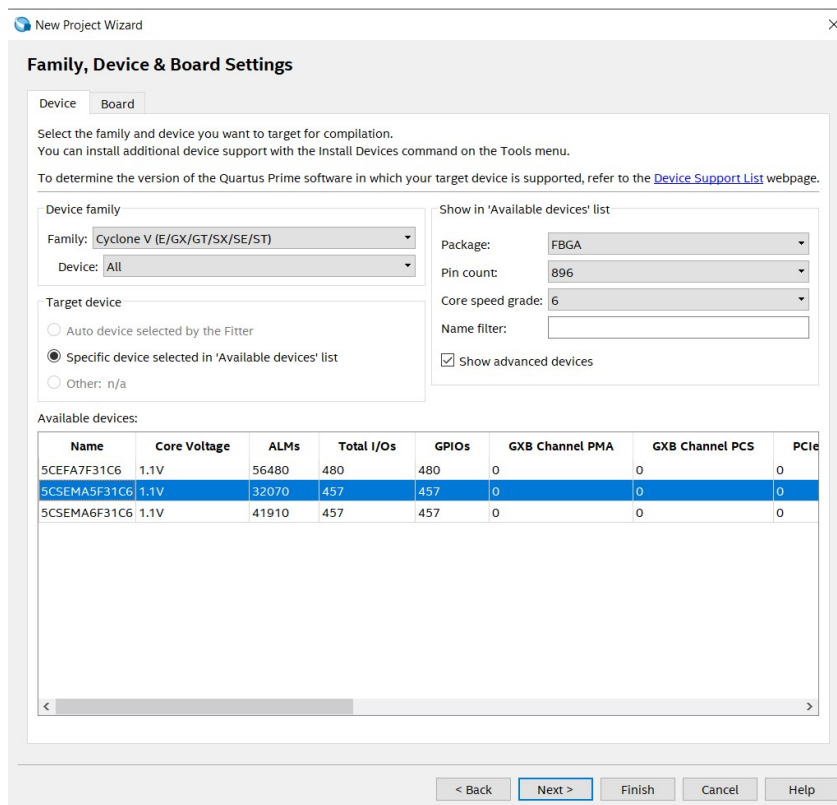
☐ Project template  
Create a project from an existing design template. You can choose from design templates installed with the Quartus Prime software, or download design templates from the [Design Store](#).

< Back   **Next >**   Finish   Cancel   Help

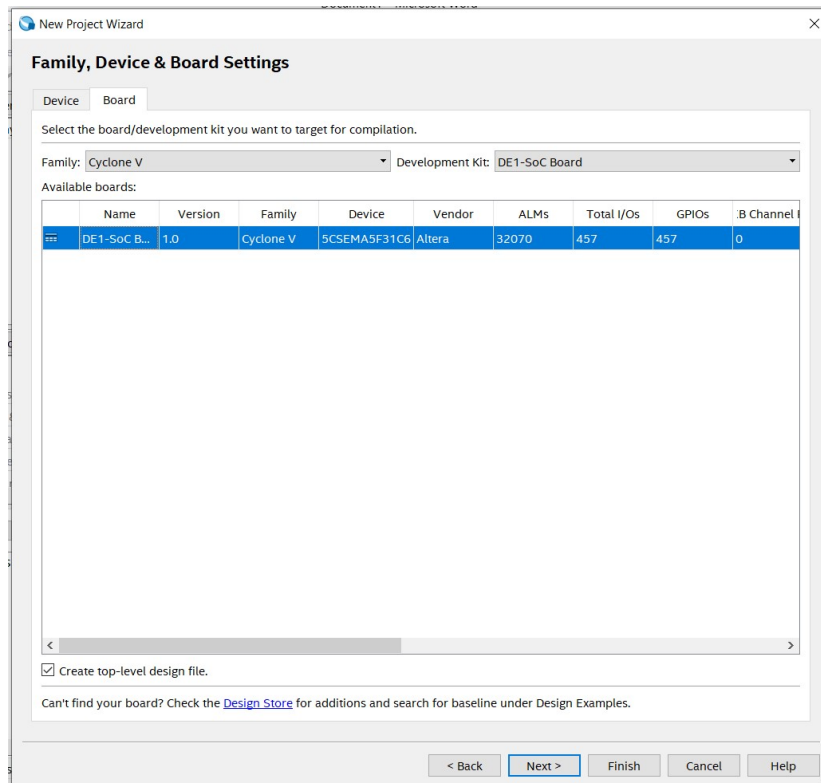
#### 5) Il n'y a pas de fichiers à ajouter au projet. Cliquer Next



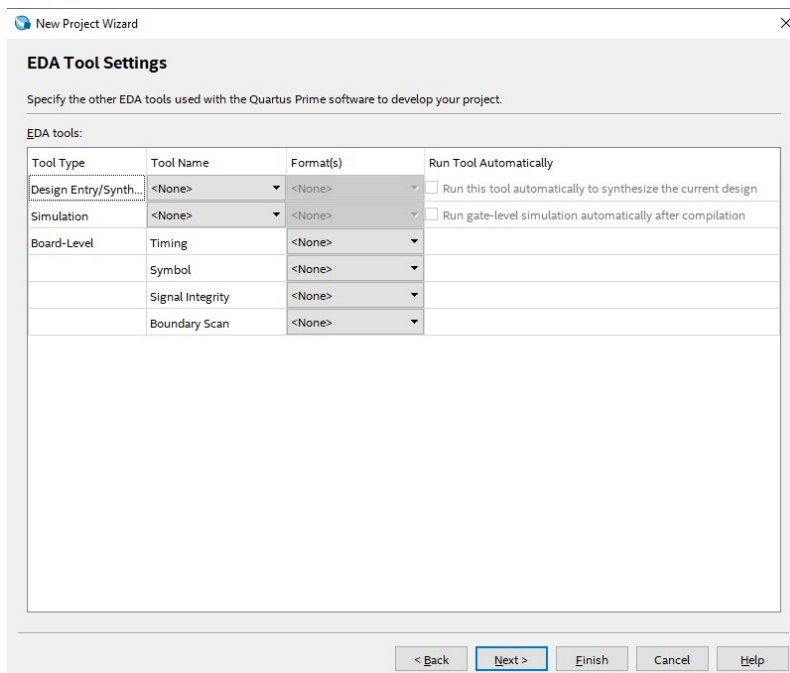
6) Faire les sélections comme est indiqué dans la figure suivante.



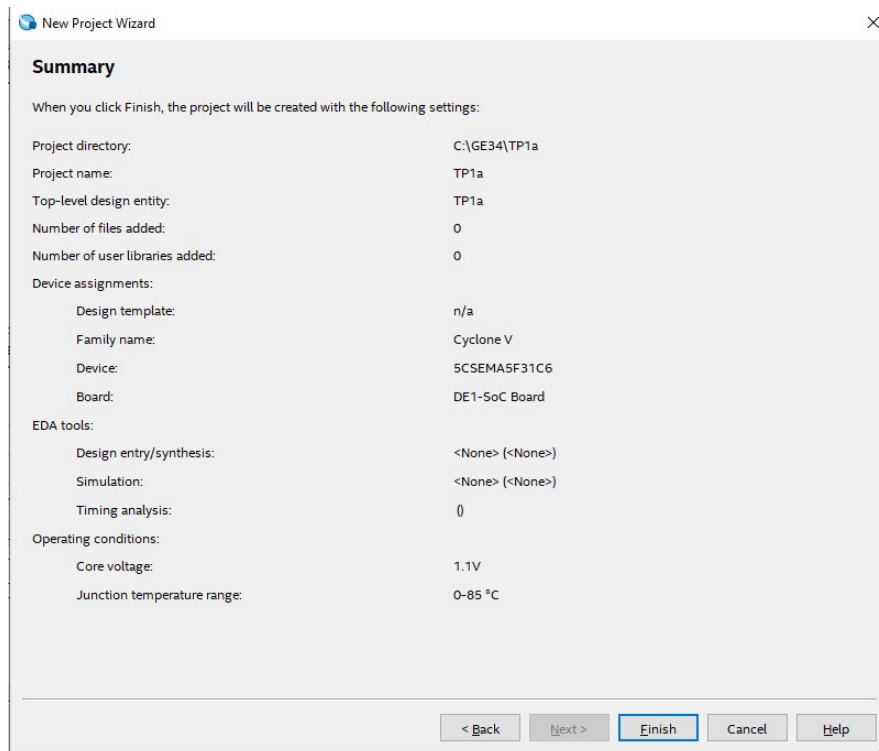
7) Cliquer sur Board. Faire les sélections comme est indiqué dans la figure suivante. Cliquer Next



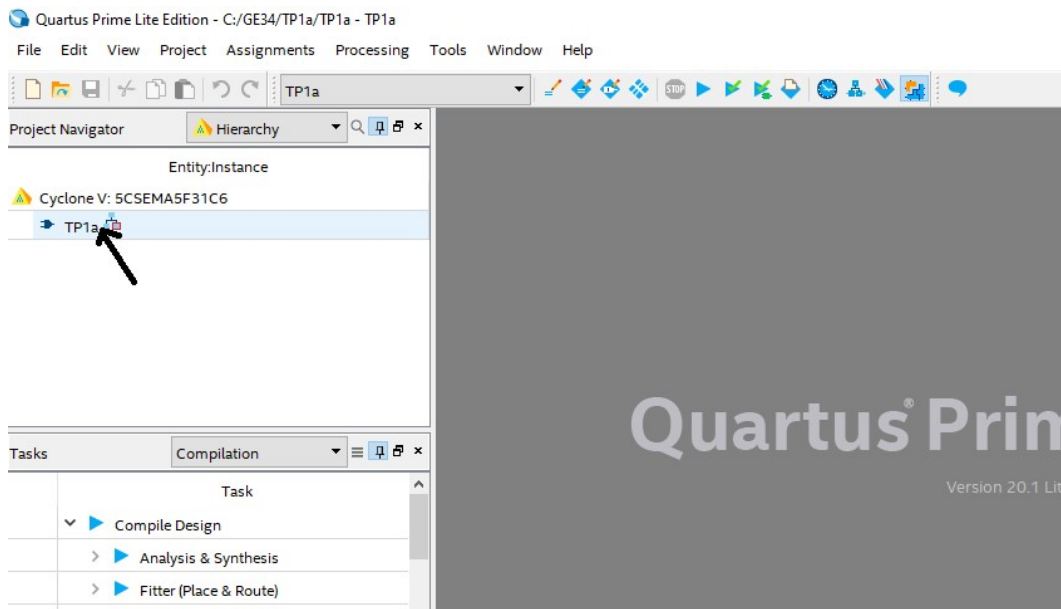
8) Cliquer Next sur la fenêtre « EDA Tool Settings »



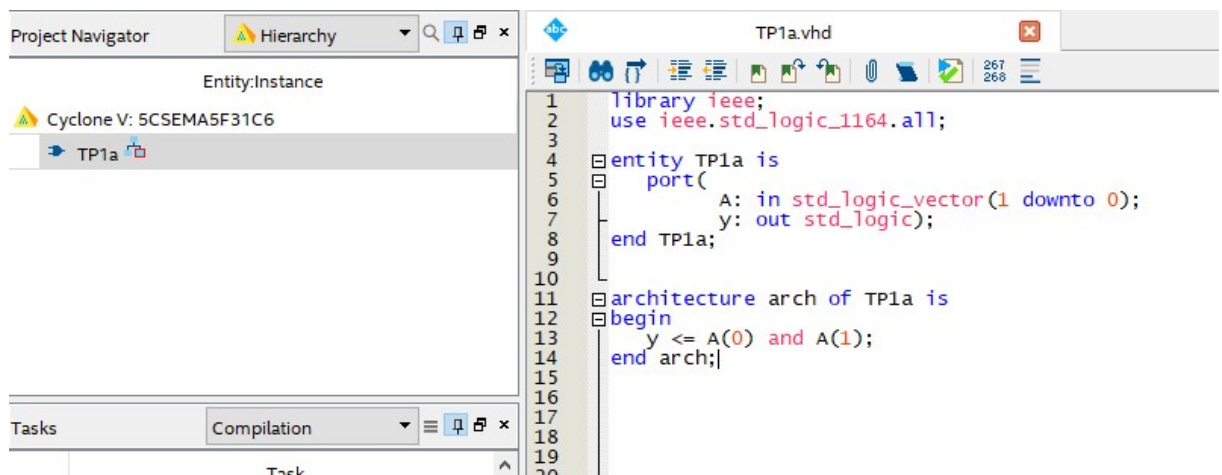
9) Cliquer Finish sur la prochaine fenêtre



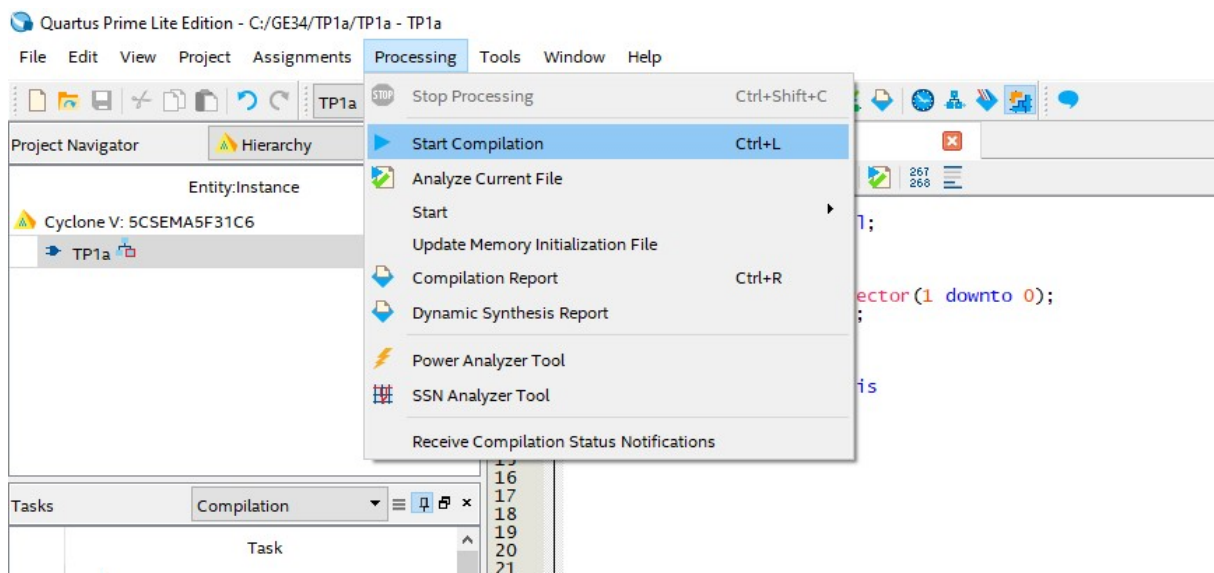
10) Double cliquer sur TP1a



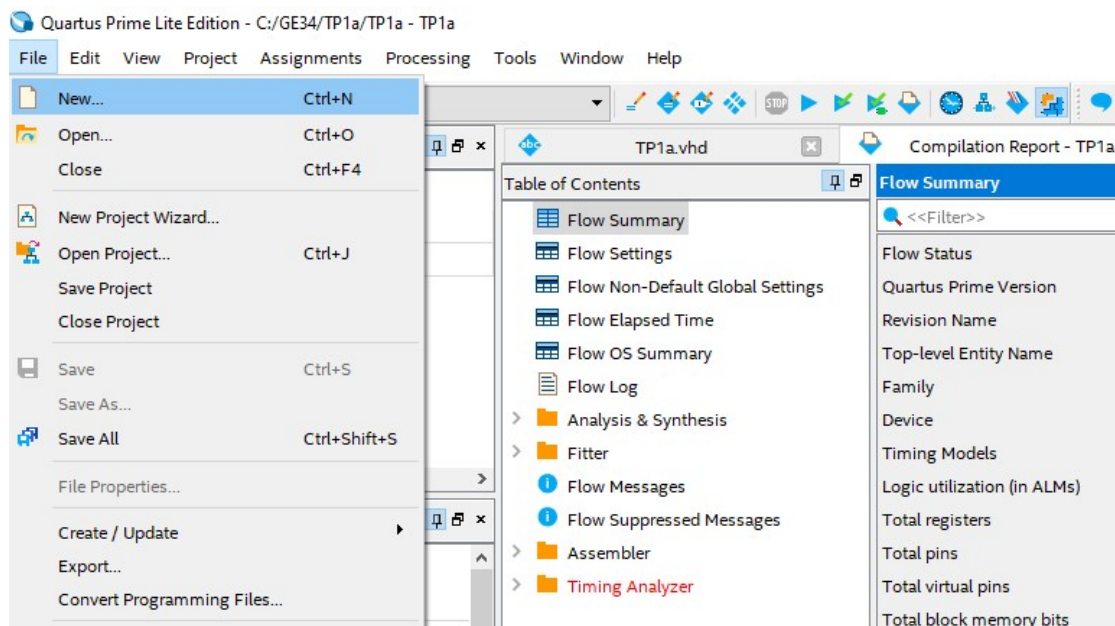
11) Le fichier TP1a s'ouvre.



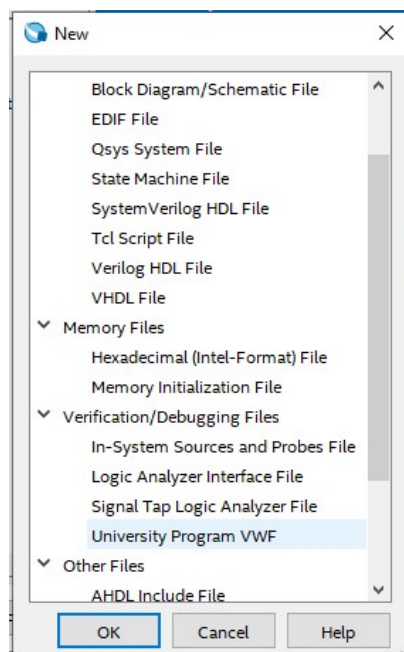
12) Choisir Start Compilation comme indiqué dans la figure



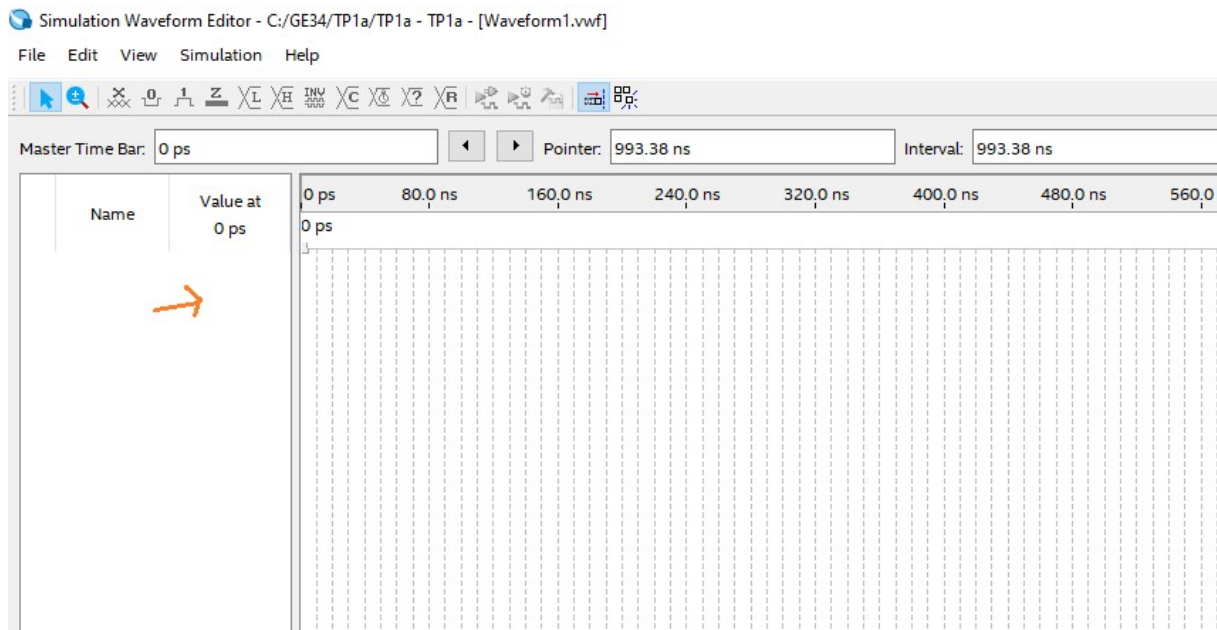
13) Attendre jusqu'à la fin de la compilation (100%) et choisir New sous File



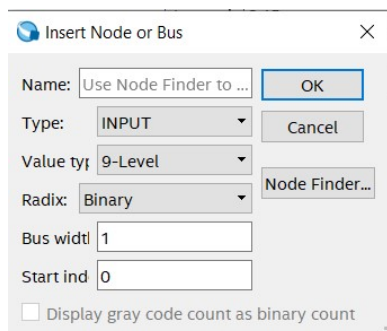
14) Une fenêtre s'ouvre. Sélectionner *University Program VWF* comme indiqué et cliquer OK



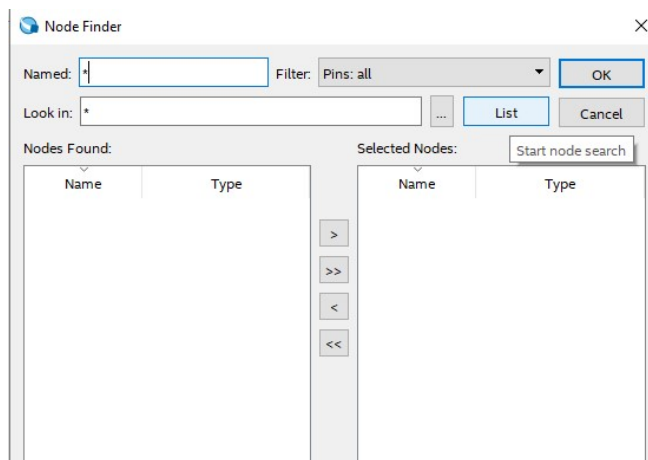
15) La fenêtre suivante s'ouvre.



16) Double cliquer sur la partie indiquée par la flèche rouge. Une autre fenêtre s'ouvre comme indiquée dans la figure suivante. Cliquer sur « *Node Finder* »

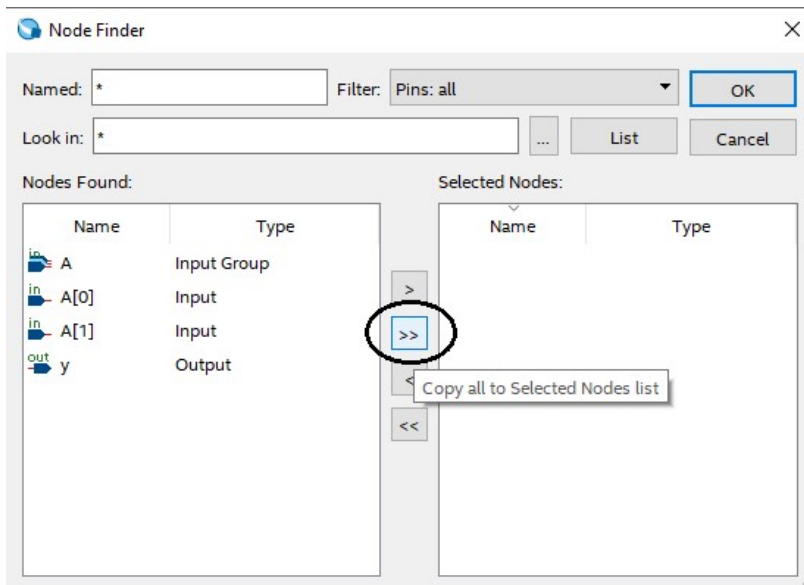


17) Une autre fenêtre s'ouvre. Cliquer sur « List »

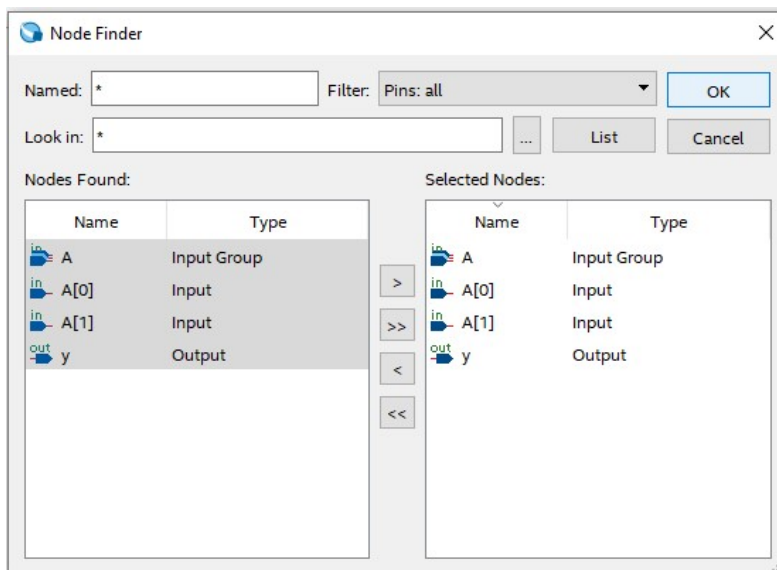




18) S'assurer que le Filtre indique Pins : all. Cliquer sur le symbole >> encerclé



19) Cliquer OK



20) Cliquer OK

**Insert Node or Bus** [X]

Name:  [OK]

Type:  [Cancel]

Value type:  [Node Finder...]

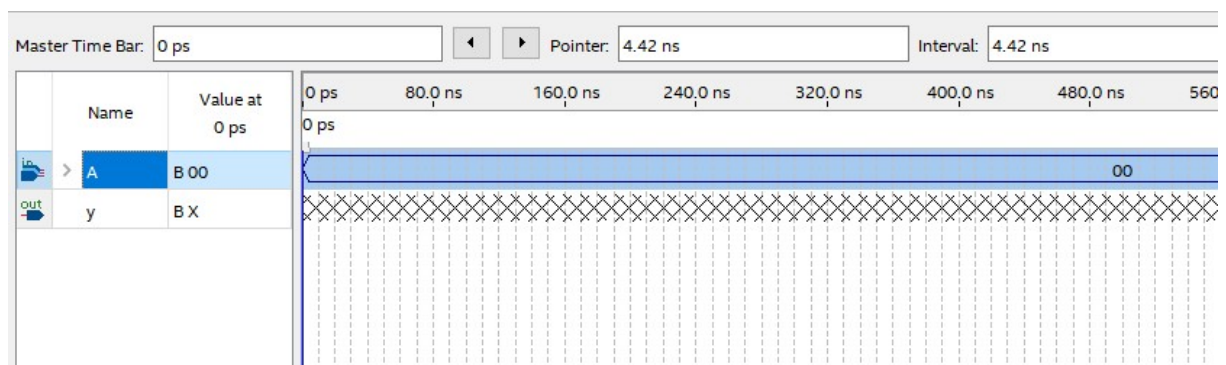
Radix:

Bus width:

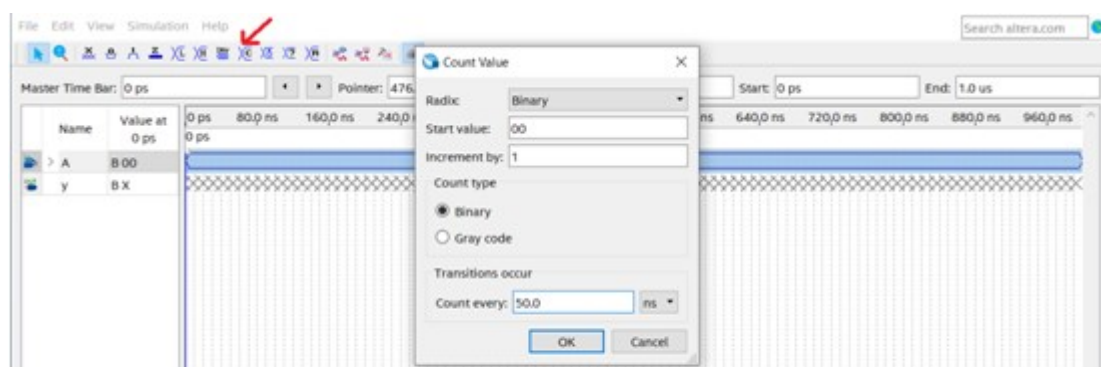
Start index:

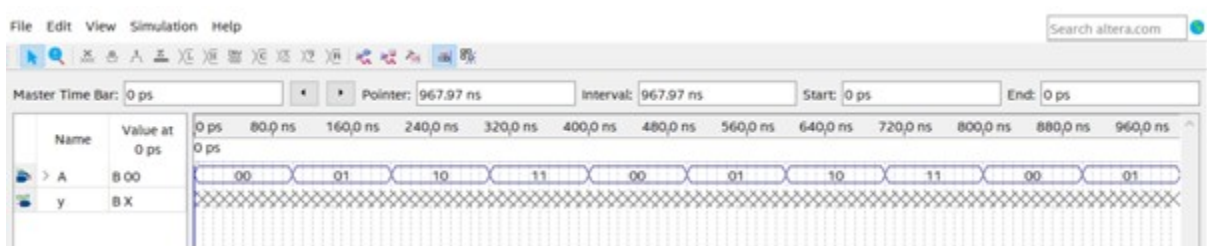
☐ Display gray code count as binary count

21) Sélectionner le port d'entrée A

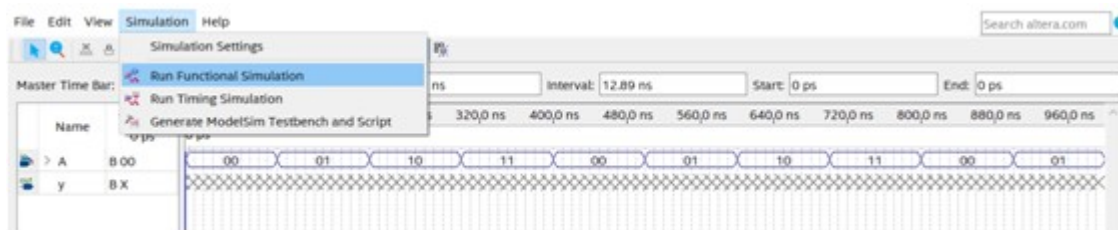


22) Sélectionner le symbole indiquée par la flèche rouge, et choisir count every 50 ns.

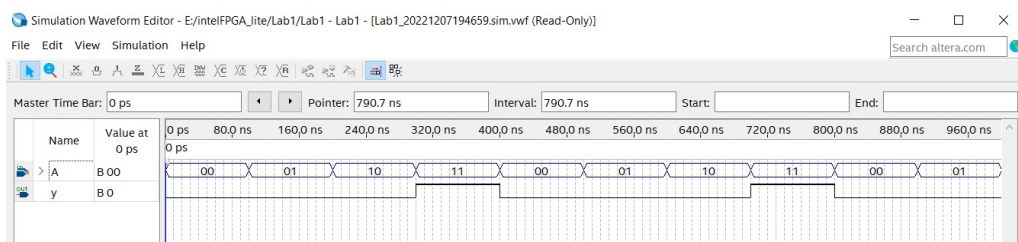




23) Choisir Run Functional Simulation sous Simulation



24) Enregistrer le fichier comme *waveform.vwf*. Le résultat de la simulation est affiché. Vérifier si c'est correct

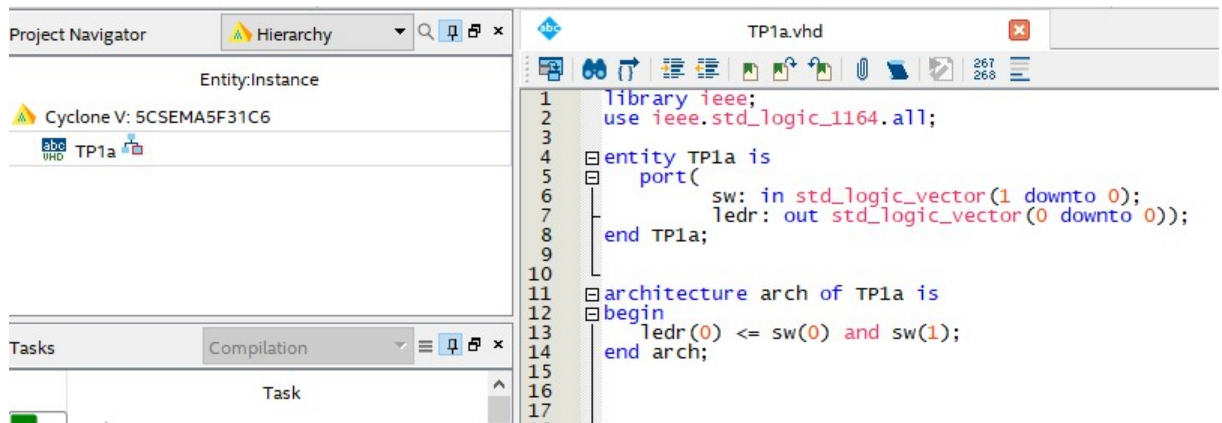


25) Changer le fichier TP1a comme ceci et l'enregistrer dans le même emplacement

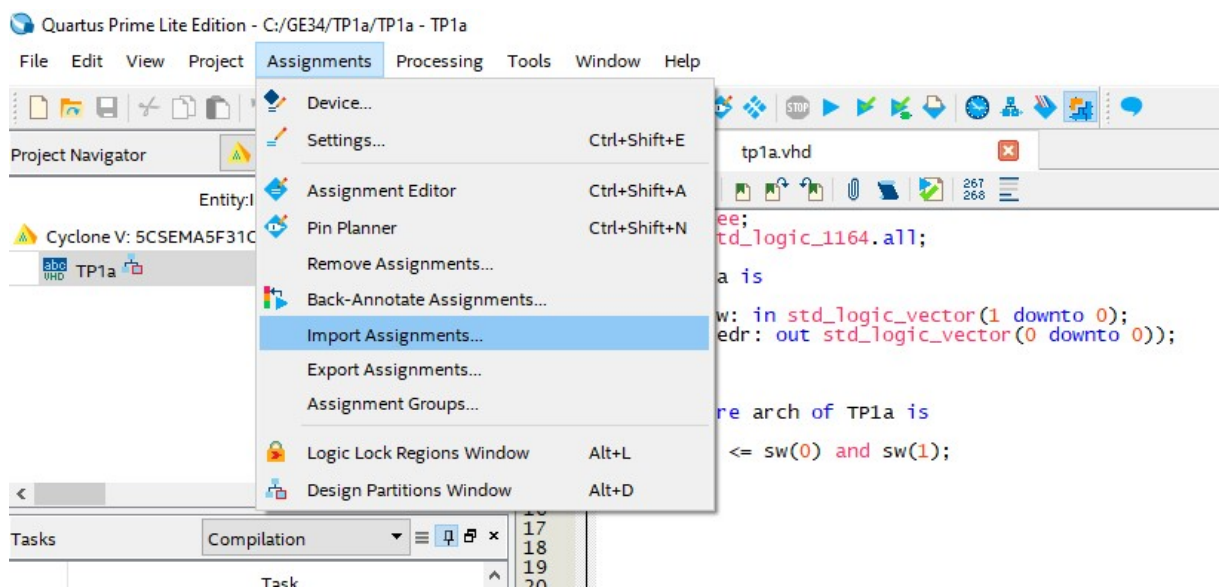
```
library ieee;
use ieee.std_logic_1164.all;

entity TP1a is
    port(
        sw: in std_logic_vector(1 downto 0);
        ledr: out std_logic_vector(0 downto 0));
end TP1a;

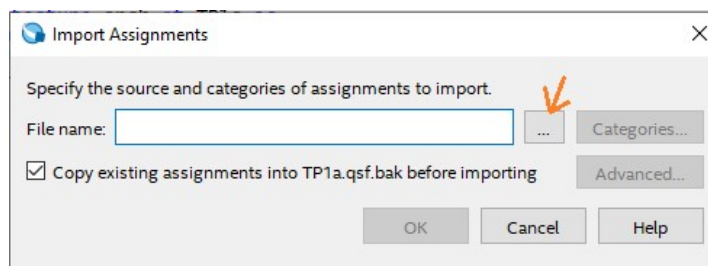
architecture arch of TP1a is
begin
    ledr(0) <= sw(0) and sw(1);
end arch;
```



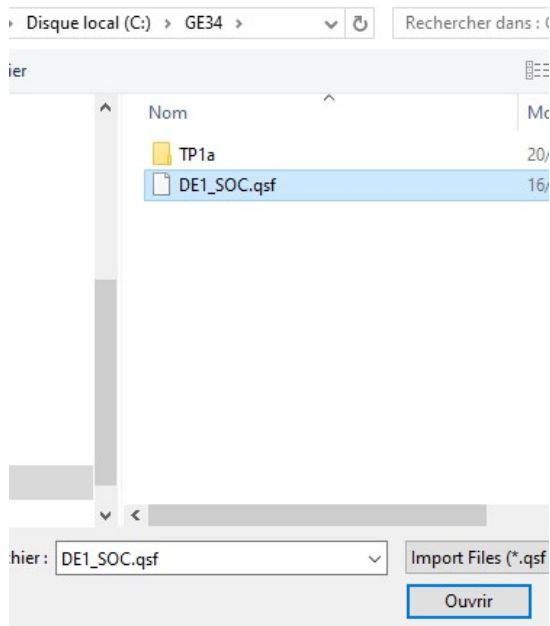
26) Choisir « Import assignments »



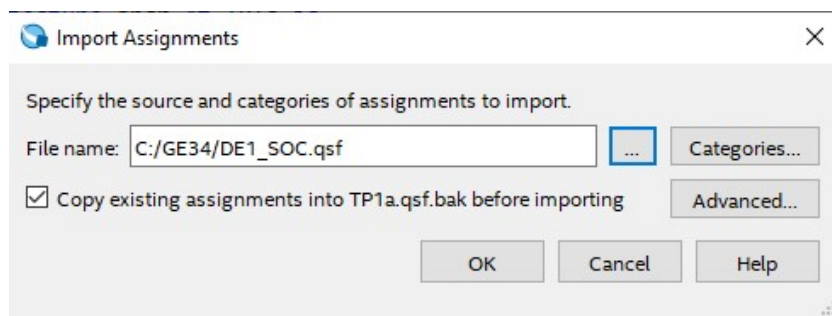
27) La fenêtre suivante s'ouvre. Cliquer sur le champs indiqué



28) Importer le fichier suivant qui est sous le dossier GE34. Cliquer sur Ouvrir

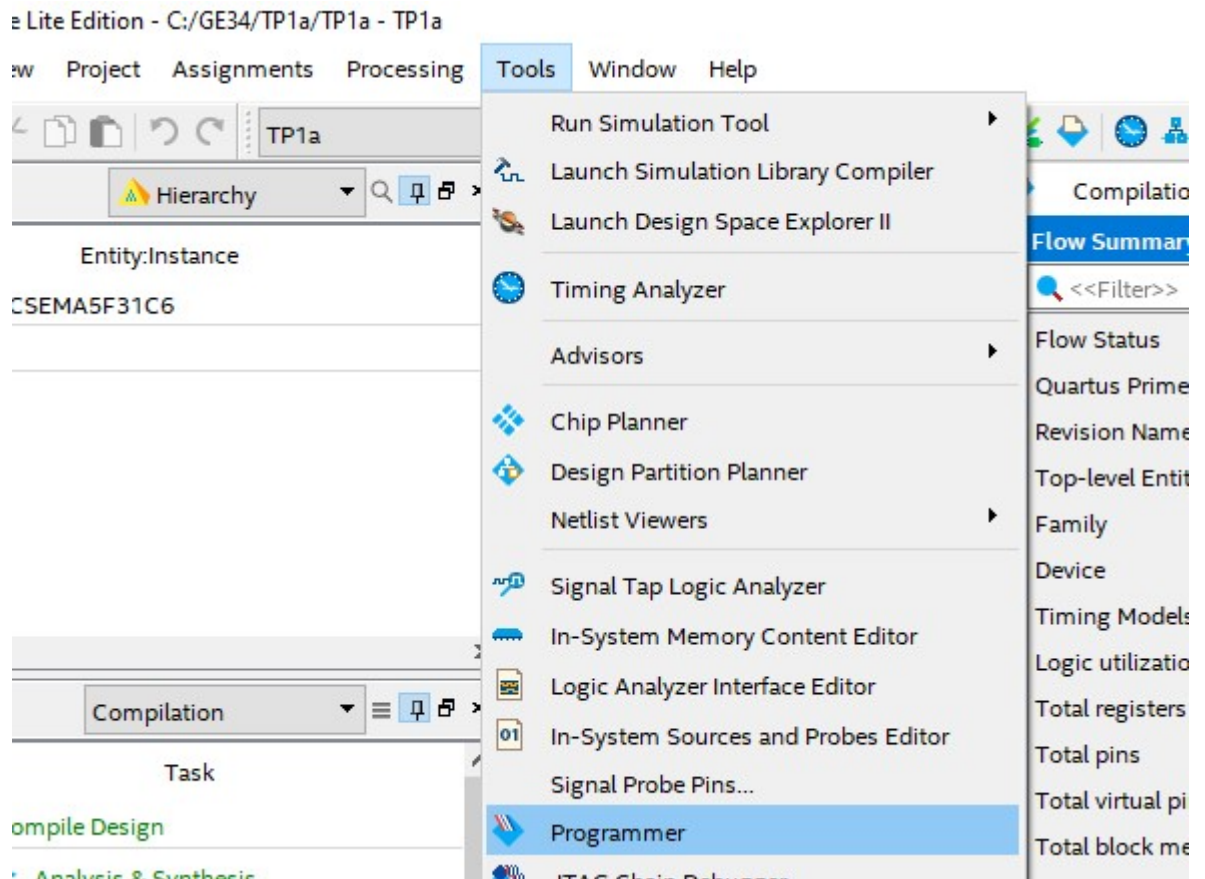


29) La fenêtre suivante s'ouvre. Cliquer OK

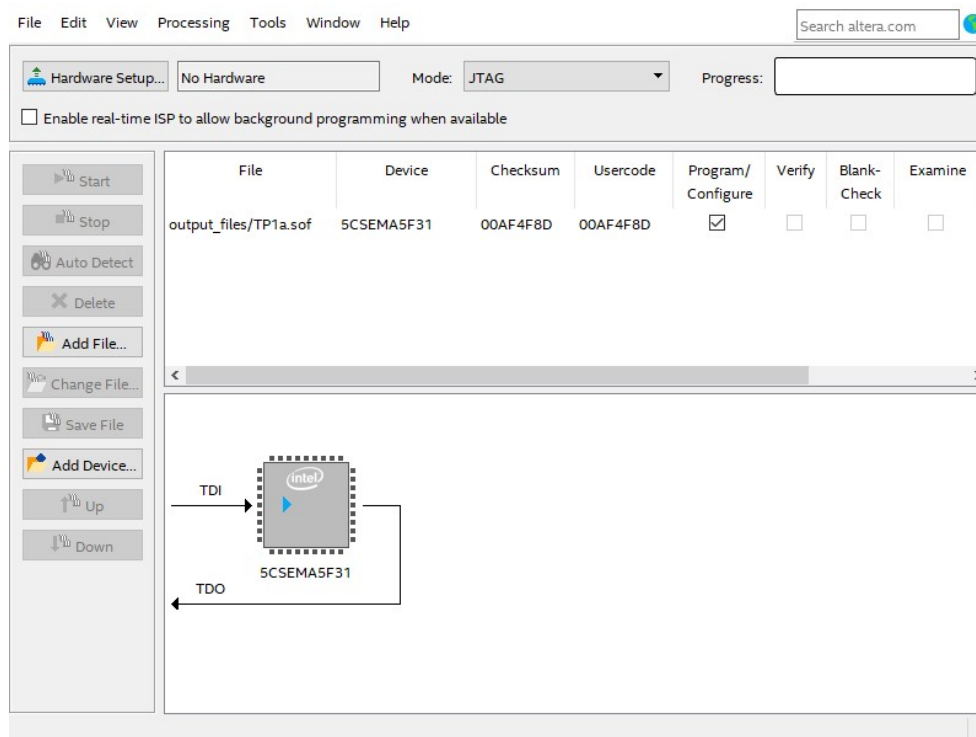


30) Cliquer sur « Start compilation »

31) Cliquer sur Programmer comme indiqué sur la figure suivante

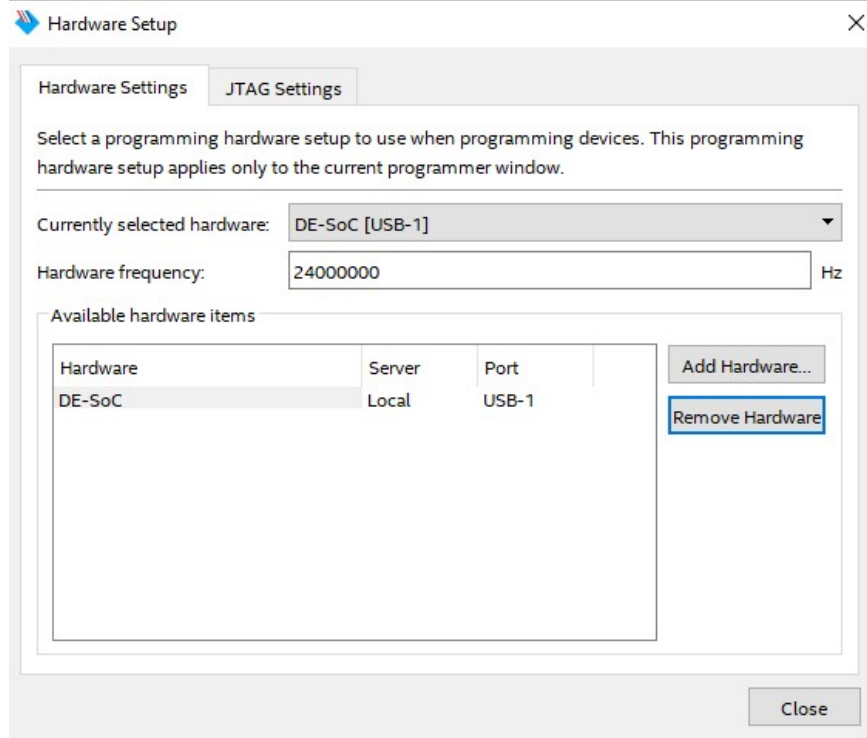
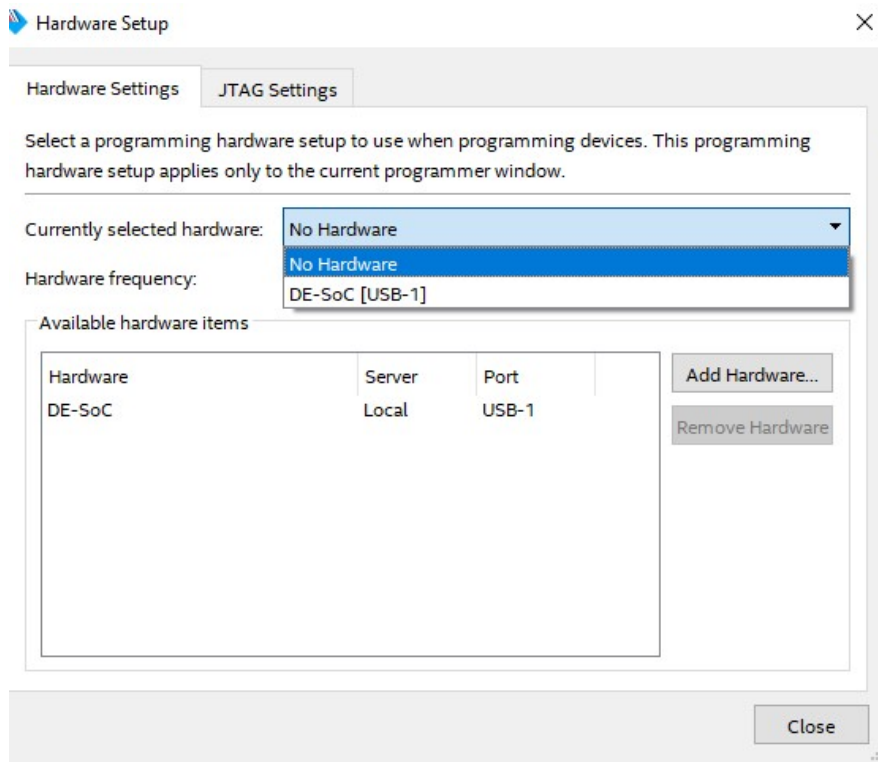


32) La fenêtre suivante s'ouvre



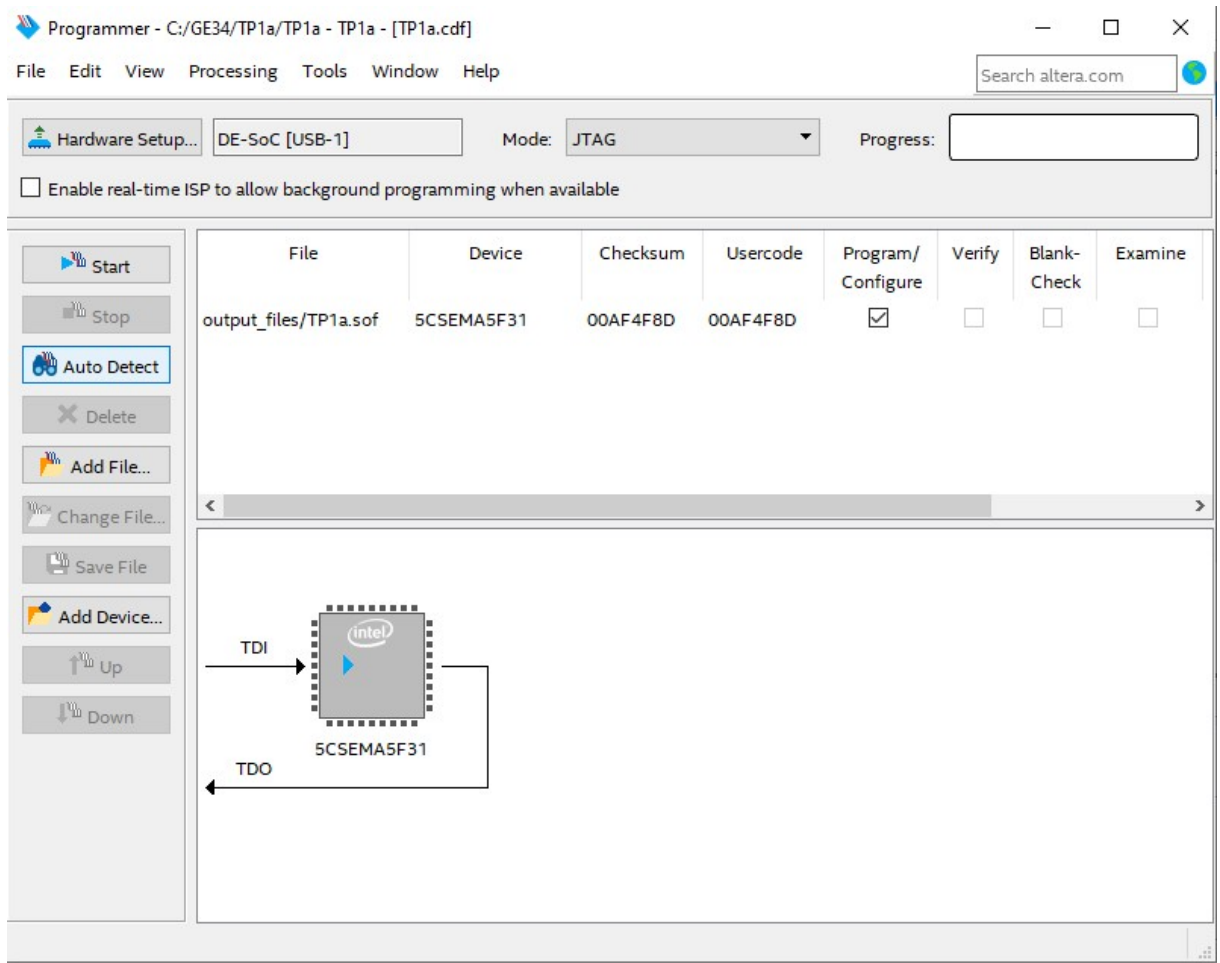


- 33) Si la fenêtre indique « No Hardware », cliquer « Hardware Setup » et choisir DE1 SOC dans le champs Currently Selected Hardware

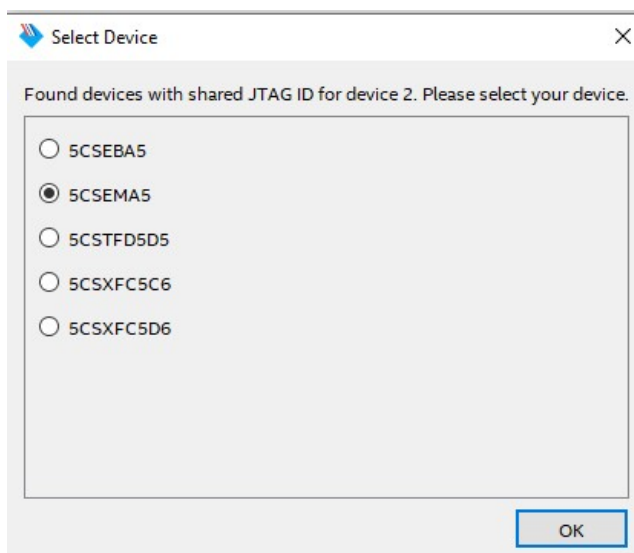


Fermer la fenêtre

- 34) Choisir dans la fenêtre suivante Auto Detect

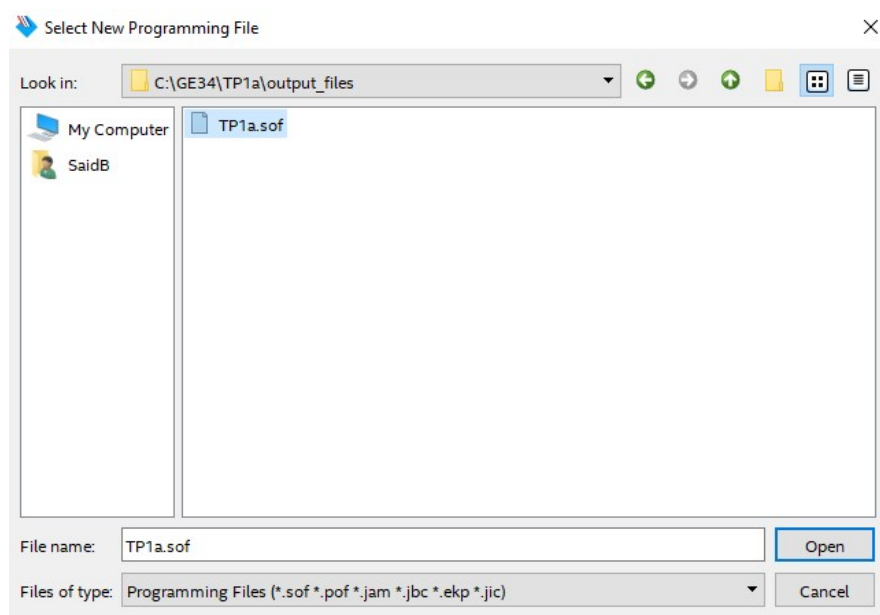
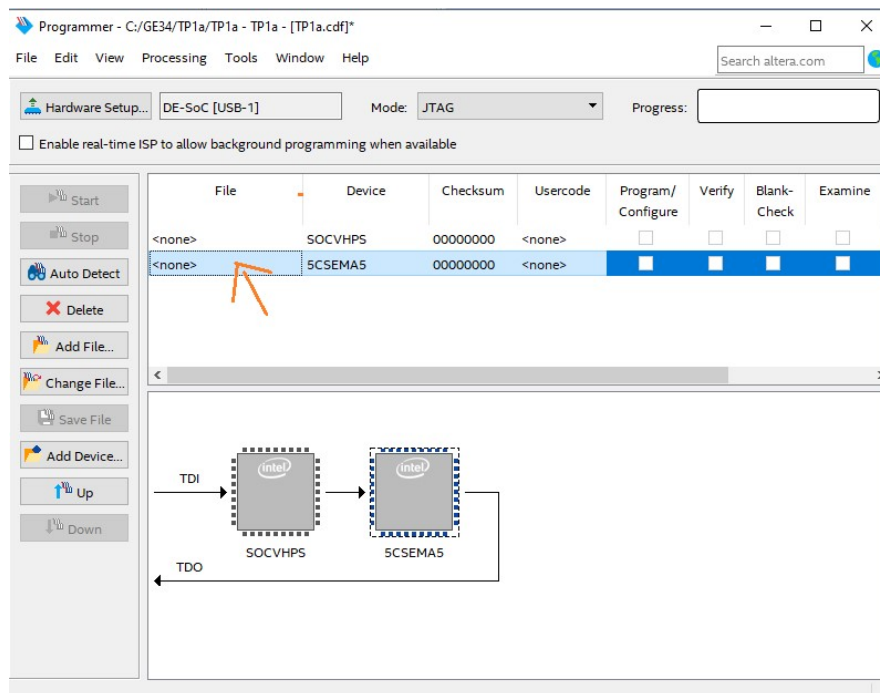


35) Choisir le dispositif indiqué et cliquer OK, puis cliquer Yes sue la prochaine fenêtre

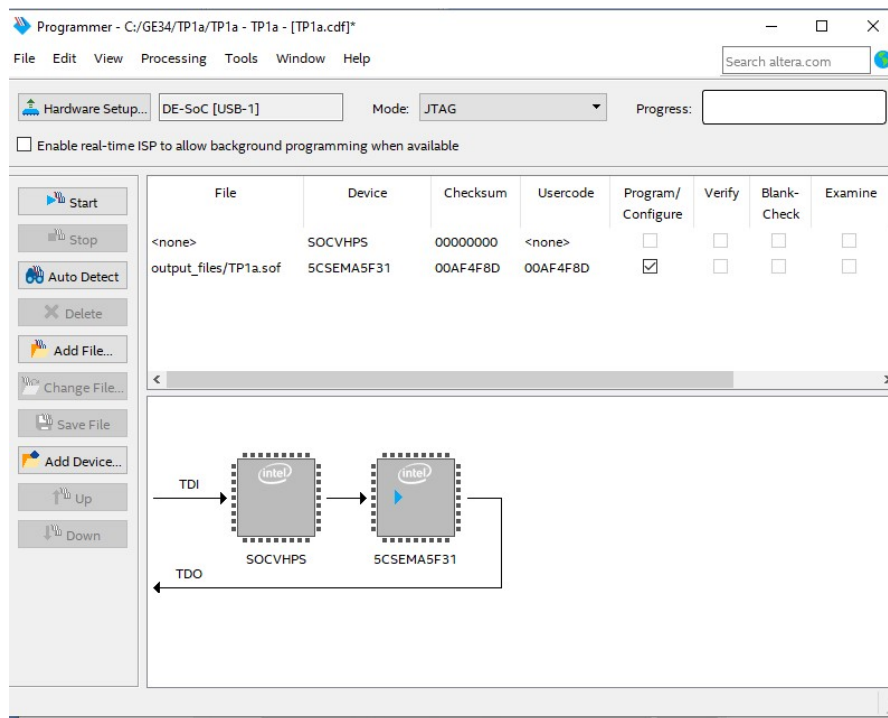


36) Choisir le dispositif indiqué par la flèche rouge et cliquer sur « Add file"





37) Voici la fenêtre obtenue



38) Cliquer Start pour programmer le FPGA et vérifier que le programme fonctionne bien correctement

## Partie b)

### Conception d'un convertisseur de code sur FPGA

Dans cette première partie du TP, l'objectif est de concevoir en VHDL un circuit combinatoire, le compiler, le simuler à l'aide de Quartus II et finalement l'implémenter sur SOC-FPGA. Le bloc combinatoire a 7 entrées et a 7 sorties. Les entrées sont connectées à des switches et les sorties sont connectées à un afficheur sept segments, comme indiquées dans la figure 1 et dans la figure 2. Le circuit contient deux convertisseurs de code et un multiplexeur. Chaque sortie contrôle un segment de l'afficheur. Si le signal est à 0 le segment est allumé, tandis que si le signal est 1 le segment est éteint.

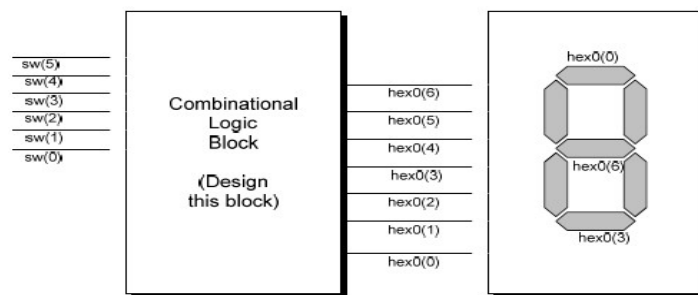


Figure 1

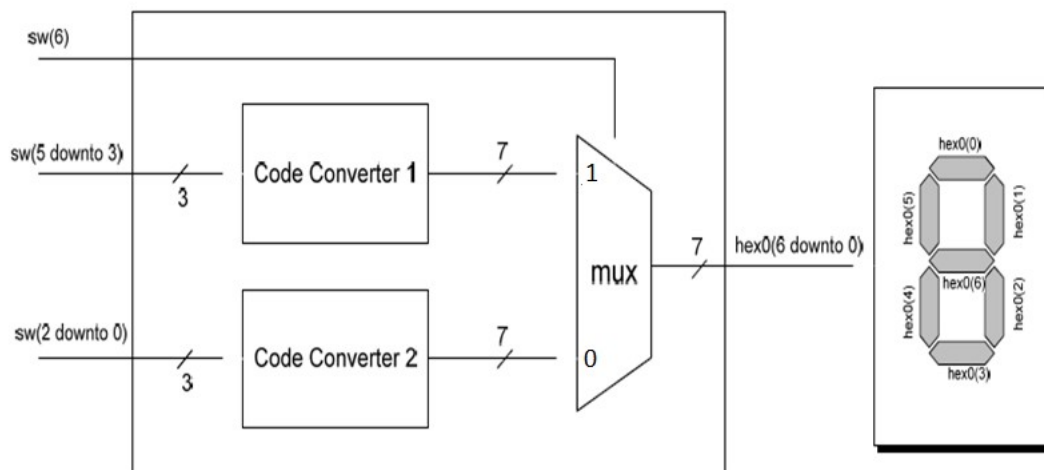












Figure 2

Le premier convertisseur de code convertit les valeurs sur les 3 switches (`sw(5 downto 3)`) en nombre de 0 à 7 si le bit de sélection du multiplexeur est à 1 (`sw(6)`). Le deuxième convertisseur convertit les 3 switches (`sw(2 downto 0)`) en lettre de A à H si le bit de sélection du multiplexeur est à 0. Le tableau suivant illustre les affichages en fonction des valeurs des switches:

sw(2 downto 0)				sw(5 downto 3)			
sw(2 downto 0)	Output	sw(2 downto 0)	Output	sw(5 downto 3)	Output	sw(5 downto 3)	Output
000		100		000		100	
001		101		001		101	
010		110		010		110	
011		111		011		111	

Concevoir ce circuit combinatoire en VHDL, le compiler, le simuler à l'aide de Quartus II et finalement l'implémenter sur la carte DE1 SOC.