

FPGA AVEC CYCLONE II ALTERA : EXERCICES DE PRISE EN MAIN

Objectif : Etre capable :

**De synthétiser un schéma logique de base à l'aide de Quartus et de l'implanter sur la maquette Cyclone 2 DE1
à partir d'un fichier VHDL
à partir d'un schéma**

Matériel :

Platine FPGA Cyclone II d'altera

Cable USB fournit

Driver présent dans : ?/altera/12.1/quartus/driver/usbblaster (? = disque d'installation)

Fascicule de cours VHDL et d'utilisation de quartus : TutorialQuartusWeb

Fichier d'assignement

Rappel important :

**Avant de programmer : Assigner les broches non utilisées en entrée 3 états.
(Assignement + Pin unsed + Input tristate)**

1. EXERCICE 1

Lancer le logiciel Quartus (Web edition version 12)

1.1. Créer un schéma :

Créer un nouveau projet (renseigner les différents éléments : 1/5 la localisation

2/5 rien 3/5 cyclone 2C20F484C7 4/5 Simulation : Mdelsim_ altera - VHDL)

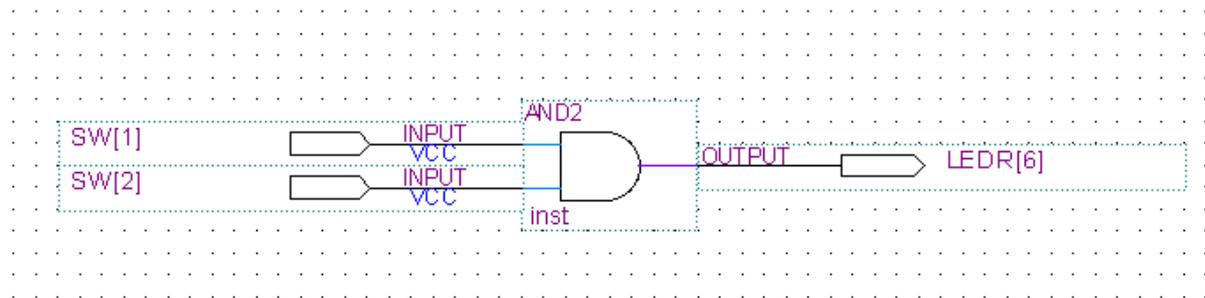
Créer un schéma logique comprenant une unique porte AND :

File + new + Blockdiagram/ schematic file

**Symbol tool + choisir les composants + librairy
(primitives/logic/and2**

Placer les broches d'entrées et de sorties et les nommer (les noms doivent correspondrent à des interrupteurs et leds de la carte ex : SW[0] KEY[1] LEDR[2]...)

Penser à enregistrer vos .dbf (diagram block file)



1.2. Assigner les broches

Assigner les broches non utilisées :

**Assignement + device + device and pins options... + unused pin +
input as tri-stated + ok + ok**

Assigner le fichier correspondant à la carte DE1 (présent sur le site)

**Assignement + import assignement + choisir
«DE1pinassignement.qsf» + ouvrir + ok**

Le numero des broches apparaît alors à coté du nom de la broche.

1.3. Compiler

Processing + start compilation...

voir le déroulement des opératons en bas à gauche
corriger les éventuelles erreurs

1.4. Programmer



Le programmeur se lance :

Vérifier que le USB Blaster est sélectionné dans Hardware setup

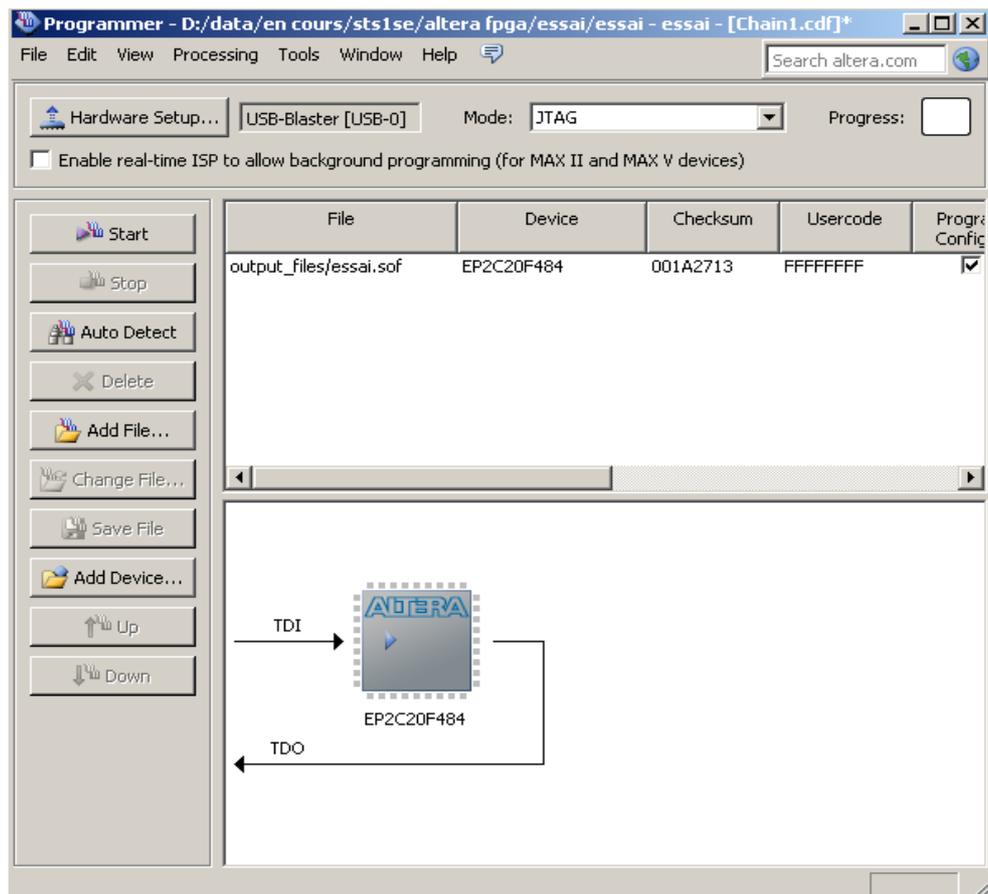
1.5. Programmation en RAM :

Vérifier le mode JTAG (inter sur RUN) : programme mis en RAM : fichier SOF (SRAM)

Remarque : le fichier POF sert pour implanter le programme en ROM (mode AS)

Si le fichier SOF n'est pas présent faire :

Add File + sélectionner le .SOF dans le dossier « output_file »



Cliquer sur :

START

Tester le bon fonctionnement du programme sur la carte.

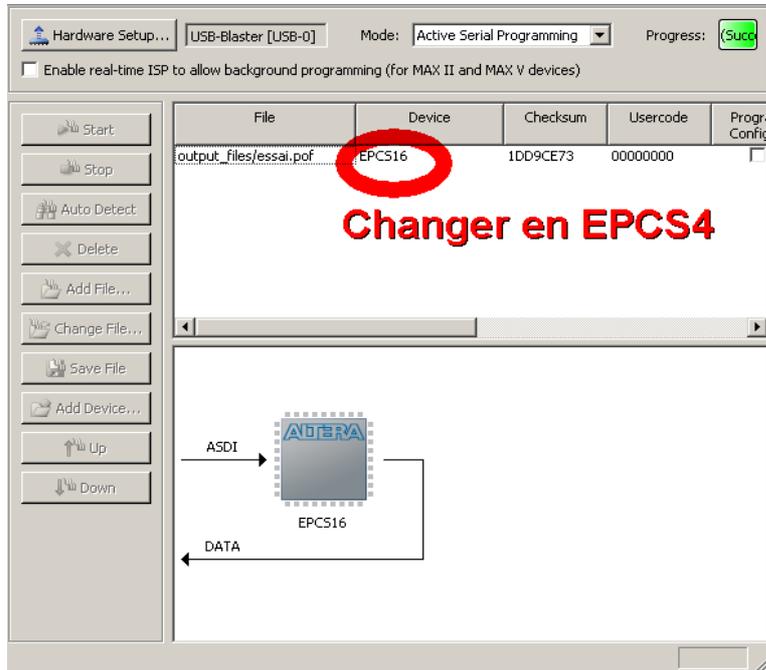
Débrancher la carte : le programme d'efface.

1.6. Programmation en ROM :

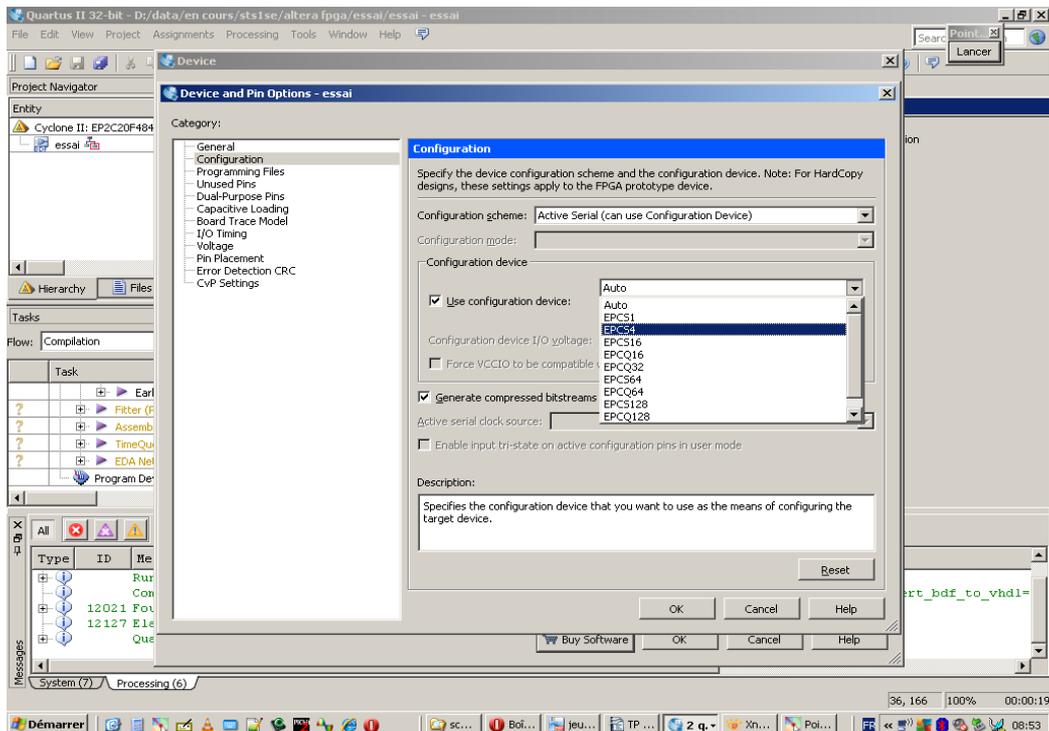
Reprogrammer la carte en mode : ASP (active serial programming) inter sur PROG

Attention il faut que le device corresponde à un :

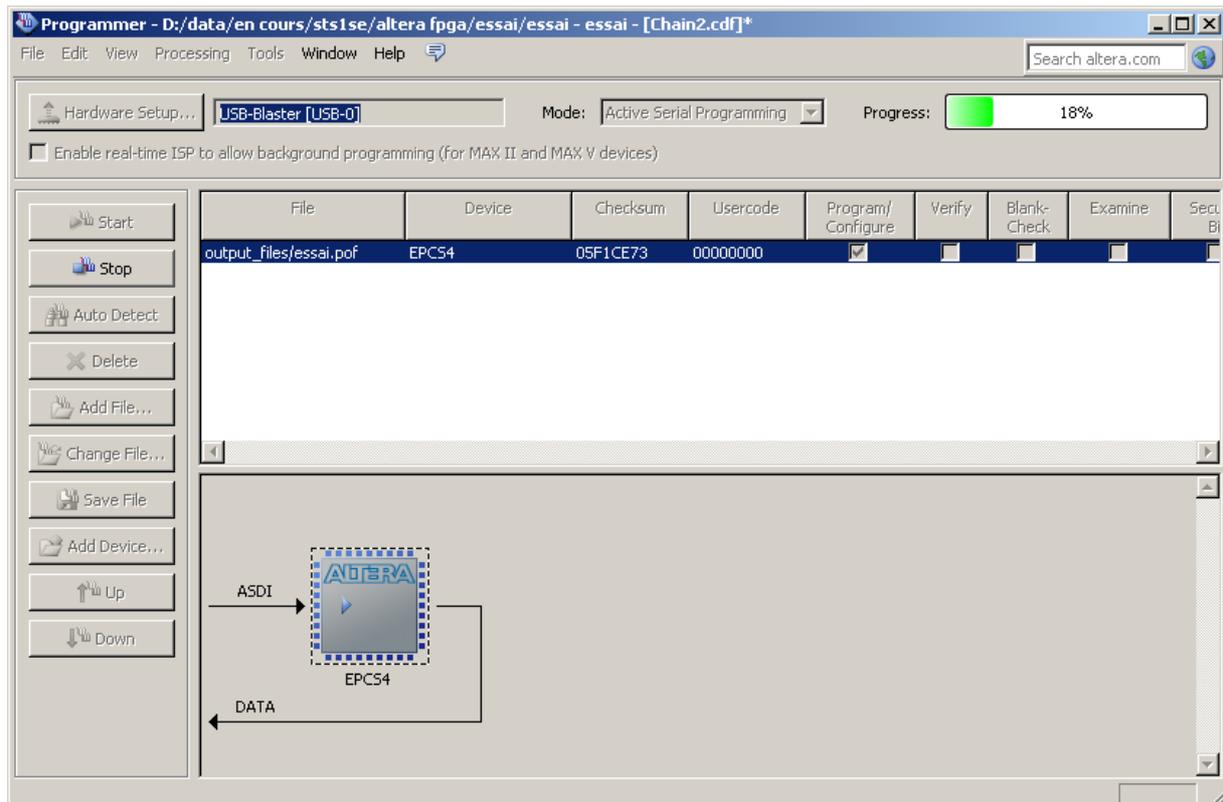
Pour cela : retourner dans Quartus



Assignment + Device + Device and pins options ... + Configuration + choisir EPCS4



Recompiler et programmer en ROM (mode ASP + inter PROG + fichier .POF)



La programmation est plus longue.

Tester le programme : et voilà vous savez programmer un FPGA à partir d'un schéma.

1.7. Complément exercice 1

On va maintenant générer le fichier VHDL à partir du schéma de l'exercice 1.

Cliquer sur l'onglet du schéma .bdf puis

File+ Create/update + Create HDL design... + File type : VHDL + ok

Le fichier .HDL est créé : ouvrir le et identifier la synthèse VHDL.

2. EXERCICE 2 : PROGRAMMER ET TESTER LES EXEMPLES PRÉSENTS DANS LE COURS VHDL :

2.1. Exemple PORTES page 10

La procédure est la même sauf que tout se fait en ligne dans un éditeur de texte.

New project + ... + File + new + VHDL file

Recopier l'exemple entity PORTES : page 10 du cours

```
entity PORTES is
port (A,B :in std_logic;
Y1,Y2,Y3,Y4,Y5,Y6,Y7:out std_logic);
end PORTES;
architecture DESCRIPTION of PORTES is
begin
Y1 <= A and B;
Y2 <= A or B;
Y3 <= A xor B;
Y4 <= not A;
Y5 <= A nand B;
Y6 <= A nor B;
Y7 <= not(A xor B);
end DESCRIPTION;
```

Suivre la procédure donner dans le fascicule d'utilisation de TutorielQuartusWeb paragraphe 2.2.

2.2. Exemple du compteur

```
Library ieee;
Use ieee.std_logic_1164.all;
Use ieee.numeric_std.all;
Use ieee.std_logic_unsigned.all;
entity CMP4BITS is
PORT (
CLOCK : in std_logic;
Q : inout std_logic_vector(3 downto 0));
end CMP4BITS;
architecture DESCRIPTION of CMP4BITS is
begin
process (CLOCK)
begin
if (CLOCK ='1' and CLOCK'event) then
Q <= Q + 1;
end if;
end process;
end DESCRIPTION;
```

3. EXERCICE CRÉATION D'UN COMPTEUR DÉCIMAL DE 0 À 9 EN MODE SCHÉMA.

On vous demande de réaliser un compteur modulo 10 en utilisant les composants présents dans les bibliothèques. Ex : Compteur 4étages + décodeur BCD /7seg ...

la CLOCK sera le bouton : KEY[0]