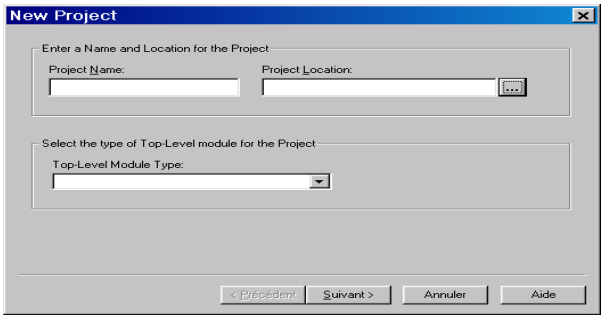
Université M'hamed BOUDIAF de M'SILA

Faculté de Technologie

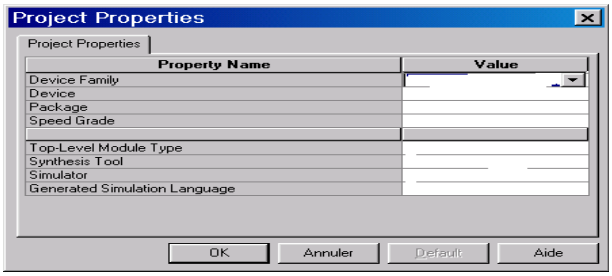
Département Electronique

TP n°6 Electronique Numérique Avancée

TP6 : Programmation d'un circuit FPGA (Basys 2)

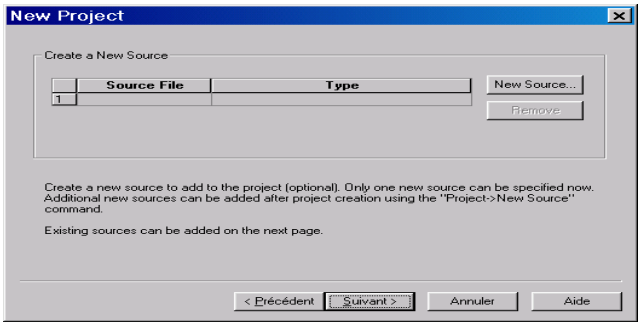


Avant de commencer il est impératif de créer un répertoire de travail D:\ena\sgxy\nom\tp6.
Ouvrir la fenêtre New Project dans le menu File et la compléter en indiquant le nom du projet, le répertoire de travail, et la nature du module du plus haut niveau.

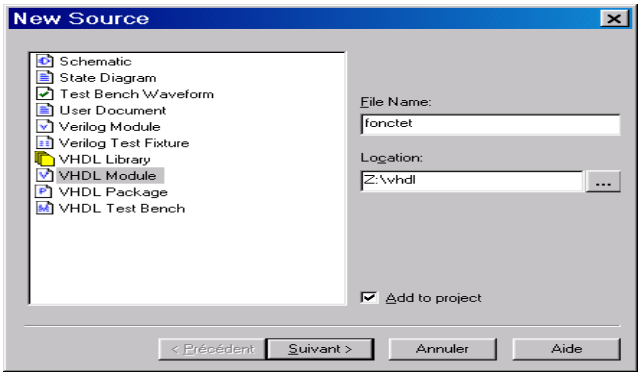


La carte cible Basys utilise un composant FGA xilinx de la famille Spartan3E, Spartan3E-100 en boiter CP132

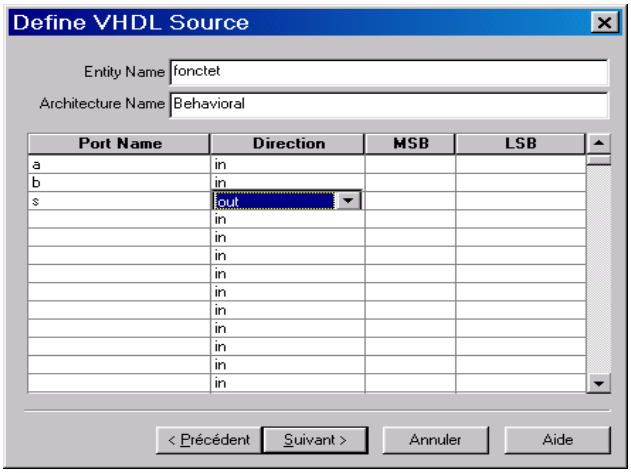
Le langage du module du plus haut Niveau est le HDL, le simulateur associé sera ISE Simulator.



Pour créer un fichier source dans le projet, valider le bouton
New Source par un clic du bouton gauche.



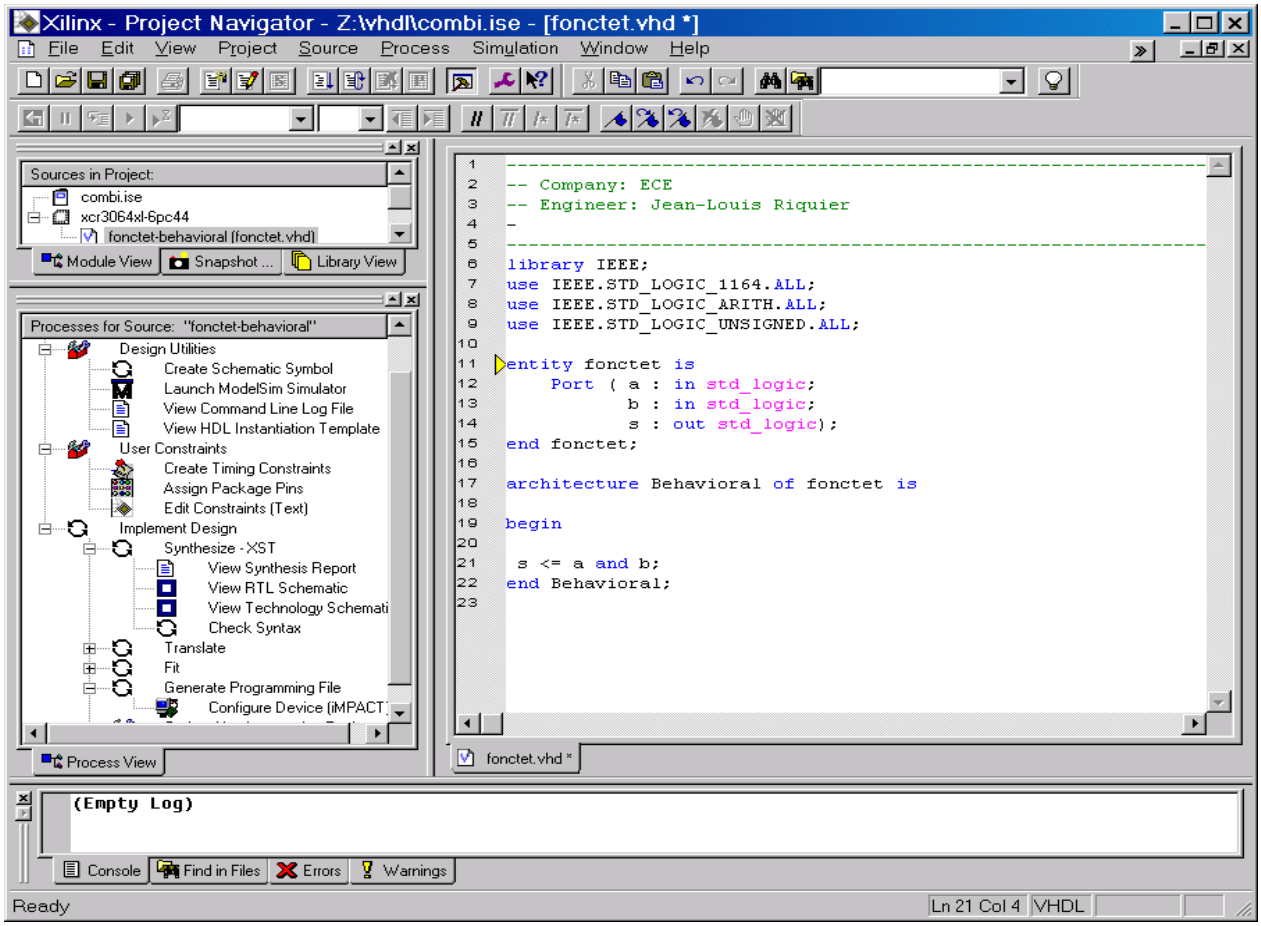
La description des fonctions à réaliser peut se faire sous forme d'un schéma, d'un diagramme d'état ou dans un langage de description comportementale, verilog ou vhdl.
Pour débuter on choisi une entrée en langage VHDL en validant VHDLModule dans la fenêtre ci contre et l'on donne un nom court au fichier.



L'étape suivante permet de définir les entrées et les sorties de la ou des fonctions à réaliser, soit sous forme de bit, soit sous forme de bus dont on donne la taille en complétant la case MSB.
Les broches peuvent être des entrées des sorties ou des entrées sorties lorsque que la sortie doit être réinjectées dans la fonction.
Pour ne pas bloquer au démarrage la structure choisie sera simple deux entrée a,b et une sortie s.

Le logiciel génère automatiquement le squelette du fichier source vhdl.

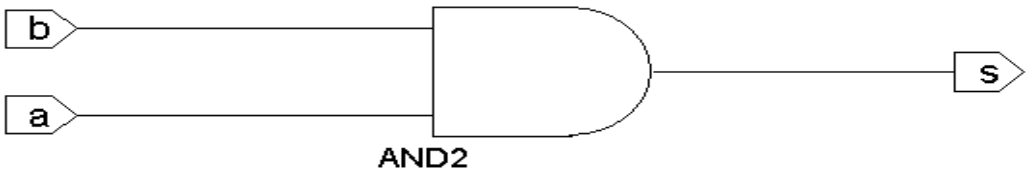
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

---- Uncomment the following library declaration if instantiating
---- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity fonctet is
    Port ( a : in std_logic;
           b : in std_logic;
           s : out std_logic);
end fonctet;

architecture Behavioral of fonctet is
begin

end Behavioral;
```



Pour tester une fonction logique combinatoire il faut compléter l'architecture avec l'équation de la fonction ici s <= a and b ;

Le schéma synthétisé est visible grâce à l'éditeur de schéma que l'on appelle grâce à la commande view RTL Schématic.

Pour visualiser l'architecture il faut effectuer un clic avec le bouton gauche sur l'entité précédente. Et valider la commande push into selected instance.



La dernière étape consiste à programmer le circuit sur la carte cible
Le schéma fonctionnel ci-dessous montre les ressources disponibles sur la carte basys.

La fonction et sera câblée entre les interrupteurs 0 et 1 et la led 0 qui correspondent aux broches P11 L3 et M5 du circuit Spartan3.

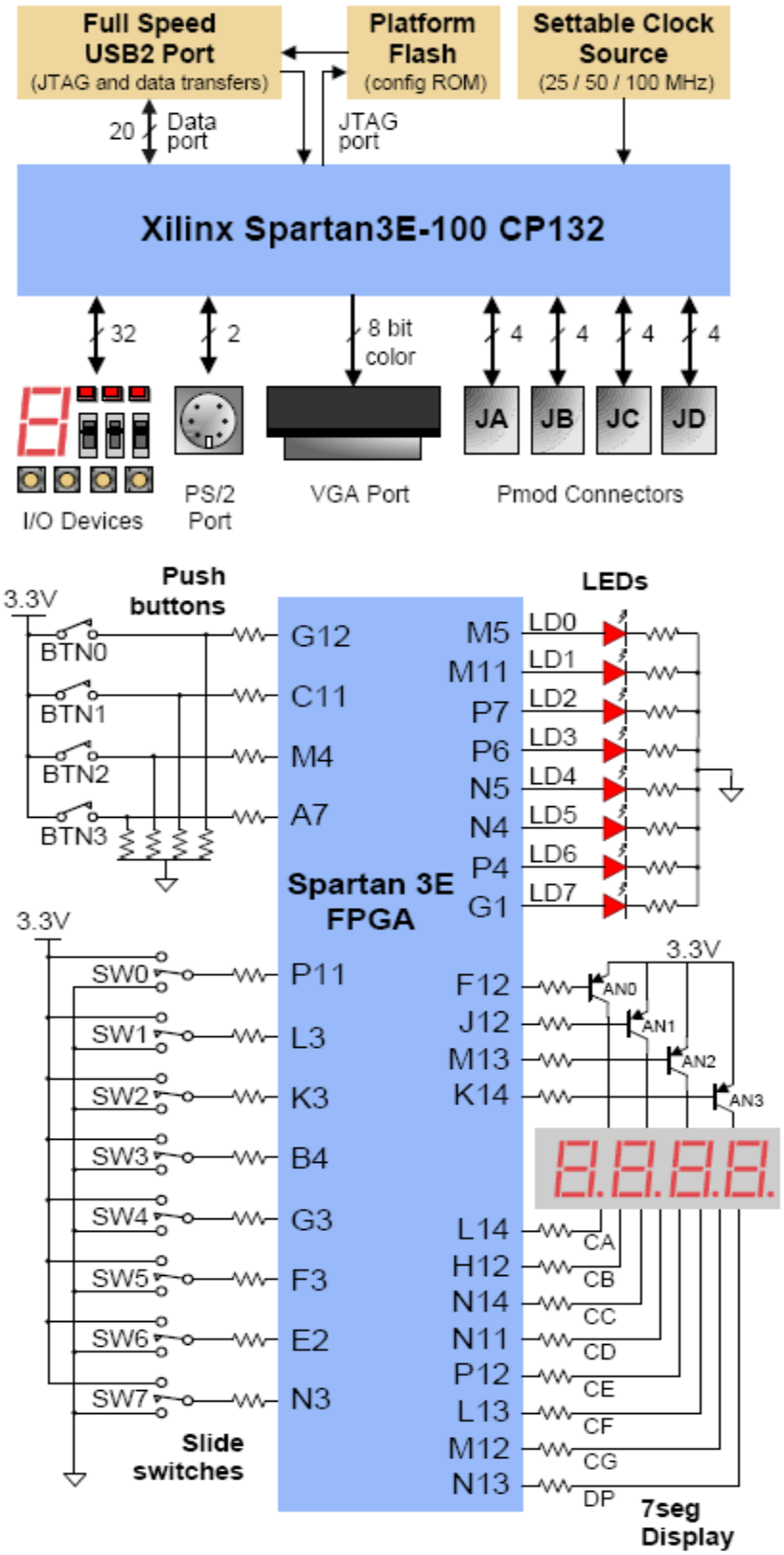
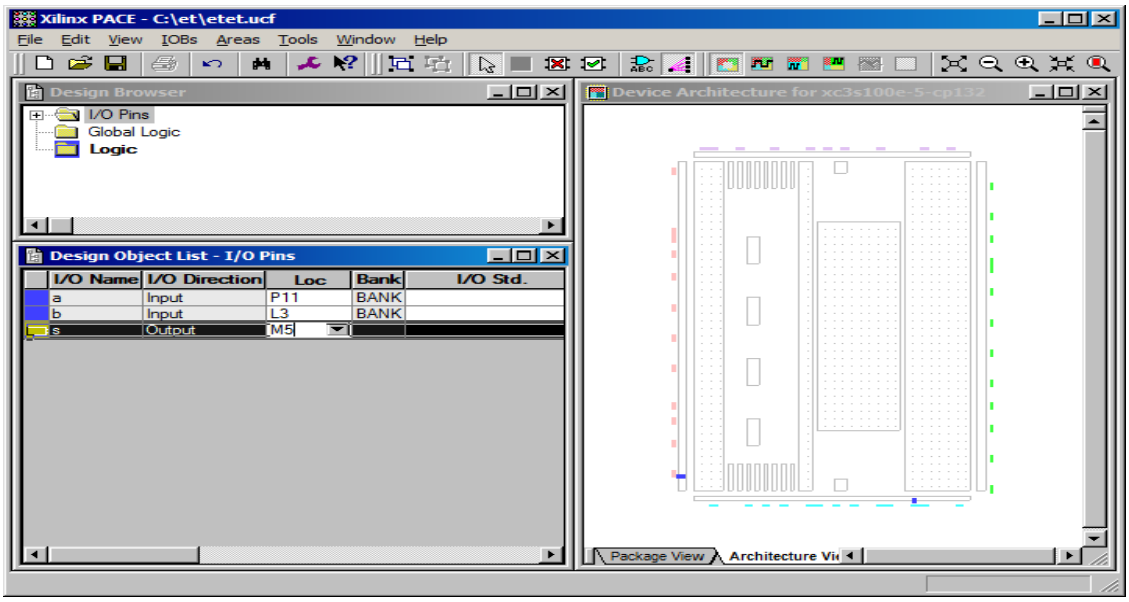


Figure 6. Basys2 I/O circuits



Pour affecter des broches aux entrées sorties de la fonction on utilise la commande
Assign Package Pin de la fenêtre Process. Il suffit de compléter les attributs Loc pour a, b et s : P11, L3, M5 pour une carte basys2.
L'ensemble de ces informations sera sauvegardé dans un fichier portant l'extension .UCF (user constraint file).
Ne pas oublier d'enregistrer ce fichier en quittant l'éditeur de contraintes.

Pour programmer le circuit FPGA il ne reste plus qu'à suivre la procédure suivante.

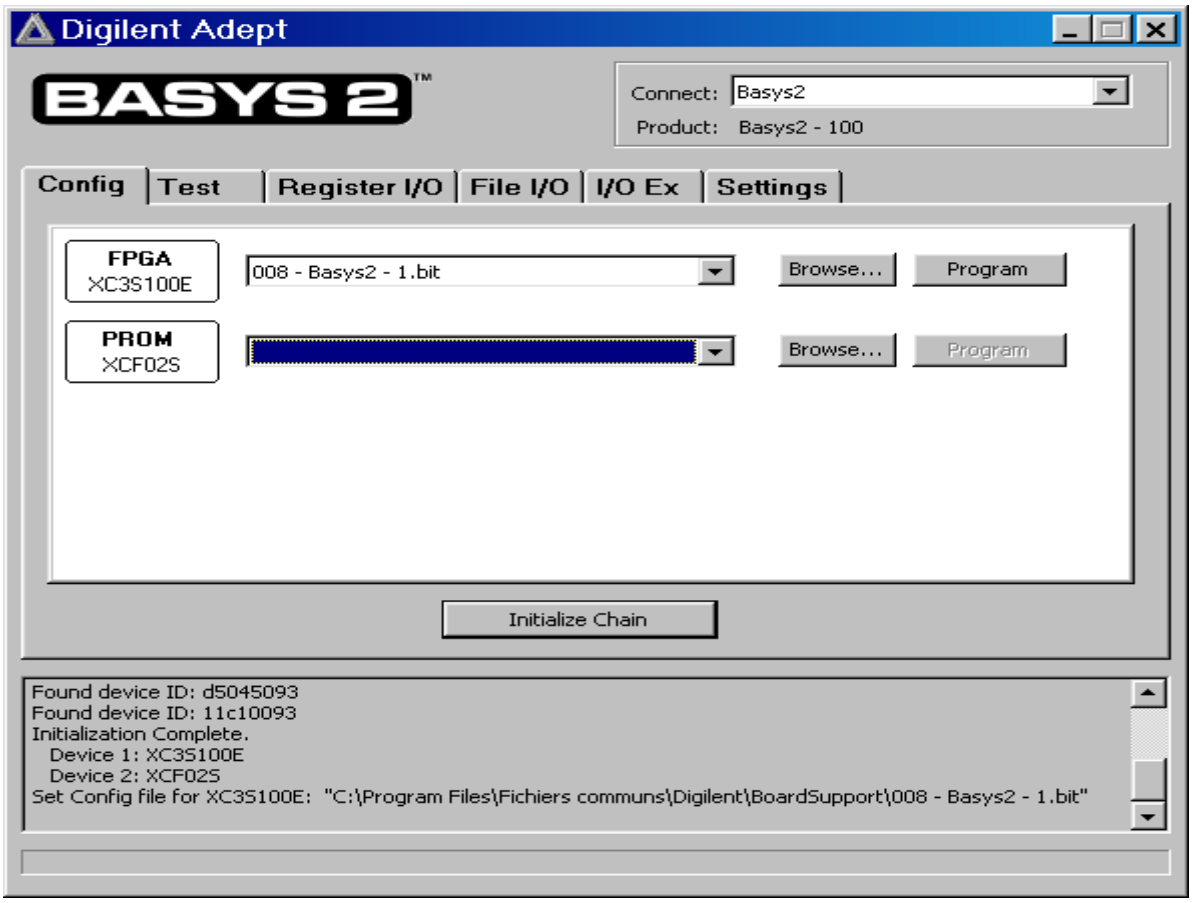
- Faire un clic droit sur « Generate Programming File » puis cliquer sur « Properties »
- Dans l'onglet « Startup Options », dans « FPGA Start-Up Clock » choisir « JTAG Clock »

- Double cliquer sur « Generate Programming File »afin de générer le fichier bit.

Programmation de la puce Spartan de la carte basys :

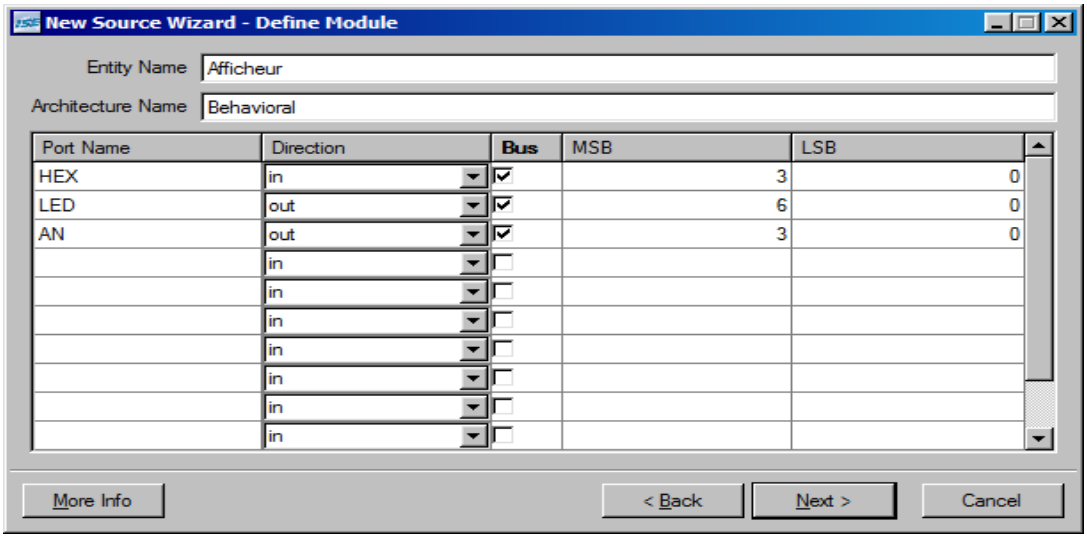
On utilise le programme Adept de digilent pour programmer la carte avec le fichier bit.

- Connecter le cavalier « JTAG » de la carte
- Ouvrir le program Adept
- Brancher la carte en USB
- Cliquer sur « Initialize Chain » si besoin
- Cliquer sur « Browse… » à coté du petit dessin « FPGA »
- Choisir le fichier « *.bit »dans le répertoire de travail
- Cliquer sur « Program »



La dernière étape consiste à manipuler les deux interrupteurs pour vérifier la table de vérité de la fonction réalisée.

Maintenant vous pouvez réaliser la fonction décodeur BCD/7 segments et la tester sur la carte cible.



s

Schéma structurel d'un afficheur anode commune,

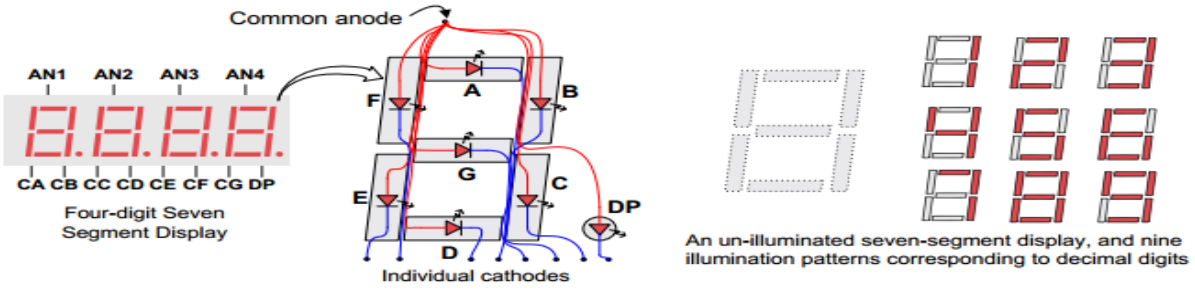
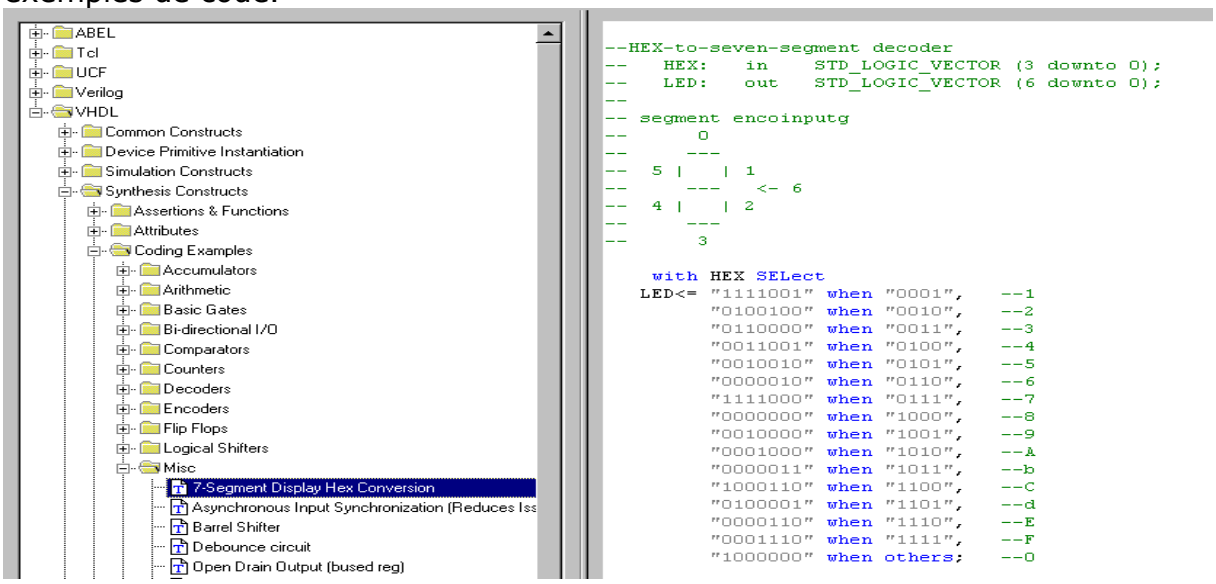


Figure 7. Seven-segment display

la fonction Language templates du menu edit de ISE permet d'accéder à des exemples de code.



Il ne reste plus qu'à comprendre ce code définir les broches d'entrée et de sortie, Valider un ou deux afficheurs (AN <= ''1110'' ;) puis tester ce code sur la carte.