Samuel MARTINEAU
Johan MEDIONI

26 septembre 2011

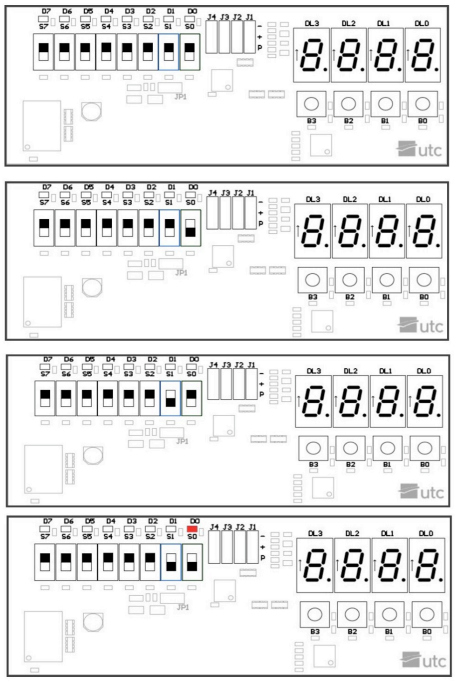
Compte-rendu TP01 MI01

Exercice 1

Il est demandé dans cet exercice de prise en main du logiciel ISE Design Suite de programmer un composant de type FPGA en lui affectant une fonction ET à deux entrées. Voici le code VHDL :

`entity exo1 is`	**Entité**
`PORT (SW_0, SW_1 : IN BIT ;`	Déclaration des entrées
`LED_0 : OUT BIT) ;`	Déclaration de la sortie
`end exo1;`	
`architecture Behavioral of exo1 is`	**Architecture**
`begin`	
`LED_0 <= SW_0 AND SW_1 ;`	On utilise directement le AND défini dans le VHDL.
`end Behavioral;`	

Fonctionnement après programmation de la carte FGPA : en changeant les valeurs des entrées SW_0 et SW_1, la diode correspondant à LED_0 s'allume bien quand la sortie du ET est 1. La figure suivante est une représentation schématique du comportement de la carte ainsi programmée (jointe en annexe et2.jpg) :



Compte-rendu TP01 MI01

Remarques diverses sur l'utilisation de ISE Design Suite :

- Simulation : après avoir écrit le code, on peut le tester en syntaxe et en fonctionnement. Cela évite de générer le fichier à charger sur la carte FGPA si le code n'a pas le fonctionnement attendu. En revanche, ce fichier n'est dans tous les cas pas généré si la syntaxe est incorrecte. (De même la simulation ne marchera pas.)

- Charger le fichier : cela se fait au moyen de l'utilitaire TeraForm. Il permet de charger sur la carte FGPA le code VHDL. C'est pourquoi le nom des signaux du code doit correspondre aux noms des signaux de la carte.

Compte-rendu TP01 MI01

Exercice 2

Cet exercice demandait de modéliser un additionneur 2 bits.

Résultats attendus (a et b entrées, s en sortie) : s prend la valeur de la somme a+b. Si a+b est strictement supérieur à 3, il y a débordement.

Le choix a été fait d'utiliser un VHDL comportemental, en travaillant directement sur la valeur. Voici le code :

`entity exo2 is`
 	
`PORT( SW_10, SW_32 : IN INTEGER RANGE 0 TO 3;`
 	
`LED_10 : OUT INTEGER RANGE 0 TO 3);`
 	
`end exo2;`
	Entité
	
	SW_10, SW_32 et LED_10 sont des vecteurs de bits. On les déclare en tant qu'entiers pour utiliser directement les fonctions arithmétiques du VHDL.

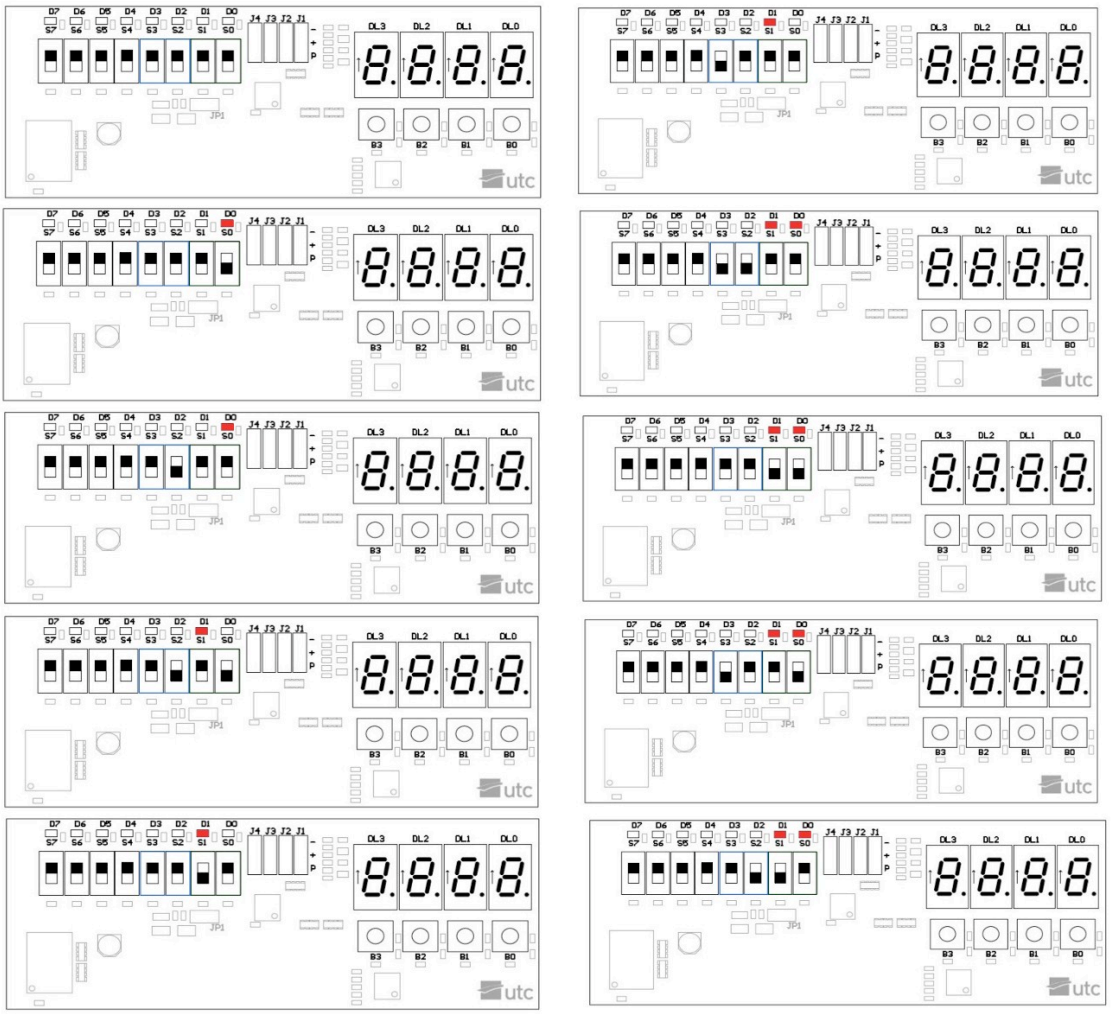
`architecture Behavioral of exo2 is`
 	
`begin`
 	
`LED_10 <= SW_10 + SW_32;`
 	
`end Behavioral;`
	Architecture
	
	Utilisation de la fonction +.

Fonctionnement après programmation de la carte FGPA : On retrouve bien les résultats attendus. Par exemple, SW_10=01 et SW_32=10 faisait s'allumer les deux diodes de LED_10, et SW_10=11 et SW_32=01 n'allumait aucune diode (débordement, LED_10=$(4)_{10}$=100, le 1 du 3° bit tombe).
En page suivante une figure représente le fonctionnement de la carte ainsi programmée (jointe en annexe add2.jpg). Seuls les cas où la sortie ne déborde pas ont été représentés.

Samuel MARTINEAU
Johan MEDIONI

26 septembre 2011

Compte-rendu TP01 MI01



Exercice 3

On peut considérer cet exercice comme un prolongement du précédant : il faut ici que sélectionner l'opération que l'on veut effectuer parmi l'addition, la multiplication et la soustraction. On se sert pour cela des bits SW_4 et SW_5. Choix fait :

SW_4	SW_5	Opération
0	0	Addition
1	0	Soustraction
0	1	Multiplication

Voici le code VHDL correspondant :

```
entity exo3 is

PORT( SW_10, SW_32 : IN INTEGER RANGE 0 TO 3;

      SW_5, SW_4 : IN BIT;

      LED_10 : OUT INTEGER RANGE 0 TO 3);

end exo3;



architecture Behavioral of exo3 is

begin

LED_10 <=

           SW_10 + SW_32   WHEN ((SW_5 = '0')
AND (SW_4 = '0')) ELSE

           SW_10 - SW_32   WHEN ((SW_5 = '0')
AND (SW_4 = '1')) ELSE

           SW_10 * SW_32   WHEN ((SW_5 = '1')
AND (SW_4 = '0'));

end Behavioral;
```

Entité

Architecture

Affectation conditionnelle.
Condition 1 : on fait l'addition.

Condition 2 : on fait la soustraction.

Condition 3 : on fait la multiplication.

Fonctionnement après programmation de la carte FGPA : La génération du fichier à charger sur la carte FPGA n'a pas fonctionné. Une ***erreur de License*** est apparue. Néanmoins, lors de la simulation, tout fonctionnait correctement. Il n'a juste pas été possible de le vérifier sur la carte FPGA.