Labo 1 sur Quartusll

A. Pour commencer

- Décompressez le fichier ~pift6223/quartus2/labo1/labo1.zip dans votre zone.
- Ouvrez l'application Altera > QuartusII
- Ouvrez le projet labo1

B. Dessiner un circuit logique combinatoire et le compiler

- Ouvrez le fichier « intro.bdf »
- Cliquez sur l'endroit où il faut insérer le composant avec le bouton droit de la souris (insert/symbol).
- Dans le menu C:\altera\quartus41\librairies\primitive\logic, choisissez la fonction logique souhaitée.
- Connectez les entrées aux boutons « PUSH » et les sorties aux boutons « MSD… » et « LSD… »
- Sauvegardez
- Compiler dans « Processing/Start Compilation ».

C. Simuler le circuit logique

- Créez un nouveau fichier de simulation (File > New > Other files > Vector WaveForm File)
- Cliquez avec le bouton droit de la souris sur la partie gauche de la fenêtre.
- Insert Node or Bus > Node Finder > List > >> > OK
- Initialisez les inputs avec les boutons sur la gauche.
- Simulez : Processing >Start Simulation

D. Exercices

Quelques circuits logiques à réaliser exclusivement avec des portes logiques élémentaires (AND, NAND, OR, NOR, XOR, NXOR, NOT):

- 1. Un convertisseur de binaire (Zéro ou Un) vers 7 segments.
- 2. Un additionneur binaire (trois entrées, deux sorties) avec affichage sur le convertisseur réalisé au point 1.
- 3. Une fonction pour résoudre le problème suivant :

Trois processeurs (A,B,C) veulent avoir accès à une mémoire. A est prioritaire sur B et B est prioritaire sur C. Réaliser un circuit avec trois entrées (A,B,C) et deux sorties (R1, R0) pour coder l'entrée qui peut accéder à la mémoire. Utilisez le codage suivant : 00 Pas d'accès; 01 A a l'accès; 10 B a l'accès; 11 C a l'accès

Écrivez la table de vérité de R1 et R0, Déduisez-en le circuit le plus simple possible. Implémentez le circuit sur QuartusII. Utilisez le convertisseur réalisé au point 1 pour l'affichage. On souhaite ensuite que si l'accès est accordé à A et que C fait une demande, ce soit lui qui devienne prioritaire...

4. Utilisez le bloc « vga_sync ». Chaque bit est représenté par une moitié (gauche ou droite) de l'écran. Utilisez la couleur rouge pour représenter un '1' et la couleur verte pour représenter un '0'. Utilisez ce bloc pour tester la fonction réalisée au point 3.

Commencez par simuler votre architecture avant de la programmer dans le FPGA.

Bon Travail !