M1 ISME *Travaux pratiques VHDL*

Outil logiciel : Quartus II version 9.0 sp2 web edition d'Altera
Outil matériel : carte DE1 de Terasic

Documents utiles au TP :

http://jeanlouis.boizard.free.fr/m1_isme/master_1/Affectation des broches carte de1.pdf

http://jeanlouis.boizard.free.fr/m1_isme/master_1/DE1_UserManual_v1017.pdf

http://jeanlouis.boizard.free.fr/m1_isme/master_1/manuel_quartus2.pdf

D'autres exercices peuvent être trouvés ici :

http://jeanlouis.boizard.free.fr/m1_isme/master_1/td2_vhdl.txt

Travaux pratiques proposés

TP1 : Additionneurs 2x1, 2x3, 2x5 et 2xn
TP2 : Mini Projet (Diviseur, Compteur, Transcodeur BCD/7 segments)
TP3 : Générateur de parité, registre à décalage 8 bits

Note importante :

L'environnement de développement Quartus II est dédié à la conception de systèmes numériques complexes. Cela signifie qu'une très grande rigueur doit être observée dans la création et la gestion des projets.
Pour cela il est vivement conseillé :

- De créer un dossier général par binôme
- De créer un projet pour chaque exercice proposé

!!! Attention : éviter l'emploi de noms de dossiers ou fichiers à rallonge et comportant des accents, -, /, \ et blancs.

Boizard Jean-Louis/ Peytavi Didier

TP1 - Additionneurs :

Réaliser un additionneur de deux mots de 1 bit :

- entrées : 2 bits a_i et b_i et 1 retenue d'entrée C_{in}.
- sorties : 1 bit s_i et 1 retenue de sortie C_{out}.

a) En graphique à l'aide de portes
b) En langage VHDL structurel (utilisation de portes)

Pour cela créer un nouveau projet sous Quartus II appelé add_1. Ouvrir une fenêtre graphique et l'enregistrer dans le dossier avec le nom add_1.bdf. Saisir le schéma associé à l'additionneur et créer les entrées sorties nécessaires. Enregistrer le document une fois la saisie de schéma terminée.
Réaliser la compilation et la simulation et si le fonctionnement est correct, visualiser les synthèses logique et physique générées par Quartus. Enfin créer le symbole associé à add_1.
Réaliser l'implémentation sur le composant et valider sur la carte DE1, on choisira pour a_i , b_i et C_{in} les switchs SW0, SW1 et SW2 et pour s_i et C_{out} les leds LEDR0 et LEDR1.

Réaliser un additionneur de deux mots de 3 bits :

Entrées : 2 mots de 3 bits A2 A1 A0 et B2 B1 B0 et une retenue d'entrée Cin.
Sorties : 1 mot de 3 bits S2 S1 S0 et une retenue de sortie Cout.

Ici on crée l'additionneur en exploitant le circuit add_1 précédemment créé.

Créer un nouveau projet sous Quartus II appelé add_3. Lors de la création du projet ajouter les fichiers issus d'autres projets qui seront utilisés (ici le fichier symbole add_1). Ouvrir une nouvelle fenêtre graphique et saisir le schéma de l'additionneur add_3 en exploitant le symbole graphique add_1.
Réaliser la compilation et la simulation. Si OK valider sur la carte DE1.On choisira pour A0, A1, A2, B0, B1, B2 et Cin (SW0, SW1, SW2, SW3, SW4, SW5 et SW6) et pour S0, S1, S2 et Cout (LEDR0, LEDR1, LEDR2 et LEDR3).

Réaliser un additionneur de deux mots de 5 bits :

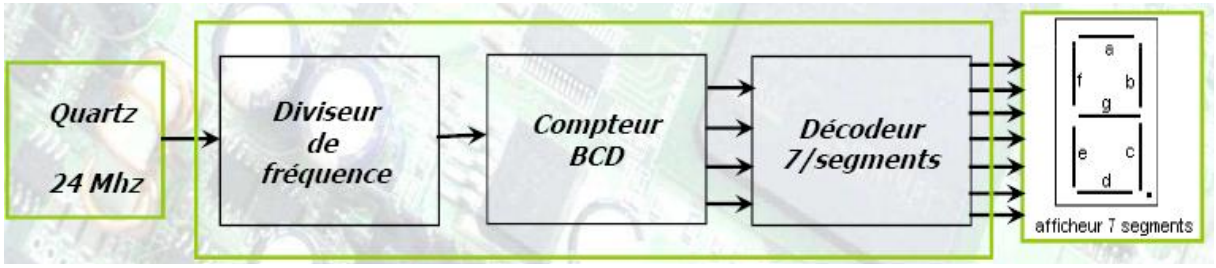
Reprendre les mêmes procédures que décrites précédemment. Exploiter les symboles des composants add_1 et add_3.

Réaliser un additionneur de deux mots de n bits (n générique):

On se propose de décrire ici en langage VHDL comportemental un additionneur de 2 mots de n bits, n étant passé en paramètre.

TP 2 - Mini-Projet

Projet : On veut réaliser un projet permettant, à partir d'un oscillateur à quartz de 24Mhz présent sur la carte DE1, de visualiser l'écoulement des secondes sur un afficheur 7 segments. On pourra utiliser la même méthode que précédemment, à savoir créer un projet par fonction à réaliser.

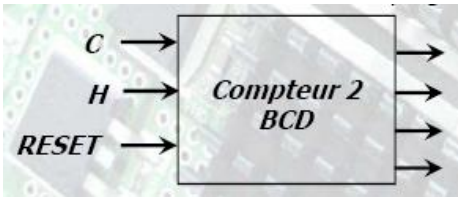


Fonction **Génération du signal 1 Hz :**
Dans une première étape générer un signal de fréquence 1 Hz à partir de l'horloge 24 Mhz. Pour cela réaliser un diviseur de fréquence. La sortie de celui-ci sera connectée à une led de la carte DE1.

Réaliser en VHDL par une description comportementale le programme du diviseur. Tester le fonctionnement et créer le symbole associé.

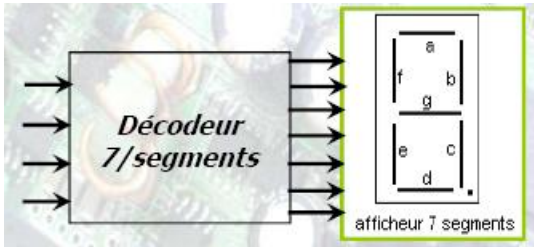
Fonction **comptage BCD :**
Réaliser un compteur BCD (comptage de 0 à 9) possédant 1 entrée d'horloge H et 4 sorties (Q0, Q1, Q2, Q3). On proposera une description comportementale.

Même chose avec un compteur/décompteur et reset asynchrone:



Valider sur la maquette.

Fonction **décodage BCD / 7segments :**



- 4 entrées (on utilisera 4 interrupteurs de la carte)
- 7 sorties (on utilisera un des deux afficheurs 7 segments)

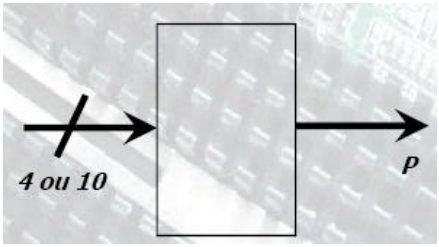
Réaliser le programme VHDL. Toute combinaison autre que celles allant de 0 à 9 entrainera un affichage d'un E signalant une erreur.

Création du projet global

Créer un projet qui reprend l'ensemble des fonctions réalisées ci-dessus. Créer le schéma à partir des symboles générés précédemment. Compiler puis tester sur la maquette.

TP3 – Générateur de parité, registre à décalage

Générateur de parité : La sortie P donne une information sur la parité du vecteur d'entrée E. On prendra P=1 si le nombre de 1 est impair.
A partir d'une boucle FOR LOOP, donner une description VHDL de ce générateur de parité (on prendra E sur 4 bits puis sur 10 bits). Enfin on utilisera la forme générique.



Compilation et simulation de ce générateur. Intégration et validation sur le circuit FPGA.

Registre à décalage :

Une entrée d'horloge clk , une entrée data_in et une entrée de raz asynchrone.
Sorties Q sur n bits (n étant un paramètre générique)

Fonctionnement :

- Raz=1 met toutes les sorties à 0
- Si raz=0 et qu'un front montant est sur clk, alors la sortie Q0 prend la valeur de data_in, Q1 prend la valeur de Q0, etc …
-