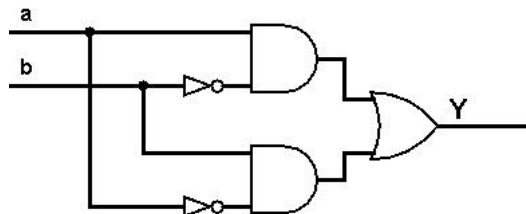


TD 02 : Langage VHDL Instruction concurrente

Exercice 1 :

Soit la fonction logique Y réalisée par le logigramme suivant :



1. Ecrire l'équation de cette fonction logique.
2. Implémenté le circuit logique avec VHDL en utilisant une architecture structurel et une architecture flot de données.

Exercice 2 :

1. Donner la table de vérité du décodeur 2-4

Ecrire son code VHDL en utilisant :

1. L'affectation concurrentielle (utilisation de la fonction canonique)
2. L'affectation conditionnelle (utilisation de la table de vérité, avec des vecteurs)
3. L'affectation sélective (en déclarant les entrées comme des entiers)

Exercice 3 :

Ecrire le code VHDL d'un décodeur Mux 4 vers 1 en utilisant :

1. l'affectation sélective.
2. L'affectation concurrente.
3. L'affectation conditionnelle.
4. L'architecture structurelle avec un composant Mux 2 vers 1.

Exercice 4 :

Ecrire le code VHDL d'un décodeur BCD-7 segment en utilisant l'affectation case.

Exercice 5 :

Ecrire un code VHDL du transcodeur en utilisant :

1. L'architecture flot de données.
2. L'architecture comportementale avec l'affectation if/then/elseif.

A	B	C	D	a	b	c	d
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	1	0	1	1
0	1	1	0	1	1	0	0
0	1	1	1	1	1	0	1
1	0	0	0	1	1	1	0
1	0	0	1	1	1	1	1



TD 03 : Langage VHDL Instruction concurrente

Exercice 1 :

Soit un décodeur d'adresse mémoire 4 bits. On suppose que la mémoire considérée est décomposée en quatre pages, la deuxième page étant elle-même décomposée en quatre segments.

Table des adresses de la mémoire.

12-15	Page 4
8-11	Page 3
7	Page 2
6	
5	
4	
0-3	Page 1

Ecrire le code VHDL du décodeur

Exercice 2 :

L'unité arithmétique et logique (Arithmetic and Logic Unit - ALU) constitue le cœur d'une unité de calcul telle que l'on trouve dans un microprocesseur. Elle est capable d'effectuer des opérations arithmétiques (addition, soustraction, incrémentation, décrémentation, décalages) et logiques (AND, OR, NOT, XOR) de base sur deux bus de données.

Ecrire le code VHDL de cette ALU pour des arguments non signes

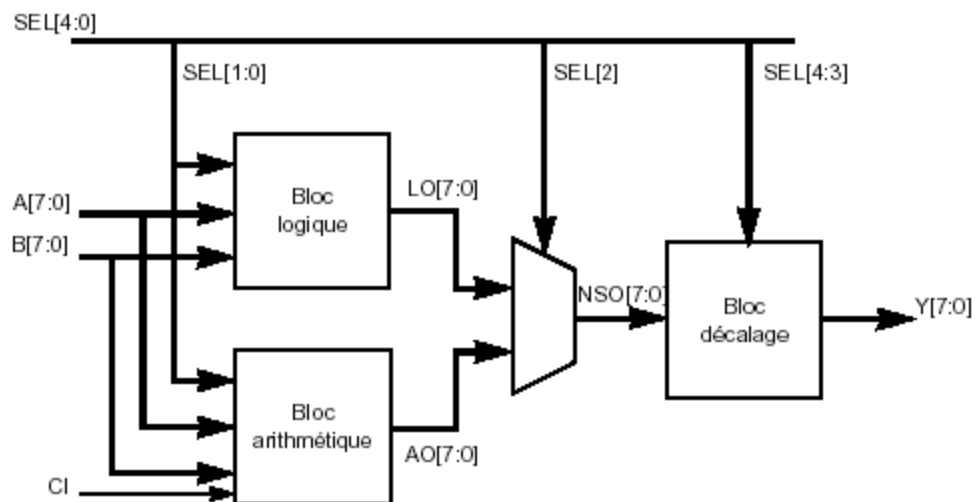
Exercice 3 :

Soit une ALU de 8 bits fonctionnant comme suit :

Table 4.2: Table des fonctions d'une ALU 8 bits.

S4	S3	S2	S1	S0	CI	Opération	Fonction	Unité
0	0	0	0	0	0	$Y \leftarrow A$	Transfert de A	arithmétique
0	0	0	0	0	1	$Y \leftarrow A + 1$	Incrémentation de A	
0	0	0	0	1	0	$Y \leftarrow A + B$	Addition	
0	0	0	0	1	1	$Y \leftarrow A + \overline{B} + 1$	Addition avec retenue	
0	0	0	1	0	0	$Y \leftarrow A + \overline{B}$	A + compl. à 1 de B	
0	0	0	1	0	1	$Y \leftarrow A + \overline{B} + 1$	Soustraction	
0	0	0	1	1	0	$Y \leftarrow A - 1$	Décrémentation de A	
0	0	0	1	1	1	$Y \leftarrow B$	Transfert de B	
0	0	1	0	0	0	$Y \leftarrow A \text{ and } B$	AND	logique
0	0	1	0	1	0	$Y \leftarrow A \text{ or } B$	OR	
0	0	1	1	0	0	$Y \leftarrow A \text{ xor } B$	XOR	
0	0	1	1	1	0	$Y \leftarrow \overline{A}$	Complément	
0	0	0	0	0	0	$Y \leftarrow \text{NSO}$	Transfert de A	décalage
0	1	0	0	0	0	$Y \leftarrow \text{Sll NSO}$	Décalage à gauche de A	
1	0	0	0	0	0	$Y \leftarrow \text{Srl NSO}$	Décalage à droite de A	
1	1	0	0	0	0	$Y \leftarrow 0$	Transfert de zéros	

Son bloc diagramme est donne par :



Donner le code VHDL correspondant



TD 04 : Langage VHDL Instruction séquentielle

Exercice 1 :

Ecrire le code VHDL de la bascule JK synchrone. Sensible sur le front montant de l'horloge. Cette bascule présente des entrées de présélection de remise à zéro RESET (active à zéro) prioritaire a l'entrée mis à un SET (active à zéro), tous les deux sont asynchrone de l'horloge.

Exercice 2 :

Ecrire le code VHDL d'un compteur binaire 8 bits (qui compte de 0 a 255) et ayant une entrée clear; en utilisant une description comportementale. Avec

1. La méthode synchrone
2. La méthode asynchrone

Exercice 3 :

Soit un compteur/décompteur avec chargement synchrone binaire 4 bits ayant les entrées suivantes :

P(4bits), load(1bit), ck, clear, count_ena(1bit), direction (1bit)

Le compteur fonctionne comme suit :

- Si clear =0, on une remise a zero
- Si load=1 et clear=1, alors chargement par p (sortie=p)
- Si le front montant alors :
 - ✓ Si count_ena=1 et direction =0 décomptage
 - ✓ Si count_ena=1 et direction =1 comptage

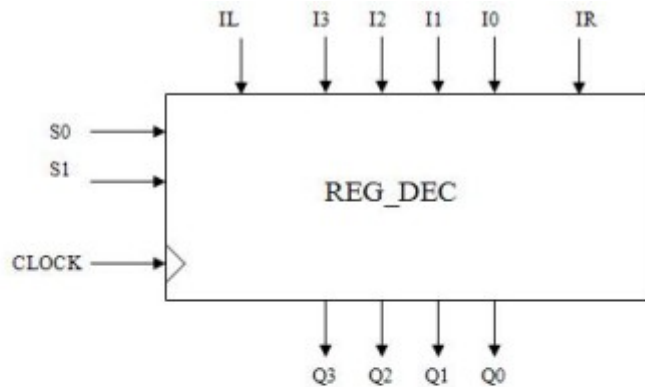
Ecrire son code VHDL en utilisant une description comportementale

Exercice 4 :

Ecrire son code VHDL d'un compteur BCD à deux digits (2 fois 4bits).

Exercice 5 :

Le circuit d'un registre à décalage synchrone (front montant) est donné sur la figure suivante :



- 4 entrées/4 sorties
- 1 entrée horloge
- 2 entrées de sélection qui permettent de choisir le mode :

S0	S1	Fonction réalisée après le top horloge	Opération
0	0	$Q[3..0] \leq Q[3..0]$	Pas de changement
0	1	$Q[3..0] \leq I[3..0]$	Chargement parallèle
1	0	$Q[3..1] \leq Q[2..0], Q[0] \leq IR$	Décalage à gauche avec IR
1	1	$Q[3] \leq IL, Q[2..0] \leq Q[3..1]$	Décalage à droite avec IL

1. Ecrire l'entité qui décrit la vue externe de ce registre.
2. Donner une architecture comportementale (tableau ci-dessus) en utilisant un process. Dans le process, vous pouvez utiliser l'instruction if.... Then.....elsif et l'instruction for.....loop.

Exercice 6 :

Ecrire un code VHDL de reconnaissance de forme de 8 bits qui fonctionne comme suite.

- Les bits sont reçus séquentiellement dans un seul signal "dataIN"
- Les 8 bits d'identification "pattern" sont reçus en parallèle, "load"=1 indique la réception de nouvelle bits à identifier.
- La correspondance se fait par le "found" qui sera mis à 1 si le signal "pattern" est le même que le signal "dataIN", c'est-à-dire il y a une identification.
- Un signal "clk" synchronise le tous.
- "Reset" permet de mètre à zéro le système.

Inputs: Reset, clk, pattern (8 bits), dataIN, load

Output: found