

# TD 03 : Langage VHDL

*TD VHDL / FPGA*

Dr. Lezzar

# Table des matières



<b>Objectifs</b>	3
<b>Introduction</b>	4
<b>I - Exercice 01</b>	5
<b>II - Exercice 02</b>	7
<b>III - Exercice 03</b>	8
<b>IV - Exercice 04</b>	10
<b>V - Exercice 05</b>	11
<b>VI - Exercice 06</b>	12

# Objectifs

- Savoir écrire un programme en langage VHDL.
- Connaître l'architecture flot de données et comportementale.
- implémenter les affectation sélective et conditionnelle dans une architecture VHDL.

# Introduction



Ce TD donne une initiation a la programmation au langage VHDL et de connaître les différente partie de programme a savoir la librerie, l'entity et l'architecture.

# Exercice 01

I

Donner la table de vérité du décodeur 2-4

## Complément

---

le décodeur possède des signaux d'entrées et quatre signaux de sortie.

## Exemple : Table de vérité d'un décodeur 2-4

---

Soit les entré A, B et les sorties Y0, Y1, Y2, Y3, il décorda l'information comme suite :

0 0 ----> 0 0 0 1

0 1 ----> 0 0 1 0

1 0 ----> 0 1 0 0

1 1 ----> 1 0 0 0

Écrire son code VHDL en utilisant :

1. L'affectation conditionnelle.

## Méthode

---

Déclaration des entrées comme entier et la sortie comme des vecteurs

## Rappel

---

L'affectation conditionnelle utilisation des termes when/else

## Exemple

---

```
Y<= "0000" when A= 0 else
```

```
Y<= "0001" when A= 1;
```

2. L'affectation sélective.

## Méthode

---

En déclarant les entrées et les sorties comme vecteurs.

◆ *Rappel*

---

L'affectation sélective utilise les terme with/select/when

☞ *Exemple*

---

```
with S select
```

```
Y<= "0000" when A= '0' ;
```

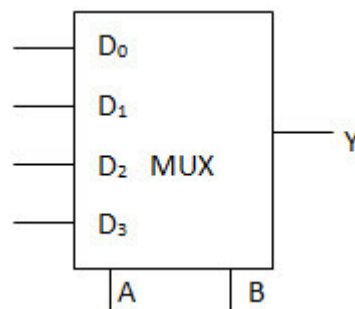
```
Y<= "0001" when A= '1' ;
```

# Exercice 02

II

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

A et B sont des signaux de sélections.




1. *L'affectation sélective (utilisation des vecteurs).*

2. *L'affectation conditionnelle.*

# Exercice 03

III

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

 Complément : Décodeur BCD 7-segment



Chiffre converti	Entrées				Sorties						
	E <sub>3</sub>	E <sub>2</sub>	E <sub>1</sub>	E <sub>0</sub>	S <sub>0</sub>	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>
0	0	0	0	0	1	0	1	1	1	1	1
1	0	0	0	1	1	0	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	1	1	1	0	0	1	0
5	0	1	0	1	0	1	1	1	0	1	1
6	0	1	1	0	0	1	1	1	1	1	1
7	0	1	1	1	1	0	1	0	0	0	1
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1
A	1	0	1	0	1	1	1	0	1	1	1
B	1	0	1	1	0	1	1	1	1	1	0
C	1	1	0	0	0	0	0	1	1	1	1
D	1	1	0	1	1	1	1	1	1	0	0
E	1	1	1	0	0	1	0	1	1	1	1
F	1	1	1	1	0	1	0	0	1	1	1

Table de vérité de décodeur BCD 7-segments

 **Rappel**

L'affectation case doit être à l'intérieur d'un process elle utilise les terme case\is\when

 **Exemple**

case A is



```
when '1' =>
```

```
  B<= '0' ;
```

```
  C<= "0001" ;
```

```
when '0' =>
```

```
  B<='0' ;
```


```
  C<= "0010" ;
```

# Exercice 04

IV

Écrire un code VHDL du transcodeur en utilisant :

1. L'architecture comportementale avec l'affectation *if/then/elsif*.

 Complément : La table de vérité suivante d'un Transcodeur

A	B	C	D	E	F	G	H
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

A,B,C,D représente les entrées et E,F,G,H représente les sorties.

 *Remarque*

L'affectation *if/elsif* doit être dans un process

 *Exemple : Affectation if/elsif*

```

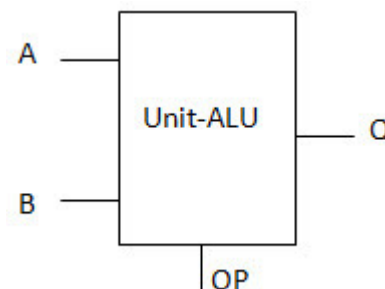
if A="0000" then
B<="0001" ;
elsif A="0001" then
B<="0011" ;
end if ;

```

# Exercice 05



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.



## *Méthode*

---

Définir un package qui contient un nouveau type de signal de sélection (OP) comme suite (ADD, SOUS, INC, DEC, DECAL, ET, OU, NON, EXOR).

## *Complément*

---

Un bus de données est un signal de 7 bits.

1. Écrire le code VHDL de cette ALU.

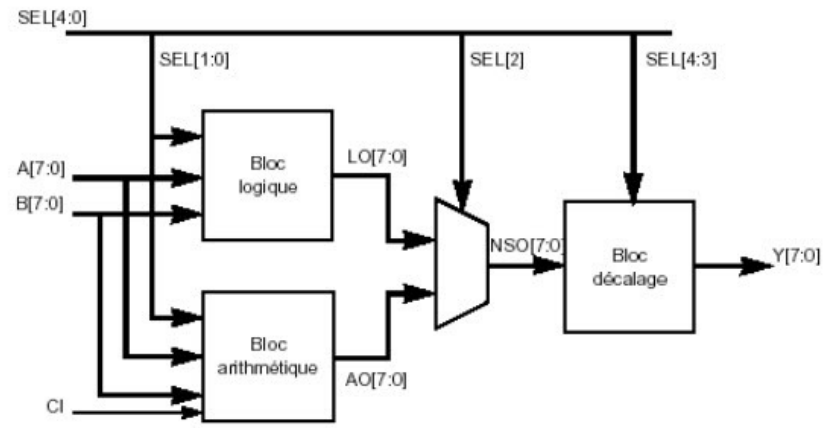
# Exercice 06

VI

Soit une ALU de 8 bits fonctionnant comme suit :

S4	S3	S2	S1	S0	CI	Opération	Fonction	Unité
0	0	0	0	0	0	$AO \leq A$	Transfert de A	Arithmétique
0	0	0	0	0	1	$AO \leq A+1$	Incrémentation de A	Arithmétique
0	0	0	0	1	0	$AO \leq A+B$	Addition	Arithmétique
0	0	0	0	1	1	$AO \leq A+B+1$	Addition avec retenue	Arithmétique
0	0	0	1	0	0	$AO \leq A+\text{not}(B)$	A+ compl. à 1 de B	Arithmétique
0	0	0	1	0	1	$AO \leq A+\text{not}(B)+1$	Soustraction	Arithmétique
0	0	0	1	1	0	$AO \leq A-1$	Décrémentation de A	Arithmétique
0	0	0	1	1	1	$AO \leq B$	Transfert de B	Arithmétique
0	0	1	0	0	0	$LO \leq A \text{ and } B$	AND	Logique
0	0	1	0	1	0	$LO \leq A \text{ or } B$	OR	Logique
0	0	1	1	0	0	$LO \leq A \text{ xor } B$	XOR	Logique
0	0	1	1	1	0	$LO \leq \text{not}(A)$	Complément	Logique
0	0	0	0	0	0	$Y \leq \text{NSO}$	Transfert de A	Décalage
0	1	0	0	0	0	$Y \leq \text{sll NSO}$	Décalage à gauche de A	Décalage
1	0	0	0	0	0	$Y \leq \text{slr NSO}$	Décalage à droite de A	Décalage
1	1	0	0	0	0	$Y \leq 0$	Transfert de 0	Décalage

Son bloc diagramme est donne par :



1. Donner le code VHDL correspondant