

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE  
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE  
UNIVERSITÉ AKLI MOHAND OULHADJ BOUIRA



FACULTÉ DES SCIENCES ET DES SCIENCES APPLIQUÉES  
DEPARTEMENT DE GENIE ELECTRIQUE

# Electronique Numériques Avancées : FPGA, VHDL

Cours présenté par :  
**M. Arezki FEKIK**

Dans le cadre de la formation **Master 1 Electroniques des Systèmes Embarqués**

# Introduction générale

- ▶ **VHDL** est l'acronyme de **VHSIC HDL** (*Very High Speed Integrated Circuit Hardware Description Language*), c'est un langage de description matérielle qui a été créé dans les années 1980 à la demande du département de la défense américaine (**DOD**).
- ▶ La première version du **VHDL** accessible au public a été publiée en 1985, et a fait l'objet d'une norme internationale en **1986** par l'institut des ingénieurs électriciens et électroniciens (**IEEE**).
- ▶ De nos jours, le langage **VHDL** devient un outil indispensable pour la conception des systèmes électroniques intégrés, il est proposé par la grande majorité des sociétés de développement et la commercialisation d'**ASIC** et d'**FPGA** telle que la société américaine **Xilinx**.
- ▶ Avec un langage de description matérielle et un **FPGA** (*Field Programmable Gate Array*), un concepteur peut développer rapidement et simuler un circuit numérique sophistiqué, de l'implémenter sur une carte de prototypage, et de vérifier son fonctionnement.

# PLAN DE COURS



**Chapitre. I: Concepts de base du langage VHDL**

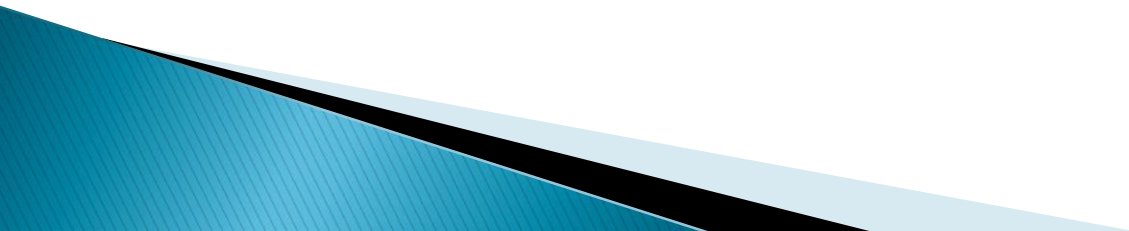
**Chapitre. II: Objets et types de données**

**Chapitre. III: Différentes descriptions d'une architecture**

**Chapitre. IV: Modélisation des circuits séquentiels**

**Chapitre. V: Exemple De FPGA : La Famille Spartan-6**

# **Chapitre. III: Les différents styles de descriptions d'une architecture**



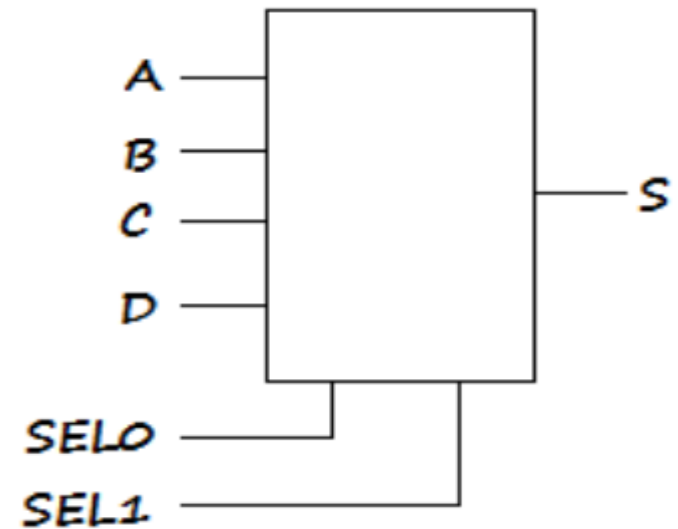
# Chapitre. III: Les différents styles de descriptions d'une architecture

- ▶ Les trois descriptions principales sont :
  - Description par flot de données.
  - Description comportemental (appelée aussi procédural).
  - Description structurelle.
- ▶ Il existe aussi des styles de descriptions mixtes, ces derniers combinent les trois styles de descriptions citées auparavant, et enfin une description très importante pour la simulation qui est l'architecture de test.

# Chapitre. III: Les différents styles de descriptions d'une architecture

## ► Description par flot de données :

Prenons comme exemple un multiplexeur 4 vers 1 (**Figure III.1**) et décrivons-le avec le style de description flot de données (**Listing III.1**).



**Figure III.1** Symbole logique d'un multiplexeur 4 vers 1.



# Chapitre. III: Les différents styles de descriptions d'une architecture

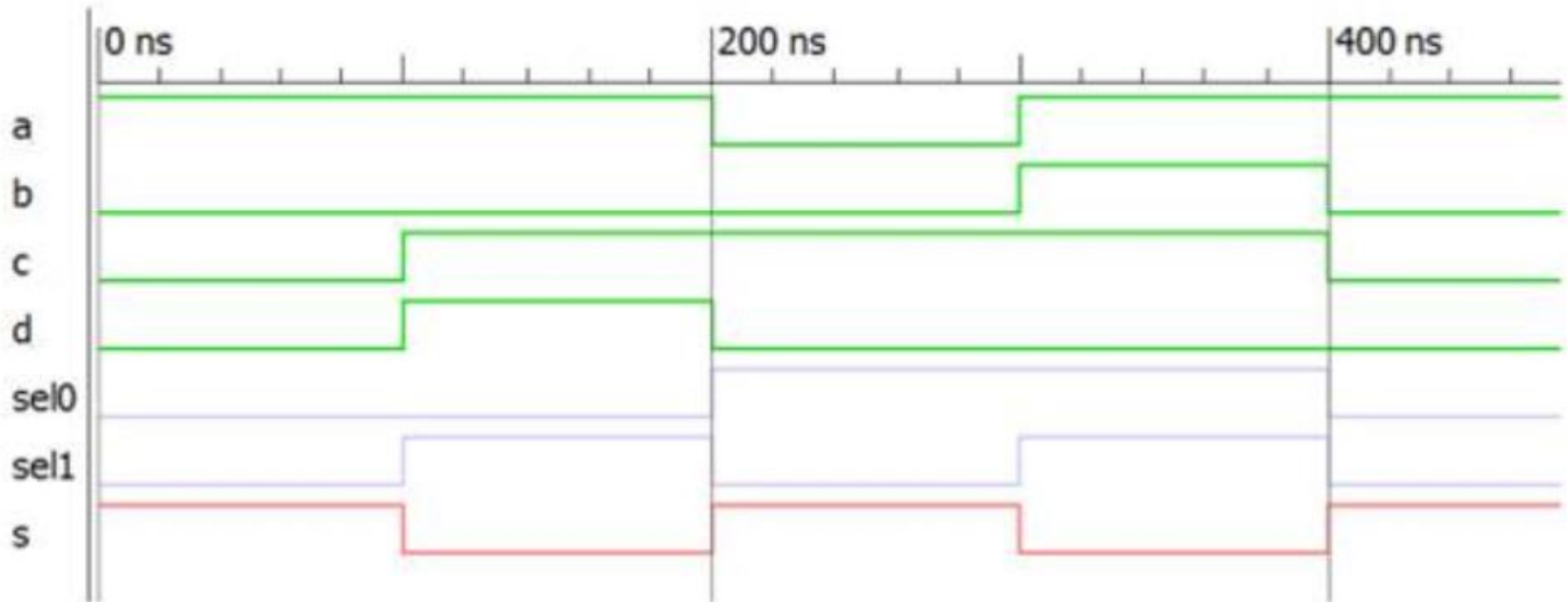
## ► Description par flot de données :

```
1  library ieee;  
2  use ieee.std_logic_1164.all;  
3  
4  entity mux_4v1 is  
5      port ( A, B, C, D, SEL0, SEL1 : in std_logic;  
6            S : out std_logic);  
7  end mux_4v1;  
8  
9  architecture data_flow of mux_4v1 is  
10 begin  
11     S <= (not(SEL0) and not(SEL1) and A) or  
12          (not(SEL0) and SEL1 and B) or  
13          (SEL0 and not(SEL1) and C) or  
14          (SEL0 and SEL1 and D);  
15 end data_flow;
```

**Listing III.1** Description VHDL avec le style de description flot de données d'un multiplexeur

# Chapitre. III: Les différents styles de descriptions d'une architecture

## ► Simulation:



**Fig III.2** Simulation du code VHDL avec le style de description flot de données d'un multiplexeur



# Chapitre. III: Les différents styles de descriptions d'une architecture

## ► Description comportementale :

Appelée aussi description procédurale, elle décrit le comportement d'un circuit selon des conditions (**IF**), des cas (**CASE**, **WHILE**), et des boucles (**FOR**).

> On utilise le **PROCESS** pour faciliter la modélisation d'un système.

> On peut le considérer comme une boîte noire où il y'a des instructions à l'intérieur qui s'exécutent séquentiellement

**Exemple :**

```
21  process (A, B)
22      begin
23          S <= A and B;
24  end process;
```

# Chapitre. III: Les différents styles de descriptions d'une architecture

## ► Description avec l'instruction if :

```
if condition_1 then  
    séquence_instructions_1 ;  
elseif condition_2 then  
    séquence_instructions_2 ;  
else  
    séquence_instruction_3 ;  
end if ;
```

### Exemple 1 :

```
if signal_x = '1' then  
    y <= '0' ;  
end if ;
```

# Chapitre. III: Les différents styles de descriptions d'une architecture

## ► Description avec l'instruction CASE

case instruction is

when choix\_1 => instruction\_1 ;

when choix\_2 => instruction\_2;

when choix\_3 => instruction\_3;

...

when others => instruction\_n;

```
10 architecture Exemple_Case of Case is
11 begin
12     process (E1, E2, SEL)
13     begin
14         case SEL is
15             when "00" => S <= E1;
16             when "01" => S <= E2;
17
18             when others => S <= '0';
19         end case;
20
21     end process;
22 end Exemple Case ;
```

# Chapitre. III: Les différents styles de descriptions d'une architecture

## ► Description avec les boucles (loop) :

En VHDL, les boucles sont utilisés quand il y'a répétition d'une opération. Les répétitions peuvent être itératives (**for**), conditionnelles (**while**)

### A- Répétitions itératives (for) :

La syntaxe simplifiée d'une boucle for est :

```
for indice in intervalle_boucle loop  
  
    instructions_séquentielles ;  
  
end loop;
```

```
1  architecture boucle of Exemple_for is  
2  begin  
3      process (A, B)  
4      begin  
5          for I in 3 downto 0 loop  
6              S(I) <= A(I) xor B(I) ;  
7          end loop;  
8      end process;  
9  end boucle;  
10
```

# Chapitre. III: Les différents styles de descriptions d'une architecture

## ► Description avec les boucles (loop) :

### **B- Répétitions itératives (While) :**

La syntaxe simplifiée d'une boucle **while** est :

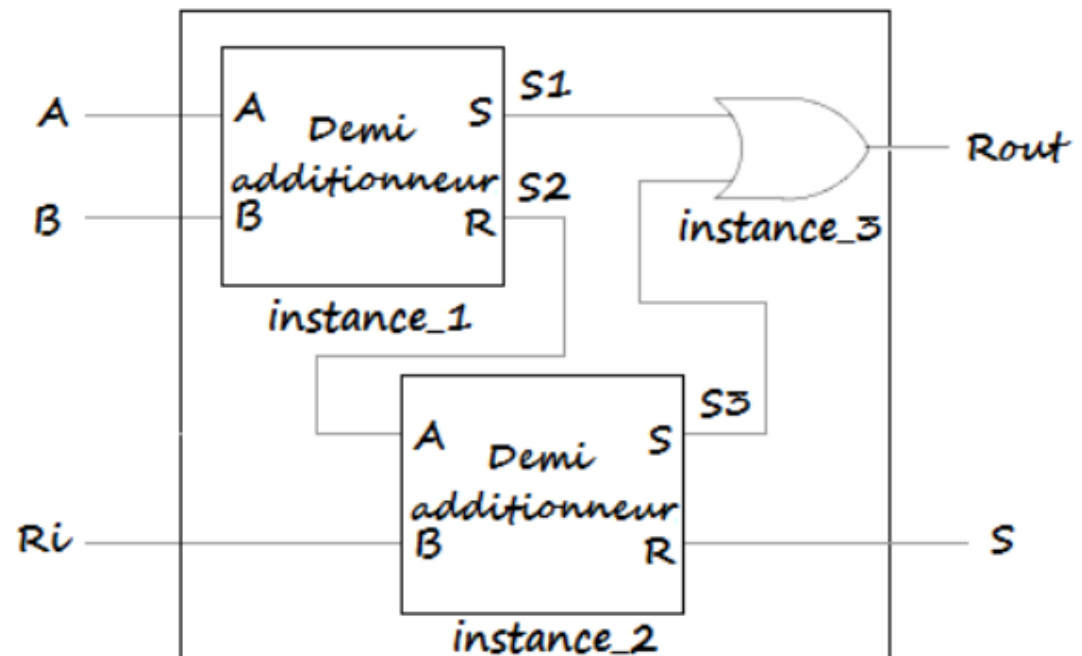
```
while condition_booléenne loop  
    instructions_séquentielles ;  
end loop;
```

```
1  architecture boucle of Exemple_While is  
2  begin  
3      process (A, B)  
4          variable I : integer := 0;  
5      begin  
6          while I < 5 loop  
7              S(I) <= A(I) xor B(I);  
8          end loop;  
9      end process;  
10 end boucle;
```

# Chapitre. III: Les différents styles de descriptions d'une architecture

## ► Description structurelle :

Lorsqu'on a un système électronique composé, on peut structurer et séparer ces composants en des blocs plus petits pour les décrire plus simplement.





# Chapitre. III: Les différents styles de descriptions d'une architecture

## ► Description structurelle :

Chaque bloc doit être décrit séparément (c'est-à-dire avoir son entité et l'architecture associée)

```
1  library ieee;  
2  use ieee.std_logic_1164.all;  
3  
4  entity porte_ou is  
5      port(E1, E2 : in std_logic;  
6           Y : out std_logic);  
7  end porte_ou;  
8  
9  architecture archi_ou of porte_ou is  
10 begin  
11     Y <= E1 or E2;  
12 end archi_ou;
```

# Chapitre. III: Les différents styles de descriptions d'une architecture

## ► Description structurelle :

```
1  library ieee;  
2  use ieee.std_logic_1164.all;  
3  
4  entity half_adder is  
5      port(A, B : in std_logic;  
6           S, R : out std_logic);  
7  end half_adder;  
8  
9  architecture archi_addi of half_adder is  
10 begin  
11     S <= A xor B;  
12     R <= A and B;  
13 end archi_addi;
```

Listing A

```
component half_adder  
    port (A, B : in std_logic;  
          S, R : out std_logic);  
end component;  
  
component porte_ou  
    port (E1, E2 : in std_logic;  
          Y : out std_logic);  
end component;
```

Listing B

# Chapitre. III: Les différents styles de descriptions d'une architecture

## ► Description structurelle :

### **A- Instanciation :**

Il y'a trois types d'instanciation :

- Instanciation par position
- Instanciation par nom
- Instanciation mixte

# Chapitre. III: Les différents styles de descriptions d'une architecture

## A-1: Instanciation par position

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  entity full_adder is
5      port(A, B, Ri : in std_logic;
6           S, Rout : out std_logic);
7  end full_adder;
8
9  architecture struct_adder of full_adder is
10
11      component half_adder
12          port (A, B : in std_logic;
13               S, R : out std_logic);
14      end component;
15
16      component porte_or
17          port (E1, E2 : in std_logic;
18               Y : out std_logic);
19      end component;
20
21      signal S1, S2, S3 : std_logic;
22
23  begin
24
25      instance_1 : half_adder port map (A, B, S1, S2);
26      instance_2 : half_adder port map (S2, Ri, S3, S);
27      instance_3 : porte_or port map (S1, S2, Rout);
28
29  end struct_adder;
```

# Chapitre. III: Les différents styles de descriptions d'une architecture

## A-2: Instanciation par nom

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  entity full_adder is
5      port(A, B, Ri : in std_logic;
6           S, Rout : out std_logic);
7  end full_adder;
8
9  architecture struct_adder of full_adder is
10
11      component half_adder
12          port (A, B : in std_logic;
13               S, R : out std_logic);
14      end component;
15
16      component porte_ou
17          port (E1, E2 : in std_logic;
18               Y : out std_logic);
19      end component;
20
21      signal S1, S2, S3 : std_logic;
22
23  begin
24
25      instance_1 : half_adder port map (A => A, S => S1, B => B, R => S2);
26      instance_2 : half_adder port map (A => S2, B => Ri, S => S3, R => S);
27      instance_3 : porte_ou port map (E1 => S1, Y => Rout, E2 => S3);
28
29  end struct_adder;
```

# Chapitre. III: Les différents styles de descriptions d'une architecture

## A-3: Instanciation mixte

```
23 begin
24
25     instance_1 : half_adder port map (A, B, S1, R => S2);
26     instance_2 : half_adder port map (S2, Ri, S => S3, R => S);
27     instance_3 : porte_ou port map (S1, E2 => S3, Y => Rout);
28
29 end struct_adder;
```