

## 1. Structure d'un module VHDL

### 1.1. Ressources externes

```
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;  
use IEEE.STD_LOGIC_ARITH.ALL;  
use IEEE.STD_LOGIC_UNSIGNED.ALL;
```

### 1.2. Entité

```
entity nom_entite is  
  
port( clk :    in STD_LOGIC;  -- entree  
      oe :    in STD_LOGIC;  
      raz :    in STD_LOGIC;  
      sq :    out STD_LOGIC;  -- sortie  
      data :   out STD_LOGIC_VECTOR(3 DOWNTO 0) -- sorties (bus)  
      sd :    out STD_LOGIC;  
);  
end nom_entite;
```

### 1.3. Architecture

```
architecture Behavioral of nom_entite is  
    signal conv, total : STD_LOGIC_VECTOR(3 DOWNTO 0);  
  
begin  
    -----  
    -- processus  
    cpt : process(clk)  
    begin  
        if raz = '1' then total <= "0000"; -- description asynchrone  
        elsif clk'event and clk='1' then -- description synchrone  
            sd <= not oe;  
            total <= total+1;  
            -- etc  
        end if;  
    end process;  
    -----  
  
    sq <= total(3); -- hors processus  
  
end Behavioral;
```

## 2. Objets et types en VHDL

### 2.1. Objets

- **signal** objet physique, associé à des évènements
- **variable** intermédiaire de calcul, non physique
- **constant**

### 2.2. Types

Types de base : **bit**, **bit\_vector**, **integer**, **boolean**

Types IEEE : **std\_logic**, **std\_logic\_vector**, **signed**, **unsigned**

Types définis par l'utilisateur :

- type énuméré, exemple : `type jour is (lu, ma, me, je, ve, sa, di);` (souvent utilisé dans les machines à état)

- sous-type : `subtype octet is bit_vector(0 to 7);`

### 2.3. Notations

bit : '0' ou '1' ; bit\_vector : "0100" ; ASCII : "Texte" ; Décimal : 423 ; Hexadécimal : x"1A"

## 3. Opérateurs en VHDL

**LOGIQUES** : `and`, `nand`, `or`, `nor`, `xor`, `xnor`, `not`

**DÉCALAGE** : `sll`, `slr`, `sla`, `sra`, `rol`, `ror`

**RELATIONNELS** : `=`, `/=`, `<`, `>`, `<=`, `>=`

**ARITHMÉTIQUES** : `+`, `-`, `*`, `/`, `MOD`

**CONCATENATION** : `&`

**AFFECTATION** : `<=`

## 4. Instructions en VHDL

### 4.1. Hors processus

Ces instructions décrivent le plus souvent des éléments combinatoires (concurrentes).

#### 4.1.1 Affectation conditionnelle

```
s <=  a when cond1 else
      b when cond2 else
      ...
      z ;
```

#### 4.1.2 Affectation sélective

```
with expr select
s <=  a when val1 ,
      b when val2 ,
      ...
      z when others ;
```

## 4.2. Dans un processus

L'exécution des instructions s'effectue dans un ordre séquentiel (ordre d'écriture). Le processus est activé lorsqu'un des éléments de la liste de sensibilité est modifié.

La mise à jour des objets s'effectue simultanément à la fin du processus.

### 4.2.1 Syntaxe

```
label : process(liste des signaux de sensibilité)
.. objets/signaux internes
begin
.. description séquentielle
end process;
```

### 4.2.2 Test : SI

```
if x="00" then
  y <= '0';
elsif x="01" then
  y <= '1';
end if;
```

### 4.2.3 Test : CAS

```
case x is
  when "00" => y <= "00";
  when "01" => y <= "10";
  when others => y <= "11";
end case;
```

## 4.3. Autres codes

### 4.3.1 Blocs répétitifs / Boucle spatiale

```
for i in 0 to 5 loop
  instructions;
end loop;
```

```
while i < 5 loop
  instructions;
end loop;
```

### 4.3.2 Temporisation

```
wait until CLK'event and CLK='1';
instructions;
```

## 5. Tournures fréquentes en VHDL

### 5.1. Détection d'un front

```
if clk 'event and clk='1' then ... ; -- front montant
if clk 'event and clk='0' then ... ; -- front descendant
```

### 5.2. Remplissage d'un vecteur (bit\_vector)

```
x <= (others => '0'); -- tous les bits a 0
x <= (others => '1'); -- tous les bits a 1
```

## 6. Brochage de la carte d'étude

K = interrupteur    LED = diode électroluminescente    CLK = horloge  
an = commande des afficheurs 7 segments    a..g = segments des afficheurs

élément	broche	type
K(0)	p24	entrée
K(1)	p25	entrée
K(2)	p26	entrée
K(3)	p27	entrée
K(4)	p28	entrée
K(5)	p29	entrée
K(6)	p33	entrée
K(7)	p34	entrée

élément	broche	type
LED(0)	p22	sortie
LED(1)	p20	sortie
LED(2)	p19	sortie
LED(3)	p18	sortie
LED(4)	p14	sortie
LED(5)	p13	sortie
LED(6)	p12	sortie
LED(7)	p11	sortie

élément	broche	type
g, seg(0)	p9	sortie
f, seg(1)	p35	sortie
e, seg(2)	p36	sortie
d, seg(3)	p37	sortie
c, seg(4)	p38	sortie
b, seg(5)	p43	sortie
a, seg(6)	p44	sortie
dp, seg(7)	p8	sortie

élément	broche	type
an(0)	p4	commande anode gauche
an(1)	p3	commande anode
an(2)	p1	commande anode
an(3)	p2	commande anode droite

élément	broche	fonction principale	caractéristique
CLK_1M	p5	horloge	fréquence 1MHz
CLK_256	p6	horloge	fréquence 256Hz
CLK_SLOW	p7	horloge	pas à pas ou 2Hz
RESET	p39	remise à zéro	impulsion