ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Enseignes et afficheurs à LED

Introduction au VHDL

Dr. Mamadou Lamine NDIAYE

Introduction au VHDL

Dr. Mamadou Lamine NDIAYE

Introduction au VHDL

- Introduction au VHDL
- Les concepts de base
  - Unités de conception
  - Les objets et les types de base
  - Les opérateurs
  - Les tableaux

Introduction au VHDL

- VHDL signifie VHSIC Hardware Description Language

Introduction au VHDL

- VHDL signifie VHSIC Hardware Description Language
- VHSIC = Very High Speed Integrated Circuit

Introduction au VHDL

- VHDL signifie VHSIC Hardware Description Language
- VHSIC = Very High Speed Integrated Circuit
- VHDL est un langage international défini par la norme IEEE qui permet
  - Synthèse logique des circuits numériques
  - Synthèse physique des circuits numériques
- Les objectifs du langage VHDL
  - Conception de circuits intégrés reconfigurables ou non (ASIC, FPGA…) : SYNTHESE
  - Mise au point de modèle de simulations numériques (circuits virtuels) : MODELISATION

Introduction au VHDL

VHDL : Les concepts de base

Unités de conception : deux parties dépendantes (obligatoires)

- L'entité (Entity) qui représente la vue extérieure du composant
  - Déclare les modes des ports en entrée et/ou sortie (in, out, inout, buffer)
  - IN port à lecture seule
  - OUT port à écriture seule
  - INOUT port à lecture/écriture
  - BUFFER écriture/lecture sur un port

VHDL : Les concepts de base

Unités de conception : deux parties dépendantes (obligatoires)

- L'entité (Entity) qui représente la vue extérieure du composant
  - Déclare les modes des ports en entrée et/ou sortie (in, out, inout, buffer)
  - IN port à lecture seule
  - OUT port à écriture seule
  - INOUT port à lecture/écriture
  - BUFFER écriture/lecture sur un port
- L'architecture (Architecture) qui décrit le fonctionnement du composant
  - Représente la vue interne du composant
  - Décrit le comportement de la fonction à synthétiser
  - Décrit les signaux internes, les composants, les constantes, les types d'objets, les déclarations de sous programmes dans la partie dite déclarative (début)
  - A une entité peut correspondre plusieurs architectures

Structure générale d'un programme VHDL

```
Entity Nom_entite is
-- déclarations de paramètres, ports..
Begin

End  Nom_entite;

Architecture Nom_archi of Nom_entite is
-- Zone déclarative...
begin
--instructions concurrentes...
P1: process
-- déclarations...
begin
--instructions séquentielles...
end process P1;
-- instructions concurrentes...
P2: process
-- déclarations...
begin
--instructions séquentielles...
end process P2;
-- instructions concurrentes...
End Nom_archi ;
```

Structure générale d'un programme VHDL

Exemple :

Multiplexeur 2 vers 1

Structure générale d'un programme VHDL

Exemple :

Multiplexeur 2 vers 1


```
Library ieee;
Use ieee.std_logic_1164.all;
Use ieee.numeric_std.all;

-- multiplexeur 2 vers 1 (commentaire)

Entity mux2 is      -- commentaire


end mux2;

-- Déclaration de l'architecture (commentaire)

architecture archi_mux2 of mux2 is




end archi_mux2;
```

Déclaration des bibliothèques

Commentaires en VHDL

Déclaration de l'entité du multiplexeur

Déclaration de l'architecture du multiplexeur

Structure générale d'un programme VHDL

Exemple :

Multiplexeur 2 vers 1


```
Library ieee;
Use ieee.std_logic_1164.all;
Use ieee.numeric_std.all;

-- multiplexeur 2 vers 1 (commentaire)

Entity mux2 is                  -- commentaire
Port( e1, e2  : in std_logic;
sel : std_logic;                -- par défaut le mode in est pris en compte
s :  out std_logic              -- mode out à préciser
);
end mux2;

-- Déclaration de l'architecture (commentaire)

architecture archi_mux2 of mux2 is
begin
with sel select
s <= e1 when  '1',      -- s recopie e1 lorsque sel =  '1'
     e2 when others;    -- recopie e2 dans tous les autres cas
end archi_mux2;
```

Déclaration des bibliothèques

Commentaires en VHDL

Déclaration de l'entité du multiplexeur

Déclaration de l'architecture du multiplexeur

Objets et types de base

Quatre objets « typés »

- Les constantes (CONSTANT), objet dont la valeur ne change pas durant la simulation.

- Les variables (VARIABLE) objet dont la valeur peut changer par affectation au cours de la simulation. Une variable n'existe que dans un PROCESS.

Objets et types de base

Quatre objets « typés »

- Les constantes (CONSTANT), objet dont la valeur ne change pas durant la simulation.
- Les variables (VARIABLE) objet dont la valeur peut changer par affectation au cours de la simulation. Une variable n'existe que dans un PROCESS.
- Les signaux (SIGNAL) désignent des signaux internes qui ne sortent pas du composant.
- Les fichiers (FILE) permettent l'échange de données entre l'extérieur et le simulateur VHDL.

Objets et types de base

Quelques exemples

```
CONSTANT cte : INTEGER  :=10;

VARIABLE vari : BIT_VECTOR(3 DOWNTO 0):= "0000";

VARIABLE n : INTEGER range 0 to 65535;

SIGNAL sig : std_logic;

SIGNAL  a : bit_vector (3 downto 0);

FILE fich : TEXT IS IN  "entree.txt";
```

Objets et types de base

Quelques exemples

Bit : peut prendre les valeurs `0´ ou `1´.

Bit_vector : groupe de bits défini entre " "

Boolean: peut prendre les valeurs true ou false.

Integer : Valeur entière codée sur 32 bits (de - 2.147.483.648 à 2.147.483.647)

Std_logic, std_ulogic : Similaires au bit mais avec 9 états possibles.

Time : comptage du temps sur 64 bits, en ps, ns, us, ms, sec, min, hr (non synthétisable) ;

Opérateurs en VHDL

Les opérateurs arithmétiques et logiques

VHDL manipule six opérateurs hiérarchisés

- 1 Opérateurs divers : **, abs, not
- 2 Opérateurs de multiplication : *, /, mod, rem
- 3 Opérateurs d'addition : +, -, &
- 4 Opérateurs relationnels : =, /=, < , <= , >, >=
- 5 Opérateurs de décalage : sll, srl , sla,sra,rol,ror
- 6 Opérateurs logiques : and , or, nand, nor, xor, xnor

Littéraux en VHDL

Les littéraux (**valeurs explicites**)

- Caractères : ‘1’ , ‘c’ , ‘b’ , ‘# ‘
- Chaînes : "10110100", "bonjour", "x@&"
- Chaînes de bits : B"0010_1101", X "2F", O "265"
- Décimaux : 27, -5, 4e3, 76_562, 4.25
- Basés : 2#1001#, 8#65_07, 16#C5#e2

- Une chaine de bit est représenté par un vecteur de bits
- std_logic_vector(7 downto 0); -- bit poids fort => poids faible
- std_logic_vector(0 to 7); -- bit faible poids => poids fort

Tableaux en VHDL

Les tableaux (ARRAY)

- Les tableaux sont des collections de données de même type
- Les données sont rangées suivant un index (entier) ou des énumérés

Tableaux en VHDL

Les tableaux (ARRAY)

- Les tableaux sont des collections de données de même type
- Les données sont rangées suivant un index (entier) ou des énumérés

```
type bus is array (0 to 31) of bit;
type COULEURS is (ROUGE, JAUNE, BLEU, VERT, ORANGE);                    -- type énuméré
TYPE memoire IS ARRAY(0 TO 31) OF STD_LOGIC_VECTOR(7 DOWNTO 0);         -- déclaration du tableau mémoire
type TABLEAU1 is array(0 to 31) of bit_vector(7 downto 0);
type TABLEAU2 is array(0 to 31, 0 to 7) of bit;

SIGNAL Tab_Mem: memoire;        -- création d'un tableau de type memoire
Signal S1 : TABLEAU1;
Signal S2 : TABLEAU2;
Begin
S1(0) <="01101011";             -- Modification des bits de la première ligne
S1 (31)(5) <= '1';              -- Modification du bit la 32ieme ligne 6ime colonne de S1
S2(4,7) <= '0';                 -- Modification du bit la 5ieme ligne 8ime colonne de S2
```

Introduction au VHDL

- Introduction au VHDL
- Les concepts de base
  - Unités de conception
  - Les objets et les types de base
  - Les opérateurs
  - Les tableaux