

LE VHDL PAR L'EXEMPLE

I. Introduction

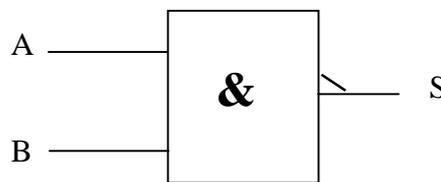
Pour décrire une fonction logique combinatoire ou séquentielle, nous connaissons plusieurs représentations :

Table de vérité ou table de transitions, Logigrammes, chronogrammes, diagramme d'états, textes etc..

Exemple : La fonction NAND :

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

Table de vérité



Logigramme

La sortie S est à 0 si et seulement si toutes les entrées sont à l'état 1.

Texte

Le langage VHDL permet de décrire le comportement d'un circuit logique.

Exemple : la fonction NAND en VHDL sera décrit comme suit :

```
Library IEEE ;
USE ieee.std_logic_1164.all ;
USE work.std_arith.all ;

entity PORTE is
    Port ( A, B : in std_logic ;
          S : out std_logic ) ;
end PORTE

architecture ARCH_PORTE of porte is
begin
    S<= A nand B ;
end ARCH_PORTE ;
```

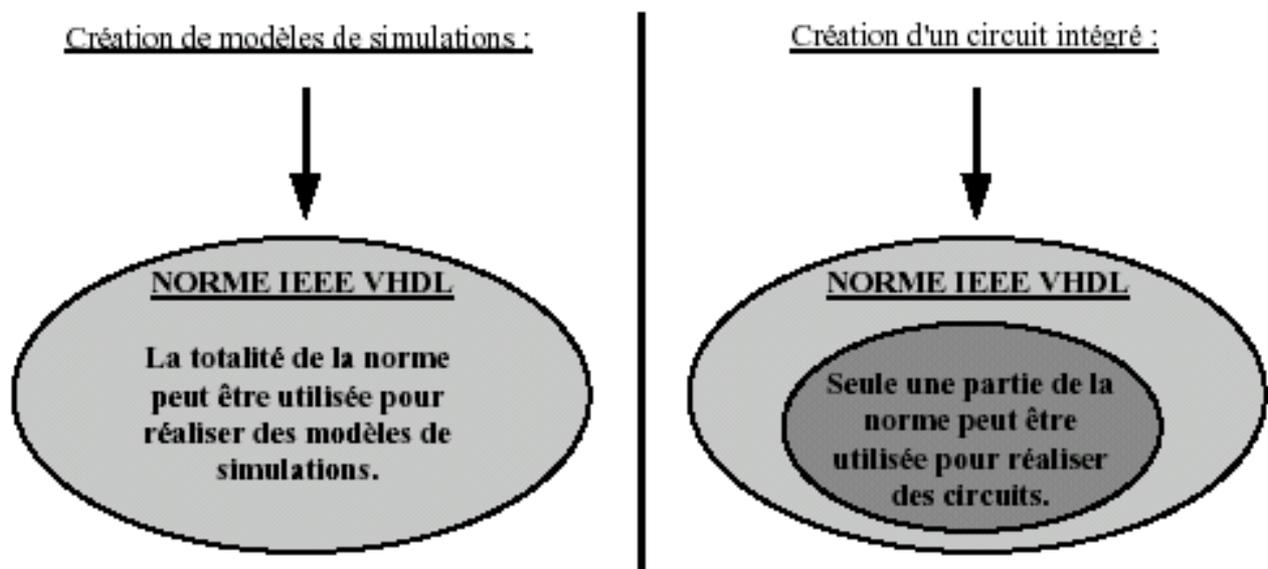
Ce langage est utilisé pour développer les circuits logiques programmables. A partir d'un fichier VHDL, un outil de développement informatique nous permettra la programmation d'un circuit logique programmable.

II. Présentation

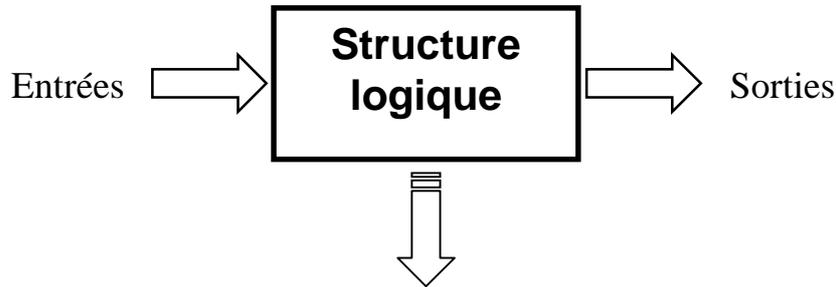
Le langage VHDL :

- ⊗ est un langage qui décrit un comportement
- ⊗ est utilisé pour développer les circuits logiques programmables (PLDs)
- ⊗ est normalisé : normes IEEE 1076 & 1164

Les limites actuelles



III. Structure d'une description VHDL



L'accès aux bibliothèques

library *nom de la bibliothèque* ;
use permet l'accès aux fonctions de la bibliothèque **all** ;

La déclaration d'entité

entity *Nom_de_l'entité* **is**
 generic déclaration des paramètres (éventuellement) ;
 port description des E/S de la structure en explicitant pour chacune d'entre elles, **le nom**, la direction ou **le mode** et **le type** ;
 attribute *Pin_numbers* **of** *Nom_de_l'entité*: **entity is**
 attribution d'un numéro de broche à chaque E/S
 "*nom* : *n°broche* *nom* : *n°broche* " &
 "*nom* : *n°broche* *nom* : *n°broche* " ;
end *Nom_de_l'entité* ;

Outil de développement « WARP »

Une même entité peut avoir plusieurs architectures

L'architecture

zone de déclaration optionnelle
begin
description de la structure logique
end *Nom_de_l'architecture* ;

Nous allons voir sur quelques exemples la mise en œuvre du VHDL.

IV. Exemples de descriptions VHDL en logique combinatoire

IV.1 Fonctions logiques combinatoires de base

On donne la description VHDL suivante :

```
Library ieee ;
USE ieee.std_logic_1164.all ;
USE work.std_arith.all ;

entity PORTES is
    Port ( A, B : in std_logic ;
          Y1,Y2,Y3,Y4,Y5,Y6,Y7 : out std_logic ) ;

Attribute pin_numbers of PORTES : entity is
    " A: 2  B :3  Y1:23 Y2:22 Y3:21 "&
    " Y4:20  Y5:19  Y6:18 Y7:17"
end PORTES ;

architecture ARCH_PORTES  of PORTES is
begin
    Y1 <= A and B;
    Y2 <= A or B;
    Y3 <= A xor B;
    Y4 <= not A ;
    Y5 <= A nand B;
    Y6 <= A nor B;
    Y7 <= not (A xor B);

end ARCH_PORTES ;
```

Q.1) A partir du fichier, donner le nom des bibliothèques utilisées.

Q.2) Quel est le nom de l'entité ? Quel est le nom de l'architecture ?

Q.3) Représenter le schéma fonctionnel de la fonction. On placera naturellement les entrées à gauche et les sorties à droite. On reportera sur ce schéma les numéros des broches du circuit.

Q.4) Représenter le schéma structurel du circuit.

Q.5) A l'aide de l'outil Warp, saisir le fichier, compiler et simuler la structure. On se reportera à la documentation des logiciels fournis. On imprimera tous les documents intéressants.

Q.6) Programmer le circuit Pal, l'installer sur la maquette et valider le fonctionnement.

IV.2 Ecriture d'une structure VHDL combinatoire.

On désire réaliser un programme de test de la maquette. Le but est de tester toutes les entrées et toutes les sorties du circuit.

On donne :

La procédure de test de la carte est donnée, le schéma structurel de la maquette.:

Extrait de la procédure de test de la maquette :

II.4 Test fonctionnel

Cavalier J2: présent, cavalier J1: présent.

Cavalier J8 : présent J10 : Absent

RV1 minimum (clignotement rapide)

Vérifier que la LED D1 clignote.

Appuyer sur BP1, vérifier que D2 clignote en opposition de phase par rapport à D1.

Relâcher BP1, faire basculer SW1 vers la droite, même constatation. Re-basculer SW1 vers la gauche.

Même manipulation avec BP2/ D3/ SW2 puis BP3/D4/SW3 puis BP4/D4/SW4.

Faire basculer vers la droite le 1° interrupteur du switch-dip SW5, vérifier que D6 clignote en opposition de phase par rapport à D1. Remettre l'interrupteur dans son état initial.

Même manipulation avec 2° interrupteur /D7, 3° interrupteur/D8, 4° interrupteur/D9.

Faire basculer vers la droite les 5°, 6°, 7° interrupteurs (simultanément) et vérifier que D10 clignote opposition de phase par rapport à D1. Remettre l'interrupteur dans son état initial.

Q.1) On demande d'écrire la description VHDL correspondant à la procédure de test, de valider le fonctionnement par simulation puis par programmation du PAL. Faire vérifier le fonctionnement final. On s'inspirera fortement de l'exemple vu précédemment.

IV.3 Décodeur BCD-7segments

On donne la description d'un décodeur BCD-7segments

```
Library ieee ;
USE ieee.std_logic_1164.all ;
USE work.std_arith.all ;

entity DEC7SEG4 is
    Port ( DEC : in std_logic_vector(3 downto 0);
          SEG :out std_logic_vector(0 to 6) );
End DEC7SEG4 ;

architecture ARCH_DEC7SEG4 of DEC7SEG4 is
begin
    SEG <= "1111110" WHEN DEC=0
    ELSE "0110000" WHEN DEC=1
    ELSE "1101101" WHEN DEC=2
    ELSE "1111001" WHEN DEC=3
    ELSE "0110011" WHEN DEC=4
    ELSE "1011011" WHEN DEC=5
    ELSE "1011111" WHEN DEC=6
    ELSE "1110000" WHEN DEC=7
    ELSE "1111111" WHEN DEC=8
    ELSE "1111011" WHEN DEC=9
    ELSE "-----";
```

Q.1) Comment sont déclarés les variables d'entrées et de sorties ?

Q.2) Expliciter le format de DEC et SEG. Où sont les poids forts, les poids faibles pour DEC? Ou sont a,b,c..g pour SEG ?

Q.3) Quelle différence faites vous entre <= et = ?

Q.4) Que se passe t'il pour les valeurs de DEC>9 ? A votre avis comment le compilateur va décider ?

Q.5) Ecrire le « attribute pin numbers » de ce fichier pour le faire fonctionner sur la maquette. Le mot d'entrée sera donné par les 4 interrupteurs de la maquette. Le poids faible en haut.

Q.6) Valider le fonctionnement par simulation puis par programmation du PAL. Faire vérifier le fonctionnement final. On veillera à positionner correctement les cavaliers sur la maquette.

Q.7) Modifier le fichier pour afficher A,b,C,d,E,F pour les valeurs de DEC égales à 10,11,12,13,14,15

V. Exemples de descriptions VHDL en logique séquentielle

V.1 Le compteur 4 bits

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;
USE work.std_arith.all ;

entity COMPT_4 is
-- H = Horloge
-- R = Reset
    port( H,R :in std_logic ;
          Q : buffer std_logic_vector(3 downto 0) ) ;
end COMPT_4 ;

architecture ARCH_COMPT_4 of COMPT_4 is
begin
    process(H,R)
    begin
        if R='1'
            then Q <= "0000" ;
        elsif (H'event and H='1')
            then Q <= Q+1 ;
        end if ;
    end process ;
end ARCH_COMPT_4 ;
```

Remarque : Le symbole - - indique un commentaire.

Q.1) Identifier chaque partie de la description.

Q.2) Quel est le type de la variable Q ?

Remarque : Les entrées pures sont nommées **IN**, les sorties pures **OUT**. **BUFFER** est réservé aux variables qui sont des sorties rebouclées. Ici Q est une sortie (Q représente les 4 sorties du compteur), mais en même temps Q est relu par le système puisqu'à chaque front d'horloge $Q \leq Q+1$.

Les « process »

Un « process » est une suite d'instructions séquentielles. Lorsque toutes les instructions à l'intérieur d'un « process » sont exécutées, le « process » recommence. L'affectation des sorties ne se fait qu'à la fin du « process ».

Toutes les entrées qui modifient les sorties du process doivent être déclarées dans la **liste de sensibilité**. L'exécution du process reprend lorsqu'une des variables de cette liste de sensibilité change d'état.

Q.3) Pour le process étudié, donner la liste de sensibilité.

Q.4) Que fait-on si $R='0'$, si $R='1'$

Q.5) Comment traduire « if (H'event and H='1') »?

Q.6) Sur ce compteur, quels sont les entrées synchrones et les entrées asynchrones ?

Q.7) Dessiner le schéma « IEC » équivalent de ce compteur.

Q.8) A l'aide de l'outil Warp, saisir le fichier, compiler et simuler la structure. On se reportera à la documentation des logiciels fournis. On imprimera tous les documents intéressants.

Q.9) Si le temps le permet, programmer le circuit Pal, l'installer sur la maquette et valider le fonctionnement.

V.2 Une machine à états : le portail de garage

Le système portail de garage a été étudié comme exemple dans le cours sur la synthèse de système séquentiels synchrones. On donne la description VHDL de ce système. Ce fichier a été généré par un programme à partir du diagramme d'états.

```

library IEEE;
use IEEE.std_logic_1164.all;

entity garage2 is
  port (CLK: in STD_LOGIC;
        PF: in STD_LOGIC;
        PO: in STD_LOGIC;
        TC: in STD_LOGIC;
        F: out STD_LOGIC;
        O: out STD_LOGIC);
end;

architecture garage2_arch of garage2 is

  -- SYMBOLIC ENCODED state machine: Sreg0
  type Sreg0_type is (S0, S1, S2, S3);
  signal Sreg0: Sreg0_type;

begin
  --concurrent signal assignments
  --diagram ACTIONS;

  Sreg0_machine: process (CLK)
  begin
    if CLK'event and CLK = '1' then
      case Sreg0 is
        when S0 =>
          if TC='1' then
            Sreg0 <= S1;
          end if;
        when S1 =>
          if PO='1' then
            Sreg0 <= S2;
          end if;
        when S2 =>
          if TC='1' then
            Sreg0 <= S3;
          end if;
        when S3 =>
          if PF='1' then
            Sreg0 <= S0;
          end if;
        when others =>
          null;
      end case;
    end if;
  end process;

  -- signal assignment statements for combinatorial outputs
  O_assignment:
  O <= '0' when (Sreg0 = S0) else
        '1' when (Sreg0 = S1) else
        '0' when (Sreg0 = S2) else
        '0' when (Sreg0 = S3) else
        '0';

  F_assignment:
  F <= '0' when (Sreg0 = S0) else
        '0' when (Sreg0 = S1) else
        '0' when (Sreg0 = S2) else
        '1' when (Sreg0 = S3) else
        '1';

end garage2_arch;

```

Q.1) Dessiner le schéma fonctionnel de la structure.

Q.2) Il y a une variable interne. Quel est son nom ? Quel est son type ? Sur combien de bits (de bascules) sera t'elle codée ? Comment appelle t'on cette variable dans une machine à états ?

Q.3) Dans cette structure, on a donné un nom au process. Quel est ce nom ?

Q.4) A partir de la description VHDL, représenter le diagramme d'état du système. Est-il conforme à ce qui avait été vu précédemment.