**VHDL – DOC**

**Définition des entré et des sortie  de cette façon**

|  |
| --- |
| -- On déclare tout d’abord la bibliothèque IEEE qui contient  -- la définition du type std\_logic (entre autres).  library IEEE;  -- On indique que l’on souhaite utiliser tous les objets  -- définis dans le package STD\_LOGIC\_1164. Cela permet d’utiliser  -- le type std\_logic pour les signaux (0,1,'Z', etc.).  use IEEE.STD\_LOGIC\_1164.ALL;  -- Début de la définition de l’entité (entity)  entity TP07\_V1 is  -- La section Port définit les entrées et sorties de cette entité.  Port (  -- Entrées (in) de type std\_logic  A : in STD\_LOGIC; -- Signal d'entrée A  B : in STD\_LOGIC; -- Signal d'entrée B  C : in STD\_LOGIC; -- Signal d'entrée C    -- Sorties (out) de type std\_logic  LedA : out STD\_LOGIC; -- Sortie correspondant à A  LedB : out STD\_LOGIC; -- Sortie correspondant à B  LedC : out STD\_LOGIC -- Sortie correspondant à C  );  end TP07\_V1; |

**Architecture**

L’architecture décrit comment le circuit fonctionne en interne. C’est là qu’on écrit le comportement qui dit : « La sortie X dépend de A et B selon telle règle ». On y écrit la logqiue.

**Affectation**

En VHDL, on peut écrire les affectations (assignations) de signaux de plusieurs façons. Voici la différence entre :

1. Affectation simple
2. Affectation conditionnelle
3. Affectation par sélection

**Affectation simple**

L’affectation simple (souvent utilisée) dit juste : « ma sortie prend la valeur de telle expression ». Par exemple :

|  |
| --- |
| LedA <= A and (B nor C);  LedB <= B and (A nor C);  LedC <= C and (B nor A); |

**Affectation conditionnelle**

Ici, on utilise la forme [... <= ... when ... else ....] On pose une ou plusieurs conditions et on indique la valeur à donner en fonction de ces conditions.

Exemple :

|  |
| --- |
| LedA <= '1' when (A = '1') else '0'; |

Si A vaut '1', alors LedA prend '1'; sinon, LedA prend '0'.

On peut chaîner plusieurs conditions pour gérer plusieurs cas.

**Affectation par sélection (with ... select)**

Cette forme de code permet de faire une sorte de « commutateur » (switch) sur une variable ou un signal, puis d’assigner différentes valeurs.

Exemple :

|  |
| --- |
| with A & B & C select      LedA <= '1' when "100",           '0' when others; |