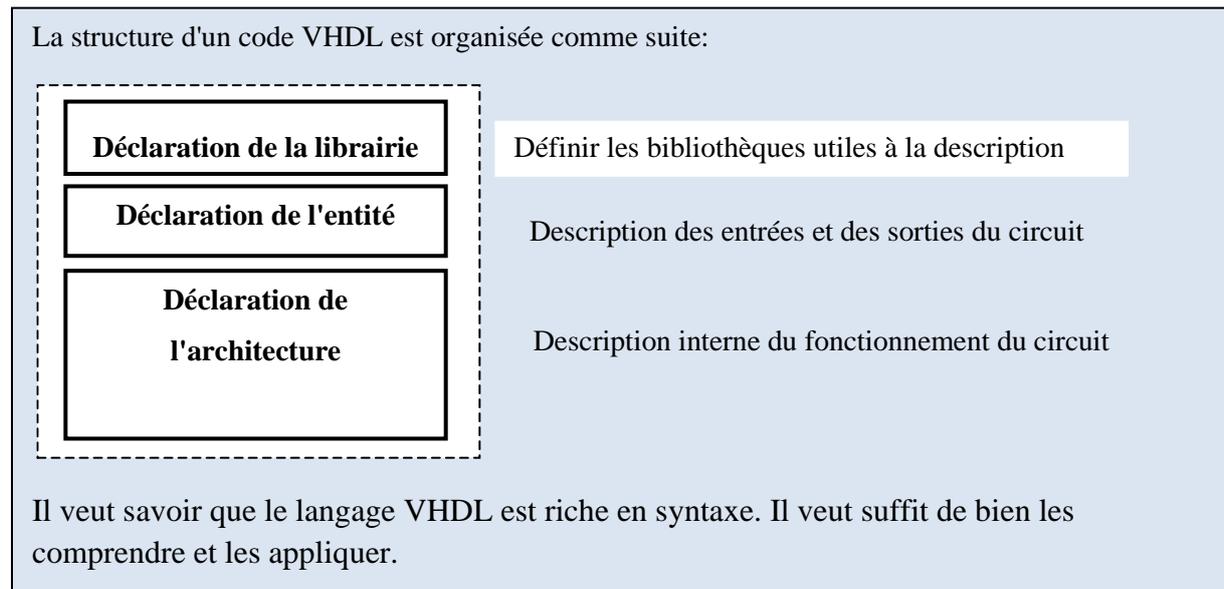


UEF 12.12 - DSP et FPGA  
**Exercice 1 corrigé - TD N°01****Exercice N°01 :**

1 - Codes VHDL des portes logiques suivantes: ET, OU et XOR.

**Rappel :**

Nous allons découvrir ensemble comment interpréter ces syntaxes

**Exemple 1:** Code VHDL de la porte logique ET (AND)**-- :** Ces deux lignes signifie un commentaire en VHDL**-- Librairie**Library ieee; **-- Le langage VHDL a été normalisé par l'institut IEEE. Elle regroupe toute les fonctionnalités du VHDL.**

Use ieee.std\_logic\_1164.all;

**-- std\_logic\_1164** est le nom du package. Ce qui veut dire nous allons utilisé seulement **les systèmes logiques à valeurs multiples** de la bibliothèque ieee.**-- all** (tout en français): nous allons utilisé **toute** la partie du std\_logic\_1164**-- Entité**

Entity Exercice1 is

Port (

a, b : in std\_logic;

s : out std\_logic);

```

End Exercice1;
-- Architecture
Architecture Description of Exercice1 is
Begin
    s<=a and b;    -- <= signe d'affectation, and est un opérateur logique
                  -- VHDL est un langage de description matérielle (Rappel).
                  -- Description de la porte AND.
                  -- En VHDL, le terme description qui est utilisé plutôt que
                  -- programmation.
End Description;

```

**Exemple 2 :** Code VHDL de la porte logique OR (OU)

```

-- Librairie
Library ieee;
Use ieee.std_logic_1164.all;

-- Entité
Entity Exercice1 is
Port (
    a, b    : in    std_logic;
    s      : out   std_logic);
End Exercice1;

-- Architecture
Architecture Description of Exercice1 is
Begin
    s<=a OR b;    -- Description de la porte OR .
                  -- En VHDL, écrire un opérateur en Majuscule OR ou en
                  -- minuscule or n'a pas d'importance.
End Description;

```

**Exemple 3 :** Code VHDL de la porte logique XOR

```

-- Librairie
Library ieee;
Use ieee.std_logic_1164.all;

-- Entité
Entity Exercice1 is
Port (
    a, b    : in    std_logic;
    s      : out   std_logic);

```

```

End Exercice1;

-- Architecture
Architecture Description of Exercice1 is
Begin
    s<=a XOR b; -- Description de la porte XOR .
End Description;

```

Jusqu'à ici, nous avons appris à écrire un code VHDL ou à décrire des portes logiques. Vous trouverez les signes des opérateurs logiques ainsi que les différents opérateurs en VHDL dans le support du cours.

## 2 - Code VHDL du circuit représenté dans la Figure ci-dessous

Nous allons passer à la description d'un circuit numérique.

Comme vous pouvez remarquer, le circuit est composé de deux portes logiques xor et and.

```

-- Librairie
Library ieee;
Use ieee.std_logic_1164.all;

```

```

-- Entité
Entity Circuit is
Port (

```

```

    A, B      : in  std_logic; -- Si les entrées sont de même type de signal,
                                vous pouvez les écrire sur la même ligne.

```

```

-- Sinon vous les écrivez séparément
    A        : in  std_logic;
    B        : in  std_logic;
-- Sorties sout et cout
    sout, cout : out std_logic);

```

```
End Circuit;
```

```

-- Architecture
Architecture Description of Circuit is
Begin

```

```

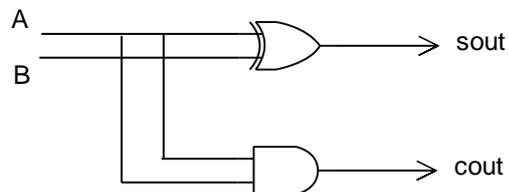
    sout<=A xor B; -- Description de la porte XOR.
    cout<=A and B; -- Description de la porte AND.

```

```

-- Pareille si vous écrivez
    cout<=A and B;

```



```
sout<=A xor B;  
End Description;
```

### Rappel

Les instructions du langage VHDL sont *concurrentes*, ce qui signifie l'ordre de description de ces instructions n'est pas important. Contrairement aux langages informatiques, exemple, le C et JAVA sont des instructions *séquentielles*.

Décrire la sortie **cout** au premier ensuite **sout** ou le contraire n'est pas important.