

TD AEV

En VHDL le comportement du composant est décrit dans une *Architecture* contenant la partie interne du module. L'architecture doit être nommée, ici "*arc*", ce qui permet de définir plusieurs architectures pour la même entité. Le code est défini entre les mots-clef begin et end

l'instruction <= est une instruction d'affectation qui se dit aussi "*reçoit*" permettant d'exprimer la causalité du calcul entre les entrées et la sortie combinatoire. Le mot-clef = existe aussi mais est utilisé comme opérateur de comparaison entre 2 variables.

Les expressions logiques utilisables sont propres à chaque type de variable. Pour le type prédéfini bit, les opérateurs utilisables sont les mêmes qu'en C :

- le ET est noté and
- le OU est noté or
- le XOR est noté xor
- le NON est noté not

Pour une porte NOR3, on aurait eu :
s <= not(e1 or e2 or e3);

Exercice 1

Voici un petit programme :

Remplissez les blancs avec le choix proposé.

```
entity exo1 is
port (clk : in bit;
a : _____ bit_vector ( _____ downto 0),
s : out bit_vector ( _____ downto 0);
end exo1 ;
```

```
architecture aexo1 of exo1 is
begin
with _____ select
s <= _____ when "01001",
    "011" when "10010",
    _____ when others ;
end aexo1;
```

Table de vérité devant correspondre au programme à gauche	
a	s
01001	101
10010	011
10011	001
10100	001
10101	001
10110	001

Exercice 2

Soit la description VHDL suivante :

TD AEV

```
-- fonction.vhd

library ieee;
use ieee.std_logic_1164.all;

entity fct is
    port (a, b : in std_logic;
          s : out std_logic);
end fct;

architecture archi_fct of fct is
begin
    s <= '1' when a=b else '0';
end archi_fct;
```

1. Que réalise cette fonction ?

Dans le même projet, on écrit la description VHDL suivante:

```
-- function.vhd

library ieee;
use ieee.std_logic_1164.all;

entity circuit is
    port (e1, e2, e3, e4 : in std_logic;
          s : out std_logic);
end circuit;

architecture archi_circuit of circuit is

    signal s1, s2, s3 : std_logic;

    component fct
        port (a, b : in std_logic;
              s : out std_logic);
    end component;

begin
    cmp1 : fct port map (a=>e1, b=>e2, s=>s1);
    cmp2 : fct port map (a=>e3, b=>e4, s=>s2);
    cmp3 : fct port map (a=>e1, b=>e3, s=>s3);

    s <= s1 and s2 and s3;

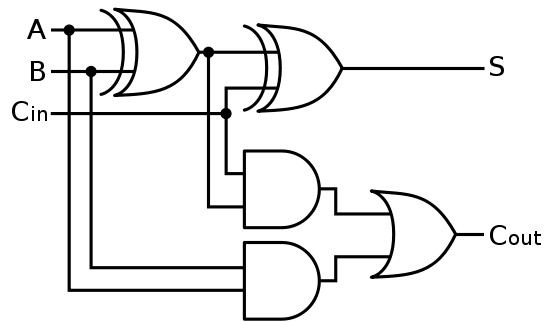
end archi_circuit;
```

2. Faites une analyse structurelle du circuit réalisé.
3. Quelle fonction est décrite par ce circuit ?

Exercice 3 : Additionneur complet 1 bit

Un additionneur complet 1 bit (voir figure ci-dessous) a trois entrées (a, b, et cin la retenue entrante), et deux sorties (le bit de somme s et le bit de retenue sortante cout).

TD AEV



- 1 – Donner le code VHDL correspondant
- 2 - Réutiliser ce composant pour concevoir un additionneur 4 bits. Donner le code VHDL

Exercice 4

Soit le code VHDL suivant

```
ENTITY exo4 IS
    PORT
    (
        a : IN STD_LOGIC;
        b : IN STD_LOGIC;
        c : IN STD_LOGIC;
        d : IN STD_LOGIC;
        adr : IN STD_LOGIC_VECTOR (1 downto 0);
        s : OUT STD_LOGIC
    );
END exo4;

ARCHITECTURE Archi1exo4 OF exo4 IS
    BEGIN
        s <= ( a AND NOT adr(1) AND NOT adr(0) )
            OR ( b AND NOT adr(1) AND   adr(0) )
            OR ( c AND   adr(1) AND NOT adr(0) )
            OR ( d AND   adr(1) AND   adr(0) );
    END Archi1exo4;
```

- 1 – Dessiner le circuit
- 2 – Que fait-il?
- 3 – Ecrire l'architecture en utilisant les mots clés :
 - WITH SELECT
 - WHEN ELSE