

Description par flot de données

- Le modèle d'un circuit par flot de données décrit sa fonction sans nécessairement définir sa structure.
- Les valeurs des signaux et des ports du circuits sont établies par des affectations concurrentes de valeurs.
- Trois types d'énoncés concurrents:
 - Avec des opérateurs logiques:
and, or, nand, nor, xor, xnor, not
 - Selection (équivalent à un énoncé case):
With-select
 - Conditionnel (équivalent à un if else):
When-else

Avec des opérateurs logiques

```
library ieee;
use ieee.std_logic_1164.all;

entity combinatoire1 is
    port (
        A, B, C : in std_logic;
        F : out std_logic
    );
end combinatoire1;

architecture flotdedonnees1 of combinatoire1 is
begin
    F <= not(A and (B xor (not C)));
end flotdedonnees1;
```

Sélection

architecture flotdedonnees2 of combinatoire1 is
signal entree : std_logic_vector (2 downto 0);
begin

entree <= (A,B,C);

with entree **select**

F <=

'1' when "000" ;

'1' when "001" ;

'1' when "010" ;

'1' when "011" ;

.....

.....

'0' when others ;

end flotdedonnees2;

A	B	C	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Enoncé conditionnel

architecture flotdedonnees3 of combinatoire1 is
begin

 F <= '1' **when** (A = '0' or B/= C) **else** '0';
end flotdedonnees3;

Description comportementale

- Une description comportementale utilise des énoncés similaires à ceux d'un langage procédural comme C ou java, incluant les structures de contrôle (condition, répétition).
- Les descriptions comportementales en VHDL se font à l'aide de l'énoncé " **process** " à l'intérieur d'une architecture.
- Un processus (process) décrit une partie du circuit qui s'exécute de façon concurrente à d'autres processus et à des affectations concurrentes de signaux.

- Chaque composant interne du système est un processus (process) de l'architecture.
- Une architecture est un ensemble de processus.
- Les processus s'exécutent en parallèle.
- Les processus de l'architecture sont interconnectés par le biais des signaux (signal).

Bibliothèque IEEE {
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

Entité {
entity exemple is
port (a, b : in std_logic_vector (3 downto 0);
op, clk, reset : in std_logic;
c : out std_logic_vector(3 downto 0)
);
end exemple

architecture test of exemple is
signal moinsb, opb, aopb : std_logic_vector(3 downto 0);
begin

moinsb <= not b + "0001";

opb <= b when op='0'
else moinsb;

aopb <= a + opb;

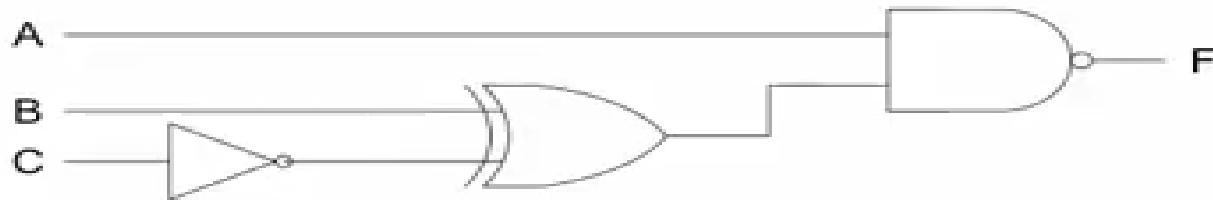
process (reset, clk)
begin
if reset='0'
then c <= "0000";
else if (clk'event and clk='1')
then c <= aopb;
end if;
end if;
end process;

end test;

Process implicite

Process explicite

Exemple 1 de description comportementale



```
library ieee;  
use ieee.std_logic_1164.all;  
  
entity combinatoire1 is  
    port (  
        A, B, C : in std_logic;  
        F : out std_logic  
    );  
end combinatoire1;
```


architecture comportementale1 of combinatoire1 is

begin

process (A, B, C) – (A,B,C) : liste de sensibilité (sensitivity list)

begin

F <= not(A and (B xor not (C)));

end process;

end comportementale1;

Similaire à une description par flot de données avec opérateurs logiques

architecture comportementale2 of combinatoire1 is
begin

 process (A, B, C)

 begin

 if A='0' or B /= C then

 F <= '1';

 else

 F <= '0';

 end if;

 end process;

end comportementale2;

Similaire à une description par flot de données avec énoncé conditionnel.