Exemple **Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**-1 – circuit combinatoire décrit en VHDL

Exemple1 : Circuit1

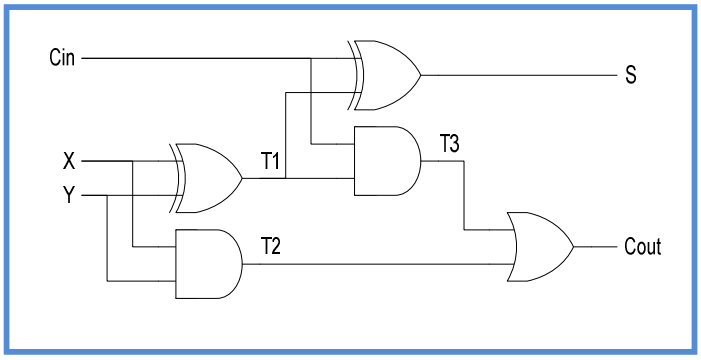


Figure 1: logigramme de l'exemple1

```
library IEEE;
use IEEE.std_logic_1164.all;
entity exemple1 is
port(
cin, X,Y :in std_logic;
S,cout :out std_logic
);
end exemple1;

architecture arch1 of exemple1 is
signal T1, T2, T3 : std_logic;
begin
T1<=X xor Y;
T2<=X and Y;
T3<=cin and T1;
S<=T1 xor cin;
cout<= T3 or T2;
end arch1;
```

Code VHDL de l'exemple1

Figure2 : Simulation Waveform de l'exemple1

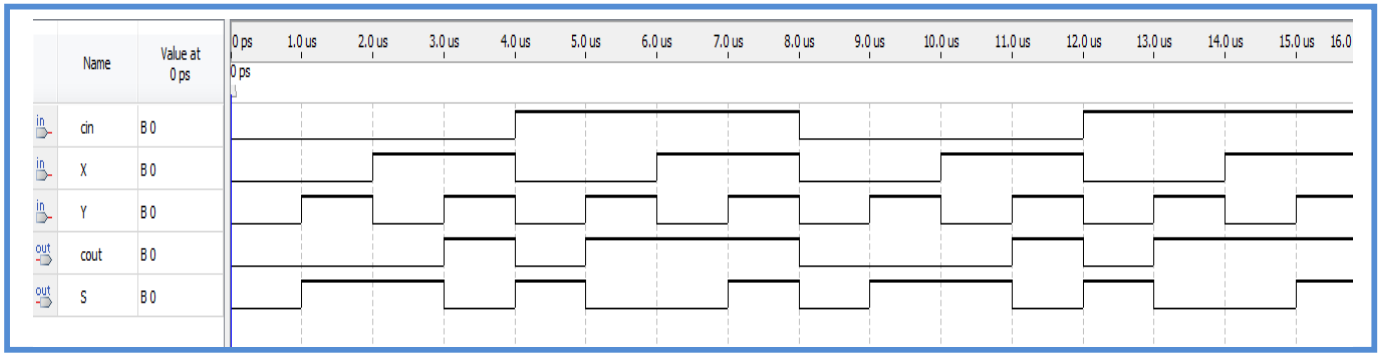


Figure2 : Simulation Waveform de l'exemple1

Exemple2 : Circuit2

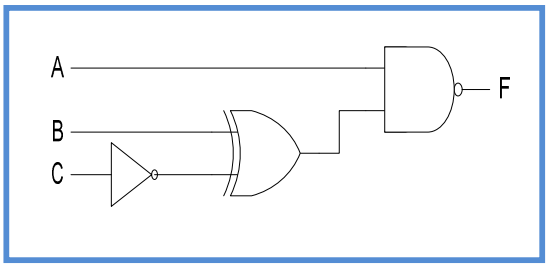


Figure 3: logigramme de l'exemple2

```
library ieee;
use ieee.std_logic_1164.all;
entity exemple2 is
port (
A,B,C : in std_logic;
F : out std_logic
);
end exemple2;

architecture arch2 of exemple2 is
signal T1,T2: std_logic;
begin
T1 <= not(C);
T2 <= T1 xor B;
F <= T2 nand A;
end arch2;
```

Code VHDL de l'exemple2

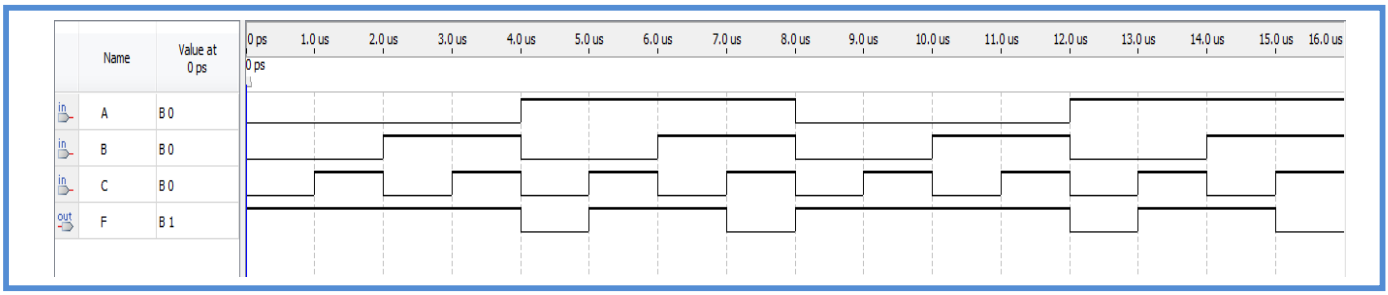
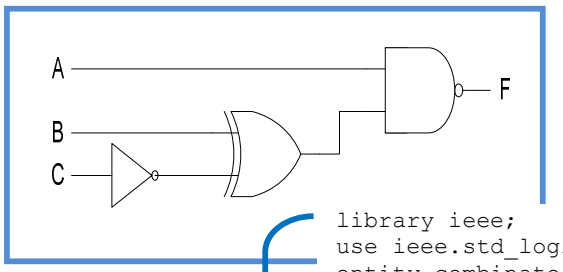


Figure4 : Simulation Waveform de l'exemple2




```
library ieee;
use ieee.std_logic_1164.all;
entity combinatoire1 is
    port (
        A : in std_logic;
        B : in std_logic;
        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;
```

. **Exemple** Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**-2 – description par flot de**

Exemple3 : Table de vérité

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

```
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 "100",
            '1' when "101",
            '1' when "110",
            '0' when "111",
            '0' when others;
end flotDeDonnees2;
```

Exemple Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**-3 – assignation choisie**

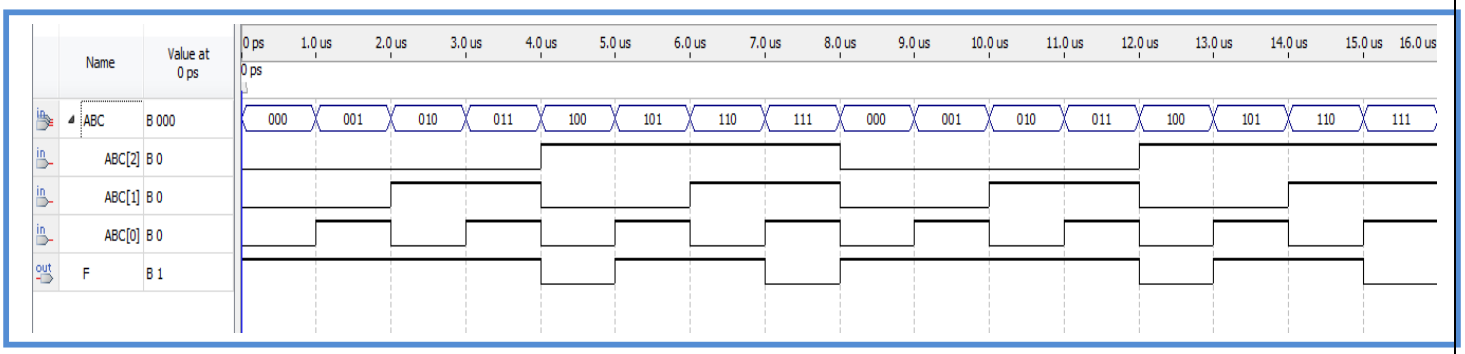
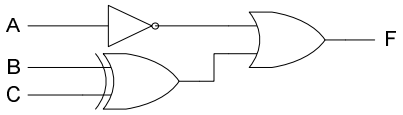


Figure6 : Simulation Waveform de l'exemple3

Exemple 2 –5.

```
architecture flotDeDonnees3 of combinatoire1 is
begin
    F <= '1' when (A = '0' or B /= C) else '0';
end flotDeDonnees3;
```



Exemple Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**-5 – assignation conditionnelle**

Exemple5 : Porte AND à 4 entrées

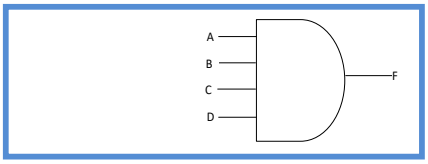
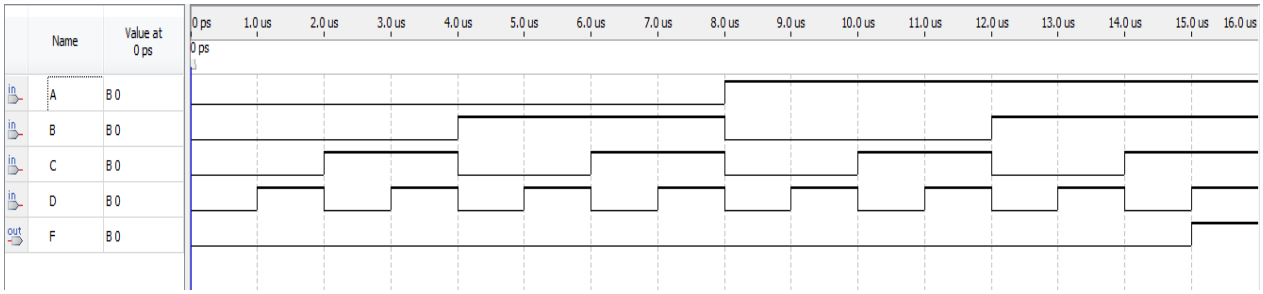


Figure 9: logigramme de l'exemple5

```
library ieee;
use ieee.std_logic_1164.all;
entity exemple5 is
port (
A,B,C,D : in std_logic;
F: out std_logic
);
end exemple5;

architecture arch5 of exemple5 is
begin
F <= (A and B and C and D);
end arch5;
```

Code VHDL de l'exemple5



Exemple Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**-4 – deux façons de décrire une porte ET à quatre entrées**

```
library ieee;
use ieee.std_logic_1164.all;
entity exemple6 is
generic( n: integer:=4);
port ( a : in std_logic_vector( n downto 1);
c : out std_logic);
end entity;

Architecture arch6 of exemple6 is
Signal s : std_logic_vector((n+1) downto 1);
begin
s(1)<='1';
g1: for i in 1 to n generate
s(i+1)<=a(i) and s(i);
end generate;
c<=s(n+1);
end arch6;
```

Code VHDL

Exemple Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**-5 – une porte ET à nombre d'entrées variable**

.

L'énoncé `generic` est un paramètre pour indiquer le nombre d'entrées de la porte ET..

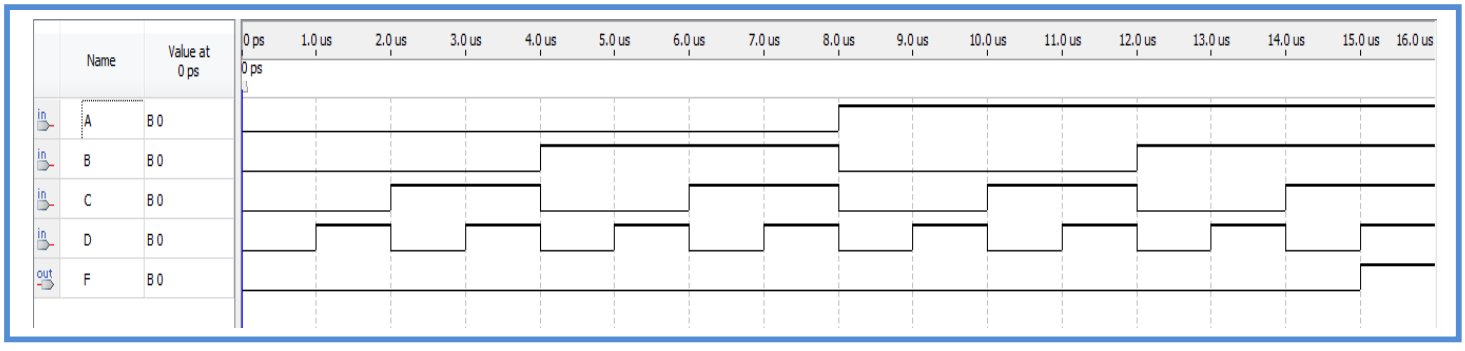
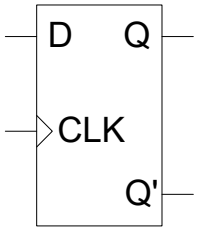


Figure10 : Simulation Waveform de l'exemple5

L'Exemple **Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**-6 démontre l'utilisation de l'attribut `event` sur le signal `CLK`, dénoté par `CLK'event`.

```
library ieee;
use ieee.std_logic_1164.all;
entity basculeD is
   port (
      CLK : in STD_LOGIC;        -- horloge
      D : in STD_LOGIC;          -- entrée
      Q : out STD_LOGIC          -- sortie
   );
end basculeD;

architecture basculeD of basculeD is
begin
   process(CLK) is
   begin
      if (CLK = '1' and CLK'event) then
         Q <= D;
      end if;
   end process;
end basculeD;
```



Exemple Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**-6 – bascule D**

```
library IEEE;
use IEEE.std_logic_1164.all;
entity basculeDR is
   port (
      reset : in STD_LOGIC;      -- signal de remise à zéro
      CLK : in STD_LOGIC;        -- signal d'horloge
      D : in STD_LOGIC;          -- entrée
      Q : out STD_LOGIC          -- sortie
   );
end basculeDR;
architecture basculeDRasynch of basculeDR is
begin
   process(CLK, reset) is
   begin
      if (reset = '0') then
         Q <= '0';
      elsif (rising_edge(CLK)) then
         Q <= D;
      end if;
   end process;
end basculeDRasynch;
architecture basculeDRsynch of basculeDR is
begin
   process(CLK, reset) is
   begin
      if (rising_edge(CLK)) then
         if (reset = '0') then
            Q <= '0';
         else
            Q <= D;
         end if;
      end if;
   end process;
end basculeDRsynch;
```

Exemple Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**-7 – bascule D avec initialisation asynchrone ou synchrone**

1.1.2 Loquet D

Le loquet D (*D–latch*) a deux entrées : un signal de données *D* et un signal de contrôle *G* (*Gate*).Dans l'Exemple **Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**-8, un processus a dans sa liste de sensitivité le signal de contrôle *G* et le signal de donnée *D*.Une simple modification permet de choisir la valeur '0' comme valeur active pour le signal *G*, si désiré.

```
library ieee;
use ieee.std_logic_1164.all;
entity loquetD is
    port (
        G : in STD_LOGIC;    -- contrôle
        D : in STD_LOGIC;    -- donnée
        Q : out STD_LOGIC
    );
end loquetD;
architecture loquetD of loquetD is
begin
    process(G, D) is
    begin
        if (G = '1') then
            Q <= D;
--      else   -- implicite, infère élément à mémoire ne pas changer Q
        end if;
    end process;
end loquetD;
```

Exemple Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**-8 – loquet D**

Exemple Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**-9 – Fonction ET**

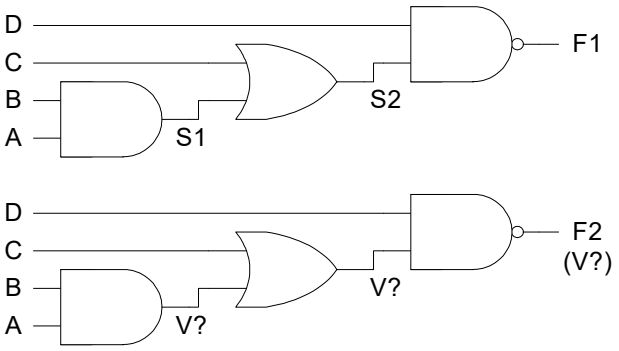
```
function porteET(V: std_logic_vector) return std_logic is
variable resultat : std_logic := '1';
begin
    for k in V'range loop
        resultat := resultat and V(k);
    end loop;
    return resultat;
end;
```

Exemple Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**-10 – définition d'une fonction**

Exercice 3

Exemple 11 – signaux et variables pour la modélisation

```
library ieee;
use IEEE.STD_LOGIC_1164.ALL;
entity demoSignalVariable is
    port (
        A, B, C, D: in std_logic;
        F1, F2 : out std_logic
    );
end demoSignalVariable;
architecture demo of demoSignalVariable is
signal S1, S2 : std_logic;
begin
    S1 <= A and B;
    S2 <= S1 or C;
    F1 <= S2 nand D;
    process(A, B, C, D)
    variable V : std_logic;
    begin
        V := A and B;
        V := V or C;
        V := V nand D;
        F2 <= V;
    end process;
end demo;
```



Exemple Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**-12 – signaux et variables pour la modélisation**

Exemple 5 1 – multiplexeur 2:1

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity mux21 is
    port(D0, D1, S : in STD_LOGIC; F : out STD_LOGIC);
end mux21;
architecture flotDeDonnees of mux21 is
begin
   with S select
   F <= D0 when '0', D1 when others;
end flotDeDonnees;
```

Exemple Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**-13 – multiplexeur 2:1**

Exemple 5 2 – description comportementale d'un multiplexeur général

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
entity mux is
   generic (
      n : positive := 3 -- nombre de signaux de contrôle
   );
   port (
      D : in std_logic_vector(2 ** n - 1 downto 0);
      S: in unsigned(n - 1 downto 0);
      F : out std_logic
   );
end mux;
architecture comportementale of mux is
begin
   process (D, S)
   begin
      F <= D(to_integer(S));
   end process;
end comportementale;
```

Exemple Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**-14 – description comportementale d'un multiplexeur général**

Exemple 5 3 – décodeur 3:8

Un décodeur a n signaux d'entrée et 2^n signaux de sortie active un signal spécifique correspondant à un code.

#	A_2	A_1	A_0	F_7	F_6	F_5	F_4	F_3	F_2	F_1	F_0
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
2	0	1	0	0	0	0	0	0	1	0	0
3	0	1	1	0	0	0	0	1	0	0	0
4	1	0	0	0	0	0	1	0	0	0	0
5	1	0	1	0	0	1	0	0	0	0	0
6	1	1	0	0	1	0	0	0	0	0	0
7	1	1	1	1	0	0	0	0	0	0	0

Tableau Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**-1 – table de vérité d'un décodeur 3:8**

```
library ieee;
use ieee.std_logic_1164.all;
entity decodeur38 is
    port(
        A : in std_logic_vector(2 downto 0);
        F: out std_logic_vector(7 downto 0)
    );
end decodeur38;
architecture flotDeDonnees of decodeur38 is
begin
    with A select F <=
        "00000001" when "000",
        "00000010" when "001",
        "00000100" when "010",
        "00001000" when "011",
        "00010000" when "100",
        "00100000" when "101",
        "01000000" when "110",
        "10000000" when "111",
        (others => 'X') when others;
end flotDeDonnees;
```

Exemple Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**-15 – décodeur 3:8**

Exemple 5 4 – décodeur général

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
entity decodeur is
    generic (
        n : positive := 3; -- nombre de signaux d'entrée
        valeurActive : std_logic := '1'
    );
    port(
        A : in std_logic_vector(n - 1 downto 0);
        F: out std_logic_vector(2 ** n - 1 downto 0)
    );
end decodeur;
architecture comportementale of decodeur is
begin
    process(A)
    begin
        F <= (others => not(valeurActive));
        F(to_integer(unsigned(A))) <= valeurActive;
    end process;
end comportementale;
```

Exemple Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**-16 – décodeur général**

Exemple 5 5 – encodeur à priorité

Un encodeur identifie un signal actif parmi des signaux Tableau **Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**-2, et produit un code associé **Erreur ! Source du renvoi introuvable.**

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_2	A_1	A_0	V
0	0	0	0	0	0	0	0	-	-	-	0
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	1	-	0	0	1	1
0	0	0	0	0	1	-	-	0	1	0	1
0	0	0	0	1	-	-	-	0	1	1	1
0	0	0	1	-	-	-	-	1	0	0	1
0	0	1	-	-	-	-	-	1	0	1	1
0	1	-	-	-	-	-	-	1	1	0	1
1	-	-	-	-	-	-	-	1	1	1	1

Tableau Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**-2 – table de vérité d'un encodeur à priorité 8:3**

Exemple 5 6 – tampon à trois états

On utilise la valeur 'Z' du type `std_logic` qui correspond à un état de haute impédance pour l'état trois.

```
library ieee;
use ieee.std_logic_1164.all;
entity tampon3etats is
    port(
        I : in std_logic; -- signal d'entrée
        S : in std_logic; -- signal de contrôle
        O : out std_logic -- signal de sortie
    );
end tampon3etats;
architecture flotdonnees of tampon3etats is
begin
    O <= I when (S = '1') else 'Z';
end flotdonnees;
```

Exemple Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**-17 – tampon à trois états**

Exemple 5 7 – registre à chargement parallèle

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity registre is
    generic (     W : integer := 8
    );
    port(
        reset : in STD_LOGIC;
        CLK : in STD_LOGIC;
        charge : in STD_LOGIC;
        D : in STD_LOGIC_VECTOR(W - 1 downto 0);
        Q : out STD_LOGIC_VECTOR(W - 1 downto 0)
    );
end registre;
architecture arch of registre is
begin
    process (CLK, reset)
    begin
        if reset='0' then
            Q <= (others => '0');
         elsif CLK='1' and CLK'event then
            if charge='1' then
                Q <= D;
            end if;
        end if;
    end process;
end arch;
```

Exemple Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**-18 – registre à chargement parallèle**

Exemple 5 8 – registre à décalage

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity registreadecallage is
    generic (     W : integer := 8 -- nombre de bits du registre
    );
    port(
        reset : in STD_LOGIC;
        CLK : in STD_LOGIC;
        mode : in STD_LOGIC_VECTOR(1 downto 0); -- mode
        entreeSerie : in STD_LOGIC; -- entree serielle
        D : in STD_LOGIC_VECTOR(W - 1 downto 0);
        Q : out STD_LOGIC_VECTOR(W - 1 downto 0)
    );
end registreadecallage;
architecture arch of registreadecallage is
begin
    process (CLK, reset)
    variable Qinterne : STD_LOGIC_VECTOR(W - 1 downto 0);
    begin
        if reset='0' then
            Qinterne := (others => '0');
            Q <= (others => '0');
        elsif CLK='1' and CLK'event then
             case mode is
               when "00" => -- garde
                   Qinterne := Qinterne;
               when "01" => -- charge
                   Qinterne := D;
               when "10" => -- decale gauche
                   Qinterne := Qinterne(W - 2 downto 0) & entreeSerie;
               when "11" => -- decale droite
                   Qinterne := entreeSerie & Qinterne(W - 1 downto 1);
            when others =>
                   Qinterne := Qinterne;
            end case;
            Q <= Qinterne;
        end if;
    end process;
end arch;
```

Exemple Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**-19 – registre à décalage**

Exemple 5 9 – bloc des registres (4 registres) avec commande de chargement et deux ports de sortie A et B

```
type lesRegistres_type is array(0 to Nreg - 1) of signed(Wd - 1 downto 0);
signal lesRegistres : lesRegistres_type;
signal A : signed(Wd - 1 downto 0);
signal choixA : integer range 0 to Nreg - 1;
signal B : signed(Wd - 1 downto 0);
signal choixB : integer range 0 to Nreg - 1;
signal donnee : signed(Wd - 1 downto 0);
signal choixCharge : integer range 0 to Nreg - 1;
signal charge : std_logic;
-- FIN de la partie déclarative de l'architecture
    process (CLK, reset)
    begin
        if rising_edge(CLK) then
            if reset = '1' then
                lesRegistres <= (others => (others => '0'));
            else
                if charge = '1' then
                    lesRegistres(choixCharge) <= donnee;
                end if;
            end if;
        end if;
    end process;
    A <= lesRegistres(choixA);-- signaux de sortie du bloc des registres
    B <= lesRegistres(choixB);
```

Exemple Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**-20 – bloc des registres**

Exemple 5 10 – mémoire des données generic en taille de la mémoire et taille du mot binaire

```
-- dans la partie déclarative de l'architecture

type memoireDonnees_type is array(0 to 2 ** Md - 1) of signed(Wd - 1 downto 0);
signal memoireDonnees : memoireDonnees_type;
signal sortieMemoireDonnees : signed(Wd - 1 downto 0);
signal adresseMemoireDonnees : integer range 0 to 2 ** Md - 1;
signal lectureEcritureN : std_logic;

-- dans le corps de l'architecture

    -- mémoire des données
    process (CLK)
    begin
        if rising_edge(CLK) then
            if lectureEcritureN = '0' then
                memoireDonnees(adresseMemoireDonnees) <= B;
            end if;
        end if;
    end process;
    sortieMemoireDonnees <= memoireDonnees(adresseMemoireDonnees);
```

Exemple Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**-21 – mémoire des données**

Exemple 5 11 – unité arithmétique

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

entity unitearithmetique is
   generic (
      W : positive := 8 -- largeur des opérandes
   );
   port(
      A, B : in signed(W - 1 downto 0); -- les opérandes
      choix : in std_logic_vector(2 downto 0); -- le sélecteur d'opération
      F : out signed(W - 1 downto 0) -- le résultat
   );
end unitearithmetique;

architecture arch of unitearithmetique is
begin
   process(A, B, choix)
   variable t : signed(2 * W - 1 downto 0);
   begin
      t := A * B;
      case to_integer(unsigned(choix)) is
         when 0 => F <= A + B;
         when 1 => F <= A - B;
         when 2 => F <= A + B + 1;
         when 3 => F <= A + 1;
         when 4 => F <= abs(A);
         when 5 => F <= -A;
         when 6 => F <= t(2 * W - 1 downto W);
         when 7 => F <= t(W - 1 downto 0);
         when others => F <= (others => 'X');
      end case;
   end process;
end arch;
```

Exemple Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**-22 – unité arithmétique**

Exemple 5 16 – unité logique pour entrées A et

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
entity unitelogique is
    generic (
        W : positive := 8 -- largeur des opérandes
    );
    port(
        A, B : in std_logic_vector(W - 1 downto 0); -- les opérandes
        choix : in std_logic_vector(2 downto 0); -- le sélecteur d'opération
        F : out std_logic_vector(W - 1 downto 0) -- le résultat
    );
end unitelogique;
architecture arch of unitelogique is
begin
    process(A, B, choix)
    begin
        case to_integer(unsigned(choix)) is
            when 0 => F <= A and B;
            when 1 => F <= A or B;
            when 2 => F <= A nand B;
            when 3 => F <= A nor B;
            when 4 => F <= A xor B;
            when 5 => F <= A xnor B;
            when 6 => F <= not(A);
            when 7 => F <= not(B);
            when others => F <= (others => 'X');
        end case;
    end process;
end arch;
```

Exemple Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**-23 – unité logique**

Exemple 5 17 – comparateur pour entrées A et B

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
entity comparateur is
    generic (    W : positive := 8 -- largeur des opérandes
    );
    port(
        A, B : in signed(W - 1 downto 0);
        eq, neq, gt, lt, ge, le : out std_logic
    );
end comparateur;
architecture arch of comparateur is
begin
eq <= '1' when A = B else '0';
neq <= '1' when A /= B else '0';
gt <= '1' when A > B else '0';
lt <= '1' when A < B else '0';
ge <= '1' when A >= B else '0';
le <= '1' when A <= B else '0';
end arch;
```

Exemple **Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**-24 – comparateur

Exemple 5 18 – compteur synchrone à quatre modes

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.numeric_std.all;
entity compteurSynchrone is
    generic (
        W : integer := 4 -- nombre de bits du compteur
    );
    port(
        reset : in STD_LOGIC;
        CLK : in STD_LOGIC;
        mode : in unsigned(1 downto 0);
        D : in unsigned(W - 1 downto 0);
        Q : out unsigned(W - 1 downto 0)
    );
end compteurSynchrone;
architecture comportementale of compteurSynchrone is
begin
    process (CLK, reset)
    variable Qinterne : unsigned(W - 1 downto 0);
    begin
        if reset='0' then
            Qinterne := (others => '0');
        elsif CLK='1' and CLK'event then
            case mode is
                when "01" => Qinterne := Qinterne + 1;
                when "10" => Qinterne := Qinterne - 1;
                when "11" => Qinterne := D;
                when others => Qinterne := Qinterne;
            end case;
        end if;
        Q <= Qinterne;
    end process;
end comportementale;
```

Exemple Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**-25 – compteur synchrone à quatre modes**