

### EXERCICE 1:

Considérez le système d'équations logiques suivant :

$$P = ABCD + \bar{B}\bar{D} + \bar{A}\bar{C}D + E\bar{F}G$$

$$Q = BC + A\bar{B} + \bar{C}D$$

Considérez le circuit montré à la page 3, qui représente une portion d'un réseau prédéfini programmable (FPGA) général. En utilisant ce circuit, montrez comment implémenter ce système d'équations.

1. Indiquez clairement la position de chaque port, le contenu des tables de correspondance (Look-Up Table – LUT) et le routage des différents signaux.
2. Considérez l'extrait de code VHDL suivant. Donnez son équivalent en matériel à l'aide d'un schéma de portes logiques.

### EXERCICE 2:

Soit le code VHDL suivant.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
entity MAC is
generic (W : positive := 4 );
port (
clk,rst : in std_logic;
entree1, entree2, entree3: in signed(W-1 downto 0);
sortie : out signed(2*W downto 0));
end MAC;
```

a - Donnez le diagramme du circuit MAC correspondant au code VHDL ci-avant. Indiquez la largeur en bits des ports et des signaux intermédiaires en sachant que W est égal à 4. Vous pouvez utiliser des opérations logiques et arithmétiques, la comparaison, des multiplexeurs, décodeurs et encodeurs, éléments mémoire, etc.

```
library ieee;
use ieee.std_logic_1164.all;
entity exo1 is
port (
A,B,C: in std_logic;
F,G,H: out std_logic);
end exo1;
architecture arch of exo1 is
begin
F <= not(A and (B xor not(C)));
G <= '0' when (A = '1' or B /= C) else '1';
with A select H <= B when '0', C when others;
end arch;
```

```
architecture beh of MAC is
begin -- beh
sortie <= (entree1 * entree2) + entree3;
end beh;
```

b – Donnez le diagramme du circuit itératif combinatoire à 3 étages qui cascade 3 MAC en sachant que le port entree3 du premier étage de MAC prendra la valeur zéro. En sachant que le port entree3 du prochain étage prendra la valeur du port sortie du module de l'étage précédent.

c – Donnez le diagramme du circuit itératif séquentiel de 3 étages qui cascade 3 MAC. En sachant que le port entree3 du premier étage de MAC sera prendre la valeur zéro.

**EXERCICE 3:**

Soit l'entité module3 ainsi que le banc d'essai module3\_tb permettant de le vérifier. Les codes VHDL de ces modules sont comme suit:

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

entity module3 is
port (
A, B : in std_logic;
F : out std_logic
);
end module3;
architecture arch of module3 is
signal S1, S2: std_logic;
begin
S1 <= A and B;
S2 <= not(B);
process (S1, S2)
begin
F <= S1 xor S2;
end process;
end arch;
```

```
library IEEE;
use IEEE.std_logic_1164.all;
entity module3TB is
end module3TB;
architecture arch of module3TB is
component module3
port (
A, B : in std_logic;
F : out std_logic
);
end component;

signal A,B,F : std_logic;
begin
UUT : module3 port map (A, B, F);
A <= '0' after 0 ns, '1' after 20 ns;
B <= '1' after 0 ns, '0' after 10 ns,
'1' after 15 ns;
end arch;
```

- Donnez le chronogramme des ports A, B, F et des signaux internes S1 et S2 durant une simulation comprise entre les temps 0 ns jusqu'à 25 ns.

**EXERCICE 4:**

Dans le code VHDL suivant, indiquez les quatre erreurs détectées au moment de la compilation. Pour chaque ligne comportant une erreur, donnez le code corrigé correspondant.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
entity unite_arithmetique is
generic (W : positive := 8);
port(
A, B : in integer(W - 1 downto 0);
choix : in std_logic_vector(2 downto 0);
F : out std_logic_vector(W - 1 downto 0));
end unite_arithmetique;
architecture arch of unite_arithmetique is
begin
process(A, B, choix)
variable t : integer range 0 to 2**W-1;
begin
t <= to_integer(A * B);
case to_integer(unsigned(choix)) is
when 4 => F <= std_logic_vector(abs(A));
when 6 => F <= t;
when others => F <= 'X';
end case;
end process;
end arch;
```

-- largeur des opérandes

-- les operandes de type signé

-- le sélecteur d'opération

-- le résultat

**Exercice 5:**

a. Dessinez le système représenté par le programme VHDL suivant :

```
library IEEE;
use IEEE.std_logic_1164.all;
entity exo5 is
port (a, b, c, d, clk : in std_logic;
q : out std_logic);
end exo5;
```

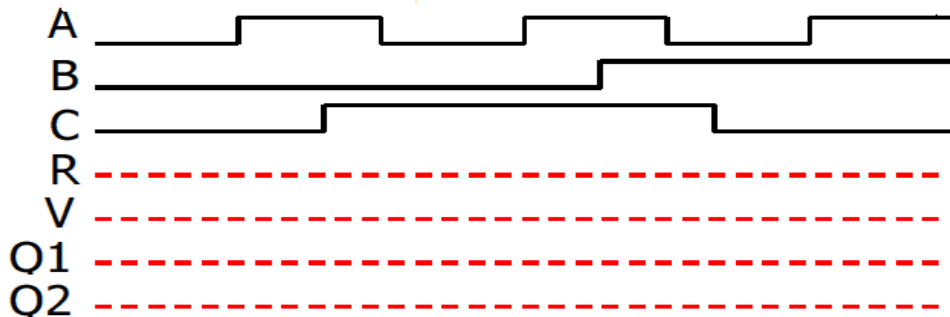
```
architecture bev of exo5 is
signal local : std_logic;
begin
local <= '1' when ((a='1' and b='1') or
c='1')
else '0';
process (clk, local)
begin
if local='1' then
q <= '0';
elsif clk'event and clk='1' then
q <= d;
end if;
end process;
end;
```

b. Ecrivez un programme équivalent à l'aide d'un seul processus

**Exercice 6 :**

Soit le code VHDL suivant.

```
library IEEE;
use IEEE.std_logic_1164.all;
entity exo_1 is
port (A, B, C : in std_logic;
Q1, Q2 : out std_logic);
end exo_1;
architecture synth of exo_1 is
signal V, R : std_logic;
begin
process (V, C)
begin
if (V='1') then
Q2 <= C;
end if;
end process;
R <= B xor C;
process (A)
begin
if (A'event and A='1') then
Q1 <= C;
V <= R;
end if;
end process;
end synth;
```



**Figure.1:** Chronogramme de l'exercice 6

Dessiner le système représenté par ce code. Compléter le diagramme des temps de la figure 1.

