



EXAMEN FINAL

(Correction)

(A)

Matrice de Programmable

Exercice N°1 (3pt)

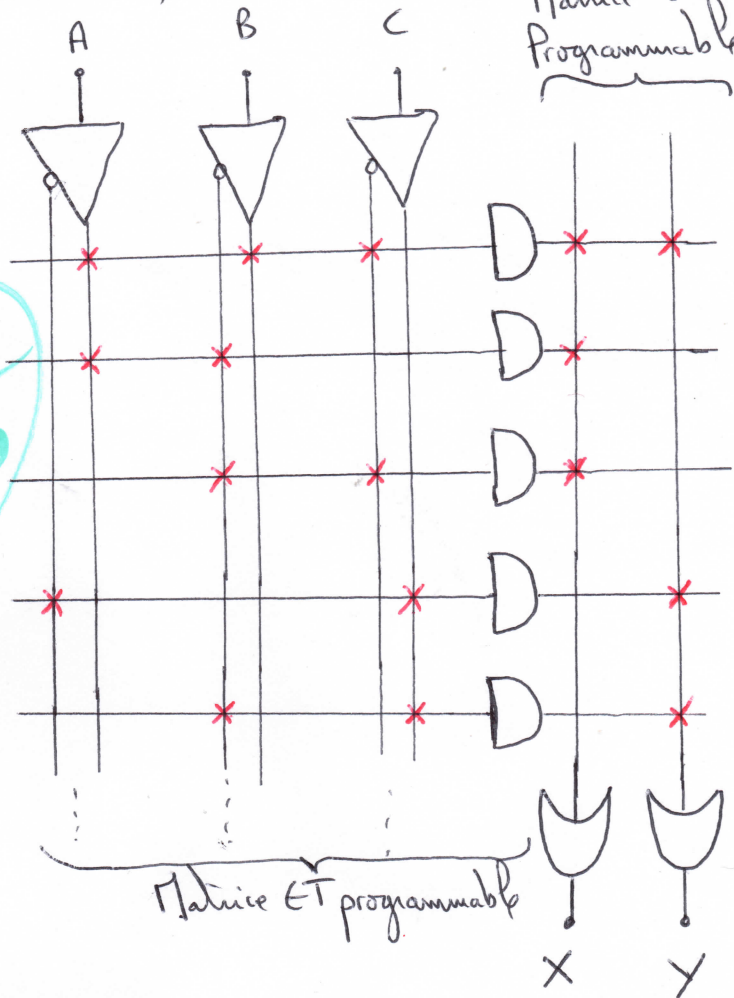
Dans un circuit PLA (Programmable Logic Array), les matrices ET et OU sont programmables, ce qui offre une très grande souplesse.

$$X = ABC + \bar{A}\bar{B} + \bar{C}\bar{B}$$

$$Y = (AB) \oplus C = ABC + AB\bar{C}$$

$$Y = (\bar{A} + \bar{B})C + ABC$$

$$Y = \bar{A}C + \bar{B}C + ABC$$



3pt

Exercice N°2 (8pt)

$$S = AC + \bar{B}C + B\bar{C}$$

Table de vérité

A	B	C	S
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

a) Description par flot de données

2pt

```

1
2
3
4 --EXERCICE N°2a--
5 --Flot de données--
6
7 library ieee;
8 use ieee.std_logic_1164.all;
9
10 entity exo2_a is
11     port ( A, B, C : in std_logic;
12           S       : out std_logic);
13 end exo2_a;
14
15 architecture flot_de_donnees of exo2_a is
16 begin
17     S <= (A and C) or (not B and C) or (B and not C);
18 end flot_de_donnees;
19
    
```

b) Description comportementale sans proces.

```

1
2 --EXERCICE N°2b--
3 --1ère Version SANS PROCESS
4 library ieee;
5 use ieee.std_logic_1164.all;
6
7 entity exo2_b is
8     port(A, B, C : in std_logic;
9           S : out std_logic);
10 end exo2_b;
11
12 architecture comp_sans_process1 of exo2_b is
13 begin
14     S <= '0' when (A='0' and B='0' and C='0') or (A='1' and B='0' and C='0') or (A='0' and
15                B='1' and C='1')
16                else '1';
17 end comp_sans_process1;

```

1^{ère} Version

3ph

```

--EXERCICE N°2b--
--3ème Version SANS PROCESS
library ieee;
use ieee.std_logic_1164.all;

```

3^e Version

```

entity exo2_b is
    port(A,B,C : in std_logic;
          S : out std_logic);
end exo2_b;

architecture comp_sans_process3 of exo2_b is
    signal entrees : std_logic_vector(2 downto 0);
begin
    entrees(2) <= A;
    entrees(1) <= B;
    entrees(0) <= C;
    with entrees select
        S <= '0' when "000" | "100" | "011",
            '1' when others;
end comp_sans_process3;

```

```

18
19 --EXERCICE N°2b--
20 --2ème Version SANS PROCESS
21 library ieee;
22 use ieee.std_logic_1164.all;
23
24 entity exo2_b is
25     port(ABC : in std_logic_vector(2 downto 0);
26           S : out std_logic);
27 end exo2_b;
28
29 architecture comp_sans_process2 of exo2_b is
30 begin
31     with ABC select
32         S <= '0' when "000" | "100" | "011",
33         '1' when others;
34 end comp_sans_process2;
35

```

2^e Version

c) Description comportementale avec proces

```

1 --EXERCICE N°2c--1ère Version AVEC PROCESS
2 library ieee;
3 use ieee.std_logic_1164.all;
4
5 entity exo2_c is
6     port(ABC : in std_logic_vector(2 downto 0);
7           S : out std_logic);
8 end exo2_c;
9
10 architecture comp_avec_process1 of exo2_c is
11 begin
12     process(ABC)
13     begin
14         case ABC is
15             when "000" => S <= '0';
16             when "011" => S <= '0';
17             when "100" => S <= '0';
18             when others => S <= '1';
19         end case;
20     end process;
21 end comp_avec_process1;

```

1^{ère} Version

3ph

```

22
23 --EXERCICE N°2c--2ème Version AVEC PROCESS
24 library ieee;
25 use ieee.std_logic_1164.all;
26
27 entity exo2_c is
28     port
29         (A, B, C : in std_logic;
30          S : out std_logic);
31 end exo2_c;
32
33 architecture comp_avec_process2 of exo2_c is
34 begin
35     process(A,B,C)
36     begin
37         if (A='0' and B='0' and C='0')
38             or (A='0' and B='1' and C='1')
39             or (A='1' and B='0' and C='0')
40             then
41             S <= '0';
42         else S <= '1';
43         end if;
44     end process;
45 end comp_avec_process2;

```

2^e Version

Exercice N°3 (9pt)Description des Composants

```

1  --EXERCICE N°3--
2  --DECLARATION DES COMPOSANTS--
3
4  --PORTE XOR----
5  library IEEE ;
6  use IEEE.std_logic_1164.all;
7
8  entity porte_xor is
9  port ( A,B : in  std_logic;
10         S: out std_logic);
11  end porte_xor;
12
13  architecture description of porte_xor is
14  begin
15      S <= A xor B ;
16  end description;
17
18  --BASCULE D----
19  library IEEE ;
20  use IEEE.std_logic_1164.all;
21
22  entity bascule_D is
23  Port ( D ,CLK : in std_logic;
24         Q : out  std_logic );
25  end bascule_D;
26  architecture description of bascule_D is
27  begin
28      process(CLK)
29  begin
30          if (CLK'event and CLK='1') then
31              Q <= D;
32          end if;
33      end process;
34  end description;
35
36  --MULTIPLEXEUR-----
37  library IEEE ;
38  use IEEE.std_logic_1164.all;
39
40  entity multiplexeur is
41  port (D0,D1,D2,D3 : in std_logic;
42         SEL : in std_logic_vector(1 downto 0);
43         MX_OUT : out std_logic);
44  end multiplexeur;
45
46  architecture description of multiplexeur is
47  begin
48      with SEL select
49          MX_OUT <= D0 when "00",
50                  D1 when "01",
51                  D2 when "10",
52                  D3 when others;
53  end description;

```

4,5pt

ENA (FPGA-VHDL)

D

EXAMEN FINAL
(Correction) (Suite)

Exercice N°3 (9pt) (Suite)

Description structurelle

```

1  --EXERCICE N°3--
2  --DESCRIPTION STRUCTURELLE--
3
4  library IEEE ;
5  use IEEE.std_logic_1164.all;
6
7  entity exo3 is
8  port (V,W,X,Y,H : in std_logic;
9        SEL: in std_logic_vector(1 downto 0);
10       U1,U2 : out std_logic);
11 end exo3;
12
13 architecture description of exo3 is
14  -- déclaration des composants -----
15  component porte_xor
16  port ( A,B : in std_logic;
17        S : out std_logic);
18  end component;
19  component bascule_D
20  Port ( D ,CLK : in std_logic;
21        Q : out std_logic );
22  end component;
23  component multiplexeur
24  port (D0,D1,D2,D3 : in std_logic;
25        SEL : in std_logic_vector(1 downto 0);
26        MX_OUT : out std_logic);
27  end component ;
28  -- signaux intermédiaires -----
29  signal S1,S2,S3 : std_logic;
30  begin
31  D1: bascule_D
32  port map ( D => S1,CLK => H,Q => S2 );
33  D2: bascule_D
34  port map ( D=>S2,CLK=>H,Q=>S3);
35  M : multiplexeur
36  port map ( D0=>V,D1=>W,D2=>X,D3=>Y,SEL=>SEL,MX_OUT=>S1);
37  PXOR1: porte_xor
38  port map ( A => V,B => S3,S => U1 );
39  PXOR2: porte_xor
40  port map ( A => S3,B => S1,S => U2 );
41  end description;
42

```

4,5pt