Compte rendu de TP3 – Full subtractor

1.1 Objectif de ce TP

Le but de ce TP est de nous familiariser avec les logiciels LT-Spice et VIVADO, et de faire un exercice similaire à Full additionneur précédent avec sa simulation sous LT-Spice et ensuite sur VIVADO.

1.1.1 Exercice part 1 (logiciels LT-Spice)

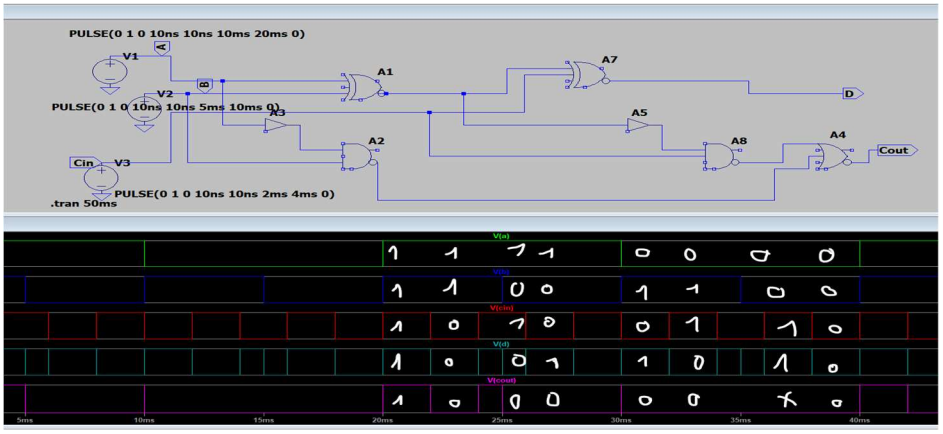
Construire un full-subtractor 4bits

- Construire la table de vérité

- Réaliser le schéma électronique

- Réaliser l'analyse de chemin critique

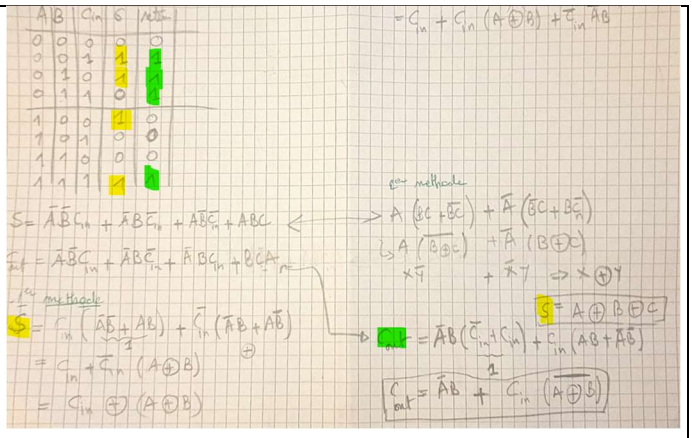
- Simuler le circuit sous LT-spice, prouver son fonctionnement



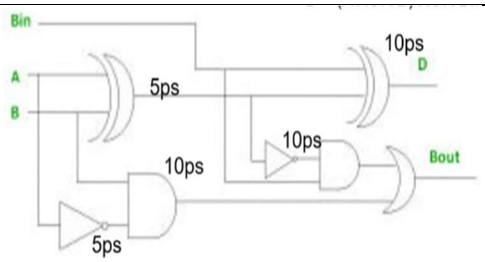
Compte rendu Eve CHAR 28/04/2023

INPUT			OUTPUT	
A	B	Bin	D	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

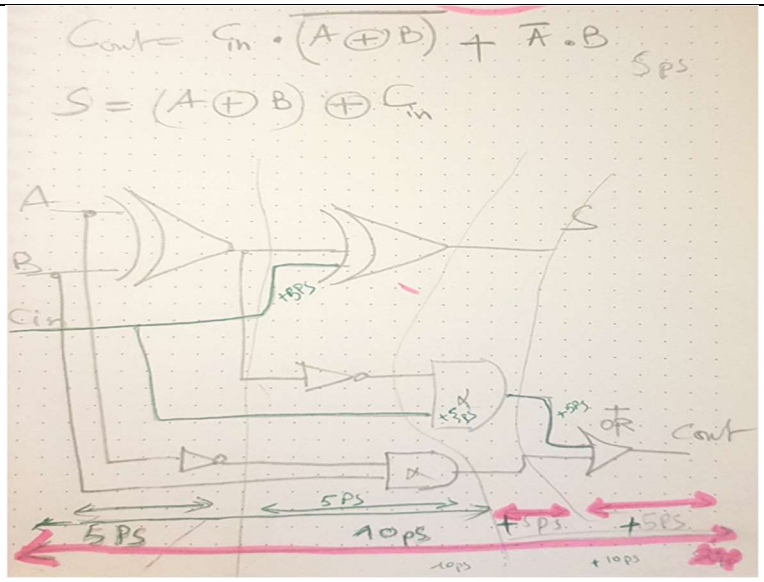
0 - 0 =0
0 – 1= 1 avec retenue 1
1-1-1=-2=10=> D=0 et retenue1

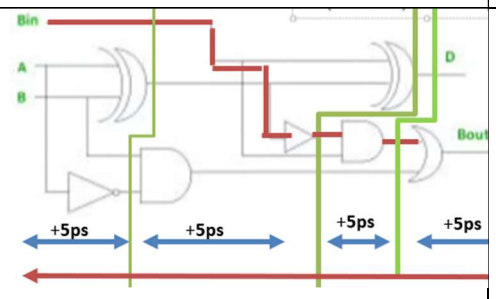


➔l'analyse de chemin critique



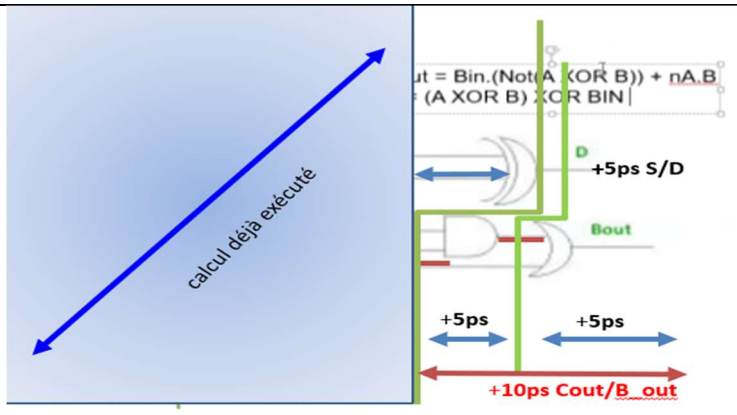
D=S= (A XOR B) XOR Cin
Cout= Bout= Cin . [not (A XOR B)]+B.not A





Latence1er **LSB** (bit0A-bit0B)

- = 20ps pour la retenue (Cout ou Bin)
- = 10ps pour la sortie D ou S

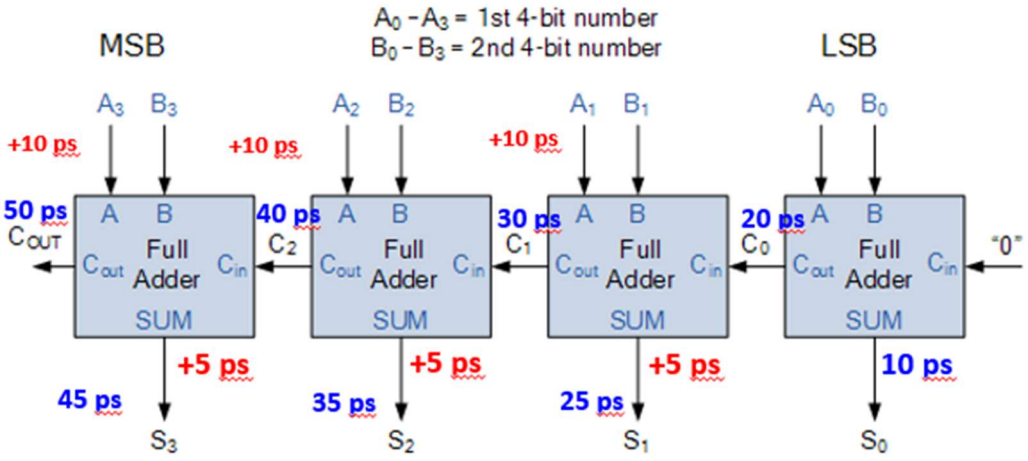


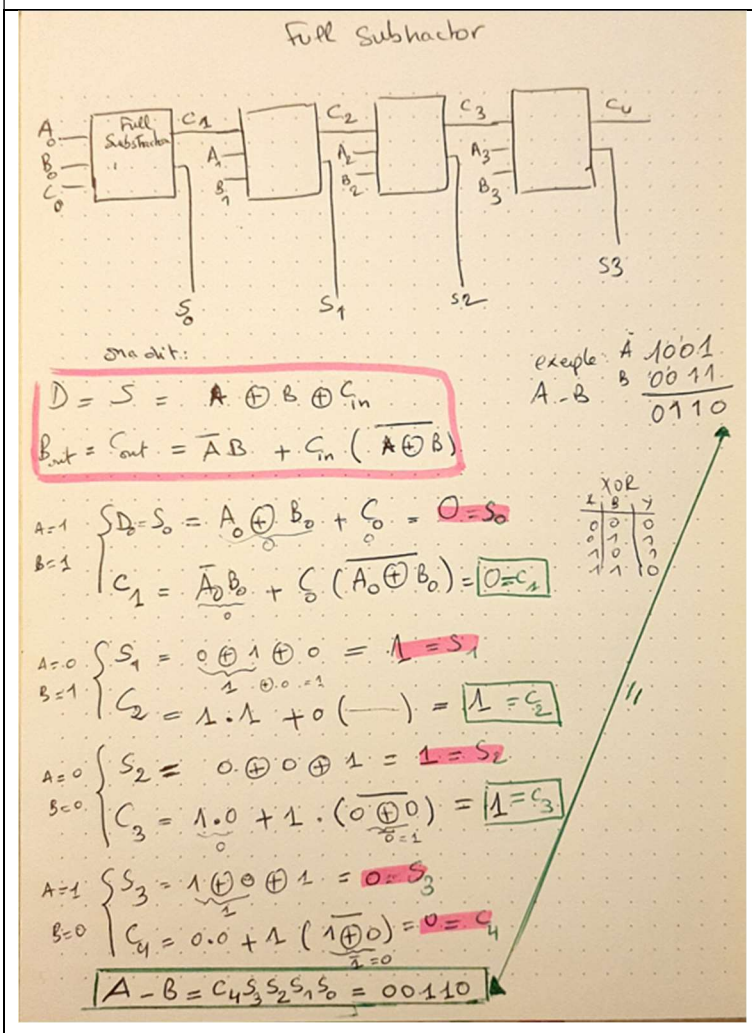
Latence pour chaque passage d'un block full soustracteur on a :

Latence (bit1A-bit1B)=Cout = 20ps +10ps & S=20ps+5ps

Compte rendu Eve CHAR 28/04/2023

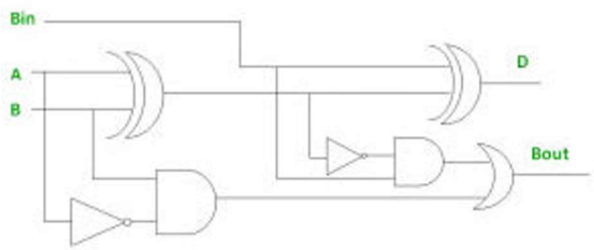
	Latence (bit2A-bit2B)=Cout = 30ps +10ps & S=30ps+5ps Latence (bit3A-bit3B)=Cout = 40ps +10ps & S=40ps+5ps





1.1.2 Exercice part 2 (logiciels Vivado)

- l'architecture RTL d'un full subtractor est la suivante :



INPUT			OUTPUT	
A	B	Bin	D	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Le code subtractor :

```
1
2   -- Déclaration des paquetages utiles pour le module
3   library ieee;
4   use ieee.std_logic_1164.all;
5   -- use ieee.std_logic_arith.all;
6   --use ieee.std_logic_unsigned.all;
7
8   --Description externe
9   entity subtractor is
10      Port ( A : in STD_LOGIC;
11             B : in STD_LOGIC;
12             Cin : in STD_LOGIC;
13             S : out STD_LOGIC;
14             Cout : out STD_LOGIC);
15  end subtractor;
16
17  architecture Behavioral of subtractor is
18  begin
19      S <= A XOR B XOR Cin ;
20      Cout <=((not A)and B) or (Cin AND (not(A XOR B)));
21
22  end Behavioral;
23
```

S <= A XOR B XOR Cin ;

Cout <= ((not A)and B) or (Cin AND (not(A XOR B)));

Le code test bench - subtractor :

```
1   library ieee;
2   use ieee.std_logic_1164.all;
3
4   entity testbench_full_subtractor is
5   end testbench_full_subtractor;
6
7   architecture behavior of testbench_full_subtractor is
8
9       -- component declaration for the unit under test (uut)
10
11      component full_subtractor
12          port(
13              A   : in std_logic;
14              B   : in std_logic;
15              Cin : in std_logic;
16              S   : out std_logic;
17              Cout: out std_logic
18          );
19      end component;
20
21      --Inputs
22      signal A   : std_logic := '0';
23      signal B   : std_logic := '0';
24      signal Cin : std_logic := '0';
25
26      --Outputs
27      signal S   : std_logic:= '0';
28      signal Cout : std_logic:= '0';
29
30  begin
31
32      -- Instantiate the Unit Under Test (UUT)
33      uut: full_subtractor
34          port map (
35              A => A,
36              B => B,
```

```
11      component full_subtractor
12          port(
13              A   : in std_logic;
14              B   : in std_logic;
15              Cin : in std_logic;
16              S   : out std_logic;
17              Cout: out std_logic
18          );
19      end component;
20
21      --Inputs
22      signal A   : std_logic := '0';
23      signal B   : std_logic := '0';
24      signal Cin : std_logic := '0';
25
26      --Outputs
27      signal S   : std_logic:= '0';
28      signal Cout : std_logic:= '0';
29
30  begin
31
32      -- Instantiate the Unit Under Test (UUT)
33      uut: full_subtractor
34          port map (
35              A => A,
36              B => B,
37              Cin => Cin,
38              S => S,
39              Cout => Cout
40          );
41      process
42      begin
43      -- hold reset state for 100 ns.
44      wait for 100 ns;
```

- ajouter des tests automatiques :

assert S='x' and Cout= 'Y' report "erreur calcul :expected S='X' and Cout='Y'" severity failure;

INPUT			OUTPUT	
A	B	Bin	D	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

1) **A <= '1'; B <= '0'; Cin <= '0'**

assert S='1' and Cout= '0' report "erreur calcul : expected S='1' and Cout='0'" severity failure;

2) **A <= '0'; B <= '1'; Cin <= '0'**

assert S='1' and Cout= '0' report "erreur calcul : expected S='1' and Cout='0'" severity failure;

3) **A <= '1'; B <= '1'; Cin <= '0'**

assert S='0' and Cout= '1' report "erreur calcul : expected S='0' and Cout='1'" severity failure;

4) **A <= '0'; B <= '0'; Cin <= '1'**

assert S='1' and Cout= '0' report "erreur calcul : expected S='1' and Cout='0'" severity failure;

5) **A <= '1'; B <= '0'; Cin <= '1'**

assert S='0' and Cout= '1' report "erreur calcul : expected S='0' and Cout='1'" severity failure;

6) **A <= '0'; B <= '1'; Cin <= '1'**

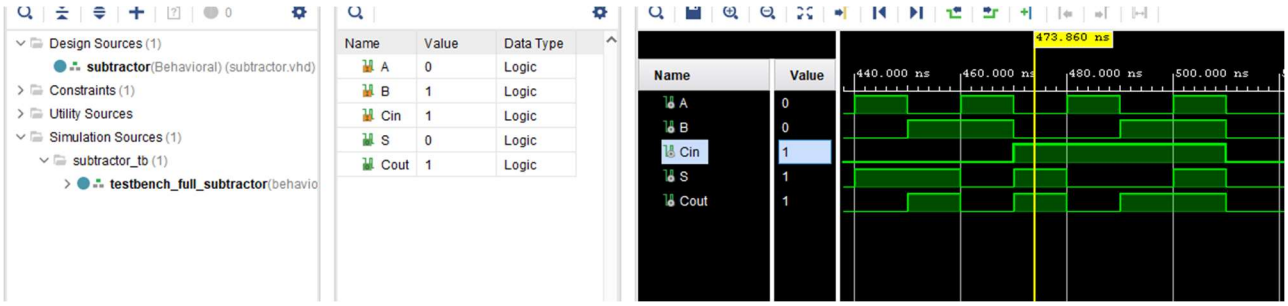
assert S='0' and Cout= '1' report "erreur calcul : expected S='0' and Cout='1'" severity failure;

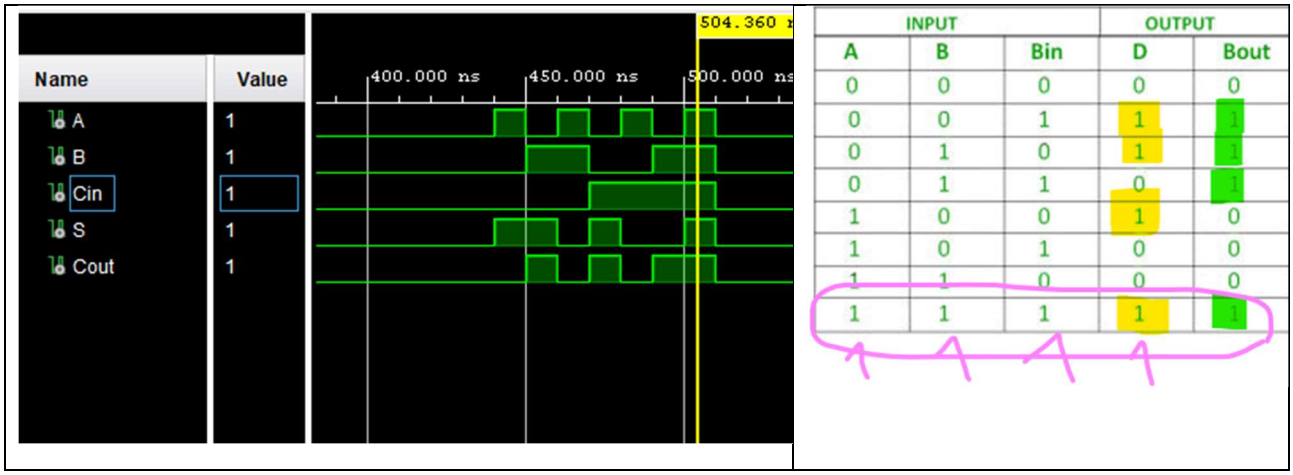
7) **A <= '1'; B <= '1'; Cin <= '1'**

assert S='1' and Cout= '1' report "erreur calcul : expected S='1' and Cout='1'" severity failure;

Compte rendu Eve CHAR 28/04/2023

- La simulation schématique associée / Behavioral Simulation :





INPUT			OUTPUT	
A	B	Bin	D	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

- Le schématique associé et La table de vérité associée