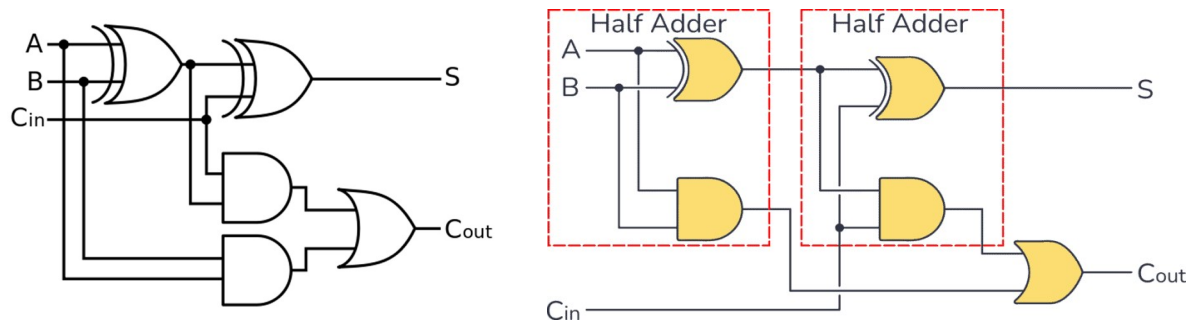


Compte rendu de TP1 – Full adder

1.1 Objectif de ce TP

Le but de ce TP est de nous familiariser avec le logiciel VIVADO, et de faire de la modélisation d'un Full-adder avec son test bench associé...

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



- Ecrivez sous forme d'équation logique le schéma du full adder :

- La sortie S du full adder :

la table de vérité du circuit :

A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$\Leftrightarrow S = C_{in} \oplus A \oplus B$

- La sortie C-out du full adder :

la table de vérité du circuit :

A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$\Leftrightarrow c0 = C_{in}(A \oplus B) + A.B.1$

2. Quelles sont les entrées et sorties du full adder ?

*Les entrées : nous avons 3 entrées A, B, Cin

*Les sorties : nous avons 2 sorties : S et Cout

3. Complétez le fichier full_adder.vhd pour décrire en VHDL le full adder :

<pre> library ieee; use ieee.std_logic_1164.all; entity full_adder is Port (--Entrées full adder A : in std_logic; B : in std_logic; Cin : in std_logic; --Sorties full adder S : out std_logic; Cout : out std_logic); end full_adder; architecture behavior of full_adder is begin S <= A XOR B XOR Cin ; Cout <= Cin and (A xor B) or (A and B); end behavior; </pre>	<pre> library ieee; use ieee.std_logic_1164.all; entity full_adder is Port (--Entrées full adder A : in std_logic; B : in std_logic; Cin : in std_logic; --Sorties full adder S : out std_logic; Cout : out std_logic); end full_adder; architecture behavior of full_adder is begin S <= A XOR B XOR Cin ; Cout <= (Cin and (A xor B)) or (A and B); end behavior; </pre>
--	--

4. Ouvrez Vivado et créez un projet

5. Complétez les commentaires dans le fichier de testbench testbench_full_adder.vhd.

6. Sur Vivado, dans l'onglet « Sources », faites un clic droit sur « Simulation Sources » puis « Add Sources »

7. Dans l'onglet « Flow Navigator », cliquez sur « Run Behavioral Simulation ».

8. Modifiez le testbench pour ajouter des tests automatiques :

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

A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
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;

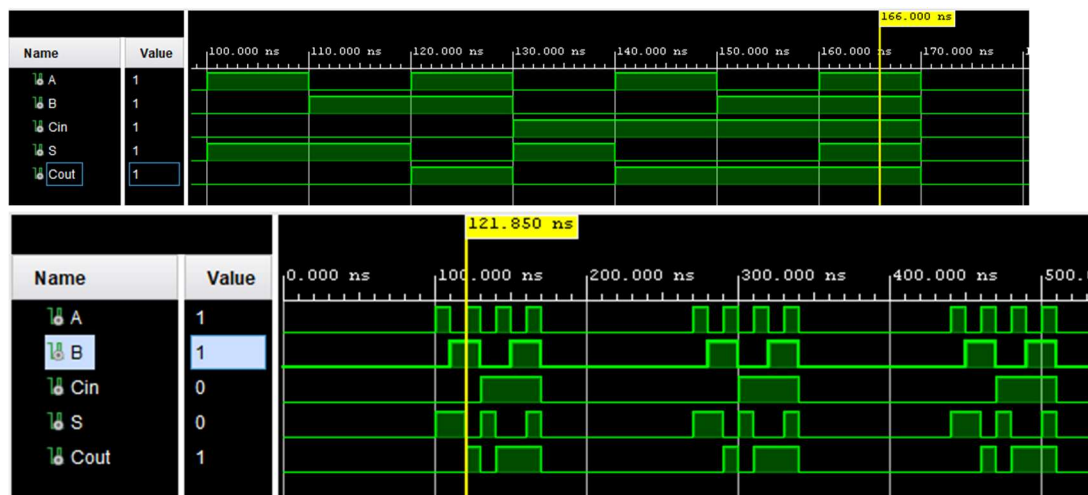
9. Dans l'onglet « Flow Navigator », cliquez « Run Synthesis » pour exécuter une synthèse de votre architecture.

10. Déroulez le menu « Synthesis » et cliquez sur « Schematic » pour ouvrir la schématique. Où sont les portes logiques de l'architecture ? Vous pouvez mettre en surbrillance les fils en cliquant dessus pour mieux suivre leur chemin

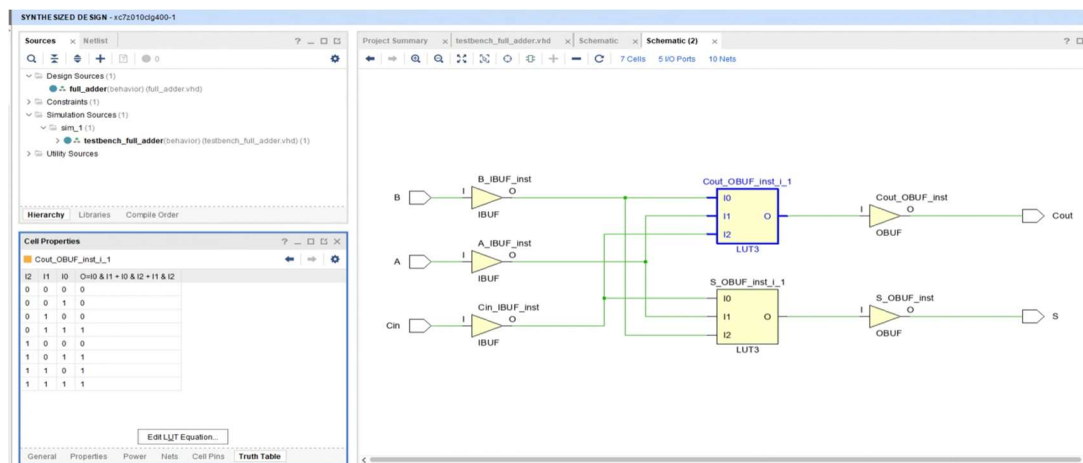
- le code test-bench full-adder

<pre> library ieee; use ieee.std_logic_1164.all; entity testbench_full_adder is end testbench_full_adder; architecture behavior of testbench_full_adder is -- component declaration for the unit under test (uut) component full_adder port(A : in std_logic; B : in std_logic; Cin : in std_logic; S : out std_logic; Cout : out std_logic); end component; --Inputs signal A : std_logic := '0'; signal B : std_logic := '0'; signal Cin : std_logic := '0'; --Outputs signal S : std_logic; signal Cout : std_logic; begin -- Instantiate the Unit Under Test (UUT) uut: full_adder port map (A => A, B => B, Cin => Cin, S => S, Cout => Cout); process begin -- hold reset state for 100 ns. wait for 100 ns; --Valeurs des sorties attendues : -- Cout <= '0'; -- S <= '0'; A <= '1'; B <= '0'; Cin <= '0'; wait for 10 ns; --Valeurs des sorties attendues : -- Cout <= '0'; -- S <= '1'; A <= '0'; B <= '1'; Cin <= '0'; wait for 10 ns; --Valeurs des sorties attendues : -- Cout <= '0'; -- S <= '1'; A <= '1'; B <= '1'; Cin <= '0'; wait for 10 ns; --Valeurs des sorties attendues : -- Cout <= '1'; -- S <= '0'; end process; end; </pre>	<pre> A <= '0'; B <= '0'; Cin <= '1'; wait for 10 ns; --Valeurs des sorties attendues : -- Cout <= '0'; -- S <= '1'; A <= '1'; B <= '0'; Cin <= '1'; wait for 10 ns; --Valeurs des sorties attendues : -- Cout <= '1'; -- S <= '0'; A <= '0'; B <= '1'; Cin <= '1'; wait for 10 ns; --Valeurs des sorties attendues : -- Cout <= '1'; -- S <= '0'; A <= '1'; B <= '1'; Cin <= '1'; wait for 10 ns; --Valeurs des sorties attendues : -- Cout <= '1'; -- S <= '1'; A <= '0'; B <= '0'; Cin <= '0'; end process; end; </pre>
---	--

- La simulation schématique associée / Behavioral Simulation :



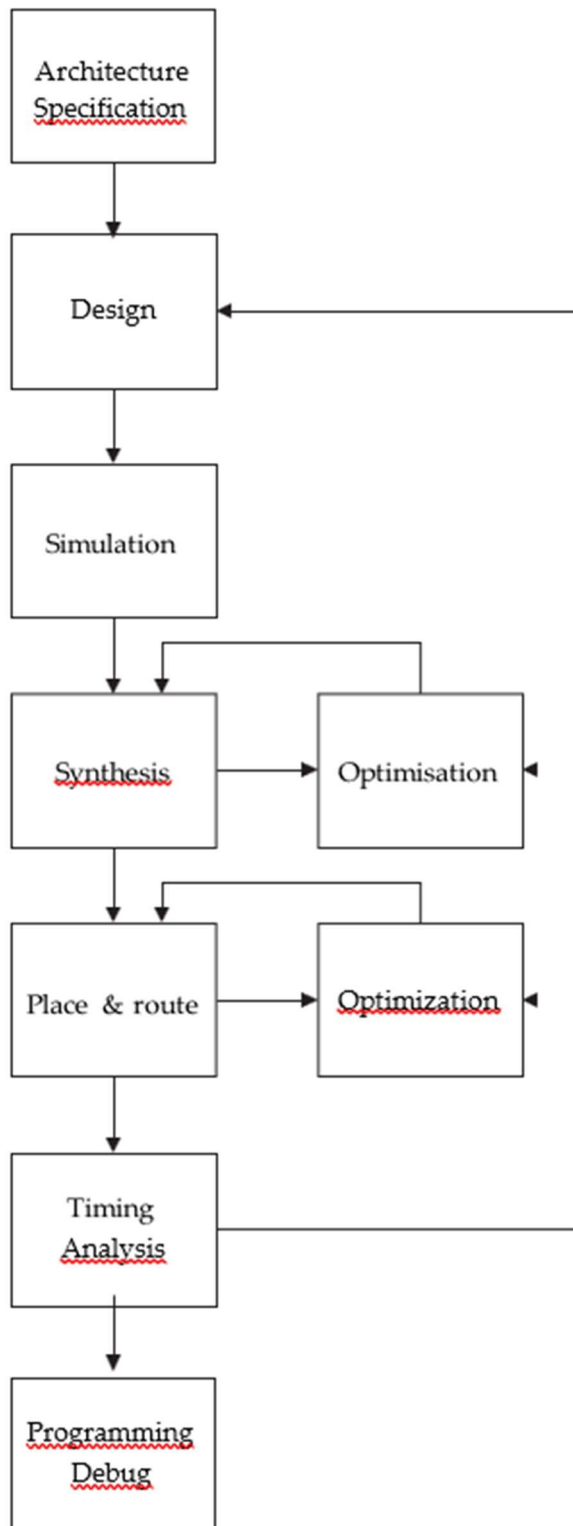
- Le schématic associé :



- La table de vérité associée

I2	I1	I0	O=I0 & I1 + I0 & I2 + I1 & I2
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

A retenir jusqu'aujourd'hui 26 /04/2023 : La conception d'un programme VHDL peut être résumée par le diagramme de la figure suivante .



1.2 Annexe servi pour ce TP :

Full Additionneur

$$S_i = (A_i \oplus B_i) \oplus C_{in}$$

$$C_{out} = A \cdot B + (A \oplus B) \cdot C_{in}$$

exemple

$$\begin{array}{r} 1101 \\ + 1100 \\ \hline 11001 \end{array}$$

$A_0=1, B_0=0 \Rightarrow \begin{cases} S_0 = 1 \oplus 0 \oplus 0 = 1 = S_0 \\ C_1 = 1 \cdot 0 + (1 \oplus 0) \cdot 0 = 0 = C_1 \end{cases}$

$A_1=0, B_1=0 \Rightarrow \begin{cases} S_1 = 0 \oplus 0 \oplus C_1 = 0 = S_1 \\ C_2 = 0 \cdot 0 + (0 \oplus 0) \cdot C_1 = 0 = C_2 \end{cases}$

$A_2=1, B_2=1 \Rightarrow \begin{cases} S_2 = 1 \oplus 1 \oplus 0 = 0 = S_2 \\ C_3 = 1 \cdot 1 + (1 \oplus 1) \cdot C_2 = 1 = C_3 \end{cases}$

$A_3=1, B_3=1 \Rightarrow \begin{cases} S_3 = 1 \oplus 1 \oplus C_3 = 1 = S_3 \\ C_4 = 1 \cdot 1 + (1 \oplus 1) \cdot C_3 = 1 = C_4 \end{cases}$

$$A + B = C_4 S_3 S_2 S_1 S_0 = 11001$$