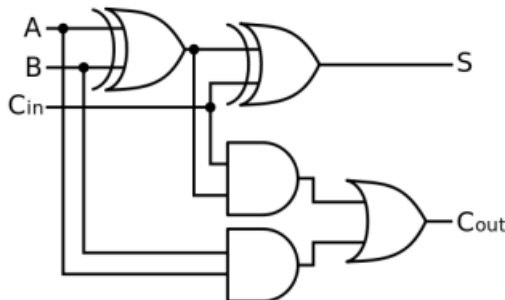


Segment 2

TP1 : Le full adder

1. Commencer par le schéma RTL / synoptique



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

- Pour le signal « S », déterminons dans quels cas S est à 1 :
$$S = (\bar{A} \text{ and } \bar{B} \text{ and } C_{in}) \text{ or } (\bar{A} \text{ and } B \text{ and } \overline{C_{in}}) \text{ or } (A \text{ and } \bar{B} \text{ and } \overline{C_{in}}) \text{ or } (A \text{ and } B \text{ and } C_{in})$$
- Pour le signal « Cout », déterminons dans quels cas Cout est à 1
$$C_o = (\bar{A} \text{ and } B \text{ and } C_{in}) \text{ or } (A \text{ and } \bar{B} \text{ and } C_{in}) \text{ or } (A \text{ and } B \text{ and } \overline{C_{in}}) \text{ or } (A \text{ and } B \text{ and } C_{in})$$

Pour plus de praticité, on utilise généralement les symboles mathématiques dans nos équations.

La syntaxe devient alors :

$$C_o = (\bar{A} . B . C_{in}) + (A . \bar{B} . C_{in}) + (A . B . \overline{C_{in}}) + (A . B . C_{in})$$
$$S = (\bar{A} . \bar{B} . C_{in}) + (\bar{A} . B . \overline{C_{in}}) + (A . \bar{B} . \overline{C_{in}}) + (A . B . C_{in})$$

2. Les entrées et sorties du full adder

Entrées : A,B et Cin

Sorties : S, Cout

3. Complétez le fichier full_adder.vhd pour décrire en VHDL le full adder. Retranscrire le schéma RTL en code VHDL.

```
library ieee;
```

```
use ieee.std_logic_1164.all;
```

```
entity full_adder is
```

```

Port (
    --Exemple d'entrees
    A      : in std_logic;
    B      : in std_logic;
    Cin    : in std_logic;

    --Exemple de sorties
    S      : out std_logic;
    Cout   : out std_logic);

end full_adder;

architecture behavior of full_adder is
begin
    --Affectation d'une sortie
    S <= A XOR B XOR Cin; --Affectation d'une sortie
    Cout <= (A AND B) OR (Cin AND A) OR (Cin AND B); --Affectation d'une sortie
end behavior;

```

4. Vérifier le comportement du système en simulation

Sur le chronogramme, à l'aide du curseur vérifiez que les valeurs des sorties correspondent aux attentes que vous avez déterminé précédemment.

5. Modifiez le testbench pour ajouter des tests automatiques.

```

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;
assert S='1' and Cout='0'
report " test failed : excepted S='1' and Cout='0'"
--Valeurs des sorties attendues :

--      Cout = 0
--      S = 1

A <= '0';
B <= '1';
Cin <= '0';
wait for 10 ns;
assert S='1' and Cout='0'
report " test failed : excepted S='1' and Cout='0'"
--Valeurs des sorties attendues :

--      Cout = 0
--      S = 1

A <= '1';
B <= '1';
Cin <= '0';
wait for 10 ns;
assert S='0' and Cout='1'
report " test failed : excepted S='0' and Cout='1'"
--Valeurs des sorties attendues :

```

-- Cout = 1

-- S = 0

A <= '0';

B <= '0';

Cin <= '1';

wait for 10 ns;

assert S='1' and Cout='0'

report " test failed : excepted S='1' and Cout='0'"

--Valeurs des sorties attendues :

-- Cout = 0

-- S = 1

A <= '1';

B <= '0';

Cin <= '1';

wait for 10 ns;

assert S='0' and Cout='1'

report " test failed : excepted S='0' and Cout='1'"

--Valeurs des sorties attendues :

-- Cout = 1

-- S = 0

A <= '0';

B <= '1';

Cin <= '1';

wait for 10 ns;

assert S='0' and Cout='1'

report " test failed : excepted S='0' and Cout='1'"

--Valeurs des sorties attendues :

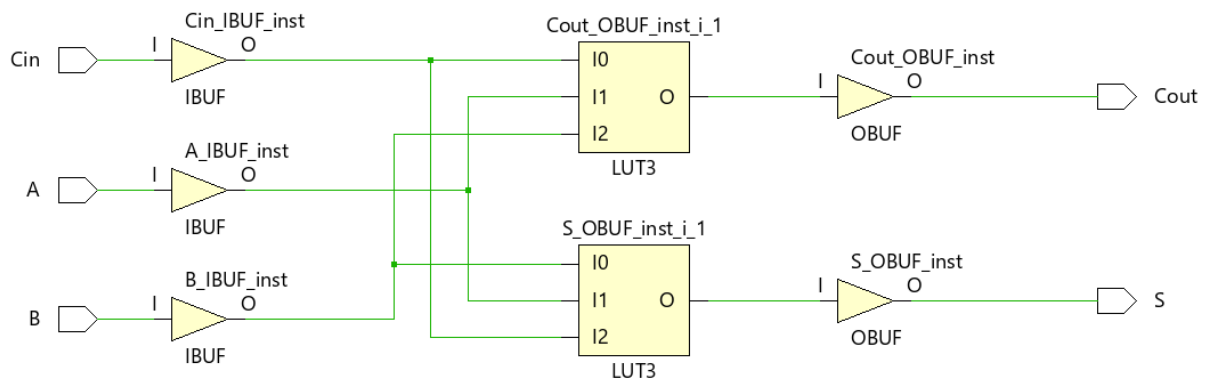
-- Cout = 1

```

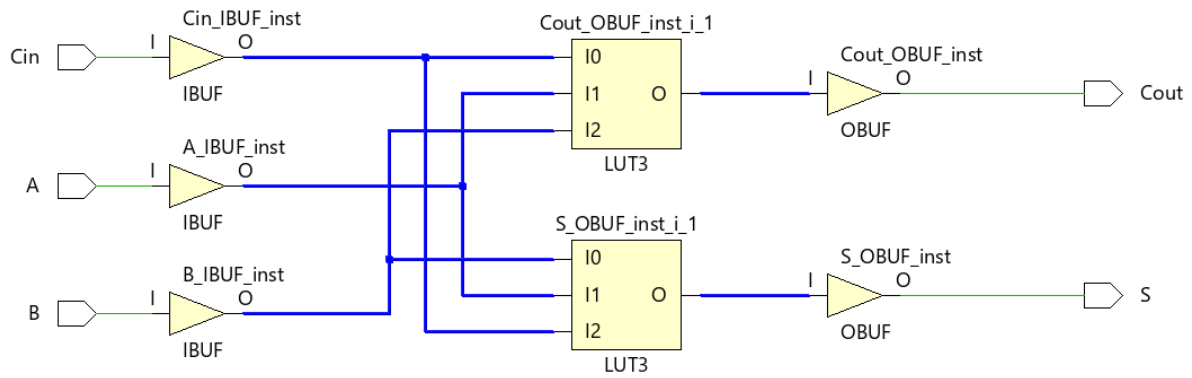
--      S = 0
A <= '1';
B <= '1';
Cin <= '1';
wait for 10 ns;
assert S='1' and Cout='1'
report " test failed : excepted S='1' and Cout='1'"
--Valeurs des sorties attendues :
--      Cout = 1
--      S = 1
A <= '0';
B <= '0';
Cin <= '0';
end process;
end;

```

6. Déroulez le menu « Synthesis » et cliquez sur « Schematic » pour ouvrir la schématique.
Où sont les portes logiques de l'architecture ?

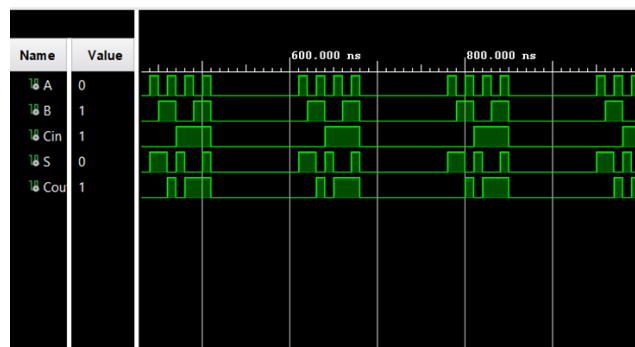


Vous pouvez mettre en surbrillance les fils en cliquant dessus pour mieux suivre leur chemin



7. Vérifier le comportement du système en simulation

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



5. Etudier la synthèse

6. Placer des sondes avec l'ILA

7. Etudier le placement routage (Place And Route -PAR-)

8. Vérifier le comportement du système sur carte