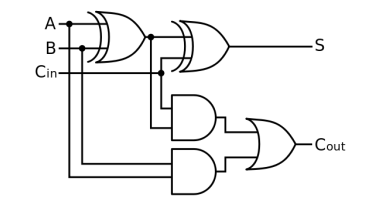
Segment 2

**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 :

* Pour le signal « Cout », déterminons dans quels cas Cout est à 1

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

|  |
| --- |
| La syntaxe devient alors : |

1. Les entrées et sorties du full adder

Entrées : A,B et Cin

Sorties : S, Cout

1. 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;*

1. 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.

1. 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';*

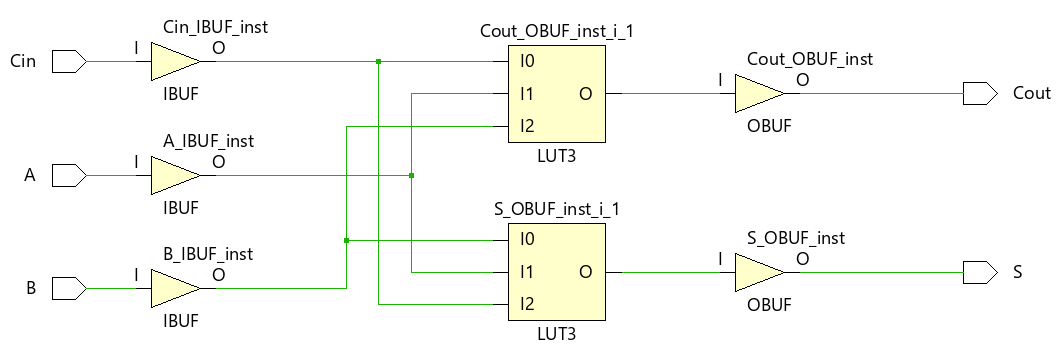
*B <= '0';*

*Cin <= '0';*

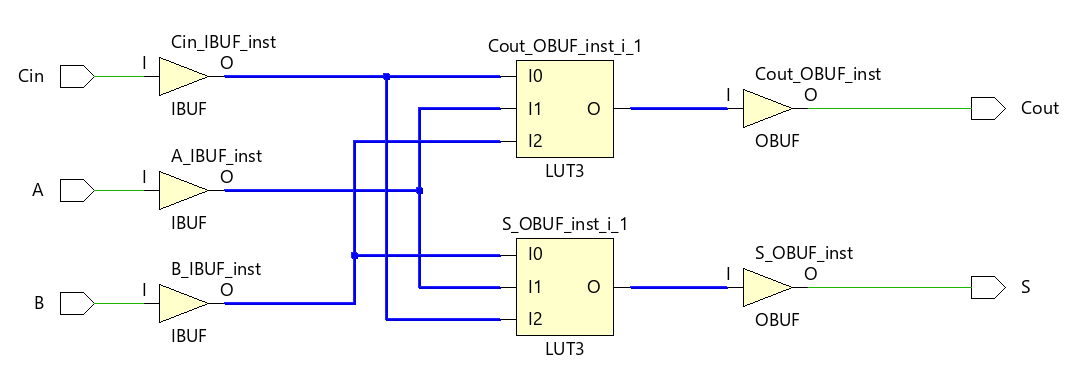
*end process;*

*end;*

1. 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



1. 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