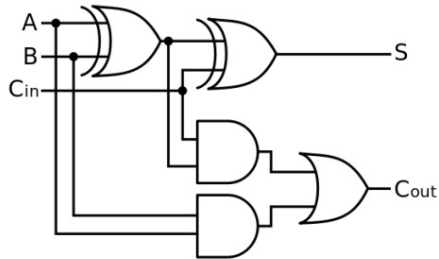


TP1 – Full adder

Pour rappel, l'architecture RTL d'un full adder est la suivante :



1. Ecrivez sous forme d'équation logique le schéma du full adder.
2. Quelles sont les entrées et sorties du full adder ?
3. Complétez le fichier *full_adder.vhd* pour décrire en VHDL le full adder.
4. Ouvrez Vivado et créez un projet



Cliquez sur « Next », nommez votre projet et choisissez son emplacement, puis « Next ».

New Project

Project Name

Enter a name for your project and specify a directory where the project data files will be stored.



Project name: full_adder

Project location: /your/path

☒ Create project subdirectory

Project will be created at: /your/path/full_adder



< Back

Next >

Finish

Cancel

Sélectionnez RTL Project, puis « Next »

New Project

Project Type

Specify the type of project to create.



- ☒ **RTL Project**
You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.
 - ☐ Do not specify sources at this time
 - ☐ Project is an extensible Vitis platform
- ☐ **Post-synthesis Project**
You will be able to add sources, view device resources, run design analysis, planning and implementation.
 - ☐ Do not specify sources at this time
- ☐ **I/O Planning Project**
Do not specify design sources. You will be able to view part/package resources.
- ☐ **Imported Project**
Create a Vivado project from a Synplify, XST or ISE Project File.
- ☐ **Example Project**
Create a new Vivado project from a predefined template.



< Back

Next >


Finish





Cancel

Cliquez sur « Add Files » pour ajouter vos fichiers sources, ajoutez votre fichier *full_adder.vhd*. Les fichiers sources pourront aussi être ajoutés plus tard, après la création du projet.

New Project

Add Sources
Specify HDL, netlist, Block Design, and IP files, or directories containing those files, to add to your project. Create a new source file on disk and add it to your project. You can also add and create sources later.





Use Add Files, Add Directories or Create File buttons below

Add Files

Add Directories

Create File

☐ Scan and add RTL include files into project

☐ Copy sources into project

☒ Add sources from subdirectories

Target language: Verilog Simulator language: Mixed

?

< Back

Next >


Finish





Cancel

Ajoutez le fichier de contrainte Cora-Z7-10-Master.xdc. Les fichiers de contraintes peuvent aussi être ajoutés après la création du projet.

New Project

Add Constraints (optional)
Specify or create constraint files for physical and timing constraints.






Use Add Files or Create File buttons below

☐ Copy constraints files into project

Add Files

Create File



< Back

Next >

Finish

Cancel

Sélectionnez la carte que vous allez utiliser pour ce projet. Vous avez à votre disposition une carte Cora-Z7-10, la puce correspondante est la xc7z010clg400-1.

New Project

Default Part
Choose a default Xilinx part or board for your project.

Parts | Boards
[Reset All Filters](#)

Category:
 Family:

Package:
 Speed:

Temperature:
 Static power:

Search:

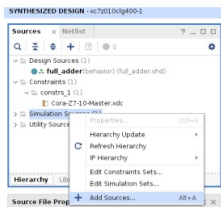
Part	I/O Pin Count	Available IOBs	LUT Elements	FlipFlops	Block RAMs	Ultra RAMs	DSPs	Gb Transceivers
xc7wv415tffv1157-1	1157	600	257600	515200	880	0	2160	20
xc7wv415tffv1158-3	1158	350	257600	515200	880	0	2160	48
xc7wv415tffv1158-2	1158	350	257600	515200	880	0	2160	48
xc7wv415tffv1158-2L	1158	350	257600	515200	880	0	2160	48
xc7wv415tffv1158-1	1158	350	257600	515200	880	0	2160	48
xc7wv415tffv1927-3	1927	600	257600	515200	880	0	2160	48
xc7wv415tffv1927-2	1927	600	257600	515200	880	0	2160	48
xc7wv415tffv1927-2L	1927	600	257600	515200	880	0	2160	48
xc7wv415tffv1927-1	1927	600	257600	515200	880	0	2160	48
xc7wv485tffg1157-3	1157	600	303600	607200	1030	0	2800	20
xc7wv485tffg1157-2	1157	600	303600	607200	1030	0	2800	20
xc7wv485tffg1157-2L	1157	600	303600	607200	1030	0	2800	20
xc7wv485tffg1157-1	1157	600	303600	607200	1030	0	2800	20
xc7wv485tffg1158-3	1158	350	303600	607200	1030	0	2800	48

<
>

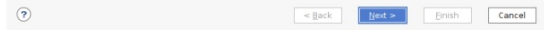
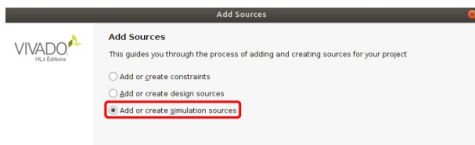
[?](#)
[< Back](#)
[Next >](#)
[Finish](#)
[Cancel](#)

Cliquez sur « Finish » pour terminer la configuration du projet.

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



Sélectionnez « Add or create simulation sources », puis « Next ».

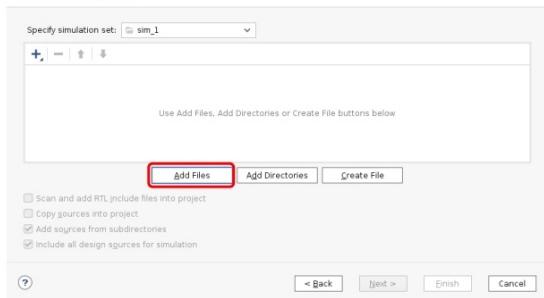


Ajoutez le fichier testbench `testbench_full_adder.vhd`.

Add Sources

Add or Create Simulation Sources

Specify simulation specific HDL files, or directories containing HDL files, to add to your project. Create a new source file on disk and add it to your project.



Cliquez sur « Finish »

C:/FORMATION_SAFRAN/TP/TP1_FullAdder/testbench_full_adder.vhd



```
1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  entity testbench_full_adder is
5  end testbench_full_adder;
6
7  architecture behavior of testbench_full_adder is
8
9      -- component declaration for the unit under test (uut)
10
11     component full_adder
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;
28     signal Cout : std_logic;
```

Définition des
entrées / sorties

Etat initial


```
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
```

```
assert Cout = '0'
```

```
    report "ERROR: Cout not equals to 0" severity failure;
```

```
assert S = '0'
```

```
    report "ERROR: S not equals to 0" severity failure;
```

Reprise des valeurs définies en page précédente (état initial)

Test unitaire de l'état initial

```
A <= '1';  
B <= '0';  
Cin <= '0';  
wait for 10 ns;
```



Entrées testées

```
--Valeurs des sorties attendues :  
-- Cout = 0  
-- S = 1  
assert Cout = '0'  
    report "ERROR: Cout not equals to 0" severity failure;  
assert S = '1'  
    report "ERROR: S not equals to 1" severity failure;
```

```
A <= '0';  
B <= '1';  
Cin <= '0';  
wait for 10 ns;
```

```
--Valeurs des sorties attendues :  
-- Cout = 0  
-- S = 1  
assert Cout = '0'  
    report "ERROR: Cout not equals to 0" severity failure;  
assert S = '1'  
    report "ERROR: S not equals to 1" severity failure;
```

```
A <= '0';
B <= '1';
Cin <= '0';
wait for 10 ns;

--Valeurs des sorties attendues :
-- Cout = 0
-- S = 1
assert Cout = '0'
    report "ERROR: Cout not equals to 0" severity failure;
assert S = '1'
    report "ERROR: S not equals to 1" severity failure;
```

```
A <= '0';
B <= '0';
Cin <= '0';
```

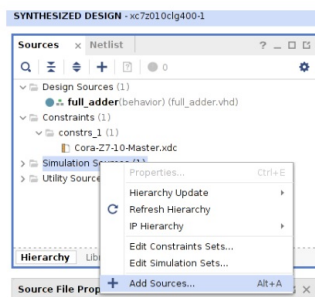
Remise à l'état initial

```
end process;
```

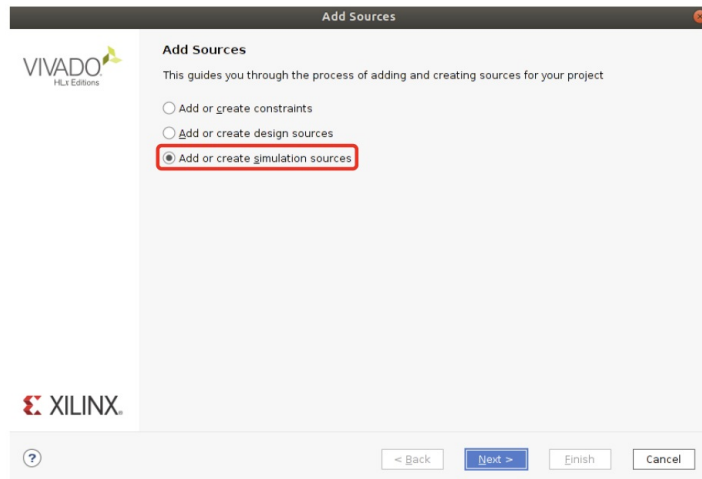
Fin de test

```
end;
```

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



Sélectionnez « Add or create simulation sources », puis « Next ».



Ajoutez le fichier testbench *testbench_full_adder.vhd*.

Add Sources

Add or Create Simulation Sources

Specify simulation specific HDL files, or directories containing HDL files, to add to your project. Create a new source file on disk and add it to your project.



Specify simulation set: sim_1

+ - ↑ ↓

Use Add Files, Add Directories or Create File buttons below

Add Files

Add Directories

Create File

- ☐ Scan and add RTL include files into project
- ☐ Copy sources into project
- ☒ Add sources from subdirectories
- ☒ Include all design sources for simulation



< Back

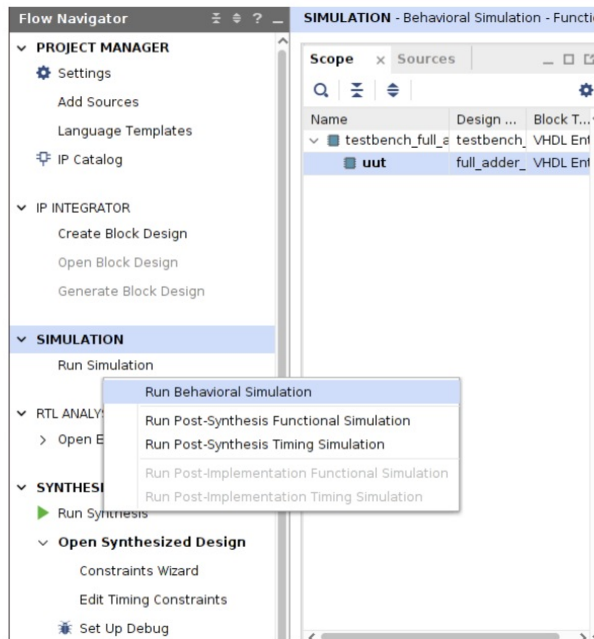
Next >

Finish

Cancel

Cliquez sur « Finish »

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



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

A ce stade, la simulation plante. Il faut modifier le fichier full_adder.vhd

full_adder.vhd x testbench_full_adder.vhd x Untitled 12 x

C:/FORMATION_SAFRAN/TP/TP1_FullAdder/full_adder.vhd

Scope
Sources
Objects
Protocol Instances

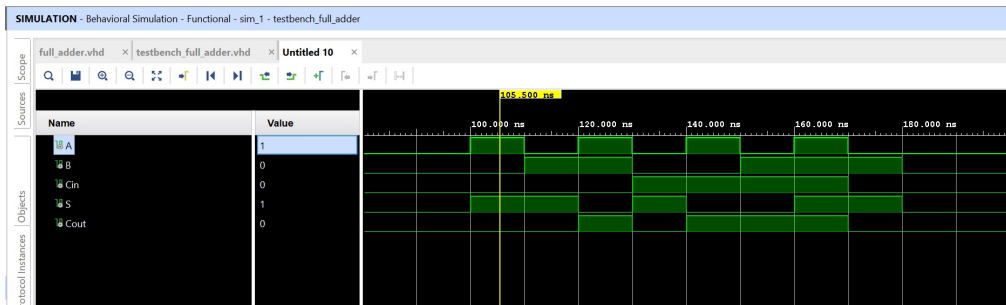
```
1  library ieee;
2  use ieee.std_logic_1164.all;
3
4
5  entity full_adder is
6
7      Port (
8          --Exemple d'entrees
9          A  : in std_logic;
10         B  : in std_logic;
11         Cin : in std_logic;
12
13         --Exemple de sorties
14         S   : out std_logic;
15         Cout: out std_logic
16     );
17
18 end full_adder;
19
20
21
22 architecture behavior of full_adder is
23
24     begin
25
26         S <= (A xor B) xor Cin;
27         Cout <= (A AND B) OR ((A XOR B) and Cin);
28
29     end behavior;
30
31
```

Reprise de la définition des entrées / sorties (un peu comme les classes en C++)


Transcription du schéma RTL

Exemple de TCL console quand tout se passe bien

```
Tcl Console × Messages Log
Q [ ] [ ] [ ] [ ] [ ] [ ] [ ]
INFO: [USF-XSim-4] XSim::Simulate design
INFO: [USF-XSim-61] Executing 'SIMULATE' step in 'C:/FORMATION_SAFRAN/TP/TPI_FullAdder/project_FullAdder/project_FullAdder.sim/sim_1/behav/xsim'
INFO: [USF-XSim-98] *** Running xsim
    with args "testbench_full_adder_behav -key {Behavioral:sim_1:Functional:testbench_full_adder} -tclbatch {testbench_full_adder.tcl} -log {simulate.log}"
INFO: [USF-XSim-8] Loading simulator feature
Vivado Simulator 2020.2
Time resolution is 1 ps
source testbench_full_adder.tcl
# set curr_wave [current_wave_config]
# if { [string length $curr_wave] == 0 } {
#   if { ([length [get_objects]] > 0) {
#     add_wave /
#     set_property needs_save false [current_wave_config]
#   } else {
#     send_msg_id Add_Wave-1 WARNING "No top level signals found. Simulator will start without a wave window. If you want to open a wave window go to File->New Waveform..."
#   }
# }
# run 1000ns
INFO: [USF-XSim-96] XSim completed. Design snapshot 'testbench_full_adder_behav' loaded.
INFO: [USF-XSim-97] XSim simulation ran for 1000ns
```



Exemple d'assert incorrect

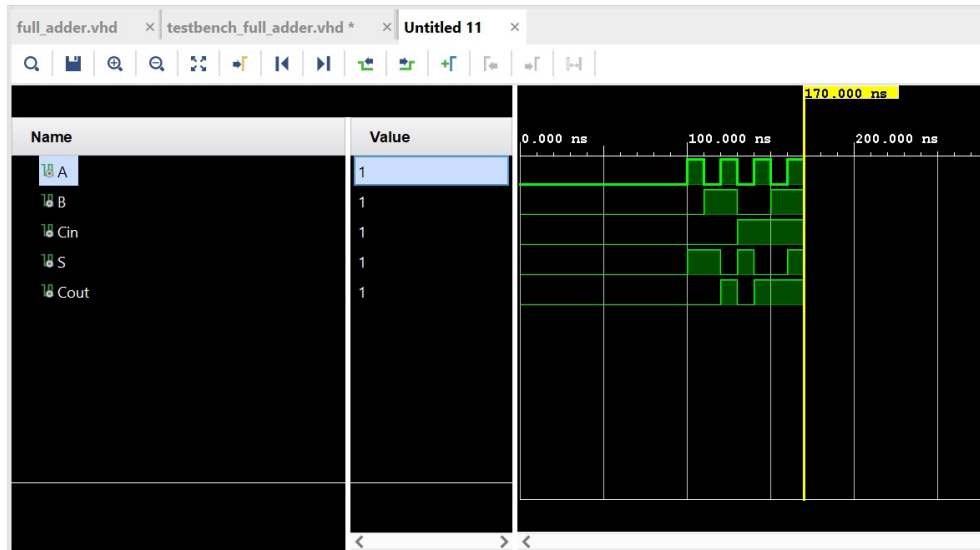


```
Tcl Console x Messages Log
# } else {
#   send_msg_id Add_Wave-1 WARNING "No top level signals found. Simulator will start without a wave window. If you want to open a wave w
# }
# }
# run 1000ns
Failure: ERROR: S not equals to 0
Time: 170 ns Iteration: 0 Process: /testbench_full_adder/line_43 File: C:/FORMATION_SAFRAN/TP1_FullAdder/testbench_full_adder.vhd
$finish called at time: 170 ns : File "C:/FORMATION_SAFRAN/TP1_FullAdder/testbench_full_adder.vhd" Line 156
INFO: [USF-XSim-96] XSim completed. Design snapshot 'testbench_full_adder_behav' loaded.
INFO: [USF-XSim-97] XSim simulation ran for 1000ns
launch_simulation: Time (s): cpu = 00:00:01 ; elapsed = 00:00:05 . Memory (MB): peak = 1211.152 ; gain = 0.648
```

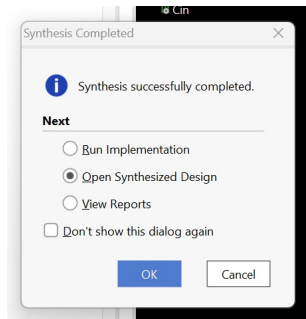
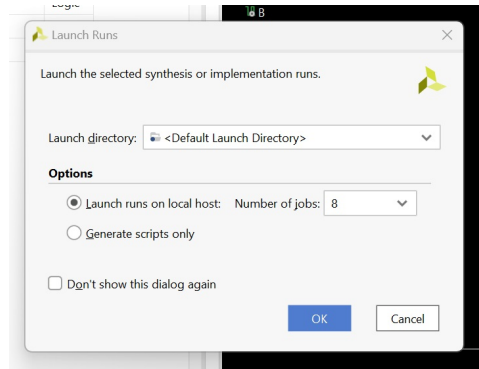
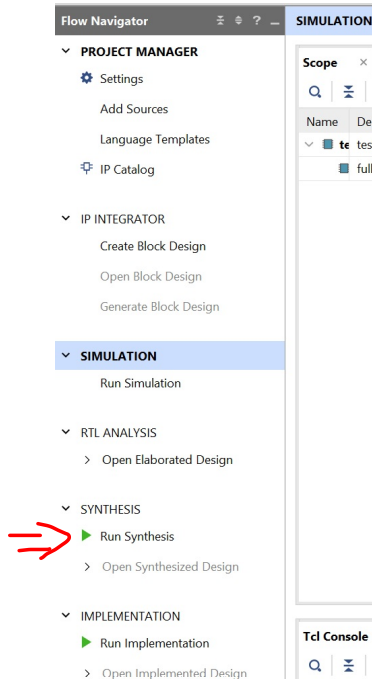
Le système indique l'assert qui n'a pas fonctionné.

```
full_adder.vhd x testbench_full_adder.vhd x Untitled 11 x
C:/FORMATION_SAFRAN/TP1_FullAdder/testbench_full_adder.vhd
142
143
144
145 ○ A <= '1';
146 ○ B <= '1';
147 ○ Cin <= '1';
148 ○ wait for 10 ns;
149
150 ○ --Valeurs des sorties attendues :
151 ○ -- Cout = 1
152 ○ -- S = 1
153 ○ assert Cout = '1'
154 ○ report "ERROR: Cout not equals to 1" severity failure;
155 ○ assert S = '0'
156 ○ report "ERROR: S not equals to 0" severity failure;
157
158
159 ○ A <= '0';
160 ○ B <= '1';
```

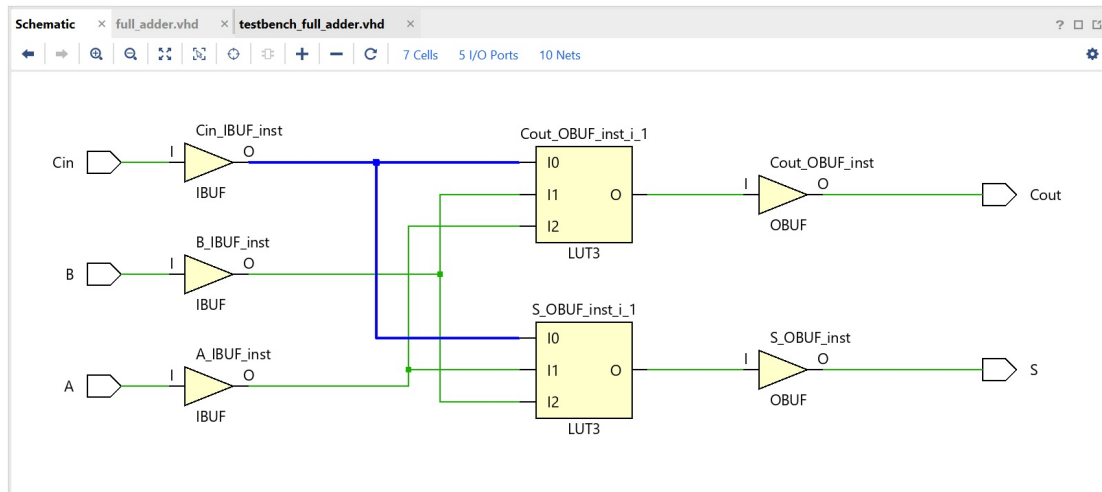
La simulation s'arrête au niveau de cet assert.



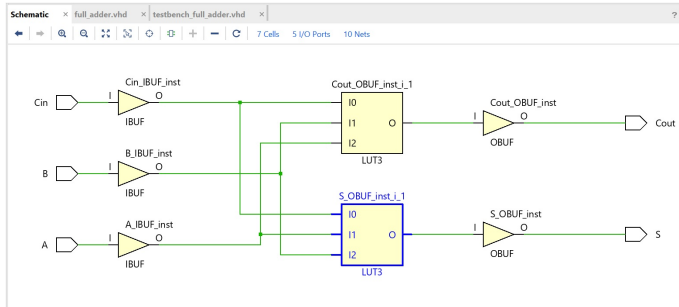
9. Dans l'onglet « Flow Navigator », cliquez « Run Synthesis » pour exécuter une synthèse de votre architecture.



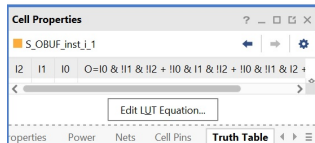
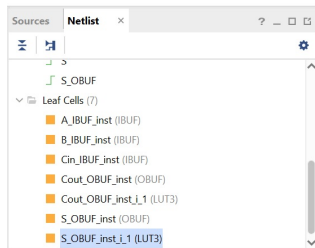
10. Déroulez le menu « Synthesis » et cliquez sur « Schematic » pour ouvrir la schématique. Où sont les portes logiques de l'architecture ? Vous pouvez mettre en surbrillance les fils en cliquant dessus pour mieux suivre leur chemin.



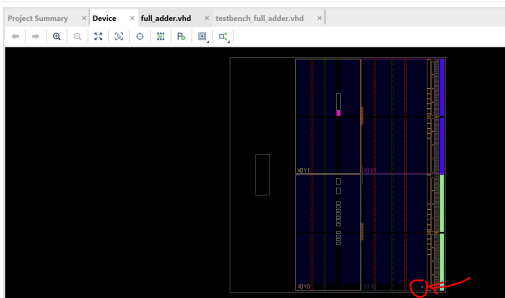
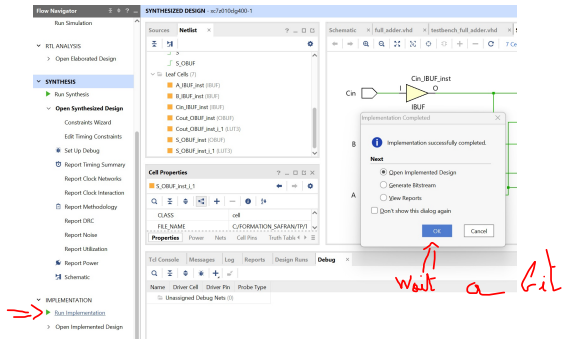
Pour retrouver la table de vérité (LUT) :
Cliquer sur la LUT



Puis, sélectionner Truth table dans Cell Properties



Cell Properties				
S_OBUF_inst_i_1				
I2	I1	I0	O = !I0 & !I1 & !I2 + !I0 & !I1 & I2 + !I0 & I1 & !I2 + !I0 & I1 & I2	
0	0	0	0	
0	0	1	1	
0	1	0	1	
0	1	1	0	
1	0	0	1	
1	0	1	0	
1	1	0	0	
1	1	1	1	



En zoomant

