

Matière : Systèmes sur puces SoC

Compte rendu du TP3

Travail réalisé par :
Mokhles El Benna

Développement d'un système purement logiciel Cas d'un processeur communiquant avec une machine hôte (UART)

Table des matières

Objectif :	3
Manipulation :	3
1- Création d'un nouveau projet :	3
Ajout des fichiers sources :	5
L'ajout d'une nouvelle source :	7

Conception d'un additionneur complet (full-adder) en VHDL dans Vivado

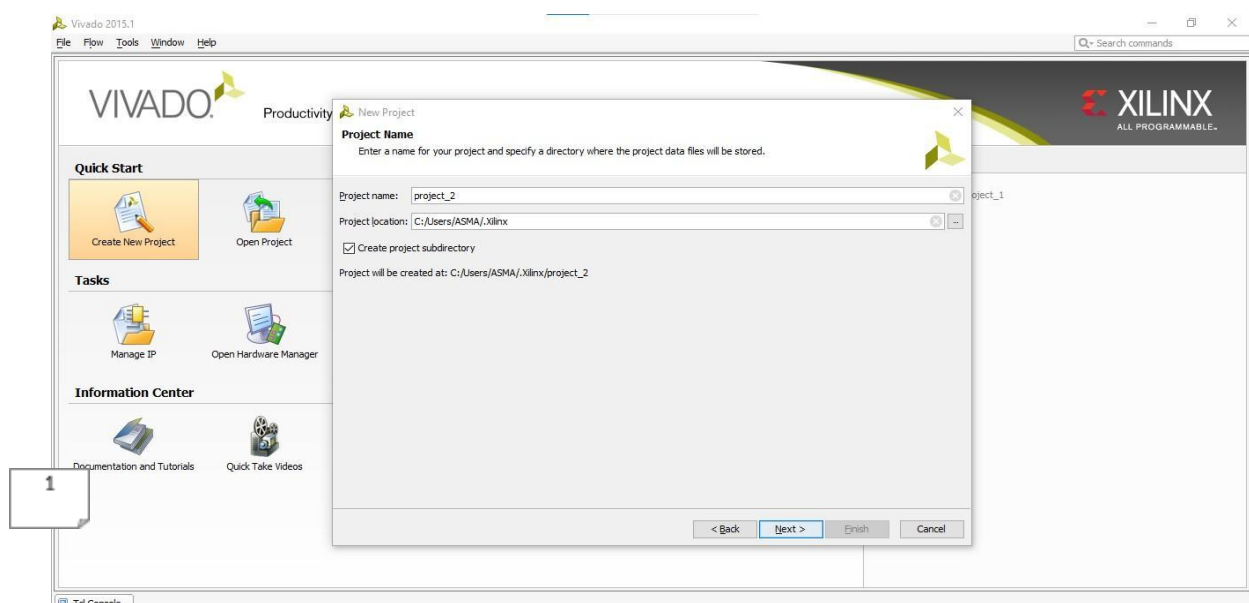
Objectif :

L'objectif de ce TP est de donner les principales clefs pour utiliser la suite d'outils Vivado de la société Xilinx. Vous allez, dans un premier temps, réaliser un simple additionneur complet (full adder) en utilisant un langage de description matérielle : le VHDL. Nous allons ensuite simuler le fonctionnement du full adder et le tester sur une carte de développement Digilent ZYBO Z7-10 intégrant un circuit FPGA Zynq™-7000. A la fin de cette manip, vous devez être capables de :

- Lancer l'outil Vivado
- Créer un nouveau projet
- Décrire un circuit combinatoire simple en VHDL
- Créer un fichier de contraintes .xdc
- Créer un test-bench pour votre description VHDL
- Simuler le comportement de votre description
- Faire la synthèse, le placement, le routage et la génération du bistream
- Reconfigurer le circuit FPGA de la carte pour tester votre système.

Manipulation :

1- Création d'un nouveau projet :



New Project

Add Sources

Specify HDL and netlist files, or directories containing HDL and netlist 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.

+

-

↑

↓

Press the + button to Add Files, Add Directories or Create File

☐ Scan and add RTL include files into project
☐ Copy sources into project
☒ Add sources from subdirectories

Target language: VHDL Simulator language: VHDL

< Back
Next >
Finish
Cancel

New Project

Default Part

Choose a default Xilinx part or board for your project. This can be changed later.

Select: Parts Boards

Filter

Vendor: All
 Display Name: All
 Board Rev: Latest

Reset All Filters

Search:

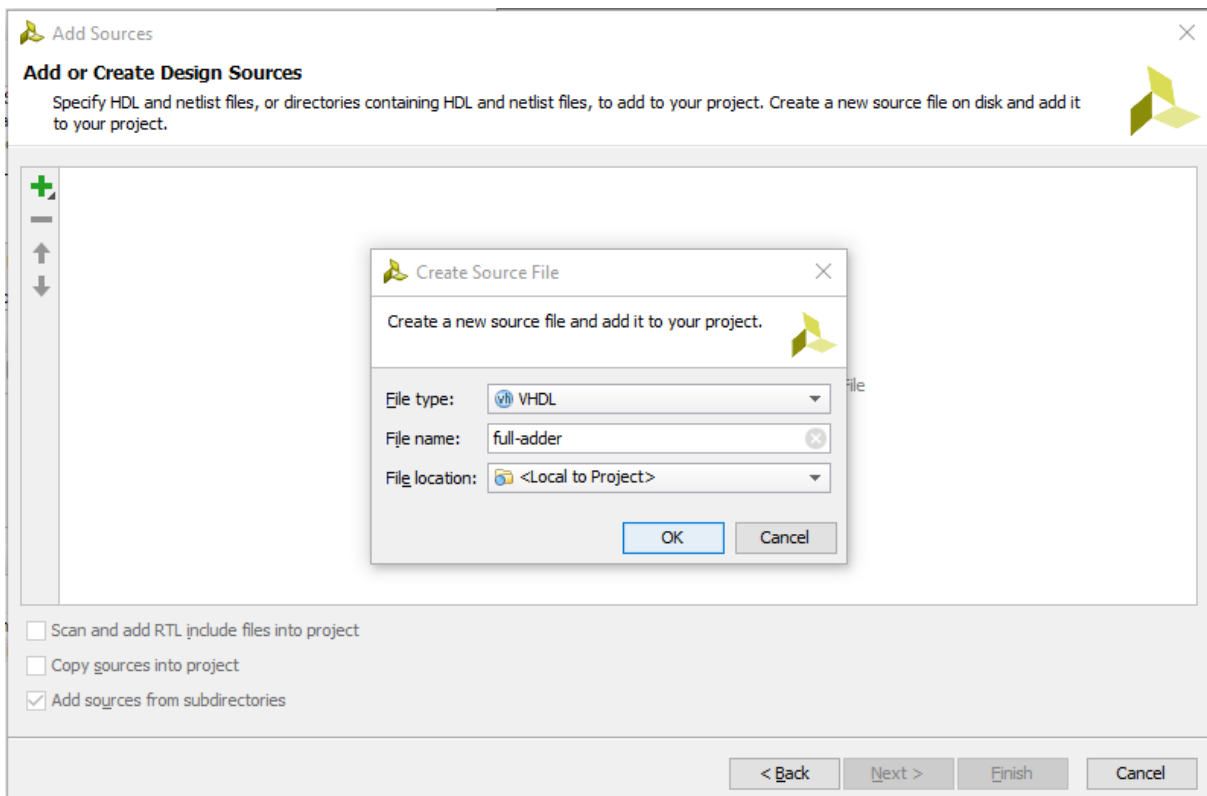
Display Name	Vendor	Board Rev	Part	I/O Pin Count	File Version	Av IC
Nexys Video	digilentinc.com	A.0	xc7a200tsbg484-1	484	1.1	28 ^
Sword	digilentinc.com	C.0	xc7k325tffg900-2	900	1.0	50
USB104 A7	digilentinc.com	B.2	xc7a100tcsq324-1	324	1.2	21
Zedboard	digilentinc.com	D.3	xc7z020d484-1	484	1.0	20
Zybo Z7-10	digilentinc.com	B.2	xc7z010d400-1	400	1.0	10
Zybo Z7-20	digilentinc.com	B.2	xc7z020d400-1	400	1.0	12
Zybo	digilentinc.com	B.4	xc7z010d400-1	400	2.0	10
ZedBoard Zynq Evaluation and Development Kit	em.avnet.com	d	xc7z020d484-1	484	1.3	20
Artix-7 AC701 Evaluation Platform	xilinx.com	1.1	xc7a200t400-2	676	1.2	40 v

< Back
Next >
Finish
Cancel



Université de Sousse
Institut Supérieure
d'informatique
et des techniques de communications

Ajout des fichiers sources :



La fenêtre suivante s'ouvre :

Project part: [Zybo Z7-10 \(xc7z010dq400-1\)](#)

Define Module

Define a module and specify I/O Ports to add to your source file.
For each port specified:
MSB and LSB values will be ignored unless its Bus column is checked.
Ports with blank names will not be written.

Module Definition

Entity name:

Architecture name:

I/O Port Definitions

Port Name	Direction	Bus	MSB	LSB
A	in	☒	0	0
B	in	☒	0	0
C	in	☒	0	0
SUM	out	☒	0	0
CARRY	out	☒	0	0

OK Cancel

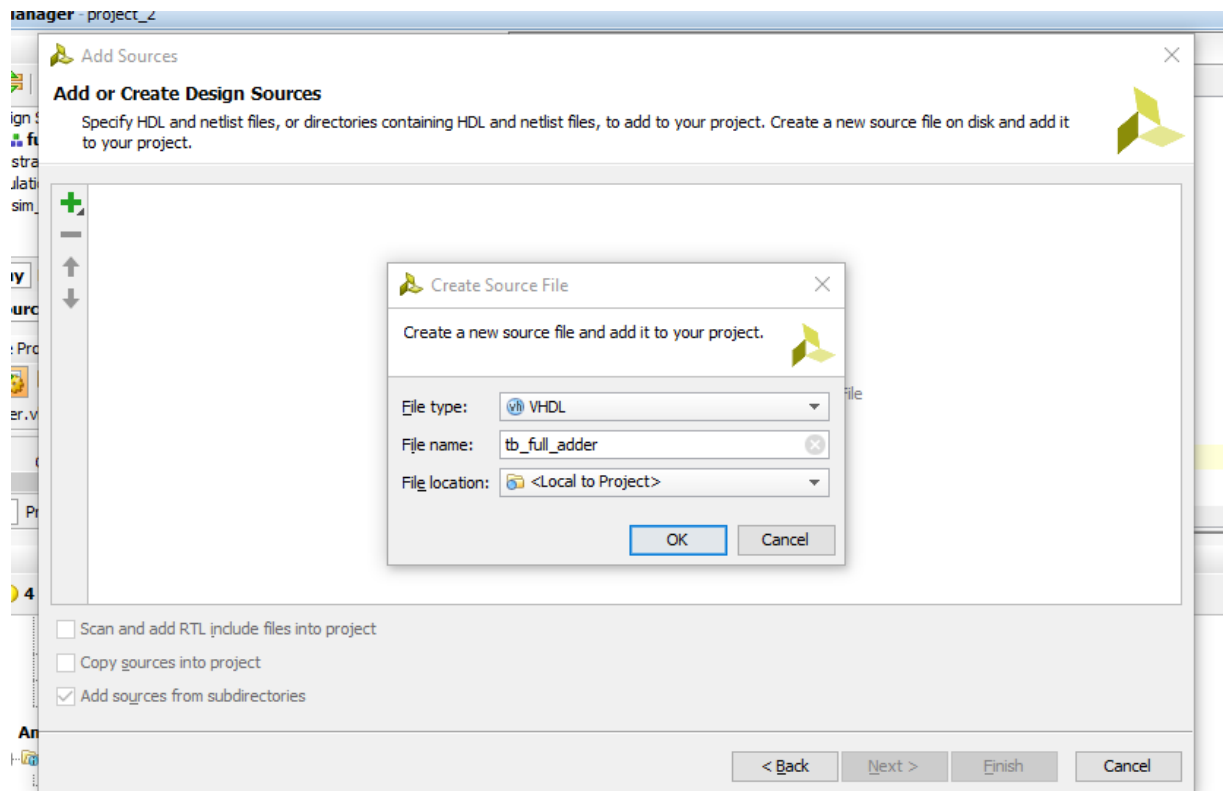
```

33
34 entity full_adder is
35     Port ( A : in STD_LOGIC;
36           B : in STD_LOGIC;
37           C : in STD_LOGIC;
38           SUM : out STD_LOGIC;
39           CARRY : out STD_LOGIC);
40 end full_adder;
41
42 architecture Behavioral of full_adder is
43     signal AXORB : std_logic;
44 begin
45     AXORB <= A xor B;
46     SUM <= AXORB xor C;
47     CARRY <= (A and B) or (C and AXORB);
48 end Behavioral;

```

ado/2015.1/data/ip/.

L'ajout d'une nouvelle source :



Define Module

Define a module and specify I/O Ports to add to your source file.
For each port specified:
MSB and LSB values will be ignored unless its Bus column is checked.
Ports with blank names will not be written.

Module Definition

Entity name:

Architecture name:

I/O Port Definitions

+	Port Name	Direction	Bus	MSB	LSB
—		in	☐	0	0
↑					
↓					

OK Cancel

```

22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24
25 -- Uncomment the following library declaration if using
26 -- arithmetic functions with Signed or Unsigned values
27 --use IEEE.NUMERIC_STD.ALL;
28
29 -- Uncomment the following library declaration if instantiating
30 -- any Xilinx leaf cells in this code.
31 --library UNISIM;
32 --use UNISIM.VComponents.all;
33
34 entity tb_full_adder is
35 -- Port ( );
36 end tb_full_adder;
37

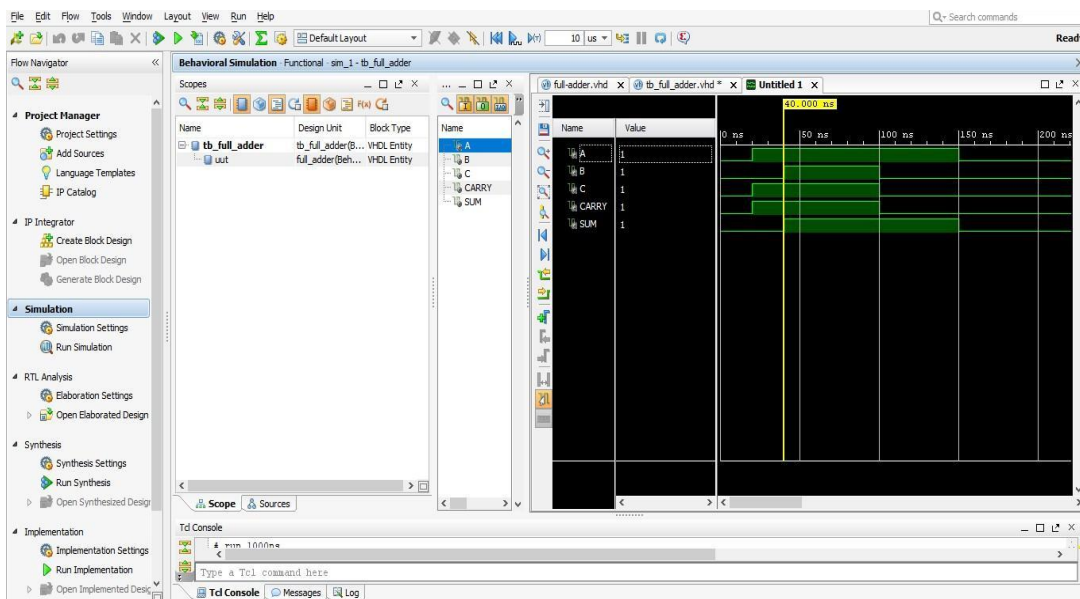
```

```

47 signal B:std_logic:='0';
48 signal C:std_logic:='0';
49 signal CARRY:std_logic;
50 signal SUM:std_logic;
51 begin
52 uut: full_adder PORT MAP (
53 A => A,
54 B => B,
55 C => C,
56 CARRY => CARRY,
57 SUM => SUM);
58 A <= '0','1' after 20 ns, '0' after 150 ns;
59 B <= '0','1' after 40 ns, '0' after 100 ns;
60 C <= '0','1' after 20 ns, '0' after 100 ns;
61 end Behavioral;
62

```

Le chronogramme ci-dessous s'affiche après la simulation :



Pour la simulation qui contient tous les combinaisons possibles, on doit modifier le programme de la façon suivante :

```

38 architecture Behavioral of tb_full_adder is
39 component full_adder
40 Port (A:in STD_LOGIC;
41       B:in STD_LOGIC;
42       C:in STD_LOGIC;
43       SUM:out STD_LOGIC;
44       CARRY:out STD_LOGIC);
45 end component;
46 signal A:std_logic:='0';
47 signal B:std_logic:='0';
48 signal C:std_logic:='0';
49 signal CARRY:std_logic;
50 signal SUM:std_logic;
51 begin
52 uut: full_adder PORT MAP (
53   A => A,
54   B => B,
55   C => C,
56   CARRY => CARRY,
57   SUM => SUM);
58 A <= '0','1' after 20 ns, '0' after 40 ns, '1' after 60 ns, '1' after 80 ns, '1' after 100 ns, '0' after 120 ns, '1'
59 B <= '0','1' after 40 ns, '0' after 80 ns, '1' after 120 ns, '0' after 180 ns;
60 C <= '0','1' after 80 ns, '0' after 180 ns;
61 end Behavioral;

```

On obtient le chronogramme suivant :

