

Compte rendu système sur puce

Année universitaire 2017-2018

Professeur :

Mme. Kh.nedia

Réaliser par:

Oueslati med melek

Zaïem donia

Tounsi safa

Meddeb hamza

Institute supérieur des études technologique en communication

Sommaire

Partie 1 :

Objectif

Matériel nécessaire

Logiciels nécessaires

Partie 2 :

Réalisation

Test

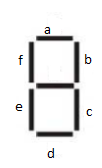
Partie 1 :

* Objectif

Notre objectif pour ce TP est de simuler un compteur modilo-16

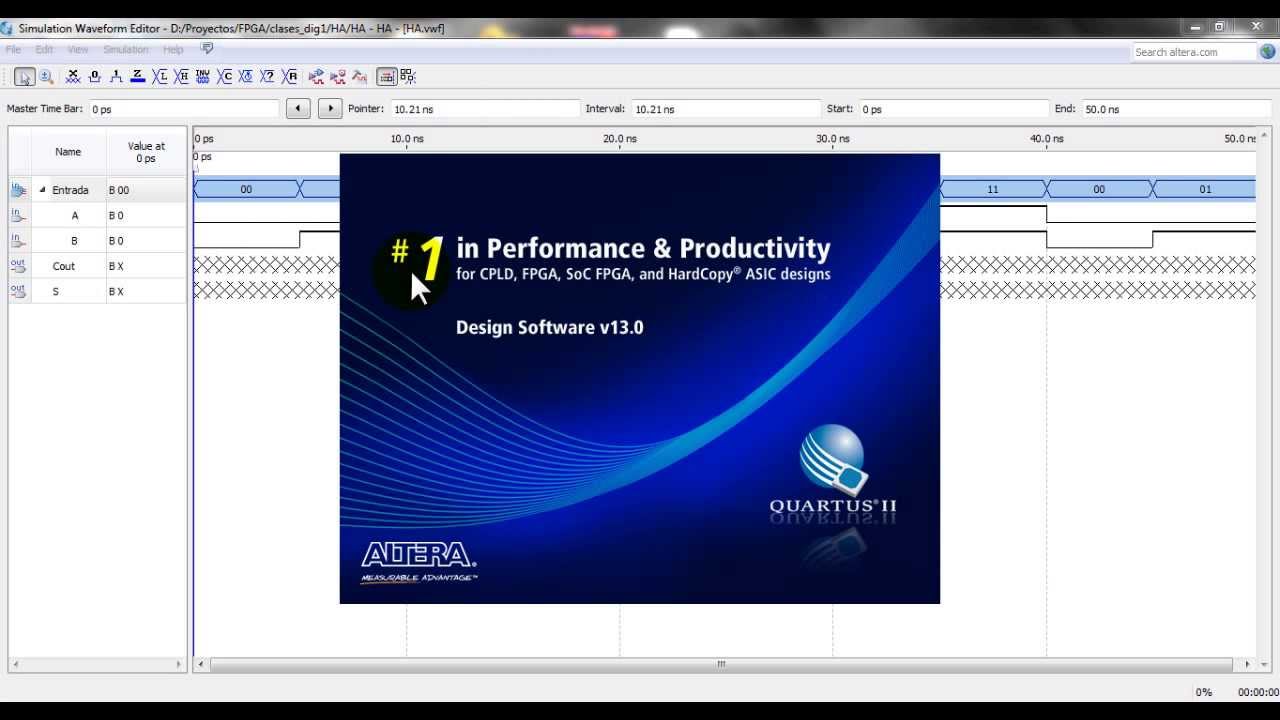
* Matériels

Carte Altera cyclone V GX

Afficheur sept segment

Sur la carte FPGA il existe 4 afficheurs à 7-segments. Chaque un de ces afficheurs est contrôlé par 7 signaux qui indiquent l’état du segment.

* Logiciel

Altera Quartus II est un logiciel de conception d'automates programmables produit par Altera. Quartus II permet l'analyse et la synthèse des conceptions HDL. Il inclut une implémentation de VHDL pour la description du matériel, et l'édition visuelle des circuits logiques.

https://www.altera.com/downloads/download-center.html

Partie 2

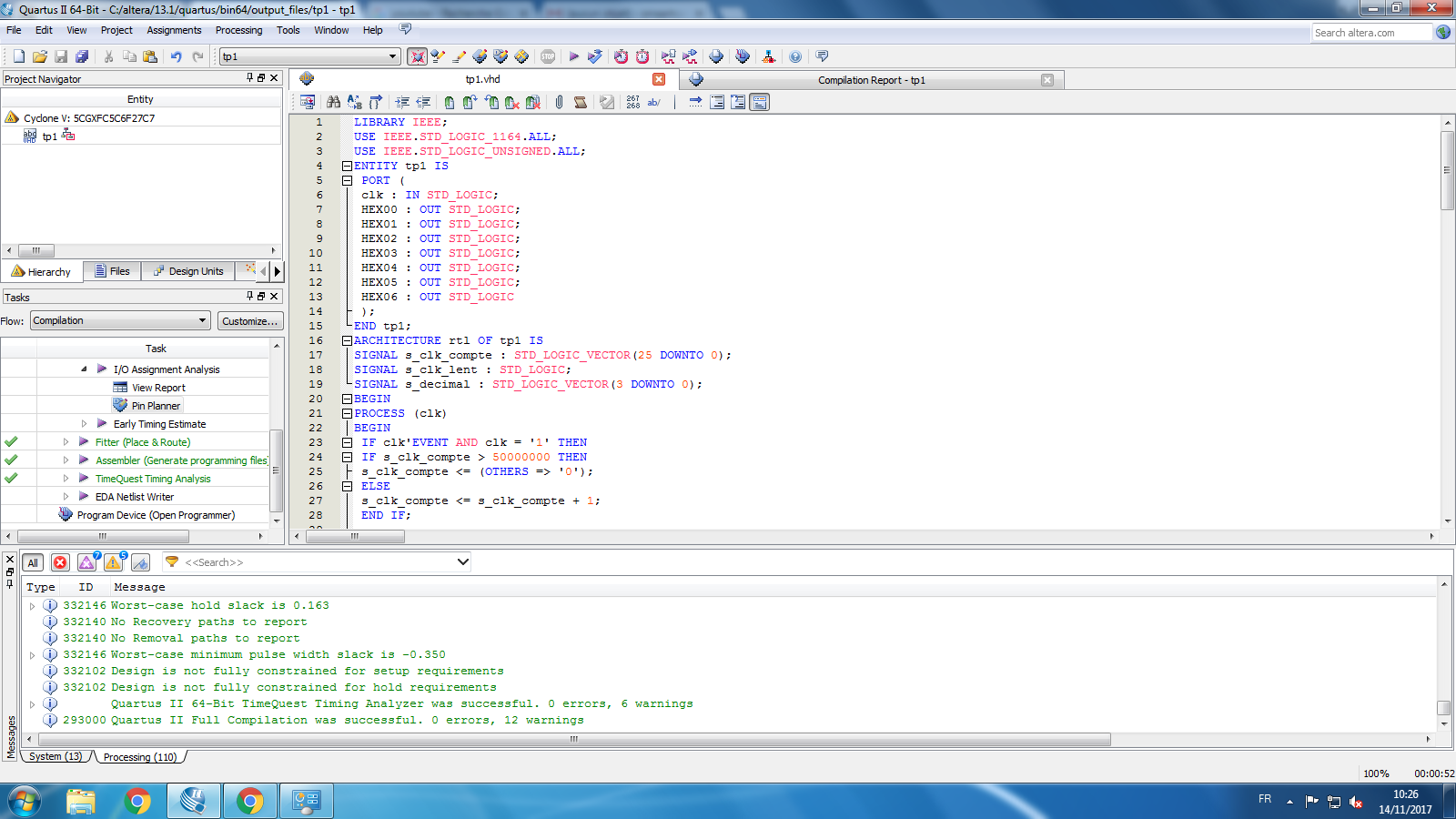
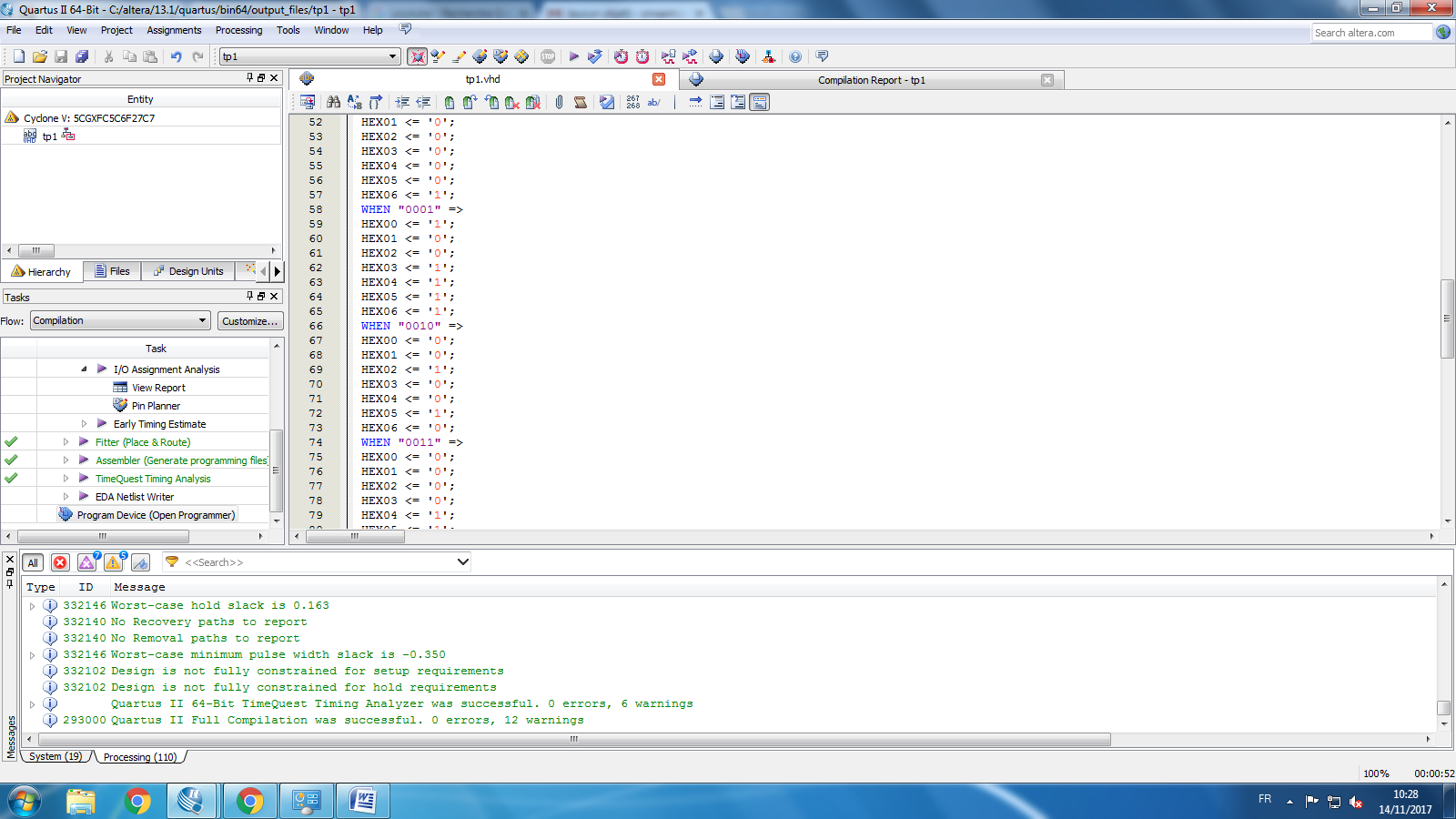
* Réalisation

Puis une nouvelle fenêtre permettant de configurer le projet apparaît :

* Puis choisir le circuit logique programmable que l’on souhaite utiliser : dans ce TP on a choisi d’utiliser la carte FPGA de la famille ALTERA 5CGXFC5C6F27C7N

Puis il faut saisir le code VHDL el l’enregistrer. Pour être sure qu’il n y a pas de faute d’écriture du code VHDL, il faut le compiler en cliquant sur le bouton

Pour faire une simulation il faut adjoindre des composants d'entrées/sorties (IO) permettant. Ces derniers se trouvent dans : Compile design 🡪 Analysis and synthesis 🡪 I/O…



LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY compteur1 IS

PORT (

clk : IN STD\_LOGIC;

HEX00 : OUT STD\_LOGIC;

HEX01 : OUT STD\_LOGIC;

HEX02 : OUT STD\_LOGIC;

HEX03 : OUT STD\_LOGIC;

HEX04 : OUT STD\_LOGIC;

HEX05 : OUT STD\_LOGIC;

HEX06 : OUT STD\_LOGIC

);

END compteur1;

ARCHITECTURE rtl OF compteur1 IS

SIGNAL s\_decimal : STD\_LOGIC\_VECTOR(3 DOWNTO 0);

BEGIN

PROCESS (clk)

BEGIN

IF clk'EVENT AND clk = '1' THEN

IF s\_decimal > 8 THEN

s\_decimal <= (OTHERS => '0');

ELSE

s\_decimal <= s\_decimal + 1;

END IF;

END IF ;

END PROCESS;

PROCESS (s\_decimal)

BEGIN

CASE s\_decimal IS

WHEN "0000" =>

HEX00 <= '0';

HEX01 <= '0';

HEX02 <= '0';

HEX03 <= '0';

HEX04 <= '0';

HEX05 <= '0';

HEX06 <= '1';

WHEN "0001" =>

HEX00 <= '1';

HEX01 <= '0';

HEX02 <= '0';

HEX03 <= '1';

HEX04 <= '1';

HEX05 <= '1';

HEX06 <= '1';

WHEN "0010" =>

HEX00 <= '0';

HEX01 <= '0';

HEX02 <= '1';

HEX03 <= '0';

HEX04 <= '0';

HEX05 <= '1';

HEX06 <= '0';

WHEN "0011" =>

HEX00 <= '0';

HEX01 <= '0';

HEX02 <= '0';

HEX03 <= '0';

HEX04 <= '1';

HEX05 <= '1';

HEX06 <= '0';

WHEN "0100" =>

HEX00 <= '1';

HEX01 <= '0';

HEX02 <= '0';

HEX03 <= '1';

HEX04 <= '1';

HEX05 <= '0';

HEX06 <= '0';

WHEN "0101" =>

HEX00 <= '0';

HEX01 <= '1';

HEX02 <= '0';

HEX03 <= '0';

HEX04 <= '1';

HEX05 <= '0';

HEX06 <= '0';

WHEN "0110" =>

HEX00 <= '0';

HEX01 <= '1';

HEX02 <= '0';

HEX03 <= '0';

HEX04 <= '0';

HEX05 <= '0';

HEX06 <= '0';

WHEN "0111" =>

HEX00 <= '0';

HEX01 <= '0';

HEX02 <= '0';

HEX03 <= '1';

HEX04 <= '1';

HEX05 <= '1';

HEX06 <= '1';

WHEN "1000" =>

HEX00 <= '0';

HEX01 <= '0';

HEX02 <= '0';

HEX03 <= '0';

HEX04 <= '0';

HEX05 <= '0';

HEX06 <= '0';

WHEN OTHERS =>

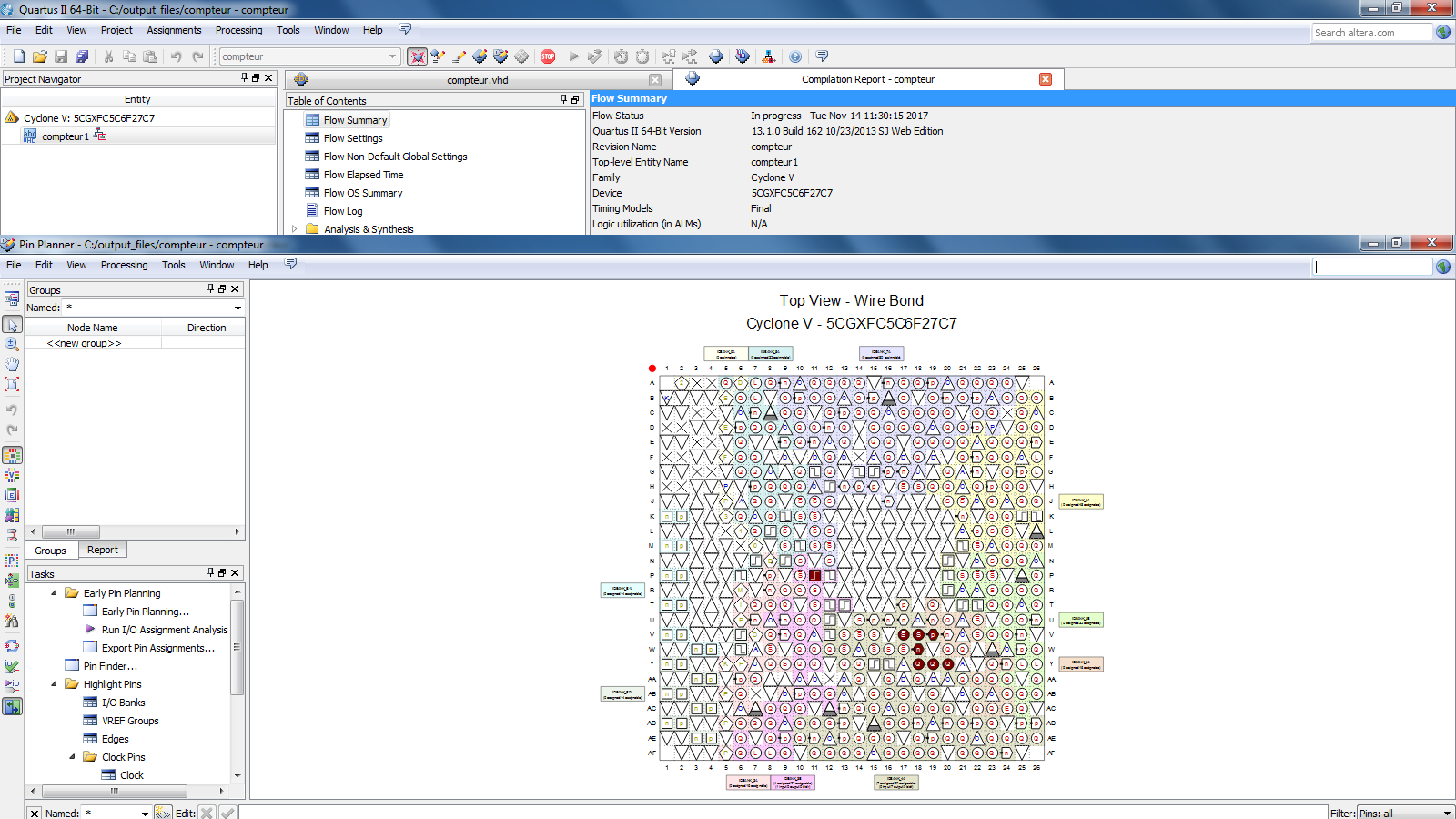
HEX00 <= '0';

HEX01 <= '0';

HEX02 <= '0';

HEX03 <= '0';

HEX04 <= '1';



* Test

