

Rapport Présenté par :

- Hamza BOUBNANE
- Tarik AZIKI
- Marouan El Ouardi

Intitulé

**Travaux Pratique Industriels (Circuit numériques 2 /
Automatismes Séquentiels)**

Encadrant : Dr. EL KAFAZI

SOMMAIRE :

Introduction générale.

Partie I : Etude Théorique

1. Introduction
2. Présentation des portes logiques utilisés. ...
3. Réalisation d'additionneur complet.

Partie II – Etude Pratique

1. Réalisation du schéma
2. Etablissement d'équations
3. Simulation et Résultat finale (Chronogramme)

Partie III – Réalisation d'un additionneur 4BITS :

1. Introduction
2. Réalisation du schéma
3. Table de vérité
4. Simulation et Résultat finale (Chronogramme)
5. Table de vérité d'un additionneur BCD

Partie I : Etude Théorique

1 - Introduction :

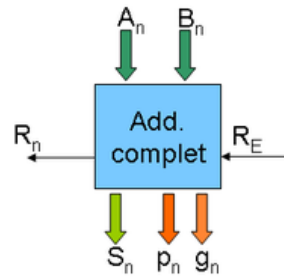
Ce chapitre a pour objectif de situer le projet dans son contexte général, ce projet a été réalisé avec le logiciel QUARTUS II qui est un logiciel de conception de dispositifs logiques programmables.

Le but de ce projet est de créer un additionneur complet avec AHDL qui est un langage de description du matériel Altera, à son tour qui est un langage de description du matériel propriétaire développé par Altera Corporation. L'AHDL est utilisé pour l'entrée de conception logique numérique pour les dispositifs logiques programmables complexes et les matrices de portes programmables d'Altera.

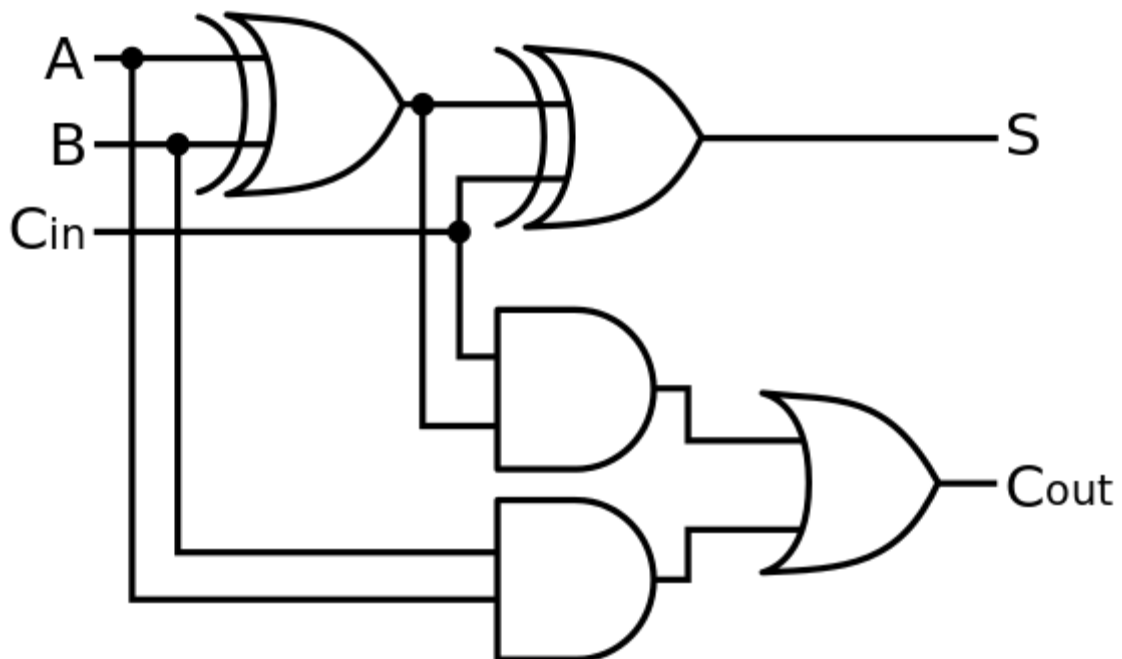
Nous allons utiliser les ports logiques afin de créer AHDL et un logigramme d'un additionneur complet (1 bit) et en vérifiant sa table de vérité et son chronogramme.

Un additionneur est un circuit logique permettant de réaliser une addition. Ce circuit est très présent dans les ordinateurs pour le calcul arithmétique mais également pour le calcul d'adresses, d'indice de tableau dans le processeur et qui est composé de plusieurs portes logiques.

Additionneur Complet :



Son logigramme :



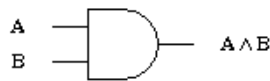
A, B, Cin sont des entrées.

S est la sortie et Cout est le retenu.

2 - Présentation des portes logiques utilisés :

Afin de réaliser un additionneur complet, nous aurons besoin des portes logiques programmées.

Porte Logique AND :



En AHDL :A & B

Porte Logique OR :



En AHDL :A # B

Porte logique XOR :



En AHDL :A \$ B

3 - Réalisation d'additionneur complet :

3-1) Table de vérité d'un additionneur complet :

A_i	B_i	C_n	S_i	C_i
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

C_n correspond à la retenue éventuelle de l'étage précédent.

C_i correspond à la retenue éventuelle de l'étage suivant.

3-2) Equations logiques de l'additionneur complet :

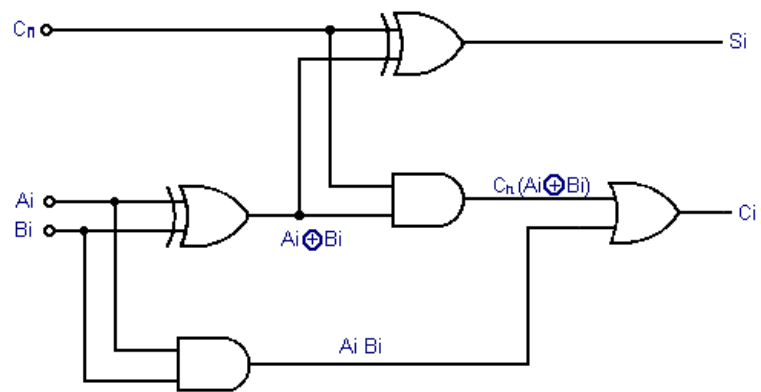
$$\begin{aligned} S_i &= \overline{A_i} \cdot \overline{B_i} \cdot C_n + \overline{A_i} \cdot B_i \cdot \overline{C_n} + A_i \cdot \overline{B_i} \cdot \overline{C_n} + A_i \cdot B_i \cdot C_n \\ &= \overline{A_i} \cdot (\overline{B_i} \cdot C_n + B_i \cdot \overline{C_n}) + A_i \cdot (\overline{B_i} \cdot \overline{C_n} + B_i \cdot C_n) \\ &= \overline{A_i} (B_i \oplus C_n) + A_i \cdot (\overline{B_i \oplus C_n}) \end{aligned}$$

$$S_i = A_i \oplus B_i \oplus C_n$$

$$\begin{aligned} C_i &= \overline{A_i} \cdot \overline{B_i} \cdot C_n + A_i \cdot \overline{B_i} \cdot C_n + A_i \cdot B_i \cdot \overline{C_n} + A_i \cdot B_i \cdot C_n \\ &= C_n \cdot (\overline{A_i} \cdot \overline{B_i} + A_i \cdot B_i) + A_i \cdot B_i (\overline{C_n} + C_n) \end{aligned}$$

$$C_i = C_n \cdot (A_i \oplus B_i) + A_i \cdot B_i$$

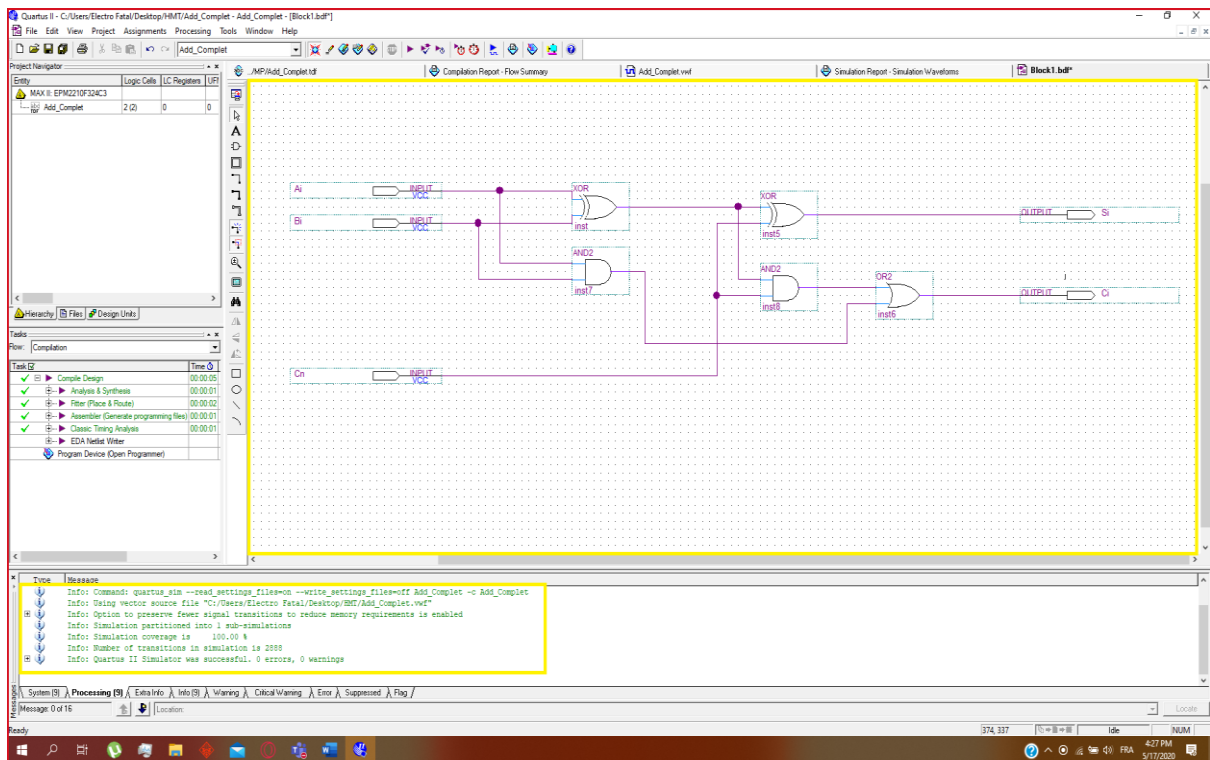
3-3) Logigramme d'additionneur complet :



Partie II – Etude Pratique :

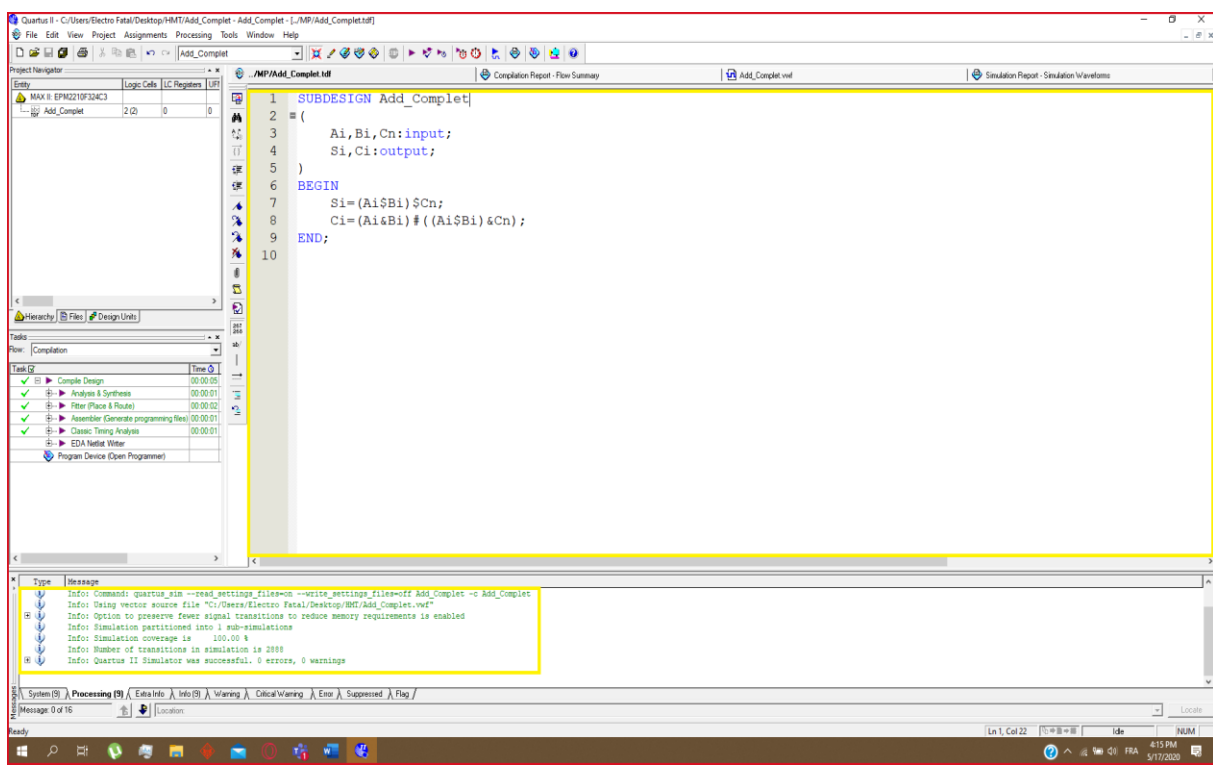
1-Réalisation du schéma :

Avant de tout commencer, la bonne réalisation du schéma vaut 70% du succès du projet, cependant, on compile le programme avant de le lancer afin de pourchasser les erreurs dues aux mauvaises manipulations des portes logiques indiquées dans la partie théorique.



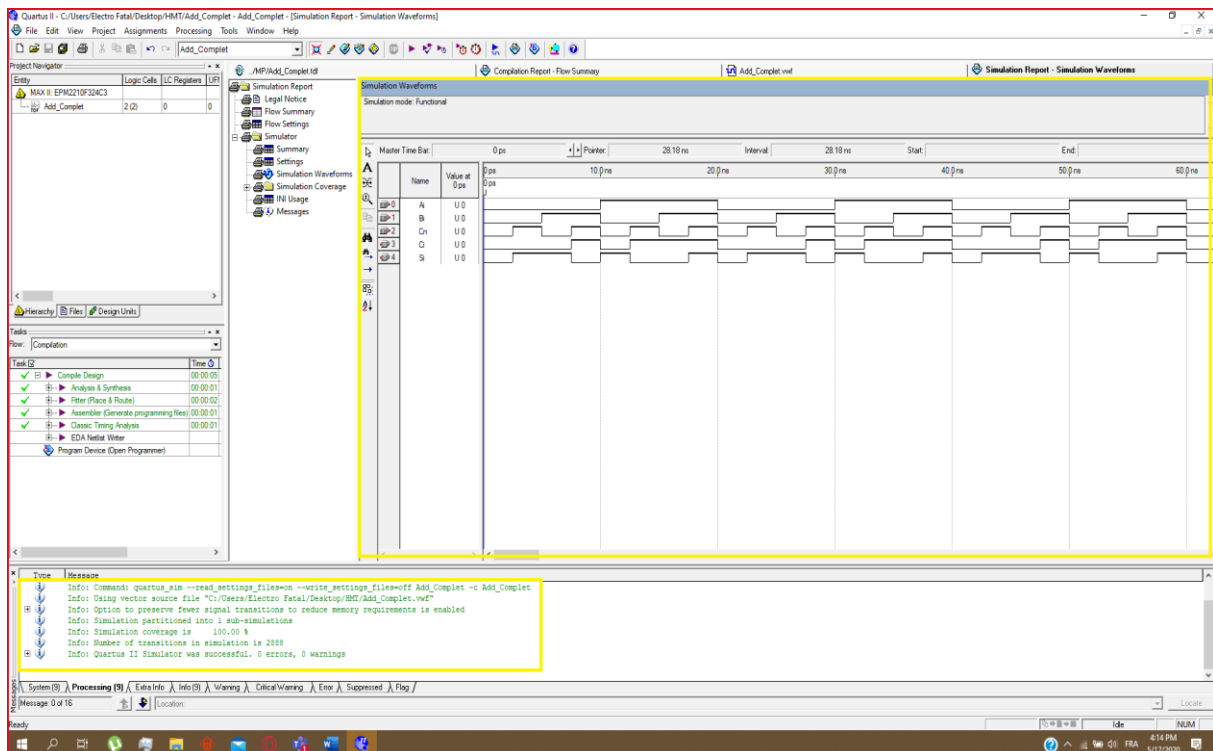
2) Etablissement d'équations :

D'après les résultats de la partie théorique, on établit les équations dans le programme en utilisant le langage AHDL. QUARTUS nous permet aussi de détecter les erreurs dans notre code.



3) Simulation et résultat final (Chronogramme) :

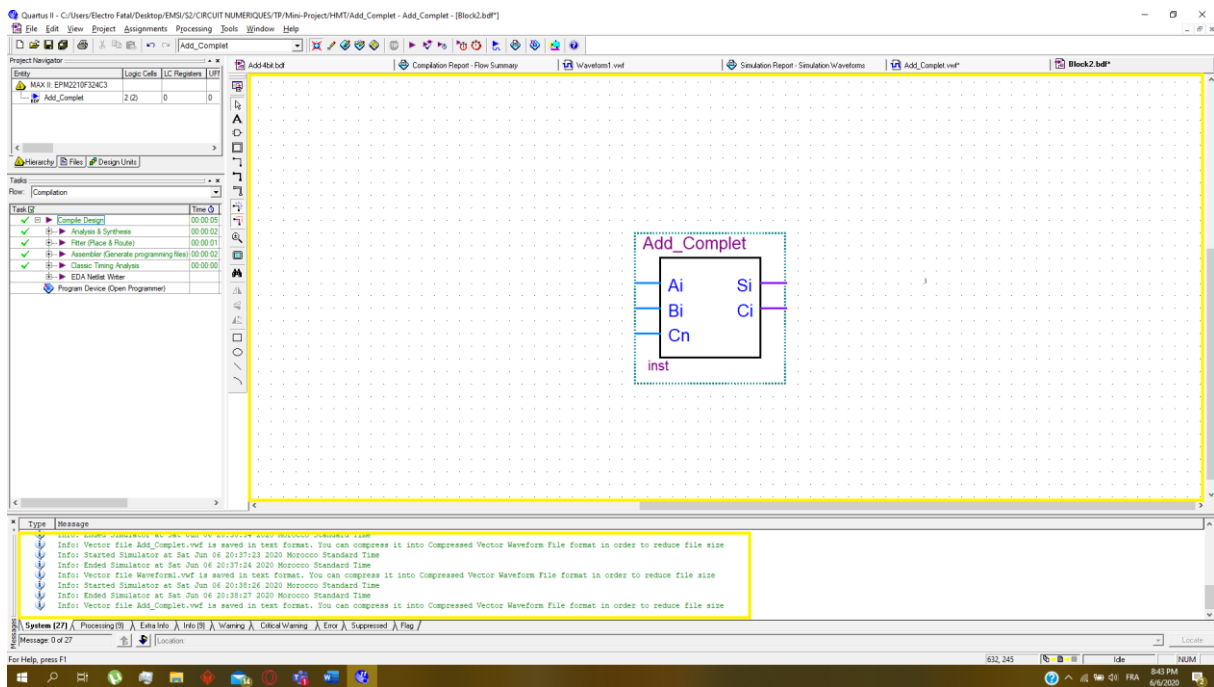
Pour conclure, on simule le tout dans un chronogramme comme indiqué ci-dessous, qui doit coopérer avec les résultats de la partie théorique.



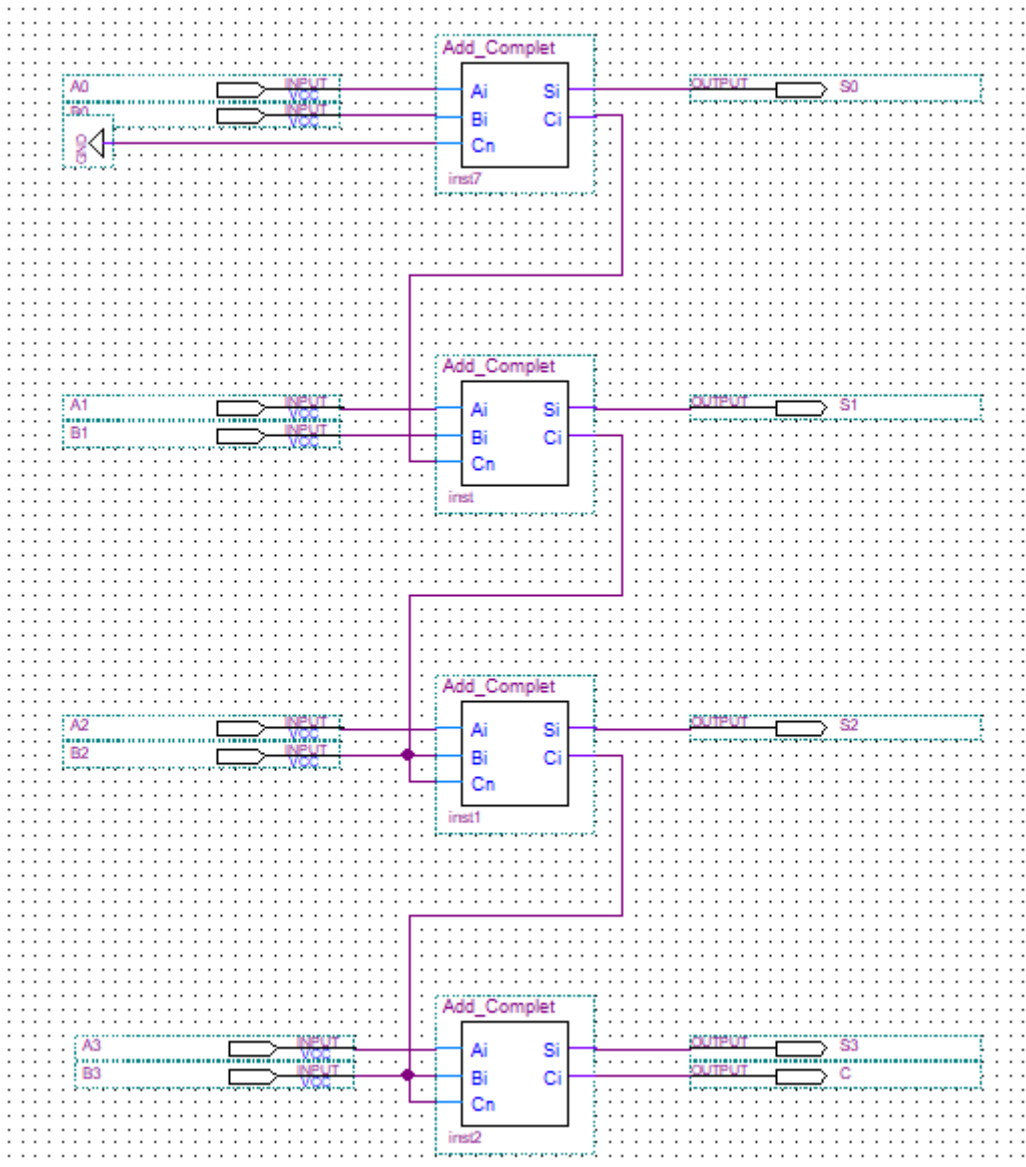
Partie III – Réalisation d'un additionneur 4BITS :

1-Introduction :

Après avoir réalisé un additionneur complet dans la 2^{ème} partie, on a converti le logigramme de cette dernière en un symbole sur Quartus comme indiqué ci-dessous :



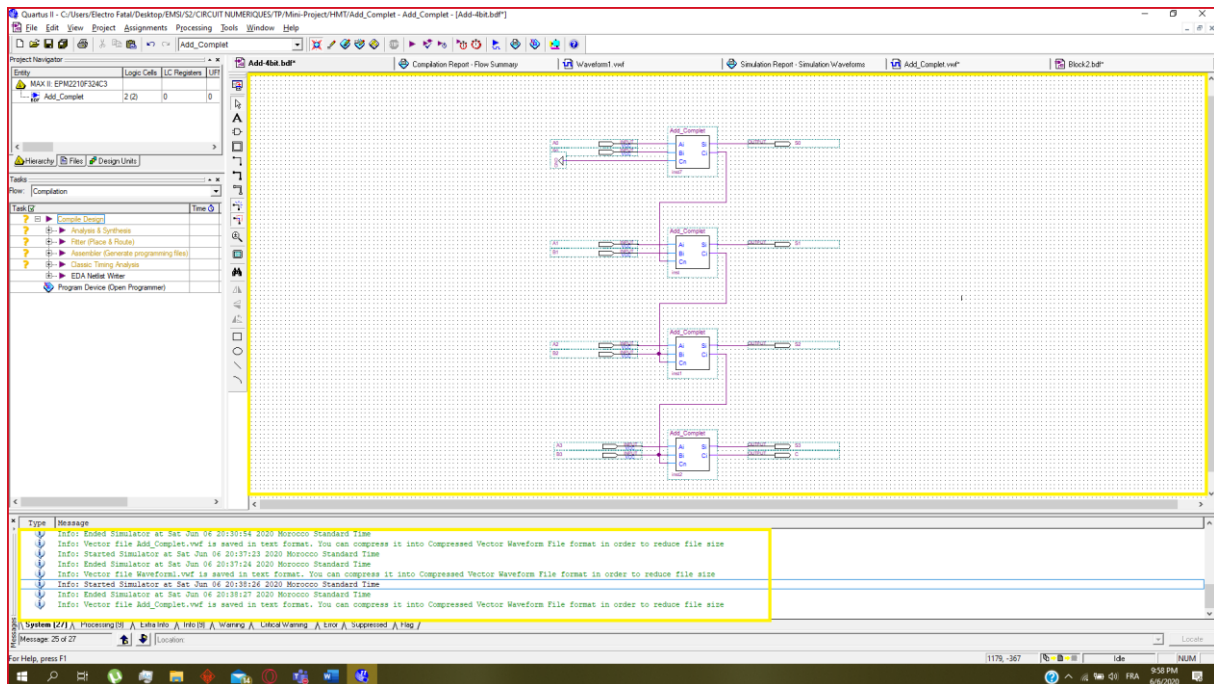
Pour réaliser un additionneur 4Bits, on a besoin de 4 additionneurs complets (1bit) donc 4 symboles. Chaque symbole contient 2 entrées principales (Ai – Bi) et une entrée secondaire (Cn) avec une sortie (Si) et une retenue (Ci).



Au tout début, on lie la première Cn au GND (ground), ensuite on lie la première retenue (C0) à l'entrée du deuxième additionneur complet (Cn), et on garde dans un coté la sortie S0 considérée comme MSB, puis on relie la retenue de cet additionneur à l'entrée de l'additionneur qui le suit etc... jusqu'à la dernière retenue qui sera réservée au LSB.

2.Réalisation du schéma :

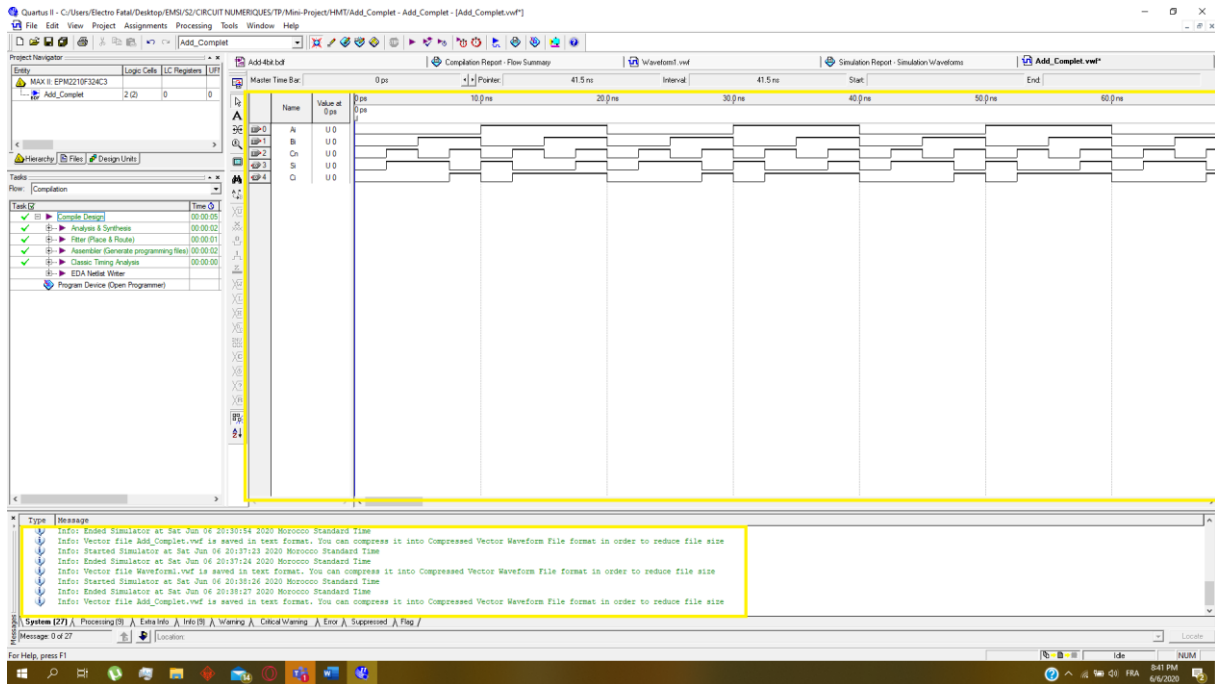
On réalise le schéma de l'additionneur 4 bits à l'aide des réformes indiquées en introduction, puis on compile le programme.



3.Tableau de vérité :

Ai	Bi	Cn	Si	Ci
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

4.Simulation et Résultat finale (Chronogramme) :



Pour conclure, on simule le tout dans un chronogramme comme indiqué ci-dessous, qui doit coopérer avec les résultats de la partie théorique.

5. Table de vérité d'un additionneur BCD :

RESULTAT DE L'ADDITION BINAIRE						RESULTAT DE L'ADDITION BCD						
⊕	Résultat possible	R4	Σ4	Σ3	Σ2	Σ1	⊕	R	S4	S3	S2	S1
	0	0	0	0	0	0		0	0	0	0	0
	1	0	0	0	0	1		0	0	0	0	1
	2	0	0	0	1	0		0	0	0	1	0
	3	0	0	0	1	1		0	0	0	1	1
	4	0	0	1	0	0		0	0	1	0	0
	5	0	0	1	0	1		0	0	1	0	1
	6	0	0	1	1	0		0	0	1	1	0
	7	0	0	1	1	1		0	0	1	1	1
	8	0	1	0	0	0		0	1	0	0	0
	9	0	1	0	0	1		0	1	0	0	1
	10	0	1	0	1	0		1	0	0	0	0
	11	0	1	0	1	1		1	0	0	0	1
	12	0	1	1	0	0		1	0	0	1	0
	13	0	1	1	0	1		1	0	0	1	1
	14	0	1	1	1	0		1	0	1	0	0
	15	0	1	1	1	1		1	0	1	0	1
	16	1	0	0	0	0		1	0	1	1	0
	17	1	0	0	0	1		1	0	1	1	1
	18	1	0	0	1	0		1	1	0	0	0

Remerciement :

Nous tenons à remercier dans un premier temps, toute l'équipe pédagogique de l'EMSI et les intervenants professionnels responsables de la formation TP Industriel / Automatismes, pour avoir assuré la partie théorique de celle-ci.

Nous remercions également Dr. EL KAFAZI pour l'aide et les conseils concernant les missions évoquées dans ce rapport, qu'il nous apporté lors des différents suivis.