

École Polytechnique de Montréal
Département de génie informatique et génie logiciel

INF1500

Logique des systèmes numériques

Laboratoire 2 : Rapport sur la simulation et implémentation des circuits en
logique combinatoire

Soumis par

Wassim Guellati

Tamer Arar

Yacine Sehboub

Section de labo #2

8 février 2016

Critère	Points
Schémas des circuits	/1
Stratégies de test	/1
Simulation des circuits	/2
Fonctionnement sur la carte FPGA	/1.5
Rapport	/1.5
Total	/7

1. Introduction

Ce laboratoire a pour but de concevoir des modules simples de logique combinatoire puis de les implémenter et les simuler sur une carte FPGA. Les principes de conception des circuits combinatoires seront utilisés à ces fins.

Nous avons commencé avec un système et nous avons dû comprendre sa logique pour ensuite déterminer les tables de vérités de ses composantes. De ces tables, nous avons produits des tables de Karnaugh qui nous ont permis de trouver des expressions simplifiées des sorties des circuits.

Par la suite, sur l'environnement d'Active-HDL, nous avons déduit les schémas de nos circuits. Nous avons pu ainsi les simuler pour finalement les implanter sur la carte FPGA.

2. Expression des sorties

Expression de la sortie B :

A1A0 \ A3A2				
	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	0	1	0	1
10	1	0	1	0

$$F_{min} = (A_3'A_2'A_1'A_0) + (A_3'A_2'A_1A_0') + (A_3'A_2A_1'A_0') + (A_3'A_2A_1A_0) + (A_3A_2'A_1'A_0') + (A_3A_2'A_1A_0) + (A_3A_2A_1'A_0) + (A_3A_2A_1A_0')$$

$$= A_3'A_2'A_1A_0' + A_3'A_2'A_1'A_0 + A_3'A_2A_1'A_0' + A_3'A_2A_1A_0 + A_3A_2'A_1'A_0' + A_3A_2'A_1A_0 + A_3A_2A_1'A_0' + A_3A_2A_1A_0$$

$$= A_3'A_2' (A_1A_0' + A_1'A_0) + A_3'A_2 (A_1'A_0' + A_1A_0) + A_3A_2' (A_1'A_0' + A_1A_0) + A_3A_2 (A_1A_0' + A_1'A_0)$$

$$= A_3'(A_2'(A_1 \oplus A_0) + A_2 (A_1 \oplus A_0)') + A_3 (A_2'(A_1 \oplus A_0)' + A_2 (A_1 \oplus A_0))$$

$$= A_3'(A_2 \oplus (A_1 \oplus A_0)) + A_3 (A_2 \oplus (A_1 \oplus A_0)')$$

$$= A_3 + (A_2 \oplus (A_1 \oplus A_0))$$

Table de vérité correspondante :

3	T1. Tables de vérité pour le circuit Bit de Parité						
4	A3	A2	A1	A0	B	Min	Max
5	0	0	0	0	0	0	$A3'+A2'+A1'+A0'$
6	0	0	0	1	1	$A3'A2'A1'A0$	
7	0	0	1	0	1	$A3'A2'A1A0'$	
8	0	0	1	1	0		$A3'+A2'+A1+A0$
9	0	1	0	0	1	$A3'A2A1'A0'$	
10	0	1	0	1	0		$A3'+A2+A1'+A0$
11	0	1	1	0	0		$A3'+A2+A1+A0'$
12	0	1	1	1	1	$A3'A2A1A0$	
13	1	0	0	0	0	1	$A3A2'A1'A0'$
14	1	0	0	1	0	0	$A3+A2'+A1'+A0$
15	1	0	1	0	0	0	$A3+A2'+A1+A0'$
16	1	0	1	1	1	1	$A3A2'A1A0$
17	1	1	0	0	0	0	$A3+A2+A1'+A0'$
18	1	1	0	1	1	1	$A3A2A1'A0$
19	1	1	1	0	1	1	$A3A2A1A0'$
20	1	1	1	1	1	0	$A3+A2+A1+A0$

Expression de la sortie C :

Expression de la sortie C0 :

A1A0 \ A3A2				
	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	0	1	0	1
10	1	0	1	0

$$F_{\min} = (A3'A2'A1'A0) + (A3'A2'A1A0') + (A3'A2A1'A0') + (A3'A2A1A0) + (A3A2'A1'A0') + (A3A2'A1A0) + (A3A2A1'A0) + (A3A2A1A0')$$

$$= A3 \oplus (A2 \oplus (A1 \oplus A0))$$

Expression de la sortie C1 :

A1A0 \ A3A2	00	01	11	10
00	0	0	1	0
01	0	1	1	1
11	1	1	0	1
10	0	1	1	1

$$F = A3A2A1' + A2A0A1' + A0A3A2' + A1A3A0' + A2A1A3' + A1A0A3'$$

$$F = A3A2A1' + A2A1'A0 + A3A2'A0 + A3A1A0' + A3'A2A1 + A3'A1A0$$

$$F = A2 (A3A1' + A3'A1) + A1 (A3A0' + A3'A0) + A0 (A3A2' + A2A1')$$

$$F = A2 (A3 \oplus A1) + A1 (A3 \oplus A0) + A0 (A3A2' + A2A1')$$

Expression de la sortie C2 :

A1A0 \ A3A2	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	1	0
10	0	0	0	0

$$F = A3A2A1A0$$

Table de vérité correspondante :

22	T1. Tables de vérité pour le circuit Somme de Contrôle						
23	A3	A2	A1	A0	C2	C1	C0
24	0	0	0	0	0	0	0
25	0	0	0	1	0	0	1
26	0	0	1	0	0	0	1
27	0	0	1	1	1	0	0
28	0	1	0	0	0	0	1
29	0	1	0	1	1	0	0
30	0	1	1	0	0	1	0
31	0	1	1	1	1	1	1
32	1	0	0	0	0	0	1
33	1	0	0	1	1	0	0
34	1	0	1	0	0	1	0
35	1	0	1	1	1	1	1
36	1	1	0	0	0	1	0
37	1	1	0	1	1	1	1
38	1	1	1	0	0	1	1
39	1	1	1	1	1	0	0

Expression de la sortie D1 pour le circuit DEP :

		B2	
B1		0	1
	0	1	0
	1	0	1

$$F = B1'B2' + B1B2 = (B1 \oplus B2)'$$

Table de vérité correspondante :

78	T4. Tables de vérité pour le circuit DEP		
79	B1	B2	D1
80	0	0	1
81	0	1	0
82	1	0	0
83	1	1	1

Expression des sorties de l'injecteur d'erreur :

Déduites selon la table de vérité suivante

41

T3. Tables de vérité pour le circuit injecteur d'erreur									
En	A3	A2	A1	A0	E3	E2	E1	E0	
43	0	0	0	0	0	0	0	0	
44	0	0	0	0	1	0	0	1	
45	0	0	0	1	0	0	0	1	
46	0	0	0	1	1	0	0	1	
47	0	0	1	0	0	0	1	0	
48	0	0	1	0	1	0	1	0	
49	0	0	1	1	0	0	1	1	
50	0	0	1	1	1	0	1	1	
51	0	1	0	0	0	1	0	0	
52	0	1	0	0	1	1	0	0	
53	0	1	0	1	0	1	0	1	
54	0	1	0	1	1	1	0	1	
55	0	1	1	0	0	1	1	0	
56	0	1	1	0	1	1	1	0	
57	0	1	1	1	0	1	1	1	
58	0	1	1	1	1	1	1	1	
59	1	0	0	0	0	0	0	1	
60	1	0	0	0	1	0	0	0	
61	1	0	0	1	0	0	0	1	
62	1	0	0	1	1	0	0	1	
63	1	0	1	0	0	0	1	0	
64	1	0	1	0	1	0	1	0	
65	1	0	1	1	0	0	1	1	
66	1	0	1	1	1	0	1	0	
67	1	1	0	0	0	1	0	0	
68	1	1	0	0	1	1	0	0	
69	1	1	0	1	0	1	0	1	
70	1	1	0	1	1	1	0	0	
71	1	1	1	0	0	1	1	0	
72	1	1	1	0	1	1	1	0	
73	1	1	1	1	0	1	1	1	
74	1	1	1	1	1	1	1	0	

$$E0 = En'A0 + EnA0' = En \oplus A0$$

$$E3 = A3$$

$$E2 = A2$$

$$E1 = A1$$

3. Schémas et description des modules

Module, Bit de parité :

Après avoir fait les tables Karnaugh de chaque module à partir de leurs tables de vérités, on cherchera alors leurs expressions simplifiées. Ainsi, grâce à ces dernières, on obtient alors plusieurs schémas. Voici le schéma du circuit du module Bit de Parité :

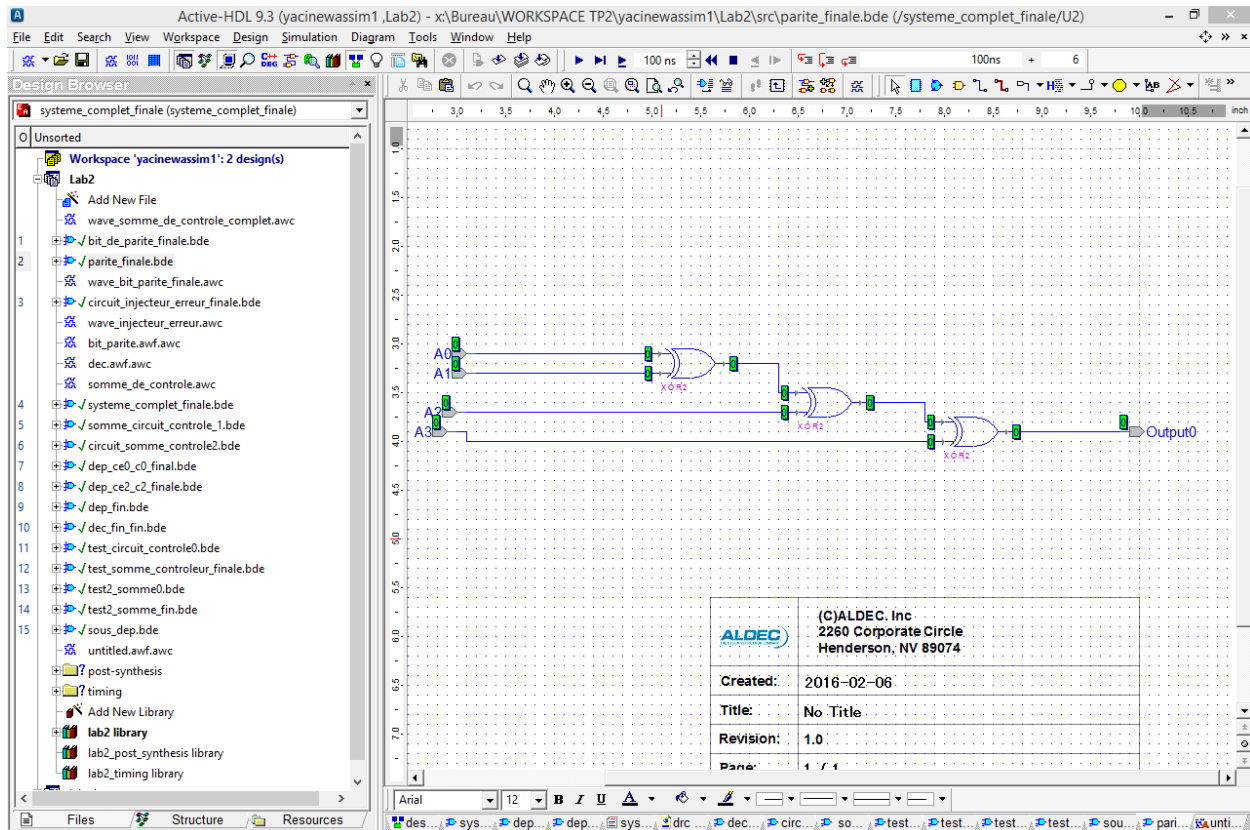


Figure 1 : Schéma du circuit du module Bit de Parité

Ce module permet de calculer la parité d'une variable d'entrée sur 4 bits. A l'aide du tableau de Karnaugh, on parvient à obtenir une formule. En sortant les facteurs commun et en utilisant le OU EXCLUSIF [(NON A ET B) OU (A ET NON B)] on arrive à la formule suivante : $A3 \oplus (A2 \oplus (A1 \oplus A0))$

L'OU EXCLUSIF représente une porte XOR, comme on peut le voir dans le schéma du circuit de Bit de parité. On a donc une porte XOR2 entre les entrées A0 et A1, on obtient en sortie $A0 \oplus A1$. Puis une porte XOR2 entre

($A0 \oplus A1$), et A2, pour obtenir en sortie $A2 \oplus (A0 \oplus A1)$. Et enfin une porte XOR2 entre A3 et $A2 \oplus (A0 \oplus A1)$ pour obtenir en sortie la formule finale $A3 \oplus (A2 \oplus (A1 \oplus A0))$.

Module somme de contrôle sur trois bits :

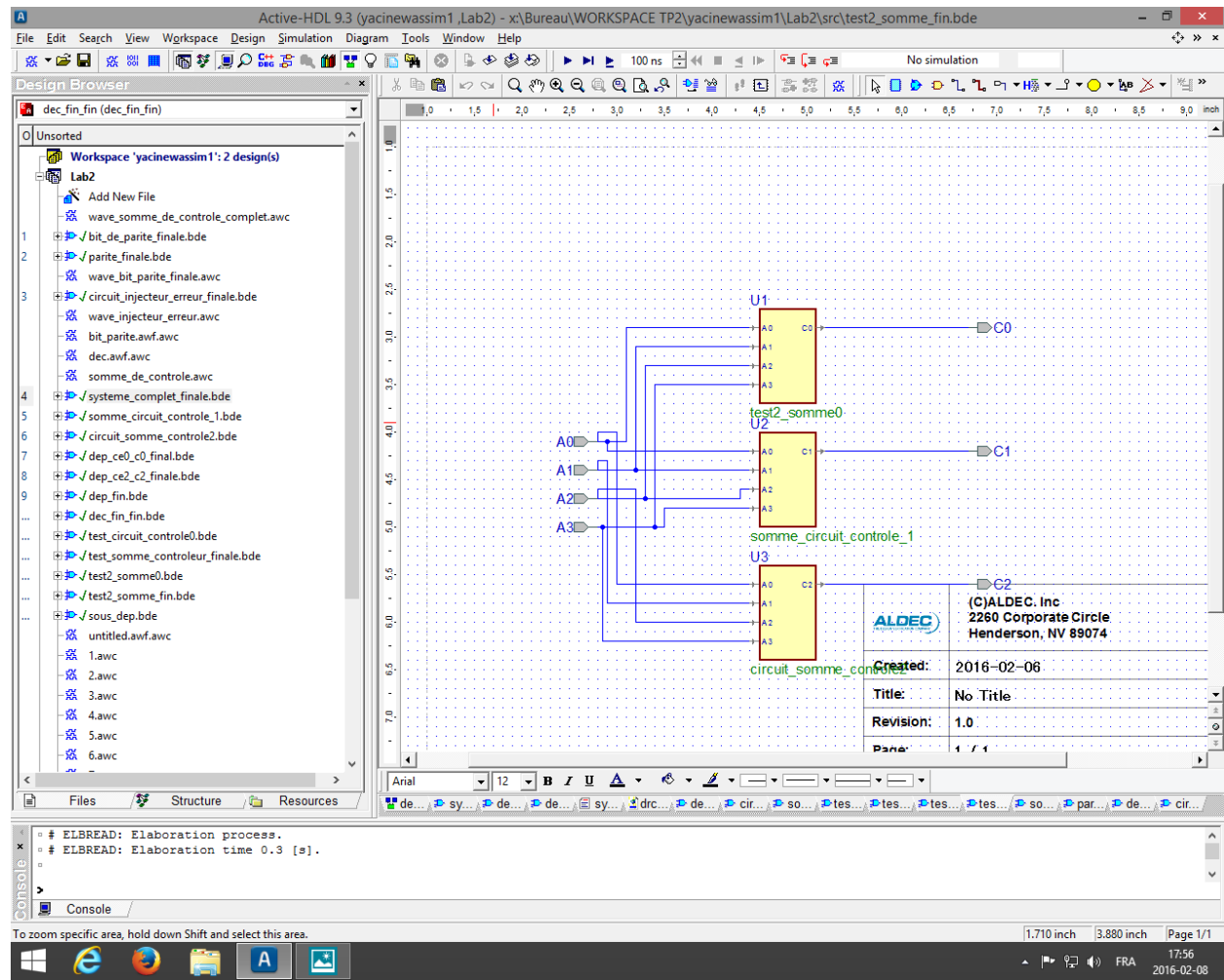


Figure 2 : Schéma du circuit somme de contrôle sur 3 bits regroupant C0, C1 et C2.

Ensuite, le module de somme de contrôle sur trois bits (SCDB) a une fonction très précise. Il transforme l'entrée d'une variable de 4 bits sur 3 bits. Il additionne entre elles la valeur de chaque bit, par exemple :

- Si l'entrée est : 1011, on obtient en sortie : 3
- Si l'entrée est 1001, on obtient en sortie 2 etc.

Ce module comporte 3 sorties, chacune avec une formule différente.

On retrouve alors dans chaque bloc de ce circuit un module

Le module en sortie C0 :

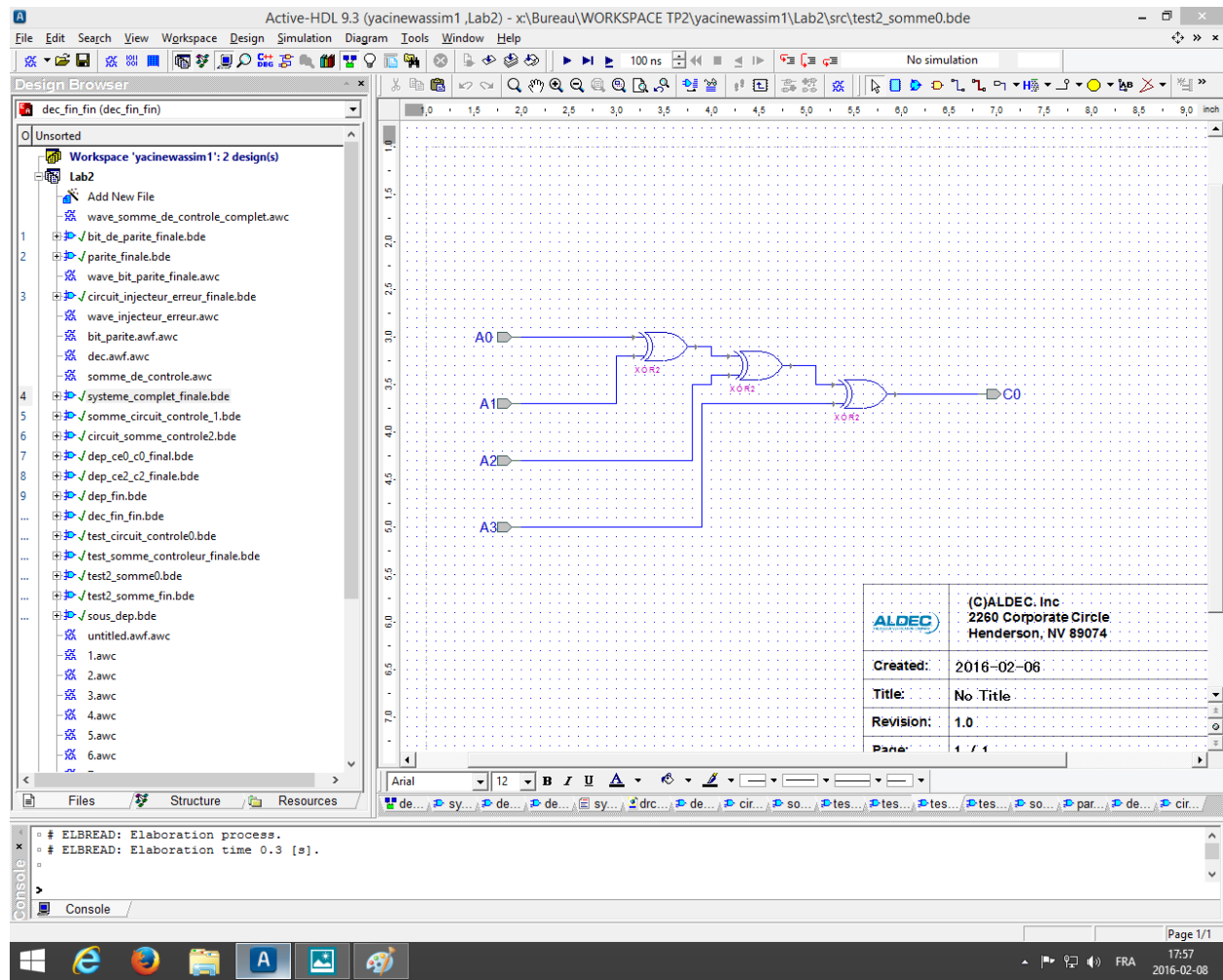


Figure3 : Schéma du circuit du module en sortie C0

La formule est la suivante : $A3 \oplus (A2 \oplus (A1 \oplus A0))$, c'est la même formule que celle du bit de parité, la logique pour réaliser cette partie là du circuit est donc la même.

L'OU EXCLUSIF \oplus représente une porte XOR, On a donc une porte XOR2 entre les entrées A0 et A1, on obtient en sortie $A0 \oplus A1$. Puis une porte XOR2 entre $(A0 \oplus A1)$, et A2, pour obtenir en sortie $A2 \oplus (A0 \oplus A1)$. Et enfin

une porte XOR2 entre A3 et $A2 \oplus (A0 \oplus A1)$ pour obtenir en sortie la formule finale $A3 \oplus (A2 \oplus (A1 \oplus A0))$

Le module en C1 :

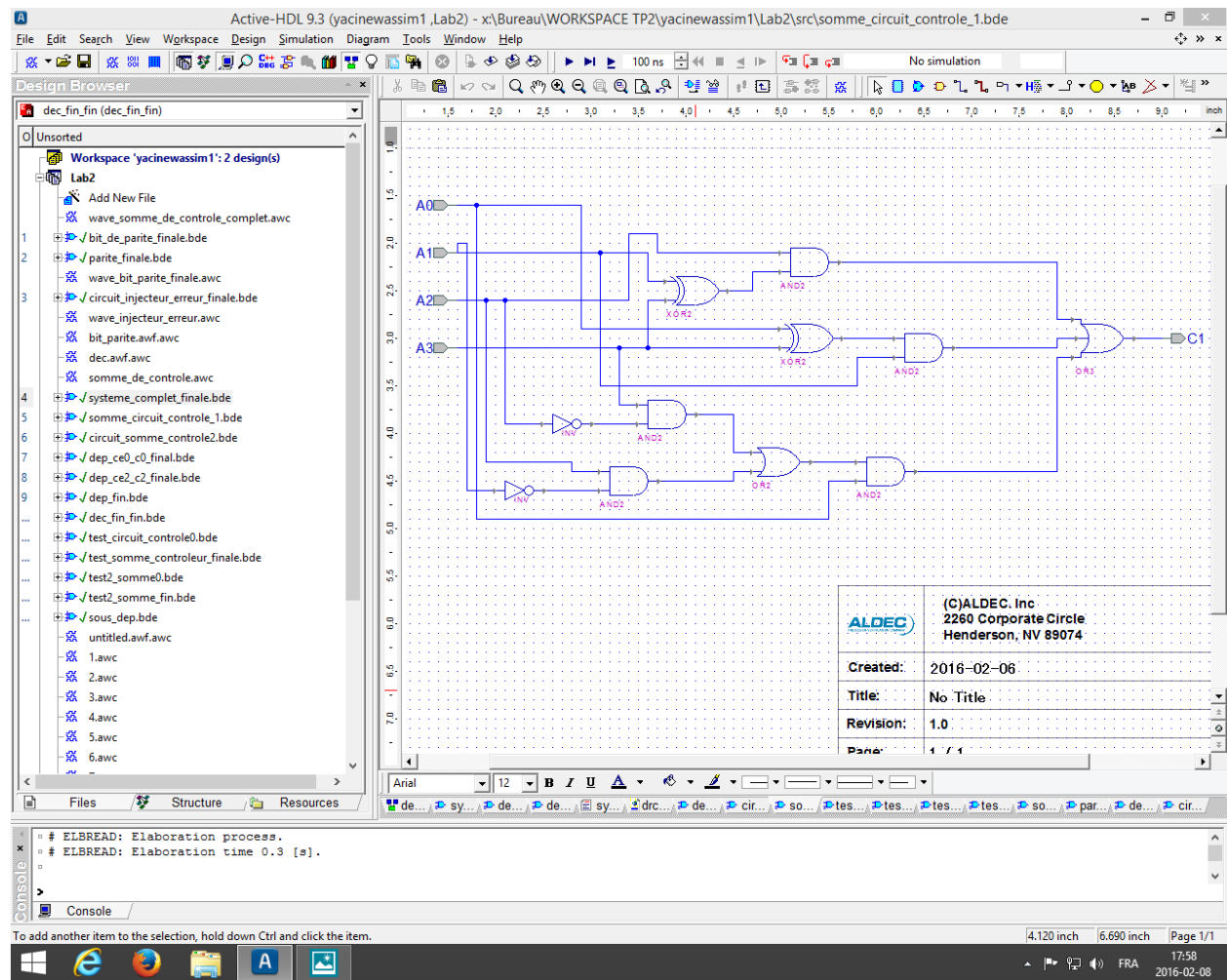


Figure4 : Schéma du circuit du module en sortie C1

Depuis le tableau de Karnaugh en sortie C1, on réalise la formule de cette sortie, en formant plusieurs figures de 2 cases, regroupant les cases contenant des 1. Pour obtenir ceci :

$$F = A_3A_2A_1' + A_2A_0A_1' + A_0A_3A_2' + A_1A_3A_0' + A_2A_1A_3' + A_1A_0A_3'$$

Par la suite, toujours en sortant le facteur commun, et en utilisant le OU EXCLUSIF [(NON A ET B) OU (A ET NON B)]

$$F = A_2(A_3 \oplus A_1) + A_1(A_3 \oplus A_0) + A_0(A_3A_2' + A_2A_1')$$

Pour réaliser ce circuit, on met donc une porte XOR2 dont les entrées seront A3 et A1 pour obtenir $A_3 \oplus A_1$ en sortie, et une porte ET entre A2 et $A_3 \oplus A_1$ pour obtenir $A_2(A_3 \oplus A_1)$

On met une porte XOR 2 entre A3 ET A0 pour obtenir $A_3 \oplus A_0$ en sortie, puis une porte ET2 entre A1 et $A_3 \oplus A_0$ pour obtenir en sortie $A_1(A_3 \oplus A_0)$

Puis, on met un inverseur sur le fil de l'entrée A2 pour obtenir A2'. Ensuite, une porte ET2 avec l'entrée A3 pour obtenir A_3A_2'

Un autre inverseur sur le fil de l'entrée A1 pour obtenir A1' et une porte ET2 avec l'entrée A2 pour obtenir A_2A_1' . Par la suite, une porte OU2 avec comme entrées A_2A_1' et A_3A_2' pour obtenir en sortie $A_2A_1' + A_3A_2'$. Et une porte ET2 entre A0 et $A_2A_1' + A_3A_2'$ pour obtenir en sortie $A_0(A_2A_1' + A_3A_2')$

Enfin, une porte OU3 avec comme entrées $A_2(A_3 \oplus A_1)$, $A_1(A_3 \oplus A_0)$, $A_0(A_3A_2' + A_2A_1')$

Pour obtenir en sortie $A_2(A_3 \oplus A_1) + A_1(A_3 \oplus A_0) + A_0(A_3A_2' + A_2A_1')$ qui est la formule simplifiée finale représentant ce module.

Module en sortie C2 :

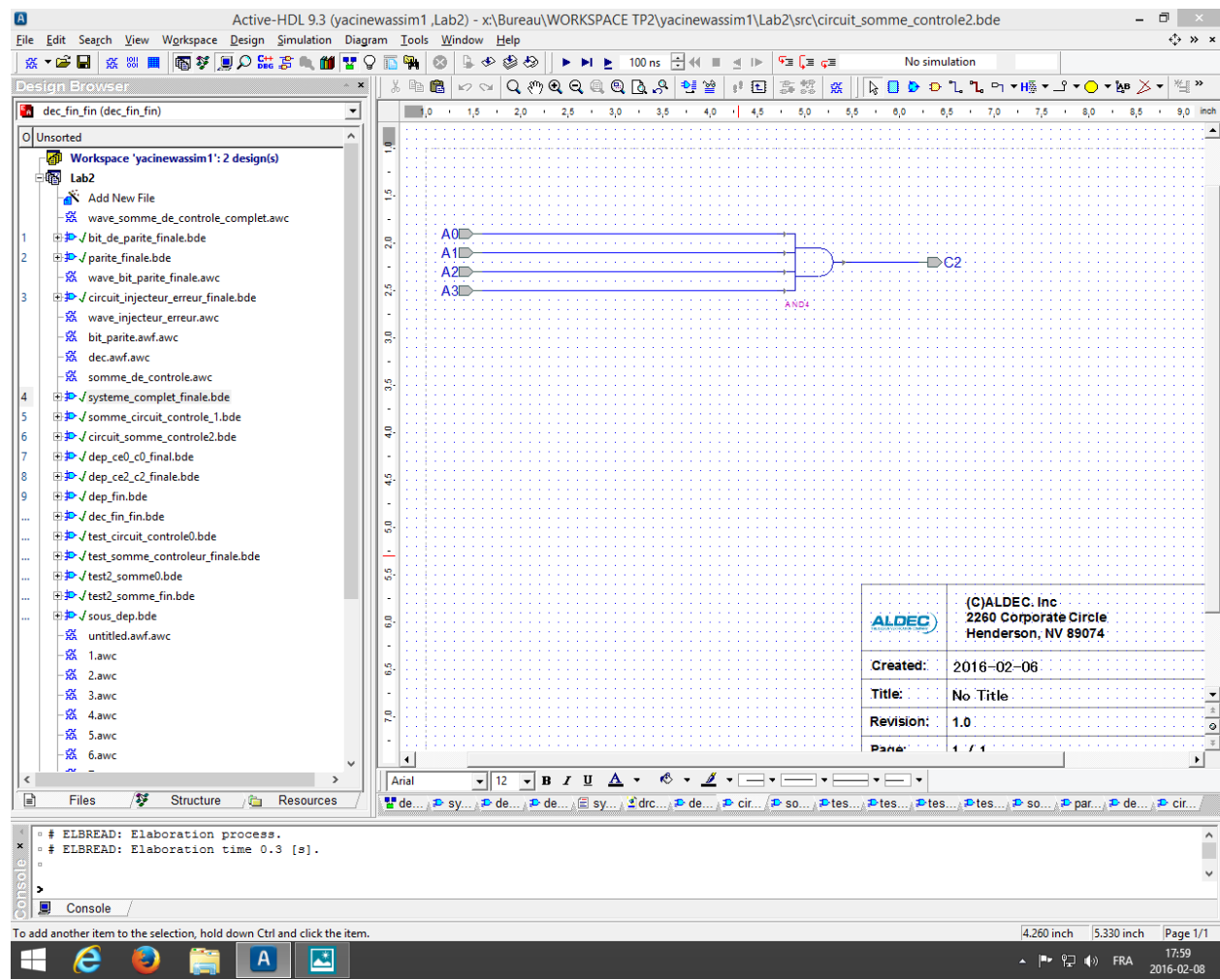


Figure 5 : Schéma du circuit du module en sortie C2

Ce circuit simple prend nos quatre entrées A0, A1, A2 et A4 pour les mettre dans une porte ET4 et obtenir en sortie A3A2A1A0 comme le montre notre formule en sortie C2: $F = A3A2A1A0$

L'injecteur d'erreur :

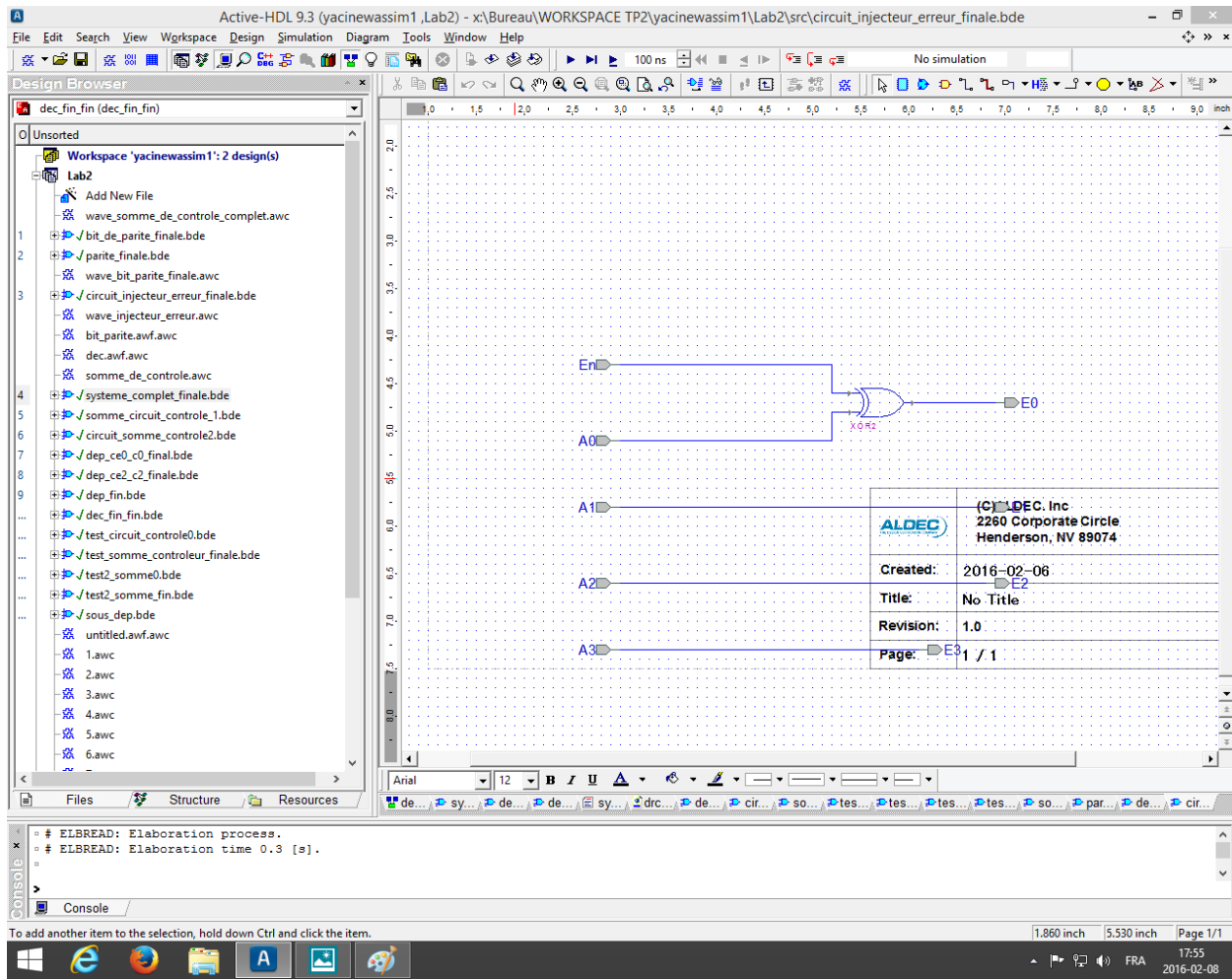


Figure 6 : Schéma du circuit de l'injecteur d'erreur

Le module injecteur d'erreur injecte une erreur dans la variable d'entrée. Il possède deux entrées ; A qui est sur 4 bits, et En qui lui est sur 1 bit. Lorsque En est a 1 le bit le moins significatif de l'entrée A à 4 bit est inversé, tandis que lorsque En est 0, le bit le moins significatif ne change pas.

Donc pour réaliser ce circuit, on sait déjà que les entrées A3 A2 A1 reste inchangées, et son directement liées respectivement aux sorties E3 E2 E1. Tandis que pour En et A0, nous somme parvenues à une formule depuis le tableau de Karnaugh qui est : $En'A0 + EnA0'$ et après simplification grâce au OU EXCLUSIF [(NON A ET B) OU (A ET NON B)] on obtient $En \oplus A0$.

Le module DEC :

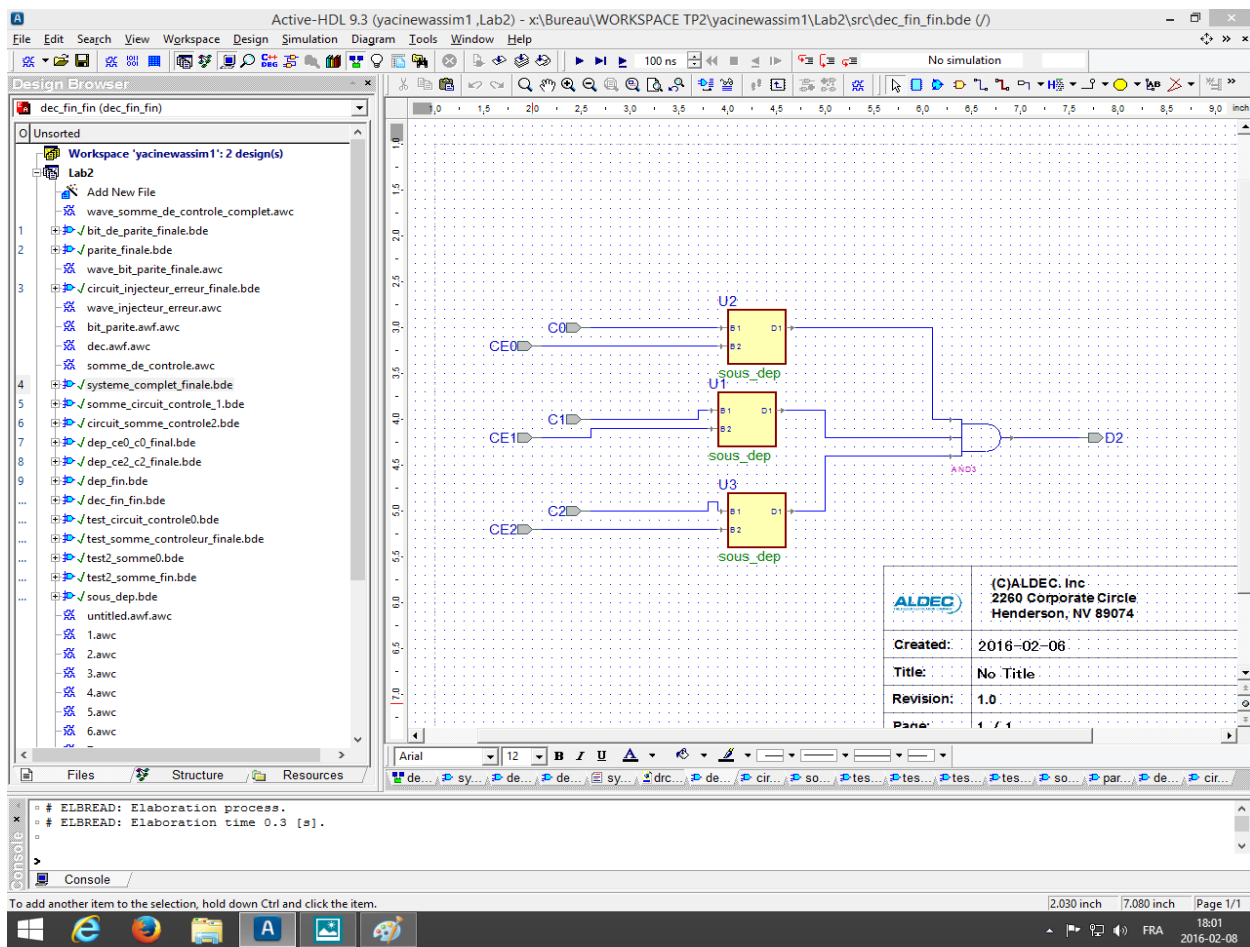


Figure 7 : Schéma du circuit DEC

Le module DEC est un comparateur 3 bits, comme vous pouvez le voir dans ce schéma, il est donc composé de trois DEP, ces DEP effectuent chacune une

comparaison entre CE2 et C2, CE1 et C1, CE0 et E0. On met donc une porte XNOR dans chaque DEP avec dans chacun des DEP les entrées à comparer pour obtenir au final 3 sorties : $(CE2 \oplus C2)'$, $(CE1 \oplus C1)'$, $(CE0 \oplus E0)'$

Les sorties de DEC sont à 1 seulement quand les sorties 3 comparaisons sont les mêmes donc, on met une porte ET3 avec comme entrées : $(CE2 \oplus C2)'$, $(CE1 \oplus C1)'$, $(CE0 \oplus E0)'$ pour avoir au final la formule : $F = (CE2 \oplus C2)'(CE1 \oplus C1)'(CE0 \oplus E0)'$ qui est celle du DEC.

Le module DEP :

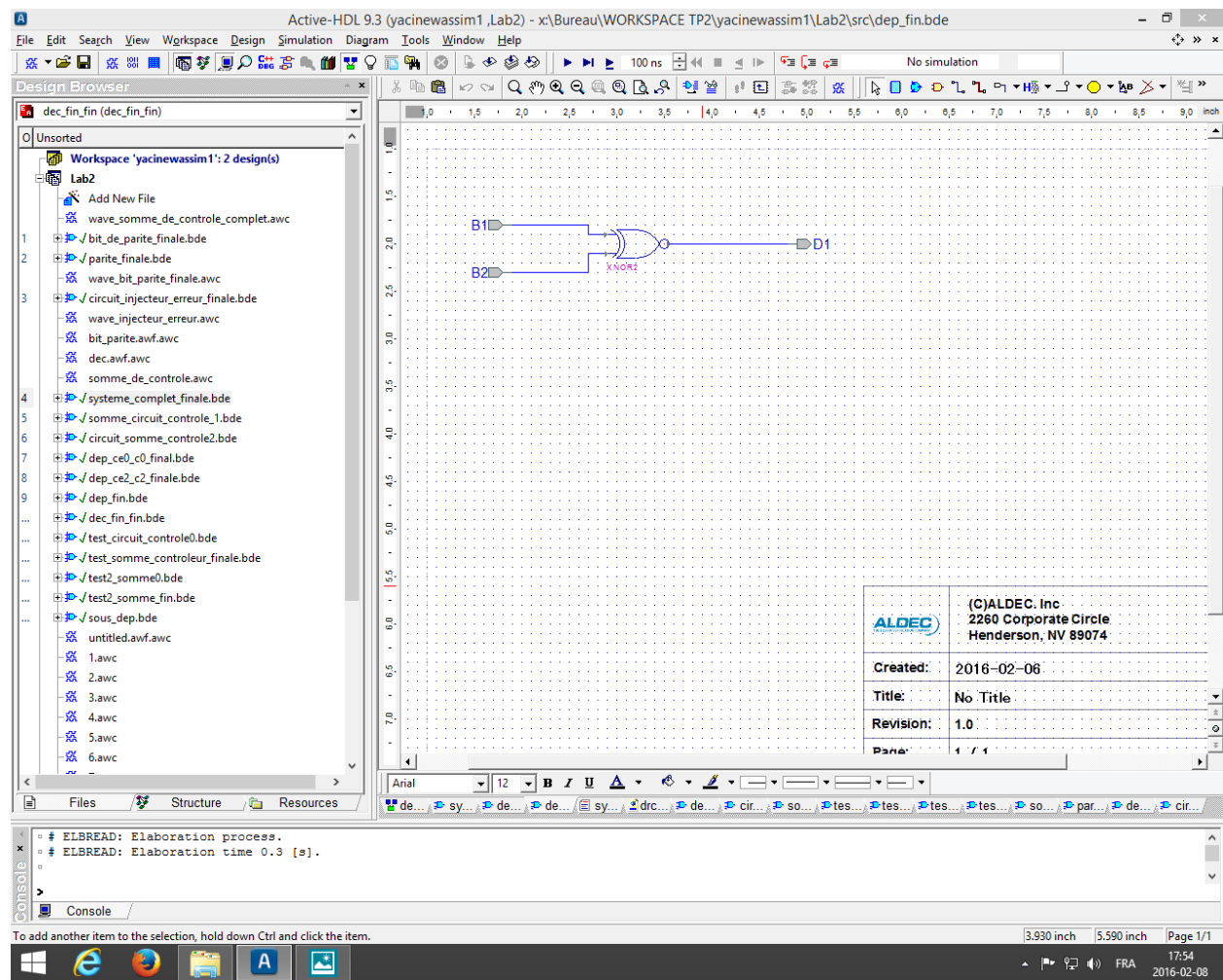


Figure 8 : Schéma du circuit du module DEP

Le DEP est un comparateur qui dépend des sorties du bit de parité 1 qui vient des entrées et du deuxième bit de parité qui lui dépend à son tour de l'injecteur d'erreur. Il indique vrais si les deux bits ont la même valeur et faux dans le cas contraire.

Son modèle est donc assez simple puisqu'il dépend que de deux entrées exclusivement. Il se résume à l'entrée des deux valeurs par une porte X NOR :

$$F = B1'B2' + B1B2 = (B1 \oplus B2)'$$

4. Stratégie de test des modules et images pour les simulations :

Stratégie de test :

La clé pour parvenir à tester si les circuits sont justes ou pas, est de lire la simulation, si celle-ci est correcte, le circuit est juste, sinon, le circuit contient forcément une erreur.

Stratégie de test pour la somme de contrôle :

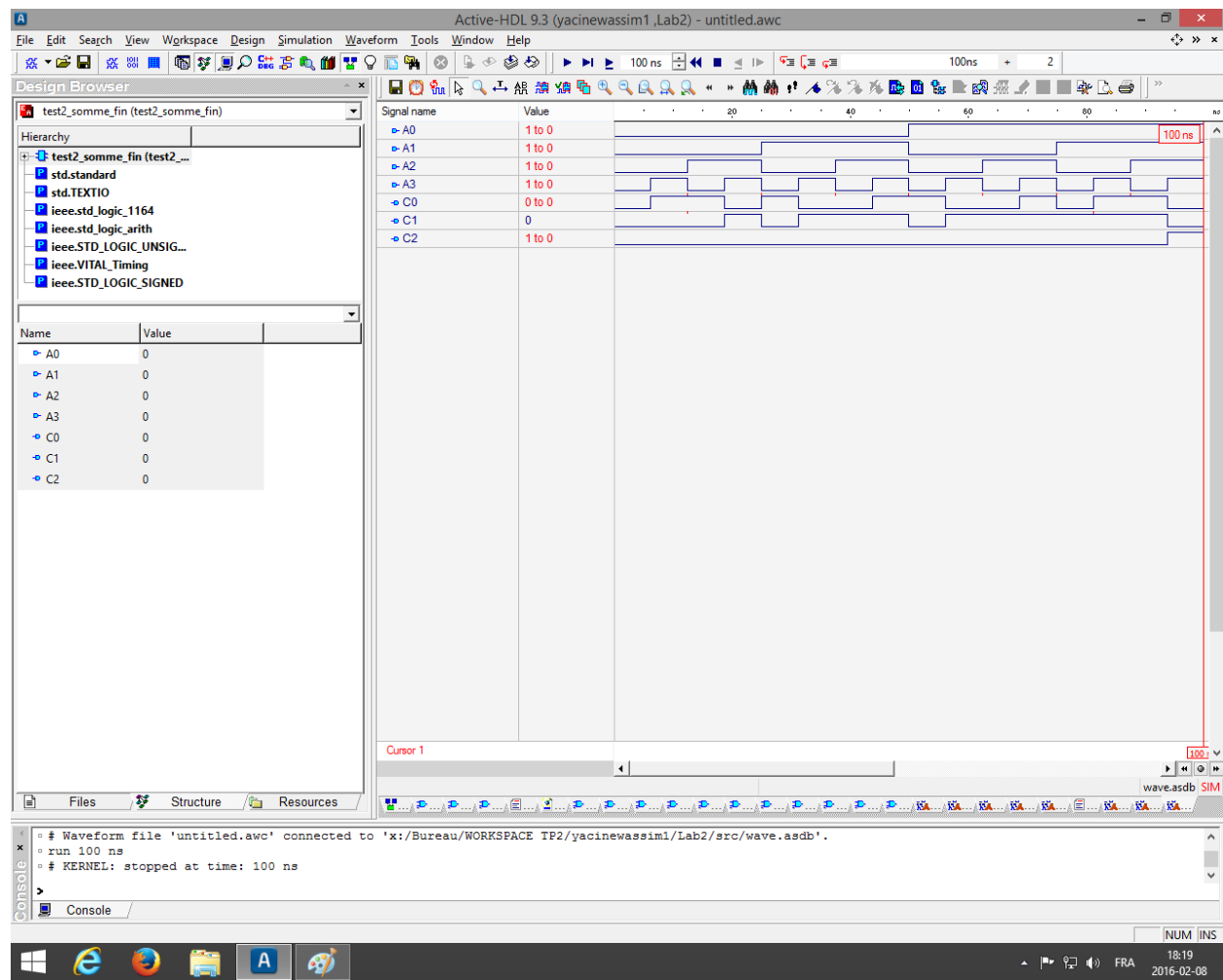


Figure 9 : Schéma du la simulation du circuit du module de somme de contrôle

Déjà, nous savons que la sortie C2 n'est censée être à 1 qu'une seule fois, lorsque les quatre entrées sont à 1, que le résultat en sortie est donc 4, en binaire 100. C'est effectivement le cas dans notre simulation.

Puis, pour tester si les résultats des 3 autres sorties étaient justes, nous avons effectué le modulo de l'addition des valeurs en entrée par 2, si le résultat est à 0, c'est-à-dire que la valeur de la somme des entrées est égale à 2; seule la sortie C1 est impérativement à 1 pour qu'en binaire nous ayons 010, si le résultat est 1, c'est-à-dire que la valeur de la somme des entrées est égale à 3, cela signifie que seules les sorties C1 et C0 doivent toutes deux être impérativement à 1 pour qu'en binaire nous ayons 011.

Dans le cas où la valeur de la somme est égale à 1 seule la sortie C0 est égale à 1 pour avoir en binaire 001. Et enfin pour le dernier cas où la valeur de la somme des valeurs des entrées est égale à 0. Toutes les sorties sont à 0 pour obtenir en binaire 000.

C'est effectivement le cas dans notre simulation, le circuit accompli est donc correctement réalisé.

Stratégie de test pour l'injecteur d'erreur :

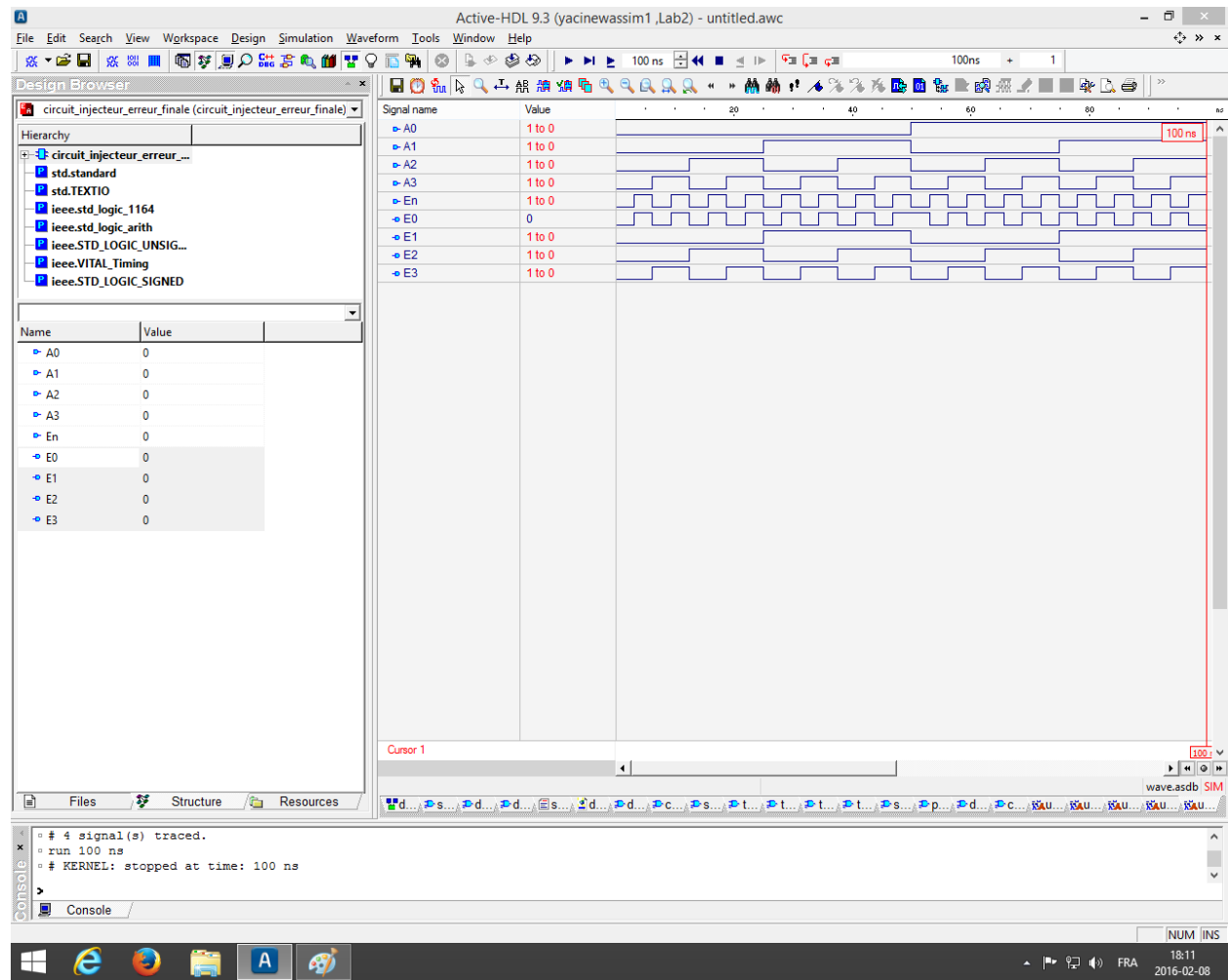


Figure10 : Schéma de simulation du circuit du module injecteur d'erreur

Comme nous pouvons le voir dans la simulation, ce circuit remplit bien le rôle qu'on attend de lui.

C'est-à-dire que l'on remarque dans la simulation que :

A1=E1

A2=E2

A3=E3

En ce qui concerne cette partie, c'est ce qu'il nous fallait comme résultat.

Passons maintenant à A0 et E0 en fonction de En. La principale fonction de l'injecteur d'erreur et d'inverser l'entrée A0 lorsque En est à 1, et de la laisser telle quelle lorsque En est à 0

Dans la simulation, nous sommes donc supposés remarquer que E0 et A0 ont les mêmes valeurs lorsque En est à 0 et qu'E0 est l'inverse d'A0 lorsque En est à 1. Ce qui est effectivement le cas, donc notre circuit est juste.

Stratégie de test pour le circuit DEP :

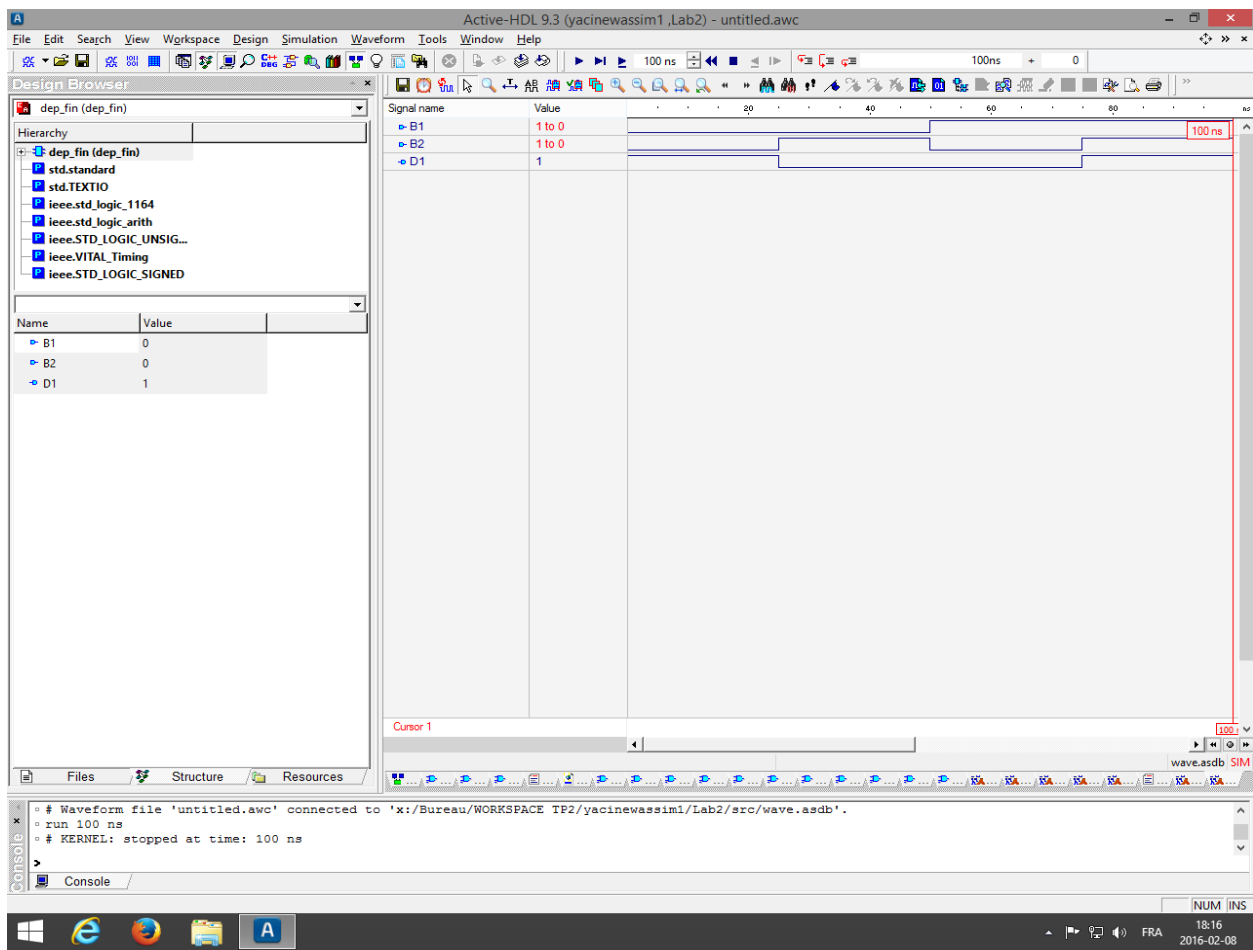


Figure 11 : Schéma de simulation du circuit du module DEP

Depuis la simulation, on remarque que les résultats affichés correspondent aux résultats voulus.

Ce circuit compare l'entrée B1 et B2, et n'affiche une sortie à 1 que lorsqu'elles ont une valeur identique.

C'est-à-dire que lorsque B1 et B2 diffèrent, la sortie est à 0, et lorsque B1 et B2 ont la même valeur, la sortie est à 1. Ce qui est effectivement le cas dans notre simulation.

Stratégie de test pour le DEC :

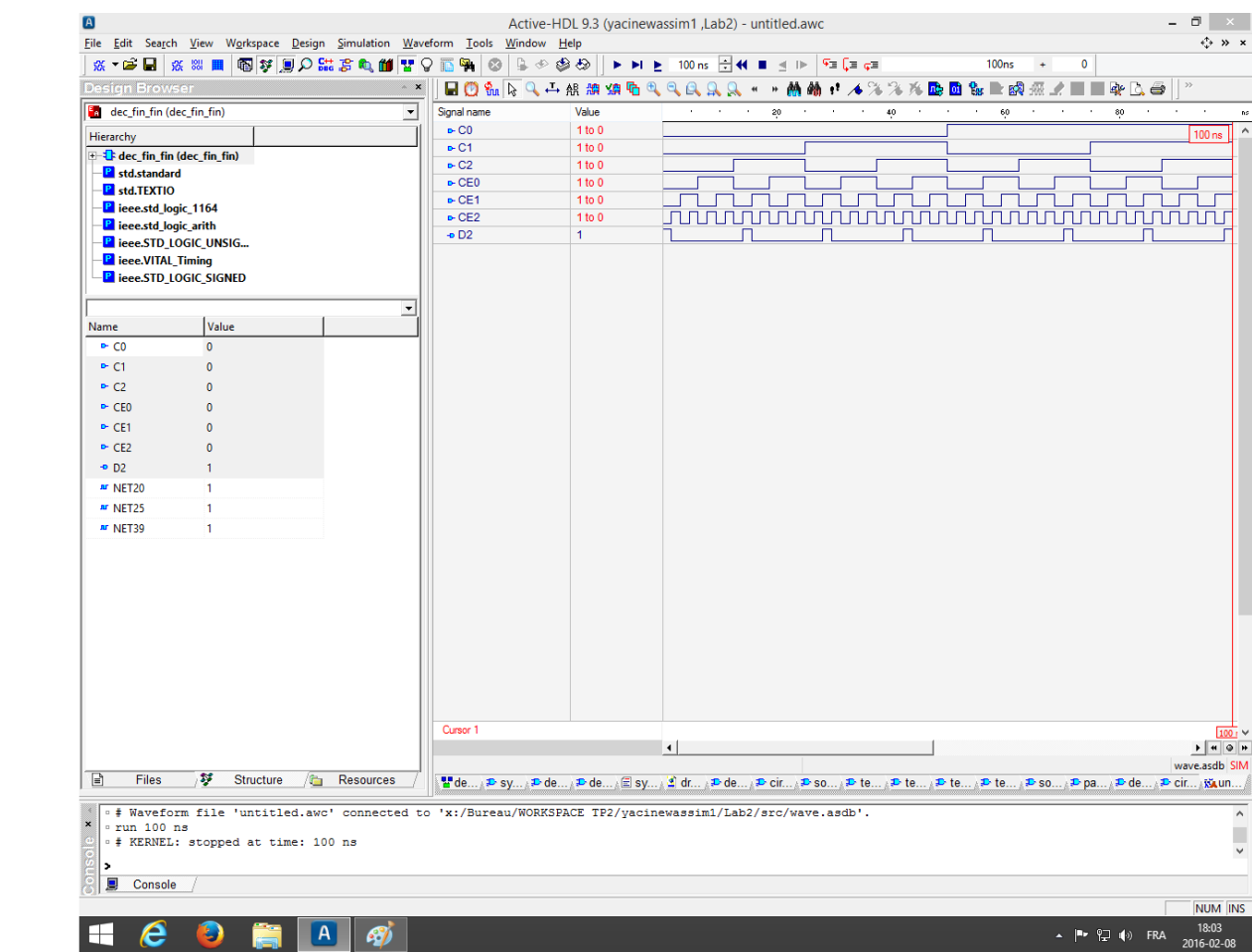


Figure 12 : Schéma de simulation du circuit du module DEC

Depuis la simulation, on remarque que l'on a obtenu le résultat escompté de ce circuit.

En effet, ce circuit effectue 3 comparaisons, entre CE0 et E0, entre CE1 et E1 et entre CE2 et C2 à l'aide DE 3 DEP. S'ils ont la même valeur la sortie de chaque DEP est à 1, puis, la sortie de DEC ne sera à 1 que si les 3 DEP sont à 1. Donc depuis la simulation remarquer que la sortie du DEC est à 1 si et seulement si :

CE0 et E0 ont la même valeur

ET

CE1 et E1 ont la même valeur

ET

CE2 et E2 ont la même valeur

C'est effectivement le cas dans notre simulation, donc le test est fructueux.

Stratégie de test pour le Bit de parité :

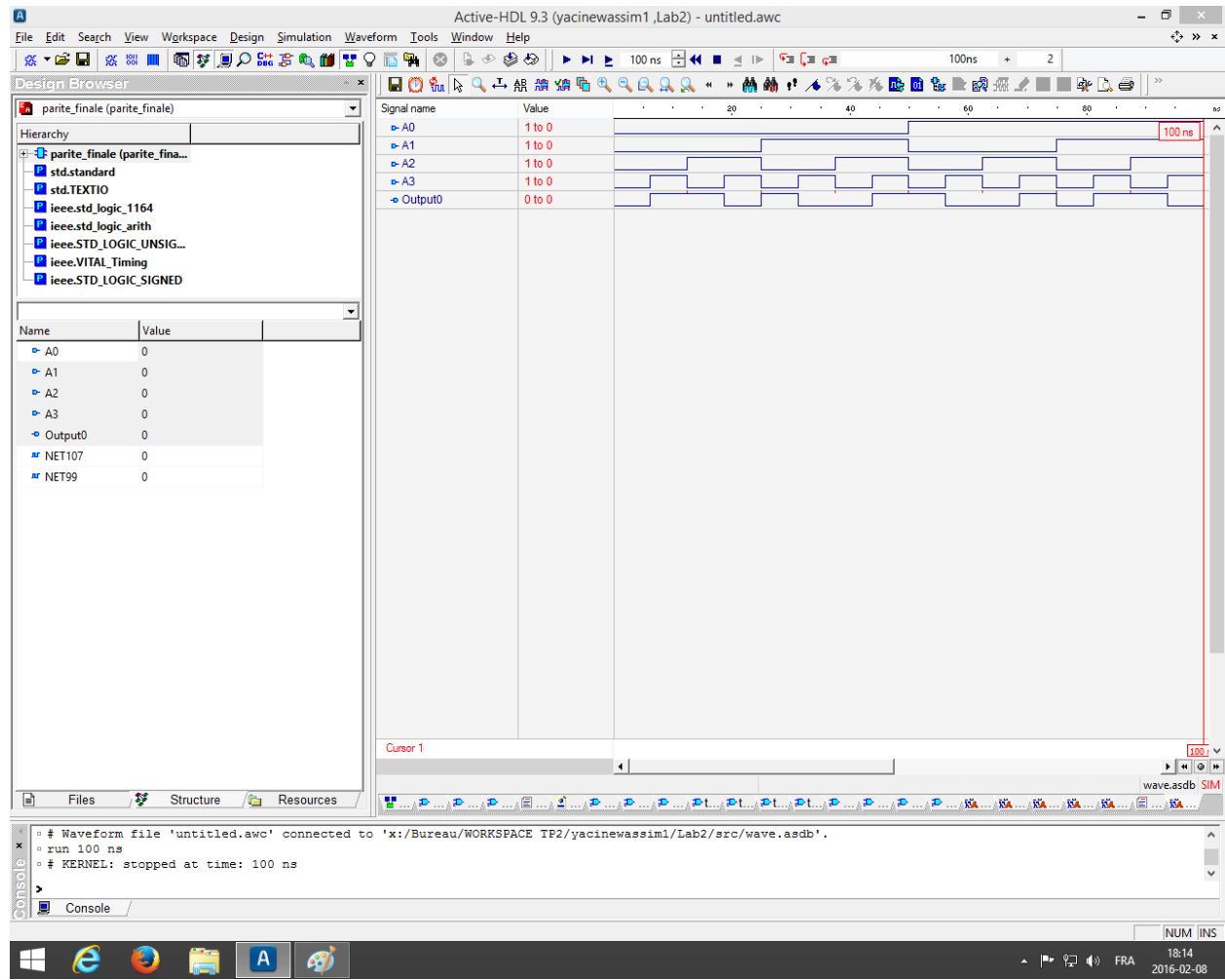


Figure 13 : Schéma de simulation du circuit du module Bit de parité

Ce circuit est censé avoir une sortie à 1 lorsque le nombre de 1 en entrées est impaire et une sortie à 0 lorsque le nombre de 1 en entrée est paire ou égale à 0.

Depuis la simulation, on remarque que lorsque la sortie est à 1 seulement lorsque, 1 ou 3 entrées sont à 1. On remarque aussi que la sortie est à 0 seulement lorsque 0, 2 ou 4 entrées sont à 1.

C'est effectivement le rôle de ce circuit, les résultats de la simulation sont ceux qu'il nous fallait obtenir, on peut en déduire que le circuit est correct.

Stratégie de test pour le circuit du système complet :

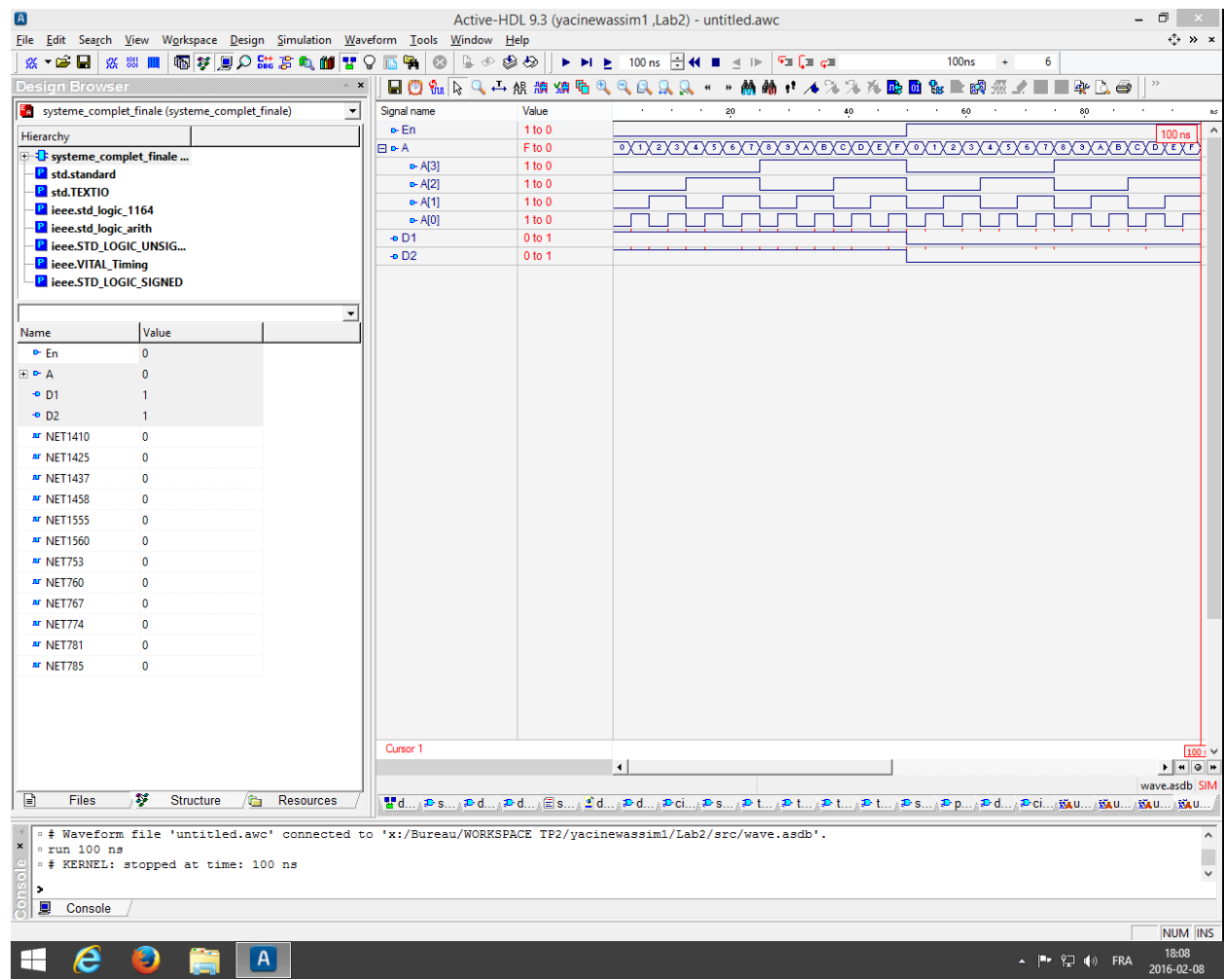


Figure 14 : Schéma de simulation du circuit du système complet

On s'est rendu compte, d'après le schéma du circuit total que quelles que soit les entrées, les sorties D1 et D2 du circuit final étaient toujours les mêmes; toujours à 1 lorsque En=0 et toujours à 0 lorsque En=1.

En effet, le DEP et le DEC, avec comme sortie D1 et D2, comparent les entrées. Lorsqu'elles sont identiques, c'est-à-dire qu'il n'y a pas d'erreur injectée, la sortie est à 1 (En=0), tandis que si l'erreur est injectée, les entrées diffèrent, donc, la comparaison échoue et la sortie est à 0 (En=1).

Comme l'une des entrées du DEC et l'une des entrées du DEP sont en relation directe avec l'injecteur d'erreur, si l'erreur est injectée, la

comparaison de leurs 2 entrées respective échouera car elles auront des valeurs différentes , par conséquent les 2 sorties D1 et D2 seront toutes les deux à 0 lorsque l'erreur est injectée ($E_n=1$).

La simulation de la figure 14 justifie notre logique.

5. Conclusion

Pour conclure, dans ce travail pratique, nous avons dû suivre une logique particulière pour réaliser le tableau de vérité de chaque module, par la suite, nous avons pu monter les tableaux de karnaugh propres à chacune des sorties. Ces tableaux ont servi à mettre aux points les formules que l'on pourra simplifier à l'aide de la factorisation et du \oplus (ou exclusif). Ces formules, ont été essentielles lors de la mise au point des circuits. Par la suite, pour tester si nos circuits étaient corrects, il a fallu faire la simulation de chaque circuit, et de voir si les résultats obtenus coïncidaient avec le rôle du module en question. Une fois, chaque circuit vérifié, dont le circuit complet, nous avons mis au point un fichier texte .ucf pour le mapping de chaque entrée et sortie du circuit complet; là , on s'est rendu compte que pour cela fonctionne , il fallait que la ligne pour l'entrée E_n soit en dernier , car effectivement , le circuit ne prend en charge cette entrée qu'après les 4 entrées de A, dans le cas contraire , le programme ne fonctionnait pas sur la carte. Puis, nous avons effectué la synthèse, suivie de l'implémentation. Lorsqu'aucune erreur n'a été signalée, nous avons pu affecter notre programme à la carte FPGA, et ainsi le tester. Le résultat de ce travail était présent, la carte fonctionnait correctement, nous sommes parvenus à implémenter des modules en logique combinatoire.