École Polytechnique de Montréal

Département de génie informatique et génie logiciel

INF1500

Logique des systèmes numériques

Laboratoire 4 : Rapport sur les circuits logiques séquentiels en VHDL

Soumis par

Tamer Arar

Éric Chao

Section de labo #2

28 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

Introduction

Ce laboratoire nous permet de nous familiariser avec les circuits logiques séquentiels et le code VHDL en nous permettant de concevoir un circuit logique séquentiel permettant d'afficher sur l'écran LCD du FPGA une séquence de nombres impairs, selon la parité choisie par l'utilisateur à l'aide d'un bouton poussoir jusqu'à un nombre sur 4 bits que ce dernier aura choisi à l'aide de quatre interrupteurs chacun correspondant à un bit du nombre.

Un ensemble de modules en logique séquentiel ont été conçus dans ce laboratoire :

Trois modules à concevoir :

- Le module diviseur de fréquence qui convertit la fréquence de l'horloge de la carte FPGA en un Hz.
- Le module de parité qui sélectionne les nombres générés par le compteur, selon la parité choisie par l'utilisateur.
- Le module top qui regroupe les deux autres modules et qui connecte le module de parité à l'afficheur LCD

Des simulations ont été réalisées afin de confirmer le fonctionnement des modules.

Module diviseur de fréquence

Tout d'abord, nous voulons créer un circuit chargé de réduire la fréquence de la carte FPGA, qui est de 100MHz, à 1Hz. Ainsi, après avoir programmé la carte, un utilisateur pourra alors voir s'afficher les nombres à un intervalle de 1 seconde entre chacun de ces nombres. Puisque la division par 100 000 000 en binaire (01011111010111100001000000000) se fait sur 32 bits et que nous n'avons pas à notre disposition un compteur 32 bits, il a fallu utiliser deux compteurs 16 bits. La création d'un comparateur sur 16 bits a alors été nécessaire :

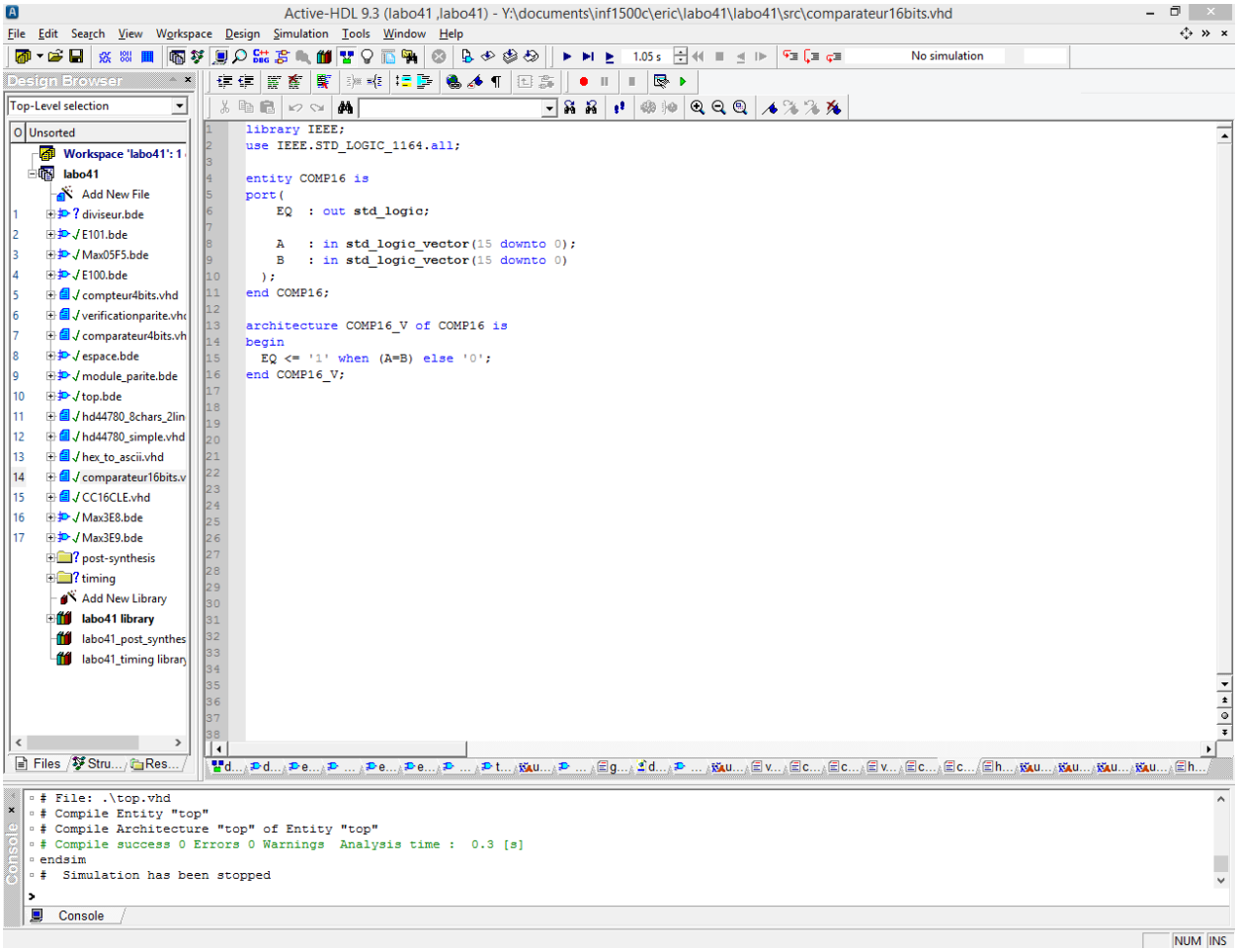

Figure 1 : Comparateur à deux entrées sur 16 bits.

Ainsi, il a été réalisé à l'aide d'un code du langage VHDL. Il compare l'entrée A sortant du premier compteur 16 bits avec l'entrée B, une valeur fixée à $(E100)_{16}$ ou $(1110000100000000)_2$. Alors, quand l'entrée A est égale à l'entrée B, la sortie EQ sera à 1, sinon à 0. Voici le circuit de l'entrée B fixée à $(1110000100000000)_2$:

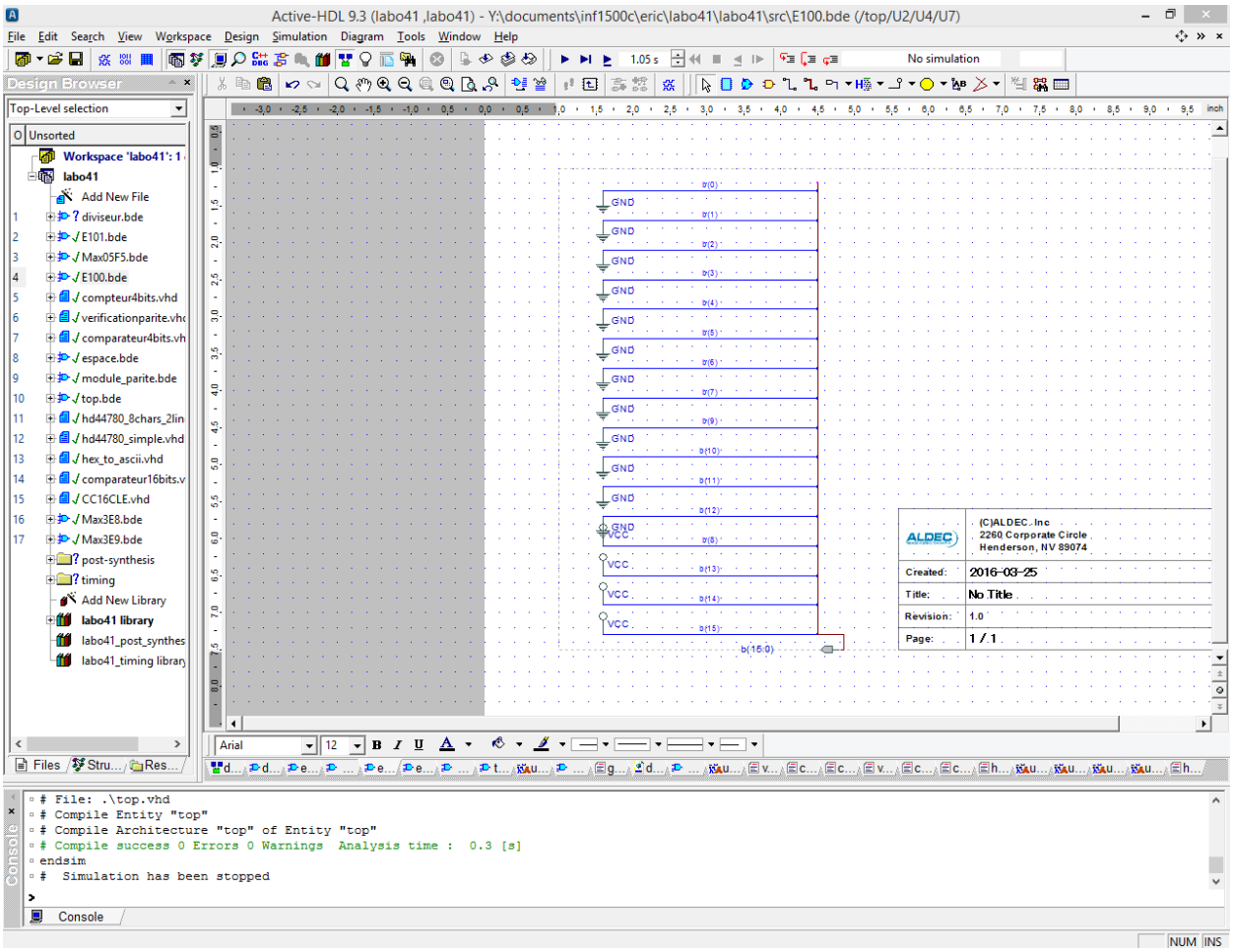

Figure 2 : Circuit de la valeur $(E100)_{16}$.

De même, un autre comparateur 16 bits est également fixé sur un deuxième compteur 16 bits. Le comparateur compare l'entrée A sortant du deuxième compteur avec l'entrée B fixée à $(05F5)_{16}$ ou $(0000010111110101)_2$. Ainsi, quand l'entrée A sera égale à l'entrée B, la sortie sera égale à 1, sinon à 0. Voici le circuit de la valeur $(0000010111110101)_2$ à l'entrée B :

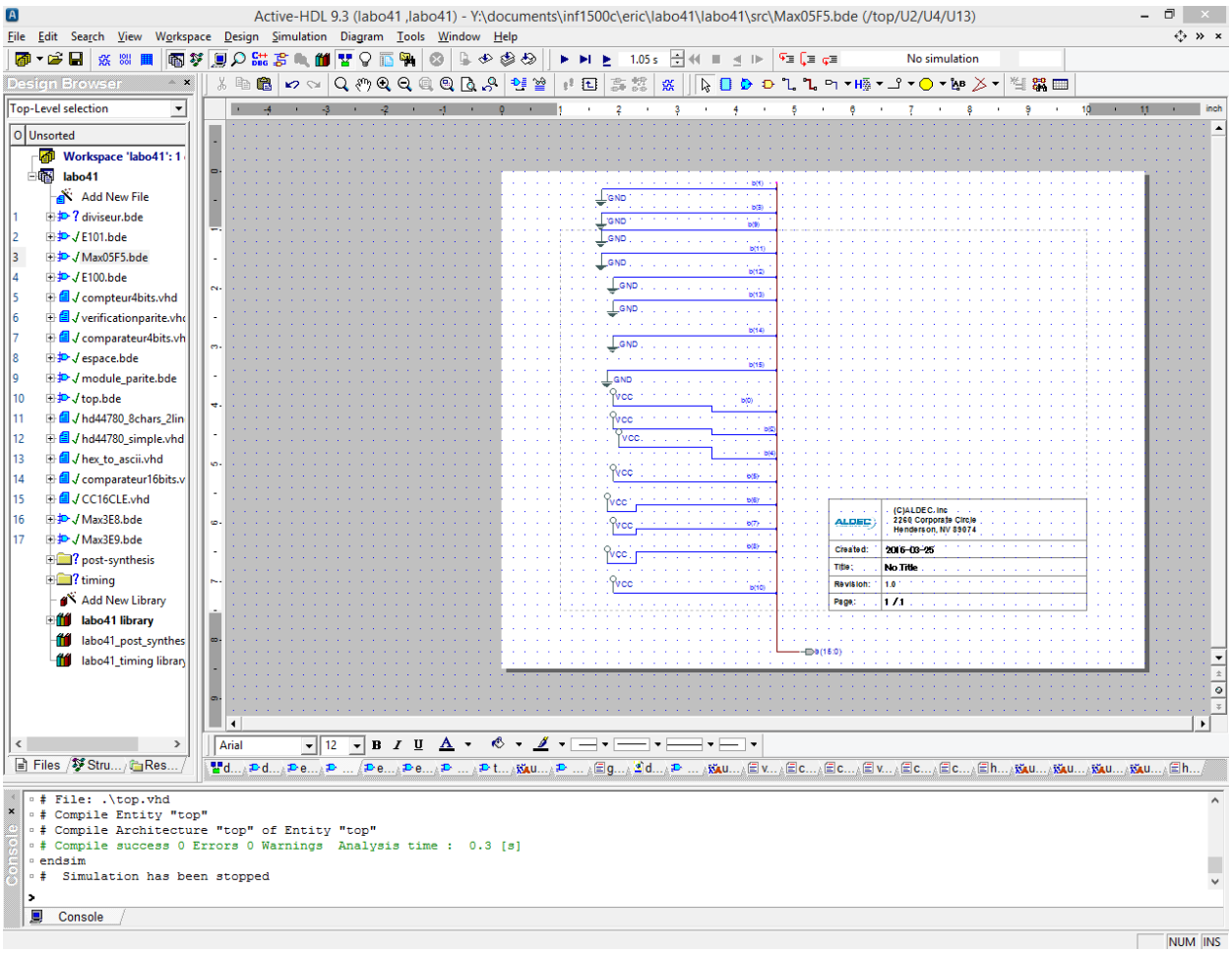

Figure 3 : Circuit de la valeur $(05F5)_{16}$.

Ensuite, les deux sorties des deux comparateurs 16 bits sont reliées à une porte logique ET. De ce fait, si et seulement si les 2 sorties EQ sont à 1, la sortie finale sera à 1. Voici le circuit diviseur de fréquence créé :

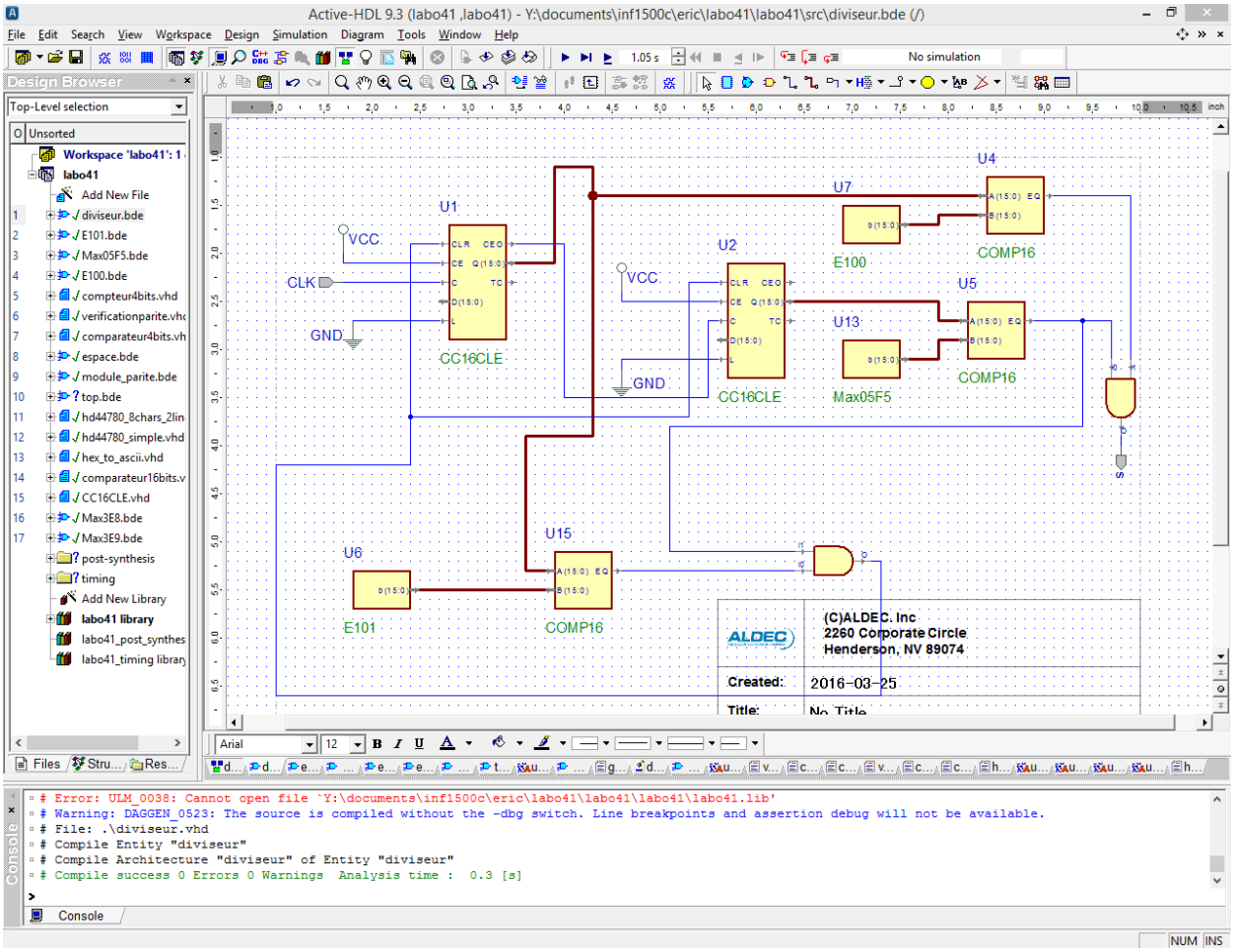

Figure 4 : Module diviseur de fréquence.

On peut observer que l'entrée CE des deux compteurs doit être fixée à 1 pour mettre en marche ces derniers. Ensuite, la sortie CEO se met uniquement à 1 quand le compteur a atteint sa valeur maximale soit $(10000)_{16}$ ou $(1000000000000000)_2$. Lorsqu'elle est à 1, la sortie CEO du compteur U1 entre par l'entrée C du deuxième compteur U2. Ainsi, lorsqu'U1 a atteint sa valeur maximale CEO envoie un signal vers l'entrée C de l'horloge d'U2, synchronisant ainsi les deux compteurs. Quand U2 aura atteint la valeur voulue : $(05F5)_{16}$, il renvoie la valeur 1 ce qui a pour effet de remplir une des deux conditions pour que la sortie « s » renvoie la valeur 1 et pour la porte logique ET reliée au comparateur U15 qui met en actif haut les entrées CLR de U1 et de U2 pour que ces derniers recommencent à compter à partir de 0 en même temps. Par ailleurs, les entrées L des 2 compteurs sont misés à 0, car elles ne sont pas utilisées. En effet, il existe un autre moyen de mettre en cascade les deux compteurs par une autre méthode que celle des comparateurs. L'entrée L aurait été alors misée à 1, car son utilité est de faire en sorte que la sortie du compteur Q est décidée par une mise en valeur fixée dans l'entrée D.

On peut également observer sur le circuit un troisième comparateur U15 sur 16 bits avec la valeur $(E101)_{16}$ ou $(1110000100000001)_2$. En effet, il permettra de vérifier la fonctionnalité désirée de ce circuit. En simulant le circuit, on devrait constater la valeur de 1 à la sortie s après une durée d'une seconde ce qui correspond à la période de notre nouvelle horloge. Toutefois, cela est presque impossible sans ce troisième comparateur, car du fait de contraintes logistiques, si nous réinitialisons les compteurs U1 et U2 lorsqu'ils atteignent 5F5E100, le circuit n'aura pas le temps de renvoyer la valeur 1. Alors, la solution est d'utiliser ce troisième compteur U15 afin de prolonger le temps de réinitialisation à 5F5E101 afin de laisser un délai au circuit pour qu'il renvoie la valeur 1 à temps. Voici la simulation avec ce troisième comparateur U15 :

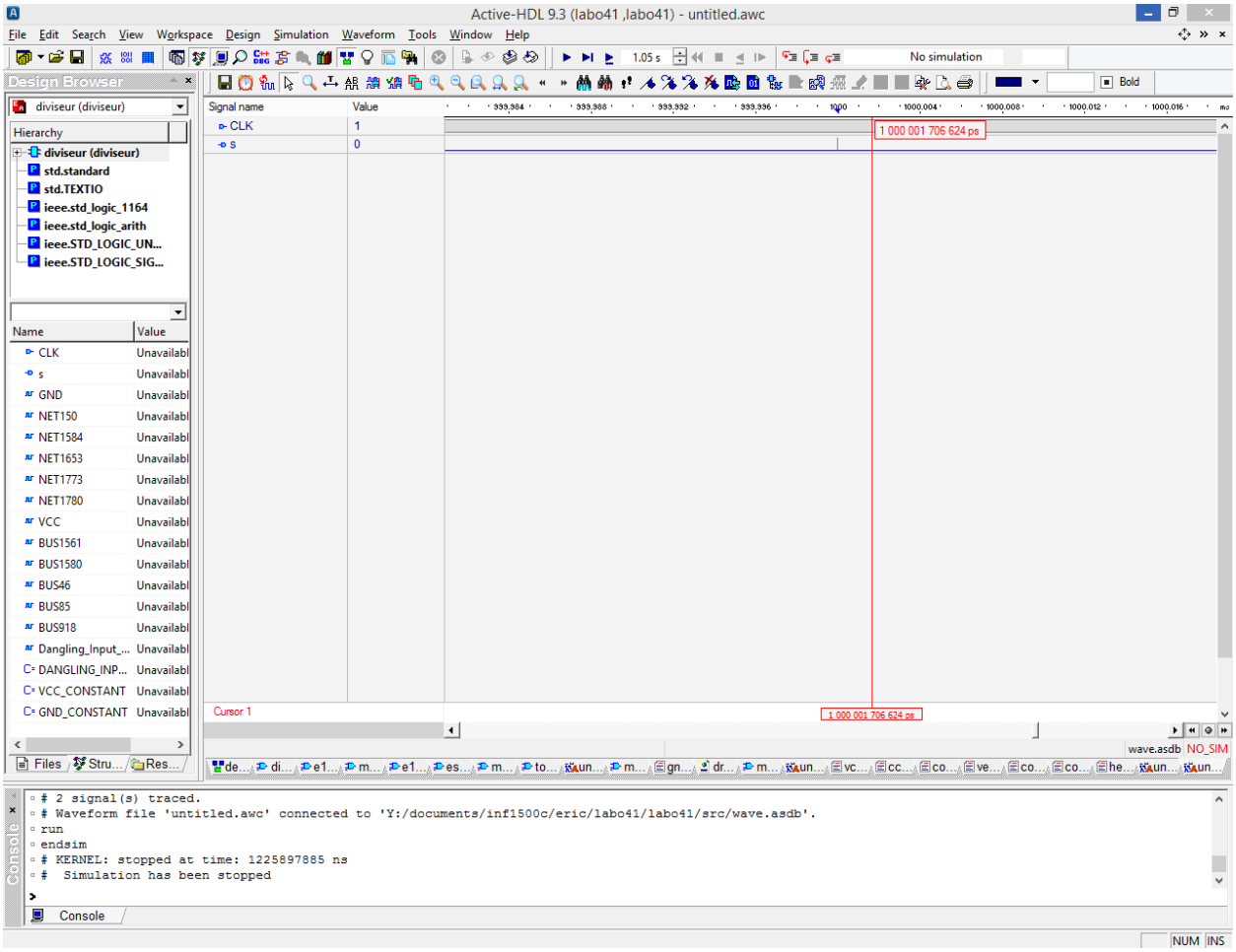

Figure 5 : Simulation du module de division de fréquence.

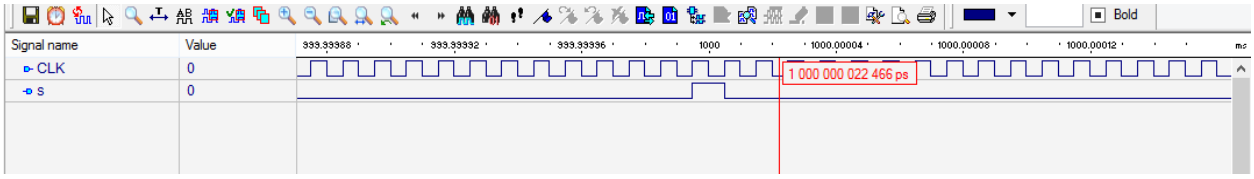

Figure 6 : Simulation du module de division de fréquence agrandi.

Ainsi, une seconde après le début de la simulation à une fréquence CLK de 100MHz, un pic est aperçu. Le circuit permettant la division de la fréquence par 100 millions est alors vérifié.

Module de parité

Par la suite, ce circuit est réalisé pour afficher une suite de nombres pairs ou impairs, au choix de l'utilisateur, sans dépasser un nombre sur 4 bits choisis par l'utilisateur. Pour commencer, il faut créer une porte logique qui permettra de choisir entre des nombres pairs ou impairs à l'aide d'un sélectionneur :

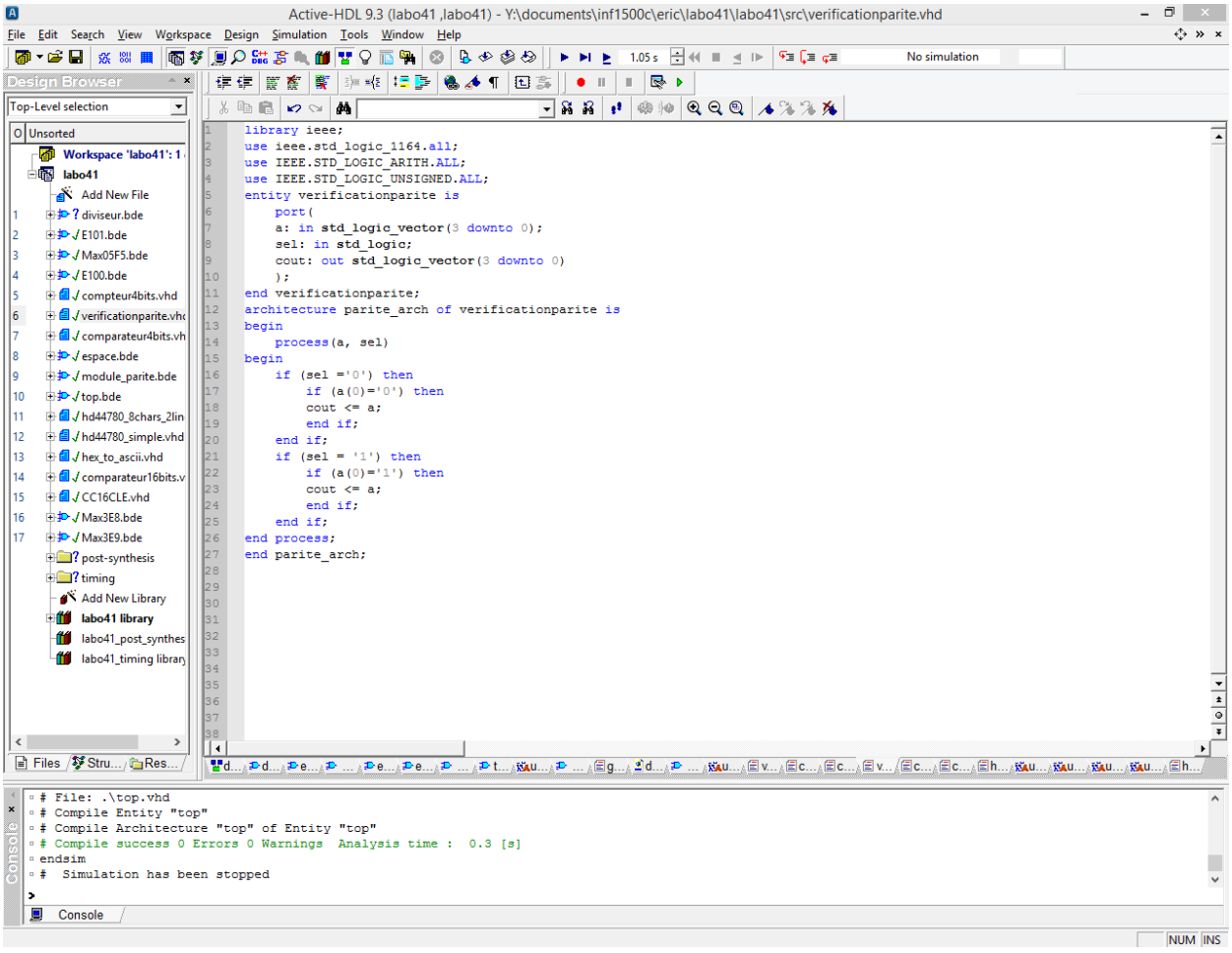

Figure 7 : Vérificateur de parité sur 4 bits.

Comme on peut le voir, le bloc a été créé par le langage VHDL. Ainsi, si l'entrée sel est à 0, on veut afficher les nombres pairs, alors que si l'entrée sel est à 1, on veut afficher les nombres impairs.

Ensuite, on a créé un compteur et un comparateur sur 4 bits :

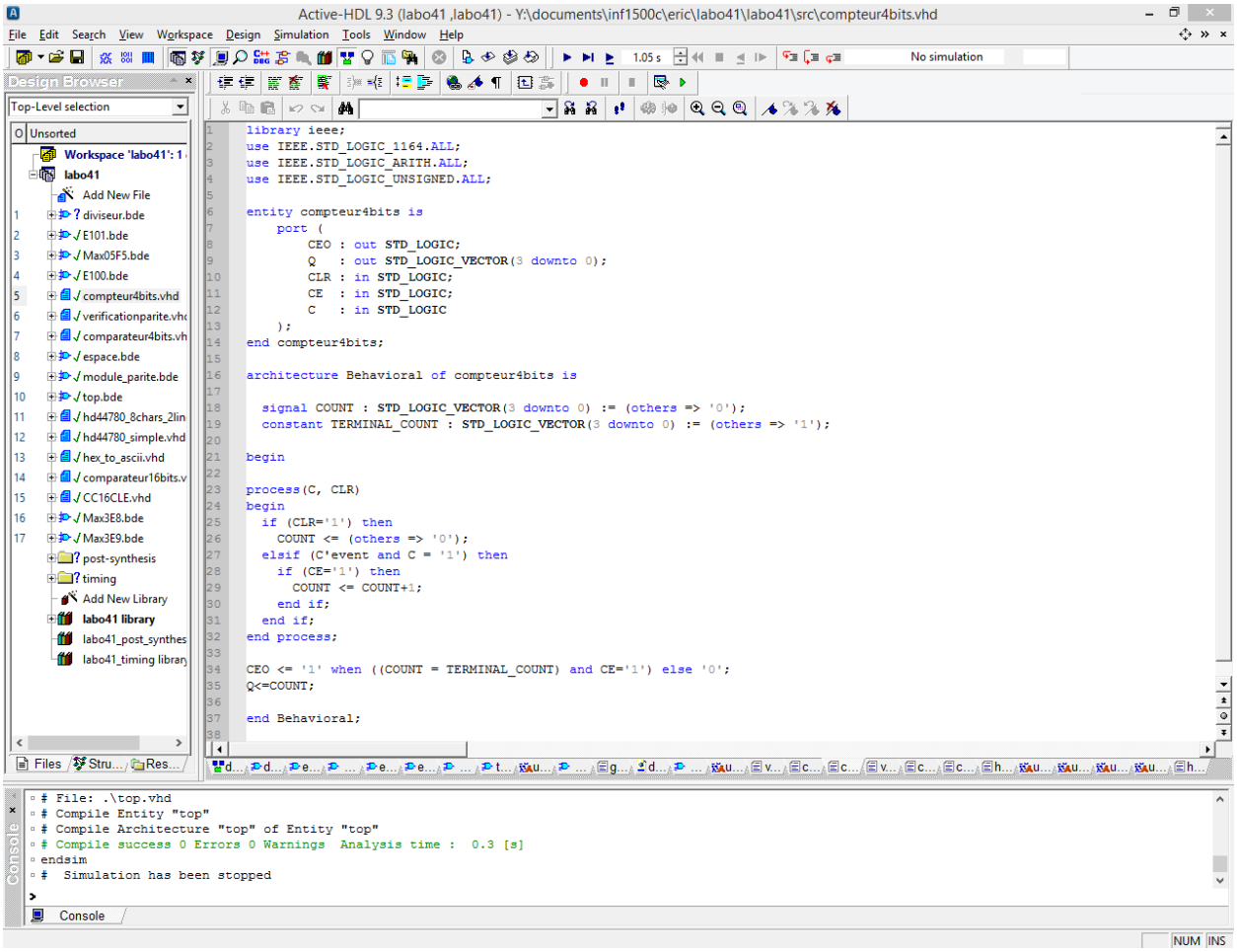

Figure 8 : Compteur sur 4 bits.

Ce compteur s'arrête de compter à sa valeur maximale i.e. $(10000)_2$.

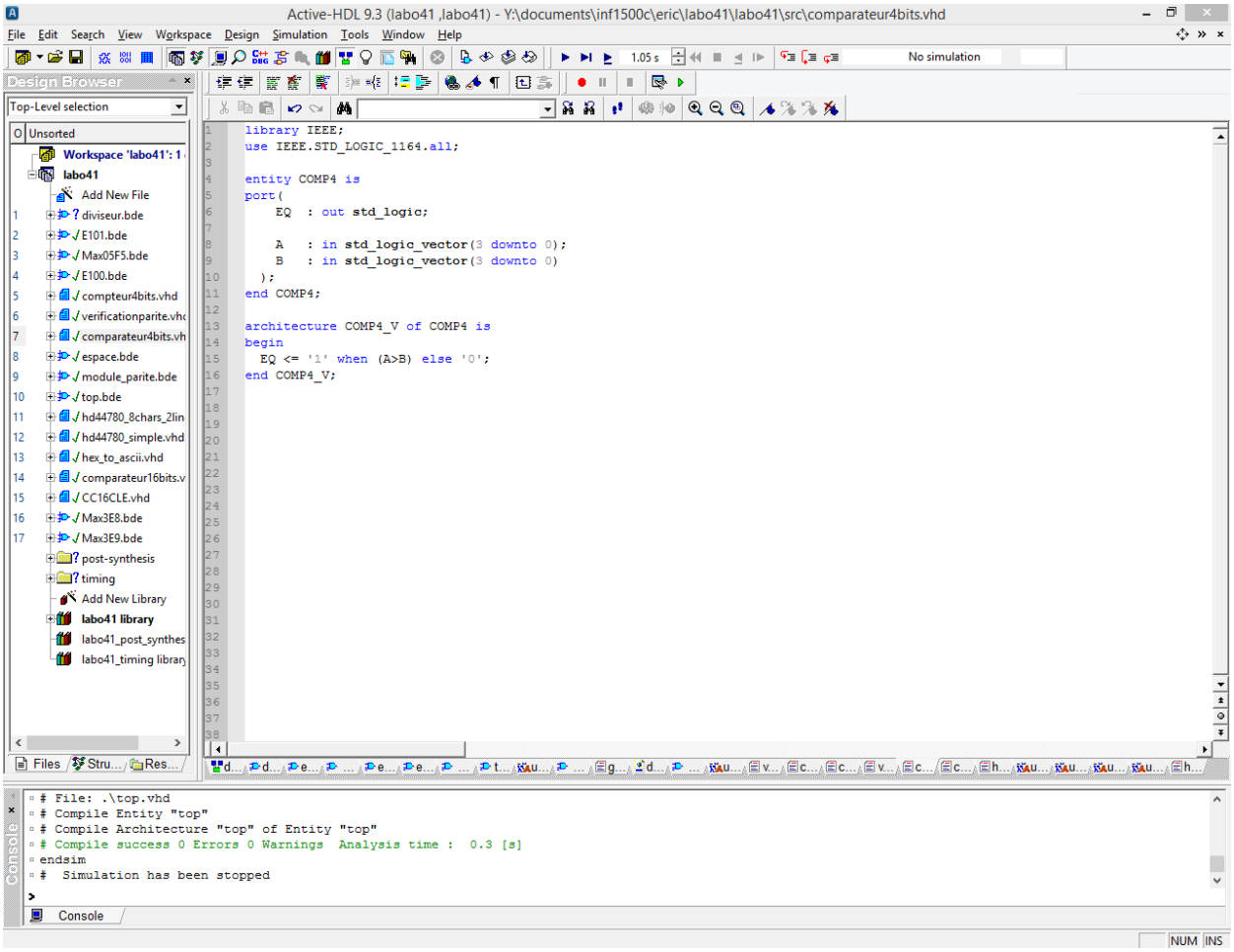

Figure 9 : Comparateur à deux entrées sur 4 bits.

Ce comparateur possède une entrée A et une entrée B. Lorsque l'entrée A devient plus grande que l'entrée B, la sortie EQ est égale à 1 (actif haut).

Ainsi, on combine ces blocs pour obtenir le module de parité final :

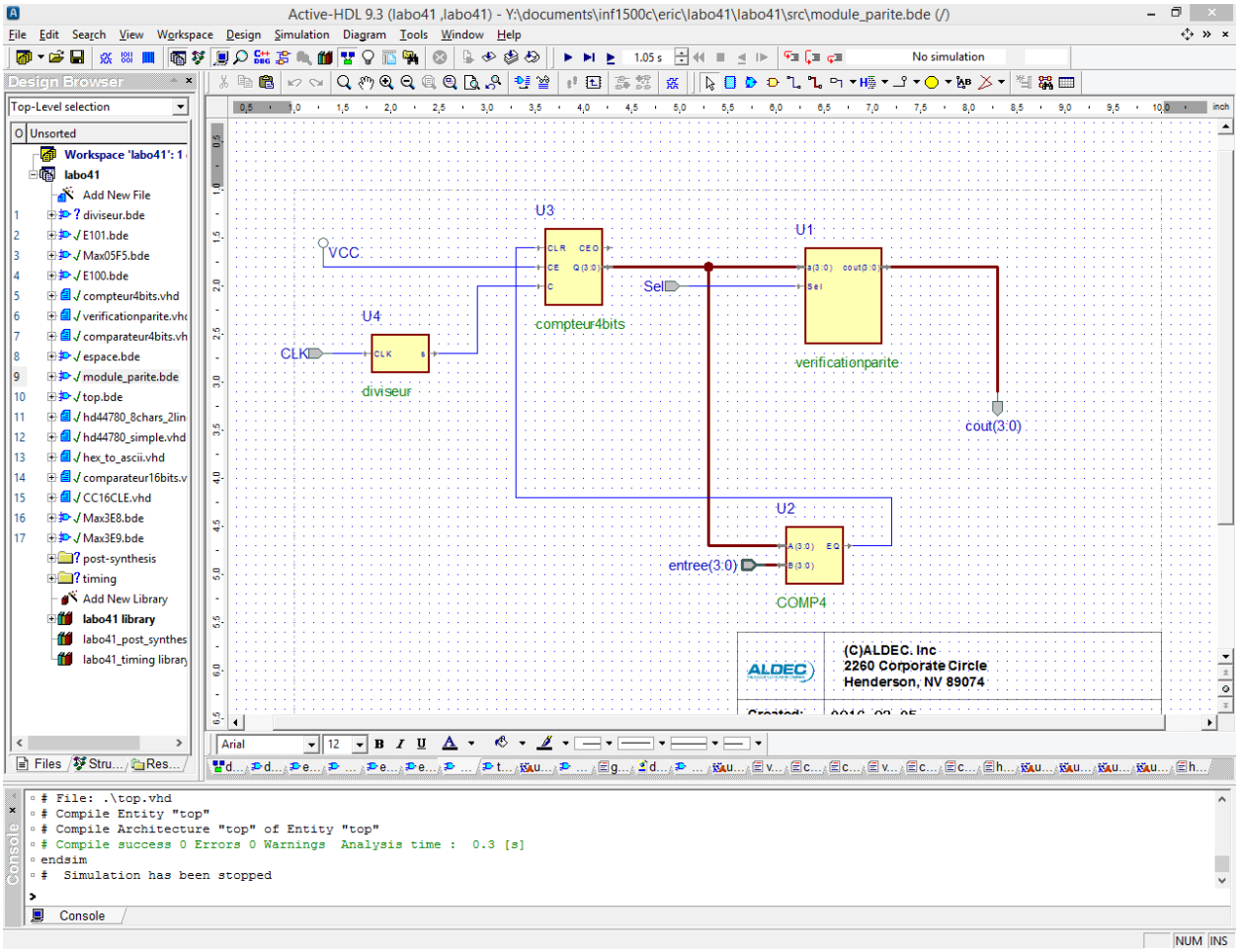

Figure 10 : Module de parité.

Par observation, le bloc U2 compare l'entrée établie par l'usager avec la sortie du compteur U3. Quand ces derniers sont égaux, le compteur recommence de compter à partir de 0, puisqu'un signal est actif haut est envoyé vers l'entrée CLR de ce compteur. De plus, les valeurs que la sortie Q prend sont envoyées au bloc U1 vérificateur de parité, qui servira à la trie de ces valeurs. Comme expliqué précédemment, quand l'entrée sel est à 1, les valeurs que la sortie cout prendra sont les valeurs impaires, et quand l'entrée est à 0, les valeurs que la sortie cout prendra sont les valeurs paires.

Par la suite, nous avons effectué une première simulation de ce module :

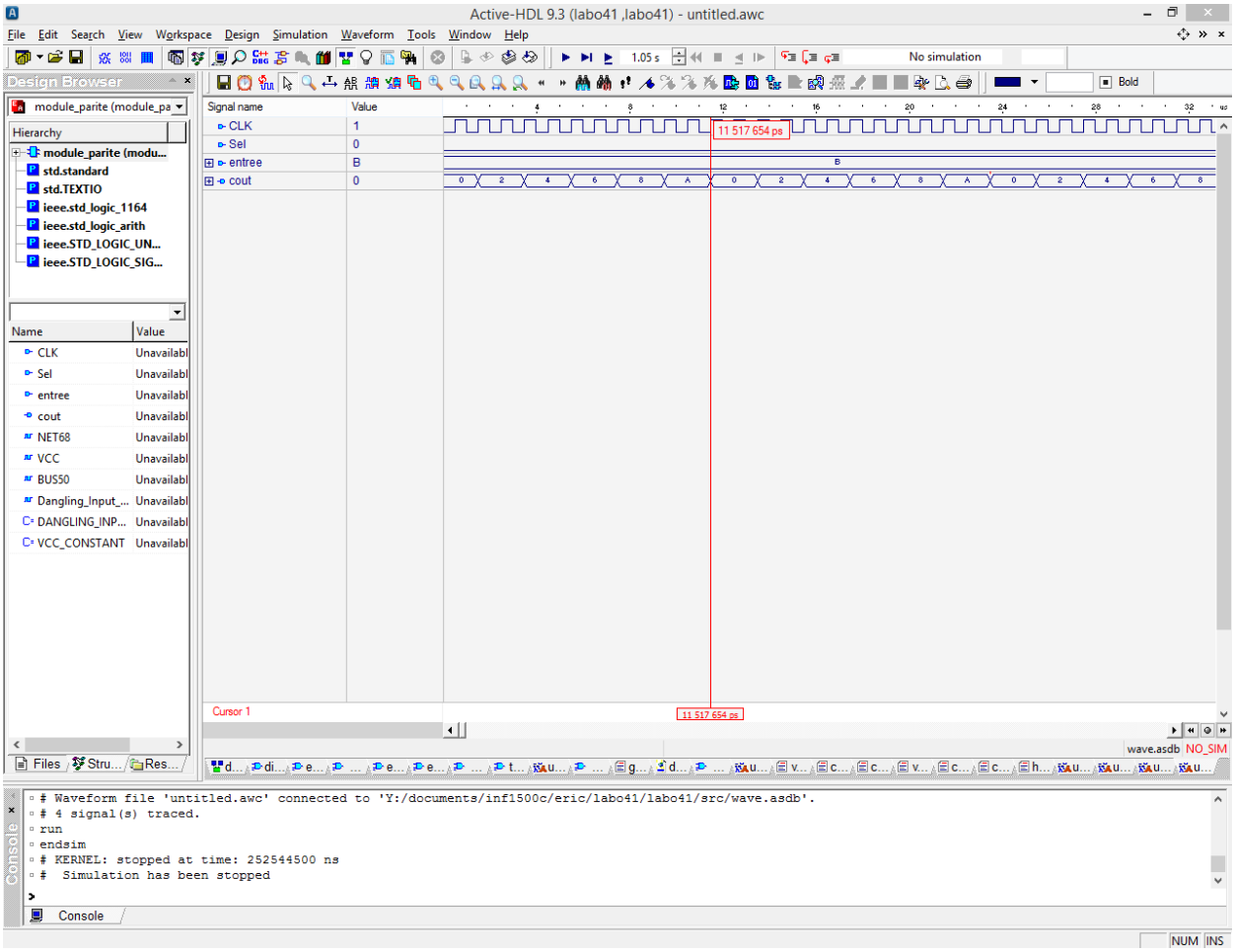

Figure 11 : Simulation du module de parité avec sel à 0 et entree à B_{16}.

En fixant notre entrée « entree » à $(B)_{16}$ et l'entrée sel à 0, on s'attend à obtenir à sortie cout une suite de valeurs pairs ne dépassant pas $(B)_{16}$. Effectivement, c'est le cas, car on obtient 0, 2, 4, 6, 8 et $(A)_{16}$.

De même, on refait l'expérimentation, mais avec l'entrée sel en actif haut :

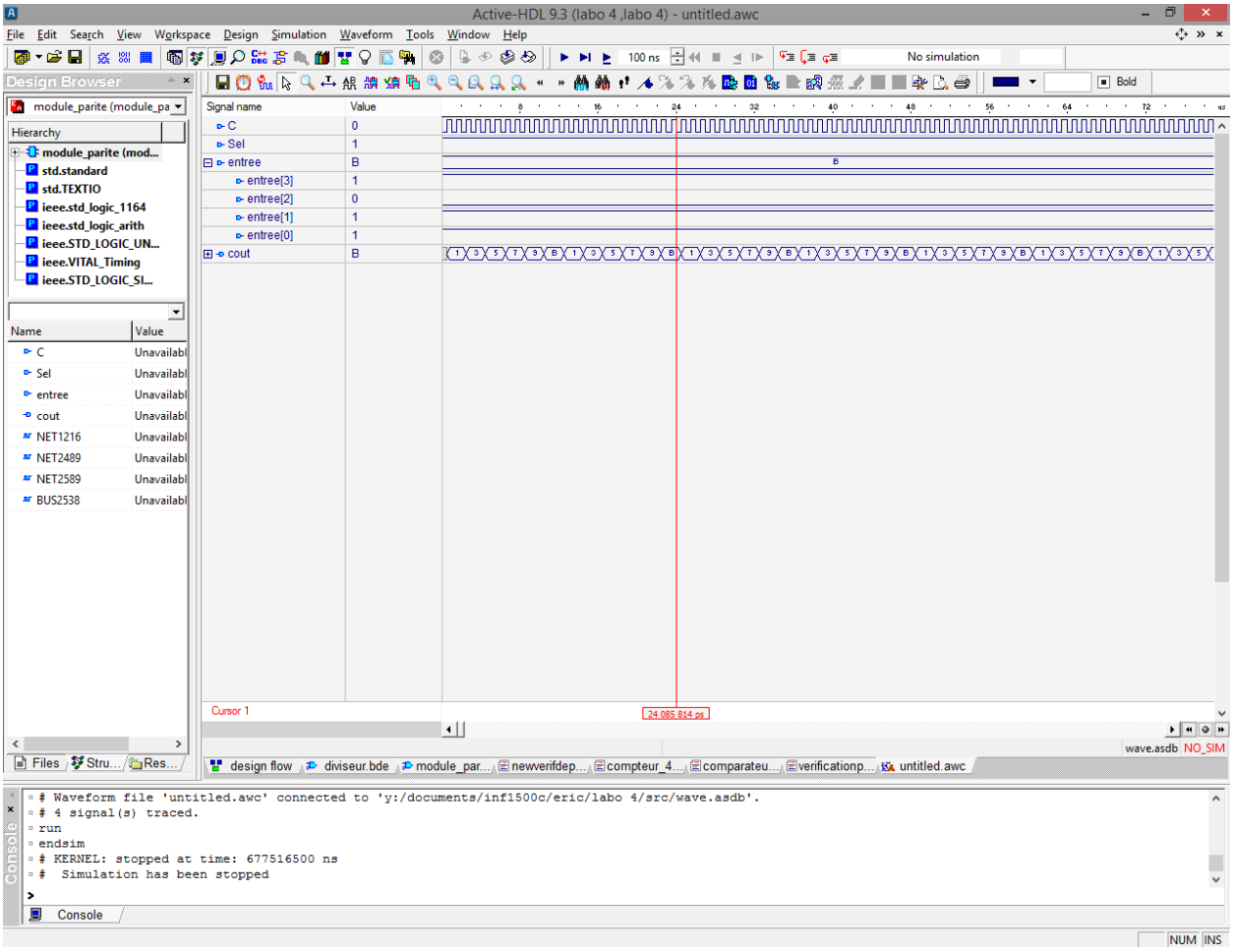

Figure 12 : Simulation du module de parité avec sel à 1 et entree à B_{16}.

En observant la figure, on obtient 1, 3, 5, 7, 9 et $(B)_{16}$. Le module de parité est alors vérifié.

Module Top

Ce module est le circuit final. Il permettra de relier le circuit diviseur de fréquence et le circuit de parité à un écran ACL. Ainsi, il est comme suivi :

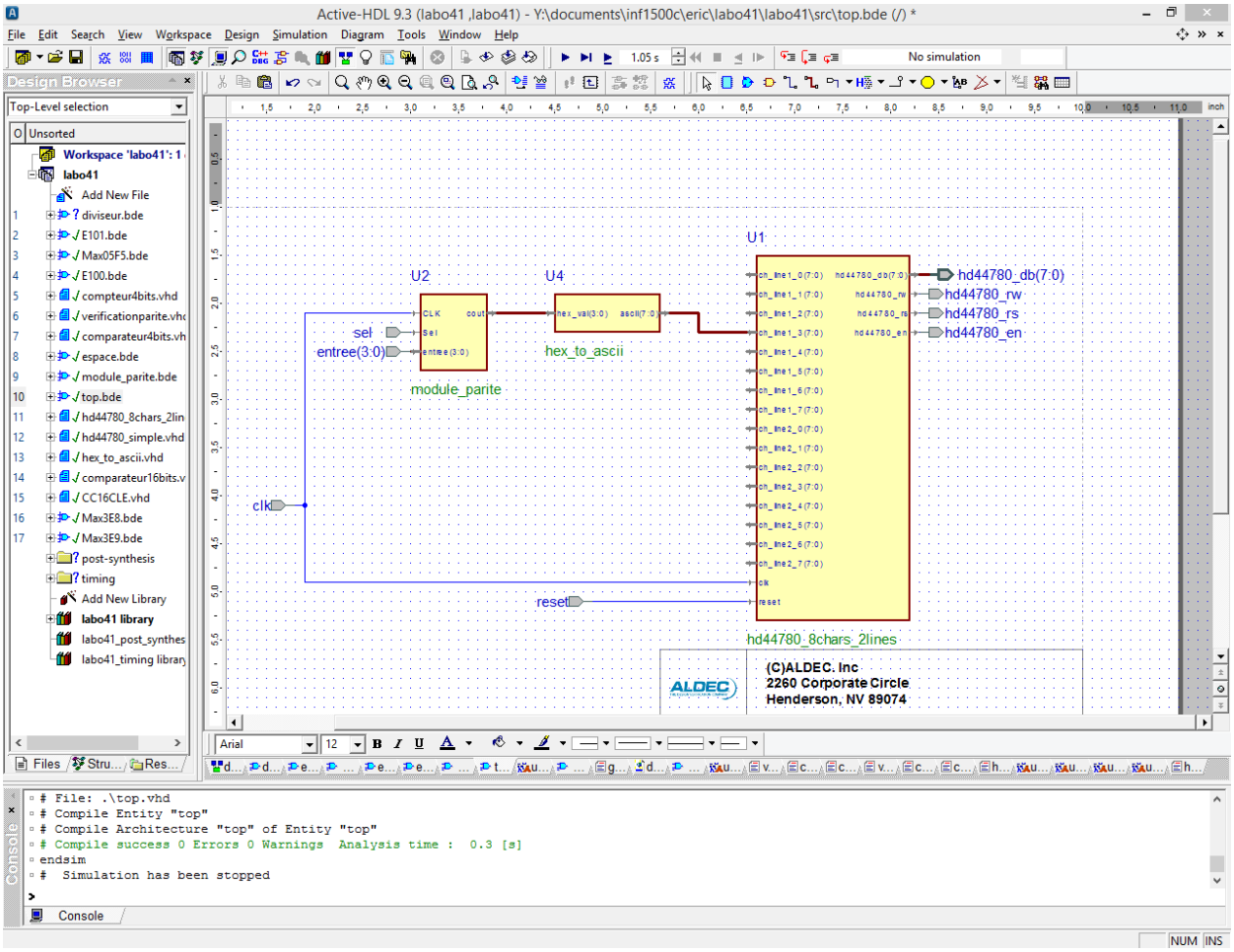

Figure 13 : Module Top.

Premièrement, une conversion en ascii des modules précédents est nécessaire puisque l'écran ne reconnait pas l'hexadécimal. En effet, le bloc U4 le permettait et il était relié à la sortie du bloc U2 du module de parité. Chaque entrée du bloc U1 correspond à un emplacement sur l'écran ACL. Par préférence, on a fixé l'affichage des chiffres à la ligne 1.3 de l'écran. Toutefois, durant l'expérimentation, l'écran n'affichait que la valeur 0. Par hypothèse, cela est dû au fait que, malgré notre module de division de fréquence, la carte génère des valeurs très rapide et impossible à identifier.

Conclusion

Les simulations ont démontré que les modules conçus agissent tel qu'indiqué dans les circuits. Les circuits permettent de réaliser les opérations demandées. Il a été implémenté sur FPGA, nous l'avons testé manuellement, mais l'écran n'affichait que la valeur 0. Par ailleurs, on peut répéter l'expérience avec une entrée à 8 bits à la place de 4 bits pour un niveau de difficulté plus élevé.