

Compte rendu du TP3 : Programmation VHDL d'applications avancées sous ModelSim

Created By: IMHAMED BOUJEMAA

March 9, 2023

1 Introduction

Dans ce compte rendu, nous allons discuter la programmation VHDL d'applications avancées sous ModelSim, en mettant l'accent sur quatres nouvelles applications : le codeur 8 vers 3, le comparateur 8, le détecteur de parité paire, et on termine par une application d'Algorithme de tri a bulles. Nous allons d'abord expliquer brièvement ce qu'est VHDL et ModelSim, puis nous passerons en revue chaque application en détail.

2 VHDL et ModelSim

VHDL (VHSIC Hardware Description Language) est un langage de description matérielle utilisé pour décrire le comportement des circuits électroniques et des systèmes numériques. ModelSim, quant à lui, est un simulateur de circuits numériques utilisé pour vérifier la fonctionnalité des conceptions VHDL.



Figure 1: ModelSim-VHDL

3 Application 1 : Codeur 8 vers 3

Le codeur 8 vers 3 est un circuit combinatoire qui prend en entrée un mot de 8 bits et qui produit en sortie un mot de 3 bits qui identifie la position de l'unique bit à 1 dans le mot d'entrée. Par exemple, si l'entrée est "01001000", la sortie sera "111", car le seul bit à 1 se trouve à la quatrième position.

En VHDL, nous pouvons décrire le comportement du codeur 8 vers 3 en utilisant une instruction de recherche de bits et des instructions de branchement. Le code VHDL pour le codeur 8 vers 3 est présenté dans Github (lien en dessous).

Pour vérifier le fonctionnement de notre codeur 8 vers 3, nous pouvons utiliser ModelSim pour simuler notre circuit et observer les résultats. Nous pouvons également utiliser des vues d'onde pour visualiser les signaux d'entrée et de sortie de notre circuit.

La Figure 2 montre une vue d'onde de la simulation du codeur 8 vers 3 pour l'entrée "01001000". Nous pouvons voir que la sortie est bien "111", ce qui correspond à la position du seul bit à 1 dans l'entrée.

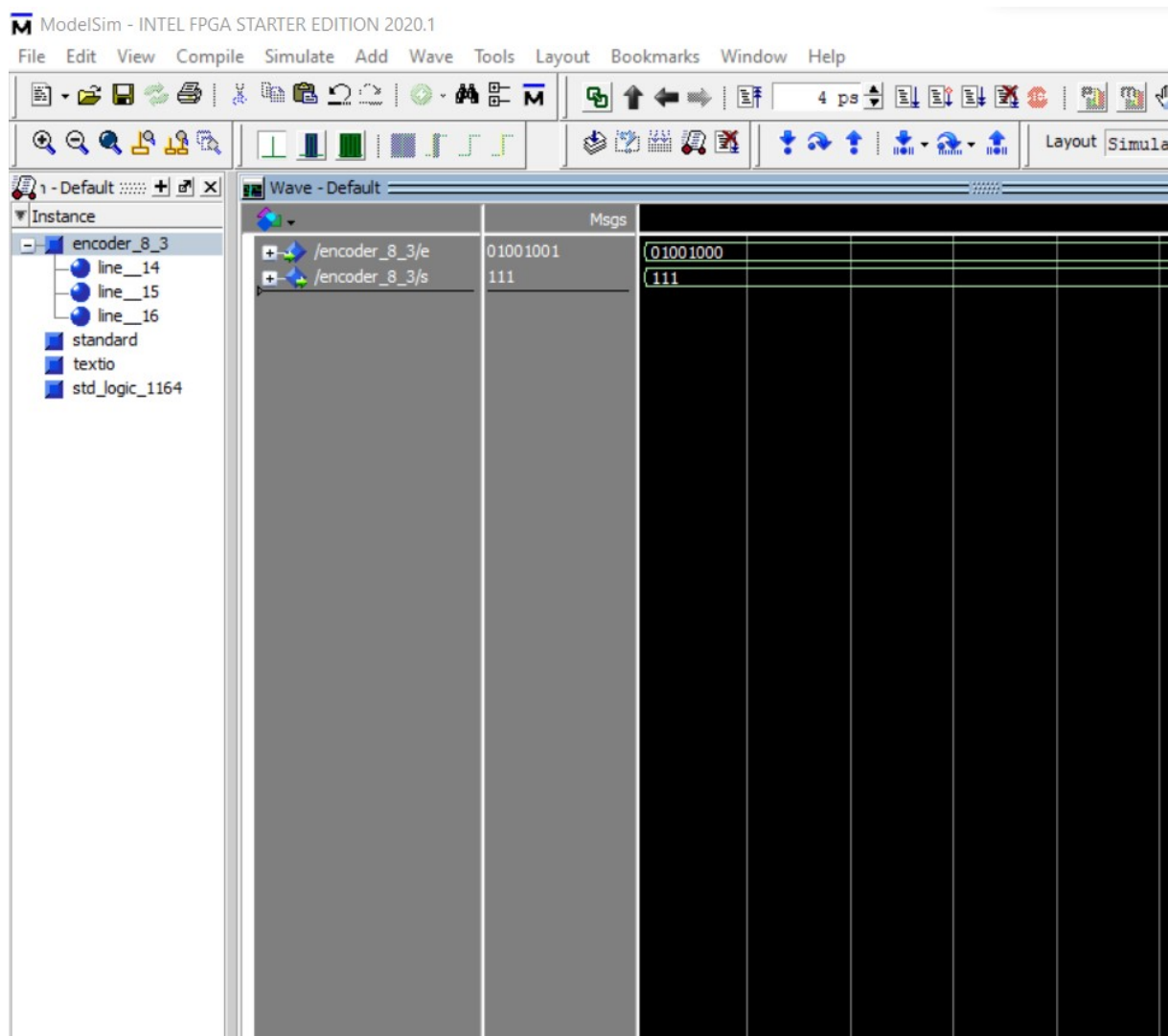


Figure 2: Résultats de la simulation d'un Codeur 8 vers 3

4 Application 2 : Détecteur de parité paire

Un détecteur de parité paire est un circuit combinatoire qui prend en entrée un mot de n bits et qui produit en sortie un bit qui indique si le nombre de bits à 1 dans le mot d'entrée est pair ou impair. Par exemple, si l'entrée est "1010", la sortie sera 0, car il y a deux bits à 1 dans le mot d'entrée.

En VHDL, nous pouvons décrire le comportement du détecteur de parité paire en utilisant des instructions de comptage de bits et de branchement. Le code VHDL pour le détecteur de parité paire est présenté dans Github.

La Figure 3 montre une vue d'onde de la simulation du détecteur de parité paire pour l'entrée "1010". Nous pouvons voir que la sortie est bien 0, car il y a deux bits à 1 dans l'entrée.

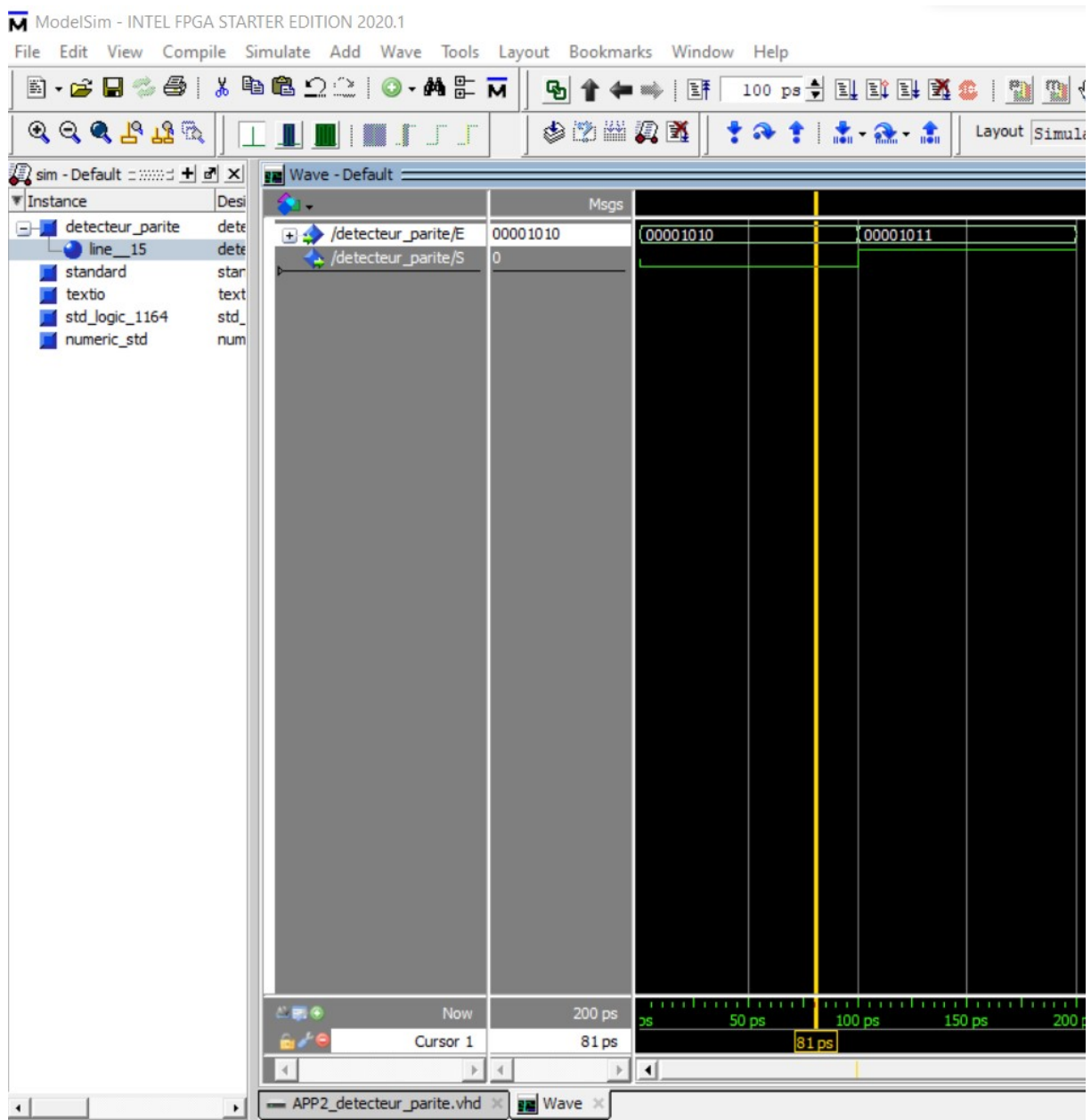


Figure 3: Résultats de la simulation du détecteur de parité

5 Application 3 : Comparateur 8 bits

Le code VHDL présenté dans Github sous le nom "APP3-comparateur-8" définit un comparateur 8 bits. Il prend en entrée deux signaux de 8 bits (A et B) et compare leurs valeurs. Le résultat de la comparaison est indiqué sur les sorties Egal, Sup et Inf.

La sortie Egal est à l'état logique haut ('1') si les deux entrées sont égales, sinon elle est à l'état logique bas ('0'). La sortie Sup est à l'état logique haut ('1') si l'entrée A est supérieure à l'entrée B, sinon elle est à l'état logique bas ('0'). La sortie Inf est à l'état logique haut ('1') si l'entrée A est inférieure à l'entrée B, sinon elle est à l'état logique bas ('0').

Le code VHDL utilise des instructions de comparaison et de branchement pour décrire le comportement du comparateur. Ainsi, lorsque A est égal à B, la sortie Egal est mise à l'état haut. De même, lorsque A est supérieur à B, la sortie Sup est mise à l'état haut, et lorsque A est inférieur à B, la sortie Inf est mise à l'état haut.

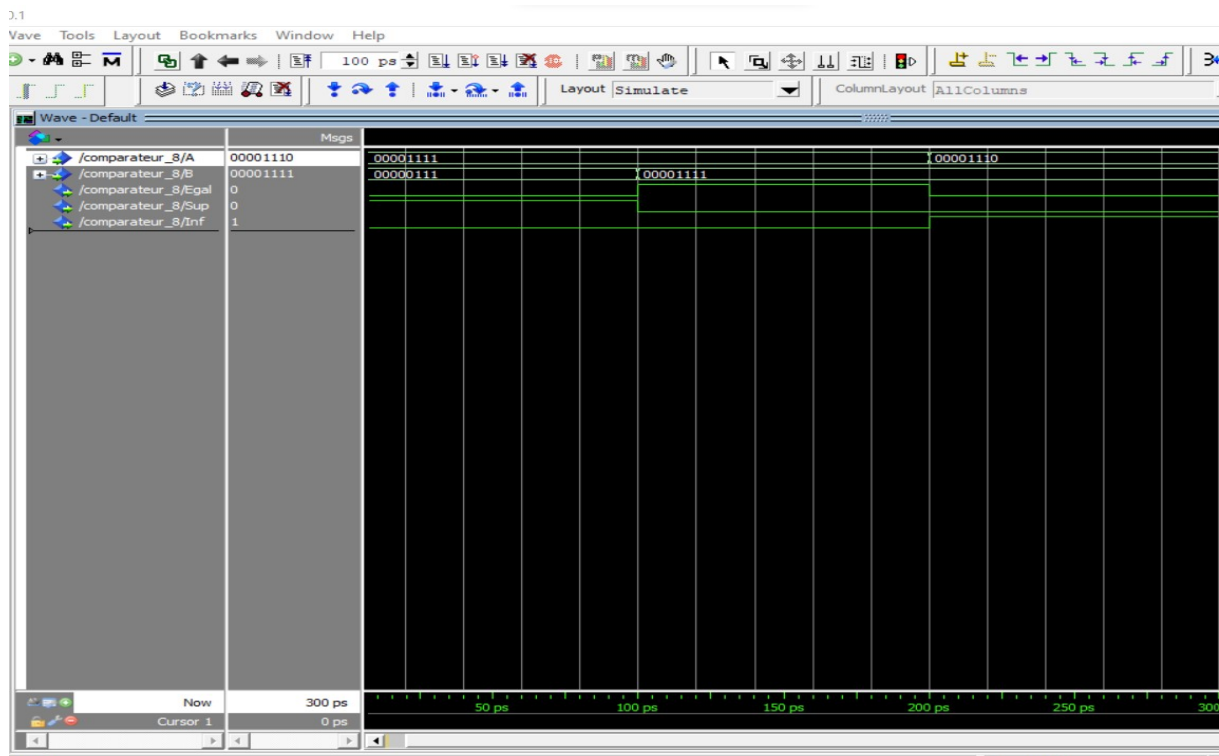


Figure 4: Résultats de la simulation du comparateur 8 bits

6 Application 4 : Algorithme de tri à bulles

6.1 Explication du contexte de l'algorithme

L'algorithme de tri à bulles est un algorithme de tri simple qui fonctionne en comparant chaque paire d'éléments adjacents d'un tableau et en les échangeant si nécessaire. Il répète cette opération plusieurs fois jusqu'à ce que le tableau soit entièrement trié.

L'algorithme de tri à bulles peut être implémenté de différentes manières, mais la méthode la plus courante est de parcourir le tableau de la fin au début, en diminuant à chaque itération la taille de la partie non triée du tableau. Pendant chaque itération, le parcours se fait de gauche à droite en comparant chaque paire d'éléments adjacents et en les échangeant si nécessaire.

Le processus d'échange consiste à comparer deux éléments adjacents et à les échanger si l'élément de droite est plus petit que l'élément de gauche. En d'autres termes, si le tableau est trié par ordre croissant, l'élément de gauche doit être inférieur ou égal à l'élément de droite. Si ce n'est pas le cas, les deux éléments sont échangés.

L'algorithme de tri à bulles est simple à implémenter et à comprendre, mais il peut être inefficace pour trier de grands tableaux car il nécessite un grand nombre d'itérations. Sa complexité temporelle est $O(n^2)$, ce qui signifie que le temps nécessaire pour trier un tableau augmente quadratiquement avec la taille du tableau. Pour cette raison, il est souvent remplacé par des algorithmes plus efficaces comme le tri rapide ou le tri fusion.

7 Conclusion

En conclusion, ce compte-rendu présente la programmation VHDL d'applications avancées sous ModelSim. Nous avons abordé quatre nouvelles applications, à savoir le codeur 8 vers 3, le détecteur de parité paire, le comparateur 8 bits et l'algorithme de tri à bulles.

Nous avons expliqué brièvement ce qu'est VHDL et ModelSim et montré comment ces deux outils peuvent être utilisés pour simuler des circuits numériques et vérifier leur fonctionnement.

Nous avons également fourni des exemples de code VHDL pour chaque application, ainsi que des vues d'ondes pour illustrer les résultats des simulations. Ces exemples permettent de mieux comprendre la programmation en VHDL et l'utilisation de ModelSim.

En résumé, la programmation VHDL est un outil essentiel pour la conception de circuits électroniques et de systèmes numériques. L'utilisation de ModelSim pour simuler les circuits permet de vérifier leur fonctionnement avant leur implémentation réelle, ce qui permet d'économiser du temps et des ressources.

The source code and associated files for this LAB can be found on Github at the following address: https://github.com/IMHAMEDBOUJEMAA/Master_Course_1_2/tree/main/4_FPGA_VHDL/TP_WORK.