

Département de génie informatique et génie logiciel

INF3500

Conception et réalisation de systèmes numériques

**Rapport de laboratoire #1**

**Laboratoire introductif à l’outil Vivado**

|  |  |
| --- | --- |
| **Critères** | **Points** |
| Partie 4 : Conception du module | /6 |
| Partie 5 : Simulation | /6 |
| Partie 6 : Synthèse et implémentation | /5 |
| Partie 7 : Question | /1 |
| Rapport : Présentation et qualité de la langue | /2 |
| Total | /20 |

**Soumis par :**

Alexandre Morinvil, #1897222

Nicolas Valenchon, #2032097

**Date :**

16 janvier 2020

1. **Objectifs**

Les objectifs de ce laboratoire étaient de se familiariser avec l’outil Vivado en implémentant un module simple permettant d’effectuer l’opération *Choose(x, y, z)*. De plus, afin de se familiariser avec ces fonctionnalités, le module implémenté doit être simulé, synthétisé et implémenté sur un support physique.

1. **Description du système**

Le module conçu pour ce laboratoire ne contient qu’un circuit. Le circuit en question implémente la fonction *choose(x,y,z)*. La fonction est une fonction logique pouvant être exprimé ainsi :

Le module est donc conçu en un seul fichier VHDL nommé ch.vhd. Ce fichier comprend la déclaration des entrées du module, soit les valeurs logiques , , , ainsi que la déclaration de la valeur de sortie .

De plus, le fichier ch.vhd comprend également l’implémentation de la fonction elle-même. L’implémentation de la fonction utilisée consiste simplement à calculer les résultats intermédiaires gardé dans les signaux et , correspondant à et . Ensuite, les deux valeurs intermédiaires sont donné en entrée à un opérateur logique ou-exclisif afin d’obtenir le résultat de la fonction .

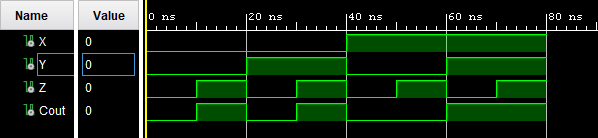
1. **Vérification par simulation**

Le banc d’essai utilisé est implémenté dans le fichier ch\_tb.vhd et il a consisté à réaliser une table de vérité du module. Pour ce le module a été soumis aux 8 combinaisons de valeurs logiques qui est possible de générer avec les trois bits d’entré x#y#z, soit {000, 001, 010, 011, 100, 101, 110, 111}. Pour chacune des combinaisons de valeurs logiques, un intervalle de temps de 10 ns a été imposé avant la transition vers combinaison de valeurs logiques suivante. Par conséquent, la simulation s’étale sur un intervalle de temps de 80 ns. À partir des 8 combinaisons d’entrée, il est donc possible de vérifier si la sortie obtenue correspond à la sortie attendue.

Les 8 tests ont été réalisé avec succès.

Cette stratégie a été sélectionnée pour les cas de teste en raison du fait que le module en question est tout simplement l’implémentation d’une fonction à trois entrées binaire. Ainsi, la sortie attendue est connue et invariable pour une combinaison d’entrée donnée. De plus, puisque le domaine de cette fonction ne comprend que 8 possibilités, cette approche peut raisonnablement être utilisée pour valider l’ensemble des possibilités.

La figure suivante affiche le résultat de la simulation réalisée avec le banc de teste décrit ci-haut :



**Figure 1 :** Resultat de la simulation du module choose(x, y, z) pour les 8 valeurs logiques des entrées

Ainsi, l’on observe que dans l’intervalle allant de 0 à 40 ns la variable est égale à 0 et la sortie correspond à la variable . De plus dans l’intervalle allant de 40 ns à 80 ns la variable est égale à 1 et la sortie correspond à la variable . Ceci correspond au résultat attendu

1. **Réponse à la question**

**Question :** La fonction est le préﬁx de choose. Expliquez de façon concrète et concise pour quelle raison cette fonction est appelée ainsi. Vous pouvez donner des exemples. INDICE! Faites une table de vérité.

**Réponse :**

Le préfixe désigne le mot « choose », ce qui signifie « choisir » en français.

Cette fonction désigne donc le mot « choisir » parce qu’elle permet de choisir de mettre en sortie une de ces entrées en utilisant la valeur d’une de ses variables comme choisisseur.

Ainsi Selon la valeur de x, 0 ou 1, on assigne à la sortie la valeur de Z ou de Y respectivement. Si l’on souhaite assigner à la sortie la valeur de Y, on assigne 1 à X, sinon si l’on souhaite assigner à la sorte la valeur de Z, alors on assigne 0 à X.

La table de vérité de la fonction est affichée ci-bas afin d’illustrer cette idée :

|  |  |  |  |
| --- | --- | --- | --- |
| x | y | z |  |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

**Figure 2 :** Table de vérité de la fonction