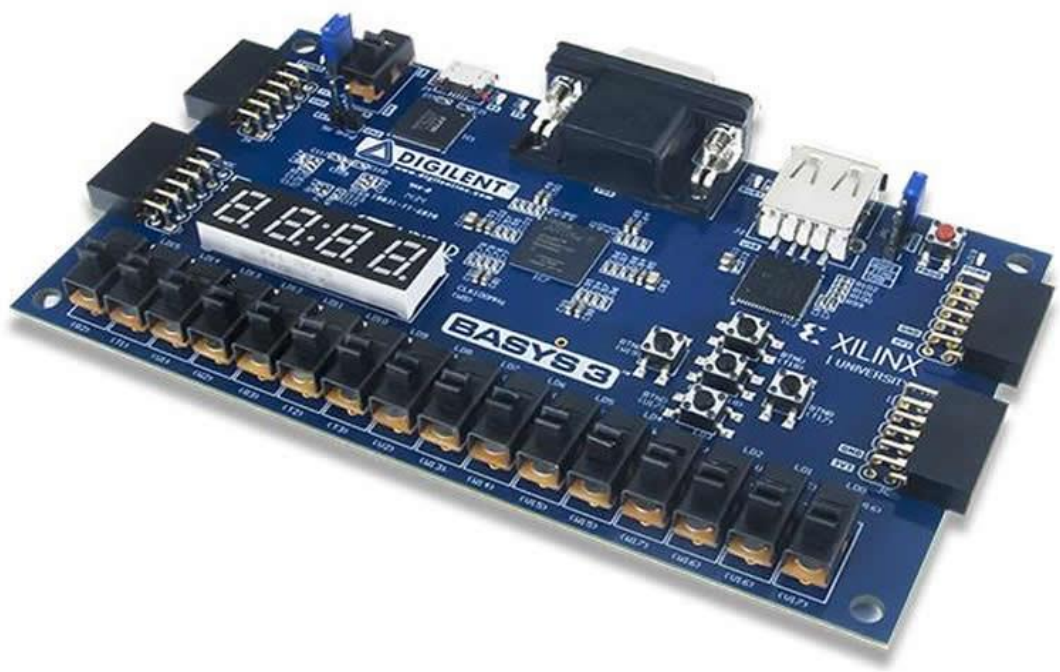
Conception de Circuits Numériques

Livrable du projet d'initiation au VHDL :
Conception d'un mini-ordinateur avec un FPGA

Année scolaire 2022-2023



NDE NOUMI Steve Darius

MOHAMMADOU Abbo OUSMANOU

Travail encadré par : Jérémie JOUSSE

Table des Matières

INTRODUCTION

Le VHDL (VHSIC Hardware Description Langage) est un langage de description de matériel (HDL) qui permet de décrire le comportement et la structure de circuits électroniques numériques. Le FPGA (Field-Programmable Gate Array) est un type de circuit intégré programmable qui peut être utilisé pour implémenter des circuits numériques.

Dans ce rapport, nous allons présenter une étude de cas de mise en œuvre d'un circuit numérique en VHDL en utilisant un FPGA. Nous montrerons comment nous avons conçu, implémenté et testé un circuit simple à l'aide de VHDL et de l'outil de développement FPGA.

Ce rapport académique a pour objectif d’évaluer nos connaissances quant à la compréhension des circuits intégrés qu’à la maitrise des outils de base du langage VHDL.

1- Cahier des charges

Objectif :

L'objectif de ce projet est de concevoir et de réaliser un mini-ordinateur qui peut être contrôlé à l'aide d'un joystick. Le mini-ordinateur doit être capable d'afficher des informations sur la température et la distance mesurée par le sonar sur une fenêtre GKterm, et de contrôler les LED de la carte Basys3 à l'aide du joystick.

Périphériques :

Le mini-ordinateur doit inclure les périphériques suivants :

- Récepteur I2C pour capter la température
- Récepteur SPI pour le joystick
- Un sonar pour mesurer la distance

Fonctionnalités :

Le mini-ordinateur doit être capable de :

- Récupérer la température grâce au récepteur I2C et de l'afficher dans une fenêtre GKterm.
- Mesurer la distance à l'aide du sonar et de l'afficher dans une fenêtre GKterm.
- Contrôler les LED de la carte Basys3 à l'aide du joystick en utilisant l'axe X :
- Les LED doivent être allumées ou éteintes en fonction de la position du joystick.

Contraintes techniques :

Le projet doit respecter les contraintes techniques suivantes :

- Le mini-ordinateur doit être réalisé en utilisant le VHDL pour la description du matériel.
- La carte Basys3 doit être utilisée pour la réalisation du mini-ordinateur.
- Les périphériques doivent être connectés à la carte Basys3 en utilisant les ports appropriés.
- Le projet doit être réalisé en utilisant l'outil de développement FPGA Vivado.
- Le logiciel GKterm doit être utilisé pour afficher les informations sur la température et la distance mesurée.

Livraison :

Le projet doit être livré avec les éléments suivants :

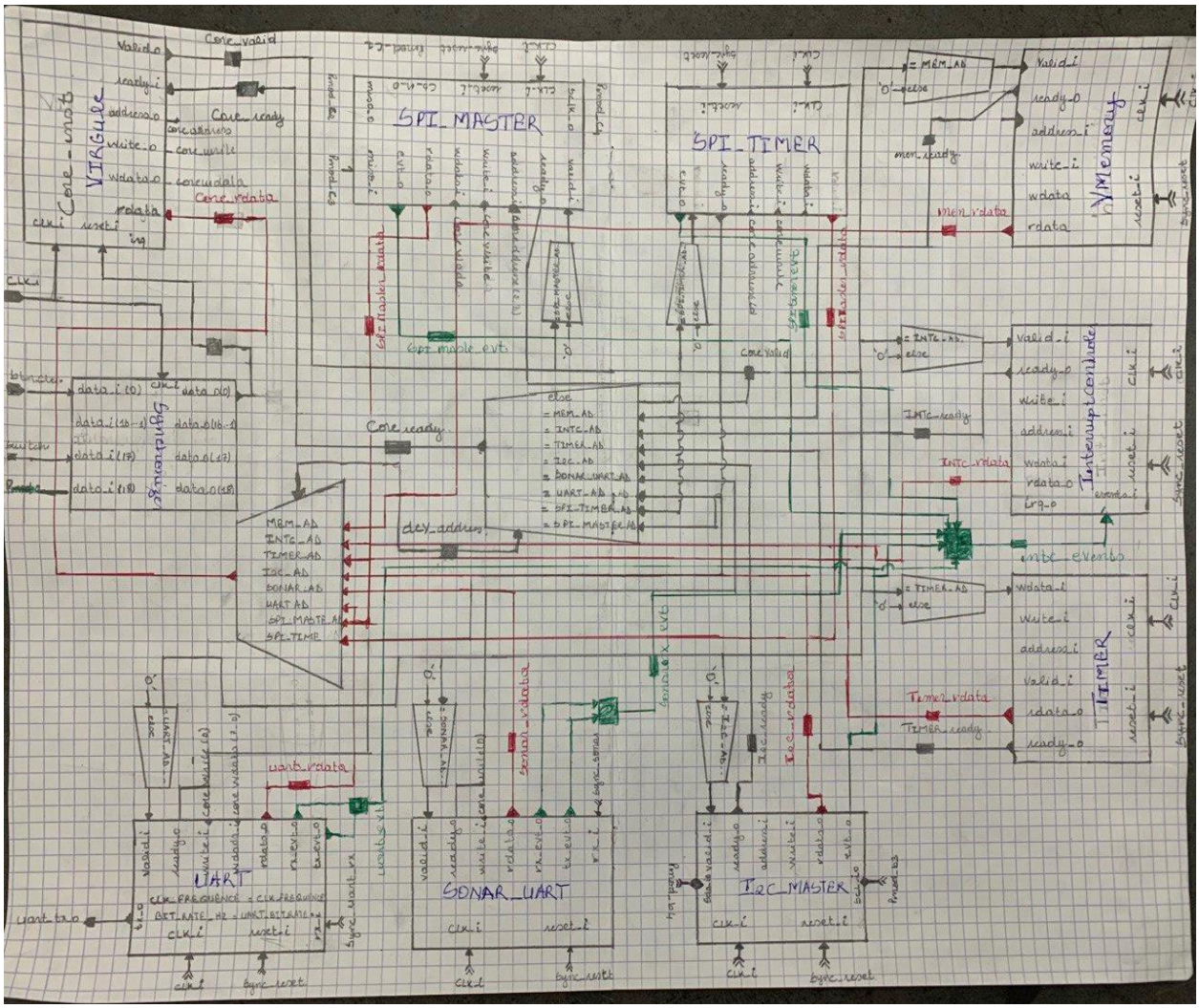
- Le code VHDL pour la description du matériel.
- La documentation complète du projet, comprenant les spécifications techniques, le plan de conception et le manuel d'utilisation.
- Les fichiers de configuration pour le projet Vivado.

Calendrier :

Le projet doit être terminé dans un délai de 3 semaines à compter de la date de début du projet.

2- Block design du mini-ordinateur

- Schéma



- **Explication du schéma**

Le protocole de communication entre les différents composants du système est basé sur les signaux valid_i, ready_o et intc_events.

- Le signal valid_i est utilisé pour indiquer qu'une donnée est prête à être échangée entre deux composants. Le composant source met à jour les signaux appropriés (adresse, données, etc.) et le signal valid_i pour indiquer qu'une opération doit être effectuée.
- Le signal ready_o est utilisé par le composant cible pour indiquer qu'il est prêt à recevoir ou à envoyer des données. Une fois que le composant source a mis à jour les signaux appropriés et le signal valid_i, il attend que le composant cible mette à jour le signal ready_o avant de poursuivre l'opération.
- Le signal intc_events est utilisé pour la gestion des interruptions. Le contrôleur d'interruption surveille les signaux valid_i et intc_events pour détecter lorsqu'une opération est demandée et pour déterminer quelle opération doit être effectuée. Lorsqu'une opération est terminée, le contrôleur d'interruption met à jour le signal ready_o et envoie un signal d'interruption au processeur pour l'informer qu'une interruption s'est produite.

Le processeur exécute des instructions stockées dans la mémoire. Il peut lire ou écrire des données à partir de la mémoire ou des contrôleurs d'entrée/sortie en utilisant le protocole de communication décrit ci-dessus. Le processeur peut également générer des demandes d'interruption en envoyant un signal intc_events approprié au contrôleur d'interruption.

3- Description des différents types de communication

- Les interfaces série asynchrone :

Une interface série asynchrone est utilisée pour transférer des données entre deux dispositifs électroniques. Elle utilise une liaison série pour envoyer des données bit par bit, contrairement à une liaison parallèle qui envoie plusieurs bits simultanément.

En résumé, une interface série asynchrone permet la transmission de données bit par bit entre deux dispositifs électroniques, en utilisant une trame de données comprenant un bit de départ, des données, un bit de parité et un bit de stop, avec

une synchronisation des horloges et une détection d'erreurs pour assurer une transmission fiable.

- Le bus I2C

Le bus I2C (Inter-Integrated Circuit) est un protocole de communication série synchrone utilisé pour connecter plusieurs périphériques électroniques entre eux sur une carte électronique ou un circuit intégré.

En résumé, le bus I2C est un protocole de communication série synchrone utilisé pour connecter plusieurs périphériques électroniques sur une carte électronique ou un circuit intégré, en utilisant deux lignes de communication pour la transmission et la synchronisation des données.

- Le bus SPI

Le bus SPI (Serial Peripheral Interface) est un protocole de communication série synchrone utilisé pour connecter plusieurs périphériques électroniques entre eux sur une carte électronique ou un circuit intégré.

Le bus SPI peut être utilisé pour communiquer avec plusieurs périphériques esclaves en utilisant une ligne de sélection distincte pour chaque périphérique. Le bus SPI prend également en charge plusieurs modes de fonctionnement, qui définissent la polarité et la phase des signaux d'horloge pour une communication plus flexible entre les périphériques.

En résumé, le bus SPI est un protocole de communication série synchrone utilisé pour connecter plusieurs périphériques électroniques sur une carte électronique ou un circuit intégré, en utilisant quatre lignes de communication pour la transmission, la synchronisation et la sélection des périphériques esclaves.

4- Plan d'adressage

Le plan d'adressage permet d'attribuer une plage d'adresses à chaque bloc de mémoire ou à chaque périphérique. Pour définir ces plages d'adresses, nous avons choisi un groupe de bits d'adresses qui a joué le rôle d'identifiant pour le bloc en question. Nous avons utilisé les 8 bits de poids fort de l'adresse comme identifiant comme suit :

- Les bits de 31 à 24 de l'adresse indiquent quel bloc de mémoire ou quel périphérique est concerné par une opération de lecture ou d'écriture.
- Les bits de 23 à 0 représentent l'adresse de la cellule à lire ou à écrire dans ce bloc de mémoire ou ce périphérique.

Adresse (bits 31 à 24)	plage	composant
$0000\ 0000_{bin}$	$00000000_{hex} - 00FFFFFF_{hex}$	Mémoire (VMemory)
$1000\ 0000_{bin}$	$80000000_{hex} - 80FFFFFF_{hex}$	IO_MASTER
$1000\ 0001_{bin}$	$81000000_{hex} - 81FFFFFF_{hex}$	VInterruptController
$1000\ 0010_{bin}$	$82000000_{hex} - 82FFFFFF_{hex}$	UART
$1000\ 0011_{bin}$	$83000000_{hex} - 83FFFFFF_{hex}$	TIMER
$1000\ 0100_{bin}$	$84000000_{hex} - 84FFFFFF_{hex}$	SONAR_UART
$1000\ 0101_{bin}$	$85000000_{hex} - 85FFFFFF_{hex}$	I2C_MASTER
$1000\ 0111_{bin}$	$86000000_{hex} - 86FFFFFF_{hex}$	SPI_TIMER
$1000\ 1000_{bin}$	$87000000_{hex} - 87FFFFFF_{hex}$	SPI_MASTER

.

5- Difficultés rencontrées

Tout au long de notre projet d'initiation au VHDL sur FPGA, nous avons été confrontés à diverses difficultés. L'une des principales difficultés était la compilation du code VHDL, tandis que la réalisation du code en langage C pour faire fonctionner le projet a également posé des problèmes, notamment en ce qui concerne le sonar, qui a été particulièrement difficile à initialiser. Pour résoudre ces problèmes, nous avons cherché des solutions en consultant des tutoriels sur YouTube, ce qui nous a permis de progresser et de surmonter les obstacles.

Conclusion

Au terme de ce projet d'initiation au VHDL sur FPGA, nous avons pu acquérir de solides connaissances en matière de programmation de circuits numériques. Nous avons également découvert les nombreuses possibilités offertes par les FPGA en termes de conception de systèmes électroniques.

Malgré les difficultés que nous avons rencontrées, nous avons réussi à mener à bien notre projet grâce à notre persévérance et à notre collaboration. Nous avons également été en mesure de trouver des solutions à nos problèmes grâce à notre travail de recherche et aux tutoriels disponibles en ligne.

Nous sommes convaincus que les compétences que nous avons acquises au cours de ce projet nous seront très utiles dans nos futurs projets et notre carrière professionnelle. En fin de compte, nous sommes très fiers du résultat final de notre projet et sommes heureux d'avoir pu relever ce défi avec succès.