

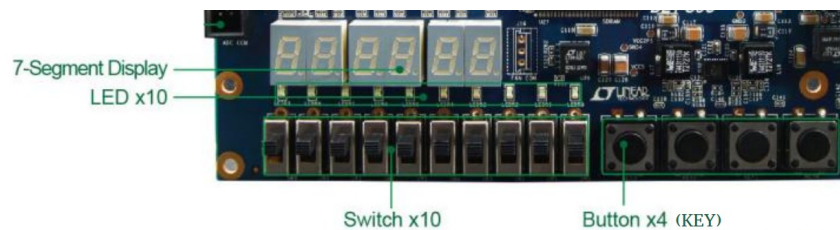
Laboratoire 04 – IP AXI4-Lite avec FPGA I/O

Objectifs du laboratoire

Ce laboratoire a pour but de réaliser une IP avec une interface AXI4-Lite et connectée sur le bus Lightweight HPS-to-FPGA à partir d'une implémentation VHDL fournie. Cette IP doit permettre d'accéder à des I/O câblées sur la partie FPGA via des registres.

Spécifications

L'objectif est d'interfacer à l'aide d'une IP AXI4-Lite toutes les I/O disponibles sur la FPGA, sans utiliser des composants PIO, soit les boutons (KEYs), les switches (SW), les LEDs et les afficheurs 7 segments.



Voici le plan d'adressage de l'IP

Offset	Data	R/W
0x00	Constant (0xBADB100D)	R
0x04	Test Register	RW
0x08	Input register (Keys)	R
0x0C	Edge capture register (Keys)	RW
0x10	Input register (Switch)	R
0x14	Output register (LED)	RW
0x18	Set register (LED)	W
0x1C	Clear register (LED)	W
0x20	Output register (HEX3-0)	RW
0x24	Output register (HEX5-4)	RW

Les IO entre parenthèses servent uniquement à vous indiquer quel IO devra être connecté sur le registre dans le top. Les registres sont implémentés de manière générique. Vous devez implémenter, dans le code VHDL de l'ip fourni, les fonctionnalités des cases en orange :

- Edge capture register : Lorsqu'un bit du registre d'input précédent passe de 0 à 1 (et uniquement dans ce sens), le bit correspondant dans ce registre est mis à

1. Même lorsque le bit du registre d'input repasse à 0, le bit de capture doit rester à 1. Le bit ne revient à 0 que lorsqu'on écrit 1 à l'emplacement du bit de capture
- Set register : Ecrire 1 sur un bit à cette adresse met à 1 le bit correspondant dans le registre d'output précédent
- Clear register : Ecrire 1 sur un bit à cette adresse met à 0 le bit correspondant dans le registre d'output précédent

Spécifications du programme

Le programme devra d'abord vérifier la présence de l'IP en lisant la constante puis en faisant un test d'écriture-lecture sur le registre de test. Si l'un ou l'autre de ces test échoue, terminer le programme. Si les tests se passent bien alors le programme doit effectuer les fonctions suivantes :

- Le programme contient une variable dont les afficheurs HEX0-3 doivent afficher la valeur décimale
- Lors d'un appui sur KEY0, la variable prend la valeur des switch
- Lors d'un appui sur KEY1, la variable est décrémentée
- Lors d'une appui sur KEY2, la variable est incrémentée
- Lors d'une appui sur KEY3, la variable est remise à 0
- Lorsque l'on essaie d'incrémenter/décrémenter la valeur alors qu'elle se trouve déjà à 1023/0, Allumer toutes les leds. Les éteindre à la prochaine opération légale

Travail demandé

- 1) Récupérer les fichiers du laboratoire en effectuant un *git pull* dans le répertoire du labo. Vous pouvez repartir du dossier labo1 et y ajouter le fichier contenu dans le dossier labo3
- 2) Créer un nouveau projet dans Quartus comme lors du premier laboratoire, puis ouvrir Platform Designer.
- 3) Implémenter les fonctionnalités supplémentaires demandées dans le fichier *IP_axi4lite_slave_interface/src/axi4lite_slave.vhd*
- 4) Dans le projet Qsys fourni, créer un nouveau composant (File → New Component) afin de rajouter votre IP à Qsys. Compléter les champs des différents onglets, et notamment :
 - Onglet "Files" : dans la partie Synthesis Files, rajouter le fichier "axi4lite_slave.vhd".
 - Onglet "Signals & Interfaces" : Ajouter une interface "AXI4Lite Slave", "Clock Input", "Reset Input" et "Conduit". Faire glisser les signaux dans l'interface leur correspondant, configurer les interfaces et les signaux associés. Supprimer l'interface Avalon créée par défaut.
- 5) Ajouter le composant fraîchement créé dans le système Qsys, réaliser les connexions, les exports nécessaires ainsi que l'adressage du composant.
- 6) Générer les fichiers HDL du projet Qsys. ATTENTION : toute modification de l'IP impose de re-générer les fichiers HDL du projet Qsys avant de faire la synthèse avec Quartus, sous peine que les modifications ne soient pas prises en compte.
- 7) Adapter le top du projet (DE1_SoC_top.vhd).
- 8) Synthétiser et faire le placement-routage du projet.
- 9) Réaliser le code C de manière à respecter les spécifications données précédemment.
- 10) Créer un nouveau projet Altera Monitor Program puis compiler le code et le tester sur la carte DE1-SoC.
- 11) Faites valider votre montage par l'assistant

Documents à rendre

Vous devez rendre un rapport à l'issue de ce laboratoire contenant les explications sur les différentes étapes de la réalisation de votre système.

Vous devez également rendre une archive avec les sources du projet pour Quartus, Qsys et le programme C. Utiliser le Makefile fourni pour générer votre archive à rendre en tapant "make zip" dans un terminal.

Les fichiers sont à rendre sur Cyberlearn.