

Utilisation des I/O et des interruptions entre la partie FPGA et le HPS

Objectif :

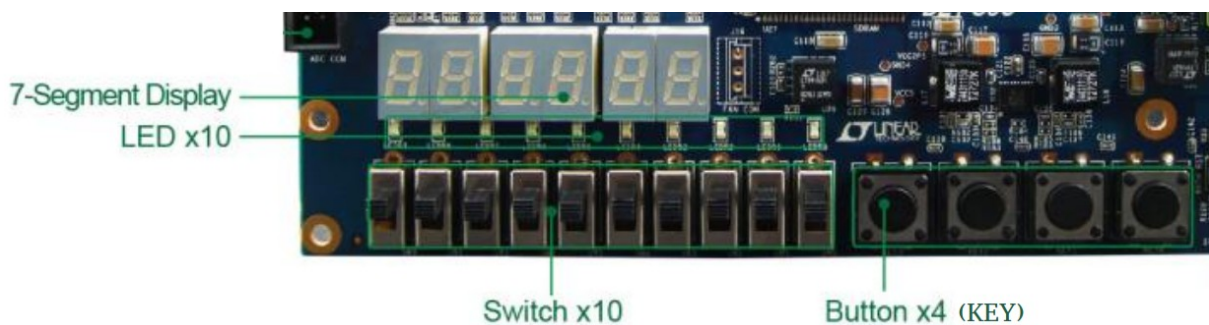
Ce laboratoire a pour but d'accéder à des I/O câblées sur la partie FPGA. Il s'agira d'ajouter des blocs PIO pour interfacer ces I/O sur le HPS. Vous devrez comprendre comment ajouter des IP disponibles dans Qsys pour construire des interfaces permettant d'accéder aux I/O de la FPGA depuis le HPS.

Dans un deuxième temps, vous utiliserez les interruptions du HPS, qui seront générées par un PIO que vous utilisez déjà. Vous devrez comprendre le mécanisme des interruptions sur la Cyclone V SoC afin de bien gérer la gestion de celles-ci.

Spécifications sans les interruptions :

Le but est d'allumer les LEDs et les afficheurs 7 segments selon l'état des boutons (KEY) et interrupteurs (switch) disponibles. La spécification du fonctionnement est la suivante :

- Appui sur KEY0 : les LEDs s'allument selon la position des différents switches. Les afficheurs HEX0 et HEX1 traduisent en hexadécimal les valeurs représentées par les LED3 à LED0 et LED7 à LED4 respectivement. Les afficheurs HEX2 et HEX3 affichent 1 lorsque la LED8 et respectivement la LED9 sont allumées, 0 sinon.
- Appui sur KEY1 : les LEDs s'allument selon la position inverse des différents switches. Les afficheurs HEX0 et HEX1 traduisent en hexadécimal les valeurs représentées par les LED3 à LED0 et LED7 à LED4 respectivement. Les afficheurs HEX2 et HEX3 affichent 1 lorsque la LED8 et respectivement la LED9 sont allumées, 0 sinon.



Spécifications avec les interruptions :

Complétez votre travail précédent avec les spécifications suivantes :

L'appui sur KEY2 ou KEY3 génère une interruption permettant de réaliser les fonctions suivantes :

- Appui sur KEY2 : l'affichage des LEDs et des afficheurs 7 segments subit une rotation à droite, les afficheurs 7 segments ne reflètent plus les valeurs des LEDs.
- Appui sur KEY3 : l'affichage des LEDs et des afficheurs 7 segments subit une rotation à gauche, les afficheurs 7 segments ne reflètent plus les valeurs des LEDs.

Documents à rendre :

Vous devrez rendre un rapport à l'issu 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 Moodle

Travail demandé :

1ère partie : utilisation uniquement des LEDs et des Switches.

L'application copie simplement l'états des Switches sur les LEDs (simple copie).

- 1) Récupérer votre projet du laboratoire d'introduction, il vous servira de base pour ce travail.
- 2) Ouvrir votre projet Qsys, éditer le composant HPS et activer le bridge AXI lightweight HPS-to-FPGA (32 bits).
- 3) Ajouter et configurer les composants nécessaires au fonctionnement du labo :
 - Clock Source : menu Library → Basic Functions → Clocks; PLL and Resets
 - Fréquence d'horloge : 50MHz
 - PIO : menu Library → Processors and Peripherals → Peripherals
 - utiliser un PIO par groupe de périphérique, soit un pour les LEDs et un pour les Switches.
- 4) Exporter les connexions des PIOs en double-cliquant dans la colonne Export en face de "external_connection". Donner un nom de votre choix, en restant explicite.
- 5) Aller dans l'onglet "Address Map" et configurer les adresses des PIO. Il faut tenir compte qu'un PIO a un minimum de 4 registres de 32 bits accessibles.
- 6) Revenir dans l'onglet "System Contents" et réaliser les connexions entre les différents composants en cliquant sur les petits ronds dans la colonne "Connections".
- 7) Générer les fichiers HDL du projet Qsys.
- 8) Adapter le top du projet (DE1_SoC_top.vhd) en connectant les nouvelles entrées et sorties :
 - CLOCK_50_i : clock du système
 - il n'y a pas de signal de reset
 - SW_i : Switches
 - LEDR_o : LEDs

Pour connaître le nom des entrées et sorties du système Qsys, il y a 2 possibilités :

- dans Qsys, menu Generate → Show Instanciation Template
- grâce au fichier qsys_system_inst.vhd dans le dossier hard/eda/qsys_system

Synthétiser et faire le placement routage du projet.

- 9) Dans un nouveau fichier `fpga_gpio.c`, réaliser le code C pour que les LEDs s'allument en recopiant la position des Switches.
- 10) Créer un nouveau projet Altera Monitor Program nommé `fpga_gpio`.
- 11) Compiler le code et le tester sur la carte DE1-SoC.

2ème partie : réaliser le fonctionnement comme décrit dans les spécifications sans les interruptions.

- 12) Adapter le système en tenant compte des spécifications complètes sans les interruptions.

Rajouter autant de PIO que nécessaire pour gérer les I/O manquants.

Reprendre les étapes 2 à 7 précédentes.

- dans `DE1_SoC_top.vhd` :

- `HEX3_o` : afficheur 7 segments n°3 ⚠ actif à l'état bas, le point n'est pas câblé
- `HEX2_o` : afficheur 7 segments n°2 ⚠ actif à l'état bas, le point n'est pas câblé
- `HEX1_o` : afficheur 7 segments n°1 ⚠ actif à l'état bas, le point n'est pas câblé
- `HEX0_o` : afficheur 7 segments n°0 ⚠ actif à l'état bas, le point n'est pas câblé
- `KEY_i` : boutons (KEY) ⚠ actif à l'état bas

- 13) Compléter votre programme `fpga_gpio.c` afin de répondre à la totalité de la spécification.

- 14) Faire valider votre montage par le professeur ou l'assistant

3ème partie : réaliser le fonctionnement complet avec la prise en charge des interruptions.

- 15) Adapter le système en tenant compte des spécifications complètes avec les interruptions. Une base de code vous est fournie qui permet d'initialiser le HPS pour utiliser les interruptions.