

Laboratoire 04 – Utilisation d'un timer avec les interruptions

Objectifs du laboratoire

Ce laboratoire a pour but d'utiliser un timer avec les interruptions du HPS. Il s'agira de mettre en oeuvre un timer du HPS, puis de générer des interruptions à partir de celui-ci. Vous devrez comprendre le mécanisme des interruptions sur la Cyclone V SoC afin de gérer correctement les différents éléments et étapes de celles-ci. Ensuite vous écrirez un programme qui utilise les interruptions du timer et qui répond aux spécifications données ci-après.

Spécifications

Le projet utilisé pour ce laboratoire reprend exactement les mêmes spécifications concernant les périphériques PIO connectés sur le bus AXI lightweight HPS-to-FPGA que celui utilisé pendant le laboratoire 2. Les spécifications sont redonnées ci-dessous en plus des détails sur le timer.

Plan d'adressage

Voici le plan d'adressage qui est utilisé dans le projet fournit.

Offset on bus AXI lightweight HPS-to-FPGA (relative to BA_LW_AXI)	Fonctionnalités
0x00_0000 – 0x00_0003	Design standard ID 32 bits (Read only)
0x00_0004 - 0x00_00FF	reserved
0x00_0100 - 0x00_011F	Périphérique PIO 0 pour interrupteurs(switch), leds et boutons (KEY).
0x00_0120 - 0x00_013F	Périphérique PIO 1 pour les 4 afficheurs 7 segments (HEX0, HEX1, HEX2 et HEX3).
0x00_0140 – 0x1F_FFFF	not used

Configuration utilisée dans le périphérique PIO 0

PIO Bits	31	24	23	20	19	10	9	0
E/S connectés	unused		Key (3...0)		Leds (9...0)		Switch (9...0)	

- Les ports sont configurés en mode "Bidirectionnel".
- "Individual bit set/clear output register" → activé.
- "Edge capture register" → désactivé.
- "Generate IRQ" → désactivé.
- Les switches et Leds sont actifs à l'état haut. Les keys sont actifs à l'état bas.

Configuration utilisée dans le périphérique PIO 1

PIO Bits	31	28	27	21	20	14	13	7	6	0
E/S connectés	unused		Hex3 (6...0)		Hex2 (6...0)		Hex1 (6...0)		Hex0 (6...0)	

- Les ports sont configurés en mode "Bidirectionnel"
- "Individual bit set/clear output register" → activé.
- "Edge capture register" → désactivé.
- "Generate IRQ" → désactivé.
- Les afficheurs 7 segments sont actifs à l'état bas.

Timer du HPS

- Utilisation du timer module 0 piloté par la clock OSC1.
- Fréquence de la clock OSC1 : 25MHz.

Programme

Le but est de réaliser un minuteur à la dixième de seconde (0.1s) avec la valeur courante affichée sur les afficheurs 7 segments. Les boutons (KEY) et interrupteurs (SW) permettent d'initialiser la valeur de départ du minuteur, ainsi que de démarrer et arrêter celui-ci. La spécification du fonctionnement est la suivante :

Au démarrage, le programme doit remplir les conditions suivantes :

- Les 10 leds sont éteintes.
- Les afficheurs 7 segments Hex0, Hex1, Hex2 et Hex3 affichent la valeur 0000.
- Afficher la constante ID au format hexadécimal dans la console de ARM-DS.
- Le minuteur est arrêté.

Ensuite pendant l'exécution du programme, à tout instant les actions suivantes doivent être respectées :

- Pression sur KEY0 : Démarrage du minuteur.
- Pression sur KEY1 : Arrêt du minuteur.
- Pression sur KEY2 : Chargement d'une nouvelle valeur de départ du minuteur.
 - La valeur de SW7-0 est utilisé comme nouvelle valeur de départ en seconde, varie de 0 à 255 secondes.
 - Cette valeur de départ est copiée sur les Leds 7-0 et affichée sur les afficheurs 7 segments (Hex3, 2, 1 et 0) au format décimal à la dixième de seconde (0.1s). Hex3, 2 et 1 affiche la valeur en seconde et Hex0 affiche le dixième de seconde.

L'affichage doit respecter les conditions suivantes :

- Lorsque le minuteur s'exécute, à chaque 100ms l'affichage de la valeur sur les 7 segments est mis à jours.
- L'état de la Led8 représente si le minuteur est actif ou non.
- Lors de chaque interruption du timer (chaque 100ms), un toggle de la Led9 est effectué.

Travail demandé

- 1) Rechercher dans la documentation du HPS (*cyclone-v_hps_trm_5v4.pdf*) les informations concernant l'utilisation du timer module 0. Identifier les configurations nécessaires pour son utilisation avec les interruptions.
- 2) Rechercher dans la documentation du HPS : Quel est le numéro de la ligne d'interruption GIC qui correspond au timer à utiliser ?
- 3) Etudier la documentation concernant la mise en œuvre des interruptions GIC dans le HPS (*cyclone-v_hps_interrupt_manual_GIC_18.1.pdf*).
- 4) Sur Cyberlearn, récupérer l'archive du projet et la décompresser. Puis générer le bitstream du projet.
- 5) Lors de la création de votre projet dans ARM-DS, respecter les informations données dans le document (*Utilisation_armds_avec_interruption.pdf*).
- 6) Ecrire vos fonctions C qui permettent d'initialiser votre timer et de l'utiliser. Pour les autres périphériques (Leds, switchs, boutons, afficheurs 7 segments) vous pouvez utiliser vos fonctions développées pendant le laboratoire 2.
- 7) Ecrire les fonctions d'initialisation et de configuration des interruptions
 - Compléter la fonction `config_GIC()` du fichier `exceptions.c` fourni. La fonction doit configurer le GIC pour utiliser l'interruption du timer.
 - Compléter la routine d'interruption `__cs3_isr_irq()` du fichier `exceptions.c` fourni. La routine d'interruption doit être la plus rapide possible.
 - Ecrire une routine d'interruption `hps_timer_ISR()` dédiée aux interruptions du timer.
- 8) Tester les interruptions sur la carte DE1-SoC en complétant l'application principal.
 - Appeler les différentes fonctions d'initialisation écrites précédemment.
 - Initialiser le mode interruption du processeur en utilisant les fonctions `set_A9_IRQ_stack(void)` et `enable_A9_interrupts(void)` du fichier `exceptions.c` fourni.
 - Lors d'une interruption du timer, réaliser seulement le toggle d'une led dans la routine d'interruption.
- 9) Ecrire le programme principal suivant les spécifications et tester sur la carte DE1-SoC.
- 10) Faire valider votre programme par le professeur ou l'assistant.

À rendre

Ce laboratoire est évalué par une évaluation orale. Il n'y a pas de rapport à rédiger. Un contrôle du travail sera effectué. Vous devez rendre une archive avec les sources du projet pour Quartus et le programme C. Utiliser le Makefile à la racine du projet pour générer votre archive à rendre en tapant « make zip » dans un terminal.

Les fichiers sont à rendre sur Cyberlearn à la date indiquée.