

## Gestion d'une interruption depuis la FPGA – REPTAR

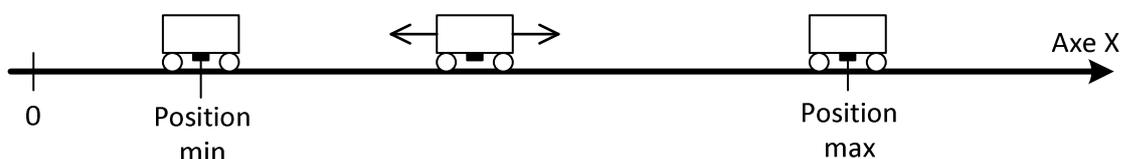
### Gestion de la position d'un chariot

#### Objectif :

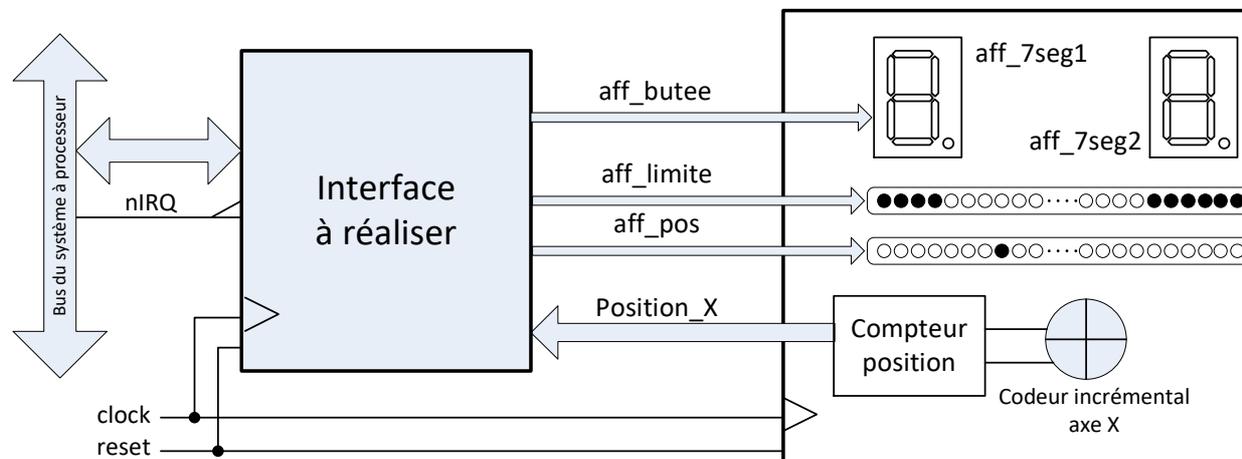
Cette manipulation a pour objectif de mettre en œuvre une interruption sur la plateforme REPTAR. Il s'agit aussi bien de réaliser la partie matériel dans la FPGA que le traitement de celle-ci dans par le CPU. L'objectif est de permettre aux étudiants de comprendre et d'utiliser des interruptions au niveau matériel et logiciel bas niveau (sans OS).

#### Description du système :

Nous allons simuler que nous disposons d'un axe sur lequel un chariot se déplace. Cet axe à une longueur finie et dispose de deux butées logicielles. Voici une vue de l'axe:



Nous allons utiliser le bouton rotatif (codeur incrémental) pour simuler le déplacement du chariot sur l'axe X. Un compteur 16 bits permettra de mesurer la position du chariot. Nous utiliserons un afficheur 7 segments pour indiquer dans quel sens doit aller le chariot (commande manuel du mouvement !)



Pour ce labo, nous allons utiliser les leds disponibles sur une carte connectée avec la carte REPTAR via le connecteur 80 pôles. Nous disposons ainsi de 2 rangées de 39 leds. Une rangée sera utilisée pour afficher les limites (butées min et max) et le second sera utilisé pour indiquer où se situe le chariot.

## **A rendre :**

Ce laboratoire sera évalué. Il sera réalisé par groupe de 2 étudiants.

Vous devez rendre à l'issu de ce travail de laboratoire 2 fichiers, soit :

- **Un fichier PDF** avec votre rapport de laboratoire comprenant toutes les explications sur les étapes demandées dans l'énoncé (1<sup>ère</sup> et 2<sup>ème</sup> parties) ainsi que la copie de tous les documents nécessaires pour la correction (scan documents manuscrit, schéma Logisim, programme C, etc.)
- **Une archive** zip ou tar.gz avec les fichiers suivants pour les parties 1 et 2, soit:
  - les fichiers \*.cir, \*.c, et autres fichiers modifiés  
**sans** le répertoire "include" (il fait 3.3MB!). Merci

Vous devez déposer vos fichiers sur la page IFS du site Moodle (Cyberlearn).

## **Description du fonctionnement :**

Le projet fourni comprend déjà la gestion de la position ainsi que la gestion des affichages 7 segments et la connections du connecteur 80 pôles pour les 2 rangées de leds.

La position du chariot varie de 0x000 à 0x26F, soit 10 bits. La position du chariot sera visualisée en allumant la leds correspondant au nombre représenté par les bits [9 .. 4] de la position. Soit : leds[0] pour 0x00, leds[1] pour 0x01, et ainsi de suite jusqu'à leds[38] pour 0x26.

Vous devez réaliser une interface qui dispose des fonctionnalités suivantes:

- Permet l'initialisation du système de déplacement du chariot par le CPU, soit: position initiale et valeur des butées min et max.
- Générer une interruption lorsque que le chariot dépasse une butée (min ou max).
- Permet la détection par scrutation de la source de l'interruption, soit de la butée min ou max.
- Permet d'afficher le sens du dépassement sur les affichages 7seg par le CPU.
- Permet la quittance de la demande d'interruption par le CPU.

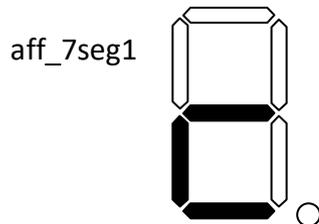
Voici les fonctionnalités demandées au programme

- Le programme doit initialiser la position du chariot (activation signal Init) et les butées min et max avec des valeurs définies par des constantes. Voici les plages de valeurs autorisées pour les butées :
  - butée min: valeur entre 0x020 à 0x0FF
  - butée max: valeur entre 0x170 à 0x24F
- Le programme doit afficher en continu la position courant du chariot et l'état des flags det\_max et det\_min dans la console (printf).
- Lorsque le chariot quitte la zone de travail comprise entre butée min et butée max, vous devez activer le flag correspondant det\_max ou det\_min. D'autre part une interruption est générée dans votre interface, soit activation du signal IRQ.

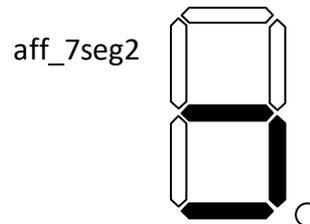
- Lors d'une interruption, vous devez déterminer par scrutation s'il s'agit d'un dépassement en min ou max. Ensuite, vous affichez le sens de dépassement du chariot sur les afficheurs 7 segments pour informer l'utilisateur. Celui-ci doit ramener le chariot dans la zone de travail (bouton rotatif).

Voici le pictogramme à afficher sur les affichages 7 segments

Dépassement butée max:



Dépassement butée min:



- Lorsque le traitement de l'interruption est terminé vous devez quitter celle-ci dans les différents éléments concernés, soit:
  - Quitter l'interruption IRQ dans le contrôleur d'interruption du DM3730 et dans le module GPIO correspondant. Voir information sur l'interruption utilisée ci-après. Cette partie est fournie.
  - Quitter la demande d'interruption dans votre interface.

Vous disposez d'un projet Logisim qui comprend déjà les circuits: Det\_Rotation, et Gestion\_position. Le premier circuit comprend une machine d'état, décrit en VHDL, pour détecter les transitions des signaux A et B avec génération des signaux impulsions Up et Down. Le second circuit comprend un compteur 10 bits permettant de simuler la position du chariot de 0x000 à 0x26F. Le compteur est configuré en mode saturé. Le switch SW5 allume le point décimal de l'afficheur 7 segments no3 (debug). Le projet comprend aussi une version de base du circuit lba\_user\_interface qui correspond au plan d'adressage fourni en annexe.

Vous devez modifier le circuit lba\_user\_interface afin d'y ajouter les fonctionnalités spécifiées ci-dessus.

Vous disposez d'un fichier C qui gère la sortie du programme (switch 4 du CPU). Le programme C comprend les initialisations nécessaires pour pouvoir utiliser l'interruption de la FPGA. Vous devez compléter le programme afin de réaliser les fonctions spécifiées ci-dessus.

Une **marche à suivre** vous est donnée dans le chapitre "Travail demandé" afin de réaliser ce laboratoire par étapes.

### Explication sur l'interruption connectée à la FPGA:

La ligne d'interruption de la FPGA est connectée sur le GPIO10\_1 (schéma: signal SP6\_GPIO18\_1) du microcontrôleur DM3730. Il s'agit de la GPIO no 18 connectée sur la *bank* 1 des entrées/sorties. Cette GPIO génère l'interruption M\_IRQ\_29. Celle-ci correspond à l'interruption générée par le module 1: GPIO1\_MPU\_IRQ.

L'interruption M\_IRQ\_29 utilisée dans ce labo est connectée sur le contrôleur INTCPS. Celui-ci génère une interruption standard IRQ sur le CPU. Il est nécessaire de quittance l'activation du IRQ via le registre INTCPS\_CONTROL.

La source de l'interruption dans le module 1 GPIO doit aussi être quittancée. C'est le registre GPIO\_IRQSTATUS1 du module GPIO1 qui doit être utilisé. Il faut écrire '1' dans le bit correspondant à l'interruption M\_IRQ\_29 (soit le bit 29 du registre).

### **Travail demandé :**

Nous allons réaliser ce projet par étapes afin de pouvoir maîtriser la complexité du projet et de réaliser des tests partiels. La recherche en cas de panne est ainsi plus simple, car il est plus aisé de déterminer la source du problème dans la partie modifiée.

Dans une première étape nous allons réaliser une version avec gestion par scrutation, sans interruption:

- 1) Compléter le circuit **lba\_user\_interface** du projet Logisim fourni afin d'ajouter les fonctionnalités suivantes:
  - Compléter l'interface afin de correspondre au plan d'adressage.
  - Détecter lorsque le chariot dépasse une butée (min ou max), mettre à jour deux flags: det\_min et det\_max
  - Prévoir une action pour désactiver les flags
- 2) Vérifier le fonctionnement de votre interface avec les fonctionnalités en le simulant dans Logisim.
- 3) Compléter le **programme C** fourni afin d'ajouter les fonctionnalités suivantes:
  - écrire les valeurs pour les butées min et max. Des constantes définiront ces valeurs dans le programme.
  - par scrutation effectuer les actions suivantes:
    - afficher la position courante du chariot et l'état des flags dans la console toujours à la même position dans la fenêtre (utiliser CR, soit \r).
    - lors de l'activation (flanc) d'un flag det\_min ou det\_max :
      - afficher un message indiquant l'activation d'un flag
      - allumer l'affichage 7 segments correspondant au flag comme indiqué dans les spécifications. Cela indiquera l'action manuelle à faire sur le bouton rotatif pour quitter la butée correspondante.
      - effacer le flag
        - dans une première étape, il est plus simple d'effacer le flag à l'aide du bouton SW3
    - Lorsque le chariot est à nouveau dans la zone de travail, effacer les affichages 7 segments.

- 4) Compiler votre projet Logisim et le programme C.  
Tester l'ensemble du projet avec une gestion par scrutation.
- 5) Faire valider le fonctionnement de votre solution par le professeur ou l'assistant

Dans la seconde étape, nous allons ajouter la gestion de l'interruption:

- 6) Modifier votre interface réalisée précédemment, pour générer une interruption lorsqu'un dépassement des butées est détectées (activation de `det_min` et `det_max`).
  - Afficher votre demander d'interruption sur une led de la FPGA
- 7) Vérifier le fonctionnement de l'interface par une simulation dans Logisim
- 8) Compléter le sous-programme d'interruption afin que l'état de la leds 0 du CPU soit inversé. L'objectif est de détecter le passage dans le sous-programme d'interruption.
- 9) Compiler votre projet Logisim et le programme C. Voici comment procéder pour le test:
  - programmer la FPGA et transférer votre programme C dans la carte REPTAR
  - lancer votre programme C sur la cible, puis quitter celui-ci en pressant sur SW4 du CPU. La routine d'interruption reste en mémoire
  - Activer l'interruption (vérifier que la led correspondante s'allume)
  - En utilisant les commandes `mw` et `md` de `uboot`, vérifier l'état des registres du contrôleur d'interruption. Quitancer dans le contrôleur l'interruption. Faite de même pour votre interface. La led correspondante doit s'éteindre.
  - Tester plusieurs fois la séquence ci-dessus.
- 10) Compléter le sous-programme d'interruption afin de réaliser la fonctionnalité demandée. Il est utile de laisser l'inversion de l'état de la leds 0 du CPU.
  - tester par scrutation afin savoir s'il s'agit d'un dépassement de la limite min ou max.
  - indiquer dans quelle butée a été dépassée par le chariot.
  - quittancer l'interruption dans votre interface
  - quittancer l'interruption `M_IRQ_29` du module `GPIO1`. Il faut utiliser le registre `GPIO_IRQSTATUS1`
  - quittancer l'interruption dans le contrôleur `INTCPS`
- 11) Compiler votre programme C. Tester le fonctionnement complet du système.
- 12) Faire valider le fonctionnement de votre solution par le professeur ou l'assistant

## Annexes

Le projet Logisim utilise la zone FPGA\_user qui utilise la zone d'adresse suivante:  
zone de 0x1900\_0000 à 0x193F\_FFFF.

Le projet Logisim fourni définit le plan d'adressage suivant:

Une zone d'adresse de 0x1900\_0000 à 0x1900\_000F. Les 7 premières adresses sont utilisées pour l'interfaçage d'éléments fourni dans le projet FPGA.

Adresse	Read		Write	
	D15	0	D15	0
0x1900 0000	Cst (0xA123, test)		not used	
0x1900 0002	[15..8] 0x00; [7..0] switch7..0		not used	
0x1900 0004	[15..8] 0x00; [7..0] Leds7..0		[15..8] not used; [7..0] Leds7..0	
0x1900 0006	[15..8] Aff_7seg2; [7..0] Aff_7seg1		[15..8] Aff_7seg2; [7..0] Aff_7seg1	
0x1900 0008	[15..10] 0x00; [9..0] Position		[15..1] not used; [0] Init	
0x1900 000A	[15..10] not used; [9..0] butee_min		[15..10] not used; [9..0] butee_min	
0x1900 000C	[15..10] not used; [9..0] butee_max		[15..10] not used; [9..0] butee_max	
0x1900 000E	not used		not used	
0x1900 0010	Libre		Libre	