

Cours 6-7 : Les interruptions

Architecture 2 : Ensimag

Motivation -1

- Pilote par scrutation

Utilisation du processeur à attendre

- L'attente peut être longue
 - exemple clavier
 - Disque dur (exemple temps d'accès moyen 8,5ms)
- Le processeur pourrait être utilisé autrement
 - Calcul
 - Faible consommation (instruction wait i386)

Lien : <http://www.ibm1130.net/functional/WAITinstr.html>

=> Pilote par interruption

Motivation -2

- Assurer le temps réel:
 - Gérer un évènement quand il arrive (signalé par un périphérique).

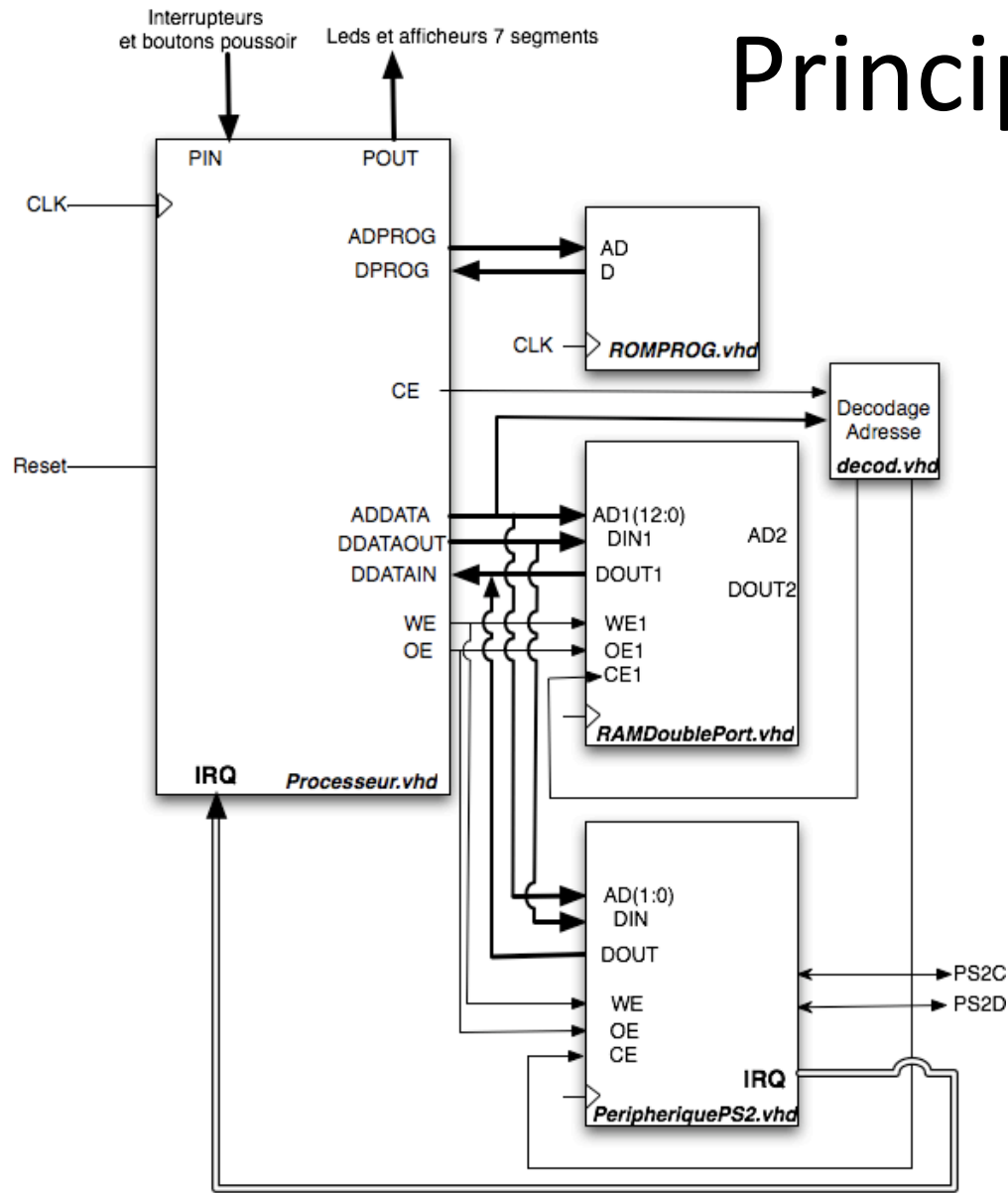
Exemple :

- Périphérique TIMER (intervalles réguliers programmables):

- Déplacement d'un objet (dans un jeu) à une vitesse indépendante de la vitesse du processeur

- Partage des tâches (Cf. cours système 2A)

Principe



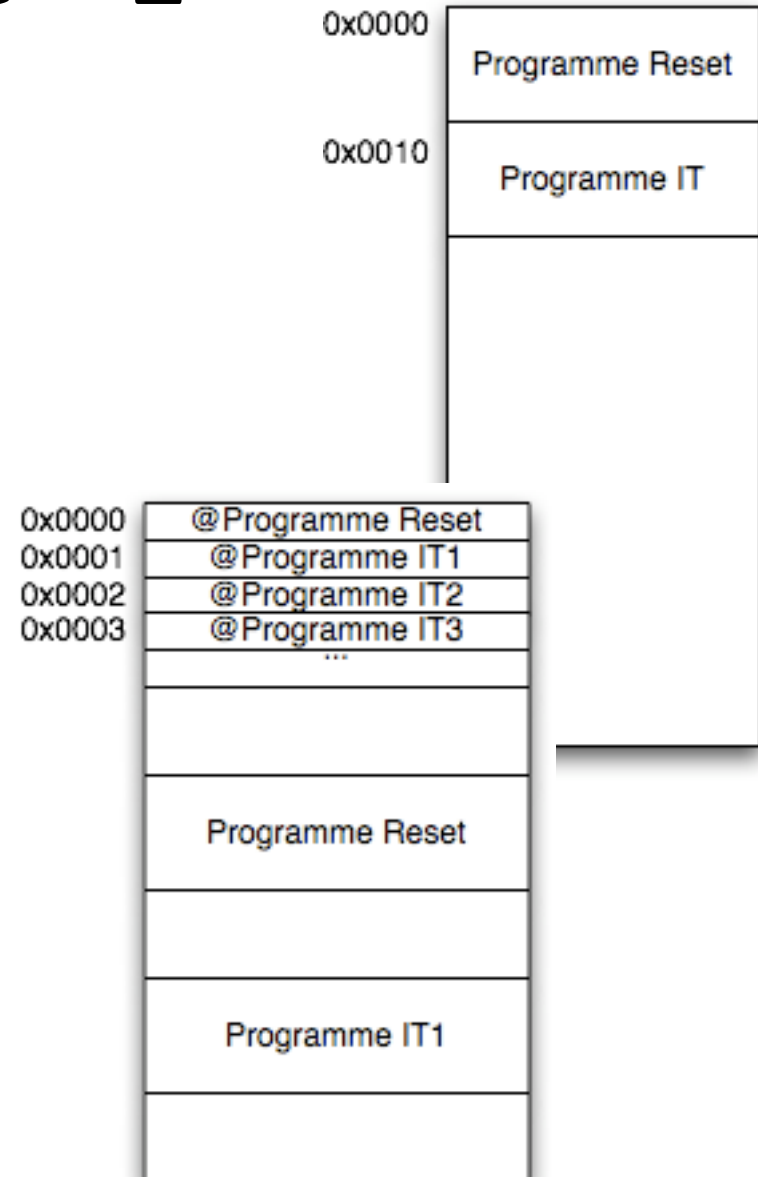
- IRQ arrive :
 - Processeur interrompt le programme en cours
 - Exécute le programme associé à l'interruption
 - Reprend le programme qui a été interrompu

En pratique -1

- Processeur interrompt le programme en cours
 - Sauvegarder PC (comme pour un appel à un sous programme)
 - Comment?
 - » Avec la pile (i386)
 - » Registre dédié (RISC) : Exemple R6 (par convention et en dur)
 - Sauvegarder ce qui risque d'être modifié sans qu'on puisse le faire dans le programme d'interruption
 - Quoi?
 - » Carry
 - Ne pas être interrompu à nouveau
 - Mettre en place un masquage des interruptions
 - » Ajout d'instructions pour masquer et démasquer les interruptions : MSKIT et UMSKIT

En pratique – 2

- Exécute le programme associé à l'interruption
 - PC va prendre la valeur du programme d'interruption
 - **Valeur fixe : exemple 0x0010**
 - Valeur fonction d'une table des vecteurs d'interruptions (table à une adresse donnée)
 - Exemple : table à l'adresse 0x0000



En pratique 3

- Exécute le programme associé à l'interruption (suite)
 - Tout ce qui n'est pas sauvegardé automatiquement et qui risque d'être modifié doit
 - être sauvegardé (à l'entrée du programme d'IT)
 - Être restauré (à la sortie du programme d'IT)

En pratique 4

- Reprend le programme qui a été interrompu
 - Reprend le programme en ré-autorisant les interruptions
 - Ajout d'une nouvelle instruction (retour d'interruption)

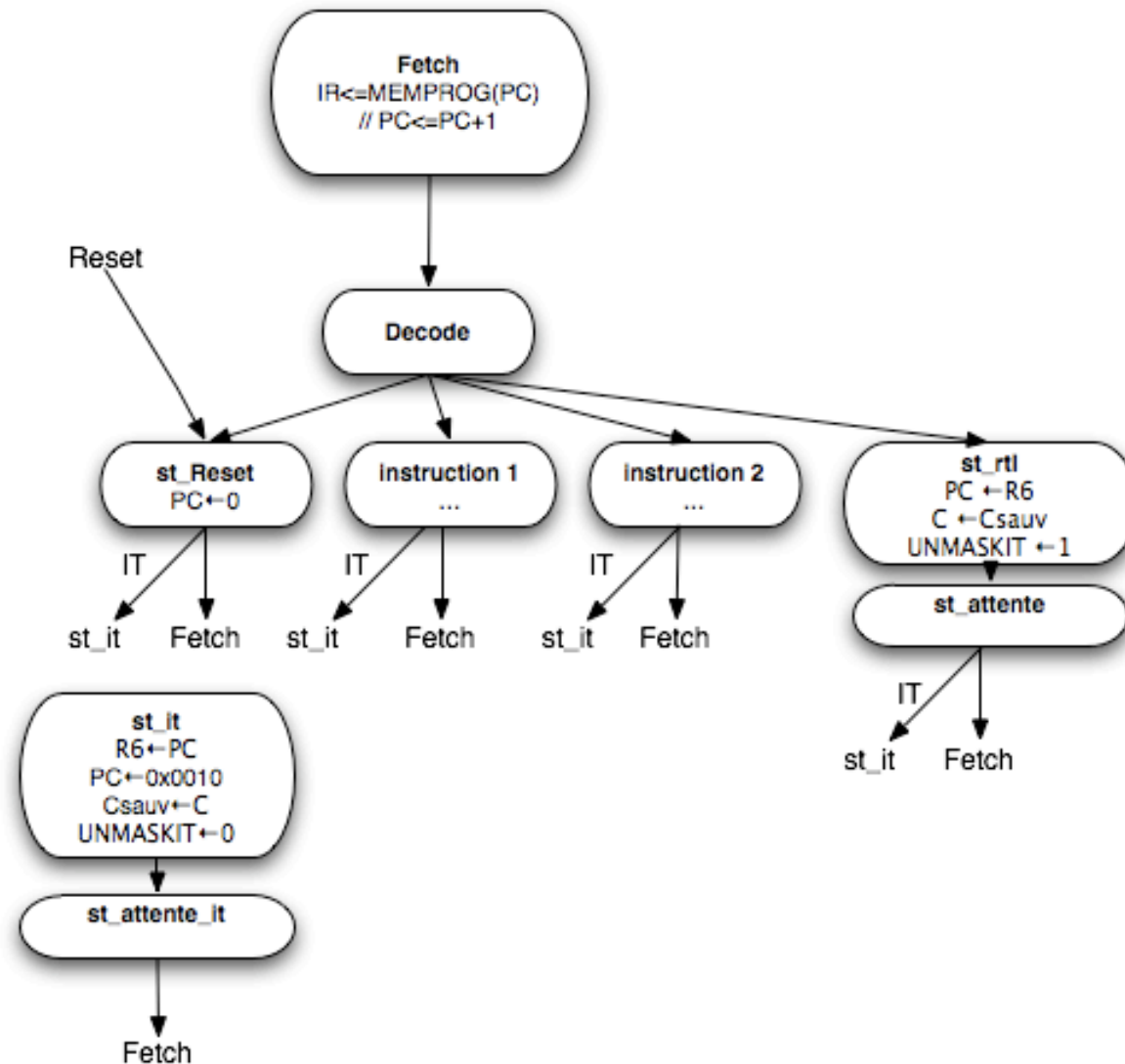
Exercice : Comment fait on un retour de sous programme avec le jeu d'instruction ?

Comment pourrait on faire un retour d'interruption ?

Ecriture du driver de clavier

- 0x0010:
 - Sauvegarde des registres susceptibles d'être modifiés
 - Appel sous programme progIT
 - Restitue les registres susceptibles d'être modifiés
 - RTI
- progIT (stocké en mémoire à un endroit quelconque)
 - Tant qu'un caractère est présent
 - **Car** =donnée lue dans la fifo
 - **Traite(Car);**
 - Fin du sous programme progIT

Réalisation : Coté FSM (Partie Contrôle)

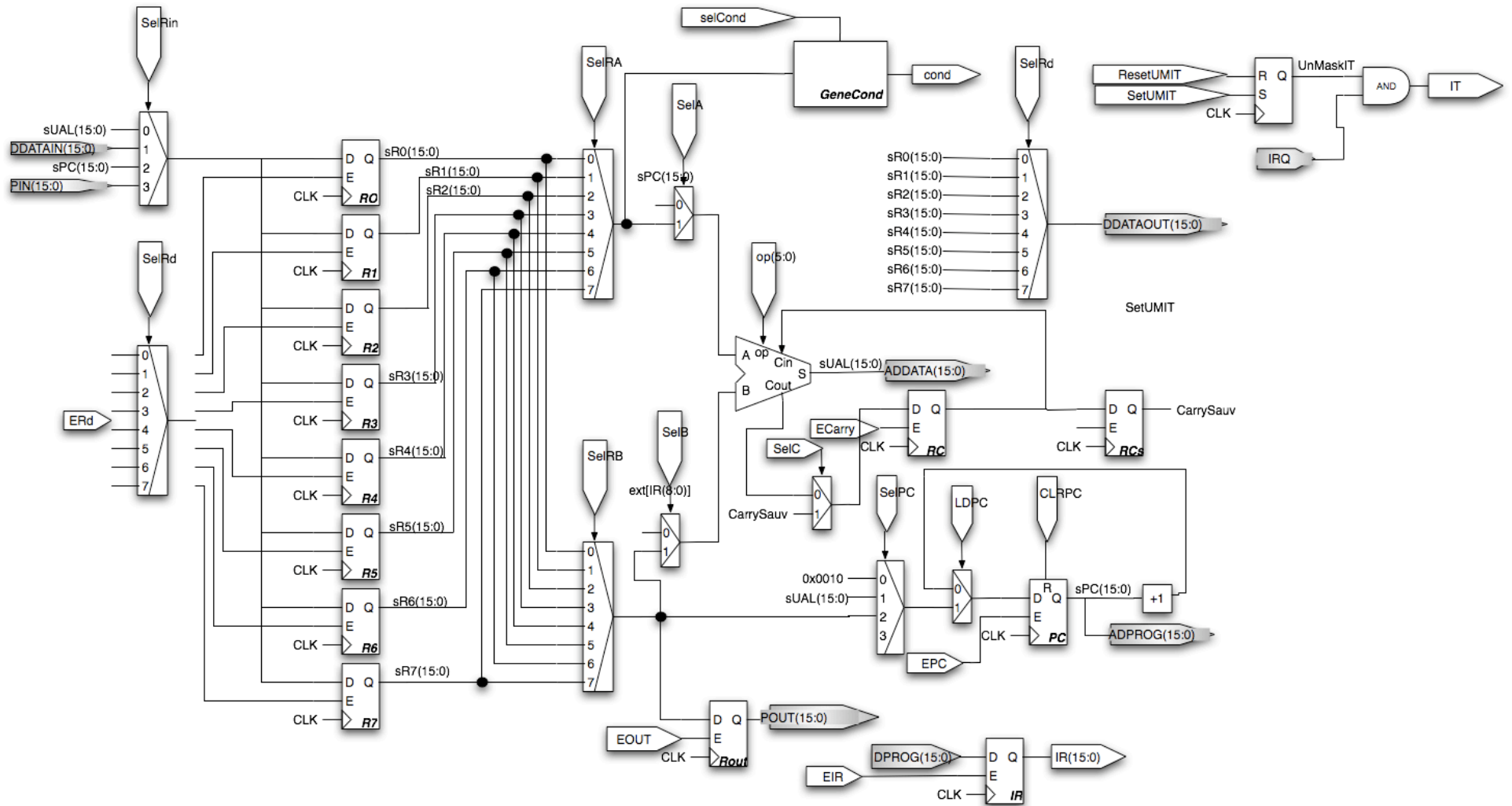


Réalisation : coté DataPath (PO)

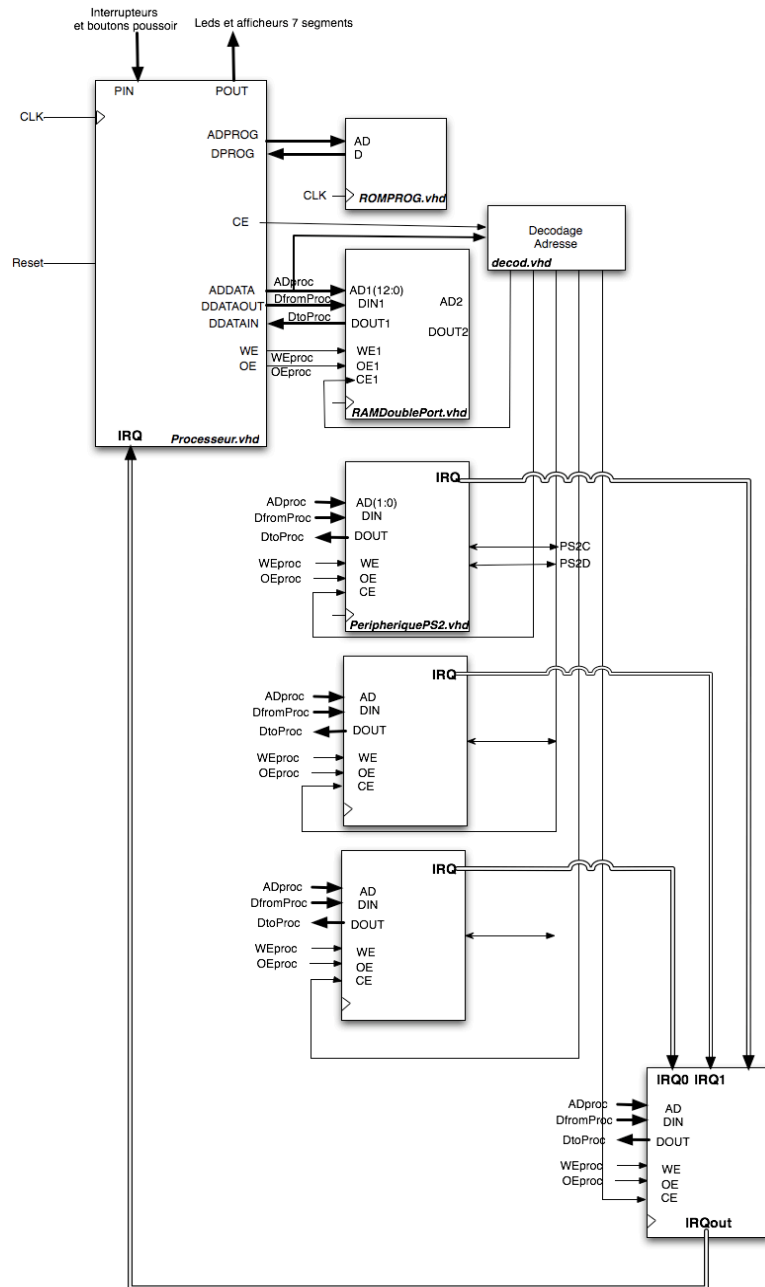
Exercice: Donner les modifications à apporter à la PO pour permettre

- 0x0010 -> PC
- PC->R6
- C->Csauv
- Csauv->C
- Gestion de UNMASKIT
- Génération du compte rendu IT

Correction



Plusieurs sources d'interruption-1



- Plusieurs sources :
Composant intermédiaire entre les périphériques et le processeur

Plusieurs sources d'interruption-2

- Gestionnaire d'interruption

- Périphérique à part entière

- Registre de contrôle

- Configuration**

- Masquer les interruptions de telle ou telle source
 - Sélectionner la sensibilité des interruptions

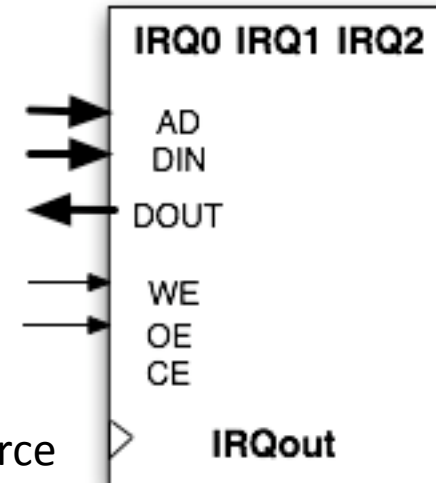
- Action**

- Acquitter le traitement d'une interruption

- Registre d'état

- Consulter la configuration
 - Une interruption est à traiter sur telle ou telle source

- Autant d'entrée que de périphériques (priorité à définir)



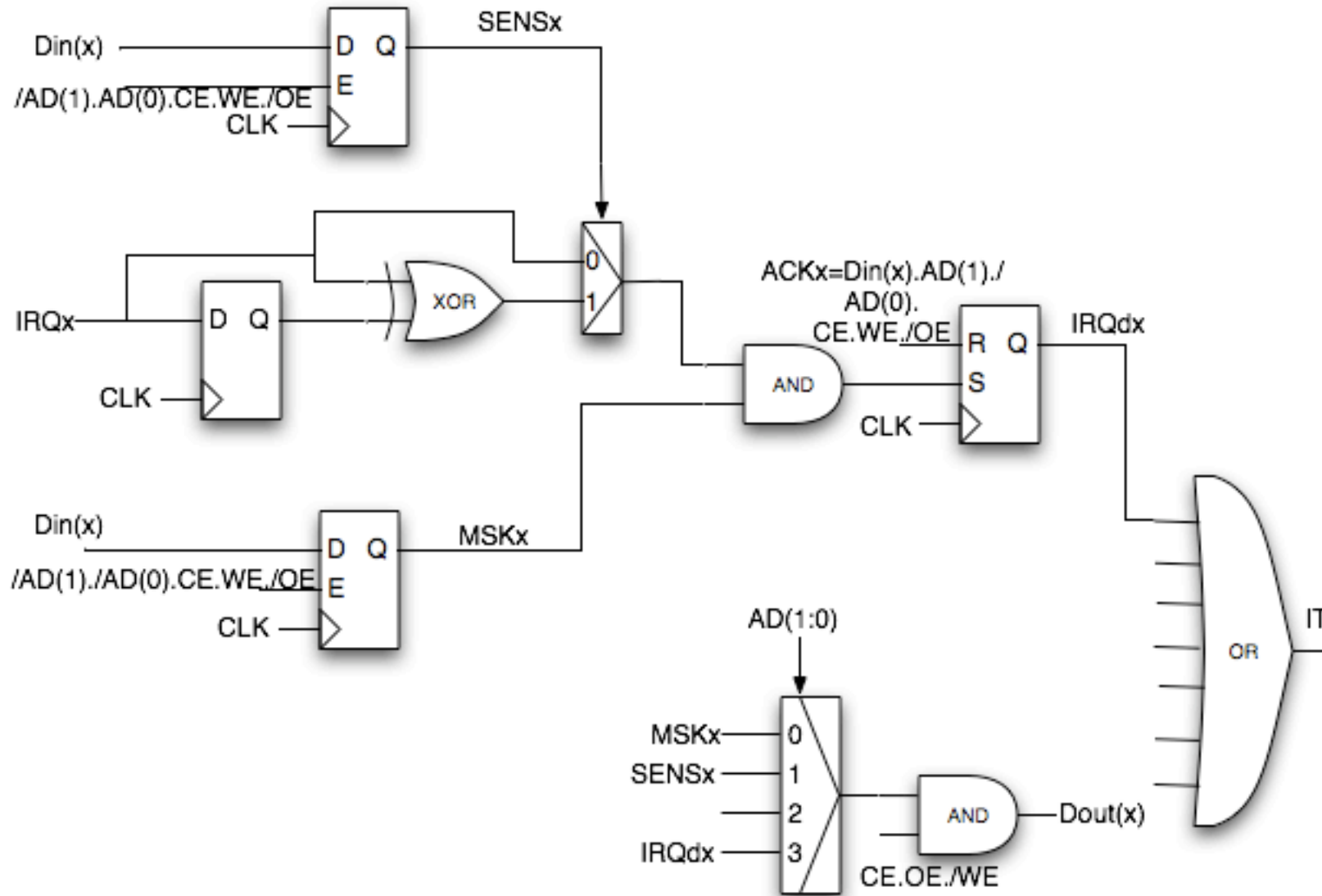
Exemple de mapping mémoire

- Adresse de base+0 (R/W)
 - Masquage des interruptions (contrôle)
 - Bit 0 : IRQ0
 - Bit 1 : IRQ1
 - ...
- Adresse de base+1 (R/W)
 - Configuration de la sensibilité des interruptions
 - Bitx:
 - 0: IRQx sensible sur niveau haut
 - 1: IRQx sensible sur changement (front)
- Adresse de base +2 (W)
 - Acquiesement des IRQ
 - Bitx : acquiesement de l'IRQx
- Adresse de base +3 (R)
 - Présence d'une interruption à traiter
 - Bitx à 1 IRQx est à traiter.

Construction du gestionnaire d'IT

- Exercice
 - Comment détecter le changement d'une IRQ?
 - Comment mettre en œuvre le masquage des IRQ?
 - Comment mettre en œuvre l'acquittement des IRQ?
 - Donner le schéma correspondant au gestionnaire d'interruption.

Correction (pour un bit)



Utilisation

- 0x0010 :
 - Sauvegarde des registres
 - Appel programme IT
 - RTI
- Programme IT
 - Pour chaque IRQ à traiter
 - Appel le programme de gestion du périphérique correspondant à l'IRQ à traiter
- Programme de gestion d'un périphérique
 - Traite la demande d'interruption
 - Acquitte l'interruption au niveau du gestionnaire d'interruption

Exercice :

- Comment détermine-t-on qu'une IRQ est à traiter?
- Comment acquitter la demande d'une IRQ?
- Comment peut masquer une IRQ?
- Est il nécessaire de masquer l'IRQ en cours de traitement?