

Les interruptions dans l'environnement Procsim

Nous voulons élargir les possibilités de la machine simulée dans Procsim (voir TP précédent) afin qu'elle puisse prendre en compte des interruptions.

Rappel : Le mécanisme d'interruption permet d'effectuer un traitement prédéfini en réponse à un événement. L'occurrence de l'événement interrompt l'exécution du programme en cours en forçant le branchement à une routine particulière appelée "traitant d'interruption". L'exécution du programme est reprise une fois le traitement de l'interruption terminé.

L'origine des interruptions peut être due à l'environnement de la machine (requête provenant d'un périphérique). Il peut s'agir aussi d'une erreur liée au déroulement du programme en cours, par exemple une division par zéro, et on parle alors plutôt de déroutement. Le mécanisme d'interruption peut également être activé explicitement par le programmeur par une instruction spéciale (dite d'appel au superviseur).

Une interruption affecte le déroulement du programme de l'utilisateur, mais pas celui de l'algorithme d'interprétation des instructions. La prise en compte d'une interruption est donc retardée jusqu'à la fin de l'instruction en cours d'exécution, autrement dit jusqu'à la recherche du prochain code-opération (sauf pour certains déroutements, dont la nature ne permet pas de mener l'instruction à son terme).

On peut résumer le traitement d'une IT par un processeur en 3 étapes principales:

1- Prise en compte de l'IT.

- 1.1- Détection par le processeur de la demande d'interruption.
- 1.2- Sauvegarde de l'état du processeur. Il faut sauvegarder certaines informations (appelé souvent contexte) qui permettront de reprendre l'exécution du programme interrompu sans problème (au minimum le compteur programme et le registre d'état).

1.3- Déterminer l'adresse du programme traitant l'interruption, affecter cette adresse au compteur programme et recommencer à interpréter les instructions (qui seront maintenant celles du programme traitant l'interruption).

2- Traitement des instructions du programme traitant l'interruption

3- Retour dans le programme interrompu

Lors de l'exécution de l'instruction de retour de programme d'interruption (RTI), rétablir le contexte du processeur sauvegardé précédemment et reprendre l'exécution du programme interrompu.

Différents mécanismes peuvent être mis en place pour chacune de ces étapes (voir cours et différents processeurs existants).

Voici ce qui pourrait être fait dans le cas de la machine MACHIN.

I- Interruption venant de l'extérieur

Nous nous intéressons tout d'abord à une interruption venant de l'extérieur du processeur.

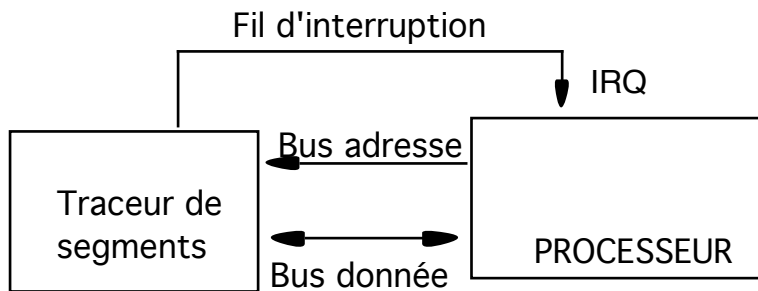
Supposons que l'on veuille mettre le co-processeur "traceur de segment" à côté du processeur. Il faut que le traceur de segment puisse signaler au processeur la fin du tracé d'un segment alors que celui-ci est en train d'exécuter un programme. Il doit donc interrompre le déroulement de ce programme.

Un autre exemple d'interruption externe sont les erreurs d'adressage provoquées par le décodage d'adresse (voir TD décodage). Un traitant particulier affichera alors par exemple à l'écran un message d'erreur (« bus error » bien connu) et rendra la main au système.

Si nous reprenons étape par étape, voici les spécifications que vous devez prendre en compte pour modifier la partie contrôle de la machine étudiée (nous ne nous intéressons pas pour l'instant au contenu du programme traitant l'interruption) :

1- Prise en compte de l'IT.

1.1- Le périphérique demandera une interruption par un fil spécialisé (voir figure ci-dessous). Le processeur sait s'il y a une demande d'IT grâce à la microinstruction **jq** : saut si interruption extérieure. Au niveau du simulateur une IT extérieure est activée à l'aide du bouton **IRQ**. Cette demande reste activée jusqu'au prochain clic sur ce bouton.



Dans la réalité, la demande d'IT restera activé par le périphérique tant que le traitant d'IT ne l'aura pas « acquité ».

1.2- Sauvegarde du contexte courant.

Le minimum de sauvegarde sera fait automatiquement (par le processeur) sur la pile : Compteur programme et registre d'état (F). Si d'autres registres sont utilisés dans le programme d'IT, ils devront être sauvegardés par le programmeur.

1.3- Déterminer l'adresse du programme d'IT.

Cette adresse est stockée en RAM par convention à l'adresse **FD**.

2 - Traitant d'IT

On imaginera un petit programme d'interruption que l'on mettra à l'adresse voulue.

Dans le cas d'une IT externe, le traitant d'IT devra s'occuper d'acquiescer le demandeur d'IT afin que celui-ci arrête de faire sa demande (sur le fil d'IT).

3- Retour au programme interrompu.

Il faudra rajouter au jeu d'instructions déjà définies une instruction particulière, RTI : Retour de Traitement d'Interruption (appelée quelquefois RTE : Retour de Traitement d'Exception). Cette instruction devra donc rétablir le contexte précédemment sauvegardé, c'est-à-dire restaurer les anciennes valeurs de PC et F.

4- Masquage d'IT

Nous rajoutons à ces 4 étapes de base un mécanisme de masquage d'IT. Ce mécanisme permet ou non la prise en compte d'une IT. Nous utilisons pour cela le bit 3 (appelé I) de F: s'il est à 1, l'IT demandée n'est pas prise en compte.

Ce bit permet donc au programmeur de rendre une partie de programme interruptible ou non. Pour cela il utilise les instructions CLI et SETI qui permettent la mise à 0 et à 1 du bit.

De plus, il est mis automatiquement à 1 pendant l'exécution d'un programme d'IT et remis à son ancienne valeur à l'exécution du RTI. Cela permet d'inhiber la demande de l'IT qui est toujours présente une fois que l'on est dans le traitant d'IT.

II - Interruption venant de l'intérieur du processeur

Nous nous intéressons maintenant à une IT venant de l'intérieur du processeur. Ces ITs, souvent appelées déroutement, sont déclenchées dans les cas où il y a des problèmes majeurs à l'intérieur du processeur:

- Division par 0

- Débordement

- Instruction interdite

- Instructions spéciales (appelé SWI): qui a le même effet qu'une requête d'interruption (masquage, sauvegarde contexte...).. Elle permet de débogger aisément des traitants d'IT si elle déclenche le même traitant d'IT qu'une IT externe. Elle peut permettre de réaliser des macro-instructions dans le cas d'un traitant d'IT différent.

- ... etc

* Cas de l'instruction illégale:

Par exemple, la détection par micro-programme d'un code ne correspondant à aucune instruction existante peut provoquer un traitement d'exception. Ce type d'interruption ne peut pas être masqué. L'adresse du traitant d'IT en cas d'instruction illégale est trouvée par convention à l'adresse **FB**.

Les ITs externes et internes seront masquées automatiquement (par le processeur) lors de la prise en compte de ce type d'IT.

QUESTION:

Modifier le micro-programme du processeur jouet afin que :

- Le processeur prenne en compte les interruptions extérieures avec les hypothèses données ci-dessus.
- Une instruction SWI soit possible. L'adresse du traitant d'IT se trouvant à l'adresse **FC**.
- Une interruption interne soit déclenchée en cas de code instruction inexistant
- Le processeur traite les instructions CLI, SETI et RTI.

Ecrire un petit traitant d'IT pour les trois cas.

Tester le processeur.