

QUESTIONS sur le TP6 / Entrées_Sorties et Interruptions

Q1) Il existe une très grande variété de périphériques d'entrée/sortie susceptibles de générer des interruptions. Quelles sont les caractéristiques communes à toutes les interruptions ?

Une interruption matérielle est toujours générée par un périphérique pour signaler un événement attendu par l'OS. Cet événement est le plus souvent la fin d'une opération d'entrée/sortie démarrée par l'OS (transfert de donnée entrant ou sortant entre la mémoire et le monde extérieur : écran, clavier, disque, réseau, etc). Ce peut-être aussi le TICK indiquant qu'un changement de contexte doit être opéré sur le processeur cible. L'événement est donc attendu, mais sa date est généralement imprédictible, car cette date dépend de la durée (extrêmement variable) de l'opération d'entrée/sortie. Puisque l'événement est attendu, l'OS a seulement besoin de déterminer quel est le périphérique source de l'interruption pour exécuter la bonne routine d'interruption.

Q2) Qu'est-ce qui différencie une interruption matérielle d'une exception matérielle ?

Ces deux types d'événements sont totalement différents:

- Une interruption matérielle est asynchrone (on ne peut pas savoir quand elle va se produire) et elle n'a que très peu d'effet sur le programme A en cours d'exécution, car le programme B qui a demandé l'opération d'entrée/sortie, et qui est la cause de l'interruption, n'est généralement pas en cours d'exécution quand l'opération d'entrée/sortie se termine. Le programme A en cours d'exécution va seulement perdre quelques centaines de cycles, qui vont donc le retarder d'autant, mais il reprendra son exécution au point où il en était, sans autre modifications quand l'ISR se termine.
- Une exception matérielle se produit quand une instruction particulière X d'un programme A essaie de faire quelque chose d'illégal ou d'impossible, généralement à cause d'une erreur de programmation (division par zéro, erreur d'adressage, etc.) C'est donc un événement synchrone et reproductible : il se reproduira à la même instruction X si on essaie de re-exécuter le programme A sans le modifier. Et c'est presque toujours un événement mortel pour le programme fautif, qui est généralement tué par l'OS, avec message d'erreur pour aider au debug.

Q3) Que se passe-t-il lorsqu'une seconde interruption IRQ_2, destinée à un coeur, se produit alors que le coeur est déjà en train d'exécuter le code de l'ISR associée à une première interruption IRQ_1?

Le gestionnaire d'interruption du GIET n'est pas re-entrant. Lorsque le processeur se branche au GIET parce qu'une interruption IRQ_1, non masquée, a été détectée, le bit d'autorisation des interruptions du register SR (bit IE) est systématiquement mis à 0, pour que le GIET commence à s'exécuter avec les interruptions masquées. Mais le gestionnaire d'interruption ne remet pas à ce bit à 1, et donc le traitement de l'interruption IRQ_1, y compris l'exécution de l'ISR associée à IRQ_1 ne peuvent être interrompus par la seconde interruption IRQ_2. Par conséquent l'interruption IRQ_2 ne sera prise en compte que lorsque le gestionnaire d'interruption redonne la main au programme utilisateur interrompu par IRQ_1. En effet l'instruction "eret" remet à 1 le bit IM, car un programme en mode utilisateur doit toujours être interruptible. En d'autre termes, le GIET traite les interruptions l'une après l'autre, et le traitement d'une interruption n'est pas interruptible. Certains systèmes d'exploitation utilisent des gestionnaires d'interruptions re-entrant, c'est à dire que le gestionnaire d'interruption est lui-même interruptible. Mais cela complique significativement l'écriture du Gestionnaire d'Interruption, car il faut alors utiliser une technique de pile pour sauvegarder les contextes.

Q4) Le composant ICU ne possède que 32 entrées IRQ_IN[i]. Que se passe-t-il si l'architecture contient beaucoup de périphériques contenant beaucoup de canaux, et et si on a besoin de plus de 32 entrées ?

Il existe différentes solutions pour résoudre ce problème sans changer profondément l'architecture:

- on peut construire une structure d'arbre à deux niveaux ou un premier composant ICU_1 prend pour N entrées, les N sorties de N composants ICU_2 de 2e niveau qui eux mêmes prennent en entrée les interruptions des périphériques. Cela complexifie le gestionnaire d'interruption logiciel qui devra interroger successivement deux composants ICU pour déterminer le périphérique source.
- on peut connecter plusieurs lignes d'interruption sur une seule entrée de l'ICU, en réalisant un "ou câblé". Cela complexifie encore plus le gestionnaire d'interruption logiciel qui devra d'abord interroger le composant ICU,

puis interroger directement les différents périphériques partageant la même entrée pour déterminer le périphérique source.

Q5) Comment est réalisée la fonction de routage des interruptions vers les coeurs dans les PC multi-coeurs actuels, puisque les contrôleurs de périphériques ne sont pas directement connectés sur le bus interne, mais sont connectés sur un bus externe.

L'architecture multi-coeurs présentée dans l'U.E. MULTI a été en effet un peu simplifiée. Dans les PC multi-coeurs actuels, seuls les coeurs (avec leurs caches L1) et le (ou les) cache(s) L2 sont connectés sur le bus interne et contenus dans la puce processeur. Un composant matériel spécialisé, appelé *north-bridge* fait le pont entre le bus interne du processeur, et les barrettes de RAM externe placées sur la carte mère. Un autre composant matériel spécialisé appelé *south-bridge* fait le pont entre le bus interne du processeur et le bus externe sur lequel sont connectés les contrôleurs de périphériques (bus PCI express ou HyperTransport?). Ces périphériques ont souvent une capacité DMA (voir TP7 et TP8), et donc beaucoup d'informations passent à travers le south-bridge, dans les deux sens. Cette frontière entre les composants internes et composants externes crée un problème pour connecter les lignes d'interruptions sortant des contrôleurs de périphériques (externes) aux coeurs (internes). Pour cela la fonction de routage des interruptions réalisée par le composant ICU (nommé PIC dans les PC, comme Programmable Interrupt Controller) est en pratique réalisée par deux composants matériels:

- le composant IOPIC est connecté sur le bus externe et c'est lui qui reçoit les lignes d'interruptions provenant des périphériques. C'est à la fois un maître et une cible sur le bus externe.
- le composant LAPIC est un composant interne attaché à chaque coeur. C'est une cible sur le bus interne.

La communication entre l'IOPIC externe et le LAPIC (interne) se fait travers le south-bridge: le IOPIC (maitre) qui reçoit une interruption d'un périphérique, transforme cet événement en une transaction d'écriture vers le LAPIC attaché au coeur qui doit recevoir cette interruption. Cette transaction d'écriture traverse le bus externe, traverse le south-bridge, traverse le bus interne, et atteint le LAPIC (cible), qui lui-même génère une interruption matérielle vers le coeur cible.