QUESTIONS sur le TP7 / coprocesseur DMA

Q1) Sous une sytème d'exploitation généraliste comme LINUX on WINDOWS, comment un programme utilisateur peut-il demander à utiliser le coprocesseur DMA ?

- Le terminal écran/clavier TTY, est mis à la disposition d'un programme utilisateur à travers une abstraction spécifique définie par des appels système tels que printf() et getc(). Le contrôleur de disque IOC est mis à la disposition des programmes utilisateurs à travers une l'abstraction des fichiers, qui définit d'autres appels sytèmes tels que open(), rea() , ou write(), close(). Le contrôleur réseau (non analysé dans MULTI) est mis à la disposition des programmes utilisateurs à travers l'abstraction des sockets, qui définit encore d(autres appels systèmes tels que send() et receive(). L'existence de ces périphériques d'entrée/sortie permettant la communication avec le monde extérieur est donc connue des programmes qui peuvent donc y accéder à travers des appels système dédiés.
- Le coprocesseur DMA est dans une situation différente: Ce n'est pas un périphérique d'entrée/sortie et mais un accélérateur matériel de copie de mémoire à mémoire, qui peut être présent ou non dans l'architecture matérielle. Il n'est donc utilisé par l'OS que s'il existe, et pour accélérer certains transferts de sonnées comme l'affichage d'une image sur l'écran graphique de l'architecture. C'est donc en général une optimisation non contrôlée par le programme utilisateur.

Q2) Pourquoi l'interruption DMA est-elle débrayable par logiciel en utilisant le registre DMA_IRQ_ENABLE ?

La plupart des périphériques sont conçus pour être utilisés par différents OS, qui peuvent avoir des politiques différentes pour la signalisation de fin d'une opération d'entrée/sortie. La technique des interruptions, est efficace quand l'OS supporte le fonctionnement multi-tâches sur un seul processeur, mais la technique de scrutation directe du registre status par l'OS peut être très efficace dans un contexte de sytème embarqué mono-tâche. C'est pourquoi le composant matériel DMA, comme les autres, ne rend pas obligatoire l'interruption pour signaler la fin de transfert.

Q3) Comment le coprocesseur DMA peut-il être partagé par plusieurs tâches dans une architecture multi-coeurs ?

Toute ressource partagée doit être protégée en accès exclusif, comme c'est le cas pour le contrôleur IOC. Le coprocesseur DMA ne fait pas exception, etl doit donc être protégé par un verrou. SI on utilise un coprocesseur DMA multi-canaux, on peut associer un canal privé à chaque coeur. Mais si chaque coeur peut exécuter plusieurs applications (par multiplexage temporel et commutation de contexte), il faut un verrou par canal, pour garantir l'exclusivité entre les différentes applications s'exécutant sur le même coeur

Q4) Pourquoi dit-on que l'appel système fb_write() est non-bloquant, alors que cette fonction peut se bloquer s'il faut attendre la fin d'un autre transfert déjà en cours ?

Dire qu'une fonction est non-bloquante ne signifie pas qu'elle ne peut pas se bloquer. Cela signifie que la fonction peut rendre la main au programme appelant avant que le travail demandé (c'est à dire la copie d'un buffet utilisateur source vers le frame bouffer destination) n'est pas encore réalisé. On peut faire la même réponse pour l'appel système ioc_write().

Last modified on 9 Jun 2020, 12:07:14