MASTER SESI

« Architecture des Systèmes Multi-Processeurs intégrés sur Puce»

Cours de A. GREINER

(Examen de Novembre 2009)

A) Protocole VCI (2 points)

A1) Quel est le nombre de mots VCI du paquet commande et du paquet réponse dans le cas d'une transaction d'écriture. Même question dans le cas d'une transaction de lecture.
Quel est le champs VCI utilisé pas le réseau d'interconnexion pour router le paquets commande ? Quel est le champs VCI utilisé pour router le paquet reponse ?

A2) A quoi sert le paquet réponse dans le cas d'une transaction d'écriture ?

A3) Quel mécanisme permet à un initiateur VCI de démarrer plusieurs transactions sans attendre d'avoir reçu la réponse N avant d'envoyer la commande N+1 ?

B) Micro-réseaux intégrés sur puce (2 points)

B1) Quelle est la principale limitation d'un mécanisme de communication de type bus partagé? Quel est le principal service fourni par les technologies de micro-réseau intégré sur puce ?

B2) Pourquoi dans les architectures à espace d'adressage partagé (supportant des transactions de type commande/réponse) utilise-t-on généralement des ressources matérielles distinctes (en particulier des tampons de stockage séparés) pour le réseau des commandes et pour le réseau des réponses ?

C) Mémoires caches (4 points)

C1) On considère un cache instruction de 1er niveau (cache L1) d'une capacité de huit Koctets. Ce cache est à correspondance directe, et la longueur d'une ligne de cache est de 32 octets. L'adresse a une largeur de 32 bits.
Rappelez la définition d'un cache à correspondance directe.
Le contrôleur du cache décompose cette adresse en trois champs : le champs X permet de désigner un octet dans la ligne de cache. Le champs Y permet de désigner le numéro de famille. Le champs Z est l'étiquette qui permet d'identifier une ligne de cache particulière au sein d'une même famille. Quelles sont les largeurs (en nombre de bits) des champs X, Y et Z ?

C2) On considère maintenant un cache de données de 1er niveau. Expliquez la différence entre une stratégie d'écriture WRITE_THROUGH et une stratégie d'écriture WRITE-BACK. Rappelez le principe du tampon d'écritures postées. Quelle est son utilité ?

C3) Quelle est la différence entre un cache bloquant et un cache non bloquant. Avec quels types de processeur un a-t-on besoin d'utiliser des caches non bloquants ?

C4) Rappelez le principe général du mécanisme de Snoop, permettant de garantir la cohérence entre les caches et la mémoire, sans intervention du logiciel. Pourquoi l'utilisation d'un micro-réseau complique-t-il le problème de la cohérence lorsque cette cohérence doit être assurée par le matériel ? Quelle est la solution ?

D) Instructions atomiques (2 points)

D1) Qu'est-ce-qu'une instruction atomique de type *test-and-set* ? Quelle est leur utilité du point de vue des applications logicielles ? Expliquez pourquoi l'implantation de ces instructions pose un problème particulier dans le cas des architectures utilisant un micro-réseau.

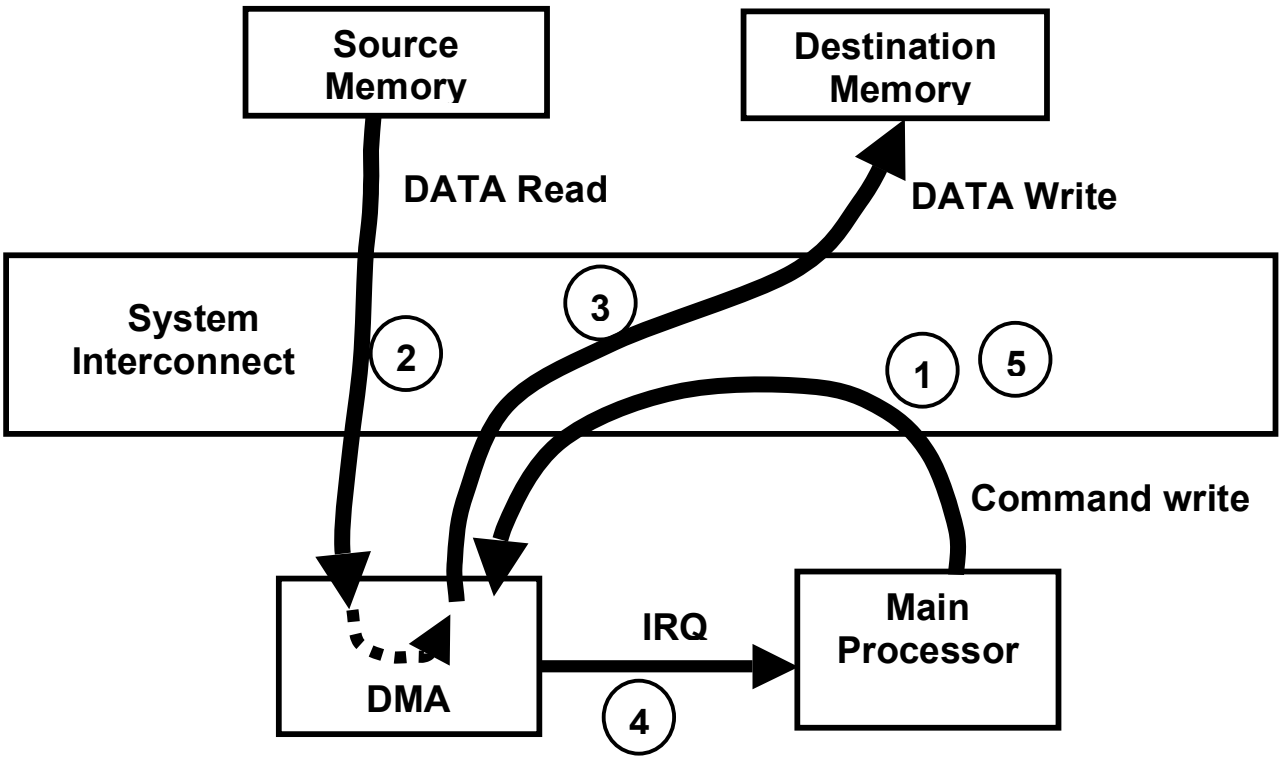
D2) Expliquez le principe du mécanisme LL / SC (Linked load / Store Conditionnal), en donnant un exemple de code en langage d'assemblage. Donnez un exemple de réalisation matérielle de ce mécanisme du côté du contrôleur mémoire.

E) Conception d'un contrôleur DMA à interface VCI (10 points)

On se propose d'étudier l'architecture interne d'un coprocesseur matériel à interface VCI capable de réaliser des transferts de données entre deux zones de la mémoire (par exemple pour déplacer une image entre la mémoire principale et la mémoire d'affichage). On appelle DMA (Direct memory Access) cette capacité d'un périphérique à adresser directement la mémoire.
Pour effectuer un tel transfert de données, le contrôleur DMA doit être configuré par le système d'explmoitation qui lui indique

- l'adresse du tampon mémoire source
- l'adresse du tampon mémoire destination
- le nombre N d'octets à transférer.

Le contrôleur DMA se comporte donc à la fois comme un initiateur (puisqu'il peut adresser la mémoire, en lecture ou en écriture), et comme une cible, puisqu'il doit recevoir des ordres du processeur principal.



Le scénario d'un transfert DMA est le suivant :

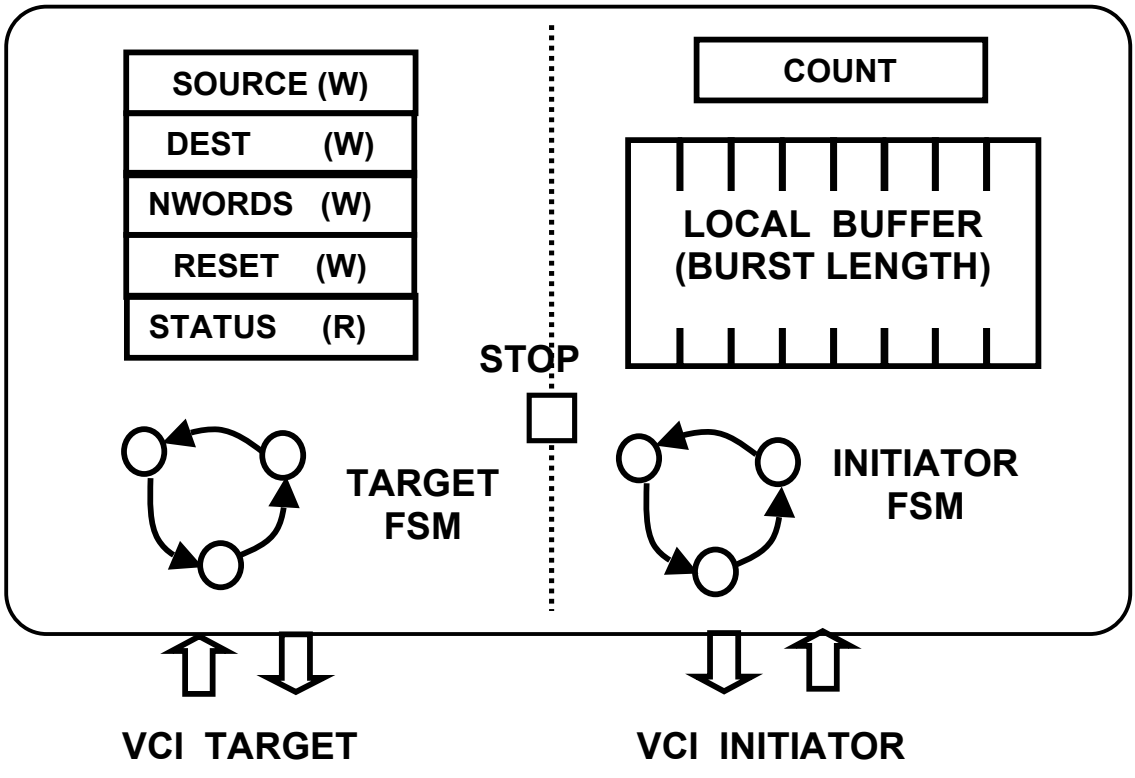
1. Le processeur principal accède au contrôleur DMA (considéré comme une cible), et écrit une série de commandes dans des registres adressables.
2. Le contrôleur DMA lit une rafale de données dans le tampon mémoire source, et les charge dans une petite mémoire locale de capacité L mots de 32 bits.
3. Le contrôleur DMA écrit dans le tampon mémoire destination les données qui ont été stockées dans la mémoire locale.
4. Les étapes 2 et 3 sont répétées jusqu'au transfert complet des données, et le contrôleur DMA signale la fin du transfert au processeur principal, en activant une interruption.
5. Le processeur principal acquitte l'interruption en écrivant dans le registre RESET du contrôleur DMA.

La longueur L d'une rafale (c'est à dire le nombre de mots de 32 bits), en lecture comme en écriture, est donc définie par la capacité de la mémoire locale contenue dans le contrôleur DMA. Pour simplifier l(exercice, on suppose que la capacité N des tampons source et destination est multiple de 4*L, et les adresses de base de ces tampons sont multiples de 4.
Dans le contrôleur DMA, un compteur circulaire COUNT permet de mémoriser un nombre de mots compris entre 0 et (L-1), et la bascule STOP permet de démarrer ou d'interrompre un transfert.
En tant que cible, le contrôleur DMA contient 5 registres 32 bits *mappés* en mémoire, qui peuvent être accédés soit en lecture, soit en écriture :

Address [3:0]	FONCTION	NOM	MODE
00 * *	adresse tampon source	SOURCE	Write
01 * *	adresse tampon destination	DEST	Write
10 * *	nombre de mots à transférer	NWORDS	Write
11 * *	acquittement interruption et initialisation	RESET	Write
* * * *	Etat du controleur	STATUS	Read

Le processeur principal commence par écrire les adresses dans les registres SOURCE et DEST, et c'est l'écriture d'une nouvelle valeur dans le registre NWORDS qui déclenche la procédure de transfert effectif des données.
L'écriture dans le registre RESET force la re-initialisation de la partie « initiateur » du contrôleur DMA : la ligne d'interruption est remise à 0, et le contrôleur est mis dans un état prêt à démarrer un nouveau transfert.

Puisque le contrôleur DMA possède deux interfaces VCI (une interface « cible » pour recevoir les commandes, et une interface « initiateur » pour lire et écrire les données) il faut deux automates pour contrôler ces deux interfaces. Le but de l'exercice est de définir précisément la structure de ces automates.



E1) Pourquoi faut-il que l'adresse de base du segment associé au contrôleur DMA en tant que cible soit un multiple de 16 ?
Représenter graphiquement l'automate TARGET_FSM, qui posséde 7 états, et qui contrôle l'interface VCI « cible ». Les 7 états sont l'état IDLE, correspondant à l'attente d'une commande VCI, 4 états correspondant à l'écriture dans un des 4 registres SOURCE, DEST, NWORDS et RESET, un état correspondant à la lecture dans le registre STATUS. Le dernier état est l'état ERROR, correspondant a l'envoi d'un code d'erreur lorsqu'on reçoit un paquet commande VCI de longueur supérieure à 1 flit. On attachera à chaque transition l'expression Booléenne dépendant des entrées CMDVAL, A2, A3, EOP, READ, et RSPACK. (Le signal READ vaut 1 quand la commande VCI indique une commande de lecture, et les bits A2 et A3 sont les bits d'adresse à décoder).
Remplir le tableau définissant – pour chaque état de l'automate - les valeurs des signaux contrôlés par l'automate TARGET_FSM : CMDACK, RSPVAL, RERROR, REOP, ainsi que les 4 signaux d'autorisation d'écriture dans les 4 registres adressables en écriture.

E2) Complêtez le graphe de l'automate FSM_INIT qui contrôle l'interface VCI « initiateur », ainsi que l'écriture dans le tampon local et dans le compteur COUNT. On attachera à chaque transition l'expression Booléenne dépendant des entrées CMDACK, RSPVAL, ainsi que des signaux STOP, LAST, et END. Le signal STOP est un bit contrôlé par l'automate FSM_TARGET. Il est mémorisé dans une bascule et prend la valeur 1 quand le contrôleur DMA reçoit un ordre d'écriture dans le registre RESET ; il reprend la valeur 0 quand le contrôleur DMA reçoit un ordre d'écriture dans registre NWORDS. Le compteur COUNT le nombre de mots envoyés dans une transaction d'écriture, et le signal LAST est activé lorsque le compteur atteint la valeur (L-1). Le registre NWORDS est décrémenté à chaque écriture effective d'un mot dans le tampon mémoire destination. Le signal END prend la valeur 1 quand le registre NWORDS atteint la valeur 0.
Pour simplifier, on ne traitera pas les réponses VCI de type ERROR renvoyées par les deux mémoires source et destination.
Remplir le tableau définissant les valeurs des signaux CMDVAL, CMD, PLEN, EOP, RSPACK, INCR (commande d'incrémentation de COUNT et de décrémentation de NWORDS), en fonction de l'état de l'automate FSM_INIT.

E3) Soient L la longueur de la rafale (en nombre de mots de 32 bits), et M la durée minimale d'un aller-retour dans le micro-réseau VCI (nombre de cycles entre l'émission du premier mot du paquet commande et le retour du premier mot du paquet réponse, en l'absence de contention). Calculer le nombre de cycle minimal pour transférer une rafale, et en déduire le débit maximal en octets/cycle..

E4) L'écriture dans le registre RESET du coprocesseur DMA peut être activée à tout instant, pour interrompre un éventuel transfert en cours. Pourquoi l'automate FSM_INIT qui contrôle les transferts de données ne prend-il en compte le signal STOP qu'à la fin du transfert (lecture puis écriture) d'une rafale de L mots?

Automate FSM_INIT du contrôleur DMA