Travaux Dirigés n°5

Traitement de l'information
et périphériques d'entrées/sorties

Exercice 1 : Cycle d'exécution d'une instruction

1. Citer les trois phases du traitement d'une instruction et le rôle de chacune.
2. Quels sont les rôles du compteur ordinal (CO) et du registre d'instruction (RI) ?
3. L'instruction `ADD R1, [2001]` est stockée à l'adresse `2000` et occupe deux mots. Décrire, dans l'ordre, les opérations de chaque phase. Quelle est la valeur du CO à la fin ?

Exercice 2 : Pipeline

Un processeur possède un pipeline à $k = 5$ étages (IF, ID, EX, MEM, WB), chaque étage durant un cycle d'horloge.

1. Combien de cycles faut-il pour exécuter $n = 6$ instructions sans pipeline ? avec pipeline ?
2. Donner la formule du gain G et calculer-le pour $n = 100$.
3. Vers quelle valeur tend G quand $n \to \infty$?
4. À partir de quelle valeur de n le gain dépasse-t-il 4,5 ?
5. Le pipeline réduit-il le temps d'exécution d'une instruction isolée ? Justifier.

Exercice 3 : Scrutation et interruption

Un périphérique possède un registre d'état `ETAT` (le bit 0 vaut 1 quand une donnée est prête) et un registre de données `DONNEE`. Il fournit un octet toutes les 10 ms (100 octets/s). Le processeur exécute une instruction en 2.

1. Écrire en pseudo-assembleur la boucle de scrutation qui attend une donnée puis la lit dans R_1.
2. La boucle comporte 3 instructions. Combien de tours de boucle sont effectués, en moyenne, pour chaque octet ? Quel pourcentage du temps du processeur est consacré à l'attente ?
3. Le transfert se fait maintenant par interruption, dont le traitement complet (sauvegarde du contexte, routine, restauration) dure 2 μs. Quel pourcentage du temps du processeur est consommé ?
4. On remplace le périphérique par un disque à 1 Mo/s (10^6 octets/s), avec une interruption par octet. Calculer la charge du processeur et conclure.

Exercice 4 : Accès direct à la mémoire (DMA)

Un contrôleur DMA transfère un bloc de 1 Ko (1024 octets) d'un périphérique vers la mémoire, à partir de l'adresse `0x2000`. Le DMA transfère un octet par vol de cycle, chaque vol de cycle durant 100.

1. Quelles informations le processeur doit-il fournir au contrôleur pour initialiser le transfert ?
2. Quelle est l'adresse du dernier octet écrit ?
3. Après le transfert de 10 octets, quelles sont les valeurs du registre d'adresse et du compteur (compteur initialisé à 1024) ?

4. Quelle est la durée totale du transfert ?
5. Comment le contrôleur signale-t-il la fin du transfert ?
6. On veut transférer un bloc de 4 Ko (4096 octets). Une interruption dure 2 μs ; en DMA, l'initialisation coûte 5 μs de temps processeur plus une interruption de fin. Comparer le temps processeur consommé avec une interruption par octet et avec le DMA. Dans quels cas le DMA est-il préférable ?

Corrigé

Exercice 1

1. **Fetch** : recherche de l'instruction en mémoire. **Decode** : décodage et recherche des opérandes. **Execute** : exécution, mise à jour des drapeaux et passage à l'instruction suivante.
2. Le **CO** contient l'adresse de la prochaine instruction ; le **RI** contient l'instruction en cours de traitement.
3. **Fetch** : CO = `2000` est placé sur le bus d'adresses, ordre de lecture ; le mot est disponible sur le bus de données et rangé dans le RI. **Decode** : le décodeur interprète le code `ADD` ; l'opérande est à l'adresse `2001` : cette adresse est envoyée sur le bus d'adresses, l'opérande revient par le bus de données et est stocké dans un registre. **Execute** : l'UAL calcule $R_1 +$ opérande, le résultat est rangé dans R_1, les drapeaux sont mis à jour, et le CO passe à `2002` (instruction sur deux mots).

Exercice 2

1. Sans pipeline : $n \cdot k = 6 \times 5 = 30$ cycles. Avec pipeline : $k + n - 1 = 5 + 6 - 1 = \mathbf{10}$ cycles.
2. $G = \dfrac{n \cdot k}{k + n - 1}$. Pour $n = 100$: $G = \dfrac{500}{104} \approx \mathbf{4{,}81}$.
3. $\lim\limits_{n \to \infty} G = k = 5$.
4. $\dfrac{5n}{n+4} > 4{,}5 \iff 5n > 4{,}5n + 18 \iff n > 36$. Pour $n = 36$, $G = 180/40 = 4{,}5$ exactement : il faut donc $\mathbf{n \geq 37}$ pour le dépasser strictement.
5. Non : une instruction traverse toujours les k étages, donc sa durée ne diminue pas. C'est le **débit** (une instruction terminée par cycle une fois le pipeline rempli) qui augmente.

Exercice 3

1. ATTENTE : LOAD R0, [ETAT] ; lire le registre d'etat AND R0, 1 ; isoler le bit 0 JZ ATTENTE ; tant que 0, on recommence LOAD R1, [DONNEE] ; donnee prete : la lire
2. Un tour dure $3 \times 2 = 6$. Un octet arrive toutes les 10 ms $= 10^7$: $\dfrac{10^7}{6} \approx \mathbf{1{,}67} \times \mathbf{10^6}$ tours par octet. Le processeur attend presque **100** % du temps : la scrutation est très inefficace pour un périphérique lent.
3. $100 \times 2\ \mu$s $= 200\ \mu$s par seconde, soit **0,02** % du temps processeur.
4. $10^6 \times 2\ \mu$s $= 2$ s de traitement par seconde : charge de **200** %, impossible. Une interruption par octet est inadaptée aux périphériques rapides ; il faut le **DMA**.

Exercice 4

1. L'identification du périphérique concerné, le **sens** du transfert, l'adresse de départ en mémoire (`0x2000`) et la taille du bloc (ou l'adresse de fin), soit le compteur = 1024.
2. Dernière adresse : `0x2000` + 1023 = `0x2000` + `0x3FF` = `0x23FF`.
3. L'adresse est incrémentée et le compteur décrémenté à chaque octet : adresse = `0x2000` + 10 = `0x200A`, compteur = 1024 − 10 = **1014**.
4. $1024 \times 100 = 102\,400 = \mathbf{102{,}4}\ \mu\mathbf{s}$.
5. Lorsque le compteur atteint zéro, le DMA envoie une **interruption** au processeur (« le transfert est terminé »).

6. Interruption par octet : $4096 \times 2\ \mu\text{s} = \mathbf{8{,}192}$ ms. DMA : $5 + 2 = \mathbf{7}\ \mu$s. Rapport $\approx \mathbf{1170}$. Le DMA est préférable pour les **transferts de blocs volumineux et rapides** (disque, réseau, vidéo) ; pour quelques octets ou un périphérique lent, l'interruption suffit, car l'initialisation du DMA ne serait pas amortie.