

Contrôle IF4-ARCH

Le 07/01/2010

Durée 2H

M. Akil

Documents autorisés : photocopiés (cours et TP) et notes de cours.

I. Processeur RISC - DLX

Pour chacune des questions ci-dessous (I.1 et I.2), vous préciserez vos choix (caractéristiques du processeur, ect...).

I.1 Soit la boucle suivante :

```
for (i=0; i < N; i++) {  
    si (A[i] < 0)  
        A[i] = -A[i];  
}
```

avec N un entier signé sur 32 bits, et les éléments de A sont des entiers signés sur 32 bits.

Donner le code en assembleur DLX de cette boucle (code commenté). Comment optimiser ce code (justifiez votre réponse) ? Donner le nombre de cycles de la version optimisée (expliquez le résultat obtenu).

I.2 Soit la boucle suivante :

Bou :

```
LD      F0, 0(r1)  
MULTD  F0,F0,F2  
ADDD   F0,F0,F4  
SD     8(r1),F0  
ADDI   r1,r1,#16  
SUB    r2,r1,r3  
BNEQ  r2,Bou
```

Proposer une programmation optimisée de cette boucle sur le processeur DLX. Expliquer succinctement cette optimisation et illustrer le déroulement de cette boucle, motiver votre choix et donner le nombre de cycles de cette programmation optimisée.

II. Mémoire Virtuelle

Soit un microprocesseur doté d'une mémoire virtuelle paginée à 2 niveaux. L'adresse linéaire ou l'adresse logique est de taille 24 bits (a_0 à a_{23}) est définie comme suit :

Bits 23 à 18	Bits 17 à 12	Bits 11 à 0
N° de l'hyperpage	N° de l'élément de la table des pages, c'est à dire le descripteur de la page recherchée	Déplacement dans la page

La taille d'un élément de l'hyperpage (descripteur de l'hyperpage) est de 4 octets. Celle d'un élément de la table des pages (descripteur de la page) est de 4 octets.

Donner :

- la taille en octets de la table des descripteurs de l'hyperpage,
- la taille en octets de la table des descripteurs des pages de l'hyperpage.

Le registre de base de la table des descripteurs de l'hyperpage contient l'adresse 008000H (H : hexadécimal). On suppose que l'on veut transformer l'adresse logique 111900H.

Donner :

- l'adresse du descripteur de l'hyperpage recherchée,
- l'adresse du descripteur de la page recherchée, on suppose que ce descripteur (c'est-à-dire le descripteur de l'hyperpage recherchée) contient la valeur : 045021H (cette valeur signifie : 045000H est l'adresse de base de la table des pages et 21H décrit les droits d'accès à la page, le 4^{ème} octet est utilisé par le système),
- l'adresse physique de la donnée à accéder en mémoire principale, on suppose que le descripteur de la page contient la valeur : 500021H (cette valeur signifie : 500000H est le cadre de la page physique en mémoire principale et 21H spécifie les droits d'accès à cette page, le 4^{ème} octet est utilisé par le système).

III. Mémoire Cache

III.1 Donner :

- les différents champs de l'adresse mémoire principale,
- la signification de chacun de ces champs,
- la taille en bits de chacun de ces champs.

Pour chacun des cas suivants :

1. mémoire cache à correspondance directe,
2. mémoire cache associative par ensembles de 8 éléments,
3. mémoire cache totalement associative.

On supposera que la taille de la mémoire principale est de 32 Méga octets et que la mémoire cache a une taille de 64k octets et que chaque bloc (appelé aussi ligne) est de 16 octets.

Dans le cas d'une mémoire cache associative par ensemble de 8 éléments, donner le synoptique de l'organisation de la mémoire cache et expliquer succinctement le processus de recherche d'une donnée dans le cache à partir d'une adresse mémoire.

III.2 soit la section de code ci-dessous :

```
for (j=0 ; j<100 ; j = j+1)
```

```
    for (i =0 ; i<5000 ; i = i+1)
```

```
        X[i][j] = a*x[i][j] ;
```

Proposer une version de code de cette section permettant de réduire les échecs dans le cache.

Quel est le type de localité que vous avez amélioré ? Expliquez votre réponse ainsi que le code proposé.