

ARCHITECTURE DES ORDINATEURS

EXAMEN Juin 2005 - 1H 30

Tous documents autorisés

PARTIE 1 : REPRESENTATION DES NOMBRES

Un format flottant 16 bits est présenté dans la figure ci-dessous. L'interprétation est similaire à celle des flottants IEEE simple et double précision. S est le bit de signe. L'exposant sur 5 bits représente la valeur de l'exposant augmentée de 15. La valeur 0 est réservée pour la représentation de 0 (Partie fractionnaire nulle) et des nombres dénormalisés (Partie fractionnaire non nulle). La valeur 31 est réservée pour l'infini (partie fractionnaire nulle) et NaN (partie fractionnaire non nulle).

Pour $0 < E < 31$, un nombre N correspond à $(-1)^S \times (1, \text{partie fractionnaire}) \times 2^{(E-15)}$

	1	5		10	
S	Exposant			Partie fractionnaire	

Q1) Donner respectivement la représentation hexadécimale (quatre chiffres)

- A) du plus grand nombre positif représentable
- B) du plus petit nombre positif normalisé représentable
- C) du plus petit nombre positif dénormalisé représentable

Q2) Donner les valeurs décimales correspondant aux nombres flottants 16 bits suivants représentés sous forme de quatre chiffres hexadécimaux :

- A) 3C00
- B) 4200
- C) C600
- D) 7C00

Q3) Quel est le résultat (sous forme de quatre chiffres hexadécimaux), de l'opération suivante :

$$4253 + C253 =$$

PARTIE 2 : JEU D'INSTRUCTIONS ET PROGRAMMATION ASSEMBLEUR

Soit le processeur DLX

Tous les registres ont 32 bits.

Le processeur a 32 registres entiers, de R0 à R31. Le registre R0 est câblé à 0 (R0=0). On peut le lire, mais l'écriture dans R0 ne produit aucun résultat. R31 contient l'adresse de retour des procédures. Il a 32 registres flottants, de F0 à F31. Il n'y a pas de registre flottant câblé à 0.

Les instructions sont de longueur fixe (32 bits). Le jeu d'instructions est donné en annexe.

La mémoire est adressable par octets.

NB : Les instructions de comparaison S__ positionnent le registre destination à 1 si la condition est vraie, et 0 si la condition est fausse.

Ex : SNE R4, R5, R6 met R4 à 1 si R5≠R6

La syntaxe assembleur est la suivante :

Instructions registres-registre

Code op, registre destination, registre source 1, registre source 2

Instructions registres-immédiat

Code op, registre destination, registre source 1, immédiat

Instructions mémoire

Code op, registre donnée, déplacement (registre adresse)

Instructions branchement

Code op, registre, déplacement

Instructions de saut

Code op, déplacement ou Code op, registre

On considère que les registres du processeur contiennent les huit chiffres hexadécimaux suivants :

R0	0000 0000
R1	1357 9B DE
R2	FCA8 6420
R3	4433 2211
R4	FFFF FFFF
R5	5678 4321

Q4) Donner le contenu des registres R6 à R11 (sous forme de huit chiffres hexadécimaux) après exécution des instructions suivantes. Indiquer les cas de débordement

- a) ADD R6, R1, R5
- b) ADDU R7, R1, R5
- c) ADD R8, R2, R4
- d) ADD R9, R3, R5
- e) SUB R10, R3, R4
- f) SUB R11, R4, R0

Q5) Donner le contenu des registres R14 à R16 (sous forme de huit chiffres hexadécimaux) après exécution des instructions suivantes :

- a) OR R14, R1, R3
- b) AND R15, R1, R3
- c) XOR R16, R1, R3

Q6) Que fait le programme suivant ?

ADDI R1, R0, 80FC

ADDI R2, R0, 8000

Boucle: SUB R3, R1, R2

SW R0, (R2)

ADDI R2, R2, 4

BNE R3, -3

Q7) Que fait le programme suivant ?

LHI R1,F000

ADDI R2,0, FFFC

ADD R2, R2, R1

ADD R3, R0, R0

Boucle : LW R4, (R1)

ADD R3,R3,R4

ADDI R1,R1,4

SUB R6, R1,R2

BNE R6, -4

Q8) Comportement du cache données

Le processeur DLX a un cache de données de 8 Ko, à correspondance directe, avec des blocs de 64 octets.

On suppose que deux tableaux X[256] et Y[256] de flottants simple précision sont aux adresses :

Adresse de X[0] : A000 0000_H

Adresse de Y[0] : B000 0000_H

a) Quel est le nombre de défauts de caches pour exécuter le programme ci-dessous (on donnera le nombre total de défauts de caches)

for (i=0; i<256; i++)

S+= X[i]*Y[i];

b) Quel est le nombre de défauts de caches si l'on suppose que l'on a maintenant un cache associatif deux voies (deux blocs par ensemble).

Jeu d'instructions DLX

Type d'instruction et code-op	Signification de l'instruction
Transferts de données	Transfère les données entre des registres et la mémoire ; le seul mode d'adressage mémoire est (registre + déplacement signé de 16 bits)
LB, LBU, SB	Chargement octet, octet non signé, rangement octet
LH, LHU, SH	Chargement demi-mot, demi-mot non signé, rangement demi-mot
LW, SW	Chargement mot, rangement mot (de/vers des registres entiers)
LF, SF	Chargement flottant simple précision, rangement flottant simple précision
Arithmétique et logique	Opérations sur les données entières ou logiques dans des registres entiers;
ADD, ADDI, ADDU, ADDUI	Addition, addition immédiats (tous les immédiats ont 16 bits). Les opérandes sont signés pour ADD et ADDI (avec extension de signe de l'immédiat). Les opérandes sont non signés pour ADDU et ADDUI (avec extension de 0 pour l'immédiat)
SUB, SUBU	Soustraction signée, non signée
AND, ANDI	Et, et immédiat (extension de 0 pour l'immédiat)
OR, ORI, XOR, XORI	Ou, ou immédiat, ou exclusif, ou exclusif immédiat (extension de 0 pour l'immédiat)
LHI	Chargement haut immédiat (charge la partie haute d'un registre avec un immédiat) et met à zéro la partie basse
SLL, SRL, SRA, SLLI, SRLI, SRAI	Décalages : sous forme immédiate (S_I) ou variable (S__); les décalages sont logique à gauche, logique à droite, et arithmétique
S__, S__I	Positionner la condition : "__" peut être EQ, NE, LT, GT, LE, GE
Contrôle	Branchements conditionnels et sauts; relatifs CP ou par registre
BEQZ, BNEZ	Branchement si registre entier égal/non égal à zéro; déplacement relatif de 16 bits ajouté à CP
J, JR	Sauts : déplacement de 26 bits ajouté à CP (J) ou destination dans le registre (JR)
Flottant	Opérations flottantes sur le format simple précision (SP)
ADDF	Addition de nombres SP
SUBF	Soustraction de nombres SP
MULTF	Multiplcation SP
DIVF	Division SP