

Compte-Rendu

TP3 : Architecture interne du contrôleur de cache

JIANG Hongbo 3602103

C) Application logicielle

Question C1 : Quelle est l'adresse de la première instruction de la fonction main ? Quelle est l'adresse de la première instruction de la boucle (instruction correspondant à l'étiquette *loop*) ?

la première instruction de la fonction main : 0x00400000

a première instruction de la boucle : 0x0040000C

Question C2 : Quelle sont les adresses de base des trois tableaux A, B, C ?

A:0x01000080

B:0x01000100 A+80+48

C:0x01000180 B+128

Question C3 : Pourquoi l'instruction *sw* est-elle placée après l'instruction *bne* (test de fin de boucle) ?

Pour éviter NOP , les codes sont optimisés par compilateur.

Question C4 : Quel est le nombre de cycles nécessaires pour exécuter une itération de la boucle, dans l'hypothèse où toutes les instructions et toutes les données sont dans le cache (pas de MISS).

7 cycles.

D) Fonctionnement du cache instruction

Question D1 : Sur combien de bits sont codés les trois champs BYTE, SET, et TAG d'une adresse 32 bits? Quelles sont leurs valeurs pour l'adresse 0x00400000 (correspondant à la première instruction du programme ?

BYTE : 4 bits

SET : 3 bits

TAG : 25 bits

Valeurs pour l'adresse 0x00400000 :

BYTE : 0000 SET : 000 TAG : 0000 0000 0100 0000 0000 0000 0

Question D2 : Représentez dans le tableau ci-dessous le contenu du cache instruction à la fin de la première itération de la boucle. Quelles instructions ont déclenché un MISS sur le cache instruction pour atteindre cet état ?

TAG	V	WORD-3	WORD-2	WORD-1	WORD-0
0x00400	1	lw	li	li	la
0x00400	1	add	addi	addi	lw
0x00400	1	addi	la	sw	bne
	0				
	0				
	0				
	0				
	0				

Les instructions de *la*, *lw* et *bne* ont déclenché un MISS sur le cache instruction.

Question D3 : Quel est le contenu du cache instruction à la fin de la 20e itération ? Calculez approximativement le taux de MISS lors de l'exécution complète du programme (on ne prend pas en compte ce qui se passe pendant la séquence de boot et pendant l'affichage du message).

Le contenu du cache reste la même que la première itération.

A la fin de la 20e itération, il y a totalement 5 MISS (7 instructions après la boucle), le sum d'instructions égale $20 \times 7 + 3 + 5 + 2 = 150$, donc le taux de MISS $5/150 = 3.33\%$

Question D4 : Pour quel type de cache l'état MISS_SELECT est-il indispensable?

Pour la cache set-associative MISS_SELECT est indispensable.

Question D5 : Complétez le graphe de l'automate ICACHE_FSM ci-dessous, en attachant à chaque transition sa condition de franchissement. Les conditions de transition dépendent des signaux Booléens suivants :

- IREQ : le processeur émet une requête instruction,
- IUNC : l'adresse correspond à un segment non caché,
- IMISS : le cache instruction fait MISS,
- VALID : l'automate PIBUS signale que la réponse est disponible,
- ERROR : l'automate PIBUS signale que la réponse correspond à une erreur.

A	IUNC·IREQ	L	VALID·	F	VALID·	J	
B	IREQ·IMISS·	I		G	VALID·ERROR	K	VALID·ERROR
C	+·	N/M		H		O	

Question D6 : Dans quel état être forcé cet automate lors de l'activation du signal RESETN? Quel doit être l'autre effet du signal RESETN sur le cache instruction ?

Dans l'état IDLE.

Le signal RESETN va aussi vider les valeurs de VALID et de ERROR.

E) Fonctionnement du cache de données

Question E1 : Quelles sont les valeurs des trois champs BYTE, SET, et TAG pour les adresses des éléments A[0] et B[0] des deux tableaux? Donnez le contenu du cache de données à la fin de la première itération de la boucle, en précisant quelles instructions entraînent un MISS sur le cache de données et donc un gel du processeur.

	BYTE	SET	TAG
A[0]	0000	000	0x01000
B[0]	0000	000	0x01000

TAG	V	WORD-3	WORD-2	WORD-1	WORD-0
0x01000	0	104	103	102	101
	0				
	0				
	0				
	0				
	0				
	0				
	0				

Dans la boucle loop, les deux premières */w* entraînent un MISS sur le cache de données et donc un gel du processeur.

Question E2 : Calculez le taux de MISS sur le cache de données pour l'exécution complète des 20 itérations de cette boucle. Donnez le contenu du cache de données à la fin de la 20e itération de la boucle.

TAG	V	WORD-3	WORD-2	WORD-1	WORD-0
0x01000	0	104	103	102	101
0x01000	0	108	107	106	105
0x01000	0	112	111	110	109
0x01000	0	116	115	114	113
0x01000	0	120	119	118	117
	0				
	0				
	0				

Question E3 : Complétez le graphe de cet automate, en attachant à chaque transition sa condition de franchissement. Ces conditions dépendent des signaux Booléens suivants:

- DREQ : le processeur emet une requête de donnée,
- WRITE : il s'agit d'une requête d'écriture,
- DMISS : le cache de donnée fait MISS,
- DUNC : l'adresse correspond à un segment non caché,
- WOK : le write buffer n'est pas plein,
- VALID : l'automate PIBUS signale que la réponse est disponible..
- ERROR : l'automate PIBUS signale que la réponse correspond à une erreur.

A	DUNC·DREQ·	L	VALID·	F	VALID·	J	
B	DREQ·DMISS·	I		G	VALID·ERR OR	K	VALID·ERR OR
C	+·	N/ M		H		O	
D	DREQ·WRITE·WOK	E	DREQ·WRITE·DMISS· WOK	P			

Question E4: Les expressions booléennes associées aux transitions de sortie de l'état WRITE_REQ sont très proches des expressions booléennes associées aux transitions de sortie de l'état IDLE. Quelle est la différence?

Dans la situation de +∞ , l'état prochain de WRITE_REQ est IDLE mais pas lui-même.

F) Accès au PIBUS

Question F1 : Pourquoi les écritures ont-elles la priorité la plus élevée ? Quel est l'inconvénient de ce choix ?

Parce que dans le write through , la capacité du tampon d'écriture est très petite, donc il faut écrire immédiatement. L'inconvénient : ça coûte cher et aussi moins efficace.

Question F2 : Quel est le mécanisme utilisé par les deux automates ICACHE_FSM et DCACHE_FSM pour transmettre une requête de lecture vers l'automate PIBUS_FSM ? Comment le serveur signale-t-il aux clients que la requête a été prise en compte ? Dans le cas d'une requête de lecture, comment le serveur signale-t-il au client que les données sont disponibles ?

On transmette la requête de lecture par signaux :

ICACHE_FSM : IUNC , IMISS

DCACHE_FSM : DUNC , DMISS

Question F3 : Pourquoi l'automate PIBUS_FSM n'a-t-il pas besoin de signaler qu'une requête d'écriture transmise par le tampon d'écritures postées s'est effectivement terminée ? Quelle est l'utilité de la réponse dans le cas d'une transaction d'écriture ?

Le serveur le signale par r_pibus_ins, r_pibus_ins=true répond au ICACHE, r_pibus_ins=false est pour DCACHE.

Question F4 : Complétez le graphe de l'automate PIBUS_FSM ci-dessous, en attachant à chaque transition sa condition de franchissement. Ces conditions dépendent des signaux suivants :

- ROK : Le tampon d'écritures postées est non vide
- SC : L'automate DCACHE_FSM demande une "écriture conditionnelle"

- IUNC : L'automate ICACHE_FSM demande une instruction non cachable
- IMISS : L'automate ICACHE_FSM demande une ligne de cache instruction
- DUNC : L'automate DCACHE_FSM demande une donnée non cachable
- DMISS : L'automate DCACHE_FSM demande une ligne de cache de donnée
- GNT : Le bus est alloué au composant PibusMips32Xcache
- LAST : un compteur initialisé dans l'état IDLE signale qu'il s'agit de la dernière adresse d'une rafale
- ACK : Réponse de la cible (3 valeurs possibles : READY, WAIT, ERROR)

X	R0K+SC	B'		E	GNT	G	LAST·
Y	(IUNC+IMISS +DUNC+DMIS S)	C		E'		G'	+ACK[WAIT]
Z		D	ACK[READY +ERROR]	F	LAST	H	ACK[READY+ERROR]
B	GNT	D'	ACK[WAIT]	F'		H'	ACK[WAIT]

Question F5 : Calculez coût minimal du miss sur le cache instruction (nombre de cycles de gel du processeur en cas de miss). Pour cela, vous dessinerez explicitement le chronogramme décrivant pour chacun des deux automate, la succession des valeurs stockées dans les registres d'etat. Même question pour le cache de données.

9 cycles.

Question F6 : Calculez le nombre total de cycles pour exécuter les 20 itérations de la boucle de la section C, en prenant en compte les cycles de gel dûs aux miss sur le cache instruction et sur le cache de données. En déduire la valeur du CPI (nombre moyen de cycles par instruction.

$$(150+18 \times 20+5 \times 9)/150=3.7$$

G) Expérimentation par simulation

Question G1 : A quel cycle le processeur exécute-t-il sa première instruction ?

-cycle 47

A quel code correspond cette instruction ?

-la \$8,A

Quel est le coût d'un MISS sur le cache instruction (coût du MISS = nombre de cycles de gel du processeur).

-

Question G2 : A quel cycle le processeur se branche-t-il au programme principal main() ?

-Cycle 56

Question G3 : Quel est le coût d'un MISS sur le cache de données?

A quel cycle le processeur termine-t-il l'exécution de la première itération de la boucle ?

-address = 0x1000180

Quelle est la durée totale de la première itération ?

-81-71= 10 cycles.

Question G4 : Quelle est la durée de la seconde itération?

Quelle est la durée de la troisième itération?

-

Comment expliquez-vous que le coût du MISS pour les itérations 2 et 3 est plus élevé que pour la première itération?

-

Question G5 : Quel est le taux de MISS sur le cache de données à la fin de l'exécution de la boucle?

-

Quelle est la durée totale du programme main (sans compter le temps d'exécution de l'affichage du message final).