

ARCHITETTURA DEGLI ELABORATORI
APPELLO 25 GIUGNO 2013
AA. 2012-2013

PRIMA PARTE

Es1

Fra i tempi di attesa a cui è soggetto il recupero di informazioni in un disco rigido, quale fra quelli elencati di seguito è aumentato nel caso in cui si diminuisca la velocità di rotazione del disco?

- a) Tempo di attesa del dispositivo
- b) Tempo di attesa del canale
- c) Tempo di posizionamento
- d) Tempo di latenza
- e) Nessuna delle precedenti

Es2

Si consideri una cache set-associativa a 4 vie da 8MB, con linee da 2KB. La cache è inserita in una gerarchia di memoria insieme ad una memoria centrale suddivisa in 2^{20} blocchi. Assumendo un indirizzamento al singolo byte, il formato degli indirizzi della memoria centrale è:

- a) 10 bit (tag) 10 bit (set) 11 bit (word)
- b) 10 bit (tag) 11 bit (set) 10 bit (word)
- c) 9 bit (tag) 11 bit (set) 10 bit (word)
- d) 11 bit (tag) 9 bit (set) 11 bit (word)
- e) Nessuna delle precedenti

Es3

Si consideri una cache di 256MB con associazione a gruppi a 256 vie e dimensione di linea di 8KB. Supponendo che il campo tag sia di 12 bit, la dimensione massima di memoria principale che la cache è in grado di gestire è

- a) 32GB
- b) 4 GB
- c) 512 MB
- d) 256 MB
- e) Nessuna delle precedenti

Es4

Si descriva in dettaglio il ciclo completo di fetch/execute delle istruzioni.

Es5

Si descriva in dettaglio le memorie DRAM

Es6

Nel contesto di una gerarchia di memoria, illustrare le possibili tecniche (incluse quelle che coinvolgono il compilatore) che si possono adottare per tentare di minimizzare il numero di miss.

Es7

Si descrivano in dettaglio i livelli 1 e 4 di RAID.

Es8

Sia dato un disco rigido con le seguenti caratteristiche

Capacità	Numero piatti/facce	Tracce per faccia/settori per traccia	Velocità rotazione
32GB	4/8	32768/512	7200 rpm

Inoltre il tempo totale medio per accedere a 32KB memorizzati in settori contigui su uno stesso cilindro è di circa 9,927083 ms.

Si calcoli il tempo medio di posizionamento della testina, descrivendo dettagliatamente tutti i passi per la soluzione.

SECONDA PARTE

Es1

Quante volte la CPU deve accedere alla memoria quando preleva ed esegue un'istruzione che ha 2 operandi, uno con modo di indirizzamento registro e uno con modo di indirizzamento immediato?

- a) 2
- b) 3
- c) 1
- d) 4
- e) nessuna delle precedenti

soluzione:

1 volta, ovvero una per il fetch, il resto è immediato.

es2

Si consideri la seguente rappresentazione in virgola mobile a singola precisione (IEEE 754) :
0100001011111000100000000000000

Il numero rappresentato è:

- a) -126.125
- b) -1.9707031
- c) 1.9707031
- d) 126.125
- e) nessuna delle risposte precedenti

Soluzione

Rappresentazione in virgola mobile a singola precisione IEEE 754 con:

☐ 1 bit per il segno

☐ 8 bit per esponente, quindi:

o esponente = esponente_polarizzato - (28-1-1) = esponente_polarizzato -127

☐ 23 bit per la mantissa (+ 1 bit (implicito) del primo 1)

0100001011111000100000000000000

segno = 0 → +

esponente_polarizzato = 10000101 in base due = $2^7 + 2^2 + 2^0 = 133$

esponente = esponente_polarizzato -127 = $133 - 127 = 6$

numero = + 1. 11111000100000000000000 x $2^6 = 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^{-3} = 126,125$

quindi risposta B

es3:

Si consideri una pipeline a 4 stadi: fetch (IF), decodifica (ID), elaborazione (EI), e scrittura dei risultati (WO), per cui:

- i salti incondizionati sono risolti (identificazione salto e calcolo indirizzo target) alla fine del secondo stadio (ID);
- i salti condizionati sono risolti (identificazione salto, calcolo indirizzo target e calcolo condizione) alla fine del terzo stadio (EI);
- il primo stadio (IF) indipendente dagli altri;

inoltre si assuma che non ci siano altre istruzioni che possano mandare in stallo la pipeline e che non sia implementato alcun meccanismo di trattamento dei salti.

Sapendo che:

- il 18% delle istruzioni sono di salto condizionale
- il 8% delle istruzioni sono di salto incondizionale
- il 60% delle istruzioni di salto condizionale hanno la condizione soddisfatta (prese)

Il fattore di velocizzazione della pipeline è:

- a) 3.230988
- b) 3.508772
- c) 3.086420
- d) 3.960031
- e) nessuna delle risposte precedenti

Percentuali:

- 8% = 0.08 salto incondizionale
- 18% = 0.18 salto condizionale, di cui:
 - o 60% = 0.60 preso ==> quindi il $0.18 * 0.60 = 0.108$ delle istruzioni totali
 - o 40% = 0.40 non preso ==> quindi il $0.18 * 0.40 = 0.072$ delle istruzioni totaliNessun trattamento dei salti. Verrà quindi processata l'istruzione successiva.
- Salti Incodizionati: Ho 1 ciclo di stallo: non ho caricato l'istruzione corretta.
- Salti Codizionati Presi: Ho 2 cicli di stallo: ho caricato l'istruzione errata, ma per scoprirlo devo aspettare la fine di EI.
- Salti Condizionati Non Presi: Non ho nessun ciclo di stallo in quanto è stata caricata l'istruzione corretta.

Percentuali con cicli di stallo:

- salto incodizionato: $0.08 * 1 = 0.08$
- salto condizionato preso: $0.108 * 2 = 0.216$
- salto condizionato non preso: $0.072 * 0 = 0$

quindi la frazione di cicli di stallo = $0.08 + 0.216 + 0 = 0.296$

Essendovi 4 stadi $\rightarrow k = 4$

$$\text{fattore di velocizzazione} = S_k = \frac{1}{1 + \text{frazioni cicli stallo}} k = \frac{1}{1 + 0,296} 4 = 3,086419$$

Quindi risposta C

Es4

Si spieghi in dettaglio la rappresentazione dei numeri reali secondo lo standard IEEE 754.

Lo standard IEEE 754 è lo standard di rappresentazione dei numeri in virgola mobile. È possibile rappresentare i numeri in due formati differenti: a precisione singola con 32 bit e precisione doppia con 64 bit. La rappresentazione singola si compone di un bit per il segno, 8 bit per l'esponente e 23 bit per la mantissa. Il formato doppio a 64 bit si compone di 1 bit per il segno, 11 bit per l'esponente e 52 bit per mantissa.

Inoltre possiamo avere altri due formati (singolo e doppio) che ci permettono di avere bit aggiuntivi per l'esponente e per la matissa, usati per rappresentare risultati intermedi di operazioni.

Per il formato singolo, a 32 bit posso avere:

- esponente polarizzato, con numeri da 1 a 254, quindi esponente da -126 a +127.
- Per esprimere i numeri non nulli utilizzo è
 $\pm 1, f...f \times 2^{e-127}$

es5

si descriva in dettaglio la modalità di indirizzamento indiretto. Discuterne pregi e difetti. La adozione di tale modo di indirizzamento è favoriti o sfavoriti in una architettura RISC ? motivare la risposta.

L'indirizzamento indiretto prevede che il campo indirizzo punti all'indirizzo dell'effettivo operando in memoria. Chiamando EA l'indirizzo effettivo ed (A) l'indirizzo che punta all'operando si ha tale schema:

$$EA = (A)$$

Il vantaggio di tale tecnica è il grande spazio di indirizzamento di cui si dispone. Il suo contro sono gli accessi alla memoria extra per il recupero degli operandi.

L'architettura RISC è nota per la sua semplicità e per i modi di indirizzamento semplici. Quasi tutte le operazioni vengono fatte utilizzando l'indirizzamento a registro, per cui l'indirizzamento indiretto è sfavorevole in una architettura risc, se non inutile contando che l'indirizzamento a registro è molto conveniente in un RISC data la caratteristica tipica di un ampio banco di registri.

Es6

Nel contesto di una pipeline, si consideri la tecnica che utilizza la tabella della storia dei salti. A cosa serve? Descriverla in dettaglio

La tabella della storia dei salti è una piccola memoria associativa a cui il processore accede in modo associativo quando si trova di fronte alla predizione di un salto. La tabella è composta da più righe: l'indirizzo di un'istruzione di salto, un certo numero di bit di storia e informazioni inerenti all'istruzione di destinazione. Ci sono due tipi di tabelle e dipendono da quanti bit vengono utilizzati, uno oppure due. Con un bit si ha solo come informazione se l'ultima istruzione ha eseguito o meno un salto. Con la tecnica di predizione a due bit è possibile memorizzare il risultato delle ultime due istanze di esecuzione dell'istruzione associata. Con l'algoritmo a due bit si ha una maggior precisione in quanto, per modificare la predizione sono necessari due errori consecutivi, anziché uno come accade con un bit. L'utilizzo di bit di storia appena descritto presenta un notevole svantaggio: se si decide di saltare, l'indirizzo del salto, non viene decodificato prima della fine dell'esecuzione dell'operando che punta al salto.

Es7

Si descrivano le possibili alternative di organizzazione di un processore multicore.

Le variabili principali nell'organizzazione multicore sono essenzialmente 4:

- 1) il numero di core sul chip : determina quanti parallelismo avremo
- 2) il numero di livelli di memoria cache: è una scelta importante poichè siamo arrivati ad avere tre livelli di cache L1, L2 e L3. Nei processori più recenti ogni core ha una sua cache L1 ed L2, mentre il livello L3 è condiviso per diminuire il numero di miss.
- 3) quantità di memoria cache condivisa: più cache è condivisa più hit ho.
- 4) scelta superscalari o multithreading simultaneo SMT: la famiglia Core i7 di intel integra un SMT poichè lo sviluppo del software ha permesso di sfruttare pienamente le risorse a disposizione, quindi una cpu con quattro core fisici può arrivare ad avere 16 thread. Nella precedente generazione Intel Core Duo, il software non era così sviluppato e quindi la scelta di core superscalari era obbligata.

Es8

Sia data la seguente sequenza di istruzioni assembler, dove i dati immediati sono espressi in esadecimali.

SW	\$8, 50(\$7)	$\text{mem}[50 + R7] \leftarrow R8$
LW	\$7, 1(\$2)	$R7 \leftarrow [1 + R2]$
ADD	\$2, \$7, \$8	$R2 \leftarrow R7 + R8$
ADDI	\$7, \$7, 3	$R7 \leftarrow [R7 + 3]$
SUBI	\$3, \$2, 4	$R3 \leftarrow [R2 - 4]$
SW	\$8, 7(\$3)	$\text{MEM}[7+R3] \leftarrow R8$
ADD	\$5, \$7, \$2	$R5 \leftarrow R7 + R2$

Si consideri la pipeline MIPS a 5 stadi vista a lezione, con possibilità di data-forwarding e con possibilità di scrittura e lettura successiva dei registri in uno stesso giro di clock.

Mostrare come si evolve la pipeline, spiegando nel dettaglio i motivi di un eventuale stallo o dell'utilizzo particolare del circuito di by-pass.

PS: terza colonna scritta da me per mostrare meglio il testo

	1	2	3	4	5	6	7	8	9	10	11	12	13
SW	IF	ID	EX	MEM	WB								
LW		IF	ID	EX	MEM	WB							
ADD			IF	ID	ID	EX	MEM	WB					
ADDI				IF	IF	ID	EX	MEM	WB				
SUBI						IF	ID	ID	EX	MEM	WB		
SW							IF	IF	ID	EX	MEM	WB	
ADD									IF	IF	EX	MEM	WB

DATA FORWARDING:

- dalla seconda alla terza riga: ho bisogno di R7 aggiornato per input EX_{add} , dato che esce da MEM_{lw} .
- per input SW_{ex} ho bisogno del valore aggiornato di R3, per cui uso il data-forwarding e lo passo in uscita da $SUBI_{ex}$

circuito by-pass

- per $ADDI_{ex}$ ho bisogno di R7 aggiornato, per cui uso il circuito di by-pass. (prima freccia verticale)
- per input EX_{subi} ho bisogno di R2 aggiornato, per cui uso il by-pass per prelevare da $WBadd$

QUESTO SECONDO ME È UN MODO PER SVOLGERE L'APPELLO, NON HO LA PRESUNZIONE DI DIRE SIA CORRETTO. LO CARICO perché SIA DI AIUTO AD ALTRI. DELL'ULTIMO ESERCIZIO NON SONO SICURO AL 100% NON AVENDO SEGUITO LO SVOLGIMENTO IN CLASSE. PER LA PRIMA PARTE SE QUALCUNO VUOLE SVOLGERLA E INVIARMELE PER AGGIUNGERLA FA COSA GRATA PER TUTTI. SE SONO NECESSARIE MODIFICHE O ALTRO CONTATTATEMI PURE SU FACEBOOK.

SALUTI

NICOLÒ SCAPIN