

Tutorato di architettura degli elaboratori

Gerarchie di Memoria

Andrea Gasparetto – andrea.gasparetto@unive.it

Esercizio 1

Considerare un sistema di memoria virtuale paginata, con dimensione dell'indirizzo virtuale di 40 b. Assumendo che ogni entry della Page Table (PT) includa 4 bit di stato (valid, dirty, protezione, uso), la sua dimensione risulta essere di 128 MB. Rispetto ad una dimensione di pagina di 16 KB, calcolare la dimensione dell'indirizzo fisico. Calcolare la dimensione (in bit) di una TLB di 256 entry, rispetto sia ad un grado di associatività 4, e sia ad un grado di associatività 0 (diretta).

Soluzione

Dimensione Page Offset → $\log_2(16KB) = \log_2(2^{14}) = 14$ b

Dimensione numero pagina virtuale → $40 - 14 = 26$ b

Numero di entry indirizzabili → 2^{26}

Considerando che $128MB = 2^7 * 2^{20} = 2^{27}$

Possiamo scrivere la seguente equazione (che considera la dimensione in Byte della PT), dove x è la dimensione del numero di pagina fisica:

$Size_{PT} = (2^{26} * (x + 4)) / 8$ → NumeroPagineIndirizzabili * (DimNumeroPaginaFisica + 4bStato) / 8

$2^{27} = (2^{26} * (x + 4)) / 8$ → $x = 12$ b

Dimensione indirizzo fisico = Dimensione numero pagina fisica + page offset = $12 + 14 = 26$ b

Il numero di pagina virtuale (26 b) calcolato in precedenza è utilizzato per accedere alla TLB.

Associatività 4

I 256 elementi sono suddivisi in 64 set ($256 / 4$, #ingressi / #vie) di quattro elementi, che richiedono 6 bit (INDEX) per essere indirizzati.

TAG → $26 - 6 = 20$ b.

Ogni entry è $20 + 4 + num_pg_fisica = 20 + 4 + 12 = 36$ b. Quindi, in totale, $256 * 36$ b.

Associatività 0

È necessario indirizzare direttamente i 256 elementi, ovvero sono necessari 8 bit ($\log_2 256 = 8$) (INDEX).

TAG → $26 - 8 = 18$ b.

Entry → $18 + 4 + num_pg_fisica = 18 + 4 + 12 = 34$ b

Totale → $256 * 34$ b

Esercizio 2

Considerare una memoria virtuale paginata con indirizzo virtuale di 36 b, e page size uguale a 4 KB. Supporre inoltre di avere una TLB associativa a 4 vie, che contiene 1024 ingressi. Rispetto a tale TLB, l'ingresso 10 della TLB è composto come segue:

Via 0	TAG=0x8ff0	PAGE NUMB=0x00ab0
Via 1	TAG=0x88ff	PAGE NUMB=0x0ff00
Via 2	non valida	–
Via 3	TAG=0xaa01	PAGE NUMB=0x0ab22

Trovare la traduzione (indirizzo fisico) dell'indirizzo virtuale 0x88ff0a001.

Data una cache associativa a 2 vie, composta da $8 \cdot 1024$ ingressi, block size uguale a 16 B, scomporre l'indirizzo fisico individuato in precedenza in TAG, INDEX e OFFSET.

Soluzione

Poiché la pagina virtuale è 4096 B ($4 \cdot 2^{10}$), abbiamo che il page offset è uguale a $\log_2 4096 = 12$ b.

pagina virtuale = 36 - page offset = 36 - 12 = 24 b.

set della TLB = #ingressi / #vie = $1024 / 4 = 256 = 2^8$

da cui INDEX = $\log_2 2^8 = 8$ b.

Rispetto all'indirizzo virtuale 0x88ff0a001, il page offset corrisponde ai 12 b meno significativi, ovvero 0x001, per cui:

#pagina_virtuale = 0x88ff0a

L'INDEX corrisponde agli 8 bit meno significativi del #pagina_virtuale, ovvero 0x0a = 10_{10} , per cui l'insieme selezionato è l'undicesimo, quello descritto nella traccia del problema.

La TAG corrisponde ai rimanenti bit del #pagina_virtuale, ovvero 0x88ff, che ritroviamo nella Via 1 dell'insieme selezionato (INDEX=10).

Il numero di pagina fisica selezionato è quindi 0x0ff00 (pick dalla tabella), per cui aggiungendo il page offset otteniamo l'indirizzo fisico a 32 b ($20b + 12b$ dell'offset): 0x0ff00001

Riguardo alla cache:

#set = $8K / 2 = 4K$

OFFSET = $\log_2 16 = 4$ b

INDEX = $\log_2(\# \text{ set}) = \log_2 (4 \cdot 2^{10}) = 12$ b

TAG = DimIndirizzoFisico – OFFSET – INDEX = $32 - 4 - 12 = 16$ b

Scomponendo l'indirizzo fisico precedente otteniamo:

TAG = 0x0ff0

INDEX = 0x000

OFFSET = 0x1

Esercizio 3

Considerare una memoria virtuale paginata con indirizzo virtuale di 32 bit. Supporre che ogni pagina contenga 2048 Byte, che l'indirizzo fisico sia anch'esso di 32 bit, e che i tre bit più significativi di ogni entry della Page Table (PT) corrispondano alla tripla: (valid, dirty, reference). Rispondere ai seguenti quesiti:

1. Calcolare il numero di ingressi e dimensioni della PT e mostrare la struttura del singolo ingresso.
2. Considerare la seguente porzione della PT (il numero a sinistra indica l'indice della entry):

0	080000
1	a1a940
2	e2b000
.....	

Facendo riferimento a queste prime tre entry della PT, quali sono le pagine fisiche allocate (ovvero, quali sono i numeri di pagina fisica corrispondenti)?

3. Usando le informazioni del punto precedente, tradurre (e scrivere in esadecimale) il seguente indirizzo virtuale in indirizzo fisico: 0x00000aff.
4. Considerare che ogni entry della TLB è di 5 B. Qual è dimensione della TAG e il numero di insiemi della TLB?

Soluzione

1)

$$\text{OFFSET} = \log_2 2048 = \log_2 2^{11} = 11$$

$$\text{Dim. \#pagina_virtuale/fisica} = \text{DimIndirizzoPagina} - \text{OFFSET} = 32 - 11 = 21$$

$$\text{Entry ((valid, dirty, reference), DimNumeroPaginaFisica)} = 3 + 21 = 24 \text{ b} = 3 \text{ B}$$

$$\text{Numero entry PT } (2^{\text{Dim. \#pagina_virtuale/fisica}}): 2^{21}$$

$$\text{Dimensione PT: } 2^{21} * 3 \text{ B (Dimensione di ogni entry)} = 6 \text{ MB.}$$

2)

Traducendo in binario la prima cifra esadecimale di ogni entry, otteniamo la configurazione della tripla di bit (valid, dirty, reference):

0	000 0 80000	(valid, dirty, reference)
1	101 0 1a940	
2	111 0 2b000	
.....		

La prima entry con valid=0 indica che nessuna traduzione è presente per la pagina virtuale 0, mentre sono valide le successive. In particolare, le pagine fisiche allocate in corrispondenza delle pagine virtuali 1 e 2 sono, rispettivamente, 0x01a940 e 0x02b000 (si noti che i numeri di pagina sono rappresentati su 21 bit).

3)

Per tradurre 0x00000AFF in indirizzo fisico dobbiamo estrarre i primi 21 bit ed utilizzarli per accedere alla PT. In binario, l'indirizzo risulta essere uguale a

000000000000000000000001 01011111111

21b 11b (OFFSET)

Otteniamo quindi che la pagina virtuale è la seconda (ingresso 1) che ha come corrispondente numero di pagina fisica 0x01a940 (tabella risposta 2), ovvero 0 (0) 0001 (1) 1010 (a) 1001 (9) 0100 (4) 0000 (0).

Per ottenere la traduzione è sufficiente "appendere" i bit di offset alla pagina fisica ottenendo:

000011010100101000000 01011111111, ovvero 0x0d4a02ff.

21b 0x01a940 11b (OFFSET)

4)

$TAG = DimEntry - Dim\#pagina_virtuale - 3 (\rightarrow valid, dirty, reference) = (5B = 5 * 8 = 40b) - 21 - 3 = 16 b.$

$INDEX = 21 (Dim_indirizzo_pagina_virtuale) - TAG = 21 - 16 = 5 b.$

Numero di insiemi della TLB (2^{INDEX}) = 2^5

Esercizio 4

Supporre di avere un sistema di memoria virtuale paginata, con una TLB 4-way associative di 256 entry. Calcolare la dimensione della TAG_{TLB} , considerando che la dimensione del numero di pagina virtuale è 22 b. Considerando che ogni entry della TLB è grande 34 b, di cui 2 bit sono Valid e Dirty, e che le pagine hanno dimensione 1 KB, calcolare la dimensione dell'indirizzo fisico. Infine, supporre di avere una cache 2-way associative, composta da 2^{16} blocchi. Calcolare la dimensione in Byte della cache (parte dati), considerando che la dimensione della TAG_{cache} è 7 b.

Soluzione

Il numero di insiemi della TLB è $256/4 = 64$ set

$INDEX_{TLB} = \log_2 64 = 6 b$ Numero di bit necessari ad indirizzare 64 set

$\# PAG. VIRTUALE = 22 b = INDEX_{TLB} + TAG_{TLB} = 6 + TAG_{TLB}$

da cui

$TAG_{TLB} = 22 - 6 = 16 b$

Ogni entry della TLB è composta, oltre che dai due bit aggiuntivi, dalla TAG_{TLB} , e dal $\# PAG. FISICA$.

$Dim. entry = 34 b = 2 + TAG_{TLB} + \# PAG. FISICA = 2 + 16 + \# PAG. FISICA$

da cui

PAG. FISICA = 34 - 18 = 16 b

Poiché l'OFFSET di pagina è $\log_2 1K = 10$ b, abbiamo che

Dim ind. fisico = # PAG. FISICA + OFFSET = 16 + 10 = 26 b

La cache ha $2^{16}/2$ vie = 2^{15} insiemi, per cui $INDEX_{cache} = \log_2 2^{15} = 15$ b.

L'OFFSET_{cache} = Dim ind. fisico - INDEX_{cache} - TAG_{cache} = 26 - 15 - 7 = 4 b.

Per cui la dim. di un blocco è 2^4 B, mentre la dim. totale è data da $2^4 * 2^{16}$ blocchi = 2^{20} B = 1 MB.

Esercizio 5

Abbiamo le seguenti misure relative all'esecuzione di un certo programma su un processore a 2 GHz:

CPI_{ideale} = 2 Perc_{IW/sw} = 20% Data_miss_rate = 30%

Instr_miss_rate = 5% IC=10 Milioni T_{exe} = 65 msec

Calcolare il Miss penalty in ns.

Modificando la cache, e mantenendo inalterato il resto del sottosistema di memoria, si osserva un miglioramento del Data miss rate. Se otteniamo un tempo di esecuzione T_{exe} = 40 ms, quanto vale il nuovo Data miss rate?

Soluzione

1)

Sappiamo che:

$$\begin{aligned} CPI_{miss} &= (Instr_miss_rate + Data_miss_rate * Perc_{IW/sw}) miss_penalty = \\ &= (0.05 + 0.06) miss_penalty = 0.11 miss_penalty \end{aligned}$$

$$CPI = CPI_{ideale} + CPI_{miss} = 2 + 0.11 miss_penalty$$

$$No_cicli = CPI * IC = (2 + 0.11 miss_penalty) * 10^7$$

$$T = 1/2GHz = 1 / (2 * 10^9) s = 0.5 ns$$

$$\begin{aligned} T_{exe} &= No_cicli * T = (2 + 0.11 miss_penalty) * 10^7 * 0.5 ns = \\ &= (2 + 0.11 miss_penalty) * 5 * 10^6 ns = (2 + 0.11 miss_penalty) * 5 ms \end{aligned}$$

Considerando che T_{exe} = 65 ms, abbiamo:

$$65 = (2 + 0.11 miss_penalty) * 5$$

$$65 = 10 + 0.55 miss_penalty$$

$$Miss_penalty = (65 - 10) / 0.55 = 100 \text{ cicli.}$$

Per esprimere il miss_penalty in ns, basta moltiplicare per T:

$$miss_penalty = cicli * T = 100 * 0.5 ns = 50 ns$$

2)

Sia X il nuovo valore del Data_miss_rate:

$$CPI_{miss} = (Instr_miss_rate + X * Perc_{i/w/sw}) miss_penalty = (0.05 + 0.2 * X) 100 = 5 + 20 * X$$

$$CPI = CPI_{ideale} + CPI_{miss} = 2 + (5 + 20 X) = 7 + 20 X$$

$$No_cicli = CPI * IC = (7 + 20 X) 10^7$$

$$T'_{exe} = No_cicli * T = (7 + 20 X) 10^7 * 0,5 = (7 + 20 X) * 5 * 10^6 ns$$

Metto nell'equazione il tempo di esecuzione dato dalla consegna e calcolo il nuovo Data_miss_rate risolvendo l'equazione per X

$$T'_{exe} = (7 + 20 X) * 5 * 10^6 ns = 40 ms (== 40 * 10^6 ns)$$

$$35 * 10^6 + 10^8 X = 40 * 10^6$$

$$10^8 X = 5 * 10^6$$

$$X = 5 * 10^{-2} = 0.05 = 5\%$$

Esercizio 6

Si consideri un sistema di memoria virtuale paginata con indirizzo virtuale di 32 b e indirizzo fisico di 27 b. Il sistema comprende una TLB 4-way associative, con 512 ingressi totali, e Tag di 13 b. Calcolare il Page size (1).

Infine, la cache del sistema è 8-way associative, la parte dati è 1 MB, e il Block size è di 16 B. Calcolare Tag e l'Index (2).

Soluzione

1)

$$\# \text{ set TLB} = 512 / 4 \text{ vie} = 128$$

$$INDEX = \log_2(128) = \log_2(2^7) = 7 \text{ b.}$$

$$Page_offset = 32 \text{ (Dim. Indirizzo Virtuale)} - INDEX - TAG \text{ (Dato)} = 32 - 7 - 13 = 12 \text{ b.}$$

$$Page \text{ size} = 2^{Page_offset} = 2^{12} = 4 \text{ KB.}$$

2)

$$\# \text{ blocchi} = 1 \text{ MB} / 16 \text{ B} = 2^{20} / 2^4 = 2^{16}$$

$$\# \text{ set cache} = \# \text{ blocchi} / 8 \text{ vie} = 2^{16} / 2^3 = 2^{13}$$

$$Block_offset = \log_2(16 \text{ B}) = 4 \text{ b (per indirizzare ognuno dei byte del blocco ho bisogno di 4 bit).}$$

$$INDEX = \log_2(\# \text{ set cache}) = \log_2(2^{13}) = 13 \text{ b.}$$

$$TAG = 27 \text{ (DimIndirizzoFisico)} - INDEX - Block_offset = 27 - 13 - 4 = 10 \text{ b.}$$

Esercizio 7

Considerare una memoria virtuale con indirizzo virtuale di 32 b. Supporre che ogni pagina contenga 2 KB, che l'indirizzo fisico sia di 34 b, e che gli ingressi della page table abbiano un valid bit come primo bit.

Calcolare il numero di ingressi della page table e la dimensione in Byte (1).

Considerando la seguente porzione della page table (il numero a sinistra indica il numero di ingresso), individuare a quali pagine fisiche si riferiscono primi tre ingressi (2). Tradurre, usando tali informazioni, l'indirizzo virtuale

0x00000AFF nel corrispondente indirizzo fisico.

0 0xA42000

1 0x855000

2 0x1FF010

Sempre riferendoci allo stesso sottosistema di memoria, si consideri l'esistenza di una cache, associativa a 2 vie, la cui parte dati è di 1 MB. Se l'Index è uguale a 13 b, qual è la dimensione di ciascun blocco? Qual è la dimensione della Tag (3).

Soluzione

1)

Le pagine hanno dimensione $2048 \text{ B} = 2^{11} \text{ B}$. Sono quindi necessari 11 bit di offset. Da qui si ottiene che i bit usati per indirizzare le pagine virtuali (numero di pagina virtuale) sono $32 - 11 = 21$ e che ci sono 221 ingressi nella page table. Ogni ingresso della page table deve contenere $34 - 11 = 23$ b per la pagina fisico, più il bit di validità, quindi un totale di $24 \text{ b} = 3 \text{ B}$. La dimensione della page table è quindi

$2^{21} \cdot 3 \text{ B} = 6 \text{ MB}$.

2)

Per rispondere al secondo punto, è sufficiente controllare se i 3 ingressi sono validi (primo bit) e, in tale caso, eliminare il primo bit per ottenere la pagina corrispondente. Otteniamo che solo i primi due sono validi e corrispondono quindi alle pagine fisiche 0x242000 e 0x055000. Il terzo ingresso ha invece bit di validità uguale a zero e non corrisponde di conseguenza a nessun indirizzo fisico.

Per tradurre 0x00000AFF in indirizzo fisico dobbiamo estrarre i primi 21 bit ed utilizzarli per accedere alla page table. In binario, tale numero risulta essere 0000 0000 0000 0000 0000 1010 1111 1111. Otteniamo quindi che la pagina virtuale è la seconda (ingresso numero 1) che ha come corrispondente pagina fisica 0x055000, ovvero 000 0101 0101 0000 0000 0000. Per ottenere la traduzione è sufficiente "appendere" i bit di offset alla pagina fisica ottenendo:

0000101010100000000 01011111111

cioè

00 0010 1010 1000 0000 0010 1111 1111. Tale indirizzo può essere scritto in esadecimale come 0x02A802FF.

3)

La dimensione di ogni entry della cache è

$$\text{Size cache} / 2^{\text{vie}} \cdot 2^{\text{Index}} = 2^{20} / 2^{14} = 2^6 = 64 \text{ B}$$

Quindi Offset è $\log_2 64 = 6$ b. TAG = 34 - Offset - Index = 34 - 6 - 13 = 15 b.