

Esercizi 03

Memory Systems

Chapter 12

1) Una CPU richiede i seguenti indirizzi in esadecimale: 54, 58, 104, 5C, 108, 60, F0, 64, 54, 58, 10C, 5C, 110, 60, F0, 64. Supponendo che all'inizio la cache sia vuota e assumendo una politica di rimpiazzo LRU, determinare se ogni indirizzo produce un Cache Hit o Miss nei seguenti casi: a) Cache a mappatura diretta; b) Cache completamente associativa; c) Cache associativa a 2 vie.

Soluzione:

Cache mappatura diretta

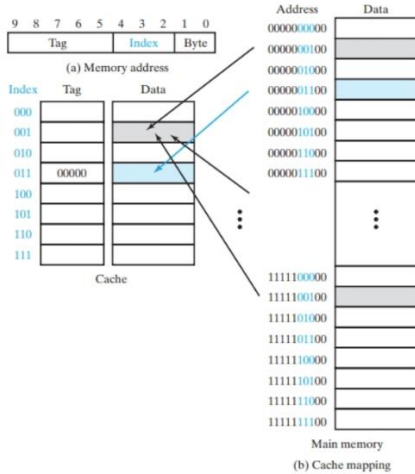


FIGURE 12-3 Direct Mapped Cache

Cache completamente associativa

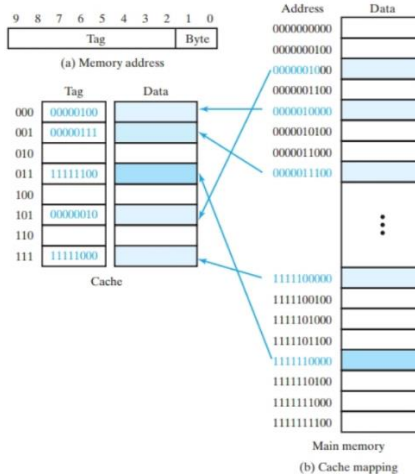
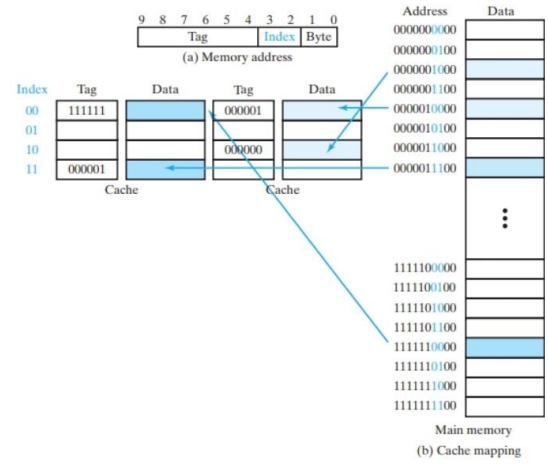


FIGURE 12-4 Fully Associative Cache

Cache associativa a 2 vie



Cache a mappatura diretta

			Index	Tag	Index	Tag	Index	Tag	Index	Tag
54	0010 101 00	M								
58	0010 110 00	M	000	0011	000	0011	000	0011	000	0011
104	1000 001 00	M	001	1000	001	0011	001	0011	001	0011
5C	0010 111 00	M	010	1000	010	1000	010	1000	010	1000
108	1000 010 00	M	011		011	1000	011	1000	011	1000
60	0011 000 00	M	100	0111	100	0111	100	1000	100	0111
F0	0111 100 00	M	101	0010	101	0010	101	0010	101	0010
64	0011 001 00	M&sub	110	0010	110	0010	110	0010	110	0010
54	0010 101 00	H	111	0010	111	0010	111	0010	111	0010
58	0010 110 00	H								
10C	1000 011 00	M								
5C	0010 111 00	H								
110	1000 100 00	M&sub								
60	0011 000 00	H								
F0	0111 100 00	M&sub								
64	0011 001 00	H								

Cache completamente associativa

			Index	Tag	Index	Tag
54	0010 101 00	M				
58	0010 110 00	M	000	0010 101	000	0010 101
104	1000 001 00	M	001	0010 110	001	0010 110
5C	0010 111 00	M	010	1000 001	010	1000 011
108	1000 010 00	M	011	0010 111	011	0010 111
60	0011 000 00	M	100	1000 010	100	1000 100
F0	0111 100 00	M	101	0011 000	101	0011 000
64	0011 001 00	M	110	0111 100	110	0111 100
54	0010 101 00	H	111	0011 001	111	0011 001
58	0010 110 00	H				
10C	1000 011 00	M&s				
5C	0010 111 00	H				
110	1000 100 00	M&s				
60	0011 000 00	H				
F0	0111 100 00	H				
64	0011 001 00	H				

Cache associativa a 2 vie

54	0010 101 00	M
58	0010 110 00	M
104	1000 001 00	M
5C	0010 111 00	M
108	1000 010 00	M
60	0011 000 00	M
F0	0111 100 00	M
64	0011 001 00	M&s
54	0010 101 00	M&s
58	0010 110 00	H
10C	1000 011 00	M
5C	0010 111 00	H
110	1000 100 00	M&s
60	0011 000 00	M&s
F0	0111 100 00	M&s
64	0011 001 00	H

Index	Tag	Tag
00	00110	01111
01	00101	10000
10	00101	10000
11	00101	

Index	Tag	Tag
00	01111	00110
01	00110	00101
10	00101	10000
11	00101	10000

Index	Tag	Tag
00	10001	00110
01	00110	00101
10	00101	10000
11	00101	10000

2) Ripetere l'esercizio precedente con i seguenti indirizzi: 20, 04, 28, 60, 20, 04, 28, 4C, 10, 6C, 70, 10, 60, 70

Cache a mappatura diretta

20	0010 0000	M
04	0000 0100	M
28	0010 1000	M
60	0110 0000	M&s
20	0010 0000	M&s
04	0000 0100	H
28	0010 1000	H
4C	0001 1100	M
10	0001 0000	M
6C	0110 1100	M
70	0111 0000	M&s
10	0001 0000	M&s
60	0110 0000	M&s
70	0111 0000	M&s

Index	Tag
000	011
001	000
010	001
011	011
100	011
101	
110	
111	000

Cache completamente associativa

20	0010 0000	M
04	0000 0100	M
28	0010 1000	M
60	0110 0000	M
20	0010 0000	H
04	0000 0100	H
28	0010 1000	H
4C	0001 1100	M
10	0001 0000	M
6C	0110 1100	M
70	0111 0000	M&s
10	0001 0000	H
60	0110 0000	H
70	0111 0000	H

Index	Tag
000	0111 00
001	0000 01
010	0010 10
011	0110 00
100	0010 00
101	0001 11
110	0001 00
111	0110 11

Cache associativa 2 vie

20	0010 0000	M
04	0000 0100	M
28	0010 1000	M
60	0110 0000	M
20	0010 0000	H
04	0000 0100	H
28	0010 1000	H
4C	0001 1100	M
10	0001 0000	M
6C	0110 1100	M
70	0111 0000	M&s
10	0001 0000	H
60	0110 0000	M&s
70	0111 0000	M&s

Index	Tag	Tag
00	0001	0111
01	0000	
10	0010	
11	0010	0110

3) Un computer ha una cache a mappatura diretta con indirizzi a 32 bits. L'indirizzamento viene effettuato per byte. La cache ha una capacità di 1KB e utilizza linee di 32 bytes. Utilizza il write-through e quindi non necessita di dirty bit. a) Quanti bits ci sono nell'Indice della cache? b) Quanti bits nel Tag? c) Qual è il numero totale di bits di immagazzinamento nella cache includendo i valid bits, i tag e le linee di cache?

Soluzione:

a) Dato che ogni linea contiene 32 bytes e l'indirizzamento viene fatto per byte (non per word), ho bisogno di $2^5=32 \rightarrow 5$ bits

Dato che la cache ha una capacità di 1KB, in totale ho $1024/32 = 2^5$ linee

b) Tags = $32 - 5 - 5 = 22$ bits

c) Ogni linea ha $(32 \times 8 + 22 + 1) = 279$ bits, in totale tutte le linee hanno $279 \times 32 = 8928$ bits

4) Un sistema con cache a mappatura diretta che contiene 32 bit di dati ha le seguenti specifiche: Ci sono 512 words (32 bits) nella main memory che sono mappate nelle 64-word locazioni nella cache. Si assume che un blocco di cache (linea) contenga solo 4 bytes (word). Determinare: a) In quale locazione la word con indice 0 (nella memoria centrale) sarà mappata; b) Quali altre words verranno mappate dalla memoria centrale nella stessa locazione della cache

Soluzione:

a) Dato che è una cache a mappatura diretta, la word con indice 0th sarà mappata nella locazione della cache con indice 0th

b)

Main Memory	
0000 0000 0000 0000	0
....	
0000 0000 0100 0000	64
...	
0000 0000 1000 0000	128
...	
0000 0000 1100 0000	192
...	

scritto male!
se ho 512 word
che sono mappate in 64
ho 8 righe

5) Si consideri un sistema con 2048 linee di cache e 8192 blocchi nella main memory. Trovare dove nella cache sono mappate i blocchi di memoria MMB-19 e MMB-5031 nel caso di a) mappatura diretta; b) mappatura completamente associativa; c) mappatura associativa a 4 vie

Soluzione:

Blocco MMB-19

a) Direttamente nella locazione della cache 19

b) In qualsiasi locazione della cache

- c) Nella mappatura associativa a 4 vie la cache viene divisa in 4 blocchi, quindi ogni blocco avrà $2048/4 = 512$ sets
 $19 \bmod(512) = 19 \rightarrow$ Quindi viene mappato in una qualsiasi locazione del set 19

Blocco MMB-5031

- a) $5031 \bmod(2048) = 935$
b) In qualsiasi locazione della cache
c) $5031 \bmod(512) = 423 \rightarrow$ In qualsiasi locazione del set 423

6) Trovare la dimensione del tag, dell'indice e l'offset per i due tipi di cache: 1) 16 KB con mappatura diretta e blocchi di 4-word

Soluzione:

- 1) Dimensione della cache = 16×2^{10} bytes
Numero di bytes per linea \rightarrow 4 words = 4×4 bytes = 16 bytes
Numero di linee \rightarrow 16×2^{10} bytes / 16 bytes = 2^{10} linee = 1024 linee
Numero di bit per indirizzare le linee \rightarrow Index bits = 10
Offset bits = 4
Tag bits = $32 - 10 - 4 = 18$ bits

7) Una memoria cache ha un tempo di accesso dalla CPU di 4 ns e la main memory ha un tempo di accesso di 40 ns. Qual è il tempo di accesso effettivo per la gerarchia cache-main memory se la percentuale di Hit è: a) 0.91; b) 0.82 e c) 0.96?

Soluzione:

- a) Effective Access Time = $0.91 * 4\text{ns} + 0.09 * 40 \text{ ns} = 7.24 \text{ ns}$
b) Effective Access Time = $0.82 * 4\text{ns} + 0.18 * 40 \text{ ns} = 10.48 \text{ ns}$
c) Effective Access Time = $0.96 * 4\text{ns} + 0.04 * 40 \text{ ns} = 5.44 \text{ ns}$

8) Ripetere il problema precedente con un tempo di accesso della cache di 1 ns e della main memory di 20 ns

Soluzione:

- a) Effective Access Time = $0.91 * 1\text{ns} + 0.09 * 20 \text{ ns} = 2.71 \text{ ns}$
b) Effective Access Time = $0.82 * 1\text{ns} + 0.18 * 20 \text{ ns} = 4.42 \text{ ns}$
c) Effective Access Time = $0.96 * 1\text{ns} + 0.04 * 20 \text{ ns} = 1.76 \text{ ns}$

9) Un sistema con memoria virtuale usa pages di 4 KB, 64-bit word and 48-bit di indirizzo virtuale. Un dato programma richiede 4263 pages. a) Qual è il numero minimo di page table richiesto? b) Qual è il numero minimo di voci richiesto nella directory page? c) Per le risposte date quanti sono le voci che ci sono nell'ultima page table?

Soluzione:

a) Ogni pagina contiene $4 \times 1024 \text{ bytes} = 4096 \text{ bytes}$

64-bit word equivale a $64/8 = 8 \text{ bytes}$

Quindi ogni pages table può mappare $4096 / 8 = 512 \text{ pages}$

Se un dato programma richiede 4263 pages, avrà bisogno di $4263 \text{ pages} / 512 \text{ pages} = 8.3 \text{ page tables} \rightarrow 9 \text{ pages tables}$

b) 9 voci nella directory table

c) $4263 - 8 \times 512 = 167$

10) La figura rappresenta il formato a 32 bit di un sistema di memoria a pagine. Calcolare la quantità di memoria fisica indirizzabile con questo sistema.

**Soluzione:**

Dato che il campo DIR è lungo 10 bits, 2^{10} o 1K di pagine sono possibili. Ogni voce nella page table è indicizzata dai 10 bits del campo FIELD, quindi ha 2^{10} voci cioè 2^{10} pagine indirizzabili. Ogni pagina contiene 2^{12} bytes. Dunque, in totale la memoria ha una capacità di $2^{10} \times 2^{10} \times 2^{12} = 2^{32} = 4\text{GB}$

11) Una TLB ha un hit rate del 95% e una penalità $T_{\text{MISS}} = 150$ cicli. Nel caso di HIT il tempo per la traduzione dell'indirizzo e $T_{\text{HIT}} = 0$. a) Qual è il tempo medio di traduzione di un indirizzo? b) Che cosa indicano il dirty bit e l'used bit? c) Le page tables sono immagazzinate in memoria e occupano spazio. Qual è la dimensione di una page table di 32-bit con pagine di 4K?

Soluzione:

a) $T_{\text{AVG}} = 0.95 * 0 + 0.05 * 150 = 7.5 \text{ cicli}$

b) Il used bit è settato ogni volta che c'è accesso a una pagina. Il dirty bit è settato ogni volta che una pagina viene riscritta. Questo significa che la pagina è cambiata dall'ultimo accesso.

c) Un indirizzo a 32 bit può indirizzare 2^{32} voci. Se ogni voce (page) ha dimensione $4\text{K} = 2^{12}$, il totale di pagine indirizzate è $2^{32} / 2^{12} = 2^{20}$. Se ogni voce ha 32 bit (4 bytes), la dimensione totale occupate è $4 \text{ bytes} * 2^{20} = 4\text{MB}$.

12) Considerare un Sistema con 85% Hit rate, 60 ns per la ricerca nella TLB e 800 ns per l'accesso alla memoria centrale. Trovare il tempo di accesso di una pagina e di lettura di una data word dalla pagina nelle seguenti situazioni: a) La pagina viene trovata nella TLB. b) La pagina non viene trovata nella TLB; c) Il tempo medio di accesso.

Soluzione:

a) $T = 800 \text{ ns} + 60 \text{ ns} = 860 \text{ ns}$ (lettura TBL + lettura in memoria)

b) $T = 800 \text{ ns} + 800 \text{ ns} + 60 \text{ ns} = 1660 \text{ ns}$ (lettura TBL + caricamento da memoria + lettura memoria)

c) $T = 0.85 * 860 + 0.15 * 1660 = 960 \text{ ns}$

13) Si supponga che una memoria cache per i dati adotti uno schema a mappatura diretta con 16 byte per blocco e 32 blocchi complessivi. Allora, per una memoria indirizzata a byte, il byte di indirizzo 0x00004321, se presente nella cache, si trova nel blocco di indice I (0..31) e in quel blocco occupa la posizione B (0..15), mentre il tag ad esso associato è T. Si indichino i valori di I, B e T come numeri interi con notazione in base dieci.

Soluzione:

In presenza di una memoria cache a mappatura diretta, un indirizzo generato dal processore individua un byte all'interno di uno dei blocchi appartenenti a una delle pagine in cui è ripartita la memoria centrale; questo indirizzo è costituito da tre campi:

Tag	Index	Byte
------------	--------------	-------------

Il campo di bit meno significativi fornisce l'indice B del byte cui il processore intende accedere all'interno del blocco che lo contiene; il campo di bit immediatamente più significativi specificano l'indice I di questo blocco all'interno della pagina in cui è suddivisa la memoria centrale; i rimanenti bit (nel campo più significativo) contengono il tag T che consente di verificare se quel blocco appartiene alla pagina cercata.

Nel nostro caso vi sono 16 byte per blocco: **quindi il campo B è di 4 bit**; vi sono 32 blocchi: **quindi il campo I è di 5 bit**; di conseguenza il campo T è di **23 bit**.

L'indirizzo 0x00004321, riscritto in binario e ripartito nei 3 campi, diventa:

000000000000000000000000100001	10010	0001
T=0x21	I=0x12	B=0x1

In base 10 abbiamo:

T = 33

I = 18

B = 1

14) Se l'address bus di un processore ha 32 linee e queste sono collegate ad una memoria cache a mappatura diretta che contiene 64 blocchi, ciascuno costituito da 16 byte, il comparatore che effettua il confronto per stabilire hit o miss confronta coppie di N bit.

In queste ipotesi, il byte di indirizzo 0xBBCCDDEE, se presente nella cache, è contenuto nel blocco di indice I. Si indichino in numero N di bit e l'indice I del blocco, in notazione decimale.

Soluzione:

Nel nostro caso vi sono 16 byte per blocco: **quindi il campo B è di 4 bit**; vi sono 64 blocchi: **quindi il campo I è da 6 bit**; il campo T è di 22 bit

Tag	Index	Byte
32	10 9	4 3 0

Il confronto per stabilire hit o miss avviene sul campo tag: $N = 22$

L'indirizzo 0xBBCCDDEE, riscritto in binario e ripartito nei 3 campi, diventa:

1011101111001100110111	011110	1110
T=0x2EF337	I=0x1E	B=0xE

In base 10 abbiamo:

$N = 22$ bit

$I = 30$

15) Una memoria cache abbia tempo di accesso $t_c = 62.5\text{ns}$. Con un hit rate del 80%, il tempo medio di accesso alla memoria è $t_{\text{avg}} = 90\text{ns}$. Allora la memoria convenzionale ha un tempo di accesso t_a pari a:

Soluzione:

Il tempo di accesso medio è pari a:

$$t_{\text{avg}} = 0.8 * t_c + 0.2 * t_a$$

$$90 \text{ ns} - 0.8 * 62.5 \text{ ns} = 0.2 * t_a$$

$$t_a = (90 \text{ ns} - 0.8 * 62.5 \text{ ns}) / 0.2 = 200 \text{ ns}$$

16) Una memoria cache, con un hit rate del 85%, consente un tempo medio di accesso da parte della CPU $t_{\text{avg}} = 80.7 \text{ ns}$, mentre la memoria centrale ha tempo di accesso $t_a = 130 \text{ ns}$. Allora il tempo di accesso della memoria cache t_c è:

Soluzione:

Il tempo di accesso medio è pari a:

$$t_{\text{avg}} = 0.85 * t_c + 0.15 * t_a$$

$$t_c = (t_{\text{avg}} - 0.15 * t_a) / 0.85 = (80.7 \text{ ns} - 0.15 * 130 \text{ ns}) / 0.85 = 72 \text{ ns}$$

17) In una memoria cache da 32 Kbyte, di tipo set-associativo a 4 vie, con 256 set da 4 linee e 32 byte per blocco, un byte si trova nella cache nel set di indice 34 e nel blocco in cui è presente occupa la posizione 11, mentre il tag associato a quel blocco è pari a 0x142A. Allora l'indirizzo del corrispondente byte in memoria è pari a IB.

Soluzione:

In presenza di una memoria cache set-associativa, un indirizzo generato dal processore, in caso di hit, individua un byte che si trova in una delle 4 linee di uno dei 256 set della cache; questo indirizzo è costituito da tre campi:

Tag	Index	Byte
------------	--------------	-------------

Il campo di bit meno significativi fornisce l'indice B del byte cui il processore intende accedere all'interno della linea che lo contiene; il campo di bit immediatamente più significativi specificano l'indice S del set in cui, se presente, si trova la linea in cui potrebbe essere stato ricopiato il blocco contenente il byte cercato; i rimanenti bit (nel campo più significativo) contengono il tag T che consente di verificare, con ricerca in parallelo (memoria associativa) se nel set individuato vi è la linea cercata.

Nel nostro caso vi sono 32 byte per blocco: il campo **B è di 5 bit**; vi sono 256 set: il campo **S è da 8 bit**; il campo **T è di 19 bit**.

Il tag T è: 0x142A, riscritto in binario su 19 bit, diventa:
T = 0000001010000101010

L'indice di set S è: 34 = 0x22, riscritto in binario, diventa:
S = 00100010

L'indice B del byte è: 11 = 0xB, riscritto in binario su 5 bit, è:
B = 01011

L'indirizzo IB del byte in memoria è, quindi:

IB = 0000001010000101010 00100010 01011
IB = 0x0285444B

18) In una memoria cache completamente associativa, costituita da 1024 linee da 32 byte, a ciascuna sua linea sono associati un TAG e 8 bit di controllo (dirty bit, bit di validità, bit per LRU, ...). La dimensione complessiva della cache (comprendente le 1024 linee, i relativi TAG e i bit di validità) è di 36 Kbyte.

In queste ipotesi, indicare il numero A di bit di cui sono costituiti gli indirizzi generati dal processore.

Soluzione:

Nella cache completamente associativa, un blocco viene inserito in una qualsiasi linea di cache vuota. Se tutte le linee in cache sono occupate, si sceglie quale linea svuotare, secondo una politica di rimpiazzo (FIFO o LRU).

In un sistema con cache completamente associativa, l'indirizzo generato dal processore è costituito da 2 campi:



Se ciascuna linea della cache è di 32 byte, il campo contenente l'indice B del byte è da 5 bit. Per trovare il numero A di bit di cui sono costituiti gli indirizzi generati dal processore ($A = T + B = T + 5$), è necessario individuare il numero T di bit del campo TAG.

Ciascuno dei 1024 elementi della cache è costituito da una linea da 32 byte ($32 \cdot 8$ bit) cui è associato un TAG (T bit) e 8 bit di controllo. Il numero di bit di uno di questi 1024 elementi è:
 $32 \cdot 8 + T + 8 = T + 33 \cdot 8$

La dimensione complessiva della cache è di 36 Kbyte; in bit è:
 $(T + 33 \cdot 8) \cdot 1024 = 36 \cdot 1024 \cdot 8$

Da cui $T = (36 \cdot 1024 \cdot 8 - 33 \cdot 8 \cdot 1024) / 1024 = 36 \cdot 8 - 33 \cdot 8 = 24$

Allora $A = 24 + 5 = 29$.

19) Il rapporto tra l'estensione della memoria logica e di quella fisica di un sistema paginato sia pari a 2^9 ; la dimensione delle pagine sia di 1 Kbyte; l'indirizzo fisico sia costituito da 16 bit. In queste ipotesi, si indichino: il numero N_e degli elementi della page table e il numero N_b di bit di cui è composto ciascuno di quegli elementi.

Soluzione:

L'indirizzo logico risulta da $2^9 \times 2^{16} = 2^{25}$ bit

Indirizzo fisico:

Physical page frame number	Offset
6 bit	1KB = 10 bit

Indirizzo logico:

Virtual page frame number	offset
15 bit	6 bit

La page table contiene tanti elementi quante sono le pagine logiche: $N_e = 2^{15}$
 Ciascun elemento della page table contiene: $N_b = 6$ bit

Page table

Physical page frame number
...
...
...

6 bit

20) Il set di registri di mappa di una MMU contiene un totale di 512 bit. Supponendo che l'indirizzo logico sia da 20 bit e che l'estensione di ciascuna pagina logica sia 2^{14} byte, l'estensione F della memoria fisica indirizzabile è:

Soluzione:

L'indice di pagina logica è costituito da $20-14 = 6$ bit:

Virtual page frame number	offset
6 bit	14 bit

La page table contiene $2^6 = 64$ elementi (quante le pagine logiche)

Il numero di bit di ciascun suo elemento è $512/64 = 8$

L'indirizzo fisico è costituito da $8+14 = 22$ bit

Physical page frame number	Offset
8 bit	14 bit

L'estensione F della memoria fisica è 2^{22} byte

21) Se in un sistema dotato di memoria virtuale, gli indirizzi generati dalla CPU sono da 32 bit e gli indici di frame contenuti nella page table sono da 16 bit, il numero massimo di byte contenuti in ogni pagina, tale per cui la memoria fisica risulta meno estesa di quella virtuale è pari ad M. Si esprima M

Soluzione:

L'indirizzo virtuale, generato dalla CPU è da 32 bit.

Il virtual page frame number sia da q bit: allora l'offset è da (32-q) bit:

Virtual page frame number	offset
q bit	(32-q) bit

L'indirizzo fisico contiene il physical page frame number da 16 bit e l'offset:

Physical page frame number	offset
16 bit	(32-q) bit

Per avere una memoria fisica meno estesa di quella virtuale, l'indirizzo fisico deve essere costituito da non più di 31 bit:

$$16+32-q \leq 31 \text{ cioè } q \geq 17$$

Allora l'offset, da cui dipende la dimensione delle pagine, deve essere:

$$\text{n. bit offset} = 32 - q \leq 32 - 17 = 15 \text{ bit}$$

La dimensione massima della pagina richiesta è: 2^{15}

22) Un sistema con memoria virtuale, utilizza indirizzi logici a 32 bit e fisici a 24 bit e una lunghezza di pagina pari a 4 Kbyte.

Supponendo che la MMU disponga di un TLB (realizzato da una cache completamente associativa) con 24 elementi, ciascuno includente i bit dei campi TAG e DATO, la dimensione K del TLB è complessivamente pari a:

Soluzione:

Un sistema con memoria virtuale, utilizza indirizzi logici a 32 bit e fisici a 24 bit e una lunghezza di pagina pari a 4 Kbyte.

A una lunghezza di pagina di 4 Kbyte ($4K = 2^{12}$) corrisponde un offset di 12 bit negli indirizzi virtuale e fisico.

Pertanto i campi Virtual Page frame number e Physical page frame number sono, rispettivamente da 20 e da 12 bit.

Indirizzo fisico:

Physical page frame number	Offset
12 bit	12 bit

Indirizzo logico:

Virtual page frame number	offset
20 bit	12 bit

Ciascuno dei 24 elementi del TLB contiene:

- un campo Virtual Page frame number (TAG, in memoria associativa): 20 bit
- un campo (DATO) con il corrispondente Physical frame page number: 12 bit

Il numero K di bit del TLB è dato dal prodotto:

(n. bit di un elemento) 32×24 (n. elementi)

$$K = 32 \times 24 = 768$$

23) In un sistema dotato di memoria virtuale, l'indirizzo virtuale ha 32 bit, quello fisico 28 bit e la dimensione delle pagine fisiche è pari a 16Kbyte.

Il MMU dispone di un TLB completamente associativo, nel quale il confronto sul TAG interessa t bit e l'indice di pagina fisica restituito dal TLB in caso di hit è costituito da f bit. Supponendo che nell'intervallo di tempo (t1, t2) non vi siano page fault e che ciascuna pagina fisica abbia la medesima probabilità di essere interessata da accessi (in modo tale che lo hit rate sia pari al rapporto tra il numero di elementi del TLB e il doppio del numero di pagine fisiche) e che, sempre in (t1, t2), lo hit rate sia pari al 50%, allora il TLB contiene n elementi. Calcolare t, f e n.

Soluzione:

In un sistema dotato di memoria virtuale, l'indirizzo virtuale ha 32 bit, quello fisico 28 bit e la dimensione delle pagine fisiche è pari a 16Kbyte.

Indirizzo fisico:

Physical page frame number	Offset
14 bit	16KB = $2^4 \times 2^{10} \rightarrow 14$ bit

Indirizzo logico:

Virtual page frame number	offset
18 bit	14 bit

Il MMU dispone di un TLB completamente associativo, nel quale il confronto sul TAG interessa $t = 18$ bit e l'indice di pagina fisica restituito dal TLB in caso di hit è costituito da $f=14$ bit.

Lo hit rate (50%) sia pari al rapporto tra il numero n di elementi del TLB e il doppio del numero di pagine fisiche):

$$0.5 = n / (2 \times 2^{14}) \text{ da cui } n = 2^{14}$$

24) In un sistema dotato di memoria virtuale, 16 accessi alla memoria consecutivi riferiscano le seguenti pagine virtuali: 3, 4, 2, 6, 4, 7, 1, 3, 2, 6, 3, 5, 1, 2, 3, 4.

Si supponga che la memoria fisica, costituita da 4 pagine, contenga inizialmente le pagine logiche 1, 2, 3, 4, caricate in questo ordine temporale.

Si indichino gli indici dei frame che vengono sovrascritti quando il processore richiede l'accesso alla pagina virtuale 7, nel caso di politica di rimpiazzo di tipo LRU e nel caso di politica di rimpiazzo di tipo FIFO.

Si indichino inoltre il numero di page fault che si verificano nel caso LRU e nel caso FIFO.

Soluzione:

I page fault si verificano quando è richiesta una pagina non presente in memoria.

Se tutta la memoria è in quel momento occupata da pagine precedentemente caricate, è necessario sostituirla una con quella richiesta.

La politica LRU sceglie come candidata alla sostituzione quella che non è stata riferita da più lungo tempo; con la politica FIFO quella che è stata caricata da più lungo tempo. Pagine virtuali riferite: 3, 4, 2, 6, 4, 7, 1, 3, 2, 6, 3, 5, 1, 2, 3, 4.

Pagine in memoria con politica LRU (x corrisponde a hit di pagina, un numero rosso a sostituzione per page fault):

	3	4	2	6	4	7	1	3	2	6	3	5	1	2	3	4
1				6				3			x					x
2			x				1					5				4
<u>3</u>	x					7				6				2		
4		x			x				2				1			

$F_{LRU}=3$
 $N_{LRU}=10$

Pagine in memoria con politica FIFO (x corrisponde a hit di pagina, un numero rosso a sostituzione per page fault):

	3	4	2	6	4	7	1	3	2	6	3	5	1	2	3	4
1				6					2					x	3	
<u>2</u>			x			7				6						4
3	x						1					5				
4		x			x			3			x		1			

$F_{FIFO}=2$
 $N_{FIFO}=10$

25) Un sistema è dotato di memoria virtuale paginata con indirizzo virtuale da 48 bit e indirizzo fisico da 32 bit. Sapendo che la dimensione delle pagine fisiche è di 1024 byte, e che ciascuno dei 64 elementi del TLB completamente associativo include un bit di validità, il numero di bit di cui è costituito il TLB è pari ad N

Soluzione:

Indirizzo fisico:

Physical page frame number	Offset
22 bit	1KB = 2^{10} → 10 bit

Indirizzo logico:

Virtual page frame number	offset
38 bit	10 bit

$$N = 64 * (38 + 22 + 1) = 64 * 61 = 3904$$

26) Si consideri un sistema in cui sono presenti i seguenti supporti di memoria, ciascuno caratterizzato dal corrispondente tempo di accesso indicato:

- memoria cache ($t_{ch} = 20ns$),
- memoria centrale ($t_{mc} = 60ns$),
- un disco usato come supporto alla memoria virtuale ($t_{di} = 12ms$ tempo medio).

In caso di miss nella cache, il tempo di accesso è pari a $t_{ch} + t_{mc}$, mentre in caso di page fault è pari a t_{di} (trascurando i pur presenti tempi $t_{ch} + t_{mc}$, richiesti per l'accesso alla memoria dopo che la pagina virtuale è stata ricopiata su un frame, in seguito ad un miss).

Assumendo che lo hit rate medio della cache sia pari a 90% e che si verifichi in media un page fault ogni 10000 accessi alla memoria, si indichi, in microsecondi, il tempo di accesso medio complessivo t_m del sistema di memoria durante l'esecuzione di $N \gg 10000$ istruzioni.

Soluzione:

$$t_{ch} = 20\text{ns}; t_{mc} = 60\text{ns}; t_{di} = 12\text{ms}.$$

Nell'eseguire $N \gg 10000$ istruzioni, la durata complessiva degli accessi alla memoria è $N \cdot t_m$.

Nell'esecuzione delle N istruzioni:

- in assenza di miss o page fault, il tempo di accesso sarebbe $N \cdot t_{ch}$
- $0.1 \cdot N$ istruzioni comportano un miss della cache e, per ciascuna di esse, il tempo di accesso diventa $0.1 \cdot N \cdot (t_{ch} + t_{mc})$
(questo è il tempo indicato nell'esercizio; si potrebbe però osservare che, nel caso di miss, non vi è un vero accesso alla cache, ma solo un confronto tra TAG e che questo confronto ha una durata trascurabile; allora la formula diverrebbe $0.1 \cdot N \cdot t_{mc}$; qui seguiamo l'indicazione dell'esercizio)
- $10^{-4} \cdot N$ istruzioni comportano un page fault e, per ciascuna di esse, il tempo di accesso diventa $10^{-4} \cdot N \cdot t_{di}$

$$\text{Allora: } N \cdot t_m = 0.9 \cdot N \cdot t_{ch} + 0.1 \cdot N \cdot (t_{ch} + t_{mc}) + 10^{-4} \cdot N \cdot t_{di}$$

da cui:

$$t_m = 0.9 \cdot t_{ch} + 0.1 \cdot (t_{ch} + t_{mc}) + 10^{-4} \cdot t_{di}$$

$$t_m = 0.9 \cdot 20 + 0.1 \cdot (20 + 60) + 10^{-4} \cdot 12 \cdot 10^6 \text{ ns} = 1226 \text{ ns}$$

$$t_m = 1.226 \text{ } \mu\text{s}$$