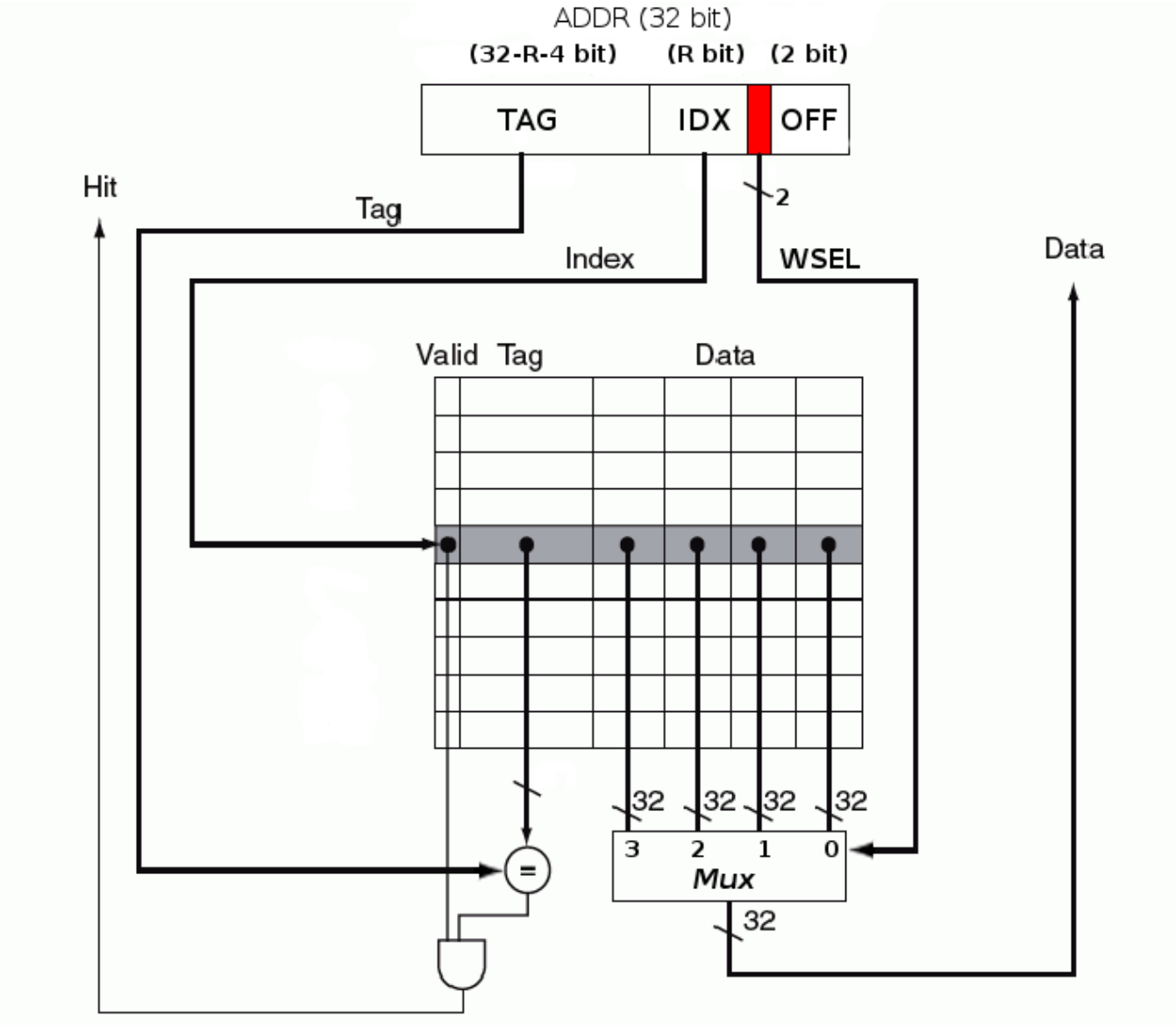
Strutture di Memoria 1

Laboratorio di Architettura

24 maggio 2013

4-Word Direct Mapped Cache



Simulazione di una cache

La struttura della cache è descritta dalla variabile globale

```
struct {
  unsigned int v;
  unsigned int tag;
  int elem[4];
} cache [4096];
```

La memoria principale è di 4MB, e corrisponde alla variabile globale

```
int mem [1048576]; /* (int = 32 bit) */
```

La memoria del sistema è indirizzabile a byte, per cui ogni indirizzo di memoria sarà composto da 22 bit (indirizzi 0 . . . 4194303).
L'indirizzo di memoria di 22 bit sarà a sua volta scomposto in:

- OFFSET (2 bit, bit 1 . . . 0) per l'indirizzamento a byte (vengono ignorati)
- WSEL (2 bit, bit 3 . . . 2) "word selector" per selezionare l'elemento ELEMX della riga;
- IDX (12 bit, bit 15 . . . 4, per indirizzare 4096 linee);
- TAG (6 bit, bit 21 . . . 16);

Attenzione!

Nella variabile `mem[]` si indirizzano direttamente le word, ma le funzioni che dovrete implementare accetteranno degli indirizzi di memoria (indirizzano i byte)
Esempio: se eseguo fetch_mem(600000); ovvero prendo dalla memoria principale la word 4 BYTE che contiene l'indirizzo 600000 passando attraverso la cache, significa che dovrò prendere l'elemento contenuto in `(600000>>2)=150000` della variabile `mem[]`, ovvero

```
x = mem[150000] ;
```

Requisiti

Memorizzare nelle variabili globali, il numero di cache hit e cache miss:

```
unsigned int rhit;
unsigned int rtot;
unsigned int whit;
unsigned int wtot;
```

Verranno aggiornate dalle funzioni `fetch_mem` e `write_mem`.

inizializzazione e stato

- void init_cache(void)
  Azzera tutti i campi della cache (V, TAG e gli ELEM per tutte le linee) e azzera anche le 4 variabili globali `rhit`, `rtot`, `whit` e `wtot`.
- void init_mem(void)
  Imposta a 1 ogni elemento della memoria principale (vettore `mem[]`).
- void cache_print(void)
  Per righe valide della cache (`v` == 1) stampare: IDX, V, TAG, elem[0], elem[1], elem[2], elem[3],

gestione cache (I)

int fetch_mem(unsigned int address)

1. calcola TAG, IDX e WSEL corrispondenti al parametro address e incrementa di un'unità il valore della variabile globale `rtot`;
2. controlla se il dato richiesto alla memoria è già presente nella cache (ovvero se, nella linea specificata da IDX, il bit di valid V è 1 e il TAG di address corrisponde al TAG contenuto in cache):
   a. in caso positivo stampa il messaggio "cache hit", restituisce l'elemento `elem[]` della cache corrispondente al campo WSEL di address e incrementa di un'unità il valore della variabile globale `rhit`;
   b. in caso negativo, pone il contenuto della memoria principale (preso dalla variabile `mem[]`) nella cache e lo restituisce come valore di ritorno. E' necessario inserire nella cache (nella stessa linea) anche gli altri elementi adiacenti di `mem[]` per i quali i campi TAG e IDX siano gli stessi. Se tale condizione si verifica, viene stampato il messaggio "cache miss". Infine, devono essere aggiornati coerentemente anche i campi V e TAG della linea modificata.
3. Ritorna il valore della word che contiene l'indirizzo richiesto.

gestione cache (II)

int write_mem(unsigned int address, int val)

1. calcola TAG, IDX e WSEL corrispondenti al parametro address e incrementa di un'unità il valore della variabile globale wtot;
2. controlla se il dato corrispondente all'indirizzo address è già presente nella cache: ciò è vero se il corrispondente bit di valid V è 1 e se il TAG della linea specificata da IDX corrisponde a quello dell'indirizzo address richiesto:
   a. in caso positivo, scrive nella cache il dato val nell'elemento `elem[]` corretto (specificato dal campo WSEL dell'indirizzo address) e modifica anche il contenuto nella memoria principale con lo stesso dato val (policy write-through), stampando il messaggio "write on hit". Viene incrementato di un'unità il valore della variabile globale whit.
   b. in caso negativo, la cache non viene modificata, ed il dato viene scritto solo nella memoria principale, quindi si stampa il messaggio "write on miss".

Main

Il main come prima cosa deve inizializzare la cache e la memoria principale usando le funzioni `init_cache` e `init_mem`.
Com test main dovrà contenere una sequenza di almeno 30 tra letture e scritture della memoria, con valori a scelta, che devono verificare tutti i casi possibili (cache hit/miss - write hit/miss).
Non è consentito l'input interattivo di tali test. Alla fine del programma, stampare il contenuto delle righe valide della cache con la funzione `cache_print`. Stampare inoltre le variabili rhit, rtot, whit, wtot.

Esempio:

```
me@mylaptop:~$ ./test
fetch_mem ( 000000 );
 cache miss
val = 1
fetch_mem ( 000004 );
 cache hit
val = 1
write_mem ( 000008, 3 );
 write on hit
write_mem ( 020004, 3 );
 write on miss

Cache (lines with v==1):
000 | 1 | 00 | 00000001 00000001 00000003 00000001

Statistiche:
 rhit: 1
 rtot: 2
 whit: 1
 wtot: 2
```

Requisiti

Requisiti fondamentali:

- Prima di chiamare `fetch_mem`, stampare l'indirizzo che si sta per richiedere, dopo aver chiamato la funzione, stampare il valore ritornato.
- Prima di chiamare `write_mem`, stampare l'indirizzo che si sta per richiedere e il valore che si vuole scrivere.
- Non è consentito l'uso di vettori di qualsiasi tipo per rappresentare i singoli bit. Tutte le operazioni vanno svolte direttamente sulle variabili intere, utilizzando le funzioni logiche e aritmetiche del C (potete usare gli operatori bit a bit & $|$ $\wedge$ $\sim$).
- Non è consentito l'uso della funzione pow o di altre funzioni per l'elevamento a potenza (usate lo shift $>>$ e $<<$).

Consegna

La relazione da consegnare è formata da:

1. listato (o listati, se più di uno), in linguaggio C, adeguatamente commentato;
2. uno (o più) file in formato testo contenente la cattura dell'output dei programmi di test. Usate la redirezione verso file:

```
me@mylaptop:~$ ./test > file_output.txt
```

   Oppure copiate l'output e incollate su file di testo. Non usate la cattura di immagini da schermo.

Non includete eseguibili.
Spedite tutto a arc1@fe.infn.it entro le ore 23:59:59 di venerdì 31 maggio.
L'oggetto della mail deve essere nella forma:
LAB1-N#esercitazione-#gruppo
(es: LAB1-N5-99)