Architettura degli Elaboratori - Correzione

Appello del 18 luglio 2006, A.A. 2005/06

Domanda 1

Una unità di elaborazione U, avente ciclo di clock τ, è collegata ad una memoria interallacciata con 8 moduli, ciclo di clock 30τ, e capacità complessiva di 128M parole. La latenza di trasmissione dei collegamenti è di 5τ.

U contiene, nella sua Parte Operativa, una memoria di capacità 16K che viene usata come cache operante su domanda, gestita con il metodo diretto e con blocchi di 8 parole.

U riceve in ingresso un valore J che rappresenta l'indirizzo base in memoria di un array A di 1K interi. Utilizzando la cache, U calcola la somma di tutti gli elementi di A e invia tale valore all'esterno.

Scrivere e spiegare il microprogramma di U, e valutare il tempo medio di elaborazione in funzione di τ.

Suggerimento: Si consiglia di utilizzare per U le seguenti interfacce: U-U0 (RDYIN, J, ACKIN), U-U1 (RDYOUT, OUT, ACKOUT), U-M (RDYOUT_M, IND, $DATAIN_0$, $RDYIN\_M_0$, $DATAIN_1$, $RDYIN\_M_1$, … $DATAIN_7$, $RDYIN\_M_7$).

Domanda 2

Il metodo diretto per l'indirizzamento della memoria cache ha, in generale, l'inconveniente che, quando il programma debba effettuare elaborazioni del tipo F(X, Y) (con X, Y oggetti di tipo anche complesso), un blocco di memoria principale di X ed un blocco di Y, da utilizzare contemporaneamente, potrebbero essere in corrispondenza con lo stesso blocco di cache, di conseguenza aumentando la probabilità di fault del programma. Dimostrare che, in un sistema con memoria virtuale e con cache, la situazione suddetta non può essere evitata con accorgimenti a tempo di compilazione.

Domanda 3

Un sistema a livello Lc comprende i seguenti processi, di cui IOP è un processo esterno e Q un processo da eseguire sulla CPU:

```
IOP :: … channel out CH_Q; …
        while true do
                {< acquisisci un blocco B di 1K parole dal dispositivo associato >;
                  send (CH_Q, B)
                }
Q :: … channel in CH_Q (1); …
        while true do
                { receive (CH_Q, B);
                  < elabora B >;
                }
```

L'unità di I/O corrispondente al processo esterno IOP è specializzata a firmware solo per

- supportare il modello Memory Mapped I/O,
- memorizzare le parole ricevute dal dispositivo in una memoria di registri presente nelle sua Parte Operativa, avente capacità di alcuni K parole ,
- interagire con la CPU via meccanismo delle interruzioni.

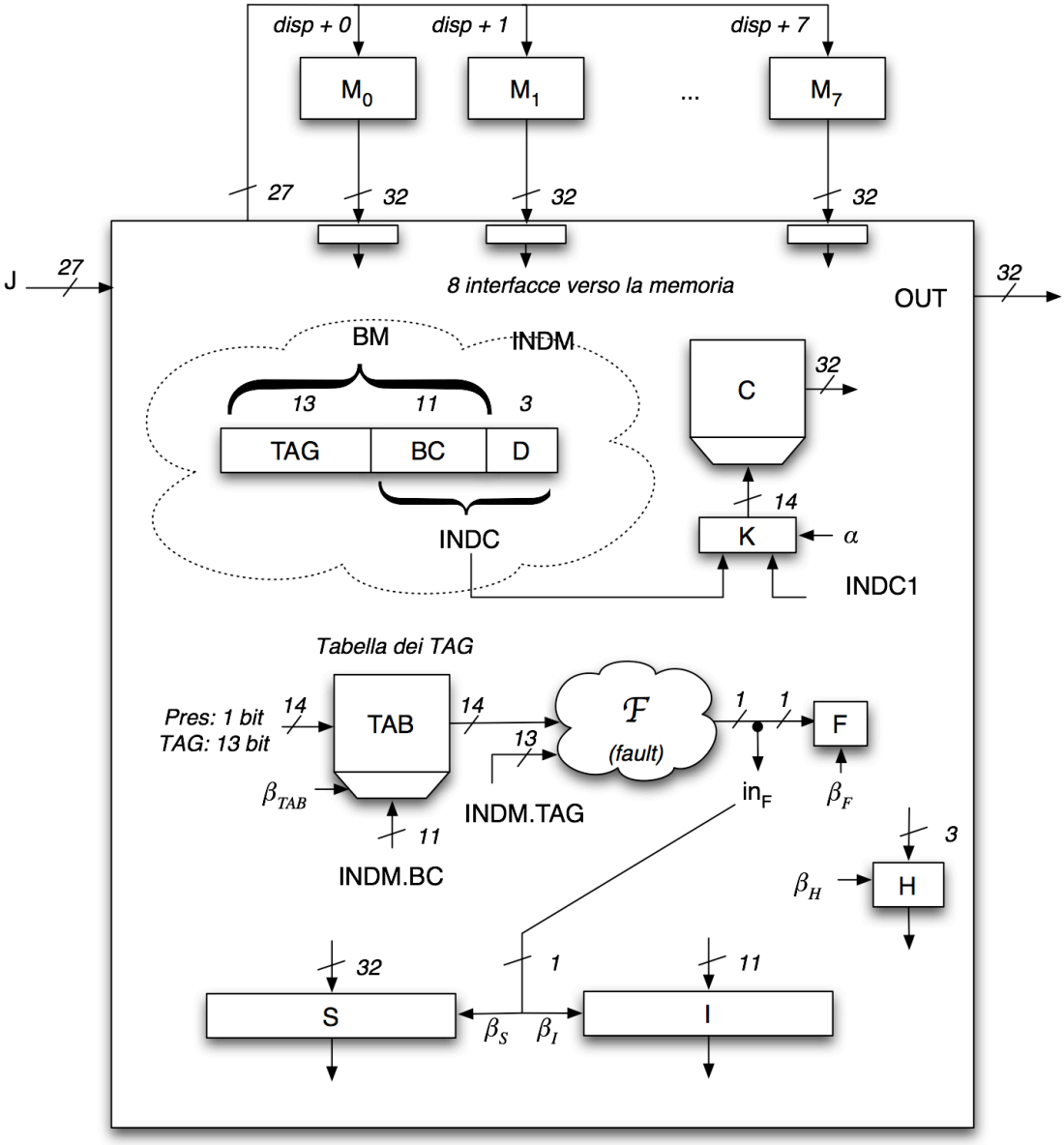
Spiegare come viene implementata la primitiva ***send*** *(CH_Q, B)* del processo IOP.

Traccia di soluzione

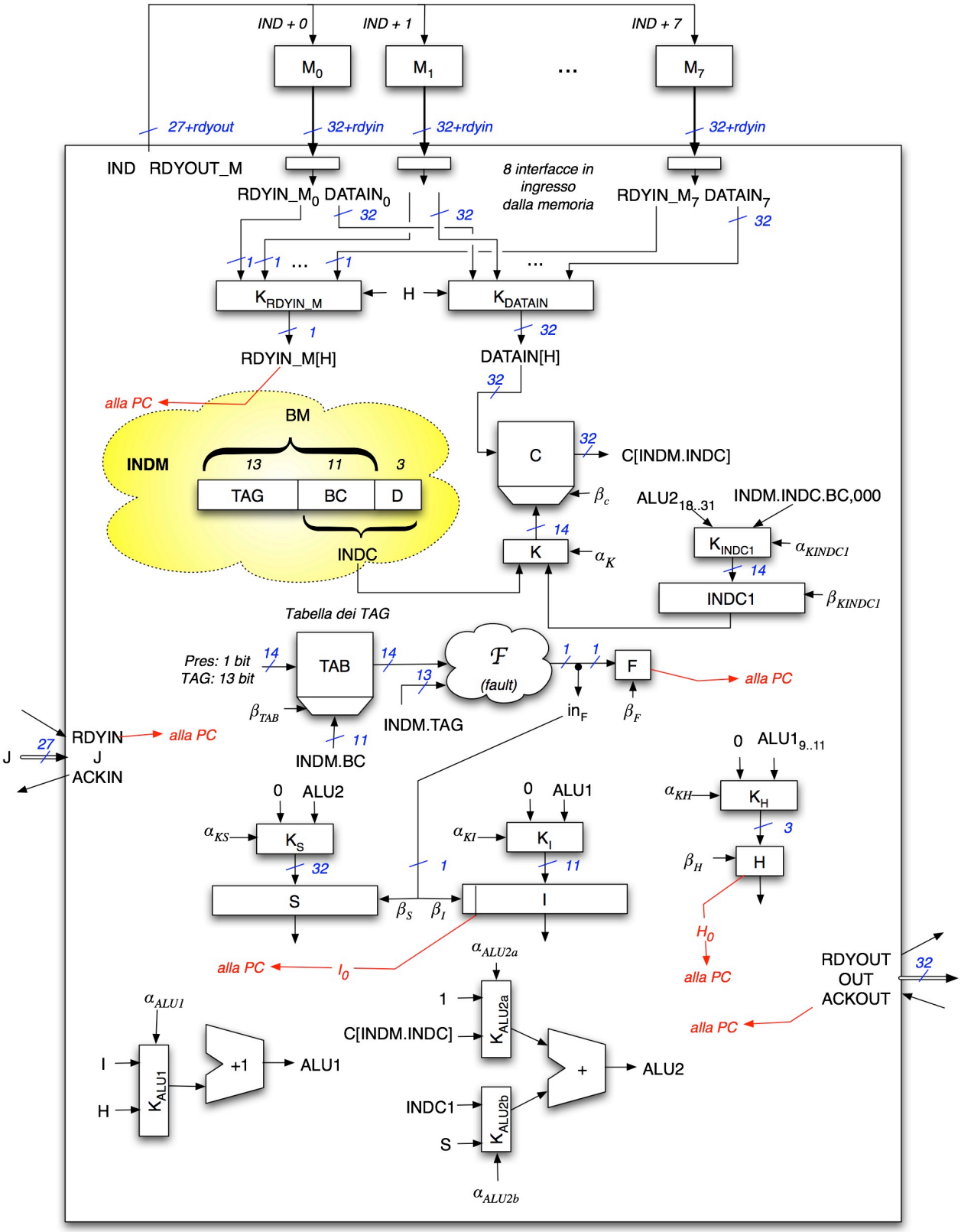
Domanda 1

Vedi esercizio n. 1 della raccolta n. 11 (quesiti su gerarchie di memoria).

Per svolgere l'esercizio è sufficiente basarsi sul seguente schema di Parte Operativa:



La Parte Operativa completa è la seguente (non necessaria per lo svolgimento).



0. (RDYIN = 0) nop, 0;

(= 1) reset RDYIN, set ACKIN, J → INDM, 0 → I, 0 → S, 1 → F, 1

1. (I_0 , F, ACKOUT = 0 1 –) S + C[INDM.INDC] → S, I+1 → I,

(**nor**(TAB[INDM.BC].TAG ⊕ INDM.TAG)) **and** (TAB[INDM.BC].PRES) → F, 1;

(= 0 0 –) (INDM.BM,000) → IND, set RDYOUT_M, ((INDM.INDC).BC,000) → INDC1, 0 → H, 2;

(= 1 – 0) nop, 1;

(= 1 – 1) reset ACKOUT, S → OUT, set RDYOUT, 0

2. (H_0 , RDYIN_M[H] = 0 0) nop, 2;

(= 0 1) reset RDYIN_M[H], DATAIN[H] → C[INDC1], INDC1 + 1 → INDC1, H + 1 → H, 2;

(= 1 –) INDM.TAG → TAB[INDM.BC].TAG, 1 → TAB[INDM.BC].PRES, 1

Dove

INDM: indirizzo memoria, I: contatore (11 bit), S: somma parziale (32 bit), F: indicatore di fault cache (1 bit)

$\mathcal{F}$ = **nor**((TAB[INDM.BC].TAG ⊕ INDM.TAG) **and** (TAB[INDM.BC].PRES)),

$\beta_S = \beta_I = in_F$

il valore (X,Y) in un assegnamento indica il concatenamento dei bit di X e Y, ad esempio ((INDM.INDC).BC,000) indica i bit del campo BC di INDC di INDM (la parte BC di INDM nella PO) a cui sono aggiunti 3 bit 0 nella parte meno significativa, che è proprio l'indirizzo base del blocco richiesto alla memoria interallacciata, che provvederà ad sommare un opportuna costante (0 – 7) in funzione del modulo in modo da poter mandare alla unità 8 parole di memoria consecutive (una per ogni modulo).

$T = T_{id} + N_{fault} * T_{transf}$

$T_{id} = (1K + 2)\ \tau = 1026\ \tau$ $\quad N_{fault} = 1024/8 = 128$

$T_{transf} = 2\ T_{tr} + \tau_M + 8\ \tau = 48\ \tau$, dato che $\sigma = 8$, $m = 8$

$T = 1026\ \tau + 128 * 48\ \tau = 7170\ \tau$

Domanda 2

Vedi esercizio n. 4 della raccolta n. 11 (quesiti su gerarchie di memoria).

Tramite accorgimenti a tempo di compilazione è solo possibile controllare l'allocazione delle strutture dati in indirizzi logici (e corrispondenti blocchi). Tali indirizzi saranno tradotti dalla MMU durante l'esecuzione in indirizzi fisici come mostrato in Figura 1 a)→b): IPL → IPF. L'associazione fra IPL e IPF è la più generale possibile e non costante nel tempo: la funzione di traduzione, infatti, può essere modificata dal gestore della memoria al fine di gestire dinamicamente quali e quante pagine logiche di ogni processo sono mappate in memoria principale. Dato che la cache è organizzata con il metodo diretto, l'indirizzo fisico dei blocchi sarà interpretato come mostrato in Figura 1 c.

Come appare evidente dalla figura, una parte di BC (Blocco Cache) è ottenuto da un certo numero di bit consecutivi di IPF, e da un certo numero di bit di D (questo perché in generale la dimensione di una pagina di memoria_virtuale/memoria_principale è un multiplo di un blocco di memoria_principale/cache).

Dunque, se gli indirizzi logici di due blocchi X e Y differiscono per la parte evidenziata in rosso in Figura 1 d) (displacement di blocco all'interno della pagina) allora tali blocchi non potranno mai essere in corrispondenza con lo stesso blocco di cache. Si noti che in questo caso X e Y violano le ipotesi del problema, quindi non c'è niente da dimostrare.

Rimane il caso in cui X e Y hanno la parte evidenziata in Figura 1 d) uguale. Tali blocchi potrebbero essere in corrispondenza con lo stesso blocco di cache. In questo caso ogni accorgimento statico rivolto ad allineare valori di IPF che non esprimano conflitti è inutile, visto che l'indirizzo logico verrà comunque trasformato dalla traduzione a)-b) in indirizzo fisico in modo non staticamente conosciuto. Questo dimostra la tesi.

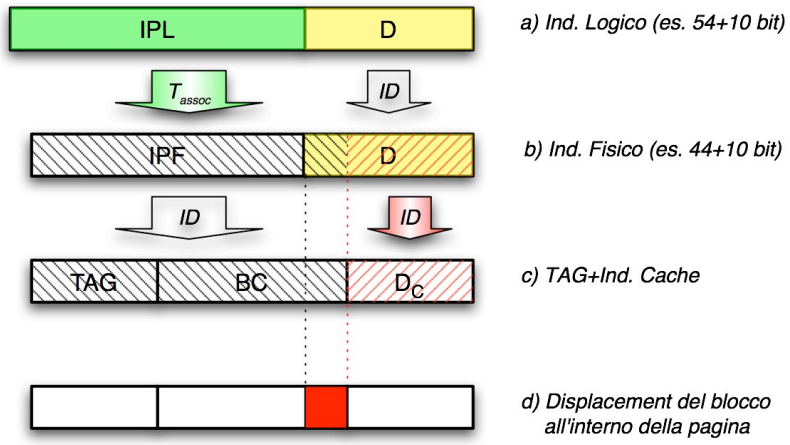


Figura 1

Con riferimento al primo caso, si noti che l'insieme dei blocchi che potranno essere resi non coincidenti in cache mediante accorgimenti statici è limitato alla cardinalità del numero di blocchi per pagina (tali blocchi non sono necessariamente nella stessa pagina logica). Ogni insieme più grande di blocchi può portare a conflitti in cache.

Domanda 3

Il processo esterno IOP non può effettuare la send direttamente perché l'unità di I/O è una unità specializzata e tale funzionalità non è fra quelle previste dalla specifica firmware. È però sufficiente associare un messaggio di interruzione dell'unità con l'esecuzione della primitiva send, che verrà quindi eseguita dalla CPU sotto il controllo di un handler di interruzione. Il msg d'interruzione contiene l'indirizzo logico base del blocco nello spazio di I/O. La routine di trattamento interruzione associa all'unità di I/O l'identificatore di CH_Q.