

*Architettura degli Elaboratori – Prima prova di verifica intermedia*  
*a.a. 2012–13, 19 dicembre 2012*

*Riportare nome, cognome e numero di matricola*

**Domanda 1**

Un'unità di elaborazione  $U_{\text{merge}}$  è interfacciata con  $U_1$  e  $U_2$ , da ognuna delle quali riceve una sequenza di interi da 32 bit, e con  $U_3$ , cui invia sia i dati ricevuti da  $U_1$  che quelli ricevuti da  $U_2$  realizzando un "merge" delle due sequenze. In presenza di dati disponibili sia da  $U_1$  che da  $U_2$ , si deve scegliere l'unità i cui dati sono stati spediti ad  $U_3$  meno di recente. Progettare l'unità minimizzando il tempo medio di elaborazione e fornendo il ciclo di clock.

**Domanda 2**

Si consideri la seguente computazione eseguita da un calcolatore con macchina assembler D-RISC:

*int A[N];*

*$\forall i = 0 .. N-1:$*

*$A[i] = \text{if}(A[i] > 0) \text{ then } F(A[i]) \text{ else } G(A[i])$*

Le funzioni  $F$  e  $G$  sono eseguite da un coprocessore di I/O. Valutare il tempo medio di completamento in funzione del ciclo di clock  $\tau$  del processore, del tempo medio di accesso  $t_a$  in qualunque memoria fisica, dei tempi medi di completamento  $T_F$  e  $T_G$  delle due funzioni, e della probabilità  $p$  di eseguire  $F$ .

**Domanda 3**

Dato un microprogramma con  $m$  microistruzioni di massimo  $f$  frasi ciascuna, che utilizza  $k$  variabili di condizionamento, considerando di poter utilizzare porte logiche con un massimo di 8 ingressi e con un ritardo pari a  $t_p$ :

- 1) si fornisca una maggiorazione per  $T_{\omega\text{PC}}$  e  $T_{\sigma\text{PC}}$ ;
- 2) si discuta come varia la maggiorazione se fosse noto che le  $k$  variabili di condizionamento vengono testate in microistruzioni diverse in modo che non ne vengano testate mai più di 4 contemporaneamente (si assuma  $k > 8$ ).

## Soluzione

### Domanda 1

L'unità ascolta nondeterministicamente messaggi dalle due interfacce d'ingresso. Se uno solo è presente, questo è inviato in uscita. Se entrambi sono presenti, la scelta è effettuata mediante una variabile booleana (registro di un bit)  $S$ , che viene posta a 0 (oppure a 1) se viene inoltrato il messaggio ricevuto da  $U1$  (oppure da  $U2$ ). Alternativamente, sarebbe stato possibile distinguere le due situazioni mediante due stati interni distinti della PC, quindi mediante due microistruzioni diverse.

La soluzione con variabile di condizionamento  $S$ , oltre a minimizzare il tempo medio di elaborazione, semplifica la realizzazione dell'unità in quanto il microprogramma risulta di una sola microistruzione, quindi l'unità è realizzata da una sola rete sequenziale:

0. (RDY1, RDY2, S, ACK3 = 0 0 --, 0 1 - 0, 1 0 - 0, 1 1 - 0) nop, 0;  
(= 1 0 - 1, 1 1 1 1) reset RDY1, set ACK1, IN1 → OUT, 0 → S, set RDY3, reset ACK3, 0;  
(= 0 1 - 1, 1 1 0 1) reset RDY2, set ACK2, IN2 → OUT, 1 → S, set RDY3, reset ACK3, 0;

Il tempo medio di elaborazione è uguale a un ciclo di clock. Il ciclo di clock dell'unica rete sequenziale è:

$$\tau = T_{\sigma} + \delta$$

La funzione di transizione dello stato interno, eccetto la gestione degli indicatori di interfaccia, è data da:

$$T_{\sigma|OUT}: in_{OUT} = \text{when } \beta \text{ do if not } \alpha_K \text{ then IN1 else IN2}$$

dove  $K$  è il commutatore all'ingresso di OUT, e

$$T_{\sigma|S}: in_S = \text{when } \beta \text{ do } \alpha_S$$

con:

$$\beta = RDY1 \text{ ACK3} + RDY2 \text{ ACK3}$$

$$\alpha_K = \alpha_S = RDY2 \bar{S} \text{ ACK3}$$

Quindi:

$$T_{\sigma} = T_{\sigma|OUT} = T_{\alpha_K} + T_K = 3 t_p$$

$$\tau = 4 t_p$$

### Domanda 2

Al coprocessore devono essere comunicati l'identificatore della funzione (ad esempio, 0 per F, 1 per G) e il valore del parametro intero. Questo può essere fatto con due istruzioni STORE in Memory Mapped I/O, che possono essere ridotte a una se l'unità di I/O riesce a interpretare opportunamente il valore dell'indirizzo contenuto nella richiesta. La sincronizzazione e il ritorno del risultato possono essere effettuati in più modi:

1. con attesa attiva dell'interruzione e risultato come seconda parola del messaggio di interruzione,
2. con sincronizzazione in attesa attiva mediante un semaforo di lock e risultato in una locazione di I/O o di memoria principale (se il DMA è disponibile),
3. con sincronizzazione in attesa passiva, commutando esplicitamente contesto, e risultato passato come seconda parola del messaggio di interruzione (evento: sveglia processo) o in memoria (di I/O oppure M).

Scegliamo la prima soluzione. Si vedano le esercitazioni per le altre due soluzioni. La terza è valida solo se la somma del tempo di servizio della commutazione di contesto e dell'handler dell'interruzione è molto minore del tempo medio di elaborazione delle funzioni richieste al coprocessore.

La compilazione del programma, con opportuna allocazione e inizializzazione dei registri generali, è:

```

LOOP: LOAD RA, Ri, Ra
      IF = 0 Ra, THEN
      STORE RbaseI/O, 0, Runo
      GO TO CONT
THEN: STORE RbaseI/O, 0, Rzero
CONT: STORE RbaseI/O, 1, Ra
      WAITINT RnomeI/O, Rclasse, Revento, Ra
      STORE RA, Ri, Ra
      INCR Ri
      IF < Ri, RN, LOOP
      END
    
```

due locazioni consecutive, con indirizzo logico base RbaseI/O, sono mappate nella memoria locale dell'unità di I/O e destinate a contenere l'identificatore della funzione e il valore di A[i];

il messaggio di interruzione è copiato in RG[Revento] e RG[Ra];

l'istruzione WAITINT abilita solo l'interruzione attesa;

nel tempo di completamento, della WAITINT viene valutata solo la fase di esecuzione (equivalente a una LOAD:  $2\tau + t_a$ ), in quanto la chiamata è sovrapposta all'elaborazione di F o G nel coprocessore.

Il tempo di completamento è dato da:

$$\begin{aligned}
 T_c &= N [ (8 - p) T_{ch} + 4 T_{ex-LD/ST} + T_{ex-WAITINT} + 2 T_{ex-IF} + T_{INCR} + (1 - p) T_{ex-GOTO} + p T_F + (1 - p) T_G ] = \\
 &= N [ (32 - 3 p) \tau + (13 - p) t_a + p T_F + (1 - p) T_G ]
 \end{aligned}$$

### Domanda 3

1) Nel caso più sfavorevole (asintotico), nelle funzioni delle uscite e dello stato interno successivo compaiono, *specificate*,  $\lceil \lg_2 m \rceil$  variabili dello stato interno e  $k$  variabili d'ingresso. Detto:

$$v = \lceil \lg_2 m \rceil + k$$

ogni termine AND di tali funzioni, realizzate come strutture ad albero con arietà 8, ha quindi un ritardo:

$$\lceil \lg_8 v \rceil t_p$$

Il numero di termini AND messi in OR è maggiorato dal numero di frasi  $m*f$ , in quanto nel caso peggiore (asintotico) tutte le frasi possono comportare l'utilizzo di una variabile (di controllo o dello stato successivo) avente valore uno. Quindi, la maggiorazione richiesta per  $T_{oPC}$  e  $T_{oPC}$  è:

$$(\lceil \lg_8 v \rceil + \lceil \lg_8 m f \rceil) t_p$$

2) Il valore di  $v$  diviene:

$$v = \lceil \lg_2 m \rceil + 4$$