

## Traccia di soluzione

La correzione viene data in forma estremamente schematica. Per le consuete spiegazioni, si vedano le soluzioni delle Esercitazioni di Verifica Intermedia e degli appelli precedenti.

### Domanda 1

Nella scrittura del microprogramma si è privilegiata la minimizzazione del numero di cicli di clock, utilizzando anche variabili di condizionamento complesse e anticipando la prima richiesta di lettura nella microistruzione iniziale.

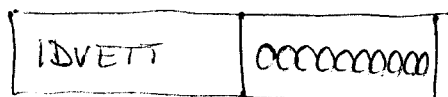
Le interfacce verso le unità destinarie devono essere implementate in modo parametrico, indiciano i registri OUT, RDYOUT e ACKOUT mediante IDDEST.

- $\emptyset$ . ( $RDYIN = 0$ )  $nop$ ,  $\emptyset$  ;  
 (=1)  $reset$  RDYIN,  $set$  ACKIN,  
 $decizien$  o  $IDVETT \rightarrow VETT$ ,  $IDDEST \rightarrow D$ ,  $VAL \rightarrow X$ ,  
 $\emptyset \rightarrow I$ ,  $\emptyset \rightarrow C$ ,  
 $decizien$  o  $IDVETT \rightarrow IND$ ,  $read \rightarrow OP$ ,  $set$  RDYOUTC, 1
1. ( $I_0$ , RDYINC,  $segno$  (DATAIN - X),  $ACKOUT[D] = 00--$ )  $nop$ , 1 ;  
 (=010-)  $reset$  RDYINC,  $I+1 \rightarrow I$ ,  $VETT+1 \rightarrow VETT$ ,  
 $VETT+1 \rightarrow IND$ ,  $read \rightarrow OP$ ,  $set$  RDYOUTC, 1 ;  
 (=011-)  $reset$  RDYINC,  $I+1 \rightarrow I$ ,  $C+1 \rightarrow C$ ,  
 $VETT \rightarrow IND$ ,  $write \rightarrow OP$ ,  $set$  RDYOUTC, 2 ;  
 (=1--0)  $nop$ , 1 ;  
 (=1--1)  $reset$  ACKOUT[D],  $set$  RDYOOT[D],  $C \rightarrow OUT[D]$ ,  $\emptyset$
2. ( $RDYINC = 0$ )  $nop$ , 2 ;  
 (=1)  $reset$  RDYINC,  $VETT+1 \rightarrow VETT$ ,  
 $VETT+1 \rightarrow IND$ ,  $read \rightarrow OP$ ,  $set$  RDYOUTC, 1

Nella correzione  $\uparrow$

la  $decizien$  "  $decizien$  o  $IDVETT$  "

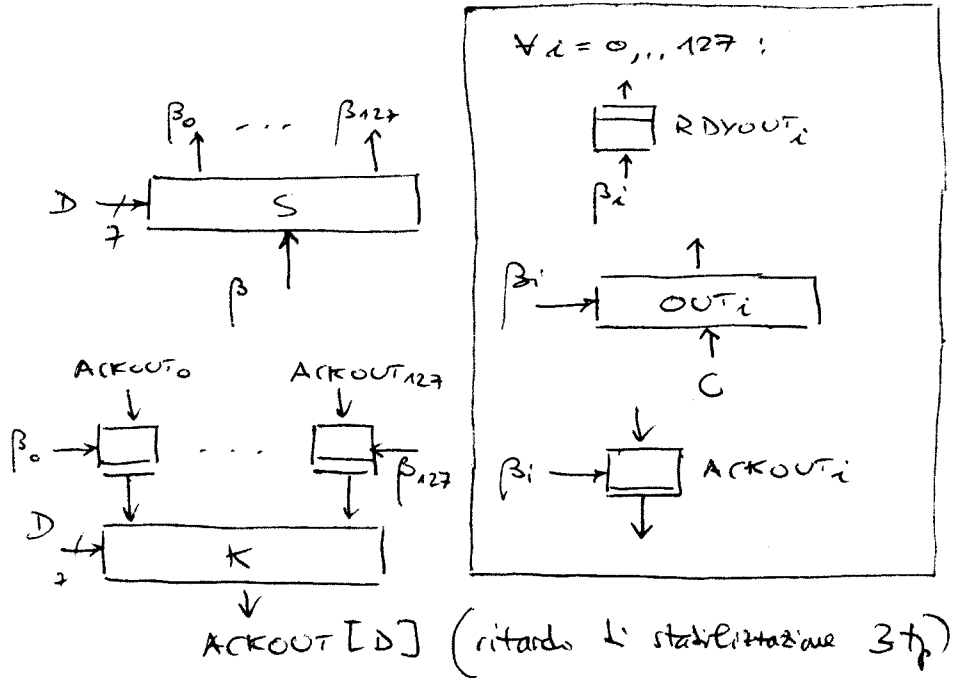
deve essere intesa come



$\uparrow$  parte meno significativa dell'indirizzo

Commenti:

- 1) "direzioni" = costante di direzioni "0"
- 2) interfaccia verso C: tipo processore
- 3) interfaccia parametrica verso  $\{\beta_0, \dots, \beta_{127}\}$ :

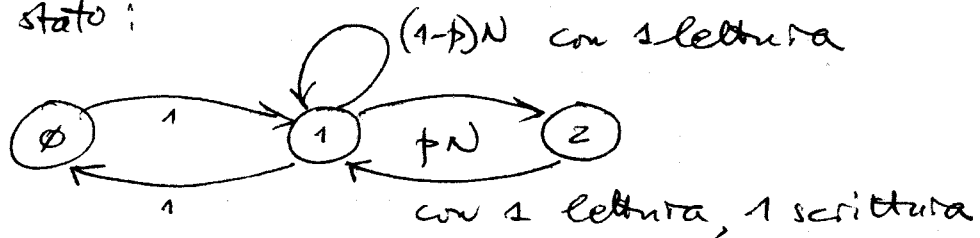


La valutazione del tempo medio di elaborazione è esposta nel seguito.

Tempo medio di elaborazione:

$$T = T_{id} + T_{swet}$$

Grafo di stato:



$$T_{id} \approx (1-p)N(\tau + t_c) + pN2(\tau + t_c)$$

$p = \text{prob}(\text{componente} < \text{VAL}), N = 1K$

$t_c = \text{tempo accesso cache diretta} = \tau$

$$T_{id} \approx 2(1+p)N\tau$$

Ciclo di clock.

$$T_{wpo} = T_{ALU} = 5t_p \quad x = \text{segno}(\text{DATAIN} - x)$$

$$T_{wpc} = T_{opc} = 2t_p \quad \begin{array}{l} \text{max. num. ingressi AND} = 7 \\ \text{" " " " OR} < 8 \end{array}$$

$T_{opo}$ : ALU contiene 3 ALU, di cui una per segno (DATAIN - x) e una per incrementare VERT oppure C, una per I+1:

$$T_{opo} = T_{ALU} + 2T_K = 9t_p$$

$$\tau = 5t_p + 2t_p + 9t_p + t_p = 17t_p$$

$$T_{id} \approx 2(1+p)N\tau = 34(1+p)Nt_p$$

$$T_{\text{fault}} = N_{\text{fault}} * T_{\text{traff}}$$

$$N_{\text{fault}} = \frac{N}{8} = \frac{N}{8} \text{ una volta letto un blocco,}$$

le scritture avvengono nello stesso blocco

$$T_{\text{traff}} = \tau_M + 2\tau_C + 5\tau = 48\tau$$

$$T_{\text{fault}} = 6N\tau$$

$$T = T_{\text{id}} + T_{\text{fault}} = 34(4+p)N\tau$$

Le scritture con il metodo Write-Through non ritardano l'elaborazione, in quanto la memoria interallocata ha una banda

$$B_M = \frac{8}{\tau_M} = \frac{0,4}{\tau}$$

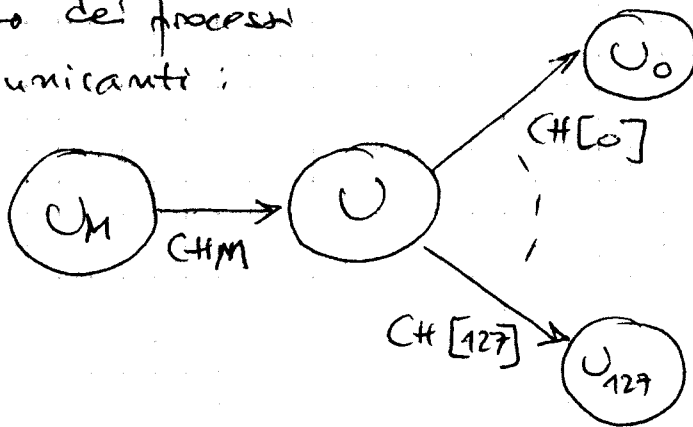
molto maggiore della massima banda di richieste da parte di  $C$ .

## Domanda 2

a) Di seguito sono riportati:

1. lo schema a processi comunicanti;
2. il codice del processo U in Lc;
3. lo schema della Memoria Virtuale di U, indicando gli indirizzi logici e registri generali di interesse;
4. la compilazione di U.

Grafo dei processi  
comunicanti:



tutti i  
canali sono  
bidirezionali con  
 $K=1$

Schema del processo U in  $\mathcal{L}_c$ :

```

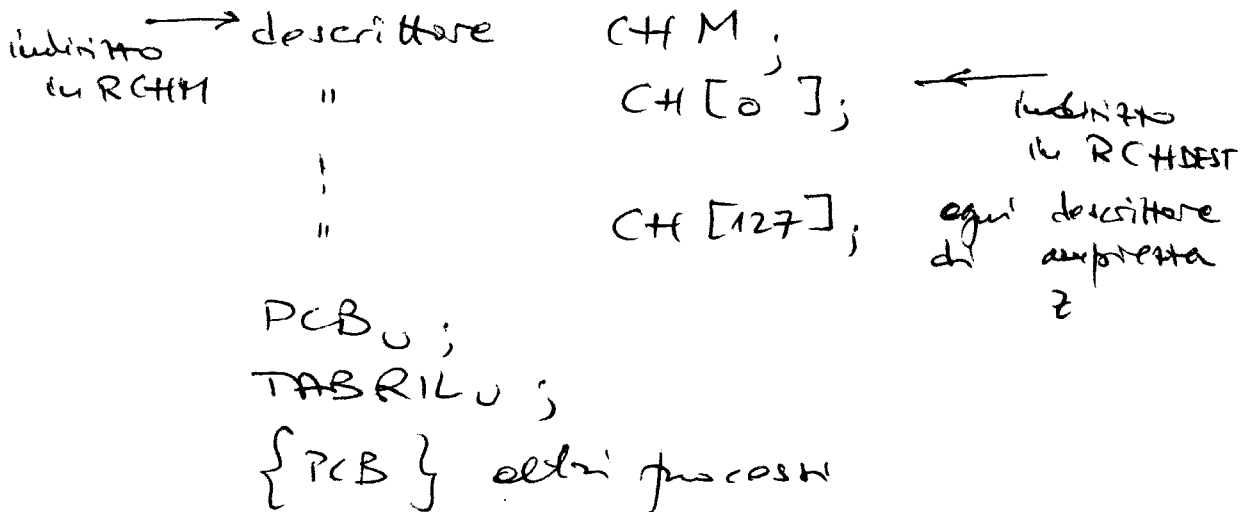
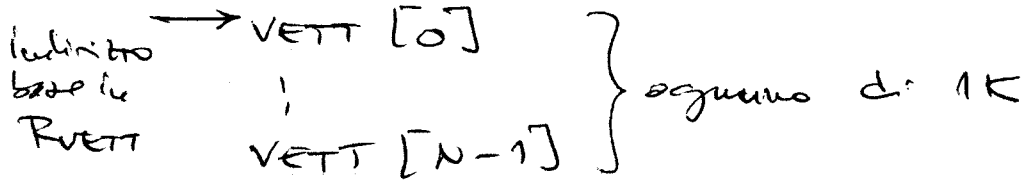
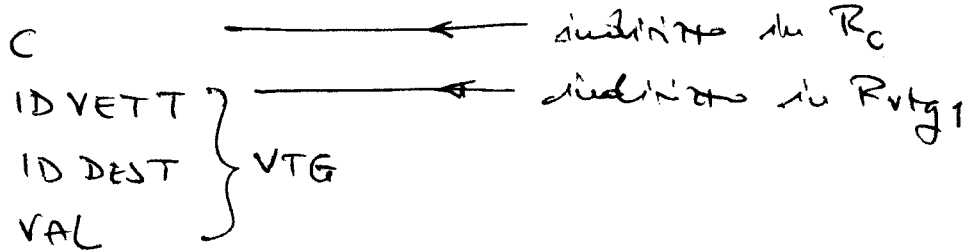
U :: channelin CH_M (1);
    channelout CH [N];
    int IDVETT, IDDEST, VAL; int VETT [N] [N*N];

    while true
    {
      receive (CH_M, (IDVETT, IDDEST, VAL));
      c = 0;
      for (i = 0; i < N; i++)
        if VETT [i] [IDVETT] < VAL
          { c = c + 1 };
      send (CH [IDDEST], c)
    }
  
```

MV<sub>0</sub> :

codice

codice send, receive, execution, autorisation



Registri generali contenenti i parametri della procedura (indirizzi):

R<sub>ch</sub>, R<sub>msg</sub>, R<sub>vtg</sub>

Altri registri inutilizzati:

R<sub>c</sub>, R<sub>i</sub>, R<sub>vett</sub>, R<sub>chm</sub>, R<sub>chdest</sub>  
 R<sub>d</sub>, R<sub>vtg1</sub>, R<sub>z</sub>

Compilazione di U:

```

INIT :  MOV RCHH, Rch      ] passaggio parametri
        MOV Rvga, Rvga    ] procedura "receive"
        CALL Rreceive, Ret
        LOAD Rvgt1, 0, RIDVETT ] parametri
        LOAD Rvga, 1, RIDDEST  ] di uscita
        LOAD Rvgt1, 2, RVAL     ] e registri
-----
        CLEAR Rcount
        MUL RN, RIDVETT, Rtemp
        ADD RVETT, Rtemp, Rbase
LOOP :  LOAD Rbase, Ri, Rtemp
        IF >= Rtemp, RVAL, CONT
        INCR Rcount
CONT :  INCR Ri
        IF < Ri, RN, LOOP
        STORE Rc, φ, Rcount
-----
        MUL RIDDEST, R2, Rtemp ] parametri
        ADD RCHDEST, Rtemp, Rch ] send
        MOV Rc, Rvgt1
        CALL Rsend, Ret
        GOTO INIT

```

La valutazione del tempo di completamento (dal testo: escluso il tempo di completamento delle procedure *send* e *receive*) non è riportata: vanno utilizzati i tempi medi di elaborazione delle varie istruzioni Risc, con  $t_c = 2\tau$ . La valutazione di  $T_{\text{faut}}$  è esattamente quella della Domanda 1.

b) Essendo le unità di I/O viste come processi esterni, il codice Lc rimane inalterato. Nell'implementazione del supporto, i descrittori di canale sono implementati in un supporto di memoria condivisa che, in ogni caso, può sempre essere realizzata nelle memorie di I/O e Memory Mapped I/O, oppure, se le unità di I/O dispongono di DMA, in memoria principale.

La differenza nell'implementazione della primitiva *send* è nella fase di sveglia: poiché i processi esterni effettuano attesa attiva, invece che operare sulle liste pronti la sveglia deve essere segnalata esplicitamente dal processore mediante una semplice istruzione di STORE nello spazio di I/O dell'unità che esegue il processo esterno destinatario.