

Architettura degli Elaboratori – A. A. 2017-2081
Prima prova di verifica intermedia e appello straordinario – 3 novembre 2017

Indicare su tutti i fogli consegnati, in alto a destra, nome, cognome, numero di matricola, corso (A o B) e, in caso di partecipazione all'appello straordinario, "Appello straordinario". I risultati verranno pubblicati via web appena disponibili, insieme al calendario degli orali per gli studenti dell'appello straordinario.

Si consideri una unità U che è collegata ad U_1 , U_2 ed U_3 mediante collegamenti dedicati. Il collegamento con U_1 e quello con U_2 è in ingresso, l'interfaccia è dotata di indicatori a transizione di livello e permette la trasmissione di una singola parola da 32 bit. U_3 interagisce con U secondo un protocollo a domanda risposta: invia richieste (senza invio di dati) ad U e riceve come risposta una parola di 32 bit.

La parte operativa di U contiene una memoria M di 1024 parole che permette la lettura di una parola e la scrittura di un'altra parola nello stesso ciclo di clock.

La memoria è gestita con politica FIFO, per memorizzare informazioni provenienti da U_1 ed U_2 e per inviare informazioni ad U_3 . In particolare, U attende una parola da U_1 ed una da U_2 e memorizza in M , secondo la disciplina FIFO, il massimo dei due valori, l'altro viene scartato. Quando riceve una richiesta da U_3 , U trasmette una parola di M , prelevata sempre utilizzando una disciplina FIFO.

Si progetti U e se ne calcoli il ciclo di clock sapendo che una ALU ha un ritardo di $5t_p$ e che le porte logiche utilizzate per la costruzione di *tutti* i componenti di U hanno al massimo 6 ingressi.

Bozza di soluzione.

Organizziamo la coda FIFO come un vettore gestito in maniera circolare. Il puntatore INS indica la posizione di inserimento. Il puntatore ESTR, indica la posizione di estrazione. Un contatore NE indica il numero di elementi presenti nella coda. INS, ESTR ed NE sono inizializzati a 0. INS ed ESTR sono da 10 bit e tutte le operazioni che li interessano saranno a 10 bit, dunque realizzando operazioni il cui risultato è considerato modulo 1024. NE è da 11 bit. NE_0 (bit più significativo) sarà indicatore di coda piena ($1 \rightarrow$ piena), $OR(NE_m)$ (10 bit meno significativi) sarà indicatore di coda vuota ($0 \rightarrow$ vuota).

Le operazioni relative all'inserimento e all'estrazione avvengono in un unico ciclo di clock essendo necessarie:

- per l'inserimento di una parola X: $X \rightarrow M[INS]$, $(INS+1)\%1024 \rightarrow INS$, $(NE+1) \rightarrow NE$, oltre agli eventuali set e reset sugli indicatori a transizione di livello necessari ad implementare il corretto protocollo di comunicazione. L'inserzione potrà avvenire se e solo se $NE_0 = 0$.
- per l'estrazione di una parola Y: $M[ESTR] \rightarrow Y$, $(ESTR+1)\%1024 \rightarrow ESTR$, $(NE-1) \rightarrow NE$, oltre agli eventuali set e reset sugli indicatori a transizione di livello necessari ad implementare il corretto protocollo di comunicazione. L'estrazione potrà avvenire se e solo se $OR(NE_m) = 1$.

Nel seguito immagineremo di utilizzare $INS(X)$ e $ESTR(Y)$ come macro che rappresentano ne microoperazioni appena descritte.

L'automa di controllo ha un unico stato, nel quale:

- se si riceve una richiesta di estrazione con coda non vuota, si esegue l'estrazione e si ritorna allo stato iniziale (unico stato)
- se si ricevono una parola da U1 ed una parola da U2 e la coda non è piena, si inserisce il massimo fra le due parole e si ritorna nello stato iniziale
- se si ricevono una parola da U1 ed una parola da U2 e una richiesta da U3 distinguiamo tre casi:
 - la coda è vuota: il massimo delle due parole ricevute da U1 ed U2 viene passato ad U3 e la coda rimane vuota, si ritorna allo stato iniziale
 - la coda contiene almeno un elemento ma non è piena: si inserisce il massimo delle due parole ricevute da U1 ed U2 e si passa il più vecchio elemento in coda ad U3, aggiornando di conseguenza INS, ESTR ed NE, si ritorna allo stato iniziale
 - la coda è piena: si passa l'elemento più vecchio in coda ad U3 e si sostituisce con il massimo delle due parole ricevute da U1 ed U2, aggiornando INS ed ESTR, ma non NE, si ritorna allo stato iniziale.
- in tutti gli altri casi, si rimane nello stesso stato senza fare alcunché

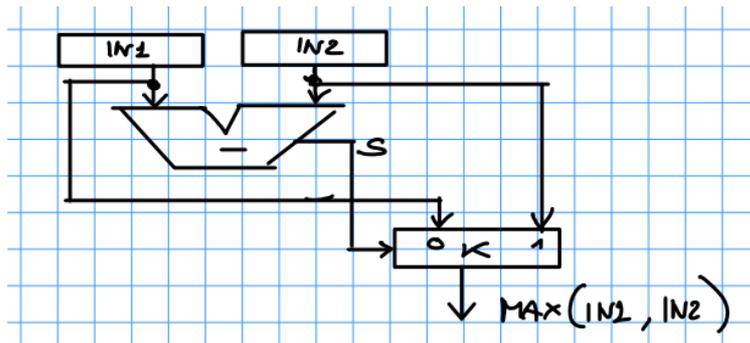
Le condizioni da testare sono dunque (assumiamo che la coppia di indicatori a transizione di livello $R_i A_i$ sia associata all'interfaccia con U_i e che R_i sia l'indicatore in ingresso ed A_i quello in uscita):

- $R_1, R_2, R_3, NE_0, OR(NE_m) = 001-1, 011-1, 101-1 \rightarrow$ estrazione
- $R_1, R_2, R_3, NE_0, OR(NE_m) = 1100- \rightarrow$ inserzione
- $R_1, R_2, R_3, NE_0, OR(NE_m) = 11100 \rightarrow$ passaggio del massimo dei valori ricevuti ad U3
- $R_1, R_2, R_3, NE_0, OR(NE_m) = 11101 \rightarrow$ inserzione+estrazione
- $R_1, R_2, R_3, NE_0, OR(NE_m) = 1111- \rightarrow$ inserzione + estrazione (stessa posizione nella memoria M)

Si assume di avere a disposizione una rete combinatoria che calcoli il $MAX(IN1, IN2)$, realizzata come segue:

- IN1 e IN2 sono ingressi di una ALU che ne fa la sottrazione e genera il flag S (segno)

- IN1 e IN2 sono anche ingressi di un commutatore il cui segnale di controllo è il bit S prodotto dalla ALU, come indicato nella figura che segue:



Sotto queste assunzioni, il microcodice di controllo potrebbe essere il seguente:

```

0. // sola richiesta di estrazione, coda non vuota
(R1,R2,R3,N0,or(N)=001-1,011-1,101-1) ESTR(OUT), set A3, reset R3, 0
// sola richiesta di inserzione, coda non piena
(R1,R2,R3,N0,or(N)=1100-) INS(max(IN1,IN2)),
set A1, set A2, reset R1, reset R2, 0
// inserzione ed estrazione con coda vuota
(R1,R2,R3,NE0,OR(NEm)=11100) max(IN1,IN2) → OUT,
set A1, set A2, set A3, reset R1, reset R2, reset R3, 0
// inserzione ed estrazione da coda non vuota e non piena
// e lo stesso per
// inserzione ed estrazione da coda piena
(R1,R2,R3,NE0,OR(NEm)=11101,1111-) max(IN1,IN2)→M[INS], (INS+1)%1024→INS,
M[ESTR]→OUT, (ESTR+1)%1024→ESTR,
set A1, set A2, set A3, reset R1, reset R2, reset R3, 0
// attesa se non ho richieste
// o non ho richieste di inserimento complete (da entrambe U1 e U2)
(R1,R2,R3,NE0,OR(NEm)=000--, 010--, 100--) nop, 0

```

Essendoci una sola microistruzione (un solo stato nell'automata di controllo) la PC sparisce come rete sequenziale. Rimane solo la ω PC come rete combinatoria parte della PO. Il ritardo introdotto dalla ω PC (rete con 5 ingressi e al uscite distinte per al massimo 5 frasi) sarà di $2t_p$.

Il ciclo di clock viene dunque calcolato come

$$\tau = T_{\omega PO} + T_{\omega PC} + T_{\sigma PO} + \delta$$

Le variabili di condizionamento richiedono al massimo $2t_p$ (calcolo di OR(NEm) con NEm da 10 bit: servono due livelli di porte OR da 6 bit).

Le risorse di stato della PO sono:

- INS, scritto sempre e solo con il risultato di una ALU che calcola l'incremento modulo 1024
- ESTR, idem
- NE, scritto sempre e solo con una ALU che calcola incremento o decremento
- M, scritta sempre all'indirizzo INS e letta dall'indirizzo ESTR, il valore in ingresso è sempre il $\max(IN1, IN2)$ calcolato come descritto nel seguito. Nel caso estrazione+inserzione è evidente la necessità di avere una memoria a doppia porta. Nel sottocaso =1111- la lettura e la scrittura avvengono nella stessa posizione, cosa possibile dal

momento che la lettura avviene sul fronte di discesa del ciclo di clock e la scrittura avviene al prossimo clock alto

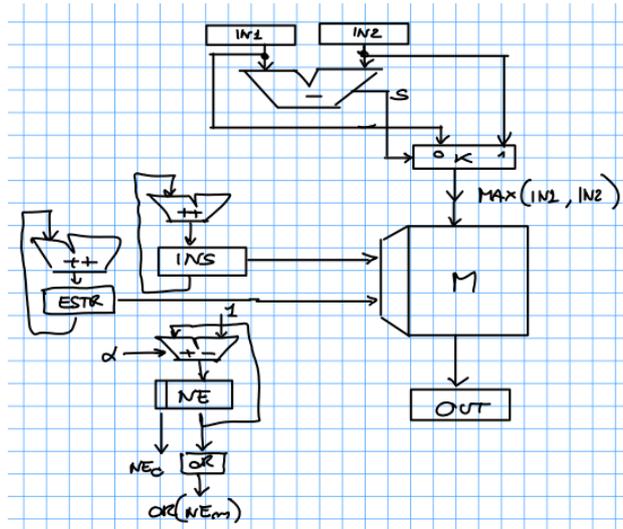
- gli indicatori a transizione di livello, su cui si opera esclusivamente con le operazioni di set e reset

Dunque non vi sono commutatori in ingresso alle risorse di stato.

Le risorse di calcolo nella PO sono:

- una ALU per incremento INS e ESTR, con operazioni a 10 bit (modulo 1024). Se si usa una sola ALU, occorre un commutatore in ingresso per scegliere fra INS ed ESTR. L'uscita è comunque ingresso sia di INS che di ESTR (scritture comandate da due β distinti, ad 1 in frasi diverse). Il ritardo introdotto con una ALU sola è $2t_p + 5t_p = 7t_p$ e con due ALU è $2t_p$.
- una ALU in grado di calcolare incremento o decremento con ingresso ed uscita da/su NE. Il ritardo introdotto è $5t_p$.
- la rete combinatoria che calcola $\max(IN1, IN2)$ costituita da una ALU che calcola $IN1 - IN2$, producendo il bit S (segno del risultato) che comanda un commutatore con IN1 e IN2 in ingresso e il bit S come ingresso di controllo. L'uscita del commutatore è il massimo fra I due registri. Il ritardo introdotto è $5t_p + 2t_p = 7t_p$.

La struttura della PO sarà dunque qualcosa del tipo:



Occorre calcolare il ritardo in scrittura della memoria M per poter valutare $T_{\sigma PO}$. La memoria ha 1K posizioni. Dunque il commutatore di lettura deve lavorare su $1K + 10$ ingressi. La tabella di verità sarà qualcosa del tipo:

i_0	i_1	...	i_9	w_0	w_1	...	w_{1024}	z
0	0		0	1	-		-	1
0	0		1	-	1	...	-	1
...								
1	1		1	-	-	1	1

dunque avrà 1024 righe e su ogni riga (corrispondente ad un 1 nella colonna delle uscite) avrà 11 bit specificati. Avendo a disposizione porte da 6 ingressi serviranno 2 livelli AND e $\lceil \log_6(1024) \rceil$ livelli di porte OR. Considerando che $6^3=216$, $6^4=1296$, serviranno 4 livelli di porte OR e dunque $t_a = 6t_p$.

Con questi risultati, la $T_{\sigma PO}$ sarà al massimo $t_{alu} + t_k + t_a$ (necessaria per calcolare il $\text{MAX}(\text{IN1}, \text{IN2}) \rightarrow \text{M}[\text{INS}]$) ovvero $5t_p + 2t_p + 6t_p = 13t_p$

Dunque il ciclo di clock sarà pari a:

$$\tau = T_{\omega PO} + T_{\omega PC} + T_{\sigma PO} + \delta = 2t_p + 2t_p + 13t_p + t_p = 18t_p$$