

Architettura degli elaboratori A.A. 2018-2019

Prima prova di verifica intermedia – 2 nov 2018

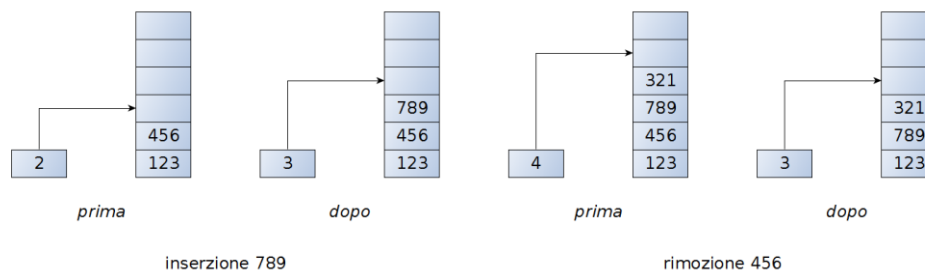
Compito Tipo A

Indicare in alto a destra su tutti i fogli ricevuti il proprio nome, cognome, numero di matricola e corso di appartenenza.

Domanda 1

Una unità firmware U mantiene una struttura dati che contiene elementi interi positivi e mette a disposizione le seguenti due operazioni: inserimento nella struttura dati, in fondo ai dati già presenti, rimozione di un elemento con un certo valore con compattazione dei dati rimanenti (vedi figura). U riceve richieste di operazione da 3 unità U_1 , U_2 e U_3 e le serve dando sempre priorità alla unità con indice maggiore. Le unità interagiscono a domanda risposta. U_1 e U_2 possono mandare solo operazioni di inserimento e U_3 solo operazioni di rimozione. Si assuma che la memoria abbia una capacità di 1K posizioni e che non possa accadere che venga richiesta una inserzione con la memoria già piena, né che possa essere richiesta la rimozione quando la memoria sia vuota. Può essere richiesta la rimozione di un elemento non presente. All'unità U_3 deve essere notificato l'esito dell'operazione: elemento rimosso o elemento non presente.

Si progetti l'unità U e se ne fornisca il tempo medio di elaborazione T in funzione di t_p sapendo che il 50% delle operazioni sono inserimenti e il 50% sono rimozioni, che le porte logiche che stabilizzano in $1t_p$ hanno al massimo 8 ingressi e che si possono utilizzare solo ALU che eseguono incrementi e decrementi di una unità in $5t_p$.



Domanda 2

Si dica se le seguenti affermazioni sono vere o false, dandone opportuna e sintetica motivazione:

1. Una memoria a doppia porta ha lo stesso tempo di accesso di una memoria standard della stessa capacità
2. Il numero di frasi in un microprogramma determina il numero di bit del registro di stato della parte controllo
3. La presenza di una variabile di condizionamento complessa (tipo segno(X-Y)) comporta necessariamente un aumento della lunghezza del ciclo di clock rispetto al microprogramma in cui $\text{segno}(X-Y) \rightarrow F$ viene calcolato in una microistruzione e alla microistruzione successiva si testa come variabile di condizionamento il flag F

Architettura degli elaboratori A.A. 2018-2019

Prima prova di verifica intermedia – 2 nov 2018

Compito Tipo B

Indicare in alto a destra su tutti i fogli ricevuti il proprio nome, cognome, numero di matricola e corso di appartenenza.

Domanda 1

Una unità U implementa una memoria di coppie $\langle \text{chiave}, \text{valore} \rangle$. Una unità U_1 può richiedere l'inserimento di una coppia $\langle K, V \rangle$. U cerca fra le proprie coppie e, quando trova la coppia con chiave pari a K, aggiorna il valore presente per quella chiave V_a con $V_a + V$. Si assume che non possano essere effettuate richieste per chiavi non esistenti. Due unità U_2 ed U_3 possono richiedere la lettura del valore relativo alla chiave K. La chiave può essere o non essere presente su U e la risposta diretta all'unità richiedente l'operazione deve indicare se la chiave è stata trovata (e con quale valore, in caso di lettura) o no. Tutte le unità interagiscono a domanda risposta. La priorità va sempre data alla unità di indice minore. La struttura dati che contiene le coppie è implementata utilizzando due moduli di memoria distinti: MK contiene alla posizione i la chiave della i-esima coppia ed MV contiene alla posizione i il valore della i-esima coppia.

107	765	107	765
201	889	201	889
12	1024	12	1024
23	129	23	129
7	250	7	262
5	270	5	270
11	231	11	231
3	111	3	111

MK	MV	MK	MV
----	----	----	----

prima dopo

inserimento(K=7,V=12)

Si progetti l'unità U e se ne fornisca il tempo medio di elaborazione in funzione di t_p sapendo che il 50% delle operazioni sono inserimenti e il 50% sono letture, che la memoria per le coppie ha una capacità di 1K coppie, che le porte logiche che stabilizzano in $1t_p$ hanno al massimo 8 ingressi e che sono disponibili solo ALU che fanno somme e incrementi/decrementi di una unità con un ritardo di stabilizzazione pari a $5t_p$.

Domanda 2

Si dica se le seguenti affermazioni sono vere o false, dandone opportuna e sintetica motivazione:

1. Un commutatore a k ingressi da n bit ciascuno ha lo stesso ritardo di un commutatore a k ingressi di 2n bit ciascuno
2. Il numero di frasi in un microprogramma determina la complessità del livello OR della σPC
3. Sostituire una microistruzione tipo
j. $(\text{segno}(X-Y) = 0) A+I \rightarrow C, j+1, (=1) B+I \rightarrow C, j+1$
con la microistruzione
j. $A \rightarrow C \mid_{\text{segno}(X-Y)=0}, B \rightarrow C \mid_{\text{segno}(X-Y)=1}, j+1$
(con la notazione $op \mid_{\text{cond}}$ indichiamo l'uso di controllo residuo per effettuare op se e solo se vale cond) comporta necessariamente una riduzione della lunghezza del ciclo di clock

Traccia di soluzione – Compito tipo A

Domanda 1

Utilizziamo la memoria M per mantenere i valori e TESTA per indicare la prima posizione libera. RDY_i (i = 1, 2 o 3) ci dice se è stata richiesta un'operazione dall'unità U_i. "uguale" è una rete combinatoria che ci dice se due registri sono identici. È costituita da n confrontatori in parallelo le cui uscite sono messe in OR. Il risultato dell'OR è negato per avere 1 (vero) quando tutti i bit corrispondenti nei due registri sono uguali. Il tempo di stabilizzazione della rete è 4 t_p (2t_p per i confrontatori, che lavorano tutti in parallelo, e 2t_p per l'albero di due livelli di porte OR che calcolano l'OR dei 32 bit risultato dei confrontatori).

L'interfaccia con le unità che richiedono l'inserimento di valori comprende, oltre agli indicatori a transizione di livello (uno in ingresso ed uno in uscita), un registro in ingresso VAL col valore da inserire nella struttura dati. L'interfaccia con l'unità che richiede l'estrazione comprende, oltre agli indicatori a transizione di livello (uno in ingresso ed uno in uscita), un registro in ingresso VAL col valore da rimuovere e un registro in uscita ESITO che comunica se il valore è stato effettivamente rimosso o se non è stato trovato nella struttura dati.

Il microcodice potrebbe essere il seguente:

- // se non ho richieste aspetto**
(RDY1, RDY2, RDY3 = 0 0 0) nop, 1
// richiesta di rimozione dall'unità a priorità più alta
(= - - 1) 0 → I, uguale (M [0], VAL3) → TROVATO, uguale (TESTA,1) → FINE, 0 → ESITO, 2.
// richiesta di inserimento da U2
(= - 1 0) VAL2 → M[TESTA], TESTA+1 → TESTA, reset RDY2, set ACK2, 1
// richiesta di inserimento da U1
(= 1 0 0) VAL1 → M[TESTA], TESTA+1 → TESTA, reset RDY1, set ACK1, 1
- // qui cerco il valore, se non l'ho trovato allora esito negativo (0)**
(TROVATO, FINE = 0 1) 0 → ESITO, set ACK3, reset RDY3, 1
// se l'ho trovato ed è l'ultima posizione, allora esito positivo e fine
(= 1 1) 1 → ESITO, set ACK3, reset RDY3, TESTA-1 → TESTA, 1.
// se l'ho trovato in una posizione intermedia comincio a compattare gli elementi restanti
(= 1 0) M[I+1] → M[I], I + 1 → I, uguale (TESTA, I+1) → FINE, 3.
// altrimenti scorro in avanti
(= 0 0) I+1 → I, uguale (TESTA, I+1) → FINE, uguale (M [I+1], VAL3) → TROVATO, 2.
- // se non sono alla fine continua**
(FINE = 0) M[I+1] → M[I], I + 1 → I, uguale (TESTA, I+1) → FINE, 3.
// altrimenti dai esito positivo e ritorna all'inizio
(=1) 1 → ESITO, set ACK3, reset RDY3, TESTA-1 → TESTA, 1

Le risorse di stato sono:

- TROVATO, registro da 1 bit, scritto sempre dall'uscita della rete combinatoria uguale (M[...], VAL)
- FINE, registro da 1 bit, scritto sempre dall'uscita della rete combinatoria uguale (TESTA,...)
- ESITO, scritto con la costante 0 o con la costante 1, dunque con un commutatore in ingresso
- M, scritta all'indirizzo TESTA o I (commutatore sull'ingresso indirizzo) con VAL o con M[I+1].
Serve una memoria a doppia porta per poter leggere la posizione I+1 e scrivere la posizione I

nella stessa micro operazione. Serve un commutatore sugli ingressi per scegliere se scrivere VAL o M[I+1]

- TESTA, registro da 10 bit che punta sempre alla prima posizione libera in memoria. Si scrive sempre con l'uscita di una ALU che implementa le operazioni incremento e decremento
- I, registro da 10 bit che si scrive con l'uscita di una ALU che esegue l'operazione di incremento o con la costante 0, dunque ha un commutatore in ingresso

Le risorse di calcolo necessarie sono:

- Due reti combinatorie UGUALE, descritte precedentemente. Ne servono due perché vengono utilizzate in due micro operazioni della stessa micro istruzione, quindi in parallelo
- Una ALU che fa incremento e decremento, utilizzata per calcolare TESTA+1 → TESTA nella microistruzione 1. e I+1 nelle diverse micro operazioni delle frasi della microistruzione 2. e 3. Pertanto, serve un commutatore sull'ingresso per scegliere se operare sul registro TESTA o sul registro I.

Abbiamo 5 variabili di condizionamento, 3 testate nella microistruzione 1., 2 testate nella micro istruzione 2. e una sola testata nella micro istruzione 3. Tutte sono o uscite di indicatori a transizione di livello o uscite di registri, dunque comportano un $T_{\omega PO}$ pari a 0 e la condizione di correttezza è rispettata (PO di Moore).

La PC ha due bit nel registro di stato interno (per rappresentare quale fra le 3 micro istruzioni del microprogramma è quella corrente) e un massimo di 3 variabili di condizionamento testate contemporaneamente. Le frasi del microprogramma sono 10. Dunque, servono un unico livello di porte AND e due livelli di porte OR (massimo $10-1=9$ ingressi), il che porta il ritardo di σPC e ωPC a $3tp$.

La σPO più lunga richiede la stabilizzazione di un commutatore (ingresso indirizzi della memoria), del commutatore di lettura della memoria da 1K, e del commutatore di selezione dell'ingresso da scrivere in memoria (il selettore di scrittura stabilizza in parallelo al commutatore di lettura), quindi

$$2tk + ta = 2(2tp) + (2tp + 4tp) = 10tp$$

Dunque, occorre un ciclo di clock pari ad almeno

$$\tau = 0 + \max\{3tp, 3tp + 10tp\} + tp = 14tp$$

L'inserzione richiede un'unica microistruzione ($k_{\text{inserimento}} = 1$). La rimozione (considerando una media di $K/2$ elementi nella memoria, ne richiederà $1 + K/2 + 1$ ($k_{\text{rimozione}} = 2 + K/2$). Il tempo medio di elaborazione sarà quindi

$$T = \tau/2 + \tau/2 + \tau K/2 = \tau + 1024 \tau / 2 = 513 \tau$$

Domanda 2

1. Vero, i due commutatori in lettura operano in parallelo
2. Falso, il numero è determinato unicamente dal numero di micro istruzioni
3. Falso, dipende dal resto del programma. Potrebbe essere vero se questa fosse l'istruzione che richiede il ciclo di clock più lungo di tutto il microcodice.

Traccia di soluzione – Compito tipo B

Domanda 1

Utilizziamo due memorie MV e MK per contenere, alla posizione i , il valore e la chiave della coppia i -esima. “uguale” è una rete combinatoria che ci dice se due registri sono identici. È costituita da n confrontatori in parallelo le cui uscite sono messe in OR. Il risultato dell’OR è negato per avere 1 (vero) quando tutti i bit corrispondenti nei due registri sono uguali. Il tempo di stabilizzazione della rete è 4τ (2τ per i confrontatori, che lavorano tutti in parallelo, e 2τ per l’albero di due livelli di porte OR che calcolano l’OR dei 32 bit risultato dei confrontatori).

L’interfaccia con l’unità che richiede l’inserimento è composta, oltre che ai due indicatori a transizione di livello in ingresso ed in uscita, da un registro in ingresso K , che contiene la chiave per la quale si vuole effettuare l’inserimento, un registro in ingresso V , che contiene il valore da utilizzare per l’aggiornamento della coppia con chiave K e un registro in uscita ESITO, che verrà utilizzato per comunicare se la chiave è stata trovata (e quindi l’aggiornamento è avvenuto) o no.

L’interfaccia con le unità che richiedono la lettura, è simile a quella della unità che richiede l’inserimento, tranne per il fatto che non è presente il registro V .

Il microcodice potrebbe essere scritto come segue:

- // nessuna richiesta, ne attendo una*
(RDY1, RDY2, RDY3 = 0 0 0) nop, 1
// richiesta di inserimento dalla unità a priorità più alta,
// inizializzo i contatori per la ricerca della chiave
(=1 - -) 0 → I, uguale (MK [0], K1) → TROVATO, 2.
// idem per la unità con la seconda priorità più alta
(=0 1 -) 0 → I, uguale (MK [0], K2) → TROVATO, 3.
// richiesta di lettura, dall’ultima unità,
// inizializzo i contatori per la ricerca della chiave
(=0 0 1) 0 → I, uguale (MK [0], K3) → TROVATO, 4.
- // fine memoria, chiave non trovata*
(I0, TROVATO = 1 0) 0 → ESITO1, set ACK1, reset RDY1, 1
// chiave trovata, aggiorna il valore
(=0 1) MV[I] + V → MV[I], 1 → ESITO1, set ACK1, reset RDY1, 1
// chiave non trovata, non sono alla fine, continua alla prossima coppia
(=0 0) I+1 → I, uguale (MK [I+1], K1) → TROVATO, 2
- // fine memoria, chiave non trovata*
(I0, TROVATO = 1 0) 0 → ESITO2, set ACK2, reset RDY2, 1
// chiave trovata, aggiorna il valore
(=0 1) MV[I] → OUT2, 1 → ESITO2, set ACK2, reset RDY2, 1
// chiave non trovata, non sono alla fine, continua alla prossima coppia
(= 0 0) I+1 → I, uguale (MK [I+1], K2) → TROVATO, 3
- // fine memoria, chiave non trovata, esito negativo*
(I0, TROVATO = 1 0) 0 → ESITO3, set ACK3, reset RDY3, 1
// chiave trovata, comunica il valore, esito positivo
(=0 1) MV[I] → OUT3, 1 → ESITO3, set ACK3, reset RDY3, 1
// chiave non trovata, non sono alla fine, continua alla prossima coppia
(= 0 0) I+1 → I, uguale (MK [I+1], K3) → TROVATO, 4

Abbiamo cinque variabili di condizionamento, che sono tutte uscite di registri o di indicatori in ingresso a transizione di livello, La condizione di correttezza è rispettata (PO di Moore). Al massimo vengono testate 3 variabili (nella prima microistruzione) contemporaneamente. $T_{\omega PO}$ è pari a 0.

Le risorse di stato sono le seguenti:

- MK (in sola lettura) e MV (scritta con l'uscita di una ALU)
- TROVATO, registro da 1 bit scritto mediante l'uscita della rete combinatoria UGUALE
- ESITO1, ESITO2, ESITO3, scritti con costanti (0 o 1)
- OUT1 e OUT2, scritti con l'uscita dalla memoria MV

La memoria MK viene letta con indirizzi 0 e I+1 (uscita di una ALU) dunque necessita di un commutatore sull'ingresso indirizzi. La memoria MV viene letta e scritta con il solo indirizzo I e non necessita di commutatori né sull'ingresso indirizzi (sempre I) né su quello dati (sempre uscita della ALU che somma il valore in ingresso al contenuto precedente della cella di memoria),

Le risorse di calcolo comprendono:

- La rete combinatoria UGUALE, che risponde 1 quando i due registri in ingresso sono uguali, come descritto precedentemente
- Una ALU per il calcolo del nuovo valore da inserire nella coppia con chiave K e per effettuare gli incrementi del contatore I. Tale ALU avrà dei commutatori sugli ingressi per poter effettuare, in frasi diverse, le operazioni I+1 e $MV[I] + V$

La parte controllo ha cinque ingressi (variabili di condizionamento) e 2 bit di stato (4 microistruzioni). Il livello AND delle σPC e ωPC sarà quindi uno solo, avendo al massimo 5 ingressi (2 bit di stato + 3 (max) variabili di condizionamento testate contemporaneamente). Essendo presenti 13 frasi, il livello OR sarà costituito da 2 livelli di porte. Il ritardo complessivo della PC sarà quindi $T_{\sigma PC} = T_{\omega PC} = 3tp$.

Per $T_{\sigma PO}$ consideriamo la micro operazione più lunga, che potrebbe essere o la uguale (MK [I+1], K) \rightarrow TROVATO o la $MV[I] + V \rightarrow MV[I]$. La prima delle due operazioni richiede 4tp per il calcolo di UGUALE. La seconda, richiede $t_a + t_{al}$. t_{al} è già più alto di t_{ug} , essendo 5tp. Il tempo di accesso alla memoria richiede la stabilizzazione di un commutatore a 1024 ingressi: 10 + 1 ingressi per il livello AND (i bit di indirizzo e il bit dell'ingresso relativo, gli altri sono don't care => due livelli di porte AND) e 1024 termini per il livello OR (i 1024 "1" corrispondenti alle 1024 righe della tabella di verità => parte intera superiore di $\log_8(1024) = 4$ livelli di porte OR), dunque 6tp. In totale la $MV[I] + V \rightarrow MV[I]$ richiede quindi 11tp (il selettore di scrittura stabilizza durante la stabilizzazione del commutatore di lettura).

Quindi per la seconda frase delle microistruzioni 2. e 3. occorreranno per σPO 11tp. Questo porta ad aver bisogno di un ciclo di clock pari a

$$\tau = 0 + \max\{3tp, 3tp+11tp\} + tp = 15tp$$

Supponiamo che l'operazione di inserimento o estrazione trovi la chiave in media dopo aver controllato $1K/2 = 512$ posizioni. Se non la trova, occorrerà invece scorrere tutta la memoria, quindi 1024 posizioni. Nel primo caso occorre considerare l'esecuzione:

- Della microistruzione 1. (1τ)
- Di 512 esecuzioni della microistruzione 2. 3. o 4. (512τ)

Per un totale di 513τ .

Visto che le due operazioni esterne richiedono lo stesso numero di microistruzioni, questo sarà anche il tempo medio di elaborazione, indipendentemente dalle percentuali di operazioni dei due tipi effettivamente richieste.

Domanda 2

1. Vero, si utilizzando il doppio dei commutatori da k ingressi da 1 bit ciascuno, ma lavorano tutti in parallelo
2. Vero, ogni frase richiede una riga nella tabella di verità sia delle σPC che della ωPC , essendo la PC una rete di Mealy
3. Falso, dipende dal resto del micro codice. Se fosse l'istruzione che richiede il ciclo di clock più lungo allora potrebbe essere vero