

Architettura degli elaboratori—A. A. 2018-19—Appello 20 giugno 2019

Riportare in alto a destra di ogni foglio consegnato Nome, Cognome, Matricola e corso di appartenenza (A o B). La traccia di correzione, i risultati e il calendario degli orali saranno resi noti via web (didawiki e/o pagine docenti) appena disponibili.

Domanda 1

Una unità firmware U è interfacciata con una unità U_a mediante un'interfaccia che comprende:

- 1 registro in ingresso IN da 32 bit
- un registro in ingresso OP da 1 bit
- una coppia di indicatori a transizione di livello.

U riceve da U_a una coppia N (nel registro IN, $N > 1$) e OP (OP=0 rappresenta "+" e OP=1 rappresenta "*") e successivamente, mediante il registro IN, riceve N valori interi x_1, \dots, x_N . L'unità calcola quindi $x_1 \text{ OP } x_2 \text{ OP } \dots \text{ OP } x_N$ controllando che non vi sia overflow. Se l'operazione si conclude senza overflow, il risultato viene inviato ad una seconda unità U_b , insieme ad un ESITO = 0. Se viceversa si ha un overflow, all'unità U_b viene semplicemente mandato ESITO = 1. Anche in caso di overflow, l'unità U accetta comunque l'intera serie di N valori da U_a .

Si realizzi l'unità firmware avendo cura di minimizzare la lunghezza del ciclo di clock. La minimizzazione di τ deve essere adeguatamente commentata. Le porte logiche si stabilizzano in $1t_p$ ed hanno al massimo 8 ingressi.

Domanda 2

Un automa riceve in ingresso sequenze di interi da 4 bit e restituisce in uscita un bit che vale uno tutte le volte che viene riconosciuta la sotto sequenza formata dai valori 3 6 9 (rappresentati sui 4 bit a disposizione), e zero in qualunque altro caso. Ad esempio, se la sequenza è: 4 3 6 8 2 3 3 6 9 3 6 9 1 3 9 6 5 5 ... l'uscita sarà 0 0 0 0 0 0 0 1 0 0 1 0 0 0 ...

Si fornisca la descrizione completa di una rete sequenziale di **Mealy** che implementa l'automa.

Domanda 3

Si fornisca il codice D-RISC per una funzione F che, ricevuti come parametri A e B (interi da 32 bit, $A \neq 0$), calcola sia il risultato divisione intera, A/B , sia il resto della divisione stessa, $A\%B$ utilizzando **esclusivamente** operazioni aritmetico logiche **corte**. Lo pseudo codice della funzione è il seguente:

```
F(int A, int B) {
    R = A;
    Q = 0;
    while(R > B) {
        Q++;
        R -= B;
    }
    return (R, Q);
}
```

Se ne valutino le prestazioni su un processore D-RISC pipeline **senza** unità EU slave.

Traccia di soluzione

Domanda 1

Microcodice

1. // attendo che mi si mandi una richiesta da Ua
(RDYin = 0) nop, 1.
// se mi arriva, copio i parametri in registri locali e mi predispongo a ricevere gli Xi
(=1) IN -> N, OP -> MYOP, reset RDYin, set ACKin, 2.
2. // ricevo il primo valore, serve per inizializzare l'accumulatore
(RDYin=0) nop, 2.
// quando lo ricevo inizializzo accumulatore, flag di overflow e conto un elemento ricevuto
(=1) IN -> ACC, 0 -> OW, N - 1 -> N, 3.
3. // se ho già ricevuto l'ultimo elemento, no overflow, disponibilità a ricevere per Ub
(NO, RDYin, ACKout, MYOP, OW = 1 - 1 - 0) ACC -> OUT, 0 -> ESITO, set RDYout, reset ACKout, 0.
// stesse condizioni ma con overflow
(= 1 - 1 - 1) 1 -> ESITO, set RDYout, reset ACKout, 0.
// fine lavori ma non c'è disponibilità a ricevere da Ub, attendo
(= 1 - 0 - -) nop, 3.
// non ho finito, aspetto se non c'è diponibilità dati in ingresso
(= 0 0 - - -) nop, 3.
// ulteriore dato in ingresso disponibile, operazione somma
(=0 1 - 0 -) ACC + IN -> ACC, OW(ACC + IN) OR OW -> OW, N - 1 -> N, set ACKin, reset RDYin, 3.
// idem, moltiplicazione
(=0 1 - 0 -) ACC * IN -> ACC, OW(ACC * IN) OR OW -> OW, N - 1 -> N, set ACKin, reset RDYin, 3.

Minimo τ :

- $T_{\omega PO}$ nullo (solo flag o indicatori a transizione di livello)
- 4. $T_{\sigma PO}$ al massimo $T_{alu} + T_k$. La ALU mi serve per accumulare i valori ricevuti. Si potrebbe tentare di rimuovere il commutatore di fronte al registro scritto con la ALU (ACC). A tale scopo potremmo modificare la seconda frase della 2. come segue
(= 1) ACC + IN -> ACC, N - 1 -> N, 3.
giocando sul fatto che ACC all'accensione vale 0 (come tutti i registri, quindi togliamo anche 0 -> OW). Tuttavia,
- 5. rimane la scrittura in N che richiede un commutatore dopo la ALU. Nella seconda frase delle 1. non possiamo mettere una IN - 0 -> N altrimenti poi ci servirebbero commutatori sugli ingressi della ALU che scrive N.
- PC con 4 stati e 5 var cond (7 ingressi al livello AND) e 10 frasi, di cui 4 frasi nop, che per definizione hanno tutti i β a 0 e gli α non specificati, quindi che possono essere messi anche loro a 0. Ciò implica che il numero massimo di "1" per ciascuna delle α o β sono al più 6 e quindi basta un unico livello OR. Il ritardo introdotto da $T_{\omega PO}$ e $T_{\sigma PO}$ sarà quindi minimo (2tp).

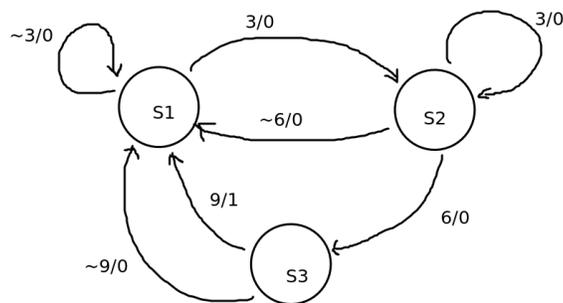
Ottimizzazione codice.

- Le ultime due frasi della 3. possono essere implementate con una frase sola utilizzando il controllo residuo. La parte che calcola il nuovo valore dell'accumulatore e l'OR del vecchio e nuovo flag OW si può realizzare con una ALU che fa somme e moltiplicazioni il cui α sia preso da MYOP (lettura da registro, non introduce ritardo nell'operazione che rimane un'operazione con ritardo $T_{alu} + T_k$). Diminuisco il numero delle variabili di condizionamento, ma questo non impatta sui ritardi della PC, che sono già minimi.

Domanda 2

L'automa ha tre stati:

- Stato iniziale. Se ricevo un 3 vado in 2. con uscita 0, altrimenti rimango in 1. con uscita 0
- Se ricevo un 6 vado in 3. con uscita 0, altrimenti vado in 1. con uscita 0
- Se ricevo un 9 vado in 1. con uscita 1, altrimenti ci vado con uscita 0.



Codifico S1 con 00, S2 con 01 e S3 con 10. I bit degli ingressi siano $x_1 \dots x_4$. La tabella di verità per σ e ω della rete di Mealy che implementa l'automa sarà data da:

S0	S1	x_1	x_2	x_3	x_4	S0'	S1'	z
0	0	0	0	1	1	0	1	0
0	0	1	-	-	-	0	0	0
0	0	-	1	-	-	0	0	0
0	0	-	-	0	-	0	0	0
0	0	-	-	-	0	0	0	0
0	1	0	0	1	1	0	1	0
0	1	0	1	1	0	1	0	0
0	1	1	-	-	-	0	0	0
0	1	-	0	-	-	0	0	0
0	1	-	-	0	1	0	0	0
0	1	-	-	1	0	0	0	0
1	0	1	0	0	1	0	0	1
1	0	0	-	-	-	0	0	0
1	0	-	1	-	-	0	0	0
1	0	-	-	1	-	0	0	0
1	0	-	-	-	0	0	0	0

Per ogni stato, una riga è per il riconoscimento della condizione che riconosce il carattere nell'ordine considerato. Questa riga è completamente specificata e dà come stato al $t+1$ il prossimo stato e come uscita 0 nei primi due e 1 nel terzo stato. Le altre 4 righe sono quelle minimamente specificate che esprimono una combinazione diversa da quella attesa per l'ingresso da riconoscere. Le colonne delle uscite hanno tutte un solo "1", tranne la colonna $S1'$, quindi serve un livello di porte OR. Gli ingressi del livello AND sono al più 6. Le due reti combinatorie σ e ω della rete di Mealy stabilizzano quindi in un $2tp$. Quindi $\tau = 3 tp$, considerando anche il tempo del clock alto. Il registro di stato è da 2 bit. Questo specifica completamente la rete sequenziale di Mealy che implementa l'automa.

La formulazione delle due reti combinatorie della ω e della singola rete della σ è data dalle somme di prodotti:

$$S0' = \sim S0 \& S1 \& \sim x1 \& x2 \& x3 \& \sim x4$$

$$S1' = \sim S0 \& \sim S1 \& \sim x1 \& \sim x2 \& x3 \& x4 + \sim S0 \& S1 \& \sim x1 \& \sim x2 \& x3 \& x4$$

$$Z = S0 \& \sim S1 \& x1 \& \sim x2 \& \sim x3 \& x4$$

Domanda 3

Compilazione in Assembler D-RISC

```
f:      mov Ra, Rr
        clear Rq
loop:   if<= Rr, Rb, fine
        inc Rq
        sub Rr, Rb, Rr
        goto loop
fine:   goto Rret
```

L'inizializzazione prende $2t$ (mov Ra, Rr è in realtà una add R0, Ra, Rr e clear Rq una add R0, R0, Rq).

Un ciclo eseguito per intero (no uscita) costa $5t$ ($4t$ per le istruzioni, che non hanno dipendenze, più una bolla da $1t$ da salto preso per il goto che torna a testare la condizione). L'ultimo ciclo costa un salto preso per $\text{if}<=$ ($2t$) più il salto per il ritorno da procedura (altri $2t$), purchè abbiamo avuto l'accortezza di allocare Rr ed Rq sui registri previsti per il ritorno dalla procedura. Dunque, una divisione con risultato n costa $2t + n(5t) + 4t$ a fronte di un numero di istruzioni pari a $2 + 4n + 2$. Per n grandi, l'efficienza è circa $4/5$ e il tempo di servizio è $5t/4$.