

**SOLUZIONI DELLA PROVA SCRITTA DEL CORSO DI
CALCOLATORI ELETTRONICI
NUOVO E VECCHIO ORDINAMENTO DIDATTICO**
17 Febbraio 2005

MOTIVARE IN MANIERA CHIARA LE SOLUZIONI PROPOSTE A CIASCUNO DEGLI ESERCIZI SVOLTI

ESERCIZIO 1 (NO: 8 punti – VO: 7 punti)

- (a) (NO: 8 punti - VO: 5 punti) Implementare un contatore modulo 3. In altri termini, dato un ingresso X , il contatore presenta un'uscita che assume ciclicamente i valori da 0 a 2, con incremento di una unità ogni volta che $X=1$. Si utilizzino flip flop T per la memorizzazione dello stato, e il metodo delle mappe di Karnaugh per minimizzare le espressioni booleane delle variabili di stato e di uscita.
- (b) (solo VO: 2 punti) Realizzare con cinque flip flop D un registro a scorrimento.

Soluzione.

Domanda (a).

Questa rete logica sequenziale richiede una coppia di bit per la rappresentazione dell'uscita che assume ciclicamente le configurazioni 00, 01, 10.

Diagramma degli stati:

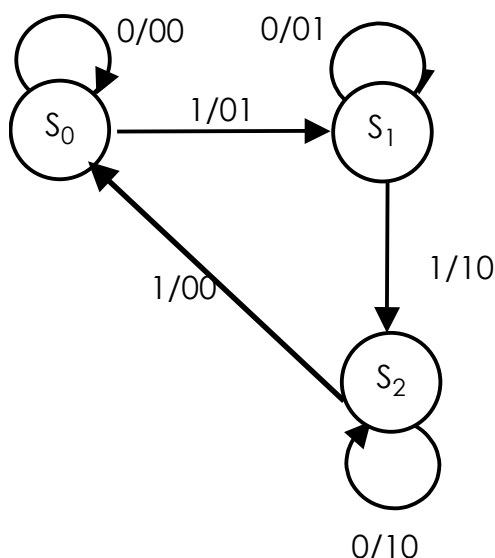


Tabella di flusso

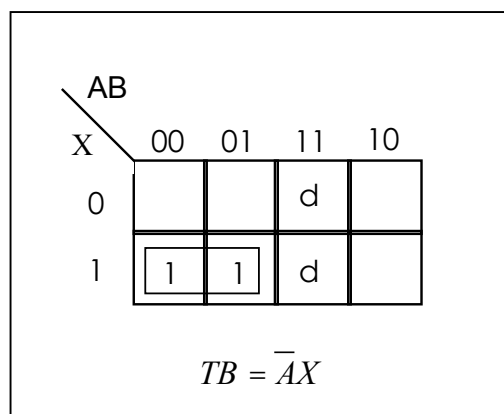
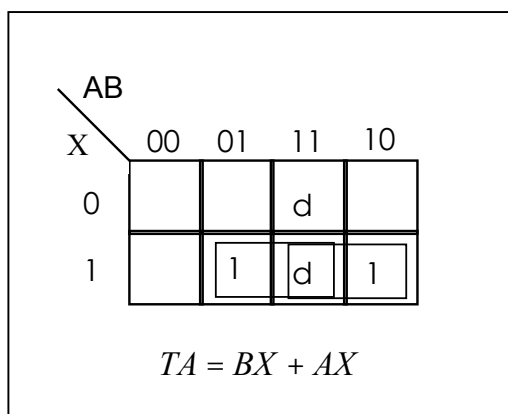
	S'/Z	
S	X=0	X=1
S0	S0/00	S1/01
S1	S1/01	S2/10
S2	S2/10	S0/00

E' facile notare che, codificando le variabili di stato con le configurazioni $S0 = 00$, $S1 = 01$, $S2 = 10$, è possibile utilizzare tali configurazioni anche come variabili di uscita.

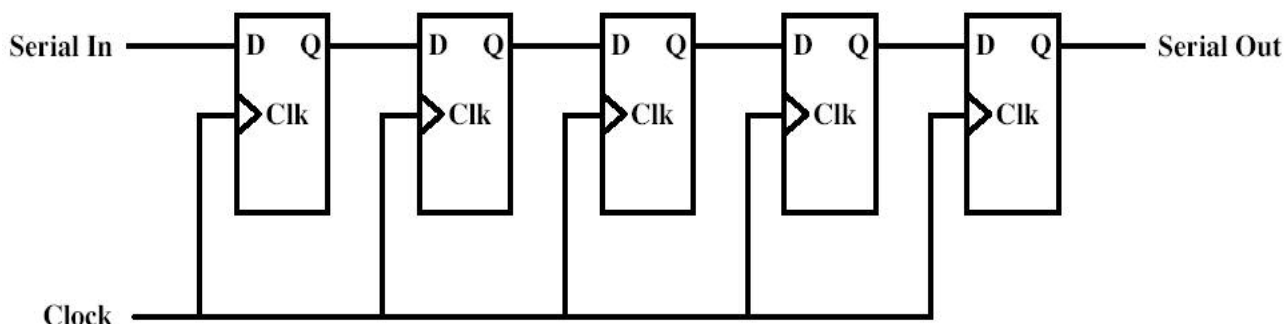
A questo punto scrivere la tabella delle transizioni è semplice. Indicando con A il bit più significativo dello stato, e con B l'altro, si ottiene:

A	B	X				
			A'	TA	B'	TB
0	0	0	0	0	0	0
0	0	1	0	0	1	1
0	1	0	0	0	1	0
0	1	1	1	1	0	1
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	D	D	D	D
1	1	1	D	D	D	D

Le espressioni booleane di TA e TB sono minimizzabili con le mappe di Karnaugh:



Domanda (b):



ESERCIZIO 2 (NO: 9 punti – VO: 8 punti)

Si consideri un calcolatore che dispone di una memoria principale di 256 Mbyte e di una memoria cache di 512 Kbyte. E' possibile accedere al singolo byte e la memoria è suddivisa in blocchi da 16 byte.

- (2 punti) Spiegare, precisando il significato e la funzione dei diversi campi, come vengono interpretati gli indirizzi logici per recuperare l'informazione contenuta nella cache nel caso venga usata la modalità di indirizzamento
 - Diretto
 - "associativo su insiemi", e ciascun insieme contenga due blocchi
- (VO: 6 punti - NO: 7 punti) Si consideri la cache di cui alla domanda precedente. Ipotezzare che il processore acceda in sequenza ai byte dall'indirizzo 0000000 a 00007FF e da 0180000 a 01807FF in questo ordine, e ripeta questa sequenza di accesso per 5 volte consecutive. Si ipotizzi inoltre che la cache sia inizialmente vuota. Calcolare il numero di miss sia nel caso di modalità ad indirizzamento diretto sia nel caso di modalità ad indirizzamento associativo su insiemi spiegata nel punto precedente. Calcolare quindi l'hit ratio.

Soluzione:

- Per indirizzare 256 Mbyte occorre un indirizzo di almeno 28 bit. Per indirizzare il singolo byte all'interno di un blocco occorrono 4 bit ($16 = 2^4$), che coincidono con i 4 bit meno significativi dell'indirizzo di memoria primaria. I restanti 24 bit costituiscono l'indirizzo del "block frame". Per indirizzare la cache, il "block frame" viene interpretato diversamente a seconda che l'indirizzamento sia di tipo "diretto" o "associativo su insiemi".
 - Indirizzamento diretto. In questo caso devo poter indirizzare ciascuno dei 32K blocchi contenuti nella cache ($512\text{Kbyte}/(16\text{byte}/\text{blocco})$). Occorrono 15 bit che coincidono con i 15 bit meno significativi del "block frame". Pertanto i 28 bit di indirizzo della memoria primaria vengono interpretati come:

tag	cache index	Offset
9 bit	15 bit	4 bit

(b) Indirizzamento "associativo su insiemi". In questo caso devo poter indirizzare ciascuno dei 16 insiemi in cui sono suddivisi i blocchi contenuti nella cache ($\frac{512\text{Kbyte}}{\frac{2\text{blocchi}}{\text{insieme}} \cdot \frac{16\text{byte}}{\text{blocco}}} = 16\text{K}$ insiemi). Occorrono 14 bit che coincidono con i 14 bit meno significativi del "block frame"

Pertanto i 28 bit di indirizzo della memoria primaria vengono interpretati come:

tag	cache index	offset
10 bit	14 bit	4 bit

- 2) La memoria è divisa in blocchi di 16 byte ciascuno in modo che la richiesta di un dato non presente in cache causa il trasferimento del blocco a cui appartiene il dato richiesto dalla memoria principale alla cache. Nel caso proposto i dati richiesti sono così suddivisi (N.B. per semplicità i valori di index sono riportati in formato decimale, mentre gli indirizzi in esadecimale):

indirizzi 0000000 ÷ 000000F ⇒ index 0; indirizzi 0000010 ÷ 000001F ⇒ index 1; ... ; 00007F0 ÷ 00007FF ⇒ index 127; 0180000 a 018000F ⇒ index 0; ... ; 01807F0 a 01807FF ⇒ index 127. (N.B. in questa sequenza l'index è identico nel caso di indirizzamento diretto e associativo su insiemi).

La prima volta che viene richiesta la parola 0000000 avremo un "cache miss" che provoca il caricamento del blocco 0 nella linea di cache di indirizzo 0. Le successive richieste dei dati di indirizzo 0000001 ÷ 000000F vengono quindi soddisfatte dalla cache ("cache hit"). Analogamente avviene per tutti i blocchi fino al blocco 127, che vengono allocati in linee consecutive della cache fino alla 127. Quando viene richiesta la parola 0180000 (index 0), nel caso di indirizzamento diretto questa sovrascrive il blocco 0000000 ÷ 000000F e così via per tutte le richieste consecutive. Nel caso di indirizzamento associativo su insiemi a due vie uno stesso index corrisponde a un insieme di due blocchi. Pertanto le parole da 0180000 a 01807FF non sovrascrivono quelle precedentemente inserite, ma vengono memorizzate nel secondo blocco disponibile in ciascun insieme memorizzato.

Nei cicli successivi al primo il processore richiede nuovamente tutti i dati, a partire da 0. Nel caso di indirizzamento diretto avremo un miss per il caricamento della prima parola di ciascun blocco e 15 hit per le parole dello stesso blocco, per tutti i 256 blocchi. Nel caso di indirizzamento associativo su insiemi si hanno solo hit perché tutti blocchi sono presenti in cache.

In sintesi:

- nel caso di indirizzamento diretto avremo per ciascun ciclo 256 miss, quindi 1280 miss in totale (256*5);
- nel caso di indirizzamento associativo su insiemi a due vie avremo soltanto 256 miss in totale, tutti relativi al primo ciclo.

Poiché il numero totale di accessi è sempre 256*5*16, si ricava facilmente l'hit ratio H dato che:

$$H = 1 - \frac{\text{numeroMiss}}{\text{numeroAccessi}}$$

Quindi $H(\text{diretto})=0,9375$; $H(\text{associativo su insiemi})=0,9875$.

ESERCIZIO 3 (solo NO: 8 punti)

Si scriva il codice Assembler MIPS di una funzione che, dati tre vettori di N interi u , v , w , scriva nella posizione $w(i)$ il valore $u(i)+v(i)$, se $u(i)<v(i)$, e il valore $u(i)-v(i)$, altrimenti. Si consideri che gli indirizzi iniziali dei vettori u , v , w siano memorizzati in \$4, \$5, \$6, rispettivamente, e che N sia memorizzato in \$7.

In altri termini il codice MIPS può implementare la funzione C:

```
void elabora(int *u, int *v, int *w, int N)
{
    int i;
    for (i=0; i<N; i++)
        if (u(i)<v(i))
            w(i)=u(i)+v(i);
        else
            w(i)=u(i)-v(i);
}
```

Soluzione.
 $\$8 \leftarrow i; \$9 \leftarrow u(i) < v(i)$
 $\$10 \leftarrow u(i); \$11 \leftarrow v(i); \leftarrow \$12 \leftarrow w(i)$

```

elabora:    addi $29, $29, -20
            sw $8, 0($29)
            sw $9, 4($29)
            sw $10, 8($29)
            sw $11, 12($29)
            sw $12, 16($29)
            move $8, $0
for:        beq $8, $7, exit
            lw $10, 0($4)
            lw $11, 0($5)
            slt $9, $10, $11
            bne $9, $0, sum
            sub $12, $10, $11
continue:   sw $12, 0($6)
            addi $8, $8, 1
            addi $4, $4, 4
            addi $5, $5, 4
            addi $6, $6, 4
            j for

exit:       lw $8, 0($29)
            lw $9, 4($29)
            lw $10, 8($29)
            lw $11, 12($29)
            lw $12, 16($29)
            addi $29, $29, 20
            jr $31

sum:        add $12, $10, $11
            j continue

```

ESERCIZIO 3 (solo VO: 7 punti)

Considerato un campo di 32 bit, siano dati i seguenti formati:

1. rappresentazione di interi senza segno;
2. rappresentazione in virgola fissa con bit di segno e 20 bit di parte frazionaria;
3. rappresentazione in virgola mobile con mantissa frazionaria e normalizzata in segno e valore (1.M) ed esponente a 8 bit in eccesso 128.

a) (3 punti) Calcolare il minimo e il massimo valore rappresentabile in valore assoluto nei tre casi.

c) (4 punti) Sommare i due numeri, $(21.5)_{10}$ e $(7.25)_{10}$, esprimendoli in virgola mobile secondo la rappresentazione 3, con l'algoritmo dei calcolatori.

Soluzione

a)

1. Minimo: 1 Max: $2^{32}-1$.

2. Minimo: 2^{-20} Max: $2^{11}-2^{-20}$

3. Minimo: 2^{-128} Max: $2^{127}(2-2^{-23})$.

b) $(21.5)_{10} = 10101.1 = 1.01011 \cdot 2^4$

$(7.25)_{10} = 111.01 = 1.1101 \cdot 2^2$

I due numeri si possono rappresentare nel seguente modo:

Segno	Esponente	Mantissa
0	10000100	010110000000000000000000
0	10000010	110100000000000000000000

Poiché il primo ha esponente maggiore del secondo ($4 > 2$) di quest'ultimo si fa scorrere la mantissa a destra di **due** posizioni.

I due numeri da sommare sono:

$$\begin{array}{r}
 1.010110 + \\
 0.011101 = \\
 \hline
 1.110011 \quad (*2^4)
 \end{array}$$

Segno Esponente Mantissa
 0 10000100 110011000000000000000000

ESERCIZIO 4 (NO: 8 punti – VO: 6 punti)

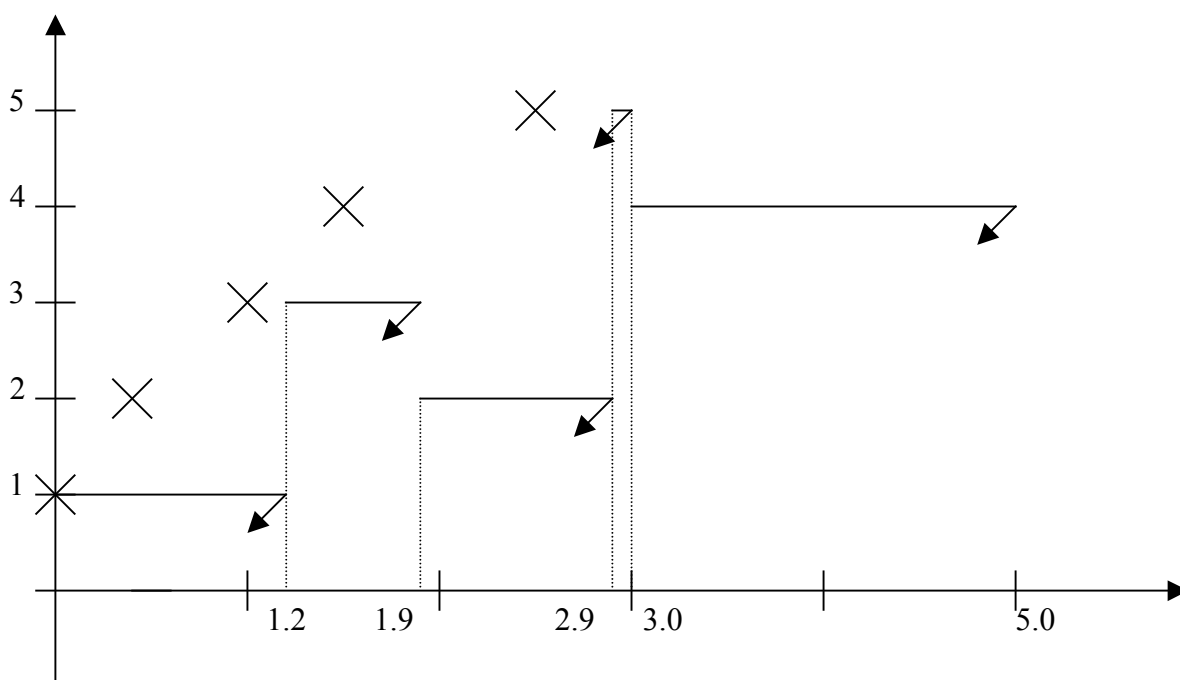
Sia data la seguente lista di processi:

Processo	Tempo di arrivo	Tempo di CPU
1	0.0	1.2
2	0.4	1.0
3	1.0	0.7
4	1.5	2.0
5	2.5	0.1

1. Mostrare, utilizzando il metodo grafico, la sequenza di esecuzione dei processi qualora si impieghi la politica di scheduling SJF monoprogrammata (NO: 5 punti – VO: 3 punti).
2. Calcolare il tempo di *turnaround* medio e il tempo di *turnaround* pesato medio (3 punti).

Soluzione.

Grafico processo-tempo con politica SJF monoprogrammata:



Processo	Arrivo	Inizio	Fine	Turnaround	WT
1	0.00	0.00	1.20	1.20	1.00
2	0.40	1.90	2.90	2.50	2.50
3	1.00	1.20	1.90	0.90	1.29
4	1.50	3.00	5.00	3.50	1.75
5	2.50	2.90	3.00	0.50	5.00
Media				1.72	2.31

ESERCIZIO 5 (solo VO: 5 punti)

Illustrare in modo chiaro e sintetico i vantaggi delle architetture RISC rispetto alle architetture CISC.

Soluzione.

Vedi dispense del corso.