

**SOLUZIONI DELLA PROVA SCRITTA DEL CORSO DI  
CALCOLATORI ELETTRONICI  
NUOVO E VECCHIO ORDINAMENTO DIDATTICO**  
21 Giugno 2006

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

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

Progettare una rete sequenziale sincrona con un ingresso X che riconosca la sequenza in ingresso 101101. Il riconoscimento avviene ponendo l'uscita Z ad 1, e 0 in **tutti** gli altri casi.

Si richiede inoltre:

- (NO: 5 punti – VO: 4 punti) il diagramma degli stati, la tabella di flusso e la tabella delle transizioni;
- (3 punti) il calcolo delle forme minime delle variabili di eccitazione dei flip flop con le mappe di Karnaugh. Si usino flip flop T. Calcolare anche la rete combinatoria per l'uscita Z.

**Soluzione.**

Diagramma degli stati:

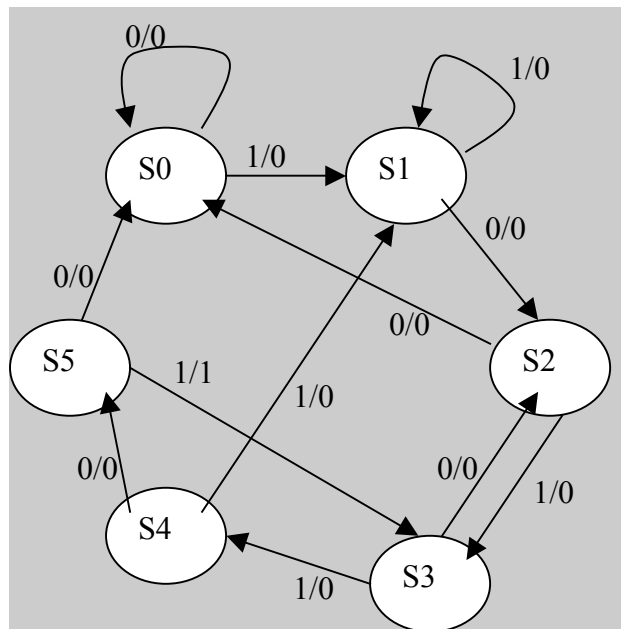


Tabella di flusso:

Stato iniziale	Stato successivo	
	X=0	X=1
S0	S0/0	S1/0
S1	S2/0	S1/0
S2	S0/0	S3/0
S3	S2/0	S4/0
S4	S5/0	S1/0
S5	S0/0	S3/1

Trattandosi di sei stati, occorrono tre bit per rappresentarli che indicheremo con ABC a partire dal bit più significativo, con le seguenti associazioni: S0=000, S1=001, ..., S5=101.

Ricordandoci la tabella di eccitazione del FF-T:

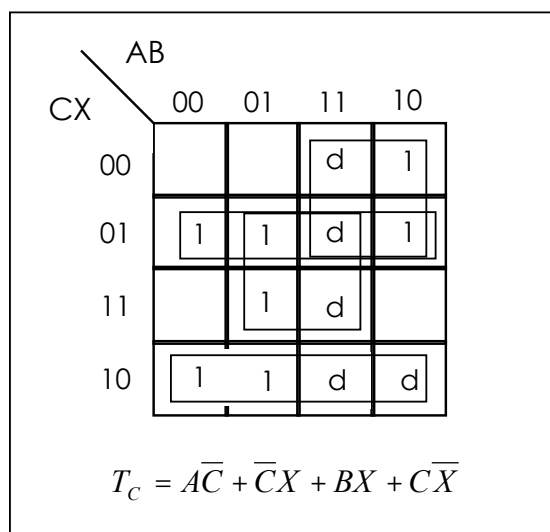
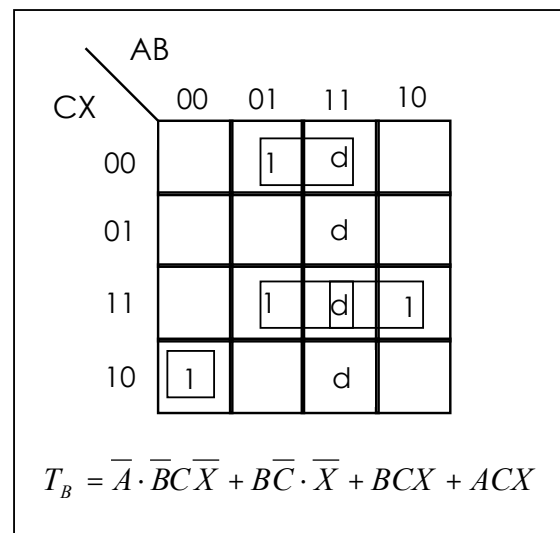
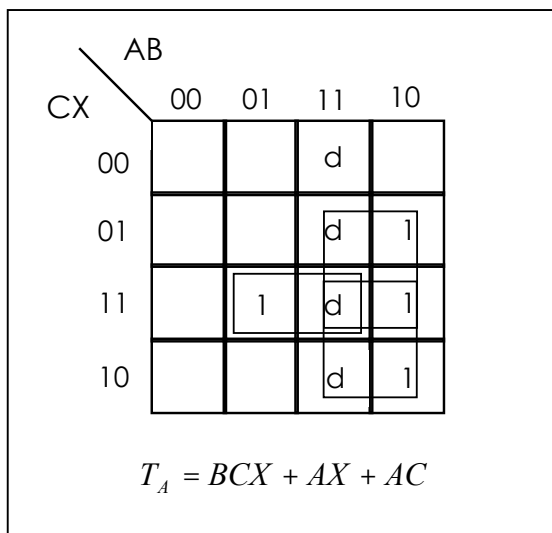
Q	Q'	T
0	0	0
0	1	1
1	0	1
1	1	0

Si ha la seguente tabella delle transizioni:

A	B	C	X	A'	TA	B'	TB	C'	TC	Z
0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	1	1	0
0	0	1	0	0	0	1	1	0	1	0
0	0	1	1	0	0	0	0	1	0	0
0	1	0	0	0	0	0	1	0	0	0
0	1	0	1	0	0	1	0	1	1	0
0	1	1	0	0	0	1	0	0	1	0
0	1	1	1	1	1	0	1	0	1	0
1	0	0	0	1	0	0	0	1	1	0
1	0	0	1	0	1	0	0	1	1	0
1	0	1	0	0	1	0	0	0	1	0
1	0	1	1	0	1	1	1	1	0	1
1	1	0	0	D	D	D	D	D	D	0
1	1	0	1	D	D	D	D	D	D	0
1	1	1	0	D	D	D	D	D	D	0
1	1	1	1	D	D	D	D	D	D	0

Poiché il testo richiede che l'uscita Z sia posta ad 1 solo nel caso in cui la sequenza sia riconosciuta, la relativa espressione di Z è  $AB'CX$  (l'apice indica la negazione di B).

Le mappe di Karnaugh e le forme minime per le variabili di eccitazione dei FF-T sono le seguenti:



**ESERCIZIO 2 (NO: 10 punti – VO: 8 punti)**

Si consideri una gerarchia di memorie cache-primaria-disco.

La memoria primaria è suddivisa in 16 blocchi di 4 parole ciascuno. La cache presenta 32 parole ed è indirizzata in modo set-associativo a quattro vie.

Il disco presenta le seguenti caratteristiche: tempo di latenza pari a 4 ms, tempo di posizionamento da una traccia a quella successiva pari a 1 ms, 100 settori per traccia, dimensione di un settore pari a 8 B.

- 1) (1 punto) Indicare, specificando il significato e la funzione dei diversi campi, come viene recuperata l'informazione nella cache a partire dall'indirizzo della parola in memoria primaria.
- 2) (NO: 6 punti - VO: 4 punti) Sia data la sequenza di chiamate ad altrettanti indirizzi di memoria (il primo indirizzo utile è 0): 24 31 8 42 21 25 55 15 50 51 27 30 39 26 2. Gli indirizzi sono espressi in notazione decimale. Considerando la memoria cache inizialmente vuota, si indichi lo stato finale della cache e l'hit ratio. La politica di sostituzione adottata è la LRU.
- 3) (3 punti) Si calcoli il tempo di lettura di un file costituito da 64 blocchi di 16 B ciascuno. Il file risulta memorizzato in due gruppi di settori contigui. I gruppi sono distanti fra loro due tracce. Si consideri la testina posizionata, all'istante iniziale, sul primo settore utile.

**Soluzione.**

1)

La memoria primaria è costituita da  $16 \times 4$  parole = 64 parole per le quali occorrono 6 bit di indirizzamento. I blocchi sono invece costituiti da quattro parole, quindi sono necessari 2 bit per indirizzare la parola all'interno del blocco.

L'indirizzamento della cache, essendo set-associativo a quattro vie, fa sì che ogni insieme contenga quattro blocchi. Il numero di insiemi è dunque dato da  $32/(4 \times 4) = 2$ . Per indirizzare questi insiemi occorre un bit dell'indirizzo di primaria. I restanti bit servono per il campo TAG. Riassumendo:

< TAG 3 bit > < Index 1 bit > < Offset 2 bit >

2)

Per ogni chiamata, scriviamo i relativi block frame, cache index ed eventuale hit. L'ultima riga segnala il blocco candidato alla sostituzione nel relativo insieme di cache, a partire dall'istante in cui esso risulta "pieno", con la politica di sostituzione LRU.

Ind.	24	31	8	42	21	25	55	15	50	51	27	30	39	26	2
B.f.	6	7	2	10	5	6	13	3	12	12	6	7	9	6	0
C.i.	0	1	0	0	1	0	1	1	0	0	0	1	1	0	0
Hit						X				X	X	X		X	
Evento								Sost. 7	Sost. 2			Sost. 5	Out →5 Sost. 13		Out →2 Sost. 10

L'hit ratio è 5/15, e lo stato finale della memoria è il seguente:

Insieme 0				Insieme 1			
6		0		7		9	
10		12		13		3	

3)

Innanzitutto ricaviamo il tempo di rotazione. Dalla relazione  $T_{rot}/2 = T_{lat}$  si ha  $T_{rot} = 8$  ms.

Inoltre il tempo di lettura di un settore è dato da  $T_{rot}/N_{sett} = 8/100 = 0.08$  ms.

Poiché ciascun blocco del file occupa due settori (la dimensione di un blocco è infatti doppia di quella di un settore), il numero di settori complessivo è pari a 128.

Il testo non precisa la dimensione dei raggruppamenti. Poniamo dunque che  $N_1$  sia il numero di settori del primo e  $N_2$  il numero di settori del secondo. Ovviamente  $N_1 + N_2 = 128$ , con  $N_1 \leq 100$  e  $N_2 \leq 100$  (capacità della traccia).

Per leggere il primo raggruppamento la testina impiega il tempo necessario alla lettura di  $N_1$  settori, in quanto all'istante iniziale la testina è posizionata sul primo settore utile. Tale tempo è pari a  $T_1 = N_1 \times 0.08$  ms.

Prima di leggere il secondo raggruppamento, di  $N_2$  settori contigui, la testina del disco impiega il tempo necessario per spostarsi di due tracce e il disco effettua una latenza. Il tempo di lettura del

secondo raggruppamento è pari a  $T_2 = 2 \text{ ms (posizionamento)} + 4 \text{ ms (latenza)} + N_2 * 0.08 \text{ ms} = 6 + N_2 * 0.08 \text{ ms}$ .

Il tempo complessivo di lettura è pari alla somma dei due tempi precedenti cioè:

$$T_1 + T_2 = (N_1 * 0.08) + (6 + N_2 * 0.08) = 6 + (N_1 + N_2) * 0.08 = 6 + 128 * 0.08 = 16.24 \text{ ms}.$$

### ESERCIZIO 3 (solo NO: 8 punti)

Sia dato un vettore di interi  $v$  così caratterizzato: il valore in posizione  $i$  risulta strettamente maggiore del valore in posizione  $2i+1$  e strettamente minore del valore in posizione  $2i+2$ . Il vettore presenta un massimo di  $N$  valori. Il primo valore del vettore è in posizione 0.

Si implementi in Assembler MIPS una funzione che effettui la ricerca nel vettore di un valore  $z$  a partire dal primo valore del vettore. La funzione presenta i seguenti ingressi: indirizzo iniziale di  $v$  in  $\$4$ ,  $N$  in  $\$5$ ,  $z$  in  $\$6$ . In uscita la funzione deve scrivere in  $\$7$  il valore dell'indice di  $v$  corrispondente a  $z$ , se esiste, altrimenti deve scrivere in  $\$7$  il valore  $N$ .

Ad esempio, la funzione MIPS potrebbe implementare il seguente codice C:

```
int cerca(int *v, int N, int z)
{
    int trovato, i;
    trovato=N;
    i=0;
    while ((i<N) && (trovato==N))
        if (v[i]==z)
            trovato=i;
        else if (v[i]<z)
            i=2*i+2;
        else
            i=2*i+1;
    return trovato;
}
```

### Soluzione.

$\$8 \leftarrow i$ ;  $\$9 \leftarrow 4*i$ ;  $v[i] \rightarrow \$10$ ;  $(i < N) \text{ e } (v[i] < z) \rightarrow \$11$

```
cerca:      addi $29, $29, -16
            sw $8, 0($29)
            sw $9, 4($29)
            sw $10, 8($29)
            sw $11, 12($29)
            move $7, $5
            move $8, $0
while:      slt $11, $8, $5
            beq $11, $0, exit
            muli $9, $8, 4
            add $9, $9, $4
            lw $10, 0($9)
            beq $10, $6, trovato
            muli $8, $8, 2
            addi $8, $8, 1
            slt $11, $10, $6
            bne $11, $0, aggiungi
            j while
aggiungi:   addi $8, $8, 1
            j while
trovato:    move $7, $8
exit:      lw $8, 0($29)
            lw $9, 4($29)
            lw $10, 8($29)
            lw $11, 12($29)
            addi $29, $29, 16
            jr $31
```

**ESERCIZIO 3 (solo VO: 6 punti)**

I trasferimenti di parole a/dalla memoria di un calcolatore sono codificate utilizzando il codice di Hamming. Si consideri la stringa di 12 bit 101001101110 (il bit meno significativo è a sinistra), risultato della codifica di una parola di N bit secondo il codice di Hamming.

- 1) (1 punto) calcolare N, supponendo di aver fatto uso del numero minimo di bit di controllo necessari.
- 2) (1 punto) scrivere la parola di N bit a partire dalla stringa data;
- 3) (4 punti) indicare eventuali errori nella stringa codificata, specificando quale dei bit è stato alterato.

**Soluzione.**

- 1) Deve venire rispettata la condizione:

$$2^K \geq N + K + 1 \quad (1),$$

dove K è il numero di bit di controllo inseriti. Essendo  $N + K = 12$ , si evince dalla (1) che il numero minimo di bit di controllo richiesto è 4. Da cui  $N = 8$ .

- 2) Nella codifica di Hamming, la sequenza in ingresso presenta la seguente struttura:

$c_0$	$c_1$	$b_0$	$c_2$	$b_1$	$b_2$	$b_3$	$c_3$	$b_4$	$b_5$	$b_6$	$b_7$
1	0	1	0	0	1	1	0	1	1	1	0

Dove  $c_0 \dots c_3$  sono i quattro bit costituenti il vettore di controllo, e  $b_0 \dots b_7$  gli otto bit trasmessi. La sequenza ricevuta è 10111110.

- 3) Per verificare la presenza di un errore, dobbiamo ricalcolare il vettore di controllo a partire dalla sequenza ricevuta. Si ha:

$$c'_0 = b_0 \oplus b_1 \oplus b_3 \oplus b_4 \oplus b_6 = 0$$

$$c'_1 = b_0 \oplus b_2 \oplus b_3 \oplus b_5 \oplus b_6 = 1$$

$$c'_2 = b_1 \oplus b_2 \oplus b_3 \oplus b_7 = 0$$

$$c'_3 = b_4 \oplus b_5 \oplus b_6 \oplus b_7 = 1$$

Il passo successivo è calcolare il vettore di errore dato dalla differenza dei vettori di controllo  $c$  e  $c'$  (ricordiamo che somma e differenza tra bit producono lo stesso risultato):

$$e_0 = c_0 \oplus c'_0 = 1$$

$$e_1 = c_1 \oplus c'_1 = 1$$

$$e_2 = c_2 \oplus c'_2 = 0$$

$$e_3 = c_3 \oplus c'_3 = 1$$

Poiché il vettore risultante 1011 non è nullo, vi è un errore nella stringa di 12 bit data e precisamente nella posizione indicata dal vettore di errore tradotto in notazione decimale. Il bit sbagliato è quindi l'undicesimo ( $b_6$ ), e la parola corretta è 10111100.

**ESERCIZIO 5 (solo VO: 5 punti)**

Descrivere in modo chiaro e sintetico la classificazione di Flynn delle architetture parallele.

**Soluzione.**

Vedi dispense del corso.

**ESERCIZIO 4 (7 punti)**

Si consideri un sistema operativo con gestione della memoria paginata. Ciascun processo ha una dimensione massima di 128 pagine. Le pagine hanno dimensione 1KB. Si abbia ad esempio la seguente situazione:

Tabella delle Pagine Processo 0			
Pagina Virtuale	Pagina Fisica	Bit di Validità	Solo lettura
0000000	0000011	0	0
0000001	0010110	0	0
0000010	1001001	1	1
0000011	1001010	1	1
0000100	1010101	1	1
0000101	0011101	1	0
0000110	0111111	1	0
0000111	1011101	1	0
0001000	1010011	1	0
0001001	0001111	1	0
0001010	0011011	1	0
0001011	0100010	1	0

Tabella delle Pagine Processo 1			
Pagina Virtuale	Pagina Fisica	Bit di Validità	Solo lettura
0000000	0001010	1	0
0000001	0010001	0	0
0000010	1011001	1	0
0000011	1011010	1	0
0000100	0011101	1	0
0000101	0101111	0	0
0000110	0110111	1	0
0000111	0100000	1	0
0001000	0100100	1	0
0001001	0001001	1	0

Tabella delle Pagine Processo 2			
Pagina Virtuale	Pagina Fisica	Bit di Validità	Solo lettura
0000000	0100001	1	1
0000001	0101110	1	1
0000010	0110110	0	1
0000011	0000110	1	1
0000100	1100011	0	1
0000101	1000011	1	1
0000110	0010101	1	1

Il bit di validità indica se la pagina virtuale richiesta è presente (1) o no (0) nella memoria principale.

- (VO: 2 – NO: 3 punti) Calcolare la dimensione in bit degli indirizzi virtuali, spiegando come vengono interpretati dal sistema operativo.
- (VO: 3 – NO: 4 punti) Si considerino le seguenti richieste e mostrare: a quale indirizzo fisico corrispondono gli indirizzi virtuali, se l'operazione può essere conclusa con successo o se viene generato un errore (pagina non valida, violazione di protezione) o un page fault:
  - (Processo 0) Read 00010110101001101
  - (Processo 2) Write 00001010101011001
  - (Processo 1) Read 00010011001110011
  - (Processo 2) Read 00001001100100110
  - (Processo 0) Write 00000110001100001
- (solo VO: 2 punti) Spiegare in modo chiaro e sintetico in cosa consiste una memoria paginata e segmentata.

**Soluzione:**

- L'indirizzo virtuale ha dimensione pari almeno a 17 bit e viene interpretato a partire dai bit più significativi nel modo seguente:  
7 bit: pagina virtuale  
10 bit: posizione all'interno della pagina.
- A partire dalla interpretazione del punto 1. Si ottiene:
  - Indirizzo fisico valido: 01000100101001101
  - Errore: scrittura su indirizzo fisico 10000110101011001 protetto in scrittura
  - Indirizzo fisico valido: 00010011001110011
  - "Page fault all'indirizzo fisico 00111011100100110
  - Errore: scrittura su indirizzo fisico 10010100001100001 protetto in scrittura.
- Vedi dispense del corso.