

PROVA SCRITTA DEL CORSO DI
CALCOLATORI ELETTRONICI
NUOVO ORDINAMENTO DIDATTICO (7 CFU)
 14 Settembre 2010

NOME:

COGNOME:

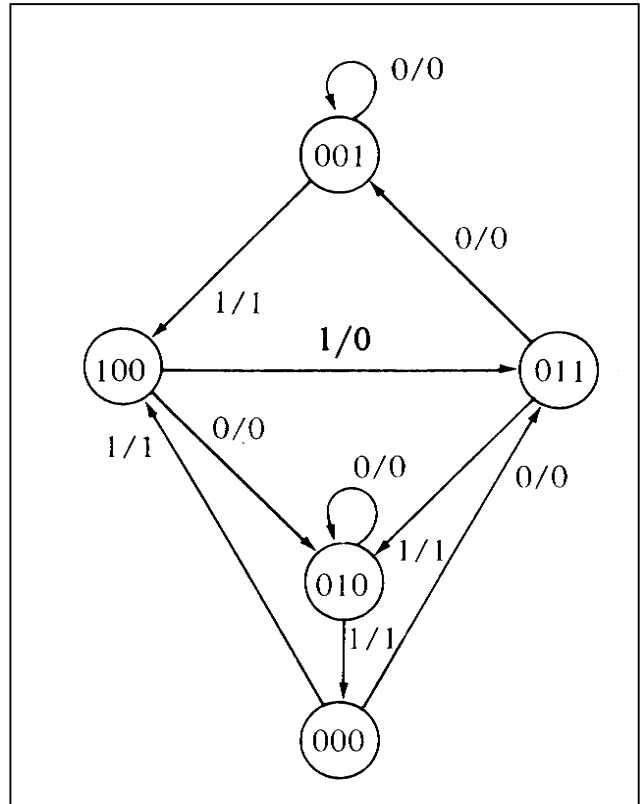
MATRICOLA:

ESERCIZIO 1 (7 punti)

Un circuito sequenziale presenta un ingresso, una uscita, e il grafo degli stati a fianco. Si richiede di:

1. (4 punti) progettare la corrispondente rete sequenziale utilizzando FF-JK;
2. (3 punti) progettare la corrispondente rete sequenziale utilizzando FF-T.

In entrambi i casi calcolare le forme minime per le variabili di eccitazione dei flip flop e l'uscita usando le mappe di Karnaugh.



ESERCIZIO 2 (6 punti)

La memoria di un calcolatore viene gestita con il metodo delle partizioni statiche. A partire dall'indirizzo 0 si hanno le seguenti partizioni (in ordine): 200K, 100K, 400K, 300K, 500K. Con la memoria inizialmente vuota si debbano allocare, nell'ordine riportato, quattro processi che occupano rispettivamente: 130K, 210K, 78K, 450K.

1. (3 punti) Mostrare il contenuto finale della memoria impiegando le strategie First-Fit, Best-Fit, Worst-Fit.
2. (3 punti) Nell'eventualità che, in qualcuno dei tre casi, uno o più processi non fosse allocabile con il partizionamento statico, sarebbe possibile allocarlo usando invece il partizionamento dinamico?

ESERCIZIO 3 (7 punti)

Implementare una procedura Assembly MIPS che, dati l'indirizzo iniziale di un vettore v in \$4 e la sua dimensione N in \$5, restituisca il massimo valore contenuto nel vettore in \$6.

ESERCIZIO 4 (8 punti)

Si consideri una memoria primaria costituita da 8 parole indirizzabili e da una cache formata da quattro parole.

1. (2 punti) Spiegare come vengono interpretati gli indirizzi logici per recuperare l'informazione contenuta nella cache nel caso venga usata la modalità di indirizzamento:
 - a. Diretto, con blocchi di due parole
 - b. "associativo su insiemi", con insiemi di due blocchi di una parola ciascuno.
2. (4 punti) Ipotizzando la cache vuota all'istante iniziale, indicare nei casi (a-b) lo stato finale della cache e il numero di hit nel caso vengano inoltrate le seguenti chiamate (indirizzi espressi in decimale, primo indirizzo 0): 4, 6, 6, 5, 1, 3, 7, 6, 3, 6. Nel caso (b) si ipotizzi una politica di sostituzione dei blocchi FIFO.
3. (2 punti) Spiegare in modo chiaro e sintetico quali sono i registri della CPU coinvolti nella lettura/scrittura di una parola dalla/sulla memoria.

ESERCIZIO 5 (5 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 011001111110 (il bit meno significativo è a sinistra), risultata 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 necessario per una stringa di 12 bit;
2. (1 punto) scrivere la parola di N bit a partire dalla stringa data;
3. (3 punti) indicare eventuali errori nella stringa codificata, specificando quale dei bit è stato alterato.

ESERCIZIO 1

Soluzione

Per risolvere l'esercizio è sufficiente sapere scrivere le tabelle delle transizioni facilmente ricavabili dal grafo degli stati.

Innanzitutto scriviamo la tabella di eccitazione dei due flip flop richiesti, il JK e il T:

Tabella di eccitazione
di un flip-flop JK

$Q(t)$	$Q(t+1)$	J	K
0	0	0	d
0	1	1	d
1	0	d	1
1	1	d	0

Tabella di eccitazione
di un flip-flop T

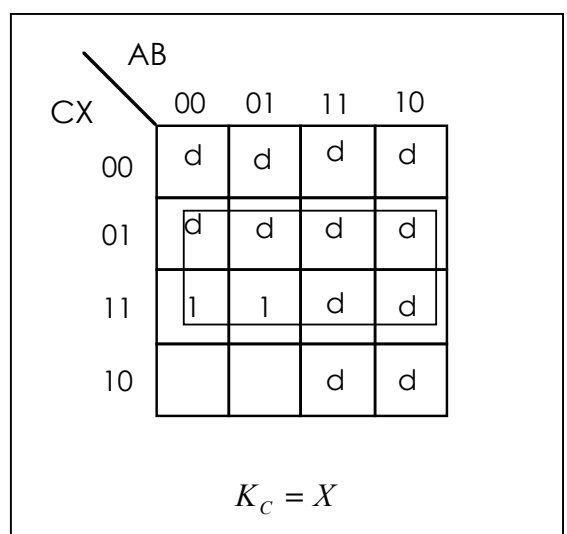
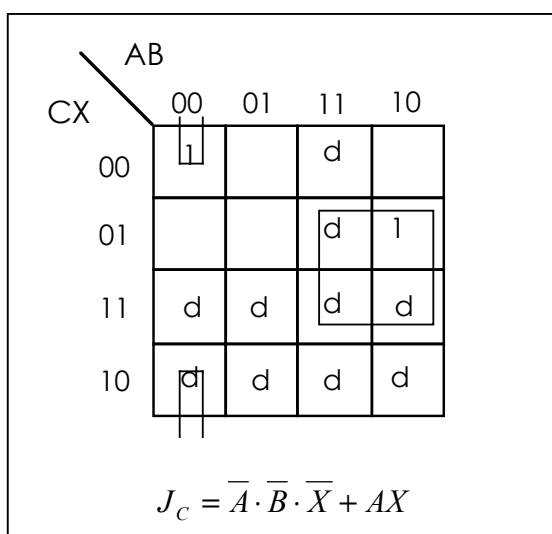
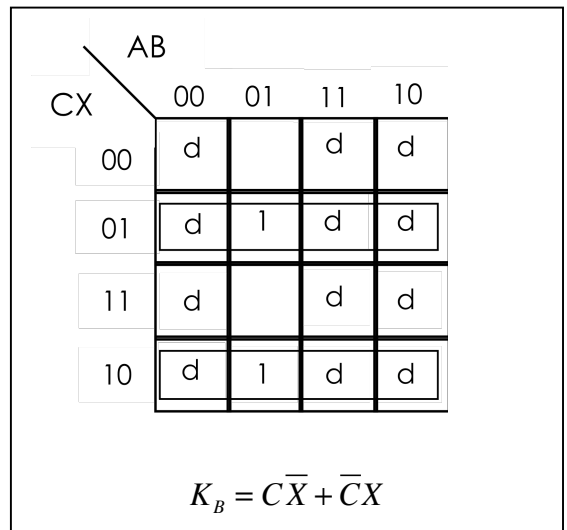
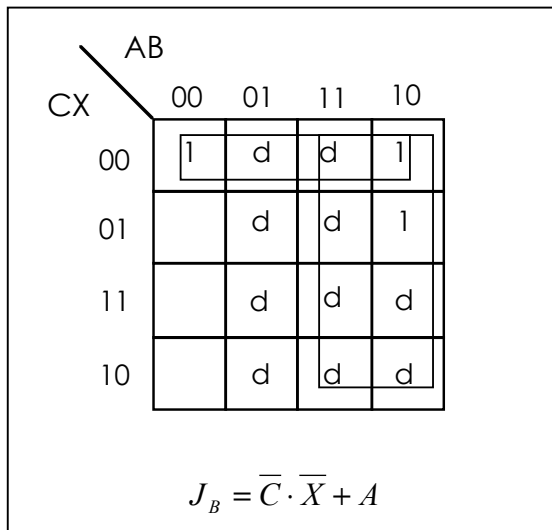
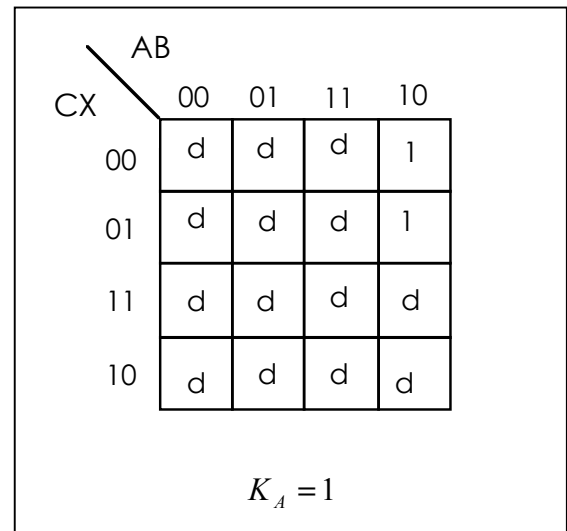
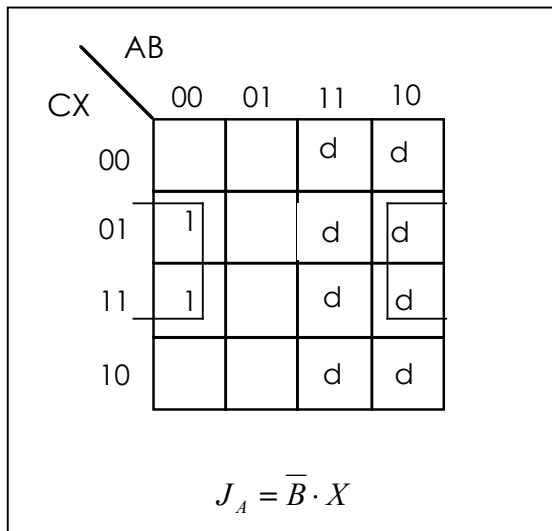
$Q(t)$	$Q(t+1)$	T
0	0	0
0	1	1
1	0	1
1	1	0

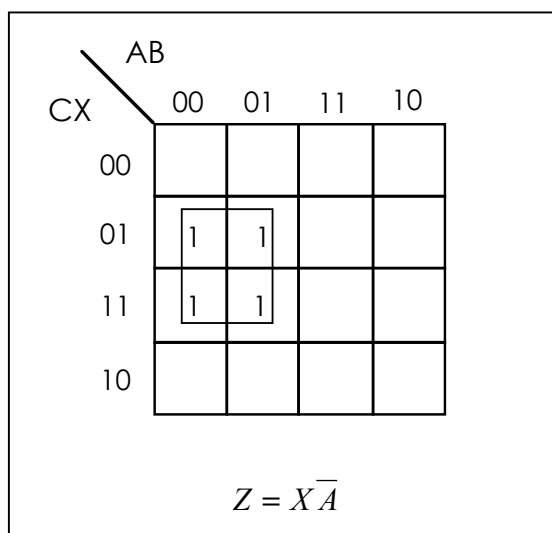
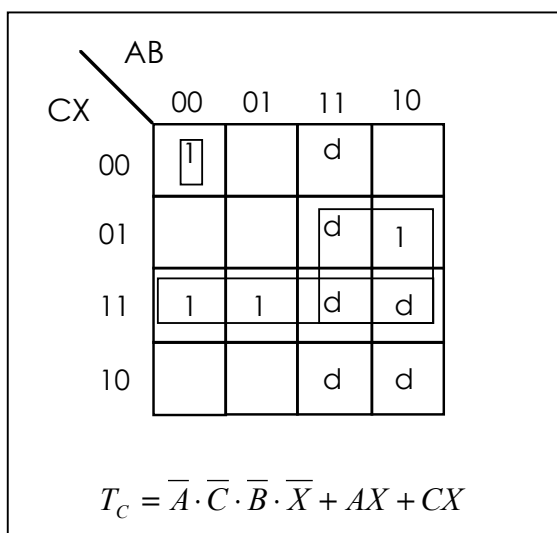
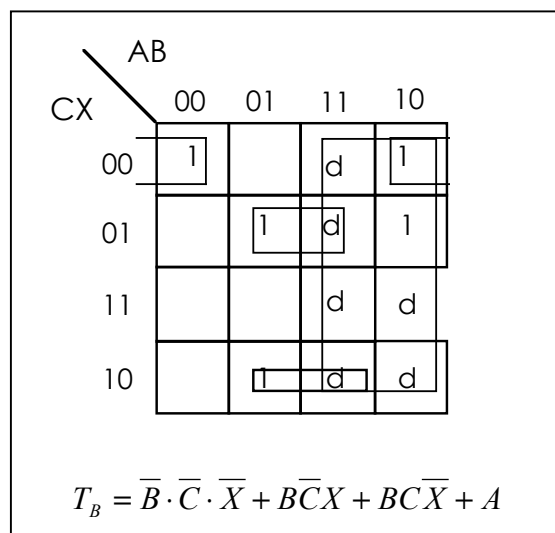
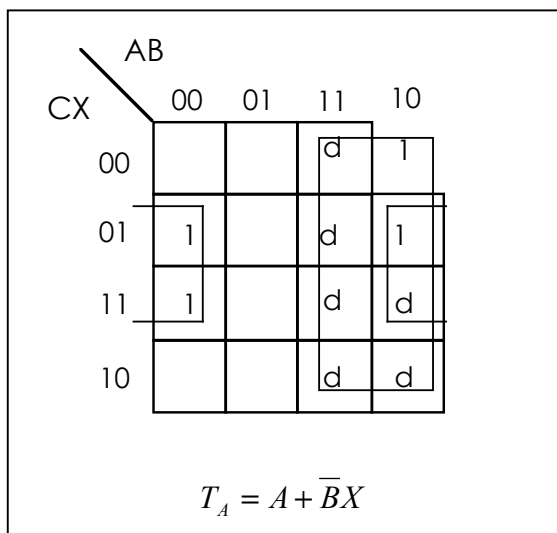
Indicando con A, B, C le variabili di stato, con X l'ingresso e con Z l'uscita, la relativa tabella delle transizioni è:

A	B	C	A'	J _A	K _A	T _A	B'	J _B	K _B	T _B	C'	J _C	K _C	T _C	Z
X = 0															
0	0	0	0	0	D	0	1	1	D	0	1	1	D	1	0
0	0	1	0	0	D	0	0	0	D	0	1	D	0	0	0
0	1	0	0	0	D	0	1	D	0	0	0	0	D	0	0
0	1	1	0	0	D	0	0	D	1	1	1	D	0	0	0
1	0	0	0	D	1	1	1	1	D	1	0	0	D	0	0
1	0	1	D	D	D	D	D	D	D	D	D	D	D	D	0
1	1	0	D	D	D	D	D	D	D	D	D	D	D	D	0
1	1	1	D	D	D	D	D	D	D	D	D	D	D	D	0
X = 1															
0	0	0	1	1	D	1	0	0	D	0	0	0	D	0	1
0	0	1	1	1	D	1	0	0	D	0	0	D	1	1	1
0	1	0	0	0	D	0	0	D	1	1	0	0	D	0	1
0	1	1	0	0	D	0	1	D	0	0	0	D	1	1	1
1	0	0	0	D	1	1	1	1	D	1	1	1	D	1	0
1	0	1	D	D	D	D	D	D	D	D	D	D	D	D	0
1	1	0	D	D	D	D	D	D	D	D	D	D	D	D	0
1	1	1	D	D	D	D	D	D	D	D	D	D	D	D	0

Si noti che nel caso dell'uscita non è stato usato il *don't care*, per rispettare il vincolo di progetto richiedente che tale variabile assuma il valore 1 solo dove indicato nel grafo degli stati.

Possiamo ora disegnare le mappe di Karnaugh in tutti i casi:





ESERCIZIO 2

Soluzione

FF	BF	WF
130K	130K	
78K	78K	
210K		210K
	210K	78K
450K	450K	130K

A fianco riportiamo lo stato finale della memoria utilizzando ciascun metodo di allocazione proposto.

Si nota che nel caso WF non è stato possibile allocare il processo da 450K. Il problema non è risolubile nemmeno col partizionamento dinamico, in quanto le due partizioni libere rimaste (la prima e la seconda), seppur compattabili, non raggiungono la dimensione adatta per potere contenere il processo.

ESERCIZIO 3

Soluzione

\$4 ← v[0]; \$5 ← N

```
max:      addi $29, $29, -16
          sw $4, 0($29)
          sw $8, 4($29)
          sw $9, 8($29)
          sw $10, 12($29)

          lw $6, 0($4)      #assumo che max <- v[0]
          addi $8,$0,1      #inizializza i=1
          addi $4,$4,4      #calcola in $4 <- &v[1]=&v[0]+4
for:      beq $8,$5, exit   #i==N, exit
          lw $9, 0($4)      #carico v[i] in $9
          slt $10, $6, $9   #$6 < $9 ->$10
          bnq $10, $0, max  #se $10 div da 0, aggiorno il max
          j inc             #altrimenti salto a inc

max:      move $6,$9

inc:      addi $8,$8,1      #i++
          addi $4,$4,4      #&v[i]++
          j for

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

          jr $31
```

ESERCIZIO 4

Soluzione.

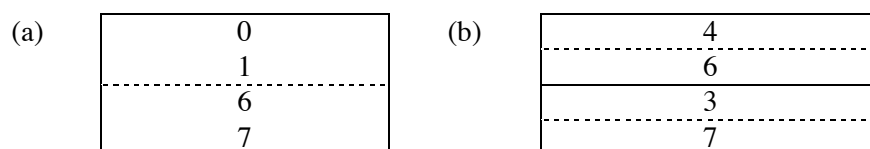
1.

Memoria indirizzabile: 8 parole = 2^3 parole \rightarrow 3 bit di indirizzamento

(a) <TAG 1 bit> <Cache Index 1 bit> <Offset 1 bit>

(b) <TAG 2 bit> <Cache Index 1 bit>

2. Nel seguito, è disegnata la cache di quattro parole, e il suo stato finale nei casi a-b. Con linee tratteggiate indichiamo la separazione tra un blocco e il successivo, con linee continue la separazione tra due insiemi.



Tale stato finale è dovuto alle seguenti dinamiche. La prima riga in tabella riporta la parola chiamata, l'ultima il verificarsi di un hit.

Metodo Diretto

4	6	6	5	1	3	7	6	3	6
4	4	4	4	0	0	0	0	0	0
5	5	5	5	1	1	1	1	1	1
	6	6	6	6	2	6	6	2	6
	7	7	7	7	3	7	7	3	7
h		h		h					

Metodo associativo su insiemi

4	6	6	5	1	3	7	6	3	6
4	4	4	4	4	4	4	4	4	4
	6	6	6	6	6	6	6	6	6
			5	5	3	3	3	3	3
				1	1	7	7	7	7
h		h		h		h		h	

Metodo diretto: 3/10 hit

Metodo associativo su insiemi: 4/10 hit

3. Vedi dispense del corso.

ESERCIZIO 5

Soluzione.

1. Deve essere 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
0	1	1	0	0	1	1	1	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):

$$e_0 = c_0 \oplus c'_0 = 0$$

$$e_1 = c_1 \oplus c'_1 = 0$$

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

$$e_3 = c_3 \oplus c'_3 = 0$$

Poiché il vettore risultante è nullo, non vi sono errori nella stringa di 12 bit data.