

**PROVA SCRITTA DEL CORSO DI**  
**CALCOLATORI ELETTRONICI**  
**NUOVO ORDINAMENTO DIDATTICO (5 CFU)**  
22 Febbraio 2011

**NOME:**

**COGNOME:**

**MATRICOLA:**

**ESERCIZIO 1 (7 punti)**

Progettare una rete sequenziale che presenti un ingresso X e un'uscita Z posta a 1 ogni volta che viene riconosciuta la sequenza 01101. Si richiede:

1. (4 punti) il diagramma degli stati, la tabella di flusso e la tabella delle transizioni;
2. (3 punti) il calcolo delle forme minime delle variabili di eccitazione dei flip flop con le mappe di Karnaugh. Si usino flip flop JK. Calcolare anche la rete combinatoria per l'uscita Z.

**ESERCIZIO 2 (8 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 3 (8 punti)**

Sia data una gerarchia di memoria costituita da memoria cache e primaria. La memoria cache ha una capacità di otto parole, con linee da due parole. La memoria primaria ha una capacità di sessantaquattro parole, con blocchi di due parole e metodo di indirizzamento diretto.

1. (2 punti) 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. (6 punti) Sia data la sequenza di chiamate ad altrettanti indirizzi di memoria espressi in decimale (prima parola, indirizzo 0): 14, 12, 15, 16, 13, 17, 17, 15, 16, 15. Si indichi lo stato finale della cache e l'hit ratio.

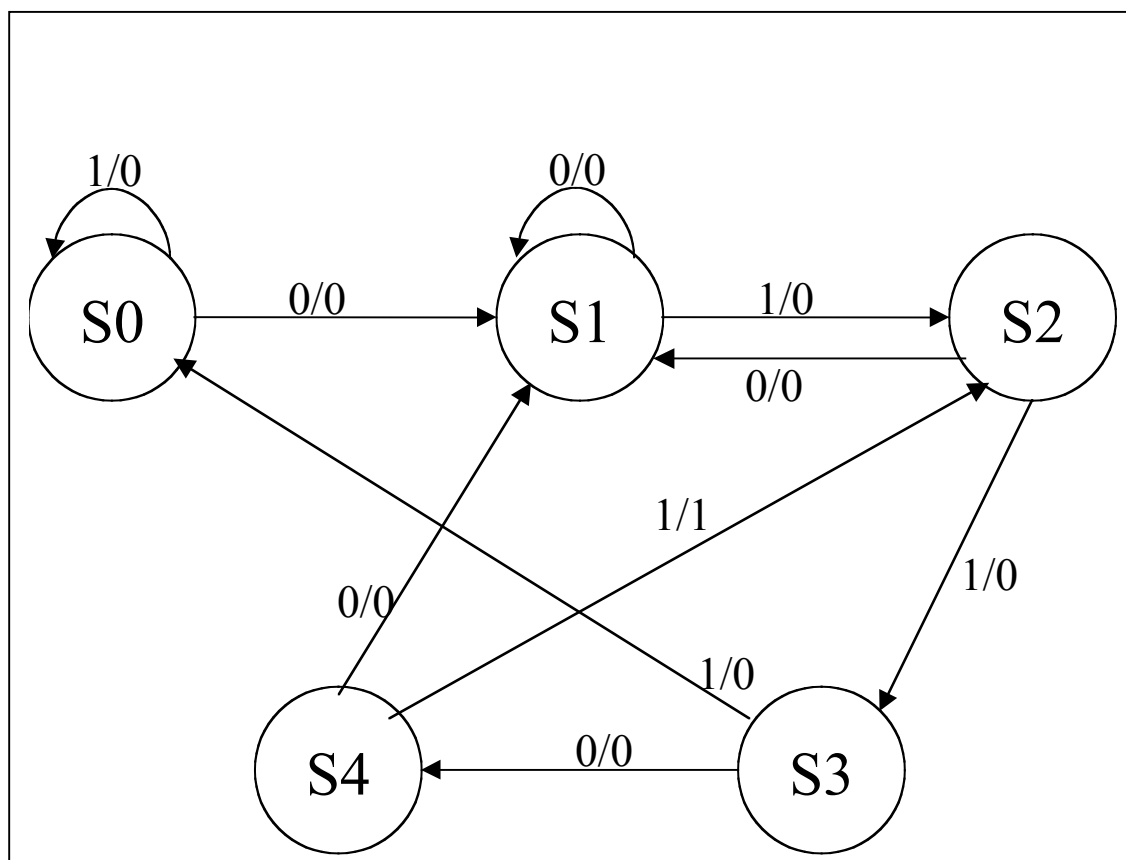
**ESERCIZIO 4 (10 punti)**

Considerato un campo di 64 bit, siano dati i seguenti formati:

- a. rappresentazione di interi senza segno;
  - b. rappresentazione in virgola fissa con bit di segno e 20 bit di parte frazionaria;
  - c. rappresentazione in virgola mobile con mantissa frazionaria normalizzata in segno e valore (0.1M) ed esponente a 8 bit in eccesso 127.
1. (3 punti) Calcolare il minimo e il massimo valore rappresentabile in valore assoluto nei tre casi.
  2. (3 punti) Sommare i due numeri,  $(12.5)_{10}$   $(5.25)_{10}$ , esprimendoli in virgola mobile secondo la rappresentazione data al punto c, con l'algoritmo dei calcolatori.
  3. (4 punti) Si dimostri analiticamente che la rappresentazione in complemento a due di un numero positivo su N bit, sommata al corrispondente valore negato, è uguale a 0.

**ESERCIZIO 1**  
**Soluzione**

Il diagramma degli stati è il seguente:



La tabella di flusso è data da:

Stato presente	Stato successivo/Uscita	
	X=0	X=1
S0	S1/0	S0/0
S1	S1/0	S2/0
S2	S1/0	S3/0
S3	S4/0	S0/0
S4	S1/0	S2/1

Per codificare 5 stati occorrono tre flip flop. La codifica è la seguente:

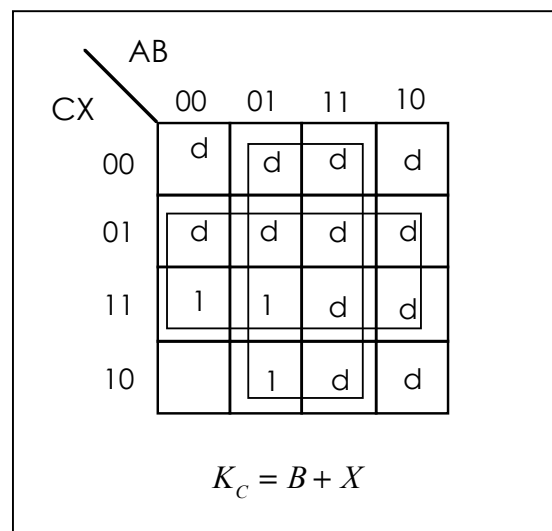
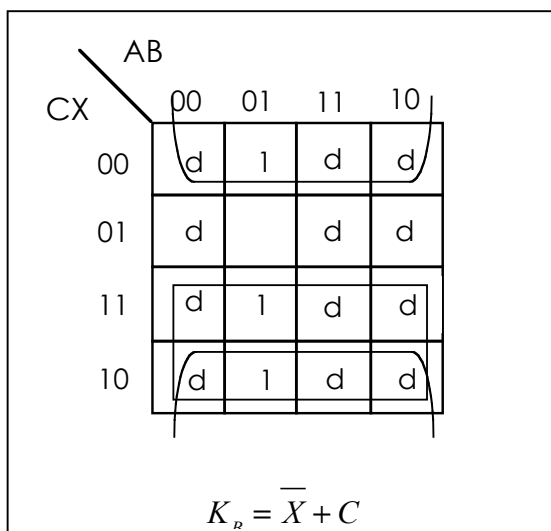
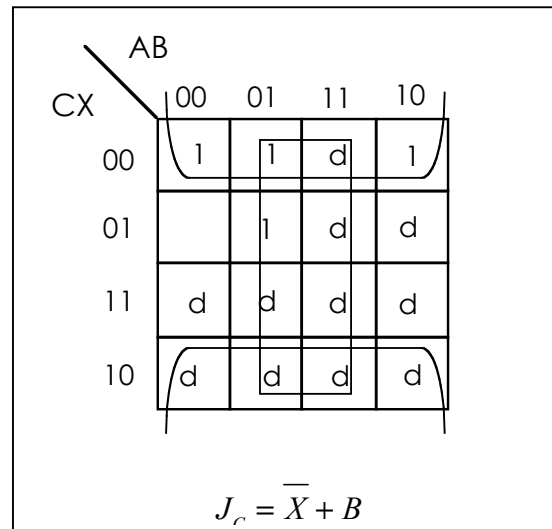
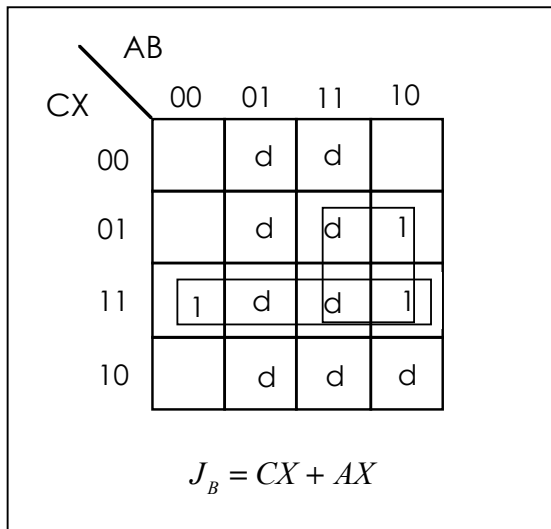
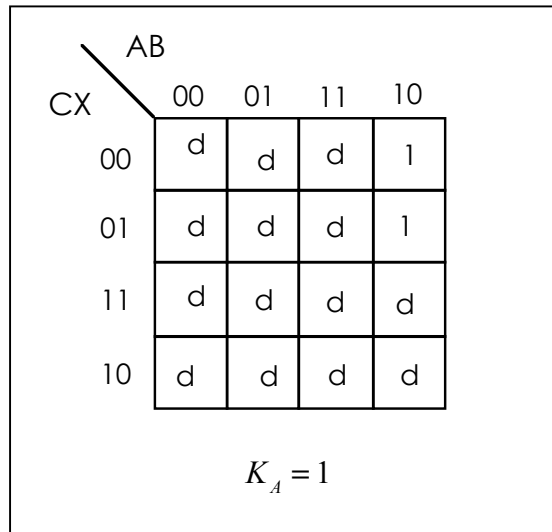
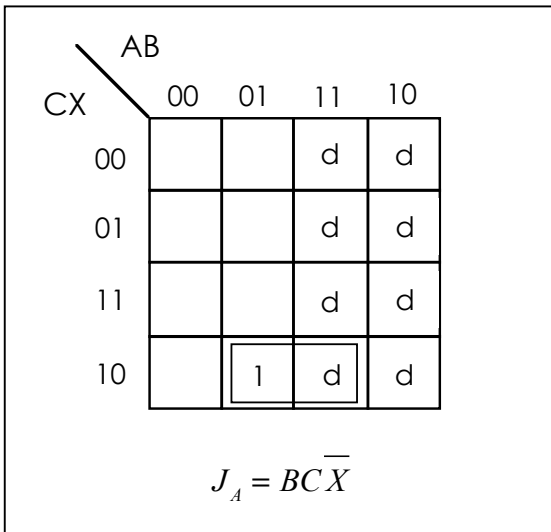
S0 → 0 0 0; ...; S4 → 1 0 0. Nel seguito indicheremo ciascun bit della codifica con le lettere A, B, C. L'apice indicherà il bit nell'istante successivo a quello considerato.

A partire dalla tabella di eccitazione del flip flop JK:

Q	Q'	J	K
0	0	0	D
0	1	1	D
1	0	D	1
1	1	D	0

A	B	C	X	A'	Ja	Ka	B'	Jb	Kb	C'	Jc	Kc	Z
0	0	0	0	0	0	D	0	0	D	1	1	D	0
0	0	0	1	0	0	D	0	0	D	0	0	D	0
0	0	1	0	0	0	D	0	0	D	1	D	0	0
0	0	1	1	0	0	D	1	1	D	0	D	1	0
0	1	0	0	0	0	D	0	D	1	1	1	D	0
0	1	0	1	0	0	D	1	D	0	1	1	D	0
0	1	1	0	1	1	D	0	D	1	0	D	1	0
0	1	1	1	0	0	D	0	D	1	0	D	1	0
1	0	0	0	0	D	1	0	0	D	1	1	D	0
1	0	0	1	0	D	1	1	1	D	0	0	D	1
1	0	1	0	D	D	D	D	D	D	D	D	D	D
1	0	1	1	D	D	D	D	D	D	D	D	D	D
1	1	0	0	D	D	D	D	D	D	D	D	D	D
1	1	0	1	D	D	D	D	D	D	D	D	D	D
1	1	1	0	D	D	D	D	D	D	D	D	D	D
1	1	1	1	D	D	D	D	D	D	D	D	D	D

Ora possiamo disegnare le mappe di Karnaugh



Infine, per quanto riguarda l'uscita Z:

AB		00	01	11	10
CX	00			d	
	01			d	1
	11			d	d
	10			d	d

$$Z = AX$$

## ESERCIZIO 2

### 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
          bne $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 3

#### Soluzione

- 1) I sei bit di indirizzamento sono suddivisi nei seguenti campi:  
< TAG 3 bit > < Cache Index 2 bit > < Offset 1 bit >.

2)

Per ottenere numero di blocco e numero di linea bisogna effettuare le seguenti operazioni:

Blocco = Parte\_intera(Indirizzo/2); Linea = Resto(Blocco/4)

<b>Chiamate</b>	14	12	15	16	13	17	17	15	16	15
<b>Blocco</b>	7	6	7	8	6	8	8	7	8	7
<b>Linea</b>	3	2	3	0	2	0	0	3	0	3
<b>Hit</b>			X		X	X	X	X	X	X

Stato finale della cache:

<b>Blocco 0</b>	<b>Blocco 1</b>	<b>Blocco 2</b>	<b>Blocco 3</b>
16		12	14
17		13	15

Hit ratio = 7/10.

#### ESERCIZIO 4

##### Soluzione

1.
  - a. Minimo: 1 Max:  $2^{64}-1$ .
  - b. Minimo:  $2^{-20}$  Max:  $2^{43}-2^{-20}$
  - c. Minimo:  $2^{-128}$  Max:  $2^{128}(1-2^{-56})$ .

2.  $(12.5)_{10} = 1100.1 = 0.11001 \cdot 2^4$   
 $(5.25)_{10} = 101.01 = 0.10101 \cdot 2^3$

I due numeri si possono rappresentare nel seguente modo:

Segno	Esponente	Mantissa
0	1000011	10010000000000000000...0
0	1000010	01010000000000000000...0

Poiché il primo ha esponente maggiore del secondo ( $4 > 3$ ) di quest'ultimo si fa scorrere la mantissa a destra di una posizione.

I due numeri da sommare sono:

$$\begin{array}{r} 0.110010 + \\ 0.010101 = \\ \hline 1.000111 \quad (*2^4) \end{array}$$

E' necessario normalizzare il risultato:

Segno	Esponente	Mantissa
0	1000100	00011100000000000000...0

3. Si veda Cap6.pdf, slide 13.