

PROVA SCRITTA DEL MODULO INTEGRATO E DEL CORSO DI
CALCOLATORI ELETTRONICI
CORSO DI LAUREA IN INGEGNERIA BIOMEDICA
CORSO DI LAUREA IN INGEGNERIA ELETTRICA, ELETTRONICA E INFORMATICA
23 ottobre 2018

NOME:

COGNOME:

MATRICOLA:

CFU:

ESERCIZIO 1 (10 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. (6 punti) il diagramma degli stati, la tabella di flusso e la tabella delle transizioni;
2. (4 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 (9 punti)

Si consideri una gerarchia di memoria composta da una memoria cache di 512 parole ed una memoria principale di 2048 parole. Ciascun blocco di primaria è costituito da 64 parole.

- 1) (2 punti) Illustrare come viene interpretato l'indirizzo di una generica parola nel caso in cui si utilizzi il metodo di indirizzamento completamente associativo.
- 2) (7 punti) Mostrare lo stato finale della memoria e il relativo cache hit nel caso in cui la sequenza di chiamate sia la seguente:
1788 – 1746 – 616 – 459 – 637 – 1198 – 865 – 1952 – 74 – 779 – 1705 – 137.
(N.B. Si ipotizzi una strategia di rimpiazzamento FIFO dove sia necessario).

ESERCIZIO 3 (9 punti)

Si scriva una funzione Assembly MIPS chiamata *ricerca* che, ricevendo l'indirizzo iniziale di un vettore v di valori interi non nulli in \$4 e un valore intero non nullo x in \$5, restituisca in \$6 il valore 1 se x è presente tra gli elementi di v , 0 altrimenti. L'ultimo valore presente nel vettore è sempre 0.

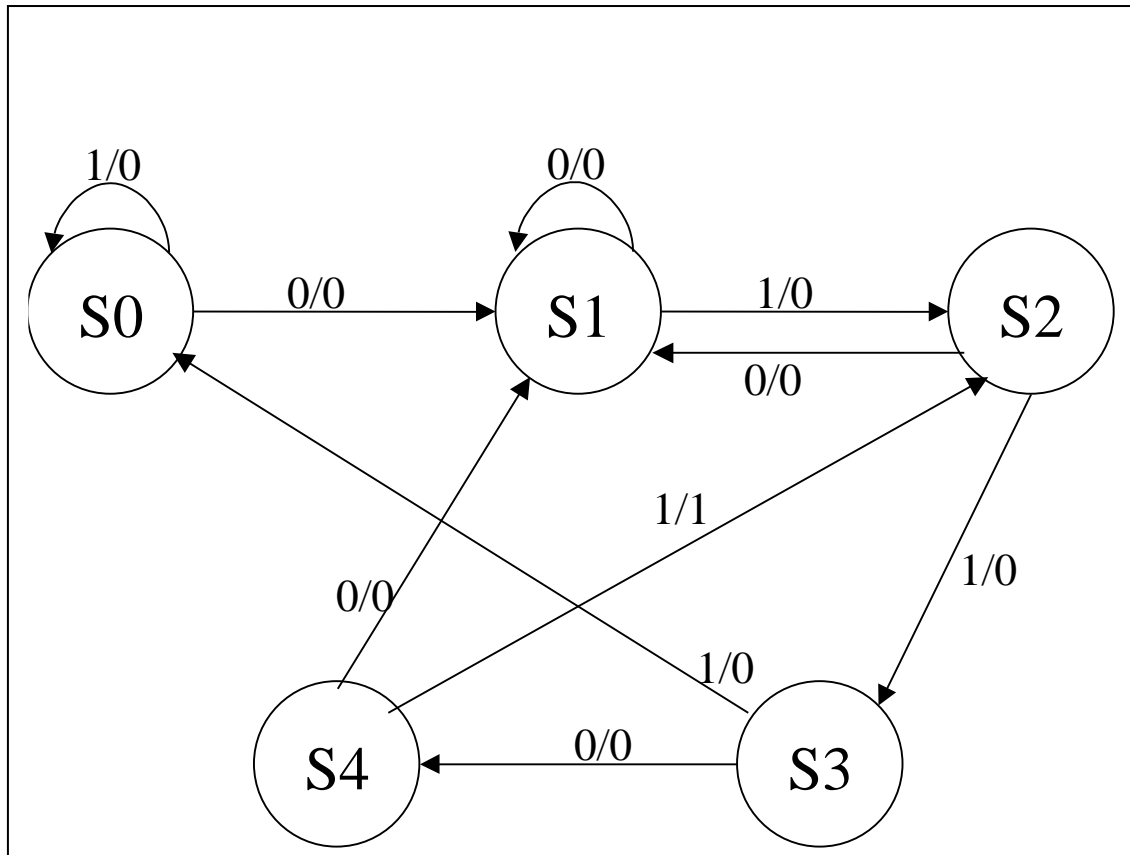
ESERCIZIO 4 (5 punti)

I trasferimenti di parole a/dalla memoria di un calcolatore sono codificate utilizzando il codice di Hamming. Si consideri la stringa in ricezione di 12 bit 000110101111, già codificata con il codice di Hamming (il bit meno significativo è a sinistra). Spiegando bene ogni passo del ragionamento:

1. (2 punti) calcolare il numero di bit della stringa di origine;
2. (3 punti) decodificare la stringa data, rilevando eventuali errori presenti in essa.

ESERCIZIO 1

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

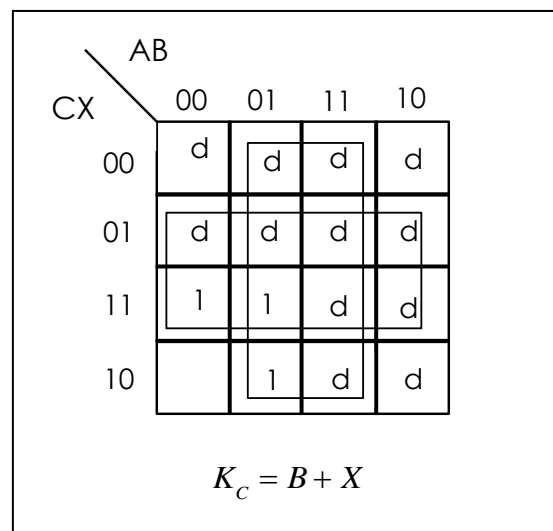
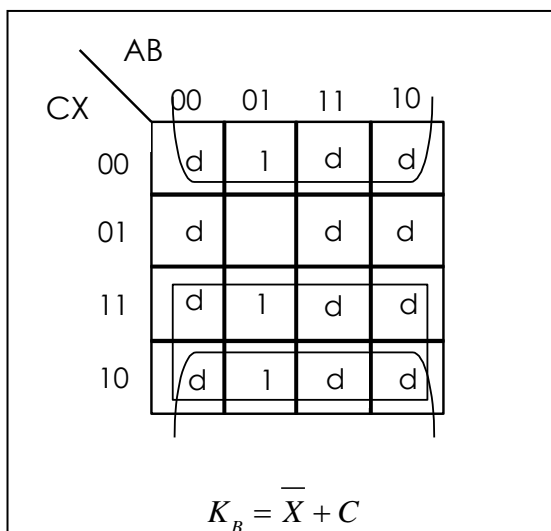
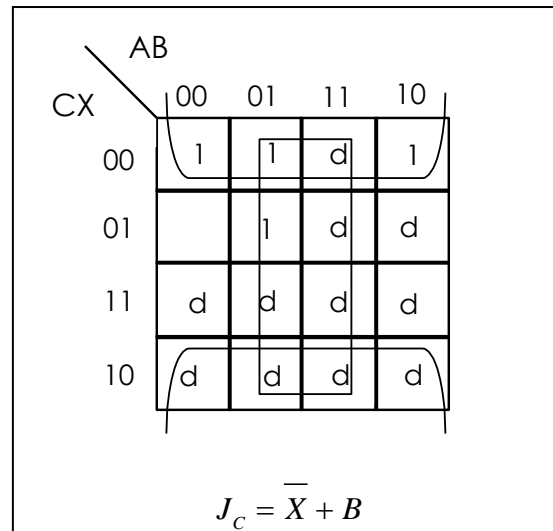
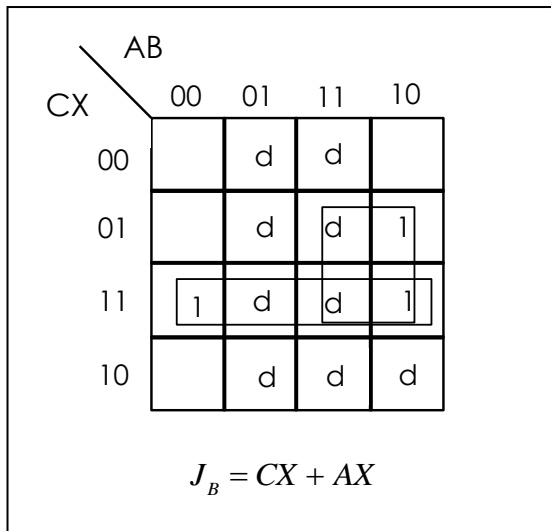
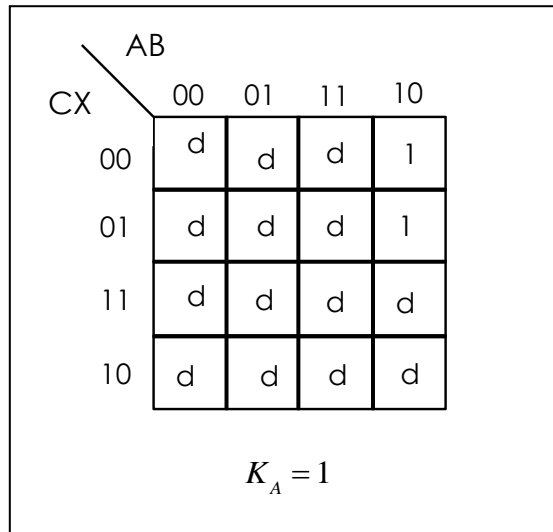
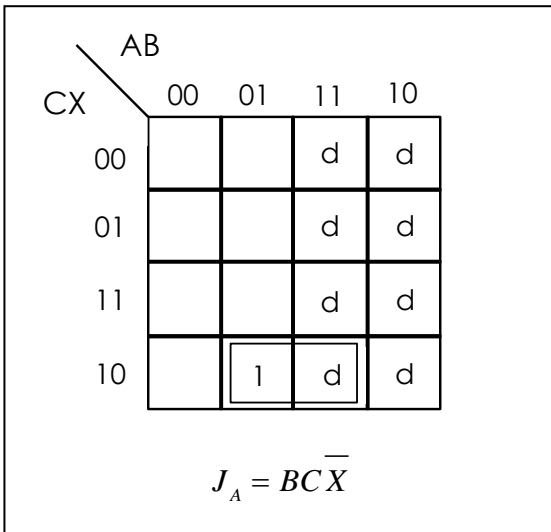
Per codificare 5 stati occorrono tre flip flop. La codifica è la seguente:
 $S0 \rightarrow 0\ 0\ 0$; ...; $S4 \rightarrow 1\ 0\ 0$. Nel seguito indicheremo ciascun bit della codifica con le lettere A, B, C. L'apice indicherà il bit nell'istante successivo a quello considerato.

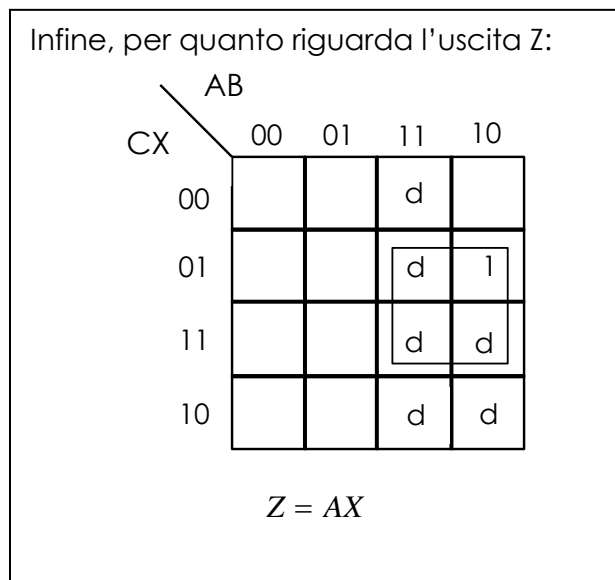
A partire dalla tabella di eccitazione del flip flop JK:

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





ESERCIZIO 2

- Il campo indirizzi avrà ampiezza 11 bit (devo indirizzare 2048 parole). Avremo 32 blocchi in primaria e 8 in cache, con blocchi da 64 parole per indirizzare le quali occorrono 6 bit. Dato che l'indirizzamento è completamente associativo, la suddivisione presenta solo due campi così suddivisi:

< TAG 5 bit > < Offset 6 bit >

2.

	1788	1746	616	459	637	1198	865	1952	74	779	1705	137
B.F.	27	27	9	7	9	18	13	30	1	12	26	2
Hit		X			X							

Blocco 0	Blocco 1	Blocco 2	Blocco 3	Blocco 4	Blocco 5	Blocco 6	Blocco 7
1728...1791	576...639	448...511	1152...1215	832...895	1920...1983	64...127	768...831
1664...1727							
1664...1727	576...639	128...191	1152...1215	832...895	1920...1983	64...127	768...831

ESERCIZIO 3

indirizzi Per implementare il codice utilizziamo anche il registro \$8 per l'offset logico del vettore v e \$9 per il relativo offset fisico e il valore $v[i]$:

```
ricerca:  addi $29, $29, -8
          sw $8, 0($29)
          sw $9, 4($29)
          move $8, $0
          move $6, $0
          lw $9, 0($4)
do:       beq $9, $0, exit0
          beq $9, $5, exit1
          addi $8, $8, 1
          muli $9, $8, 4
          add $9, $9, $4
          1 lw $9, 0($9)
          j do
exit1:    addi $6, $6, 1
exit0:    lw $9, 4($29)
          lw $8, 0($29)
          addi $29, $29, 8
          jr $31
```

ESERCIZIO 4

1)

Soluzione

1.

- a. Deve venire rispettata la condizione:

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

dove K è il numero di bit di controllo inseriti e N è il numero di bit della stringa di origine. Essendo $N+K=12$, si ha $K=4$ e $N=8$.

- b. 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	0	0	1	1	0	1	0	1	1	1	1

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. I corrispondenti bit di errore si ottengono così:

$$e_0 = c_0 \oplus b_0 \oplus b_1 \oplus b_3 \oplus b_4 \oplus b_6 = 0 \oplus 0 \oplus 1 \oplus 1 \oplus 1 \oplus 1 = 0$$

$$e_1 = c_1 \oplus b_0 \oplus b_2 \oplus b_3 \oplus b_5 \oplus b_6 = 0 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 1 = 1$$

$$e_2 = c_2 \oplus b_1 \oplus b_2 \oplus b_3 \oplus b_7 = 1 \oplus 1 \oplus 0 \oplus 1 \oplus 1 = 0$$

$$e_3 = c_3 \oplus b_4 \oplus b_5 \oplus b_6 \oplus b_7 = 0 \oplus 1 \oplus 1 \oplus 1 \oplus 1 = 0$$

Poiché il vettore risultante 0010 non è nullo, vi è un errore nella stringa di 12 bit e precisamente nella posizione indicata dal vettore di errore tradotto in notazione decimale. Il bit sbagliato nella stringa codificata è quindi il secondo (c_1), che essendo però un bit di controllo non influisce sulla stringa di origine, che può essere dunque correttamente estratta, a partire dal bit meno significativo: 01011111.