

PROVA SCRITTA DEL MODULO DI
CALCOLATORI ELETTRONICI
NUOVO E VECCHIO ORDINAMENTO DIDATTICO (5-7 CFU)
15 luglio 2015

NOME:

COGNOME:

MATRICOLA:

CFU:

MOTIVARE LA SOLUZIONE PROPOSTA A CIASCUNO DEGLI ESERCIZI SVOLTI

ESERCIZIO 1 (7 punti)

Progettare una rete sequenziale che riconosca la sequenza "SOS" dove le lettere 'S' e 'O' sono rappresentate rispettivamente dalle coppie di bit 00 e 11. Si richiede:

1. (1 punto) il numero di ingressi e di uscite della rete;
2. (3 punti) il grafo degli stati e la tabella delle transizioni;
3. (3 punti) il calcolo delle forme minime delle variabili di eccitazione dei flip flop con le mappe di Karnaugh. Si usino flip flop JK. Scrivere anche la rete combinatoria per l'uscita Z.

ESERCIZIO 2 (7 punti)

E' data una gerarchia di memorie cache-primaria. La memoria primaria è di 512 B mentre la cache è di 64 B. E' possibile indirizzare il singolo byte, e la memoria primaria è suddivisa in blocchi di dimensione d.

1. (3 punti) Indicare, specificando l'ampiezza e la funzione dei diversi campi, come vengono interpretati gli indirizzi di memoria primaria secondo il metodo di indirizzamento associativo su insiemi a due vie, al variare di d.
2. (4 punti) Porre in grafico il valore di H_c in funzione di d, ipotizzando la cache inizialmente vuota, nel caso di chiamata alle parole di indirizzo, espresso in decimale, da 0 a 31, e da 256 a 287, in quest'ordine, per due volte consecutive.

ESERCIZIO 3 (8 punti)

Implementare una procedura Assembly MIPS chiamata "fusione" che, dati l'indirizzo iniziale di due vettori v (in \$4) e w (in \$5), di dimensione N in \$6, generi un nuovo vettore z a partire da un indirizzo presente in \$7, tale che, per $i=0, \dots, N-1$, $z[2i]=v[i]$ e $z[2i+1]=w[i]$.

ESERCIZIO 4 (6 punti)

Sia dato un campo di 14 bit per la rappresentazione dei numeri in virgola mobile. Considerare i seguenti due formati:

- Bit di segno, esponente a 3 bit in eccesso 3, mantissa frazionaria in segno e valore con normalizzazione 1.M.
 - Bit di segno, esponente a 5 bit in eccesso 16, mantissa frazionaria in segno e valore con normalizzazione 1.M.
1. (3 punti) Dire se il valore 135.75, espresso in decimale, sia rappresentabile per entrambi i formati. Rappresentarlo ed esprimere l'eventuale perdita di precisione in percentuale, dove la precisione è calcolata come $p=100 \cdot (\text{valore rappresentato})/(\text{valore da rappresentare})$.
 2. (3 punti) Indicare una possibile soluzione per poter rappresentare il valore dato al punto precedente con precisione $p=100\%$, agendo sul formato a disposizione.

ESERCIZIO 5 (5 punti)

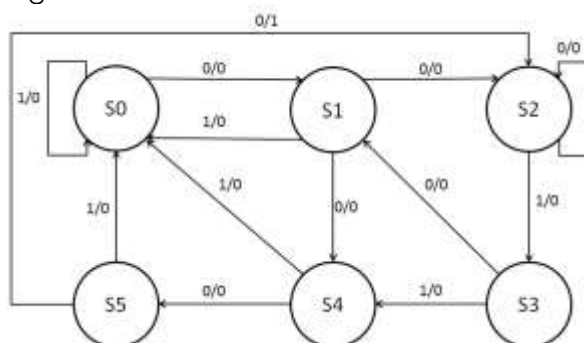
Si rappresentino i seguenti 4 bit **0111** nelle intersezioni di 3 cerchi come visto a lezione mediante l'utilizzo dei diagrammi di Venn. Si voglia poi codificare la precedente sequenza di 4 bit con il codice di Hamming.

- 1) Si calcolino i bit di controllo e si collochino nelle opportune posizioni nei cerchi.
- 2) Se due dei quattro bit d'informazione diventano errati, è possibile individuarne la posizione e correggerli? Motivare chiaramente la risposta.

ESERCIZIO 1**Soluzione.**

Sulla base delle informazioni fornite dal testo, la sequenza "SOS" coincide con la sequenza binaria "001100". Basta dunque un unico flusso di bit in entrata ed un'unica uscita che verrà posta ad 1 al riconoscimento della sequenza ottenuta.

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	S2/0	S0/0
S2	S2/0	S3/0
S3	S1/0	S4/0
S4	S5/0	S0/0
S5	S2/1	S0/0

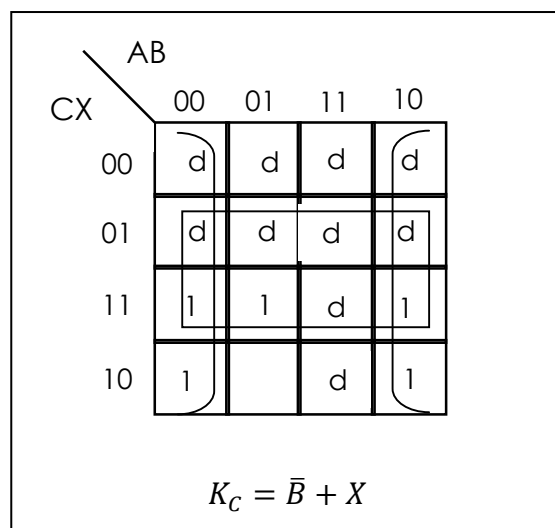
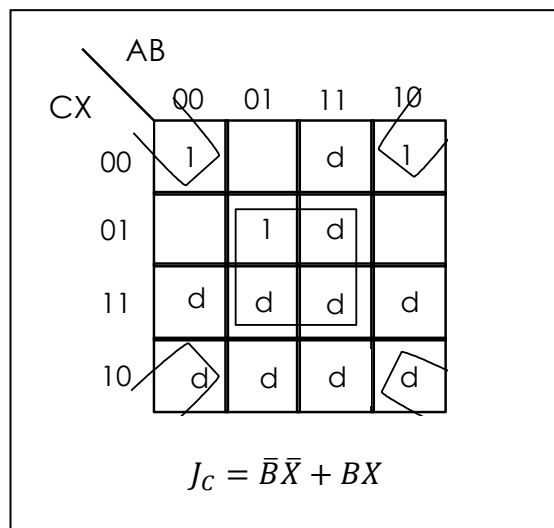
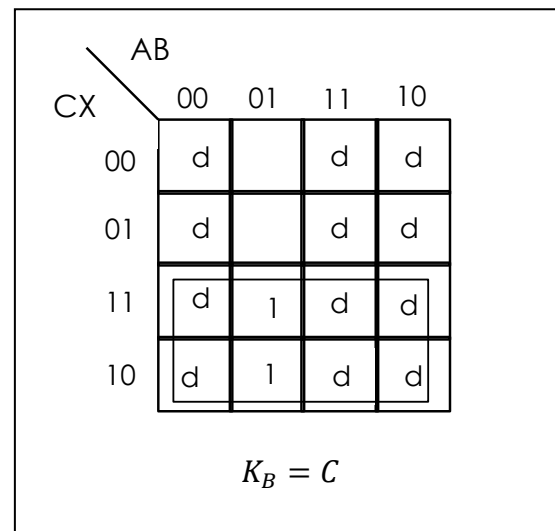
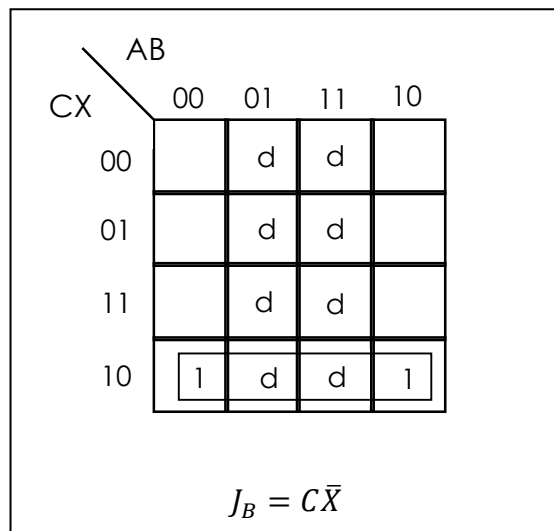
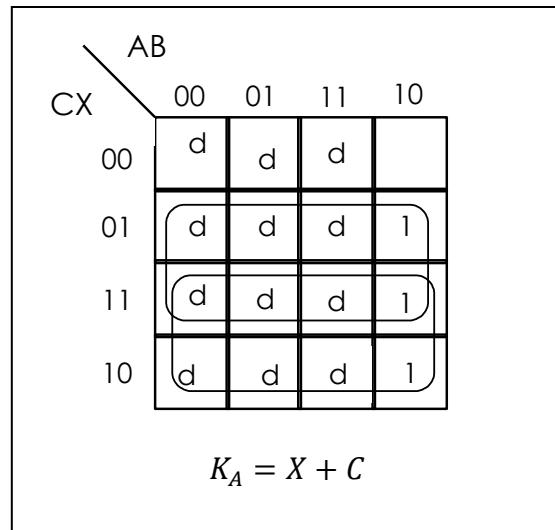
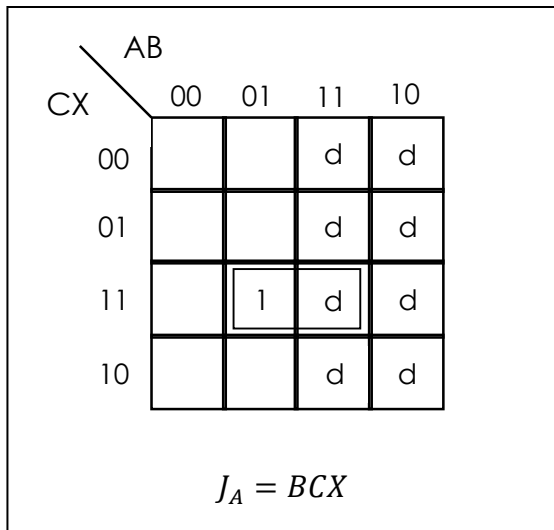
Per codificare 5 stati occorrono tre flip flop. La codifica è la seguente: S0 → 0 0 0; ...; S5 → 1 0 1. Nel seguito indicheremo ciascun bit della codifica con le lettere A, B, C. L'apice indicherà il bit nell'istante successivo a quello considerato.

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	1	1	D	0	D	1	0
0	0	1	1	0	0	D	0	0	D	0	D	1	0
0	1	0	0	0	0	D	1	D	0	0	0	D	0
0	1	0	1	0	0	D	1	D	0	1	1	D	0
0	1	1	0	0	0	D	0	D	1	1	D	0	0
0	1	1	1	1	1	D	0	D	1	0	D	1	0
1	0	0	0	1	D	0	0	0	D	1	1	D	0
1	0	0	1	0	D	1	0	0	D	0	0	D	0
1	0	1	0	0	D	1	1	1	D	0	D	1	1
1	0	1	1	0	D	1	0	0	D	0	D	1	0
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, l'uscita $Z = A\bar{B}C\bar{X}$.

ESERCIZIO 2

Soluzione.

1.

Memoria indirizzabile $512 \text{ B} = 2^9 \text{ B} \rightarrow 9$ bit di indirizzamento. La cache è pari a $64 \text{ B} = 2^6 \text{ B} \rightarrow 6$ bit di indirizzamento. La suddivisione di partenza è quindi:

$$\langle \text{TAG } 3 \text{ bit} \rangle \langle \text{Cache Index} + \text{Offset } 6 \text{ bit} \rangle$$

Poiché utilizziamo il metodo associativo su insiemi a due vie, il TAG erode un bit al campo adiacente quindi avremo:

$$\langle \text{TAG } 4 \text{ bit} \rangle \langle \text{S.I. } 5\text{-b bit} \rangle \langle \text{Offset } b \text{ bit} \rangle$$

Con $d = 2^b$, dimensione del blocco/della linea.

I casi estremi sono $b=0$ e $b=5$. Nel primo avremo 32 insiemi di due parole ciascuno, nel secondo avremo un unico insieme costituito da due vie (linee) di 32 parole ciascuna. Tutti gli altri sono casi intermedi.

2.

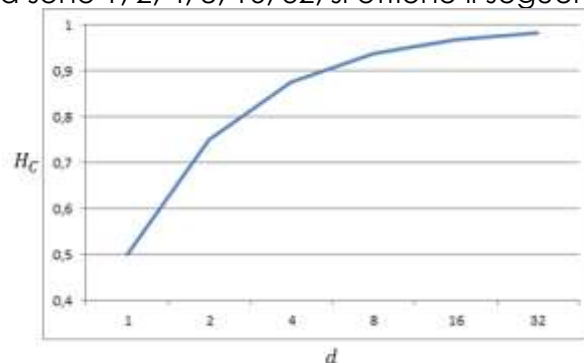
Consideriamo la dimensione generica del blocco, d . Quale che essa sia, a cache inizialmente vuota, avremo sicuramente 1 miss per blocco. Gli altri $d-1$ saranno hit, visto che le chiamate sono su parole localizzate in sequenza. Si tratta inoltre di indirizzi coincidenti a potenze di 2, quindi i blocchi saranno associati alternamente a due vie dello stesso insieme. Trattandosi di 64 chiamate in tutto alla prima iterazione, $d-1$ va semplicemente moltiplicato per il numero di blocchi corrispondenti, che si ottiene dividendo il numero di chiamate per la dimensione del singolo blocco. Alla seconda iterazione, tutte le parole sono presenti in cache. Otteniamo quindi:

- Numero hit prima iterazione: $(d-1) \cdot 64/d$
- Numero hit seconda iterazione: 64.

Si ottiene dunque il seguente valore di H_c in funzione di d :

$$H_c = 0.5 \cdot \left(\frac{d-1}{d} + 1 \right)$$

Poiché i possibili valori di d sono 1, 2, 4, 8, 16, 32, si ottiene il seguente grafico:



ESERCIZIO 3**Soluzione.**

Utilizziamo i seguenti registri:

$\$8 \leftarrow i$; $\$9 \leftarrow 4i, 8i$; $\$1 \leftarrow \&v[i], \&w[i], \&z[2i]$; $\$12 \leftarrow v[i]$; $\$13 \leftarrow w[i]$

```
fusione:    addi $29, $29, -20
            sw $8, 0($29)
            sw $9, 4($29)
            sw $1, 8($29)
            sw $12, 12($29)
            sw $13, 16($29)
            move $8, $0
for:        beq $8, $6, exit
            muli $9, $8, 4
            add $1, $9, $4
            lw $12, 0($1)
            add $1, $9, $5
            lw $13, 0($1)
            muli $9, $9, 2
            add $1, $9, $7
            sw $12, 0($1)
            sw $13, 4($1)
            addi $8, $8, 1
            j for
exit:       lw $8, 0($29)
            lw $9, 4($29)
            lw $1, 8($29)
            lw $12, 12($29)
            lw $13, 16($29)
            addi $29, $29, 20
            jr $31
```

ESERCIZIO 4

Soluzione.

1.

Le due rappresentazioni impongono la seguente suddivisione in campi:

< Bit segno 1 > < Esponente 3 bit (eccesso 3) > < Mantissa 10 bit >

< Bit segno 1 > < Esponente 5 bit (eccesso 16) > < Mantissa 8 bit >

Per sapere se il numero è rappresentabile nei due formati, trasformiamolo in virgola mobile base due:

$$135.75 = 10000111.11 \rightarrow 1.00001111 \cdot 2^7$$

Da questa espressione si deduce subito che esso non è rappresentabile con il primo formato che prevede un esponente massimo pari a 4: troppo pochi bit a disposizione per l'esponente, nonostante i nove bit della mantissa siano virtualmente memorizzabili.

Alla stessa conclusione si giunge calcolando il valore assoluto massimo rappresentabile, pari a: $(2-2^{-10}) \cdot 2^4$, evidentemente inferiore a quello fornito dal problema.

Il valore 7 è invece rappresentabile con il secondo formato in esponente (max pari a $(2-2^{-8}) \cdot 2^{15}$), tuttavia il numero di bit in mantissa è solo 8 per cui avremo una perdita di precisione. Il numero infatti è rappresentato nel seguente modo:

< Bit di segno > < Esponente: 10111 > < Mantissa: 00001111 >

Si perde l'ultimo bit della mantissa. Il valore effettivamente rappresentato è 135.5 con una perdita percentuale di precisione pari a: $100 \cdot (135.5 - 135.75) / 135.75 = -0.18\%$.

2.

Per rappresentare con massima precisione il valore dato necessitiamo di un formato in cui sia mantissa che esponente siano memorizzabili. A questo si può giungere considerando un numero di bit dell'esponente pari a 4, ed utilizzando ad esempio un eccesso pari a 7 o 8 (al massimo). Questa riduzione dell'esponente, rispetto al secondo formato, consente alla mantissa un bit in più. Il formato suggerito è dunque:

<Bit di segno> <Esponente a 4 bit (max eccesso 8)> <Mantissa 9 bit>

ESERCIZIO 5

Si veda la figura. Il codice di Hamming può individuare e correggere solo errori singoli, come si può facilmente verificare dalla figura. Nel caso di due errori non è possibile individuare la posizione degli errori. Si lascia allo studente il completamento dell'esercizio.

