

PROVA SCRITTA DEL CORSO DI
CALCOLATORI ELETTRONICI
NUOVO ORDINAMENTO DIDATTICO (5 CFU)
5 Luglio 2011

NOME:

COGNOME:

MATRICOLA:

ESERCIZIO 1 (9 punti)

Progettare una rete sequenziale che presenti un ingresso X e un'uscita Z posta a 1 ogni volta che viene riconosciuta la sequenza 10111. Si richiede:

1. (3 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.
3. (2 punti) Realizzare un flip flop T a partire da un flip flop JK.

ESERCIZIO 2 (9 punti)

E' data una gerarchia di memorie cache-primaria. La memoria primaria è di 512 KB mentre la cache è di 64 KB. E' possibile indirizzare il singolo byte, e la memoria primaria è suddivisa in blocchi di 32 B.

1. (3 punti) Indicare, sapendo che l'indirizzo della prima parola è pari a 0, l'indirizzo della prima e dell'ultima parola del blocco di memoria primaria con block frame pari a 16.
2. (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, set-associativo a otto vie, e diretto.
3. (3 punti) Ipotezzando la cache piena, indicare in quale linea di cache viene allocato il blocco indicato nel punto 1 con i metodi di indirizzamento esaminati nel punto 2. Si indichi e descriva, dove necessario, almeno un algoritmo di rimpiazzamento.

ESERCIZIO 3 (9 punti)

Implementare una procedura Assembly MIPS che, dati l'indirizzo iniziale di un vettore v (in \$4) e due indici i e j (rispettivamente in \$5 e in \$6), scambi v(i) con v(j) solo se v(i) <> v(j) (cioè se gli elementi corrispondenti ai due indici sono diversi).

In altri termini, il codice MIPS può implementare la seguente funzione C:

```
void elabora(int *v, int i, int j)
{
    if (v[i] != v[j]) {
        int tmp = v[i];
        v[i] = v[j];
        v[j] = tmp;
    }
}
```

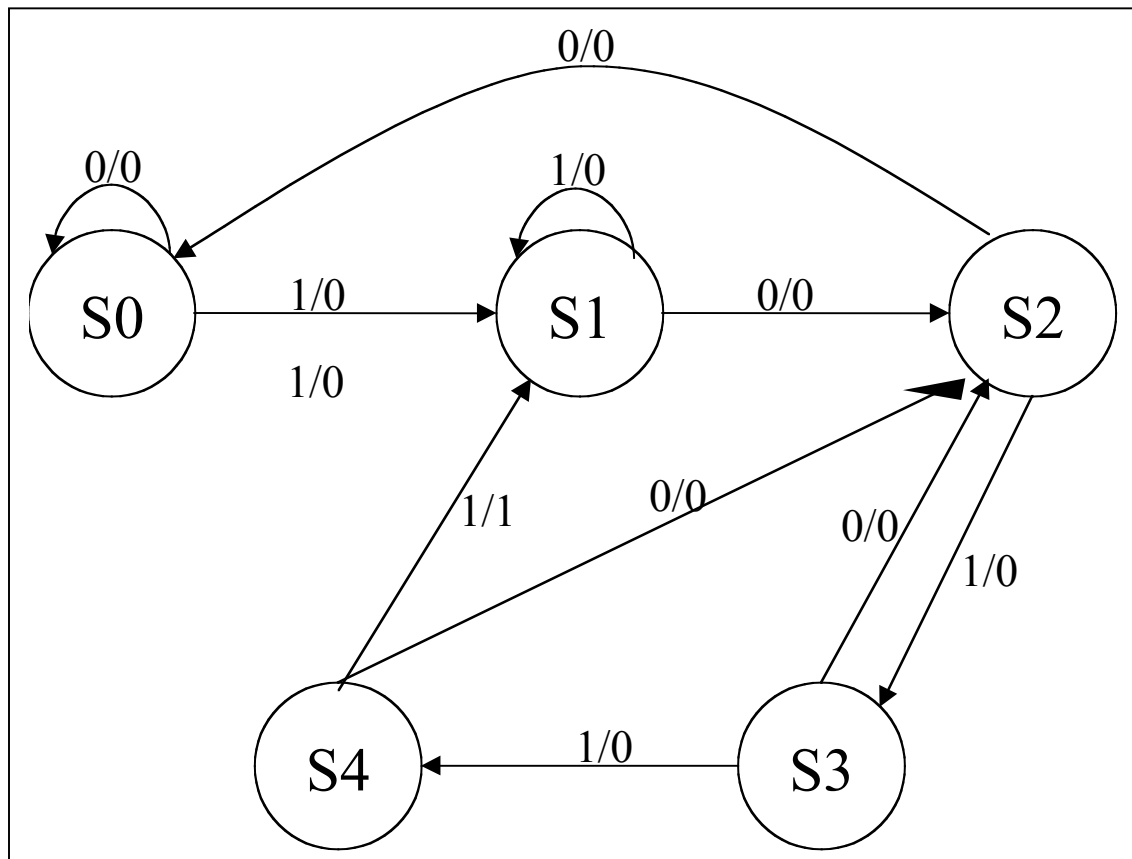
ESERCIZIO 4 (6 punti)

I trasferimenti di parole a/dalla memoria di un calcolatore sono codificate utilizzando il codice di Hamming. Si consideri la stringa di 12 bit 011001111110 (il bit meno significativo è a sinistra), risultata della codifica di una parola di N bit secondo il codice di Hamming.

1. (3 punti) Calcolare N, supponendo di aver fatto uso del numero minimo di bit di controllo necessario per una stringa di 12 bit;
2. (3 punti) scrivere la parola di N bit a partire dalla stringa data.

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

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	0	0	D	0
0	0	0	1	0	0	D	0	0	D	1	1	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	1	D	0	0
0	1	0	0	0	0	D	0	D	1	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	1	D	0	0	D	1	0
0	1	1	1	1	1	D	0	D	1	0	D	1	0
1	0	0	0	0	D	1	1	1	D	0	0	D	0
1	0	0	1	0	D	1	0	0	D	1	1	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

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

$J_A = BCX$

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

$K_A = 1$

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

$$J_B = C\bar{X} + A\bar{X}$$

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

$$K_B = \bar{C} \cdot \bar{X} + CX$$

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

$$J_C = X$$

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

$$K_C = B + \bar{X}$$

Infine, per quanto riguarda l'uscita Z:

$$Z = A \cdot \bar{B} \cdot \bar{C} \cdot X$$

Volendo utilizzare anche i don't care:

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

$$Z = AX$$

Per realizzare un flip flop T da un JK, è sufficiente connettere fra loro gli ingressi J e K, come si può vedere confrontando la tabella delle transizioni del JK con quella del T:

	J	K	Q	Q'	T	Q	Q'
{	0	0	0	0	0	0	0
	0	0	1	1	0	1	1
	0	1	0	0	1	0	1
	0	1	1	0	1	1	0
	1	0	0	1			
	1	0	1	1			
{	1	1	0	1			
	1	1	1	0			

ESERCIZIO 2

Soluzione

1. Indirizzo della prima parola: $\text{Block Frame} * \text{DimBlocco} = 16 * 32 = 512$. Indirizzo dell'ultima = Indirizzo della prima + DimBlocco - 1 = $512 + 32 - 1 = 543$.
2. Memoria indirizzabile 2^{19} B \rightarrow 19 bit di indirizzamento, di cui 5 per l'offset (i blocchi sono di 2^5 B).

Nel caso del metodo associativo, si ha:

< Block Frame (TAG) 14 bit > < Offset 5 bit >

La cache è invece formata da 2^{16} B, da cui

< TAG 3 bit > < C.I. 11 bit > < Offset 5 bit >

Infine, nel caso del metodo set-associativo a otto vie:

< TAG 6 bit > < C.I. 8 bit > < Offset 5 bit >.

3. Metodo associativo: a meno che in una delle linee della cache non sia presente un blocco con medesimo TAG, il blocco dovrà essere allocato nel blocco soddisfacente ad esempio l'algoritmo di rimpiazzamento FIFO, per il quale il primo blocco da rimpiazzare è il primo ad essere stato allocato in cache.

Metodo diretto: è necessario calcolare il resto della divisione $16/2^{11}$. Tale operazione da evidentemente 16. Quindi il blocco 16 di primaria viene allocato, sostituendo il blocco precedente, se con TAG diverso, nella linea 16 di cache.

Metodo set-associativo: è necessario calcolare il resto della divisione $16/2^8$. Anche in questo caso il valore ottenuto è 16, che però corrisponde all'indirizzo dell'insieme. Il blocco 16 sostituirà, a meno di hit, uno degli otto blocchi allocati in altrettante linee, ad esempio con strategia FIFO.

ESERCIZIO 3**Soluzione**

$\$9 \leftarrow v[i]; \$10 \leftarrow v[j]$

```
swap:      addi $29, $29, -16
           sw $9, 0($29)
           sw $10, 4($29)
           sw $5, 8($29)
           sw $6, 12($29)

           muli $5, $5, 4
           add $5, $5, $4      #indirizzo di v[i] in $5
           muli $6, $6, 4
           add $6, $6, $4      #indirizzo di v[j] in $6
           lw $9, 0($5)
           lw $10, 0($6)
           beq $9, $10, exit   #if v[i]==v[j] exit
           sw $9, 0($6)
           sw $10, 0($5)

exit:      lw $9, 0($29)
           lw $10, 4($29)
           lw $5, 8($29)
           lw $6, 12($29)
           addi $29, $29, 16

           jr $31
```

ESERCIZIO 4**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.