

PROVA SCRITTA DEL CORSO DI
CALCOLATORI ELETTRONICI
NUOVO ORDINAMENTO DIDATTICO
19 Gennaio 2010

NOME:

COGNOME:

MATRICOLA:

ESERCIZIO 1 (9 punti)

1. (6 punti) Progettare un flip flop JK a partire da un flip flop D. Disegnare il circuito finale indicando chiaramente la funzione di transizione dello stato, il numero di stati, la funzione di uscita.
2. (3 punti) Spiegare in modo chiaro e sintetico la differenza tra una rete sequenziale sincrona e una asincrona.

ESERCIZIO 2 (9 punti)

Si consideri una memoria primaria costituita da 128 parole e una memoria cache costituita da 32 parole. Il metodo di indirizzamento della cache sia quello completamente associativo con blocchi di 8 parole.

1. (2 punto) Spiegare, precisando il significato e la funzione dei diversi campi, come vengono interpretati gli indirizzi logici per recuperare l'informazione contenuta nella cache.
2. (5 punti) Si considerino le seguenti chiamate ad altrettante parole (indirizzi espressi in decimale): 125, 58, 60, 113, 70, 27, 59, 111, 5, 92. Si indichi il contenuto della cache, ovvero quali byte occupano i relativi blocchi di cache, dopo l'ultima chiamata, considerando la **LRU** come strategia di rimpiazzamento dei blocchi.
3. (2 punti) Si consideri una gerarchia di memoria a tre livelli costituita da: cache, primaria e disco. Se l'hit ratio di cache è pari a 0.9, l'hit ratio di primaria è pari a 0.95, i tempi di accesso a cache, primaria e disco valgono rispettivamente, 5 nsec, 50 nsec e 5 msec, esprimere il tempo medio di accesso alla gerarchia **in nanosecondi**.

ESERCIZIO 3 (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 001001101110 (il bit meno significativo è a sinistra), risultata della codifica di una parola di N bit secondo il codice di Hamming. **Spiegando bene ogni passo del ragionamento:**

1. (2 punti) 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 4 (9 punti)

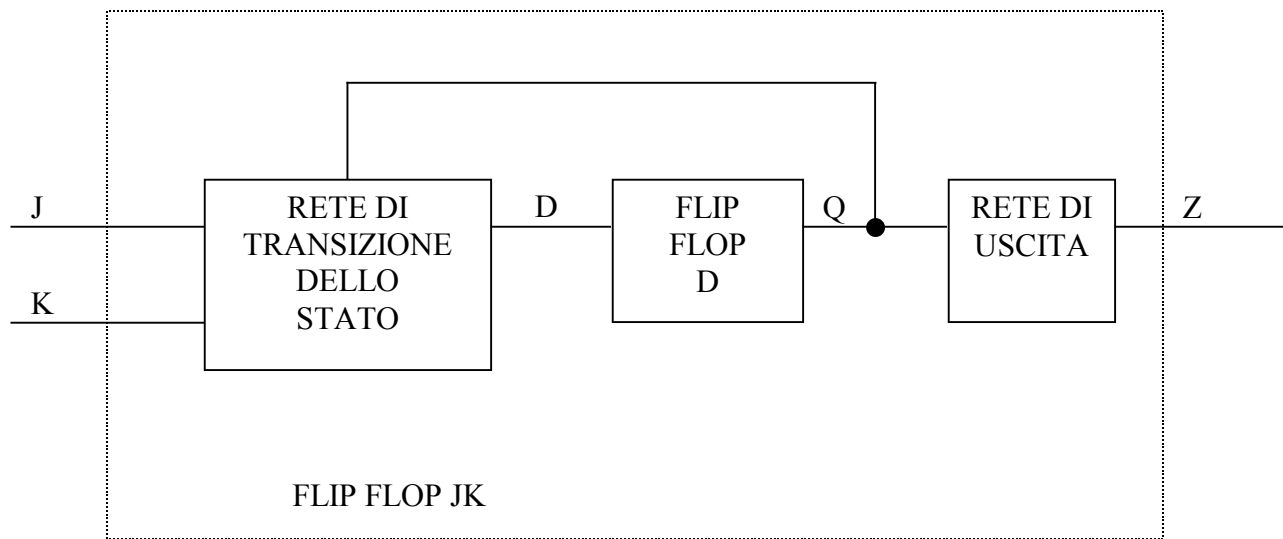
Scrivere una funzione Assembly MIPS, di nome `copy_idx`, che, dati in ingresso due vettori di dimensione N, `idx` e `x`, riordini gli elementi di `x` in un nuovo vettore `y`, secondo gli indici passati in `idx`. Più precisamente, il generico elemento `y[idx[i]]` dovrà contenere l'elemento `x[i]`.

Esempio per $N=8$:
 idx : 1, 2, 3, 4, 0, 5, 7, 6
 x : 21, 32, 45, 51, 10, 67, 88, 73.
Il risultato y conterrà: 10, 21, 32, 45, 51, 67, 73, 88.

Si supponga che `x (&x[0])` sia passato alla funzione in `$4`, `idx (&idx[0])` in `$5`, `y (&y[0])` in `$6` e `N` in `$7`. Si abbia cura di ripristinare il contenuto di tutti i registri temporanei usati nella funzione e dei registri `$4`, `$5`, `$6`, `$7` (modalità callee save).

ESERCIZIO 1**Soluzione**

1. Come in ogni rete logica sequenziale, lo schema del dispositivo è il seguente:

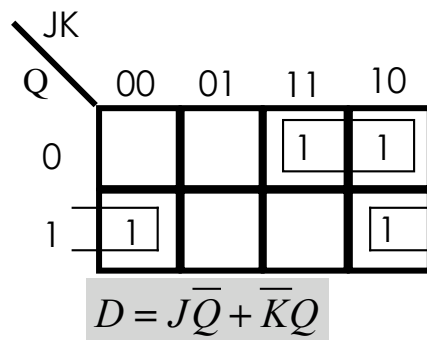


L'uscita di un flip flop JK corrisponde ai valori dello stato (due), per cui si ha $Z=Q$.

Rimane da definire solo la funzione di transizione dello stato, che si ottiene dalla tabella di transizione di un flip flop JK e dalla tabella di eccitazione del flip flop D:

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

Sintetizzando con le mappe di Karnaugh:



2. Vedi dispense del corso.

ESERCIZIO 2**Soluzione**

1.

Con 128 parole, ogni indirizzo è formato da 7 bit così suddivisi secondo il metodo completamente associativo:

< Block frame 4 bit > < Offset 3 bit >

2.

Blocco 0	Blocco 1	Blocco 2	Blocco 3
88 – 95	56 – 63	104 – 111	0 – 7

3.

Con tutti i dati a nostra disposizione è sufficiente valutare la formula:

$$\bar{T} = H_C T_C + (H_P - H_C)(T_P + T_C) + (1 - H_P)(T_D + T_P + T_C)$$

Quindi:

$$\bar{T} = 0.9 \cdot 5 + 0.05 \cdot 55 + 0.05 \cdot 5000000 = 250007.25 ns$$

ESERCIZIO 3**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	0	1	0	0	1	1	0	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):

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

$$e_1 = c_1 \oplus c'_1 = 1$$

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

$$e_3 = c_3 \oplus c'_3 = 1$$

Poiché il vettore risultante 1010 non è nullo, vi è un errore nella stringa di 12 bit data e precisamente nella posizione indicata dal vettore di errore tradotto in notazione decimale. Il bit sbagliato è quindi il decimo (b_5), e la parola corretta è 10111010.

ESERCIZIO 4**Soluzione**

```
copy_idx:
    addi $29, $29, -32
    sw $4, 0($29)
    sw $5, 4($29)
    sw $6, 8($29)
    sw $7, 12($29)
    sw $8, 16($29)
    sw $9, 20($29)
    sw $10, 24($29)
    sw $11, 28($29)

    move $8, $0
for:    beq $8, $7, exit_copy_idx
    lw $9, 0($5)           #carica l'indice in $9
    muli $9, $9, 4         #indice per 4
    lw $11, 0($4)          #$11 <-x[i]
    add $10, $6, $9        #calcola &y[idx[i]]
    sw $11, 0($10)
    addi $4, $4, 4
    addi $5, $5, 4
    addi $8, $8, 1
    j for

exit_copy_idx:
    lw $4, 0($29)
    lw $5, 4($29)
    lw $6, 8($29)
    lw $7, 12($29)
    lw $8, 16($29)
    lw $9, 20($29)
    lw $10, 24($29)
    lw $11, 28($29)
    addi $29, $29, 32

    jr $31
```