

PROVA SCRITTA DEL MODULO DI
CALCOLATORI ELETTRONICI
NUOVO E VECCHIO ORDINAMENTO DIDATTICO (5-6 CFU)
17 giugno 2015

NOME:

COGNOME:

MATRICOLA:

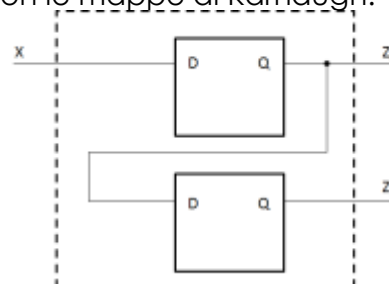
CFU:

ESERCIZIO 1 (7 punti)

(4 punti) Prima parte. Progettare una rete logica combinatoria che effettui la somma di **tutti** i valori di numeri interi non negativi rappresentati con due bit (ovvero compresi tra 0 e 3).

- (2 punti) Quanti ingressi e quante uscite presenta la rete? Motivare la risposta.
- (2 punti) Semplificare le espressioni di ciascuna uscita con le mappe di Karnaugh.

(3 punti) Seconda parte. Si consideri la rete sequenziale a fianco (per semplicità il segnale di sincronismo connesso a ciascun flip flop è stato omesso nel disegno). Scrivendo la tabella delle transizioni dello stato (1 punto) e il grafo degli stati (1 punto), indicare qual è la funzione svolta dalla rete (1 punto), motivando la risposta.



ESERCIZIO 2 (8 punti)

E' data una gerarchia di memorie cache-primaria. La memoria primaria è di 1 KB mentre la cache è di 256 B. E' possibile indirizzare il singolo byte, e la memoria primaria è suddivisa in blocchi di 32 B.

1. (2 punti) Indicare, specificando l'ampiezza e la funzione dei diversi campi, come vengono interpretati gli indirizzi di memoria primaria secondo il metodo associativo su insiemi a N vie, valutando tutte le possibili configurazioni del metodo.
2. (6 punti) Ipotizzando la cache vuota, si consideri la sequenza di chiamate seguenti: dalla parola 0 alla 63, dalla 256 alla 319, dalla 512 alla 575, dalla 768 alla 831, in quest'ordine, per dieci volte consecutive. Indicare quale sia il valore di N che massimizza Hc motivando ogni passaggio del ragionamento, e scrivere a quanto ammonti il corrispondente valore di Hc. Mostrare infine lo stato finale della cache.

ESERCIZIO 3 (6 punti)

Sia dato un disco con le seguenti caratteristiche: velocità di rotazione pari a 6000 giri al minuto, settori da 32 KB, 100 settori per traccia, 65 tracce, tempo di posizionamento da una traccia a quella adiacente pari a 1 ms. Calcolare il tempo medio di lettura di un file da 1 MB sapendo che la testina si trova inizialmente in un punto **qualunque** del disco e che il file è memorizzato per metà su settori contigui della prima traccia e per l'altra metà su settori contigui dell'ultima traccia.

ESERCIZIO 4 (8 punti)

Scrivere una funzione Assembly MIPS `min_1` che, dato l'indirizzo iniziale di un vettore `v` in `$4`, la sua dimensione `N` in `$5`, scriva su un altro vettore `w` di indirizzo iniziale memorizzato in `$6` i valori: `w[i]=v[i]` se `v[i]<v[i+1]`; `w[i]=v[i+1]` altrimenti, per `i=0,...,N-2`.

ESERCIZIO 5 (4 punti)

Scrivere e spiegare il protocollo di arbitraggio centralizzato a festone con due periferiche e tre linee ACK, REQ, GRANT, dal momento di richiesta del bus della prima periferica, e supponendo che durante la trasmissione dati della prima periferica, il bus venga richiesto dalla seconda.

ESERCIZIO 1

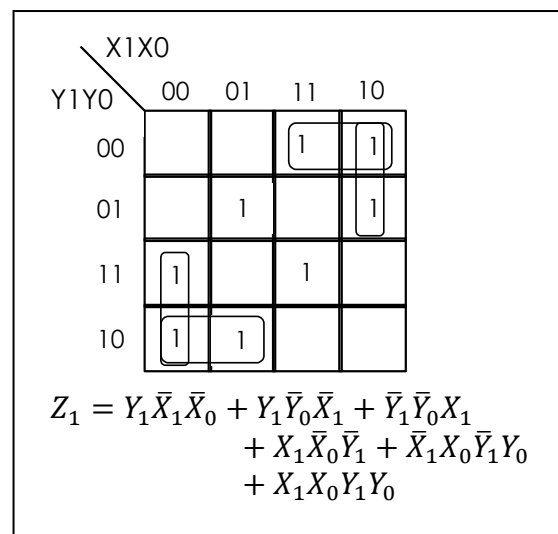
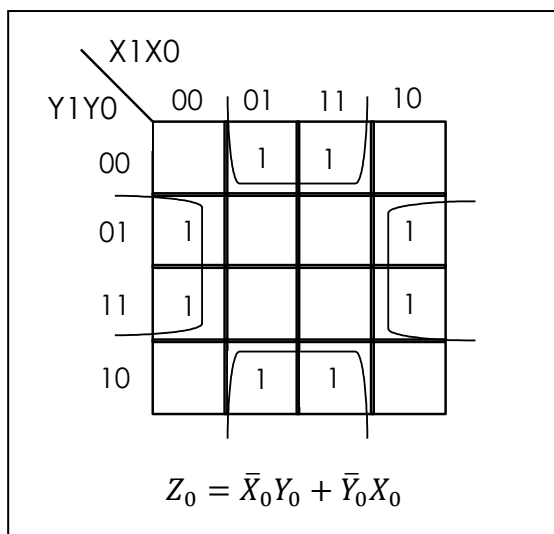
Prima parte.

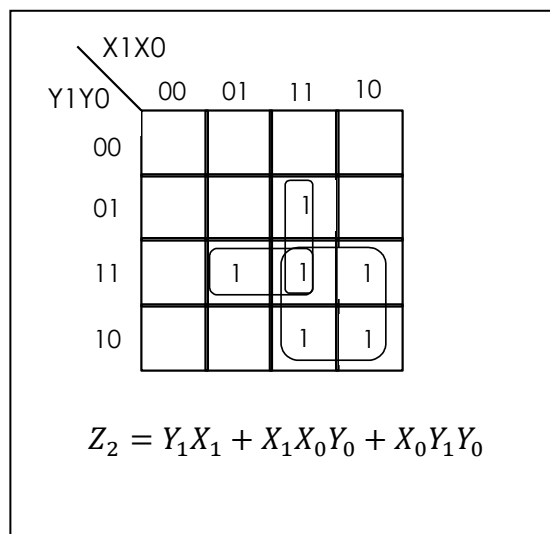
Le possibili configurazioni di numeri interi a due bit sono quattro (0, 1, 2, 3 → 00, 01, 10, 11), quindi un sommatore di valori a due bit è costituito da quattro ingressi, uno per ciascuno dei valori rappresentati tramite coppie di bit X1-X0 e Y1-Y0.

Le varie somme ottenibili, a partire da 2+2, 1+3, 3+1, e così via, non sono rappresentabili con due bit, bensì con tre, quindi le uscite della rete combinatoria saranno tre (Z2-Z1-Z0), definendo la seguente tabella di verità:

X1	X0	Y1	Y0	Z2	Z1	Z0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	1
1	0	1	0	1	0	0
1	0	1	1	1	0	1
1	1	0	0	0	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	1
1	1	1	1	1	1	0

Occorre semplificare le tre mappe di Karnaugh per ciascuna delle uscite:



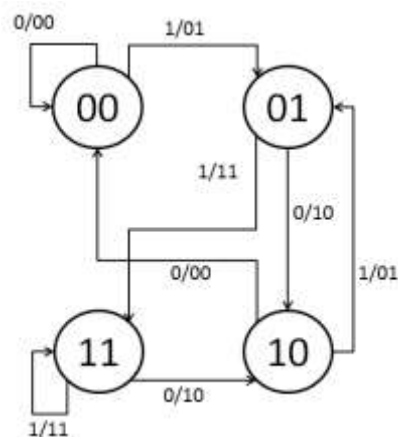


Seconda parte.

Dal momento che le uscite coincidono con lo stato, la tabella delle transizioni è la seguente:

Q1	Q0	X	Z1	Z0
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	0	0
1	0	1	0	1
1	1	0	1	0
1	1	1	1	1

Il diagramma degli stati è il seguente:



Dal grafo degli stati è evidente che lo stato (ovvero l'uscita) prevede lo scorrimento del bit di ingresso dal flip flop direttamente connesso all'ingresso al flip flop adiacente. Si tratta dunque di un **registro a scorrimento a due bit**.

ESERCIZIO 2

1. Memoria indirizzabile 2^{10} B \rightarrow 10 bit di indirizzamento, di cui 5 per l'offset (i blocchi sono di 2^5 B). Il campo restante da 5 bit va distribuito in funzione del metodo di indirizzamento. Nel caso del metodo set-associativo a N vie, possiamo avere quattro possibilità:
- 1) Un insieme, otto vie \rightarrow metodo completamente associativo
 - 2) Due insiemi, quattro vie
 - 3) Quattro insiemi, due vie
 - 4) Otto insiemi, una via \rightarrow metodo diretto (un insieme corrisponde ad una linea)

Tralasciando i casi 1 e 4, si hanno dunque due configurazioni:

< TAG 4 bit > < C.I. 1 bit > < Offset 5 bit >, con N=4

< TAG 3 bit > < C.I. 2 bit > < Offset 5 bit >, con N=2

2. Il primo blocco di chiamate 0-63 corrisponde ad i primi due blocchi di primaria, 0-31 e 32-63. Essi presentano cache index 0 e 1 rispettivamente. Secondo, terzo e quarto blocco di chiamate sono ancora costituiti da due blocchi di primaria con stessi cache index, 0 e 1. E' evidente che per memorizzarli tutti senza rischi di dover sovrascriverli, dobbiamo avere almeno due insiemi in cache per ciascuno dei quali avere almeno quattro vie. In questo modo potremo anche massimizzare H_c , visto che la sequenza è richiamata 10 volte. Delle due configurazioni al punto precedente, l'unica percorribile è dunque la prima, con due insiemi, ognuno dei quali a quattro vie.

La cache a questo punto è satura ma tutte le parole chiamate sono memorizzate. Il numero di hit è dato da 31 (hit per blocco) $\times 8$ (blocchi chiamati) = 248 alla prima invocazione. La sequenza di blocchi viene chiamata per altre 9 volte: tutte le parole sono presenti in cache, determinando $32 \times 8 = 256$ hit in totale per ciascuna delle iterazioni. $H_c = (248 + 9 \times 256) / 10 \times 256 = 0.996875$.

Stato finale della cache:

Insieme 0				Insieme 1			
L0	L1	L2	L3	L4	L5	L6	L7
0-31	256-287	512-543	768-799	32-63	288-319	544-575	800-831

ESERCIZIO 3

Si ottiene subito $Trot = 60/6000 \text{ sec} = 10 \text{ ms}$

Il file è suddiviso in settori da 32 KB per un totale di $2^{20}/2^{15} = 2^5 = 32$ settori.

Tempo latenza: $Tlat = Trot/2 = 5 \text{ ms}$.

Tempo posizionamento da traccia a traccia adiacente: $Tpos = 1 \text{ ms}$.

Tempo di posizionamento massimo: $Tposmax = 64 \text{ ms}$.

Tempo di posizionamento medio: $Tposmedio = Tposmax/2 = 32 \text{ ms}$. (Ragionamento analogo alla stima del tempo di latenza)

Tempo di lettura di un settore: $Tlett = Trot/100 = 0.1 \text{ ms}$.

Poiché il file è parzialmente frammentato per metà su due tracce ma con contiguità di sedici settori per traccia, si avrà una latenza ed un posizionamento solo in corrispondenza della lettura dei primi settori per ogni metà, secondo la seguente tabella:

Settori	Latenza	Posizionamento	Lettura
0	$Tlat$	Tempo medio tra il caso migliore (0 ms) e il caso peggiore (64 ms): $Tposmedio$	$Tlett$
1-15	0	0	$15 * Tlett$
16	$Tlat$	$Tposmax$	$Tlett$
17-31	0	0	$15 * Tlett$

Sommando tutti i tempi indicati si ottiene:

$Tletturablocco = 2 * Tlat + Tposmedio + Tposmax + 32 * Tlett = 10 + 32 + 64 + 3.2 = 109.2 \text{ ms}$

ESERCIZIO 4

Utilizzando i registri col seguente significato:

\$8 → contatore

\$9 → $i*4$

\$2 → $v[i] < v[i+1]$

\$10 → $v[i]$

\$11 → $v[i+1]$

\$12 → $\&w[i], \&v[i]$

\$1 → $\$5 - 1$

```
min_1: addi $29, $29, -28
        sw $8, 0($29)
        sw $9, 4($29)
        sw $10, 8($29)
        sw $11, 12($29)
        sw $2, 16($29)
        sw $12, 20($29)
        sw $1, 24($29)
        lw $10, 0($4)
        subi $1, $5, 1
        move $8, $0
for:     beq $8, $1, exit
        muli $9, $8, 4
        add $12, $9, $4
        lw $11, 4($12)
        add $12, $9, $6
        slt $2, $10, $11
        beq $2, $0, else
        sw $10, 0($12)
        j rip
else:    sw $11, 0($12)
rip:     move $10, $11
        addi $8, $8, 1
        j for
exit:    lw $8, 0($29)
        lw $9, 4($29)
        lw $10, 8($29)
        lw $11, 12($29)
        lw $2, 16($29)
        lw $12, 20($29)
        lw $1, 24($29)
        addi $29, $29, 28
        jr $31
```

Il riferimento è al cap. 7, ed in particolare alla seguente figura dove il protocollo richiesto è perfettamente illustrato:

