

**SOLUZIONI DELLA PROVA SCRITTA DEL CORSO DI
CALCOLATORI ELETTRONICI
NUOVO E VECCHIO ORDINAMENTO DIDATTICO**
24 Settembre 2002

MOTIVARE IN MANIERA CHIARA LE SOLUZIONI PROPOSTE A CIASCUNO DEGLI ESERCIZI SVOLTI

ESERCIZIO 1 (NO: 9 punti – VO: 8 punti)

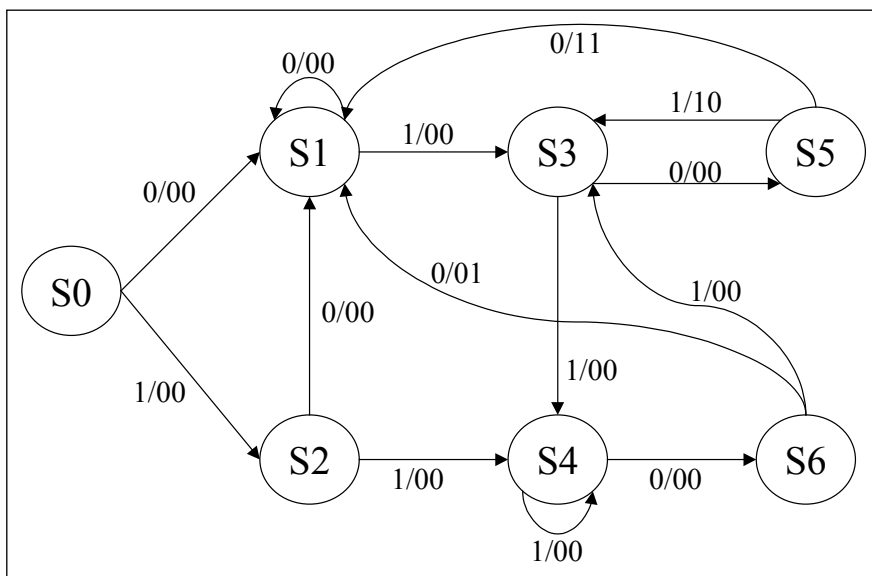
Progettare una rete sequenziale che presenti un ingresso X e due uscite Z1 e Z0 che in corrispondenza delle sequenze 0100, 0101, 1100, assumano rispettivamente la configurazione 11, 10, 01.

Si richiede:

- (NO: 6 punti – VO: 5 punti) il diagramma degli stati, la tabella di flusso e la tabella delle transizioni;
- (3 punti) il calcolo delle forme minime delle variabili di eccitazione dei flip flop e delle uscite con le mappe di Karnaugh. Si usino flip flop D.

Soluzione.

Il diagramma degli stati è il seguente:



La tabella di flusso è data da:

Stato presente	Stato successivo/Uscite	
	X=0	X=1
S0	S1/00	S2/00
S1	S1/00	S3/00
S2	S1/00	S4/00
S3	S5/00	S4/00
S4	S6/00	S4/00
S5	S1/11	S3/10
S6	S1/01	S3/00

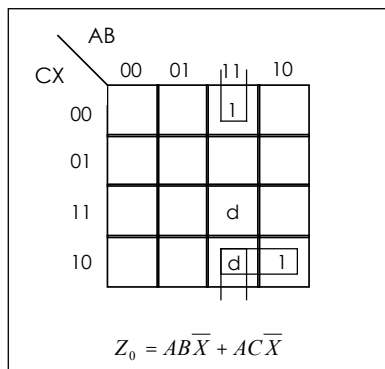
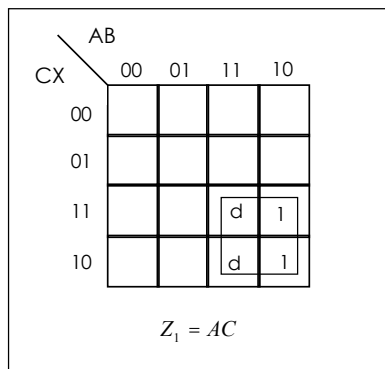
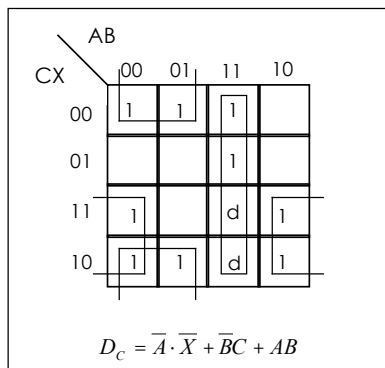
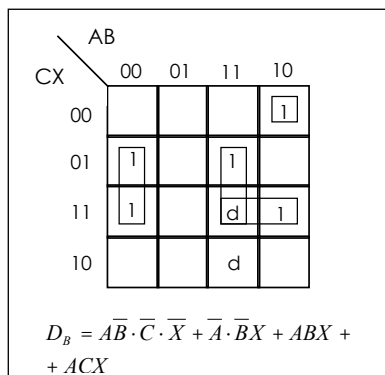
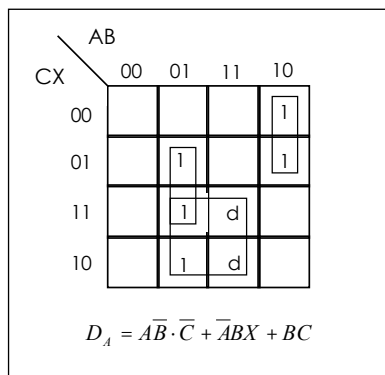
Per codificare 6 stati occorrono tre flip flop. La codifica è la seguente: S0 → 0 0 0; ...; S6 → 1 1 0. Nel seguito indicheremo ciascun bit della codifica con le lettere A, B, C. L'apice indicherà il bit nell'istante successivo a quello considerato.

A partire dalla tabella di eccitazione del flip flop D, si ricava la tabella delle transizioni.

Q	Q'	D
0	0	0
0	1	1
1	0	0
1	1	1

A	B	C	X	A'	DA	B'	DB	C'	DC	Z1	Z0
0	0	0	0	0	0	0	0	1	1	0	0
0	0	0	1	0	0	1	1	0	0	0	0
0	0	1	0	0	0	0	0	1	1	0	0
0	0	1	1	0	0	1	1	1	1	0	0
0	1	0	0	0	0	0	0	1	1	0	0
0	1	0	1	1	1	0	0	0	0	0	0
0	1	1	0	1	1	0	0	1	1	0	0
0	1	1	1	1	1	0	0	0	0	0	0
1	0	0	0	1	1	1	1	0	0	0	0
1	0	0	1	1	1	0	0	0	0	0	0
1	0	1	0	0	0	0	0	1	1	1	1
1	0	1	1	0	0	1	1	1	1	1	0
1	1	0	0	0	0	0	0	1	1	0	1
1	1	0	1	0	0	1	1	1	1	0	0
1	1	1	0	D	D	D	D	D	D	D	D
1	1	1	1	D	D	D	D	D	D	D	D

Ora possiamo disegnare le mappe di Karnaugh.



Per le due uscite, volendo, si può fare a meno dei "don't care".

ESERCIZIO 2 (NO: 8 punti – VO: 7 punti)

Si supponga di disporre di tre macchine: a pila, a uno e a due indirizzi. Per ognuna di queste si abbiano le seguenti istruzioni:

A pila		A un indirizzo		A due indirizzi	
Istruzione	Semantica	Istruzione	Semantica	Istruzione	Semantica
PUSH X	$M[X] \rightarrow \text{push}$	STORE X	$\text{ACC} \rightarrow M[X]$	MOV X1,X2	$M[X1] \rightarrow M[X2]$
POP X	$\text{pop} \rightarrow M[X]$	LOAD X	$M[X] \rightarrow \text{ACC}$	ADD X1,X2	$M[X1] + M[X2] \rightarrow M[X2]$
ADD	$\text{pop} + \text{pop} \rightarrow \text{push}$	ADD X	$\text{ACC} + M[X] \rightarrow \text{ACC}$	DIV X1,X2	$M[X1] / M[X2] \rightarrow M[X2]$
DIV	$\text{pop}(2) / \text{pop}(1) \rightarrow \text{push}$	DIV X	$\text{ACC} / M[X] \rightarrow \text{ACC}$	SUB X1,X2	$M[X1] - M[X2] \rightarrow M[X2]$
SUB	$\text{pop}(2) - \text{pop}(1) \rightarrow \text{push}$	SUB X	$\text{ACC} - M[X] \rightarrow \text{ACC}$		

ACC è il registro accumulatore; $M[X]$ indica il dato nella locazione di memoria X.

- (VO: 5 punti) Facendo attenzione a non sovrascrivere i contenuti iniziali della memoria, si scriva, per ognuna delle tre macchine, la sequenza delle istruzioni necessarie per realizzare la seguente operazione:

$$Z = A - B + C / D$$

(suggerimento: si usi un registro P dove depositare i risultati parziali)

- (solo VO: 2 punti) Spiegare in modo chiaro e sintetico i vari tipi di indirizzamento di un'istruzione.

Soluzione.

- Ecco tre possibili sequenze delle istruzioni che realizzano quanto richiesto:

A pila	A un indirizzo	A due indirizzi
PUSH A	LOAD A	MOV B,P
PUSH B	SUB B	SUB A,P
SUB	STORE Z	MOV D,Z
PUSH C	LOAD C	DIV C,Z
PUSH D	DIV D	ADD P,Z
DIV	ADD Z	
ADD	STORE Z	
POP Z		

- Vedere dispense del corso.

ESERCIZIO 3 (NO: 9 punti – VO: 7 punti)

Si consideri una gerarchia di memoria a tre livelli: cache, primaria, disco. Dopo alcune prove, vengono effettuate le seguenti misurazioni:

- numero totale di accessi alla memoria: 50
- numero di hit in cache: 45
- numero di hit in primaria, essendosi verificato un miss in cache (quindi valutato rispetto al numero di miss in cache verificatisi): 4
- tempo medio di accesso in cache: 4 ns
- tempo medio di accesso in primaria: 100 ns

- tempo medio di accesso a disco: 50 ms
- a) Qual è lo scopo di un'architettura gerarchica di memorie? Come funziona? (NO: 3 punti – VO: 2 punti)
- b) Scrivere e spiegare la formula da usare per il calcolo del tempo medio di accesso alla gerarchia con i dati forniti dal problema. (NO: 3 punti – VO: 2 punti)
- c) Calcolare il tempo medio di accesso alla gerarchia. (3 punti)

Soluzione.

- a) Lo scopo di un'architettura gerarchica di memorie è quello di far intendere all'utilizzatore di disporre di una memoria di grandi dimensioni e veloce. Tale concetto è noto come "virtualizzazione della memoria". Per questo motivo il livello più basso della gerarchia è costituito dalla memoria più piccola e veloce (la cache), e a salire di livello troviamo memorie di dimensioni superiori ma più lente (il disco). Naturalmente, per massimizzare le prestazioni di una simile architettura è molto importante massimizzare il numero di successi al primo livello della gerarchia.

Il funzionamento segue il concetto che si vuole realizzare: quando viene richiesto un accesso alla memoria, il dato viene cercato nella memoria del livello più basso della gerarchia. Ciò richiede un tempo pari al tempo medio di accesso a tale memoria, nel nostro caso la cache. Se non viene trovato nessun dato in cache, si passa al livello successivo, che richiede un ulteriore tempo pari al tempo di accesso alla primaria. Infine, se il dato non viene trovato nemmeno in primaria, si passa alla memoria disco che richiede un tempo aggiuntivo pari appunto al tempo medio di accesso al disco.

- b) In accordo con quanto espresso al punto precedente, la formula per il calcolo del tempo medio di accesso alla gerarchia è la seguente:

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

T_C , T_P , T_D , sono i tempi medi di accesso a cache, primaria e disco.

H_C è l' *hit ratio* di cache. $H_{P\bar{C}}$ è la probabilità di un hit in primaria essendosi verificato un miss in cache.

$(1 - H_C) H_{P\bar{C}}$ è la probabilità di trovare un dato in primaria e di non trovarlo in cache. $(1 - H_C) (1 - H_{P\bar{C}})$ è la probabilità di **non** trovare un dato in primaria e di non trovarlo in cache. Quest'ultimo caso implica la ricerca del dato nella sezione più alta della gerarchia (il disco).

- c) Gli hit ratio necessari si calcolano immediatamente dai dati forniti: $H_C = 45/50$; $H_{P\bar{C}} = 4/5$ (in cinque casi il dato non è stato trovato in cache).

Sostituendo questi e gli altri valori nella formula indicata al punto precedente, si ottiene:

$$\bar{T} = \frac{45}{50} * 4 + \frac{5}{50} * \frac{4}{5} * 104 + \frac{5}{50} * \frac{1}{5} * 50000104 = 3,60 + 8,32 + 1000002,08 = 1000014ns$$

ESERCIZIO 4 (NO: 7 punti – VO: 6 punti)

Si consideri un sistema operativo con gestione della memoria paginata. Ciascun processo ha una dimensione massima di 128 pagine. Le pagine hanno dimensione 1KB. Si abbia ad esempio la seguente situazione:

Tabella delle Pagine Processo 0			
Pagina Virtuale	Pagina Fisica	Bit di Validità	Solo lettura
0000000	0000011	0	0
0000001	0010110	0	0
0000010	1001001	1	1
0000011	1001010	1	1
0000100	1010101	1	1
0000101	0011101	1	0
0000110	0111111	1	0
0000111	1011101	1	0
0001000	1010011	1	0
0001001	0001111	1	0
0001010	0011011	1	0
0001011	0100010	1	0

Tabella delle Pagine Processo 1			
Pagina Virtuale	Pagina Fisica	Bit di Validità	Solo lettura
0000000	0001010	1	0
0000001	0010001	0	0
0000010	1011001	1	0
0000011	1011010	1	0
0000100	0011101	1	0
0000101	0101111	0	0
0000110	0110111	1	0
0000111	0100000	1	0
0001000	0100100	1	0
0001001	0001001	1	0

Tabella delle Pagine Processo 2			
Pagina Virtuale	Pagina Fisica	Bit di Validità	Solo lettura
0000000	0100001	1	1
0000001	0101110	1	1
0000010	0110110	0	1
0000011	0000110	1	1
0000100	1100011	0	1
0000101	1000011	1	1
0000110	0010101	1	1

Il bit di validità indica se la pagina virtuale richiesta è presente (1) o no (0) nella memoria principale.

1- Calcolare la dimensione in bit degli indirizzi virtuali, spiegando come vengono interpretati dal sistema operativo. (NO: 2 punti – VO: 1 punto)

2- (3 punti) Si considerino le seguenti richieste e mostrare: a quale indirizzo fisico corrispondono gli indirizzi virtuali, se l'operazione può essere conclusa con successo o se viene generato un errore (pagina non valida, violazione di protezione) o un page fault:

- 1) (Processo 0) Read 00010110101001101
- 2) (Processo 2) Write 00001010101011001
- 3) (Processo 1) Read 00010011001110011
- 4) (Processo 2) Read 00001001100100110
- 5) (Processo 0) Write 00000110001100001

3- (2 punti) calcolare la dimensione massima di una PMT

Soluzione:

- 1) L'indirizzo virtuale ha dimensione pari almeno a 17 bit e viene interpretato a partire dai bit più significativi nel modo seguente:
7 bit: pagina virtuale
10 bit: posizione all'interno della pagina
- 2) A partire dalla interpretazione del punto 1. Si ottiene:
(a) Indirizzo fisico valido: 01000100101001101
(b) Errore: scrittura su indirizzo fisico 10000110101011001 protetto da scrittura
(c) Indirizzo fisico valido: 00010011001110011
(d) "Page fault"
(e) Errore: scrittura su indirizzo fisico 10010100001100001 protetto da scrittura
- 3) La PMT contiene al massimo 128 'righe', una per ciascuna pagina virtuale. Ciascuna riga contiene 7 bit per la pagina virtuale, 7 bit per la pagina fisica, 2 bit di controllo, per un totale di 16 bit. La dimensione massima di una PMT è dunque pari a $128 \times 16 \text{ bit} = 256 \text{ byte}$.

ESERCIZIO 5 (solo VO: 5 punti)

Un calcolatore con architettura parallela contiene n processori. Una certa applicazione Q richiede un tempo di esecuzione pari a:

$$T(n) = 30 n^{0.6} + 10 \log_2 n.$$

Calcolare lo "speedup" che si ottiene eseguendo l'applicazione Q su tale architettura parallela con $n = 2$.

Soluzione.

Lo "speedup" $S(n) = T(1)/T(n) = 30/(30 n^{0.6} + 10 \log_2 n)$.