

4 Maggio 2000

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

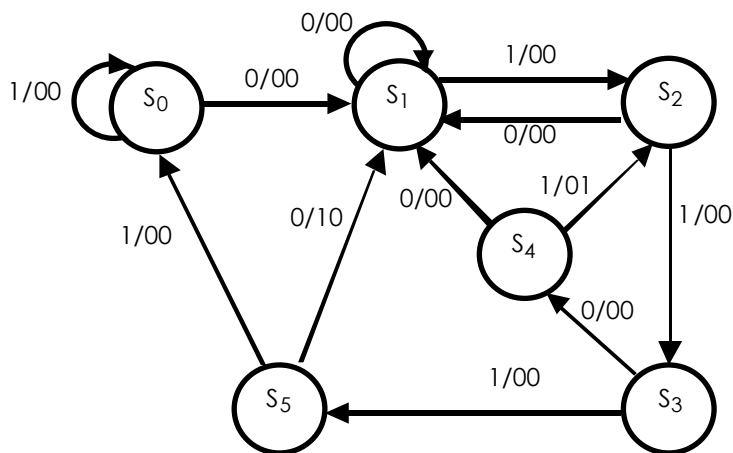
**IMPORTANTE: Leggere la NOTA riportata a pag. 5 di questo documento**

## ESERCIZIO 1

Progettare una rete sequenziale avente un ingresso  $x$  e due uscite  $z_1$  e  $z_2$  che funzioni da riconoscitore di una delle due sequenze 01110 e 01101.

1. Determinare il grafo degli stati.
2. Scrivere la tabella delle eccitazioni dei flip-flop, usando FF-JK.
3. Scrivere le espressioni delle uscite e minimizzare le funzioni delle variabili di eccitazione dei flip-flop usando le mappe di Karnaugh.

**Soluzione:**



### Codifica degli stati

$$S_0 \Rightarrow 000 \quad S_1 \Rightarrow 001 \quad S_2 \Rightarrow 010$$
$$S_3 \Rightarrow 011 \quad S_4 \Rightarrow 100 \quad S_5 \Rightarrow 101$$

Tabella di eccitazione di un flip-flop JK

$Q(t)$	$Q(t+1)$	$J$	$K$
0	0	0	$d$
0	1	1	$d$
1	0	$d$	1
1	1	$d$	0

Tabella di flusso

Stato presente	Stato futuro/uscita	
	x = 0	x = 1
000	001/00	000/00
001	001/00	010/00
010	001/00	011/00
011	100/00	101/00
100	001/00	010/01
101	001/10	000/00
110	ddd/dd	ddd/dd
111	ddd/dd	ddd/dd

(L'apice ' indica la negazione)

*Uscite:*

$$z_1 = ACx'$$

$$z_2 = AC'x$$

### Tabella delle transizioni

A	B	C	x	$A(t+1)$	$J_A K_A$	$B(t+1)$	$J_B K_B$	$C(t+1)$	$J_C K_C$
0	0	0	0	0	0	d	0	0	d
0	0	0	1	0	0	d	0	0	d
0	0	1	0	0	0	d	0	0	d
0	0	1	1	0	0	d	1	1	d
0	1	0	0	0	0	d	0	d	1
0	1	0	1	0	0	d	1	d	0
0	1	1	0	1	1	d	0	d	1
0	1	1	1	1	1	d	0	d	1
1	0	0	0	0	d	1	0	0	d
1	0	0	1	0	d	1	1	1	d
1	0	1	0	0	d	1	0	0	d
1	0	1	1	0	d	1	0	0	d
1	1	0	0	d	d	d	d	d	d
1	1	0	1	d	d	d	d	d	d
1	1	1	0	d	d	d	d	d	d
1	1	1	1	d	d	d	d	d	d

A B

C x

00 01 11 10

00			d	d
01			d	d
11		1	d	d
10		1	d	d

$J_A = BC$

A B

C x

00 01 11 10

00		d	d	
01		d	d	1
11	1	d	d	
10		d	d	

$J_B = AC'x + A'Cx$

A B

C x

00 01 11 10

00	1	1	d	1
01		1	d	
11	d	d	d	d
10	d	d	d	d

$J_C = x' + B$

A B

C x

00 01 11 10

00	d	d	d	1
01	d	d	d	1
11	d	d	d	1
10	d	d	d	1

$K_A = 1$

A B

C x

00 01 11 10

00	d	1	d	d
01	d		d	d
11	d	1	d	d
10	d	1	d	d

$K_B = x' + C$

A B

C x

00 01 11 10

00	d	d	d	d
01	d	d	d	d
11	1		d	1
10		1	d	

$K_C = Bx' + B'x = B \oplus x$

## ESERCIZIO 2

Si abbiano a disposizione 32 bit per rappresentare numeri frazionari in virgola mobile nel formato IEEE 754 con esponente di 8 bit in eccesso 127 e mantissa frazionaria e normalizzata rappresentata in segno e valore. Il calcolatore interpreta la parola di 32 bit nel modo seguente:

Segno	Mantissa	Esponente	Mantissa
-------	----------	-----------	----------

1. Calcolare il massimo e il minimo numero positivo rappresentabile.
2. Sommare le due parole di 32 bit  $3EE00000_{16}$  e  $3D800000_{16}$ , che rappresentano due numeri in virgola mobile nel formato IEEE 754, seguendo i passi dell'algoritmo usato nei calcolatori (senza dimenticare il bit implicito!) e esprimere il risultato in esadecimale.
3. Descrivere gli elementi di una ALU che effettua somme fra numeri in virgola mobile.

## Soluzione

1. Il "range" dell'esponente è:  $-127 \div +128$ . La mantissa è di  $32 - 8 - 1 = 23$  bit.

Max numero:  $(2-2^{-23})2^{128}$ ; Min numero:  $2^{-127}$

2. Il numero  $3EE00000_{16}$  viene interpretato dal calcolatore come

Segno	Esponente	Mantissa
Mantissa		
0	01111101	11000000000000000000000

Il numero  $3D800000_{16}$  viene interpretato dal calcolatore come)

Segno	Esponente	Mantissa
Mantissa		
0	01111011	00000000000000000000000

Algoritmo di somma:

**Confronto esponenti:** La differenza fra l'esponente del numero  $3EE00000_{16}$  (uguale a -2) e l'esponente del numero  $3D800000_{16}$  (uguale a -4) è  $+2_{10}$ . Il primo numero ha esponente maggiore e dunque per sommare i due numeri il secondo numero deve essere denormalizzato in modo da eguagliare gli esponenti.

#### Allineamento mantisse

Per uguagliare gli esponenti, la mantissa del secondo numero va fatta scorrere di due posizioni a destra (divisione per quattro). Pertanto le due mantisse (con esponente comune uguale a -1) diventano:

Primo numero:  $(1).110000000...$  Secondo numero:  $(0).0100000000...$

dove, fra parentesi, è stato evidenziato il valore del bit implicito

Per sommare le mantisse tenendo conto anche del bit implicito entrambi i numeri devono essere ulteriormente divisi per due (scorrimento a destra di una posizione):

Primo numero:  $0.11100000...$  Secondo numero:  $0.00100000$

A questo punto l'esponente comune dei due numeri è -1.

#### Somma delle mantisse:

$0.1110000... + 0.001000 = 1.0000000...$

A questo punto il risultato della somma sarà il seguente:

Segno	Esponente	Mantissa
Mantissa		
0	01111110	000000000000000000000000

La rappresentazione esadecimale della parola di memoria che contiene il risultato della somma è la seguente:  $3F000000_{16}$

3. La "floating point unit" (FPU) della ALU di un calcolatore può essere realizzata usando due unità aritmetiche in virgola fissa, una per l'esponente e una per la mantissa, che vengono accoppiate. L'unità per la mantissa si occupa di eseguire le operazioni aritmetiche di base sulla mantissa (somma algebrica, moltiplicazione e divisione). L'unità per l'esponente è più semplice in quanto deve eseguire solo operazioni di somma algebrica e di confronto fra numeri interi. Il confronto può essere effettuato attraverso una sottrazione degli esponenti.

Per eseguire la somma di due numeri in FP si fa la differenza degli esponenti. Il segno del risultato indica quale dei due esponenti è il più piccolo mentre il modulo indica il numero di scorrimenti verso destra che devono essere eseguiti sulla mantissa del numero più piccolo. Il registro che contiene la mantissa deve dunque essere del tipo a scorrimento. Gli scorrimenti possono essere pilotati da un contatore caricato con la differenza fra gli esponenti: ad ogni scorrimento il contatore viene decrementato finché non viene raggiunto il valore zero.

### ESERCIZIO 3

Tutti i moderni computer hanno due memorie cache:

- una, detta di primo livello (L1), montata sullo stesso chip del processore
- una, detta di secondo livello (L2), esterna al processore.

1. Detti  $H_{L1}$ ,  $H_{L2}$  e  $H_m$  gli "hit ratio" rispettivamente della cache L1, della cache L2 e della memoria principale, scrivere la formula per calcolare il tempo di accesso medio alla gerarchia di memoria spiegando il significato dei singoli termini (Suggerimento: La hit ratio del generico livello  $i$  è definita come la probabilità che il blocco richiesto si trovi nella gerarchia di memoria al livello  $i$ ).
2. Se si considera solo la gerarchia a 3 livelli (cioè se non si considerano i "page fault" che causano accessi al disco), trovare la relazione che deve esistere fra gli hit ratio delle due memorie cache L1 e L2 in modo che il tempo di accesso medio alla gerarchia di memoria sia pari al doppio del tempo di accesso alla memoria più veloce. Determinare l'intervallo di valori "ammissibili" di  $H_{L1}$  e  $H_{L2}$  usando i seguenti dati:  $t_{L1} = 5$  ns;  $t_{L2} = 25$  ns;  $t_m = 50$  ns.

3. Un'istruzione contiene nel campo indirizzi il valore  $317_{10}$ . Nella locazione di memoria all'indirizzo  $317_{10}$  si trova il valore  $1017_{10}$ . Un registro indice contiene il valore  $47589_{10}$ . Calcolare l'indirizzo di memoria in cui si trova l'operando necessario a eseguire l'istruzione, nei seguenti tre casi: a) indirizzamento diretto; b) indirizzamento indiretto; c) indirizzamento con registro indice.

### Soluzione

1. Indicando con  $t_A$  il tempo medio di accesso e con  $t_{L1}$ ,  $t_{L2}$  e  $t_m$  i tempi di accesso delle due cache e della memoria principale, la formula richiesta è la seguente:  

$$t_A = H_{L1}t_{L1} + (H_{L2} - H_{L1})t_{L2} + (H_m - H_{L2})t_m$$
 Il primo termine indica la probabilità che l'accesso in cache abbia successo. Il secondo termine indica invece la probabilità che, essendoci stato un fallimento nell'accesso alla cache, vi sia un successo nella cache L2. Infatti se si considera che i dati contenuti in L1 sono contenuti anche in L2, il termine  $(H_{L2} - H_{L1})$  rappresenta la probabilità di trovare in L2 i dati non trovati in L1. La spiegazione del terzo termine è analoga: indica la probabilità che una richiesta alla gerarchia di memoria non venga soddisfatta da L1 o da L2 ma dalla memoria principale. Tale probabilità è uguale alla differenza  $(H_m - H_{L2})$  perché i dati contenuti nella cache L2 si trovano anche nella memoria principale.
2. In questo caso l'hit ratio della memoria principale deve essere ipotizzato pari a 1 (se non ci sono page fault, tutte le richieste si trovano nella memoria principale). Pertanto sostituendo i dati del problema nell'equazione precedente si ottiene:  

$$2 \cdot 5ns = H_{L1} 5ns + (H_{L2} - H_{L1})25ns + (1 - H_{L2})50ns$$
 Raggruppando a fattor comune avremo:  

$$H_{L1} 20ns + H_{L2} 25ns = 40 ns \Rightarrow H_{L1} = 2 - 5/4 H_{L2}$$
 Dovendo essere  $H_{L1} \leq H_{L2} \leq 1$  (perché la cache di livello 1 è molto più piccola della cache di livello 2), si trova che  

$$H_{L1} = 2 - 5/4 H_{L2} \leq H_{L2} \leq 1 \Rightarrow 8/9 \leq H_{L2} \leq 1$$
 Sostituendo gli estremi nella espressione di  $H_{L1}$  in funzione di  $H_{L2}$  si ottiene:  

$$3/4 \leq H_{L2} \leq 8/9$$
3. a) Indirizzamento diretto: il contenuto del campo indirizzi indica la locazione di memoria in cui si trova l'operando ( $1017_{10}$ ).  
 b) Indirizzamento indiretto: il contenuto del campo indirizzi indica la locazione di memoria in cui trovare l'indirizzo ove risiede l'operando: il contenuto della locazione di memoria all'indirizzo  $1017_{10}$  contiene l'operando.  
 c) indirizzamento con registro indice: l'operando si trova nella locazione di memoria  $47589_{10} + 317_{10} = 47906_{10}$

### ESERCIZIO 4

1. Un bus sincrono presenta le seguenti caratteristiche:
- frequenza di clock: 100 MHz
  - durata di una trasmissione sul bus: 2 cicli di clock
  - ampiezza linea dati: 32 bit
  - ampiezza linea indirizzi: 32 bit.
- Si calcoli la velocità di trasferimento durante una operazione di lettura di un dato dalla memoria, sapendo che la memoria principale ha un tempo di ciclo pari a 70 ns nei seguenti due casi: parole di memoria di dimensione pari a 32 bit; parole di memoria di dimensione pari a 64 bit.
2. Perché nelle architetture Pentium II e UltraSPARC II l'ampiezza della linea dati del bus di memoria è doppia rispetto alla dimensione della parola?

### Soluzione:

1. La durata di un ciclo di clock è pari a  $1/(100 \text{ MHz}) = 10 \text{ ns}$

La lettura su un bus sincrono avviene secondo il protocollo seguente:

- Segnale di READ sulla linea di controllo e contemporaneamente l'indirizzo della locazione in cui risiede il dato sulla linea indirizzi:

**2 cicli di clock = 20 ns**

- Lettura della parola dalla memoria: **70 ns**

cioè 7 cicli di clock

- Trasferimento della parola dalla memoria:

se la parola da leggere ha ampiezza pari a 32 bit

**trasferimento della parola = 20 ns**

**Tempo totale per leggere una parola dalla memoria = 110 ns**

**Velocità di trasferimento =  $4 \text{ B} / 110 \text{ ns} \approx 35 \text{ MB/s}$**

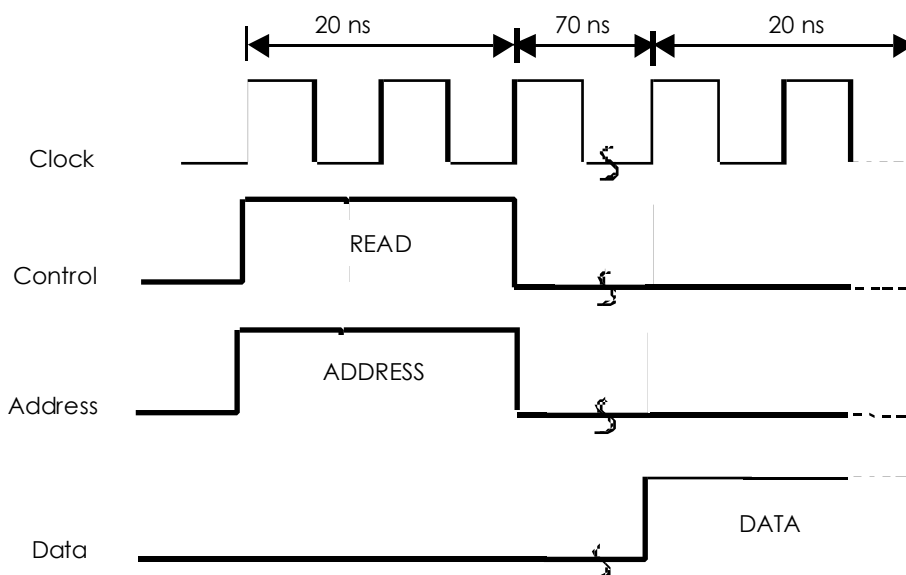
se la parola da leggere ha ampiezza pari a 64 bit occorre effettuare due trasferimenti successivi:

**trasferimento dei primi 32 bit della parola = 20 ns**

**trasferimento dei restanti 32 bit della parola = 20 ns**

**Tempo totale per leggere una parola di 64 bit dalla memoria = 130 ns**

**Velocità di trasferimento =  $8 \text{ B} / 130 \text{ ns} \approx 59 \text{ MB/s}$**



2. Nelle architetture dei calcolatori normalmente il bus di memoria ha ampiezza doppia rispetto alla dimensione della parola della memoria perché le parole non vengono trasferite singolarmente ma a blocchi. Pertanto il trasferimento di blocchi è accelerato quando l'ampiezza della linea dati del bus è maggiore della dimensione della singola parola.

### ESERCIZIO 5

1. Mostrare come si possa realizzare un pipeline aritmetico per effettuare la somma di un vettore formato da numeri in virgola mobile.
2. Descrivere la classificazione delle architetture parallele secondo Flynn e fornire un esempio per ciascuna classe.

### Soluzione

Vedi dispense del corso

NOTA	Esercizio 1	Esercizio 2	Esercizio 3	Esercizio 4	Esercizio 5
Vecchio Ordinamento	dom. 1: 2 punti dom. 2: 2 punti dom. 3: 3 punti	dom. 1: 1 punto dom. 2: 4 punti dom. 3: 2 punti	dom. 1: 3 punti dom. 2: 5 punti dom. 3: assente	dom. 1: 4 punti dom. 2: 2 punti	dom. 1: 3 punti dom. 2: 2 punti
Nuovo Ordinamento	Solo domanda 1: 7 punti	dom. 1: 2 punti dom. 2: 4 punti dom. 3: 3 punti	dom. 1: 4 punti dom. 2: assente dom. 3: 4 punti	dom. 1: 6 punti dom. 2: 3 punti	assente