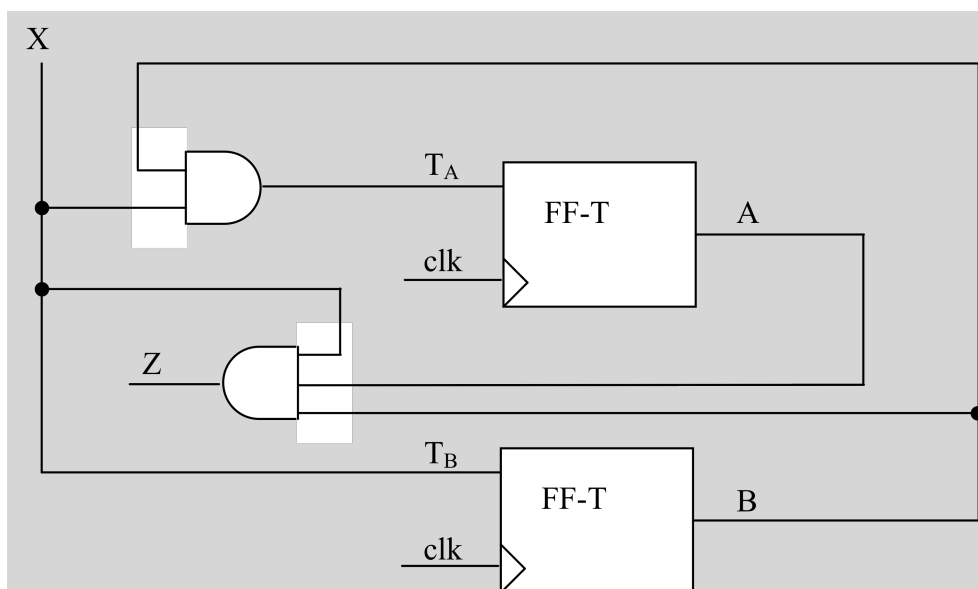


PROVA SCRITTA DEL CORSO DI
CALCOLATORI ELETTRONICI
NUOVO ORDINAMENTO DIDATTICO
 12 Settembre 2007

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

ESERCIZIO 1 (8 punti)

Considerare il circuito in figura (clk indica il clock di sistema, X l'ingresso e Z l'uscita):



1. (6 punti) Scrivere la tabella delle transizioni e il diagramma degli stati;
2. (2 punti) Quale funzione svolge il circuito?

Soluzione

1. Il circuito presenta una rete sequenziale pilotata unicamente dal clock e dagli stati in uscita dei due FF-T, di cui ricordiamo la tabella di eccitazione.

*Tabella di eccitazione
di un flip-flop T*

$Q(t)$	$Q(t+1)$	T
0	0	0
0	1	1
1	0	1
1	1	0

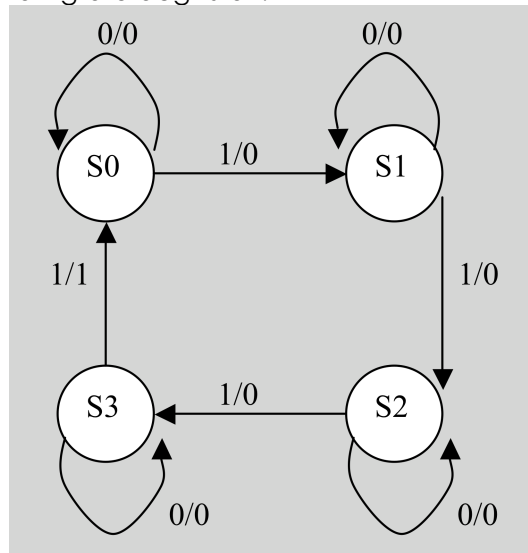
E' immediato scrivere le funzioni logiche di T_A e T_B , ovvero degli ingressi dei due FF-T: $T_A = BX$ e $T_B = X$.
 La tabella di transizione risulta:

A	B	X	A'	T_A	B'	T_B	Z
0	0	0	0	0	0	0	0
0	0	1	0	0	1	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	0	1	0
1	0	0	1	0	0	0	0
1	0	1	1	0	1	1	0
1	1	0	1	0	1	0	0
1	1	1	0	1	0	1	1

La tabella di flusso è data da:

Stato presente	Stato futuro/Uscita	
	X=0	X=1
S0	S0/0	S1/0
S1	S1/0	S2/0
S2	S2/0	S3/0
S3	S3/0	S0/1

Conseguentemente, si disegna il grafo degli stati:



La funzione svolta da questa rete logica è evidentemente un contatore per 4.

ESERCIZIO 2 (10 punti)

Si consideri una gerarchia di memoria composta da una memoria cache di 512 parole ed una memoria principale di 2048 parole. Le parole sono raggruppate in blocchi di 64.

- (4 punti) Illustrare come viene interpretato l'indirizzo di una generica parola nel caso in cui si utilizzino i metodi di indirizzamento diretto o completamente associativo.
- (5 punti) Mostrare lo stato finale della memoria e il relativo cache hit per le due strategie di cui prima, nel caso in cui la sequenza di chiamate sia la seguente:
1788 – 1746 – 616 – 459 – 637 – 1198 – 865 – 1952 – 74 – 779 – 1705 – 137.
(N.B. Si ipotizzi una strategia di rimpiazzamento FIFO dove sia necessario).

Soluzione

- Il campo indirizzi avrà ampiezza 11 bit (devo indirizzare 2048 parole). Avremo 32 blocchi in primaria e 8 in cache.
 - Indirizzamento diretto.
< TAG 2 bit > < Cache Index 3 bit > < Offset 6 bit >
 - Completamente associativo.
< TAG 5 bit > < Offset 6 bit >
- Indirizzamento diretto.

	1788	1746	616	459	637	1198	865	1952	74	779	1705	137
B.F.	27	27	9	7	9	18	13	30	1	12	26	2
C.I.	3	3	1	7	1	2	5	6	1	4	2	2
Hit		X			X							

Blocco 0	Blocco 1	Blocco 2	Blocco 3	Blocco 4	Blocco 5	Blocco 6	Blocco 7
	576...639	1152...1215	1728...1792	768...831	832...895	1920...1983	448...511
	64...127	1664...1727					
		1152...1215					
-	64...127	128...191	1728...1792	768...831	832...895	1920...1983	448...511

- Completamente associativo.

	1788	1746	616	459	637	1198	865	1952	74	779	1705	137
B.F.	27	27	9	7	9	18	13	30	1	12	26	2
Hit		X			X							

Blocco 0	Blocco 1	Blocco 2	Blocco 3	Blocco 4	Blocco 5	Blocco 6	Blocco 7
1728...1791	576...639	448...511	1152...1215	832...895	1920...1983	64...127	768...831
1664...1727							
1664...1727	576...639	128...191	1152...1215	832...895	1920...1983	64...127	768...831

ESERCIZIO 3 (8 punti)

Si scriva il codice Assembler MIPS che implementa la funzione di ordinamento di un vettore V, secondo l'algoritmo selection-sort di cui e' data una possibile implementazione in codice C. Si supponga che l'indirizzo iniziale del vettore V sia contenuto nel registro \$4 e la sua dimensione N nel registro \$5.

```
void selection(double a[], unsigned long N) {  
  
    int i, j, min;  
    double t;  
  
    for (i=0; i < N-1; i++) {  
        min = i;  
        for (j= i + 1; j < N; j++)  
            if (v[j] < v[min])  
                min = j;  
  
        t = v[min];  
        v[min] = v[i];  
        v[i] = t;  
    }  
}
```

Soluzione.

```
$4 ← &v[0];      $5 ← n;      $12 ← n-1;  
$6 ← i;           $7 ← j;      $8 ← &v[j] / v[j];      $9 ← min;  
$10 ← &v[min] / v[min];      $11 ← cond/app ;
```

```
selection:      addi $29, $29, -40          *salvataggio del contesto  
                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)  
                sw $12, 32($29)  
                sw $31, 36($29)  
  
                move $6, $0                * inizializzo i=0  
for1:           subi $12, $5, 1             * nei cicli serve <N per cui calcolo N-1  
                slt $11, $6, $12  
                beq $11, $0, fine           * condizione for1: i>=N-1 esce  
                move $9, $6               * min=i  
                move $7, $6               * inizializzo j=i  
  
for2:           addi $7, $7, 1             * j=j+1 (inizia da i+1)  
                slt $11, $7, $5  
                beq $11, $0, swap           * condizione for2: se j>=N esce  
                muli $8, $7, 4             * $8=j*4  
                add $11, $8, $4            * $11=&v[0]+j*4=&v[j]  
                lw $8, 0($11)             * $8=v[j]  
                muli $10, $9, 4            * $10=min*4  
                add $11, $10, $4           * $11=&v[min]  
                lw $10, 0($11)            * $10=v[min]  
                slt $11, $8, $10  
                beq $11, $0, for2          * salta e cicla se v[j] >= v[min]  
                move $9, $7               * aggiorniamo l'indice del minimo j  
                move $10, $8              * mettiamo in $10 il valore minimo v[min]  
                jr for2
```

<pre> swap: muli \$11, \$6, 4 add \$11, \$11, \$4 lw \$7, 0(\$11) sw \$10, 0(\$11) muli \$11, \$9, 4 add \$11, \$11, \$4 sw \$7, 0(\$11) addi \$6, \$6, 1 jr for1 fine: 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) lw \$12, 32(\$29) lw \$31, 36(\$29) addi \$29, \$29, 40 </pre>	<pre> * \$11=i*4 * \$11=&v[i] * appoggiamo in \$7 il valore v[i] * scriviamo v[min] in &v[i] * \$11=min*4 * \$11=&v[min] * scriviamo v[i] in &v[min] * i++ * ripristino contesto </pre>
--	--

ESERCIZIO 4 (NO: 7 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. Il bit di validità indica se la pagina virtuale richiesta è presente (1) o no (0) nella memoria principale. Si consideri la situazione rappresentata dalle tabelle di cui sotto.

- a) (3 punti) Calcolare la dimensione in bit degli indirizzi virtuali, spiegando come vengono interpretati dal sistema operativo.
- b) (4 punti) Si considerino le seguenti richieste e si mostri a quale indirizzo fisico corrispondono gli indirizzi virtuali e 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

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

Soluzione

- a) 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.
- b) A partire dalla interpretazione del punto 1. Si ottiene:
- 1) Indirizzo fisico valido: 01000100101001101
 - 2) Errore: scrittura su indirizzo fisico 10000110101011001 protetto in scrittura
 - 3) Indirizzo fisico valido: 00010011001110011
 - 4) "Page fault all'indirizzo fisico 00111011100100110
 - 5) Errore: scrittura su indirizzo fisico 10010100001100001 protetto in scrittura.