

08 Giugno 2010

08 Giugno 2010

MATRICOLA:

- ### ESERCIZIO 4 (8 punti)

ESERCIZIO 1

Soluzione

Il circuito è formato da flip-flop di tipo JK, da porte logiche AND. Il circuito è sincronizzato da un segnale di clock che abilita le transizioni di stato nei flip-flop. Gli ingressi ai flip-flop sono infine controllati da un segnale 'count enable'. La tabella di eccitazione di un flip-flop JK è riportata in basso.

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

Ipotesizziamo che all'inizio tutti i flip-flop si trovino allo stato 0 e che il segnale di 'count enable' sia pari a 1. Dopo il primo colpo di clock, lo stato del primo flip flop passa da 0 a 1 ($J = 1, K = 1$), dunque l'uscita A1 è pari a 1. Al secondo colpo di clock lo stato del primo flip flop torna a 0, mentre lo stato del secondo flip flop passa a 1 perché gli ingressi sono entrambi pari a 1 dovuti all'uscita precedente del primo flip flop (N.B. ovviamente la durata del clock dovrà essere tale da consentire la transizione di stato di un flip flop ma non la propagazione del nuovo stato ai flip flop successivi. Questi ultimi invece vedono in ingresso lo stato posseduto dai flip flop a cui sono collegati relativo all'istante precedente). Se proseguiamo nel ragionamento per i successivi 6 colpi di clock, osserveremo che le uscite A3 A2 A1 effettuano le seguenti transizioni: $000 \Rightarrow 001 \Rightarrow 010 \Rightarrow 011 \Rightarrow 100 \Rightarrow 101 \Rightarrow 110 \Rightarrow 111$ (si osservi che il flip-flop A3 commuta solo quando lo stato precedente dei flip-flop A2 e A1 era pari a 1). Al nono colpo di clock tutte le variabili tornano a 0. La rete sequenziale dunque realizza un contatore binario sincronizzato dal clock.

ESERCIZIO 2

Soluzione

1. <Tag 8 bit> <Cache Index 5 bit> <Offset 3 bit>

2. e 3.

113B → 0001 0001 | 0011 1 | 011 → Block frame (551)₁₀ → Cache index 7
C334 → 1100 0011 | 0011 0 | 100 → Block frame (6246)₁₀ → Cache index 6
1137 → 0001 0001 | 0011 0 | 111 → Block frame (551)₁₀ → Cache index 6
AAAA → 1010 1010 | 1010 1 | 010 → Block frame (5461)₁₀ → Cache index 21

Se le parole venissero richieste sequenzialmente non si verificherebbe alcun hit.

4. Essendo:

1A1C → 0001 1010 0001 1 | 100,

si ottiene facilmente che gli altri byte contenuti nello stesso blocco sono: 1A18 (offset 000), 1A19 (offset 001), 1A1A (offset 010), 1A1B (offset 011), 1A1D (offset 101), 1A1E (offset 110), 1A1F (offset 111).

ESERCIZIO 3

Soluzione

$\$8 \leftarrow k*4$; $\$9 \leftarrow \&v[0]+k*4$; $\$4$ - $\$6$ vengono copiati in $\$10$ - $\$12$

```
max_vettori: addi $29, $29, -24
              sw $8, 0($29)
              sw $9, 4($29)
              sw $10, 8($29)
              sw $11, 12($29)
              sw $12, 16($29)
              sw $31, 20($29)
              move $10, $4
              move $11, $5
              move $12, $6
              muli $8, $6, 4
              add $9, $8, $4
              addi $6, $6, 1
              lw $4, 0($9)
              beq $5, $6, loadV0 #se k+1 == N, carica v[0] in $5
                                #altrimenti carica v[k+1] in $5
cont:         jal media
              move $7, $6
              move $4, $10
              move $5, $11
              move $6, $12
              lw $8, 0($29)
              lw $9, 4($29)
              lw $10, 8($29)
              lw $11, 12($29)
              lw $12, 16($29)
              lw $31, 20($29)
              addi $29, $29, 24
              jr $31
loadV0:      lw $5, 0($10)
              j cont
```

ESERCIZIO 4

Soluzione.

1. Per rappresentare i numeri nel formato binario in virgola mobile è necessario innanzitutto rappresentare i due numeri nel formato binario in virgola fissa. Trattandosi di due numeri interi, si può procedere ad esempio con il metodo delle divisioni successive (cfr. Soluzioni del compito del 30/11/98). Il numero 193 si rappresenta in binario come 11000001, mentre il numero 104 è rappresentato come 1101000. La rappresentazione in virgola mobile nel formato IEEE 754 è la seguente:

193 (11000001 \Rightarrow 1.1000001 \cdot 2⁷)

Segno	Esponente	Mantissa
Mantissa		
0	10000110	100000100000000000000000

104 (1101000 \Rightarrow 1.101000 \cdot 2⁶)

Segno	Esponente	Mantissa
Mantissa		
0	10000101	101000000000000000000000

2. Algoritmo di somma:

Confronto esponenti: il numero 193 ha esponente maggiore, dunque 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 una posizione a destra (divisione per due). Pertanto le due mantisse (con esponente comune uguale a 7) diventano:

Primo numero: (1).1000001 Secondo numero: (0).1101

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

Somma delle mantisse:

(1).1000001 + (0).101 = (10).0101001

Questa mantissa non è normalizzata. Per normalizzarla nel formato IEEE 754 occorre far scorrere la mantissa di una posizione verso destra e aumentare l'esponente di una unità. Pertanto il risultato della somma verrà rappresentato nel calcolatore come (l'esponente è ora +8):

Segno	Esponente	Mantissa
Mantissa		
0	10000111	001010010000000000000000

Si può facilmente verificare che questo risultato corrisponde alla somma decimale 193+104=297

3. [cap.6, pag 39-40] 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.