

Esercizi di ricapitolazione – parte I

1. Codificare in IEEE754 il numero -37,5.
2. Trasformare in forma SOP (somma di prodotti) la seguente espressione:

$$Y = U_1 + U_2 \{ \neg [U_3 + U_3 (\neg U_2)] \}$$
 E sintetizzarne il circuito. Scrivere poi la funzione implementata in termini di tabella delle verità e le due forme canoniche corrispondenti. [2 + 1 + 4].
3. *Definire un possibile circuito firmware della moltiplicazione [1]. Quali sono i criteri con cui si può ottimizzare un circuito firmware [3]?*
4. Definire gli input e gli output di una ALU. Disegnare il circuito della ALU che opera sul MSB [1+3].
5. *Qual'è la struttura di un sommatore a propagazione di riporto? Qual'è la struttura di un sommatore ad anticipazione di riporto? [3].*
6. *Data la seguente tabella della verità, sintetizzarne il circuito logico:*

<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>y</i>
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1
7. Definire il ciclo di esecuzione di un'istruzione. Quando l'architettura "capisce" di quale istruzione si tratta? [2].
8. Definire quali sono gli elementi più importanti di una CPU. [2]
9. Circuito HW della moltiplicazione. Qual è la parte che ha più peso sul tempo di calcolo? [2].
10. Costruire una ALU a 4 bit che consenta le seguenti operazioni: Somma, sottrazione, add, or, equal, less_than, e fornisca in output il codice 1111 quando viene selezionata l'operazione *equal* ed i due operandi risultano uguali; e 1000 quando viene selezionata l'operazione *less_than* ed il primo operando è minore del secondo (NB Non si prevede un segnale di *zero*).

11. In base a quali parametri si definisce la complessità di un circuito combinatorio e perché [2]?

12. Dimostrare che la porta NAND è una porta universale [2].

13. Dimostrare che $(x + !xy) = (x + y)$ [2].

14. Riportare le configurazioni speciali codificate dallo standard IEEE754 [2].

15. Disegnare il circuito HW che implementa la moltiplicazione binaria:

101 x

110 =

Calcolarne complessità e cammino critico.

16. Data la seguente tabella della verità, sintetizzarne il circuito logico come SOP:

a	b	c	d	y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	X
0	0	1	1	X
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	X
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	1
1	1	1	1	1

Trovare il circuito logico equivalente che minimizzi la complessità ed il cammino critico.

17. Cos'è un processore Cell? Descrivere i passi più significativi dello sviluppo delle architetture. Cosa si intende per elaborazione vettoriale? Cos'è l'accumulatore?

18. Cosa si intende per uscita indifferente di una funzione combinatoria?

19. Dato il circuito ottimizzato della moltiplicazione firmware, definire come occorre modificare l'unità di controllo e i dispositivi perchè possa eseguire anche la divisione.

20. Dato un divisore firmware ottimizzato su 4 bit, definire i passi per la seguente divisione: 1101 : 111, e il contenuto dei diversi registri ad ogni passo.

21. Cos'è un numero denormalizzato? Cosa sono i bit di guardia ed arrotondamento? Perchè vengono utilizzati?

21. Sintetizzare la macchina a stati finiti che gestisce un distributore automatico di bibite. La macchina

deve funzionare in modo che quando vengono inserite monete per esattamente 30 cents la macchina eroga una bottiglia di acqua. All'istante successiva la macchina si resetta qualunque sia l'input. Le monete che si possono inserire sono: 0, 10 centesimi. [7].

22. Disegnare lo schema generale di una macchina a stati finiti. Definire i passi per la progettazione e sintesi di una FSM [3].

23. Sintetizzare una macchina a stati finiti in grado di:

- Leggere un carattere alfabetico (A, B, C o D).
- Concatenarlo agli altri caratteri letti in sequenza.
- Riconoscere la sequenza di caratteri: AAA.

24. Definire che cos'è una ISA. [1].

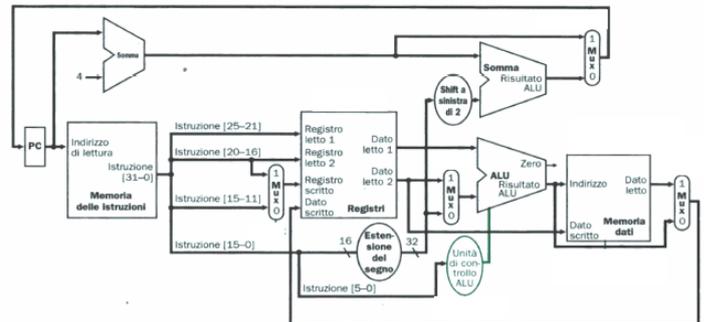
25. Definire i formati delle istruzioni e darne un esempio [4].

26. Definire i tipi di istruzioni messi a disposizione dall'ISA del MIPS R3000 [2].

27. Qual è lo spazio indirizzabile della memoria da un'istruzione di load/store?

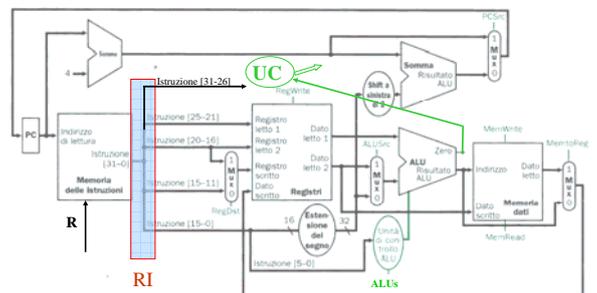
28. Descrivere il ciclo di esecuzione di un'istruzione. Di quali informazioni abbiamo bisogno in ciascuna fase?

29. Quali sono i segnali di controllo necessari per fare funzionare questa CPU? Quanti cicli di clock sono necessari per eseguire un'istruzione? Quanto valgono i segnali di controllo per eseguire l'istruzione `lw $s0, 40($s1)`? Quali sono le unità funzionali principali associate alle varie fasi di esecuzione? Indicare sul grafico la quantità che viene calcolata dalle varie unità funzionali.



30. Progettare un controllore a 2 livelli della ALU, sapendo che le operazioni consentite sono: add, and, ori, add, sub, and, or, lw, sw, beq. Il primo livello riceverà in input solamente il codice operativo, il secondo livello riceverà in input l'uscita del primo livello ed il contenuto del campo funct.

31. Seguire il data_path ed il control_path per i diversi tipi di istruzioni nella CPU a ciclo singolo riportata qui a fianco. Quali segnali di controllo **non** occorre specificare per un'istruzione di lw? E per un'istruzione di add? E per un'istruzione di addi? Per un'istruzione di beq?



32. Specificare quali sono gli operandi su cui agiscono le 3 ALU e da dove provengono. Cosa contiene l'ingresso 0 del Mux del secondo operando della ALU quando sto eseguendo l'istruzione: `add $t0, $t1, $t2`, sapendo che `$t0 = $7` e che `Funct = 0x20`. Cosa contiene l'uscita del Mux all'ingresso della porta di scrittura del register file nell'istruzione `beq $t0, $t1`? E nell'istruzione `sw $t0, 0($t1)`? E nell'istruzione: `add $t0, $t1, $t2`?

33. Come viene gestito un salto dalla CPU? Disegnare il circuito che gestisce i salti.
34. Costruire una CPU firmware, completa della sua unità di controllo, in grado di eseguire moltiplicazioni e divisioni su 4 bit. Calcolarne il cammino critico. I componenti di memoria saranno flip-flop o latch? Perché?
35. Costruire l'unità di controllo di una CPU firmware in grado di eseguire moltiplicazione e divisione mediante PLA e ROM.