

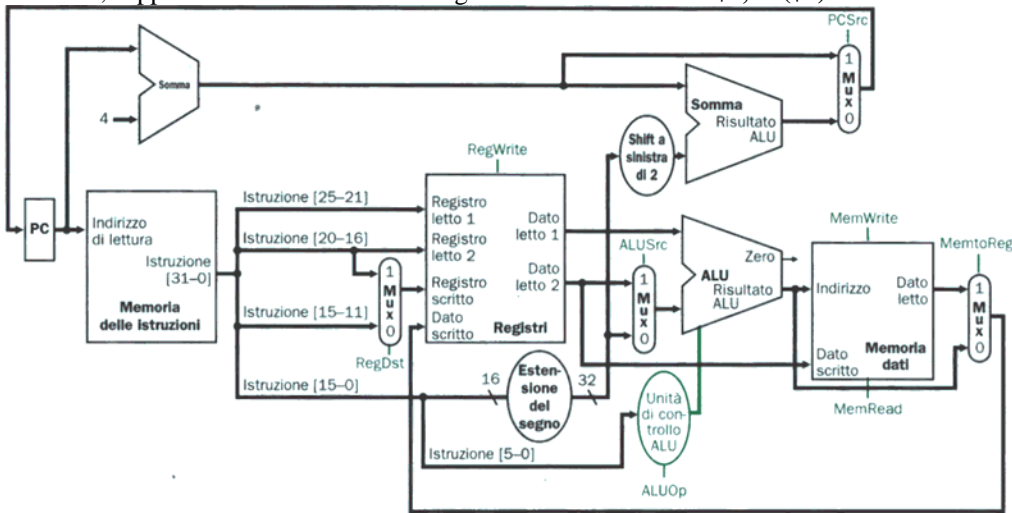
Cognome e nome dello studente:

Matricola:

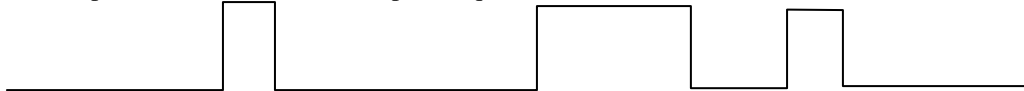
Anno di corso e turno:

A.A. 2004-2005 – Prova d’esame del 27 Gennaio 2005

- [3 p] Disegnare lo schema elettrico di un **full-adder** ad 1 bit e riportare la tabella di verità.
- [3 p] Convertire il numero decimale 99: a) in esadecimale; b) in numero binario secondo la codifica IEEE 754, singola precisione.
- [3 p] Mostrare che: $\overline{! [A(B + \overline{A})(C + \overline{B})]} = \overline{A} + \overline{B} + \overline{C}$ (senza ricorrere alla tabella di verità) e disegnare il circuito che implementa la funzione.
- [4 p] Evidenziare, nel seguente schema di CPU, i valori all’ingresso ed all’uscita di ogni ALU ed i valori di ogni segnale di controllo, supponendo che la CPU stia eseguendo l’istruzione: **lw \$5, 12(\$1)**



5. [10 p] Sintetizzare una macchina a stati finiti (di Moore) che implementi un contatore di impulsi modulo 3. Il contatore viene incrementato ad ogni fronte di salita dell’ingresso IN. L’ingresso IN viene regolarmente campionato (macchina sincrona). L’uscita è costituita da un segnale che va a “1” quando il contatore arriva a “2” [0...1...2]. Determinare il grafo (STG) e la tabella (STT) delle transizioni. Non si consideri il segnale di clock nel progetto. NB Un impulso è formato da un fronte di salita, un periodo a livello alto ed un fronte di discesa. Un possibile andamento di 3 impulsi è quindi:



6. [5 p] Il processore XYZ è progettato in modo da poter gestire fino a **16 MByte** di memoria principale con parole di **32 bit**. Tale processore viene dotato di una memoria cache a mappatura diretta di capacità **16 kByte** e con **linee di 8 parole**. Dimensionare e disegnare lo schema di tale memoria cache, evidenziando le dimensioni di tutti i campi. Mostrare inoltre come viene memorizzata nella cache la parola di memoria situata all’indirizzo (decimale) 2048, specificandone i campi **index** e **tag**.

7. [8 p] Tradurre il seguente codice C in Assembly MIPS e, quindi, in linguaggio macchina:

```

t1 = 24;
t4 = 220;
t0 = t1 + t4;
for (i=0; i<3; i++)
{
    t3 = t3 + (t0 * i);
}
    
```

sapendo che: \$t0 = \$7 e ricordando i seguenti codici operativi: beq = 4, add = 0, addi = 8, j = 2, lui = 15 ed i seguenti codici funzione: add = 32, sub = 34, slt = 42, mult = 24, mfhi = 16, mflo = 18, sll = 0, slr = 2. (Non si considerino eventuali overflow)