

CALCOLATORI ELETTRONICI A – 1 dicembre 2009

NOME:

COGNOME:

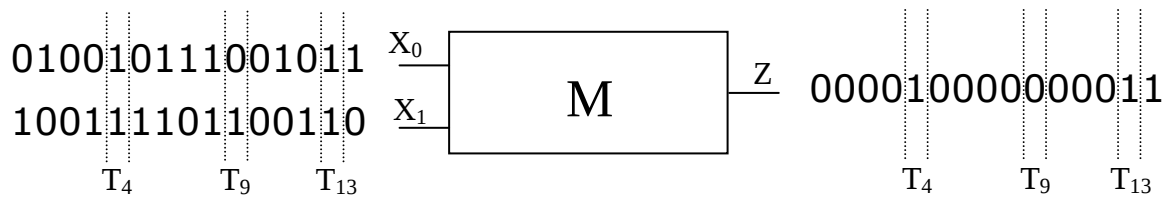
MATR:

Scrivere chiaramente in caratteri maiuscoli a stampa

1. Scrivere una procedura in assembler MIPS corrispondente alla seguente funzione espressa in linguaggio C, che riceve in ingresso un vettore di interi x ed un intero n e ritorna un valore intero. Si utilizzino le note convenzioni sui registri, cercando però di minimizzare l'uso dello stack. [6]

```
int P(int x[], int n){
    int i, somma;
    somma=x[0];
    for(i=1; i<n && x[i]>x[i-1]; i++)
        somma=somma+x[i];
    return somma;
}
```

2. Progettare la macchina sequenziale M indicata in figura, la cui uscita Z, normalmente a 0, diventa uguale a X_0 per due cicli di clock ogni volta che nell'ingresso X_1 si presentano due uni consecutivi, dopodiché si riporta allo stato iniziale. Si faccia riferimento all'esempio in figura, notando in particolare che l'uscita Z diventa uguale a X_0 già a partire dal ciclo in cui si presenta il secondo uno in X_1 . In particolare si richiede:
- il disegno del diagramma degli stati
 - uno schema di massima, comprendente registro di stato, funzione di stato futuro, funzione di uscita e relativi collegamenti
 - la specifica in formule logiche delle funzioni di stato futuro e di uscita.
- [8]



3. Si considerino, mostrati nelle figure alla pagina seguente, il datapath ed il diagramma a stati finiti che specifica l'unità di controllo secondo la tecnica a multiciclo relativamente alle istruzioni MIPS *lw*, *sw*, *beq*, *j* ed alle istruzioni *Tipo-R*.

Si vuole implementare la nuova istruzione

$\text{max } r1, (r2), (r3) \quad // \ r1 = \max\{M[r2], M[r3]\}$

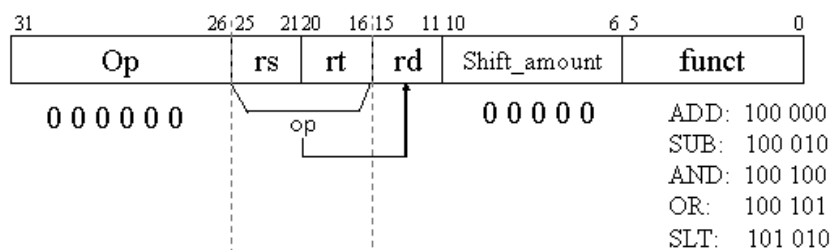
che confronta le due parole di memoria rispettivamente di indirizzo *r2* e *r3* ponendo il valore massimo nel registro *r1*.

Ricordando i tre formati di codifica delle istruzioni (riportati di seguito) si chiede di:

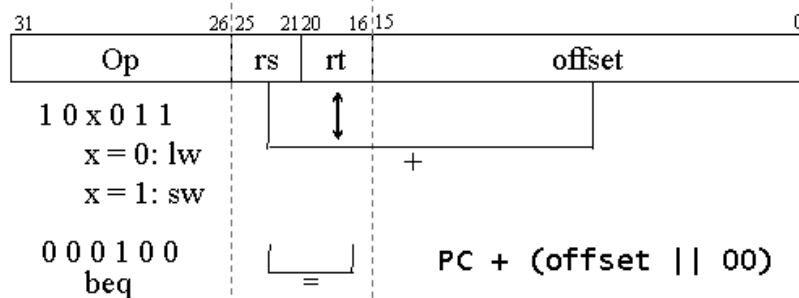
- riportare il formato della nuova istruzione macchina (specificando anche i campi destinati a *r1*, *r2* e *r3*);
- riportare, nella corrispondente figura, le modifiche necessarie al datapath;
- estendere il diagramma degli stati per implementare la nuova istruzione.

[6]

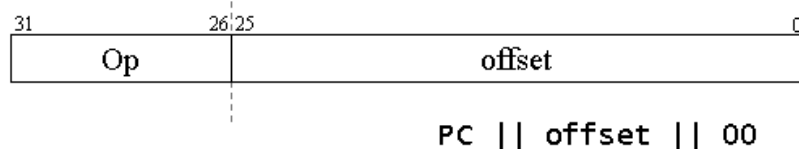
Promemoria formati delle istruzioni:



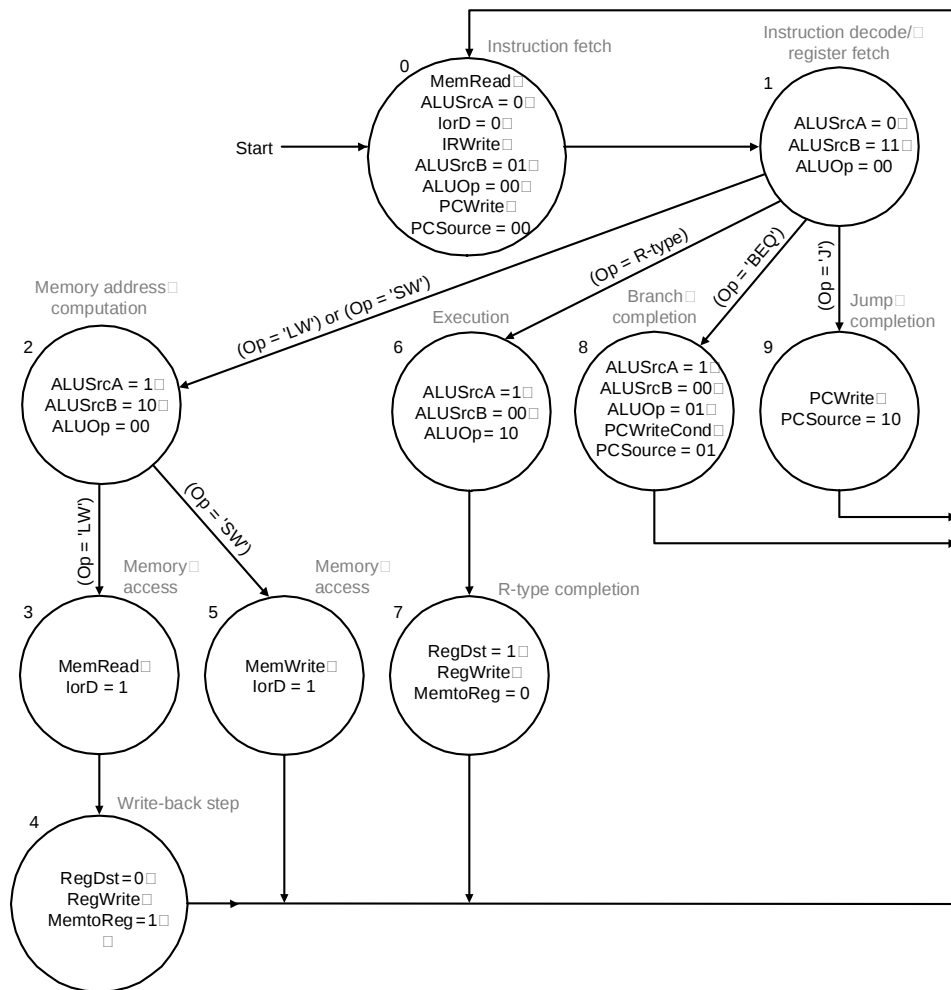
Aritmetiche:
Tipo-R

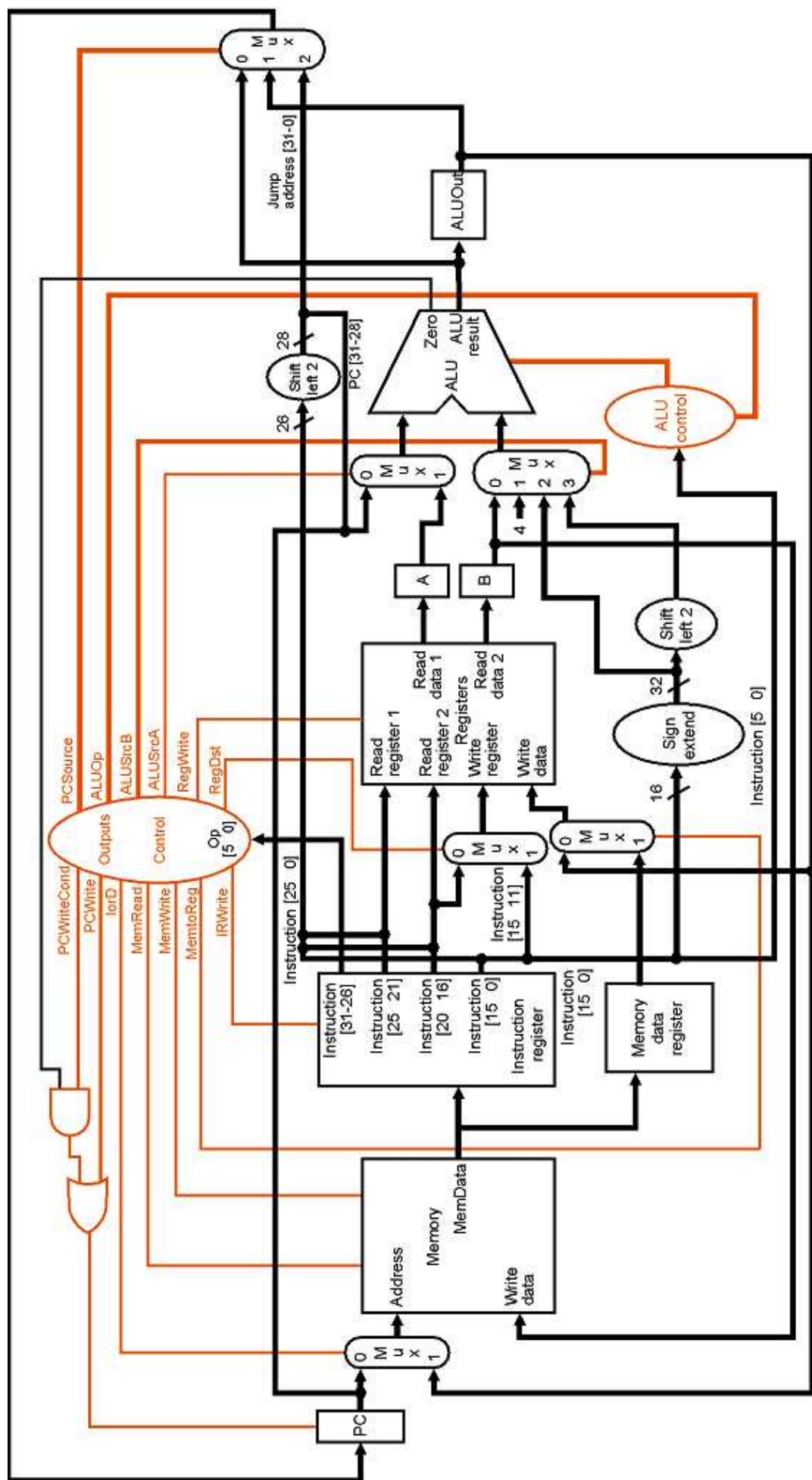


lw, sw, beq:
Tipo-I



J: Tipo-J





4. Si consideri la nota implementazione dell'unità di controllo secondo la tecnica multiciclo relativamente alle istruzioni MIPS lw, sw, beq, j e TIPO-R (si noti che sia il diagramma degli stati sia il datapath sono riportati nell'esercizio precedente). In qualità di esperti consulenti dell'implementazione del processore MIPS multi-ciclo, vi viene sottoposto il seguente problema. Il progetto attuale prevede un ciclo di clock pari a 2ns, che in particolare risulta sufficientemente lungo rispetto al tempo di accesso in lettura alla memoria, pari a 1.8 ns. Il problema che vi viene sottoposto è il fatto che, nella fase di fetch, l'unità di controllo del processore risulta molto veloce, tanto da attivare il segnale di scrittura del registro istruzioni *IRWrite* dopo appena 0.2 ns, ben prima che l'istruzione sia stata letta dalla memoria. Vi viene quindi chiesto quale accorgimento adottare per fare in modo che il segnale sia attivato dopo la lettura dalla memoria, in modo che il dato scritto nel registro istruzioni sia corretto. Cosa rispondereste? [3]
5. Illustrare le principali differenze tra RAM statica (SRAM) e RAM dinamica (DRAM), facendo anche riferimento al rispettivo utilizzo nel contesto dell'organizzazione del calcolatore. [2]

6. Si consideri una cache set-associativa a due vie in grado di memorizzare 16 KB (soltanto per la parte dati, escludendo le etichette), cui si accede con indirizzi a 32 bit. Sapendo che i blocchi della memoria cache sono 1K (1024), determinare la dimensione totale in bit della memoria. [3]
7. Si illustri sinteticamente la gestione delle eccezioni nel MIPS. Cosa si intende per interrupt vettorizzato? [4]

