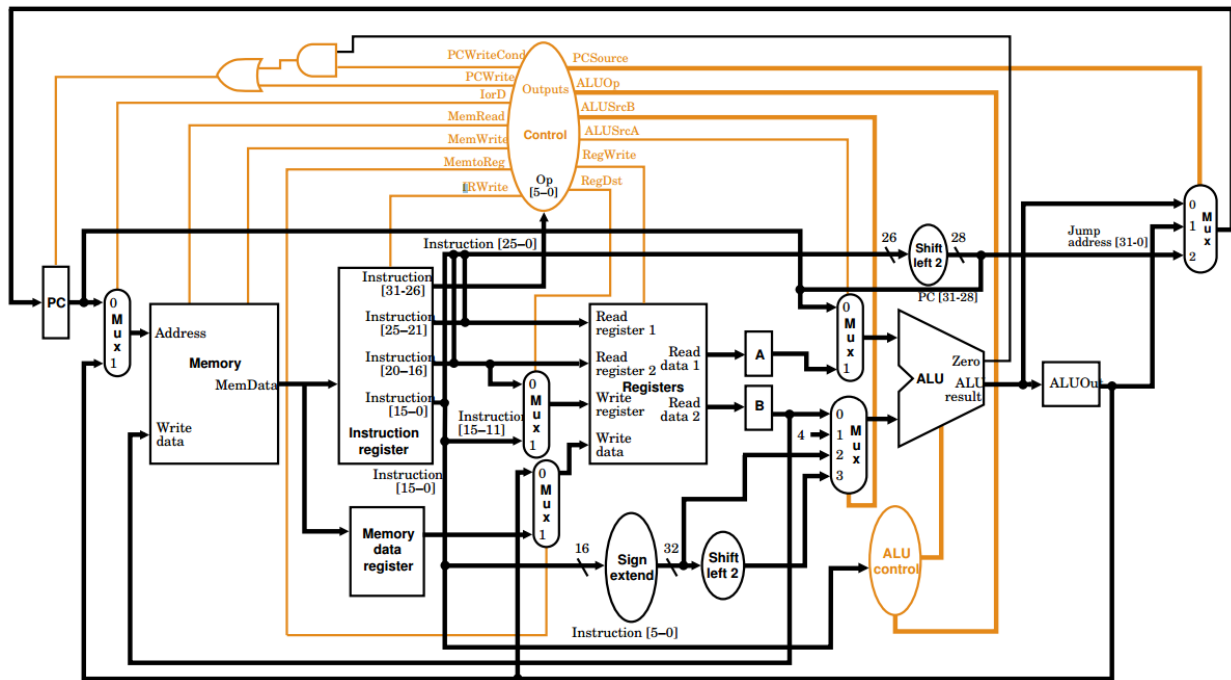




1. [5] Si consideri la CPU seguente, che esegue il codice a lato. Determinare il contenuto di tutti i registri della CPU (eccetto quelli nel Register File) dopo 11 cicli di clock compiuti dall'inizio della prima istruzione (indicandoli in formato esadecimale);

```
0xABC: add $6, $9, $0
      beq $9, $6, +4
      lw $5, 0($3)
      add $6, $6, $6
      sw $6, 0($10)
```



2. [6] Un processore caratterizzato da uno spazio di memoria indirizzabile di 256 MByte e un bus dati di 64 bit viene dotato di una memoria a mappatura diretta composta da 1024 linee di 16 parole. a) Calcolare esattamente quanti bit di memoria contiene la cache (considerando anche quelli per memorizzare il bit di validità e il campo *tag*); b) calcolare i valori (in esadecimale) di: *byte offset*, *word offset*, *index* e *tag* corrispondenti all'indirizzo di memoria: **0x1234567**.
3. [5] Rappresentare il contenuto, byte per byte in formato esadecimale, e gli indirizzi corrispondenti, della zona di memoria che viene modificata a seguito dell'esecuzione del frammento di codice MIPS32 a lato (si ricorda che il codice ASCII di "A" è 65). Determinare il valore che verrà associato dall'Assembler MIPS32 alla label **str**.
- ```
.data 0xAF8
 .half -512
arr: .word -25, 0xF5
 .space 9
str: .asciiz "CHE"
```
4. [3] Spiegare in cosa consiste un'architettura *superpipeline* e spiegare perché la sua lunghezza non può crescere oltre un certo limite.
5. [3] Elencare e descrivere i metodi che si utilizzano per minimizzare il tempo di gestione di una MISS in un sottosistema di memoria dotato di cache.
6. [4] Avendo a disposizione una capacità totale di cache (a mappatura diretta) di 256 kB, determinare la configurazione migliore (quante linee, quante parole per linea) per un'architettura a 8 bit, spiegando in dettaglio il motivo della scelta.
7. [4] Mostrare, mediante gli opportuni frammenti di codice MIPS, come allocare: a) staticamente e b) dinamicamente un array di 256k ( $2^{18}$ ) elementi.