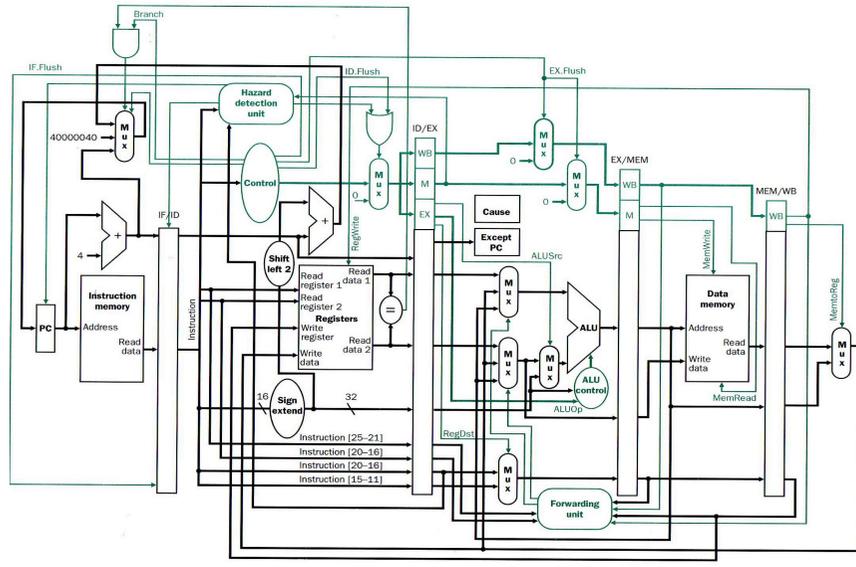




1. [6] Si consideri la CPU seguente (di cui si trascuri il circuito di anticipazione dei salti), mentre esegue il codice a lato.
- ```

0xCCC: lw $2, 0($1)
 lw $3, 0($2)
 beq $3, $3, -12
 sw $3, 80($1)
 add $3, $2, $1

```
- a) Identificare e descrivere gli eventuali casi di criticità;  
b) rappresentare il diagramma temporale dettagliato di esecuzione delle istruzioni per i primi 10 cicli di clock;  
c) determinare per quanti cicli di clock in totale l'istruzione `beq` rimane in pipeline.



2. [6] a) Disegnare lo schema circuitale di una cella di memoria dinamica e descrivere il meccanismo di lettura e quello di scrittura di un bit. b) Disegnare la struttura circuitale globale di una RAM dinamica di 16M x 1bit e descrivere come viene gestita la parola di indirizzo. Calcolare il periodo massimo di refresh, supponendo il tempo di scarica delle celle di memoria pari a 10,24 msec.

3. [4] Elencare e descrivere i criteri più comuni utilizzati nelle memorie cache a più vie per la scelta del banco in cui scrivere un nuovo blocco.

4. [4] Descrivere il principio di funzionamento delle *architetture superscalari*, sia con schedulazione statica che con schedulazione dinamica.

5. [6] Si vogliono incrementare le prestazioni di un calcolatore mediante l'introduzione di una memoria cache (di cui il calcolatore è originariamente sprovvisto). Il tempo complessivo di lettura/scrittura in cache è di 4 nsec, mentre la memoria principale impiega 60 nsec per la stessa operazione.

- a) Di quanto aumenta la velocità del calcolatore, in caso di miss-rate = 10%?  
b) Con che valore di miss-rate si ottiene un aumento di velocità del calcolatore di 10 volte?  
c) Mantenendo invece il miss-rate a 10%, quanto dovrebbe valere il tempo di lettura/scrittura della memoria cache per ottenere un incremento di velocità del calcolatore di 4 volte?

6. [6] Un programma Assembly MIPS ha bisogno di allocare staticamente un array di 80 interi e inizializzarlo con i numeri da 1 a 80, in ordine. Deve poi allocarne dinamicamente un altro, dopo aver chiesto all'utente (da terminale) il numero di elementi che deve contenere, che devono essere inizializzati a 1. Scrivere il codice che costruisce tali array.

#### System calls

|                           | codice (\$v0) | argomenti  | risultato |
|---------------------------|---------------|------------|-----------|
| <code>print_int</code>    | 1             | \$a0       |           |
| <code>print_float</code>  | 2             | \$f12      |           |
| <code>print_double</code> | 3             | \$f12      |           |
| <code>print_string</code> | 4             | \$a0       |           |
| <code>read_int</code>     | 5             |            | \$v0      |
| <code>read_float</code>   | 6             |            | \$f0      |
| <code>read_double</code>  | 7             |            | \$f0      |
| <code>read_string</code>  | 8             | \$a0, \$a1 |           |
| <code>sbrk</code>         | 9             | \$a0       | \$v0      |
| <code>exit</code>         | 10            |            |           |

#### Registri MIPS

|       |                    |       |                    |
|-------|--------------------|-------|--------------------|
| 0     | <code>zero</code>  | 24-25 | <code>t8-t9</code> |
| 1     | <code>at</code>    | 26-27 | <code>k0-k1</code> |
|       |                    | 28    | <code>Gp</code>    |
| 2-3   | <code>v0-v1</code> | 29    | <code>Sp</code>    |
| 4-7   | <code>a0-a3</code> | 30    | <code>s8</code>    |
| 8-15  | <code>t0-t7</code> | 31    | <code>Ra</code>    |
| 16-23 | <code>s0-s7</code> |       |                    |