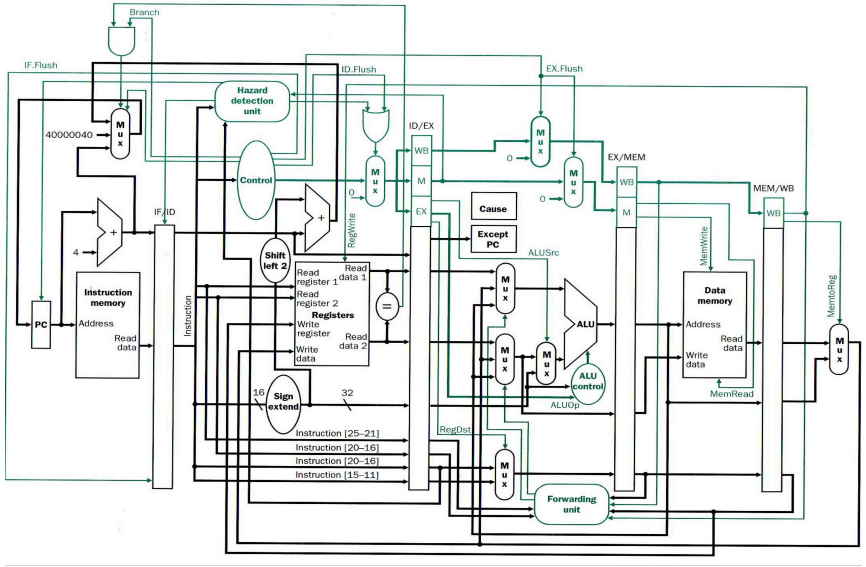
1. [6] Si consideri la CPU seguente (di cui si trascuri il circuito di anticipazione dei salti), mentre esegue il codice a lato.

```
0x8F8: lw $4, 0($1)
       lw $6, 0($4)
       bne $3, $3, -12
       lw $3, 80($1)
       add $3, $3, $9
```

a) Identificare gli eventuali casi di criticità e descrivere come vengono gestiti dalla CPU;

b) rappresentare il diagramma temporale dettagliato di esecuzione delle istruzioni del codice riportato, specificando quanti cicli di clock sono necessari per eseguire l'intero codice.



2. [5] Descrivere la struttura e il funzionamento dell'unità di propagazione ("forwarding unit") della CPU pipeline in figura, identificando tutti gli ingressi e le uscite dell'unità, definendo esattamente il suo algoritmo di funzionamento e determinandone la struttura circuitale.

3. [6] Descrivere il funzionamento della tecnica DMA per la gestione dell'I/O. In una CPU con frequenza di clock 1 GHz e bus dati di 32 bit, una periferica trasferisce informazioni al ritmo di 8.000.000 byte/s mediante DMA, trasferendo blocchi di 16.000 byte per volta. Supponendo che siano necessari 1500 cicli di clock per l'inizializzazione e altri 500 per la terminazione di ogni blocco DMA, calcolare a) quanti cicli di clock sono necessari per ogni blocco; b) la percentuale di occupazione della CPU durante il trasferimento.

4. [5] Spiegare come funziona la memoria virtuale, descrivendo come avviene, a livello circuitale, l'accesso alla memoria attraverso l'uso della Page Table, del Page Table Register e del Transition Look-aside Buffer.

5. [4] 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 64k x 1bit.
c) Calcolare il periodo massimo di refresh, supponendo il tempo di scarica delle celle di memoria pari a 40,96 msec.

6. [6] 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 inoltre: a) il valore che verrà associato dall'Assembler MIPS32 alla label **str**; b) il valore restituito al primo utilizzo della system call **sbrk**.

```
        .data 0xFEC
        .half -127,0x127
arr:    .word -40
        .space 5
        .align 3
str:    .asciiz "ABBA"

        .text
        ...
```

System calls

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

Registri MIPS

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