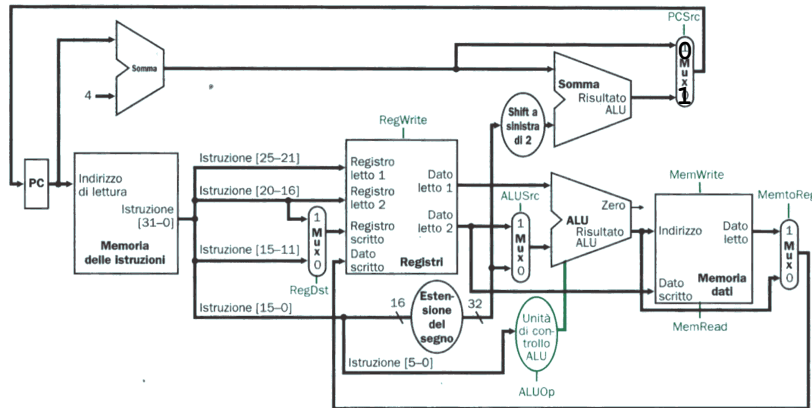


1. [10] Evidenziare, nel seguente schema di CPU:
- i valori all'ingresso ed all'uscita di ogni ALU,
  - i valori agli ingressi ed uscite del Register File (dove definito),
  - i valori di ogni segnale di controllo,
- supponendo che la CPU abbia eseguito le istruzioni seguenti e stia in questo momento eseguendo l'ultima istruzione:

```
0x200:  addi $t0, $zero, 5
        add $t1, $t0, $zero
        beq $t0, $t1, -24
```



2. [8] Disegnare lo schema circuitale di registro di 8 bit, e quindi si disegni il circuito di un Register File contenente 4 registri da 8 bit, con struttura simile al Register File MIPS, e cioè con singola porta di scrittura e doppia porta di lettura (si rappresentino i registri come moduli, senza disegnarne la struttura interna).
3. [8] Tradurre le seguenti istruzioni: a) in Assembly MIPS nativo e b) in linguaggio macchina, in formato esadecimale.

```
bgei $t1, +12, -12    # branch if gr. or eq. than immediate
divi $a0, $t0, 220+20 # divide by immediate
```

4. [8] Si traduca il seguente frammento di codice Assembly MIPS in linguaggio macchina, in formato esadecimale, calcolando prima i valori esadecimali Loc1 e Loc2 che permettono di saltare esattamente all'indirizzo indicato nel commento.

```
0x1A00:  bne $10, $11, Loc1    # salta a: 0x19B8
        j Loc2          # salta a: 0x01234560
```

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

		24-25	t8 - t9
0	zero		
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

MIPS Instruction Set:

