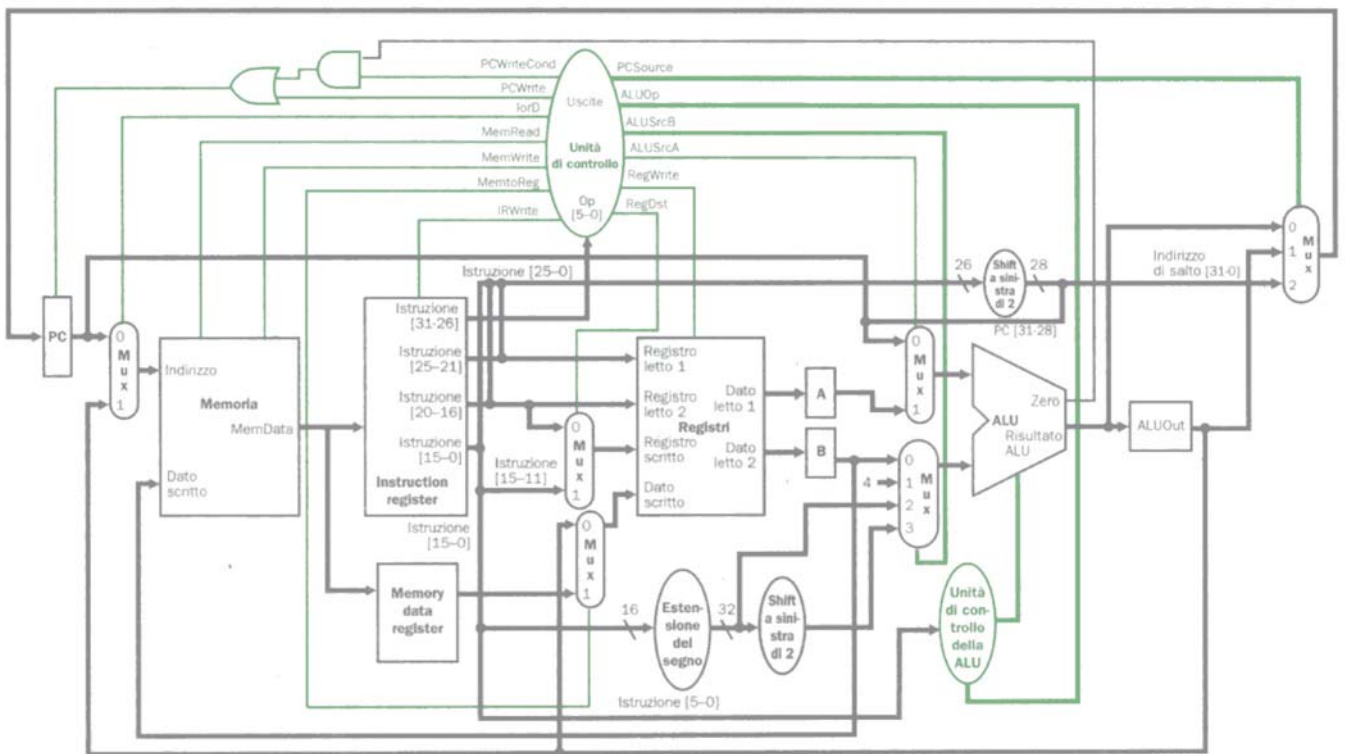


Data la CPU disegnata a fianco, disegnare la funzione logica che controlla i salti (condizionati ed incondizionati); cosa riceve in input, cosa riceve in output, quale funzione implementa? E' una funzione combinatoria o sequenziale? Perché? [3]



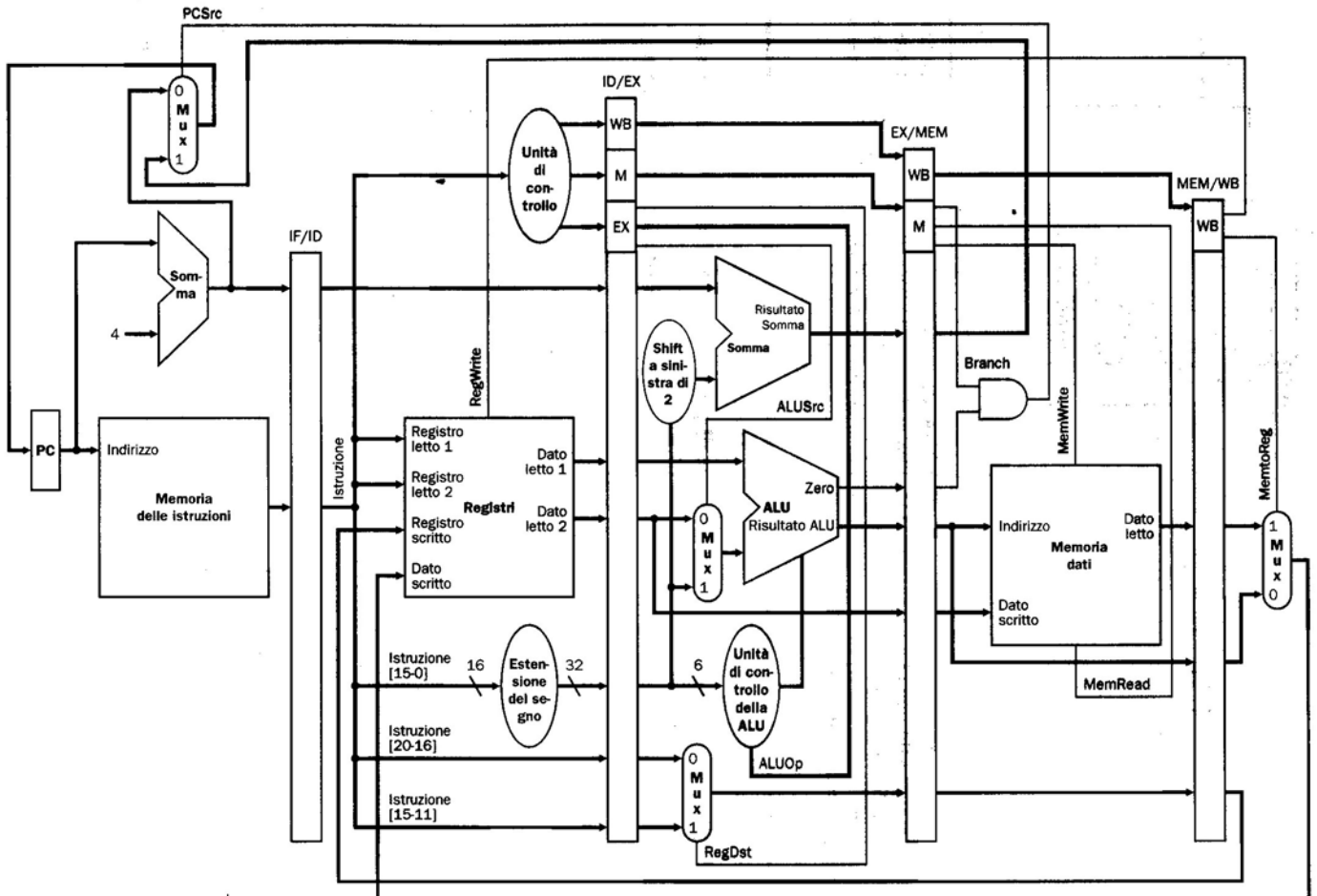
Da la CPU di cui sopra [12]:

- 1) Scrivere il contenuto di **tutte** le linee della CPU durante durante il ciclo di esecuzione dell'istruzione

`400: lw $t1, 20($t2).`

Si consideri che $\$t0 = \7 ed il registri $\$s0 = \16 . Il codice operativo della `lw` è 35, della `beq` è 4. I codici funzione della `add`, `or` e `and` sono rispettivamente: 32, 37 e 36. Dimensionare correttamente tutte le linee della CPU [7].

- 2) Modificare la CPU di cui sopra per potere gestire le eccezioni di overflow e di istruzione non valida. Cosa sono gli interrupt e le eccezioni? Come vengono gestiti dalle architetture e dal MIPS in particolare? Cosa contiene il registro di stato e dove si trova? [5]



Data la CPU disegnata sopra [20]:

- 1) Scrivere il contenuto di tutti i registri (parte master di PC + registri di pipeline) durante l'esecuzione di questo frammento di codice [6]:


```

400: lw $t1, 20($t2)
404: beq $t0, $t2, 20
408: add $s0, $s1, $s2
412: or $s3, $s4, $s5
416: and $s6, $s0, $s0
      
```
- 2) Ci sono hazard in questo frammento di codice (motivare la risposta)? Cos'è un hazard? Cos'è uno stallo della CPU? [2]
- 3) Modificare la CPU disegnata sopra in modo da potere gestire tutti gli hazard sui dati e minimizzare l'impatto degli hazard sul controllo [10].
- 4) Scrivere le funzioni logiche implementate dalle unità di controllo che gestiscono le criticità [2].
- 5) Dimensionare correttamente tutti i registri presenti nella CPU [1].