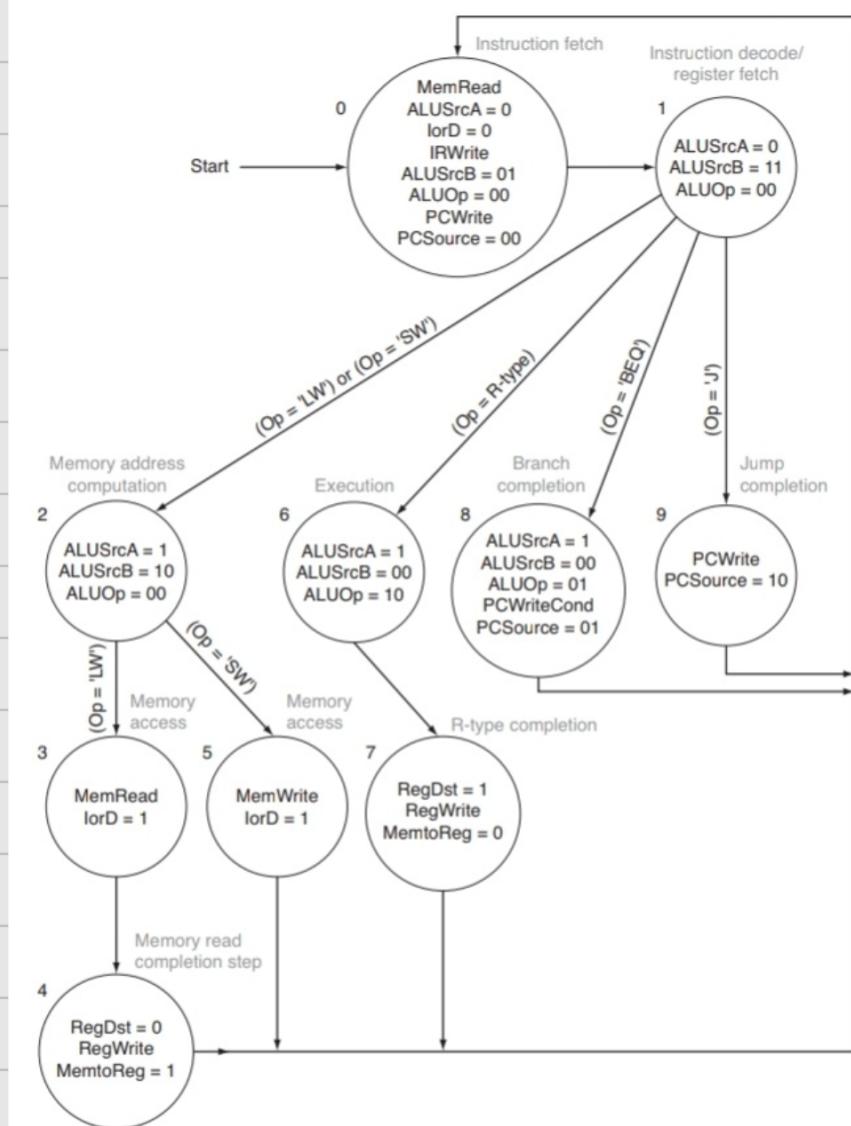


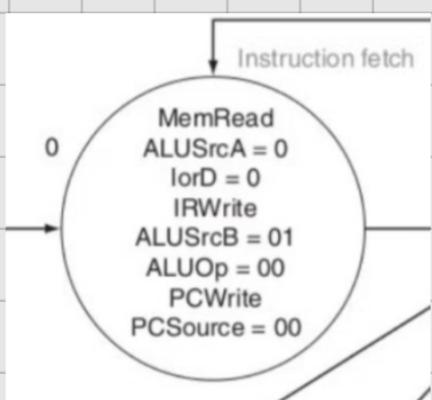
Automa a Stati Finiti: Analisi



Stato 0: Inst. Fetch

State: Instruction Fetch, uguale per tutte le operazioni.

Next State: Instruction Decode.



MemRead: la memoria deve far uscire un dato non acquisirlo.

AluSrcA: a
AluSrcB: 0s
AluOp: 00

} incremento PC + 4
 AluOp ↑
 ↑ SrcA SrcB

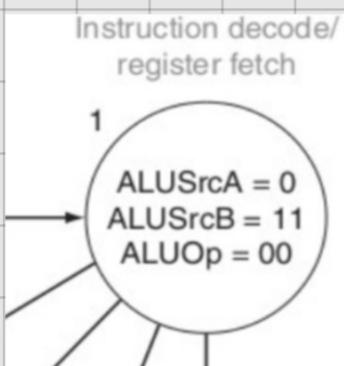
IR Write: Per scrivere verso l'Instruction Register
IorD=0, l'indirizzo da cui leggere è un istruzione (pc)
PC Write: Per salvare il nuovo valore del pc
PCSource = ∞ : pc preso dal risultato corrente di ALU

Stato 1: Inst. Decode

State: Instruction Decode, uguale per tutte le operazioni.

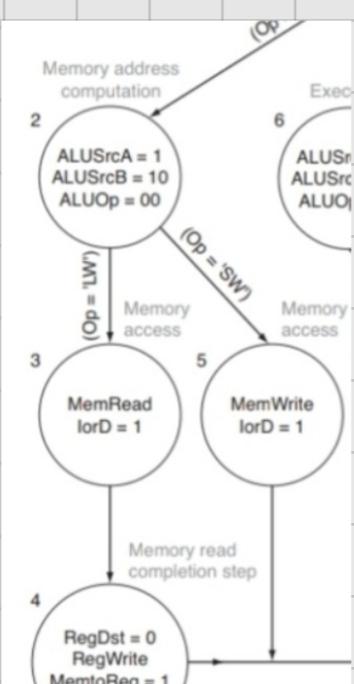
Next state: In base a opcodes, ha 4 next-state:

- op: lw/sw **stato 2**
- op: R-type **stato 6**
- op: beq **stato 8**
- op: j **stato 9**



$$\left. \begin{array}{l} \text{AluSrcA} = 0 \\ \text{AluSrcB} = 33 \\ \text{AluOp} = 0 \end{array} \right\} \begin{array}{l} \text{Si prepara ad un eventuale} \\ \text{branch} \end{array}$$

Stato 2, 3, 4, 5:



Stato 2: accesso a memoria

$$\left. \begin{array}{l} \text{AluSrcA} = 1 \\ \text{AluSrcB} = 10 \\ \text{AluOp} = 00 \end{array} \right\} \begin{array}{l} \text{Calcolo dell'offset tra reg A,} \\ \text{e valore imm.} \end{array}$$

Stato 3: lw

MemRead: la memoria deve far uscire un dato

non acquisirlo.

IorD=1, l'indirizzo da cui leggere è un dato (AluOut)

Stato 4: lw

Reg Write: Indica che scriviamo sul Reg. File

RegDst=0: Scriviamo il ris su rd

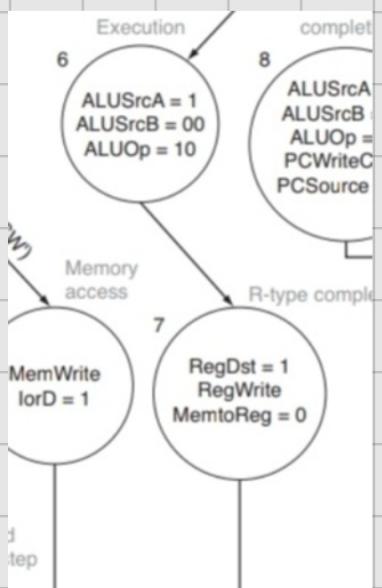
MemToReg=1: Il valore da scrivere proviene da nr

Stato 5: sw

MemWrite: la memoria deve acquisire un dato
non deve farlo uscire.

IorD=1, l'indirizzo da cui leggere è un dato (AluOut)

Stato 6-7:



Stato 6: R-type

AluSrcA=1
AluSrcB=00
AluOp=10 } Effettua l'operazione specificata
nei bit di Funct tra i due registri.

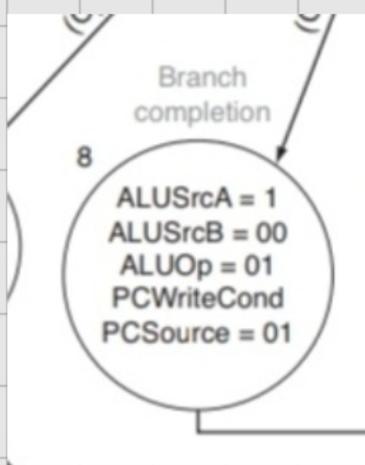
Stato 7, R-type

Reg Write: Per scrivere nel Reg. File

RegDst=1: Scriviamo il ris su rd

MemToReg=0: Il dato proviene da AluOut

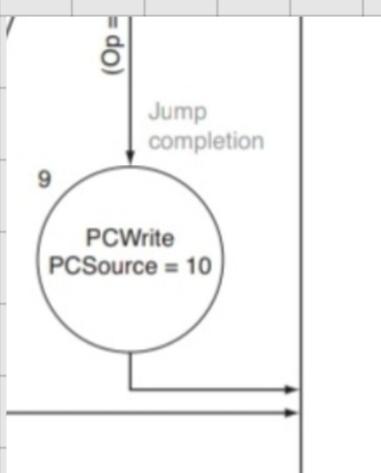
Stato 8: Beq



$AluSrcA = 1$ }
 $AluSrcB = 00$ }
 $AluOp = 01$ } Fa la sottrazione tra il reg A e il
 $reg B$ reg B
 $srcA$ srcB
 op

PcWriteCond: se la sott = 0 sono uguali (beq)
 PcSource = 01 il contenuto da scrivere sul pc è
 $AluOut$

Stato 9: jump



PcWrite: Abilita Scrittura su PC

PcSource = 10: l'indirizzo è la giost' apposizione dei primi 26 bit shiftati di 2 + i 4 del pc.



