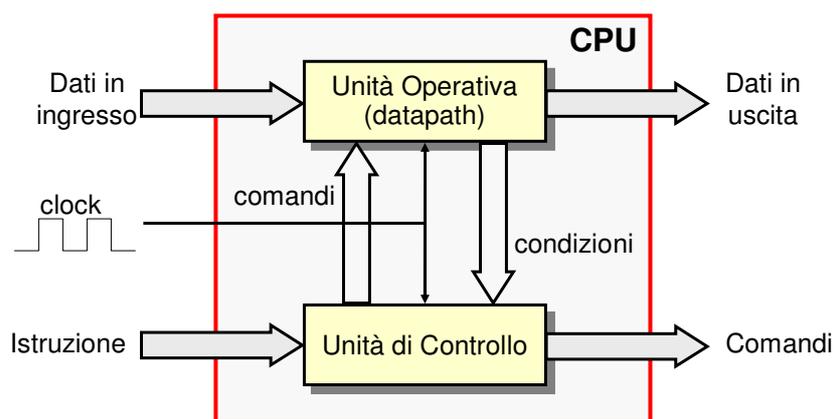

Un processore sequenziale

1

Struttura della CPU

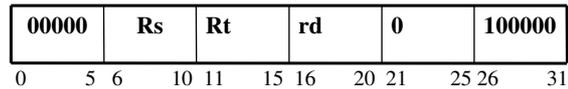


2

Istruzioni

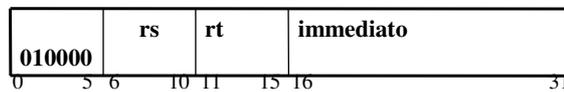
Istruzioni di tipo R

add rd, rs, rt $\text{Reg}[\text{rd}] = \text{Reg}[\text{rs}] + \text{Reg}[\text{rt}]$



Istruzioni con immediato

addi rt, rs, immediato $\text{Reg}[\text{rt}] = \text{Reg}[\text{rs}] + \text{est_segno}(\text{immediato})$



3

Istruzioni

Istruzioni di salto condizionato

beq rs,rt spiazzamento $\text{if}(\text{rs} == \text{rt}) \text{PC} = \text{PC} + \text{est_segno}(\text{spiazzamento} \ll 2)$



Istruzioni di salto

j destinazione $\text{PC} = \text{PC}[0:3] \& (\text{destinazione} \ll 2)$

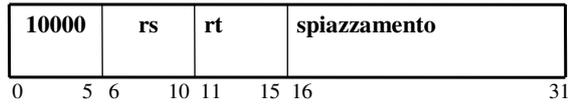


4

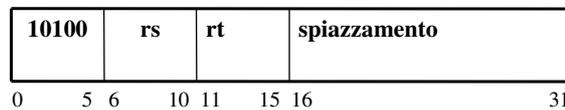
Istruzioni

Istruzioni di riferimento a memoria

lw rt, spiazamento (rs) $\text{Reg}[rt] = M[\text{Reg}[rs] + \text{est_segno}(\text{spiazamento})]$



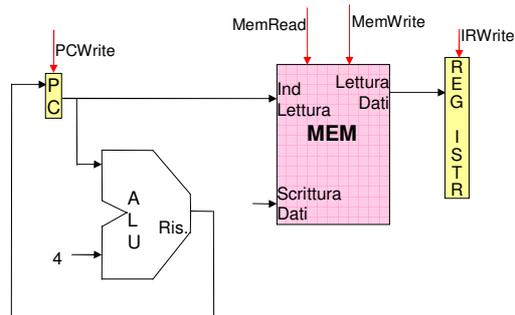
sw rt, spiazamento (rs) $M[\text{Reg}[rs] + \text{est_segno}(\text{spiazamento})] = \text{Reg}[rt]$



5

Componenti per realizzare il Fetch delle istruzioni

$\text{IR} := M[\text{PC}]; \text{PC} := \text{PC} + 4$

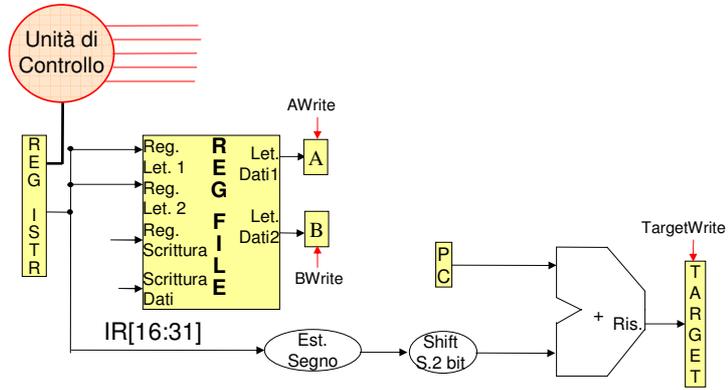


- Il contenuto del PC viene usato per indirizzare la memoria
- L'istruzione viene memorizzata nel registro IR registro istruzione

6

Componenti per realizzare il Decode delle istruzioni

$A := \text{Reg}[\text{IR}[6:10]]$; $B := \text{Reg}[\text{IR}[11:15]]$; $\text{Target} := \text{PC} + \text{est_segno}(\text{IR}[16:31]) \ll 2$

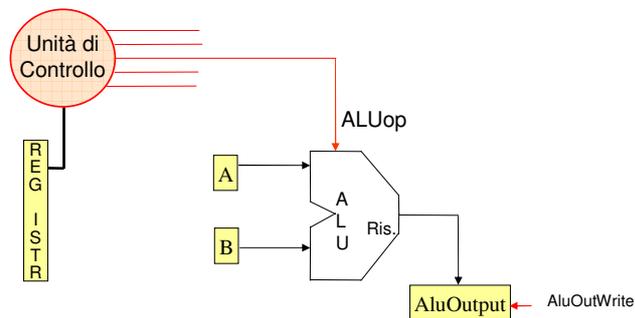


Il calcolo dell'indirizzo Target serve solo per le istruzioni branch

7

Componenti per realizzare l'Execute delle istruzioni R

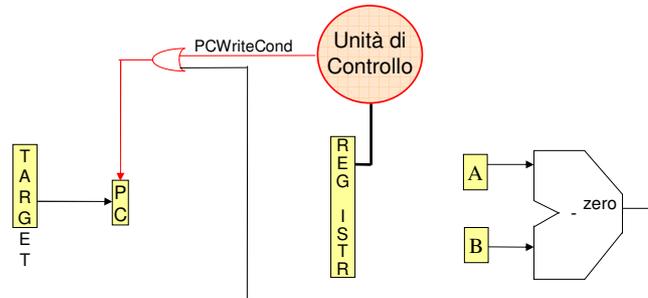
$\text{AluOutput} := A \text{ op } B$



8

Componenti per realizzare l'Execute delle istruzioni Branch

If (zero) PC:=Target

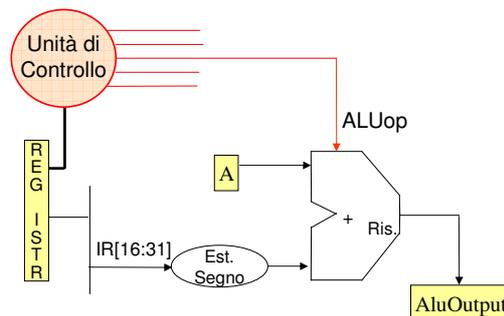


- La condizione zero è ottenuto realizzando la differenza A-B

9

Componenti per realizzare l'Execute delle istruzioni di accesso alla memoria

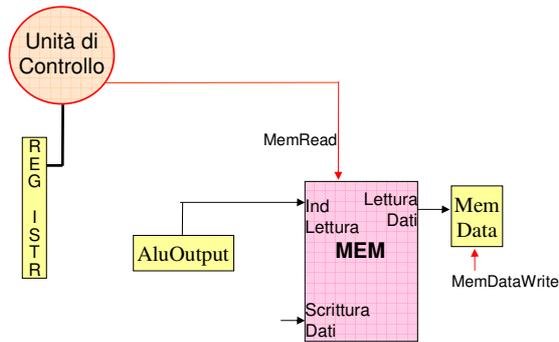
AluOutput:= A + est_segno(IR[16:31])



10

Componenti per realizzare la lettura delle istruzioni di accesso alla memoria

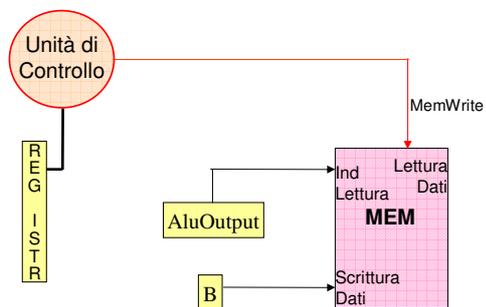
$$\text{Mem-data} = M[\text{AluOutput}]$$



11

Componenti per realizzare la scrittura delle istruzioni di accesso alla memoria

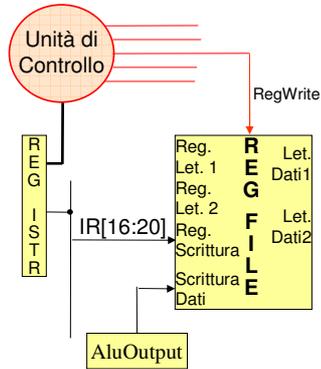
$$M[\text{AluOutput}] = B$$



12

Componenti per realizzare il Write back delle istruzioni R

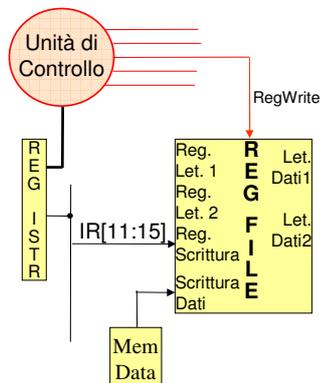
Reg[IR[16:20]]:= AluOutput



13

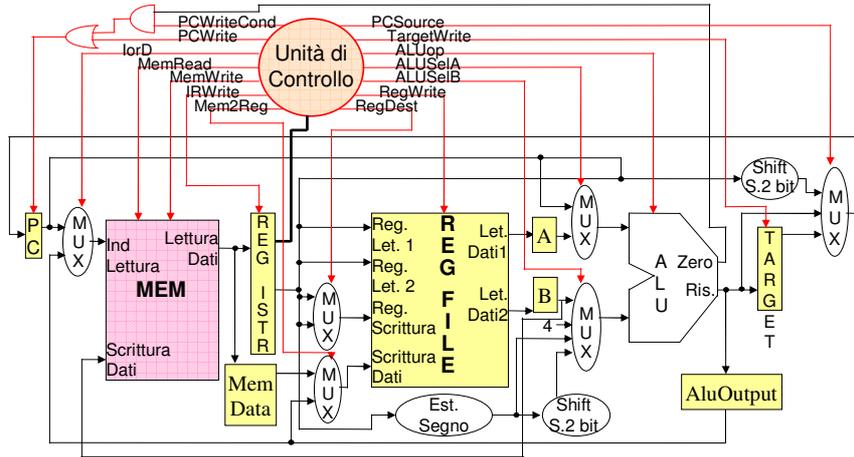
Componenti per realizzare il Write back delle istruzioni di accesso alla memoria

Reg[IR[11:15]]:= Mem-data



14

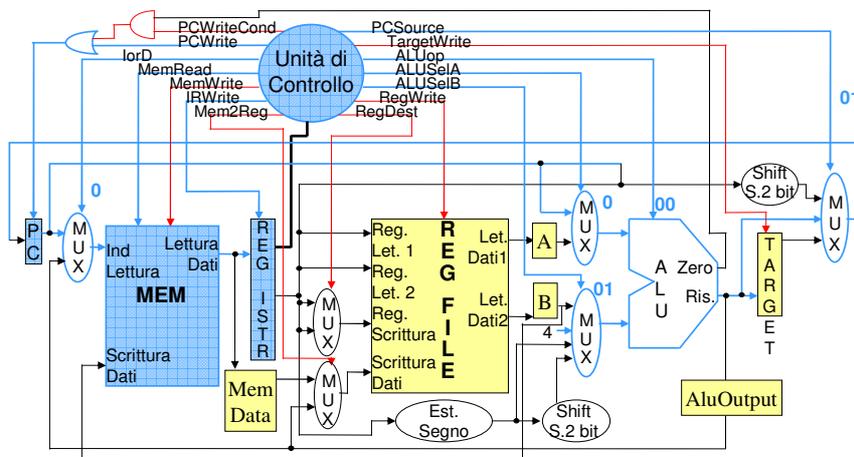
Unità Operativa+Segnali di Controllo



15

Istruzione R: Instuction Fetch

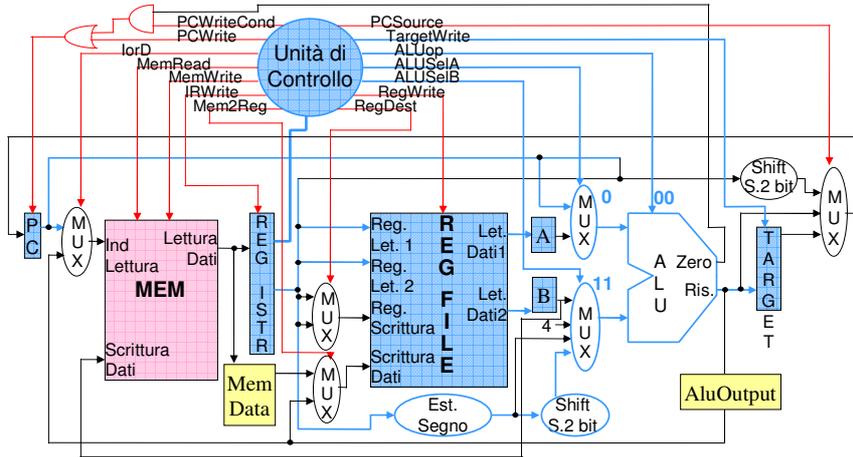
$IR:=M[PC]; PC=PC+4$



16

Istruzione R: Instruction Decode

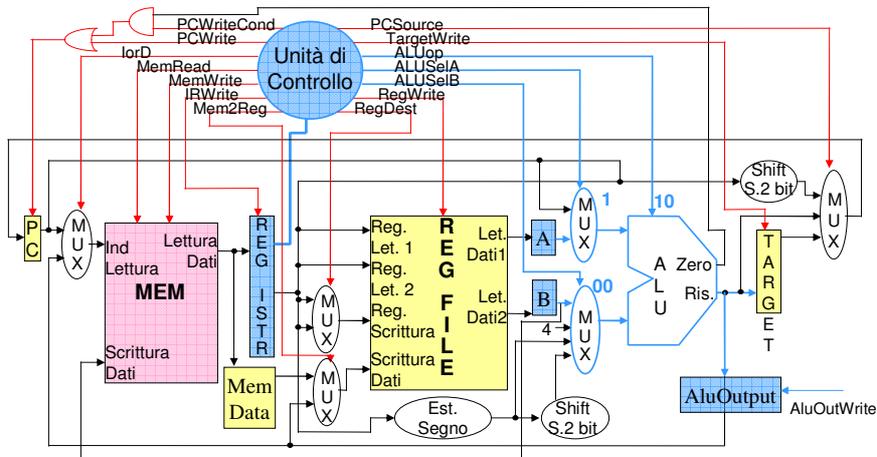
$A := \text{Reg}[\text{IR}[6:10]]$; $B := \text{Reg}[\text{IR}[11:15]]$; $\text{Target} := \text{PC} + \text{est_segno}(\text{IR}[16:31] \ll 2)$



17

Istruzione R: Execute

$\text{AluOutput} := A \text{ op } B$



18

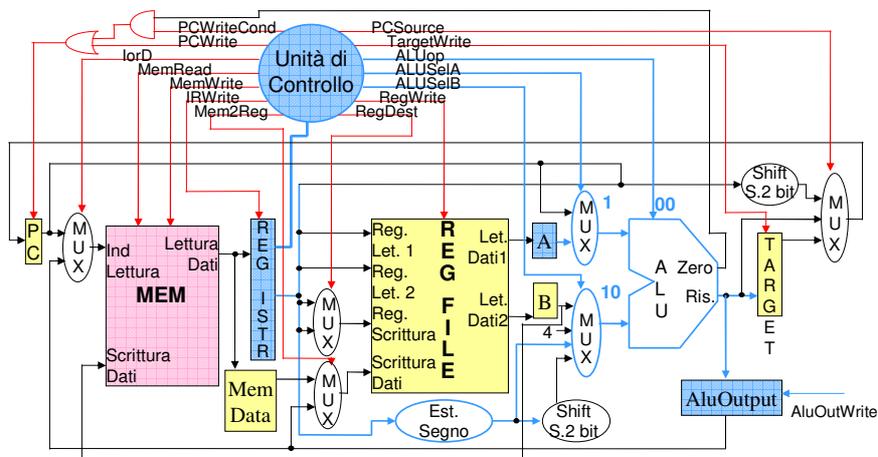
Istruzioni di lettura memoria

La fase di Instruction Fetch e Instruction Decode sono identiche a quelle degli altri tipi di istruzioni

21

Istruzioni di lettura memoria: Execute

$$\text{AluOutput} := A + \text{est_segno}(\text{IR}[16:31])$$



22

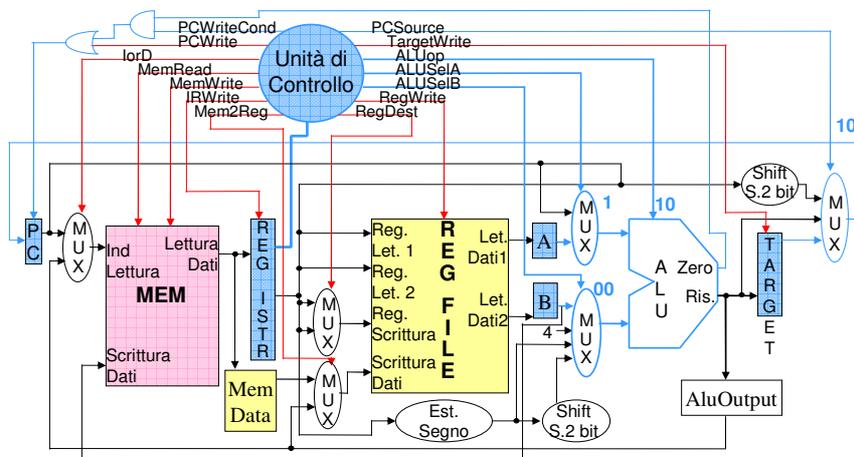
Esecuzione delle istruzioni di lettura memoria

Passo	Operazioni
Prelievo dell'istruzione	$IR := M[PC]; PC = PC + 4$
Decodifica dell'istruzione	$A := \text{Reg}[IR[6:10]]; B := \text{Reg}[IR[11:15]];$ $\text{Target} := PC + \text{est_segno}(IR[16:31] \ll 2)$
Esecuzione	$\text{AluOutput} := A + \text{est_segno}(IR[16:31])$
Accesso in lettura	$\text{Mem-data} = M[\text{AluOutput}]$
Scrittura	$\text{Reg}[IR[16:20]] := \text{Mem-data}$

25

Istruzione branch: Execute

If (zero) $PC := \text{Target}$



26

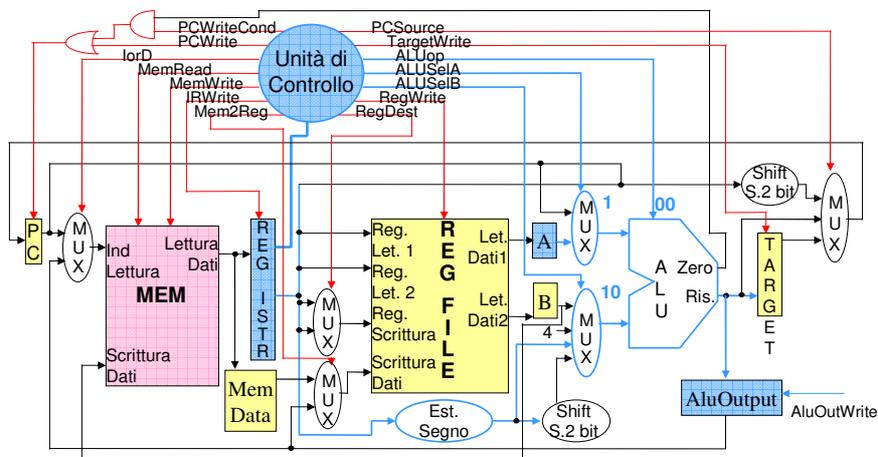
Esecuzione delle istruzioni Branch

Passo	Operazioni
Prelievo dell'istruzione <i>Instruction Fetch</i>	$IR := M[PC]; PC = PC + 4$
Decodifica dell'istruzione <i>Instruction Decode</i>	$A := \text{Reg}[IR[6:10]]; B := \text{Reg}[IR[11:15]];$ $\text{Target} := PC + \text{est_segno}(IR[16:31] < 2)$
Esecuzione <i>Execute</i>	If (zero) $PC := \text{Target}$

27

Istruzione di scrittura memoria: Execute

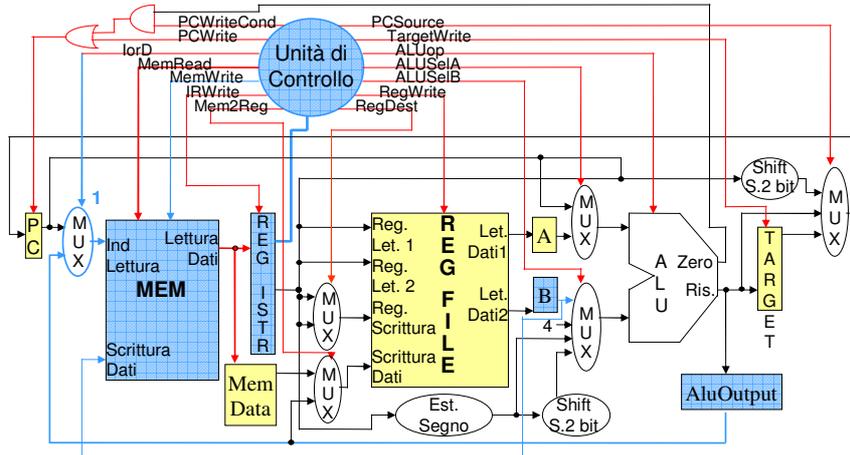
$$\text{AluOutput} := A + \text{est_segno}(IR[16:31])$$



28

Istruzione di scrittura memoria: Memory access

$$M[\text{AluOutput}] = B$$



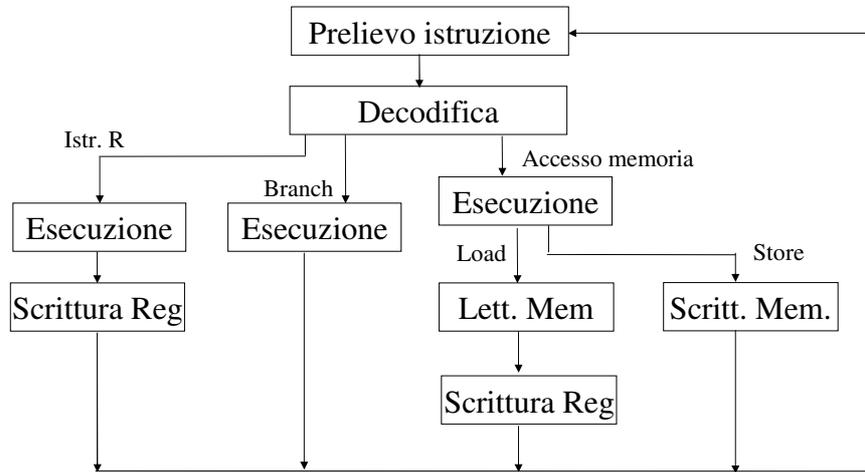
29

Esecuzione delle istruzioni di scrittura memoria

Passo	Operazioni
Prelievo dell'istruzione	$IR := M[PC]; PC = PC + 4$
Decodifica dell'istruzione	$A := \text{Reg}[IR[6:10]]; B := \text{Reg}[IR[11:15]];$ $\text{Target} := PC + \text{est_segno}(IR[16:31] \ll 2)$
Esecuzione	$\text{AluOutput} := A + \text{est_segno}(IR[16:31])$
Accesso in scrittura	$M[\text{AluOutput}] = B;$

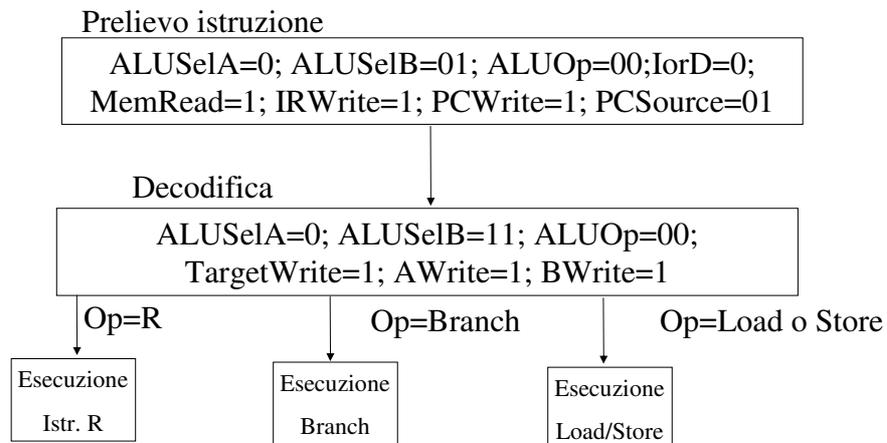
30

Macchina a stati del processore



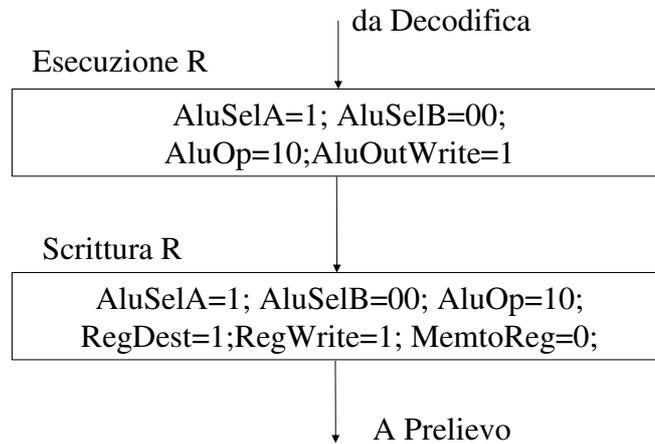
31

Segnali di controllo primi due stati



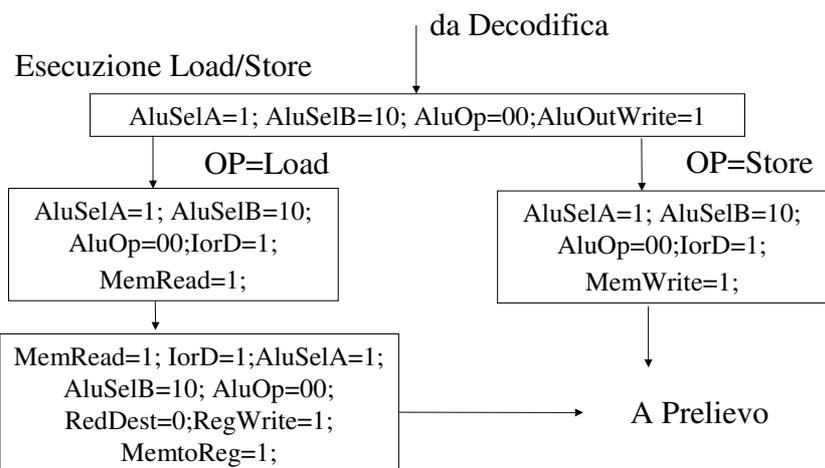
32

Segnali di controllo Esecuzione Istr. R



33

Segnali di controllo accesso memoria



34