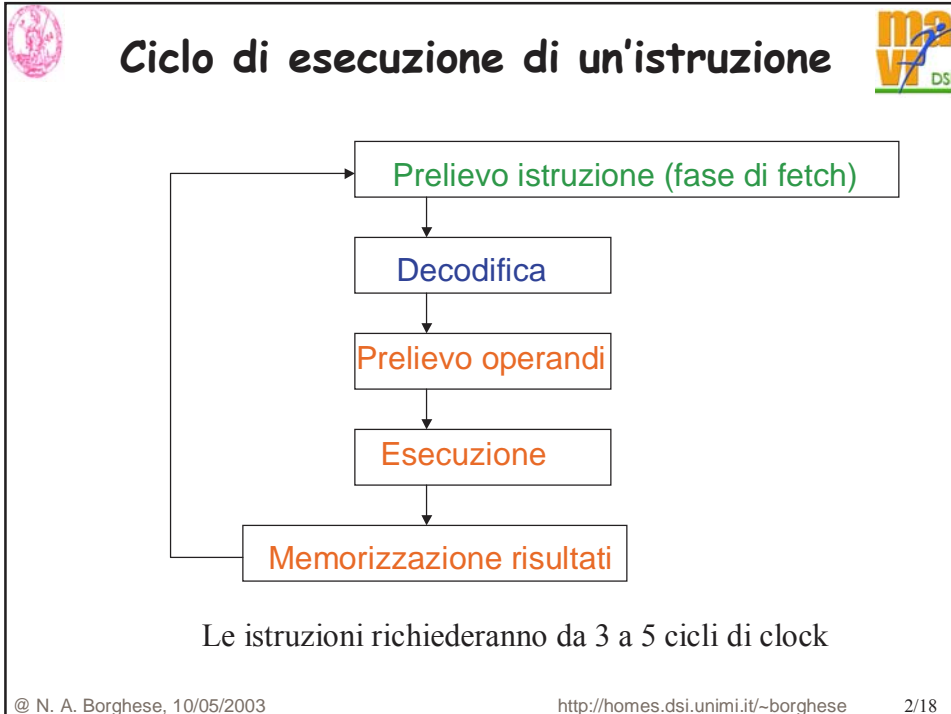


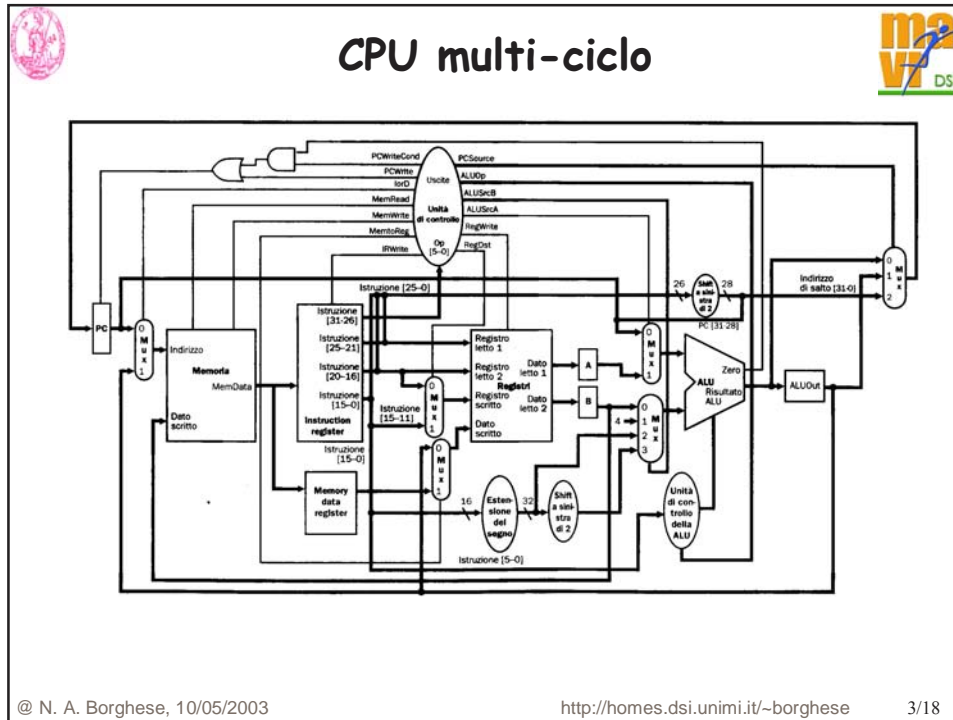
# Architettura della CPU multi-ciclo

## Architettura degli Elaboratori e delle Reti, Turno I



Alberto Borghese  
Università degli Studi di Milano  
Dipartimento di Scienze dell'Informazione  
email: borghese@dsi.unimi.it





## Riassunto dell'esecuzione

Nome del passo	Azioni per Istruzioni di Tipo R	Azioni per istruzioni di accesso alla memoria	Azioni per salti condizionati	Azioni per salti non condizionati
Fetch		IR = Memory[PC] PC = PC + 4		
decodifica & Prelievo dati dai registri		A = Reg[IR[25-21]] B = Reg[IR[20-16]] ALUOut = PC + (sign_extend(IR[15-0]) << 2)		
Esecuzione	ALUOut = A oper B	ALUOut = A + sign_extend(IR[15-0])	If (A == B) then PC = ALUOut	PC = PC[31-28]    (IR[25-0] << 2)
Conclusione	Reg[IR[15-11]] = ALUOut	Load: MDR = Memory[ALUOut] Store: Memory[ALUOut] = B		
Scrittura finale		Load: Reg[IR[20-16]] = MDR		

Le istruzioni richiedono da 3 a 5 cicli di clock

@ N. A. Borghese, 10/05/2003 http://homes.dsi.unimi.it/~borghese 4/18

Passo esecuzione	MemtoReg	RegDst	RegWrite	PCWriteCond	PCWrite	PCSource	ALUop	ALUSrcB	ALUSrcA	IRWrite	MemWrite	MemRead	MemtoReg
Fase fetch													
Decodifica													
Exec I - beq													
Exec I - j													
Exec I - R													
Exec II - R													
Exec I - sw													
Exec II - sw													
Exec I - lw													
Exec II - lw													
Exec III - lw													

Passo esecuzione	MemtoReg	RegDst	RegWrite	PCWriteCond	PCWrite	PCSource	ALUop	ALUSrcB	ALUSrcA	IRWrite	MemWrite	MemRead	MemtoReg
Fase fetch	X	X	X	0	0	00	10	01	0	1	0	1	0
Decodifica	X	X	X	0	0	X	10	11	0	0	0	0	X
Exec I - beq	X	X	0	1	0	01	01	00	1	0	0	0	X
Exec I - j	X	X	0	0	1	10	X	X	X	0	0	0	X
Exec I - R	X	X	0	0	0	X	10	00	1	0	0	0	X
Exec II - R	X	1	0	0	0	X	X	X	X	0	0	0	0
Exec I - sw	X	X	0	0	0	X	00	10	1	0	0	0	X
Exec II - sw	1	X	0	0	0	X	X	X	X	0	1	0	X
Exec I - lw	X	X	0	0	0	X	00	10	1	0	0	0	X
Exec II - lw	1	X	0	0	0	X	X	X	X	0	0	0	X
Exec III - lw	X	X	0	1	0	X	X	X	X	0	1	0	1



## Sintesi della macchina a stati finiti della CPU



I valori dei segnali di controllo dipendono:

- dal passo dell'istruzione.
- dal codice operativo (passi 3-5).
  
- Stato – passo di esecuzione.
- Uscita – segnali di controllo.
- Ingressi – OpCode.
  
- Uscita =  $f(\text{Stato})$
- Stato\_prossimo =  $f(\text{Ingressi}, \text{Stato\_presente})$



## Macchina a stati finiti

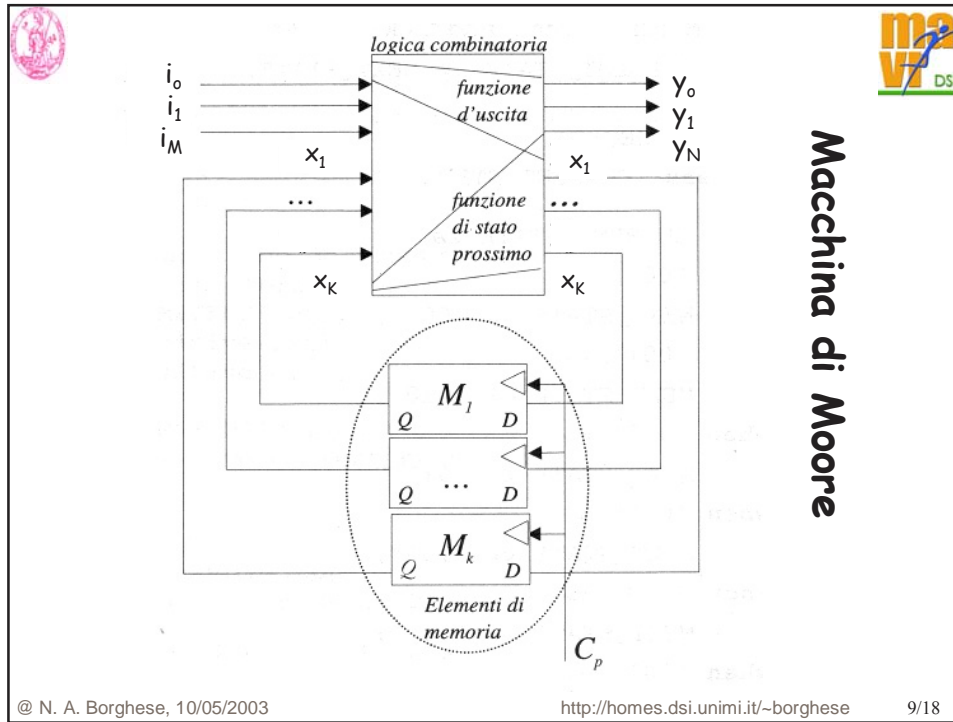


▪  $M : \langle S, I, O, f, g \rangle$

### *Come si costruisce?*

Occorre passare dalle specifiche alla STT o STG.

- Definizione degli stati.
- Definizione dello stato iniziale.
- Definizione delle transizioni tra stati (funzione stato prossimo:  $f(I, S) \rightarrow S$ )
- Definizione della funzione di uscita:  $g(S) \rightarrow Y$
- Semplificazione delle funzioni combinatorie: ( $f(\cdot)$  e  $g(\cdot)$ )

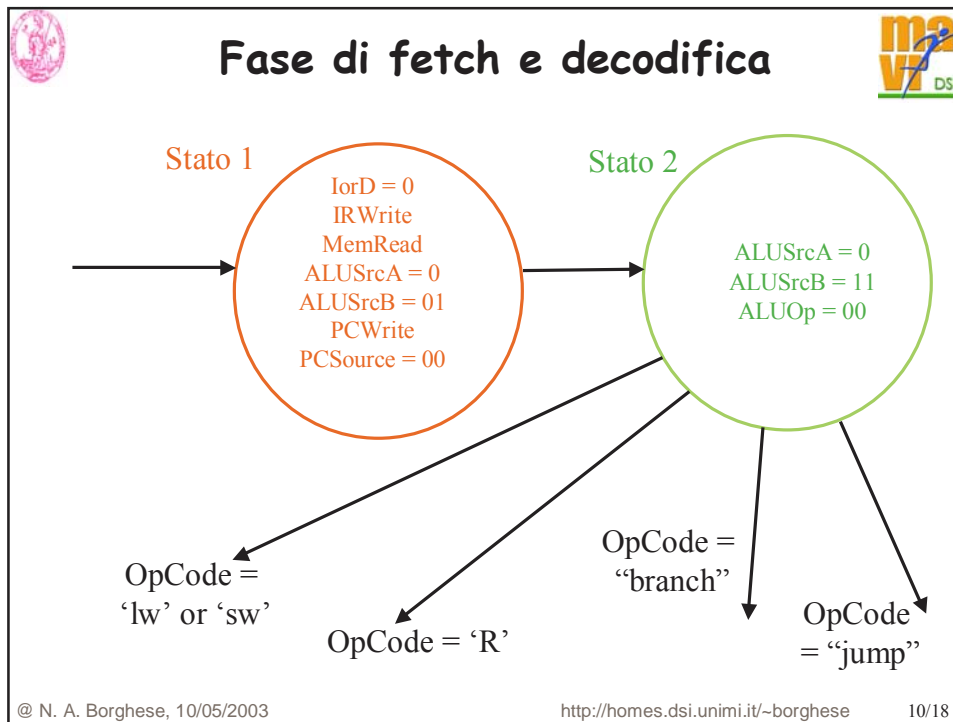


# Macchina di Moore

@ N. A. Borghese, 10/05/2003

<http://homes.dsi.unimi.it/~borghese>

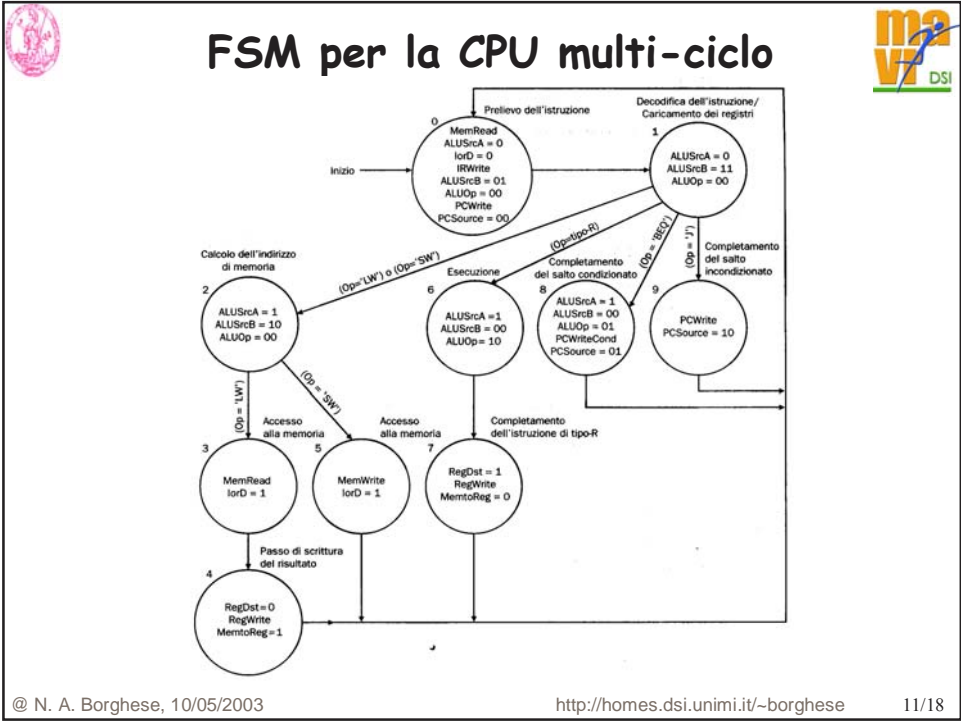
9/18



@ N. A. Borghese, 10/05/2003

<http://homes.dsi.unimi.it/~borghese>

10/18



## FSM - STT

Passo esecuzione	OpCode = R	OpCode = sw	OpCode = lw	OpCode = beq	OpCode = j	IorD	MemRead	MemWrite	IRWrite	ALUSrcA	ALUSrcB	ALUOp	PCSource	PCWrite	PCWriteCond	RegWrite	RegDst	MemtoReg
Fase fetch - 0						0	1	0	1	0	01	10	00	1	0	0	X	X
Decodifica - 1						X	0	0	0	0	11	10	X	0	0	0	X	X
Exec I - beq - 8				X		X	0	0	0	1	00	01	01	0	1	0	X	X
Exec I - j - 9					X	X	0	0	0	X	X	X	10	1	0	0	X	X
Exec I - R - 6						X	0	0	0	1	00	10	X	0	0	0	X	X
Exec II - R - 7						X	0	0	0	X	X	X	X	0	0	1	1	0
Exec I - sw - 2						X	0	0	0	1	10	00	X	0	0	0	X	X
Exec II - sw - 5						1	0	1	0	X	X	X	X	0	0	0	X	X
Exec I - lw - 2						X	0	0	0	1	10	00	X	0	0	0	X	X
Exec II - lw - 3						1	1	0	0	X	X	X	X	0	0	0	X	X
Exec III - lw - 4						X	0	1	0	X	X	X	X	0	0	1	0	1

### FSM - STT

Passo esecuzione	OpCode = R	OpCode = sw	OpCode = lw	OpCode = j	OpCode = beq	TorD	MemRead	MemWrite	IRWrite	ALUScrA	ALUSrcB	ALUop	PCSource	PCWrite	PCWriteCond	RegWrite	RegDst	MemtoReg
Fase fetch - 0	1	1	1	1	1	0	1	0	1	0	01	10	00	1	0	0	X	X
Decodifica - 1	6	2	2	8	9	X	0	0	0	0	11	10	X	0	0	0	X	X
Exec I - beq - 8	X	X	X	0	X	X	0	0	0	1	00	01	01	0	1	0	X	X
Exec I - j - 9	X	X	X	X	0	X	0	0	0	X	X	X	10	1	0	0	X	X
Exec I - R - 6	7	X	X	X	X	X	0	0	0	1	00	10	X	0	0	0	X	X
Exec II - R - 7	0	X	X	X	X	X	0	0	0	X	X	X	X	0	0	1	1	0
Exec I - sw - 2	X	5	5	X	X	X	0	0	0	1	10	00	X	0	0	0	X	X
Exec II - sw - 5	X	0	X	X	X	1	0	1	0	X	X	X	X	0	0	0	X	X
Exec I - lw - 2	X	X	3	X	X	X	0	0	0	1	10	00	X	0	0	0	X	X
Exec II - lw - 3	X	X	4	X	X	1	1	0	0	X	X	X	X	0	0	0	X	X
Exec III - lw - 4	X	X	0	X	X	X	0	1	0	X	X	X	X	0	0	1	0	1

### FSM - sintesi della funzione di uscita

Passo esecuzione	TorD	MemRead	MemWrite	IRWrite	ALUScrA	ALUSrcB	ALUop	PCSource	PCWrite	PCWriteCond	RegWrite	RegDst	MemtoReg
Fase fetch - 0	0	1	0	1	0	01	10	00	1	0	0	X	X
Decodifica - 1	X	0	0	0	0	11	10	X	0	0	0	X	X
Exec I - beq - 8	X	0	0	0	1	00	01	01	0	1	0	X	X
Exec I - j - 9	X	0	0	0	X	X	X	10	1	0	0	X	X
Exec I - R - 6	X	0	0	0	1	00	10	X	0	0	0	X	X
Exec II - R - 7	X	0	0	0	X	X	X	X	0	0	1	1	0
Exec I - sw/lw - 2	X	0	0	0	1	10	00	X	0	0	0	X	X
Exec II - sw - 5	1	0	1	0	X	X	X	X	0	0	0	X	X
Exec II - lw - 3	1	1	0	0	X	X	00	X	0	0	0	X	X
Exec III - lw - 4	X	0	1	0	X	X	X	X	0	0	1	0	1

Esempi:  
 $Y_2$  (RegWrite) = (Stato == 7) OR (Stato == 4)  
 $Y_{11}$  (ALUSrcA) = (Stato == 8) OR (Stato == 6)  
 OR (Stato == 2)

@ N. A. Borghese, 10/05/2003 http://homes.dsi.unimi.it/~borghese 14/18



## FSM - STT per lo stato futuro



Passo esecuzione	OpCode = R 000000	OpCode = sw 101011	OpCode = lw 100011	OpCode = beq 000100	OpCode = j 000010
Fase fetch - 0 - 0000	0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1
Decodifica - 1 - 0001	0 1 1 0	0 0 1 0	0 0 1 0	1 0 0 0	1 0 0 1
Exec I - lw/sw - 2 - 0010	X X X X	0 1 0 1	0 0 1 1	X X X X	X X X X
Exec II - lw - 3 - 0011	X X X X	X X X X	0 1 0 0	X X X X	X X X X
Exec III - lw - 4 - 0100	X X X X	X X X X	0 0 0 0	X X X X	X X X X
Exec II - sw - 5 - 0101	X X X X	0 0 0 0	X X X X	X X X X	X X X X
Exec I - R - 6 - 0110	0 1 1 1	X X X X	X X X X	X X X X	X X X X
Exec II - R - 7 - 0111	0 0 0 0	X X X X	X X X X	X X X X	X X X X
Exec I - beq - 8 - 1000	X X X X	X X X X	X X X X	0 0 0 0	X X X X
Exec I - j - 9 - 1001	X X X X	X X X X	X X X X	X X X X	0 0 0 0

@ N. A. Borghese, 10/05/2003 <http://homes.dsi.unimi.it/~borghese> 15/18



## FSM - funzione stato futuro (sintesi circuitale)



Esempio:  $S_0(t+1) = (\bar{S}_0 \bar{S}_1 \bar{S}_2 \bar{S}_3) + (S_0 \bar{S}_1 \bar{S}_2 \bar{S}_3)(i_0 i_1 \bar{i}_2 \bar{i}_4 i_5) + (\bar{S}_0 S_1 S_2 \bar{S}_3)$

Passo esecuzione	OpCode = R 000000	OpCode = sw 101011	OpCode = lw 100011	OpCode = beq 000100	OpCode = j 000010
Fase fetch - 0 - 0000	0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1
Decodifica - 1 - 0001	0 1 1 0	0 0 1 0	0 0 1 0	1 0 0 0	1 0 0 1
Exec I - lw/sw - 2 - 0010	X X X X	0 1 0 1	0 0 1 1	X X X X	X X X X
Exec II - lw - 3 - 0011	X X X X	X X X X	0 1 0 0	X X X X	X X X X
Exec III - lw - 4 - 0100	X X X X	X X X X	0 0 0 0	X X X X	X X X X
Exec II - sw - 5 - 0101	X X X X	0 0 0 0	X X X X	X X X X	X X X X
Exec I - R - 6 - 0110	0 1 1 1	X X X X	X X X X	X X X X	X X X X
Exec II - R - 7 - 0111	0 0 0 0	X X X X	X X X X	X X X X	X X X X
Exec I - beq - 8 - 1000	X X X X	X X X X	X X X X	0 0 0 0	X X X X
Exec I - j - 9 - 1001	X X X X	X X X X	X X X X	X X X X	0 0 0 0

@ N. A. Borghese, 10/05/2003 <http://homes.dsi.unimi.it/~borghese> 16/18





## Valutazione della prestazione della CPU Multi-ciclo



	lw	sw	beq	j	R	Singolo ciclo	Multi-ciclo
Durata	10ns	8ns	6ns	6ns	8ns		
Tipi	24%	12%	18%	2%	44%	10ns	8.08 ns

$$T_{\text{medio}} = (0.24 * 10 + 0.12 * 8 + 0.18 * 6 + 0.02 * 6 + 0.44 * 8) \text{ ns} = 8.08 \text{ ns}$$



## Riassunto: CPU multi-ciclo



L'unità di controllo di una CPU multi-ciclo è rappresentabile da una FSM.

Lo stato è codificato su 4 bit.

L'ingresso è rappresentato dai 6 bit del Codice Operativo.

L'uscita è rappresentata dai segnali di controllo (14 bit).

$Y = f(S)$  è una funzione a 6 ingressi e 10 uscite.

$S_{t+1} = f(S, I)$  è una funzione a 10 ingressi e 14 uscite.

Il tempo di esecuzione è dato dalla somma dei tempi dei passi elementari delle varie istruzioni.