



L'unità di controllo di CPU multi-ciclo

Prof. Alberto Borghese
Dipartimento di Scienze dell'Informazione
borgnese@dsi.unimi.it

Università degli Studi di Milano

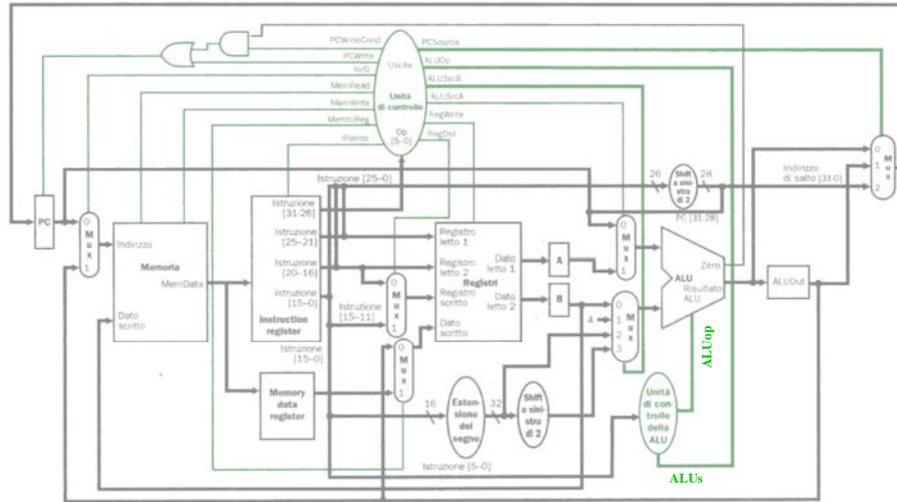


Sommario

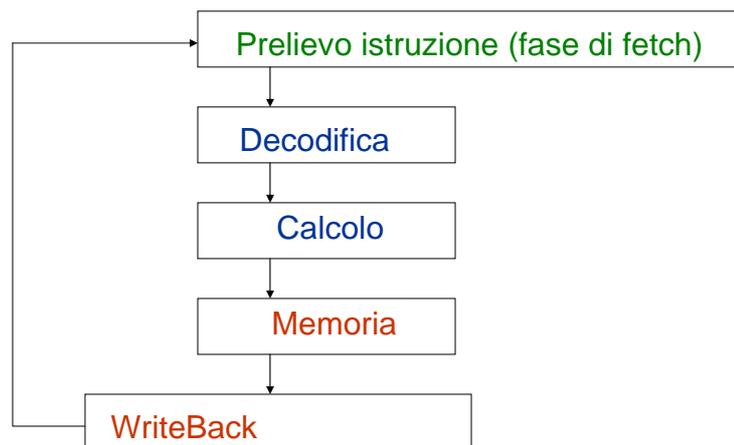
I segnali di controllo della CPU multi-ciclo

Sintesi dell'Unità di Controllo come Macchina a Stati Finiti

CPU multi-ciclo



Ciclo di esecuzione di un'istruzione



Le istruzioni richiederanno da 3 a 5 cicli di clock



Principio della suddivisione in passi



- Tutte le operazioni elementari che hanno bisogno di unità funzionali diverse possono essere eseguite in parallelo.
- Tutte le operazioni elementari che hanno bisogno della stessa unità funzionale devono essere eseguite in serie (in passi di esecuzione successivi).



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+4 + (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)
Memoria o WriteBack		Load: MDR = Memory[ALUOut] Store: Memory[ALUOut] = B		
WriteBack	Reg[IR[15-11]] = ALUOut	Load: Reg[IR[20-16]] = MDR		

Le istruzioni richiedono da 3 a 5 cicli di clock

Segnali di controllo

Nome del segnale	Valore	Effetto
ALUSrcA (1 bit)	0	Il primo operando è il valore attuale del PC
	1	Il primo operando proviene dalla prima porta di lettura del Register File
ALUSrcB (2 bit)	00	Il secondo operando proviene dalla seconda porta di lettura del RF
	01	Il secondo operando è la costante + 4
	10	Il secondo operando è l'estensione del segno del campo offset
	11	Il secondo operando proviene dall'estensione del segno e dallo shift a sx di due posizioni, dell'offset
IorD	0	L'indirizzo della memoria proviene dal PC
	1	L'indirizzo della memoria proviene dalla ALU (ALUOut)
PCSource	00	In PC viene scritta l'uscita della ALU (PC+4)
	01	In PC viene scritta il contenuto di ALUOut (indirizzo di una branch)
	10	In PC viene scritto l'indirizzo di destinazione della jump
PCWrite	0	Non abilitazione della scrittura
	1	Viene scritto il registro PC. L'indirizzo scritto in PC è controllato da PCSource
PCWriteCond	0	Non abilitazione della scrittura.
	1	Il PC viene scritto se anche l'uscita Zero della ALU è affermata

A.A. 2004-2005

7/21

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

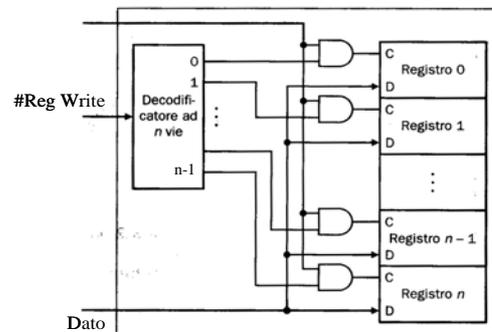
Altri segnali di controllo

Scrittura del Register File – solamente nelle istruzioni con fase di WriteBack.

Scrittura / lettura Memoria – solamente nelle istruzioni di accesso alla memoria.

Scrittura nei registri A, B,
ALUOut – ad ogni colpo di clock.

Scrittura del IR – Avviene durante
la fase di fetch.



A.A. 2004-2005

8/21

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



Sommario



I segnali di controllo della CPU multi-ciclo

Sintesi dell'Unità di Controllo come Macchina a Stati Finiti



Segnali di controllo



Passo esecuzione	MemtoReg	RegDst	RegWrite	PCWriteCond	PCWrite	PCSource	ALUop	ALUSrcB	ALUSrcA	IRWrite	MemWrite	MemRead	IOpD
Fase fetch													
Decodifica													
Calcolo – lw													
Memoria – lw													
WB – lw													
Calcolo – sw													
Memoria – sw													
Calcolo – R													
WB – R													
Calcolo – beq													
Calcolo – j													



Segnali di controllo



Passo esecuzione	IoRD	MemRead	MemWrite	IRWrite	ALUScrA	ALUScrB	ALUop	PCSource	PCWrite	PCWriteCond	RegWrite	RegDst	MemtoReg
Fase fetch	0	1	0	1	0	01	10	00	1	0	0	X	X
Decodifica	X	0	0	0	0	11	10	X	0	0	0	X	X
Calcolo – lw	X	0	0	0	1	10	00	X	0	0	0	X	X
Memoria – lw	1	1	0	0	X	X	X	X	0	0	0	X	X
WB – lw	X	0	1	0	X	X	X	X	0	0	1	0	1
Calcolo – sw	X	0	0	0	1	10	00	X	0	0	0	X	X
Memoria – sw	1	0	1	0	X	X	X	X	0	0	0	X	X
Calcolo – R	X	0	0	0	1	00	10	X	0	0	0	X	X
WB – R	X	0	0	0	X	X	X	X	0	0	1	1	0
Calcolo – beq	X	0	0	0	1	00	01	01	0	1	0	X	X
Calcolo – j	X	0	0	0	X	X	X	10	1	0	0	X	X



Sintesi della FSM della CPU



- Stato – passo di esecuzione.
- Uscita – segnali di controllo.
- Ingressi – Codice operativo.

I valori dei segnali di controllo dipendono:

- dal passo dell'istruzione (stato)

Il passo successivo dell'istruzione (stato prossimo) dipende:

- dal codice operativo (ingresso).
- dal passo presente (stato presente).

•Uscita = f(Stato)

•Stato_prossimo = f(Ingressi, Stato_presente)



Macchina a Stati Finiti (di Moore)



La Macchina di Moore è definita, in teoria degli automi, dalla quintupla :
 $\langle X, I, Y, f(\cdot), g(\cdot) \rangle$

X: insieme degli stati (in numero finito).

I: alfabeto di ingresso: tutti i simboli che si possono presentare in ingresso. Se abbiamo n ingressi, avremo 2^n possibili simboli da leggere in ingresso (configurazioni).

Y: alfabeto di uscita: tutti i simboli che si possono generare in uscita. Se abbiamo m uscite, avremo 2^m possibili simboli da presentare in uscita (configurazioni).

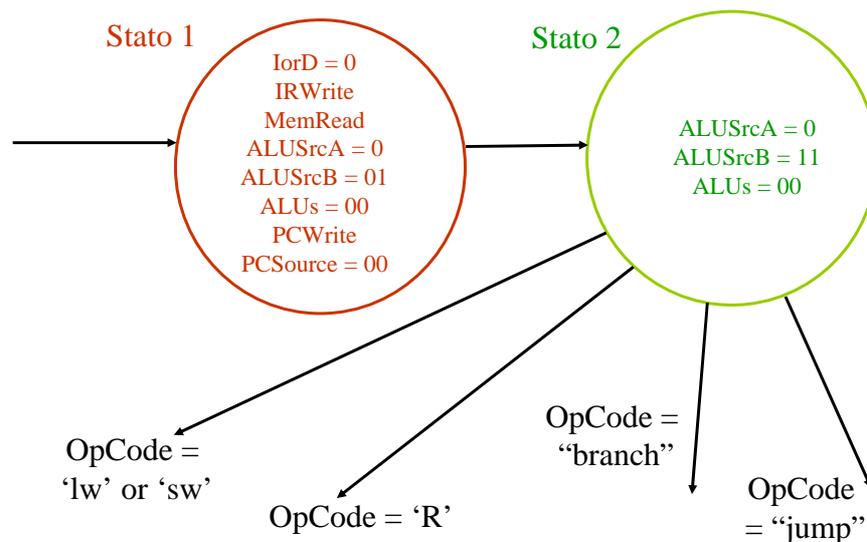
f(.): funzione stato prossimo: $X' = f(X, I)$. Definisce l'evoluzione della macchina nel tempo. L'evoluzione è deterministica.

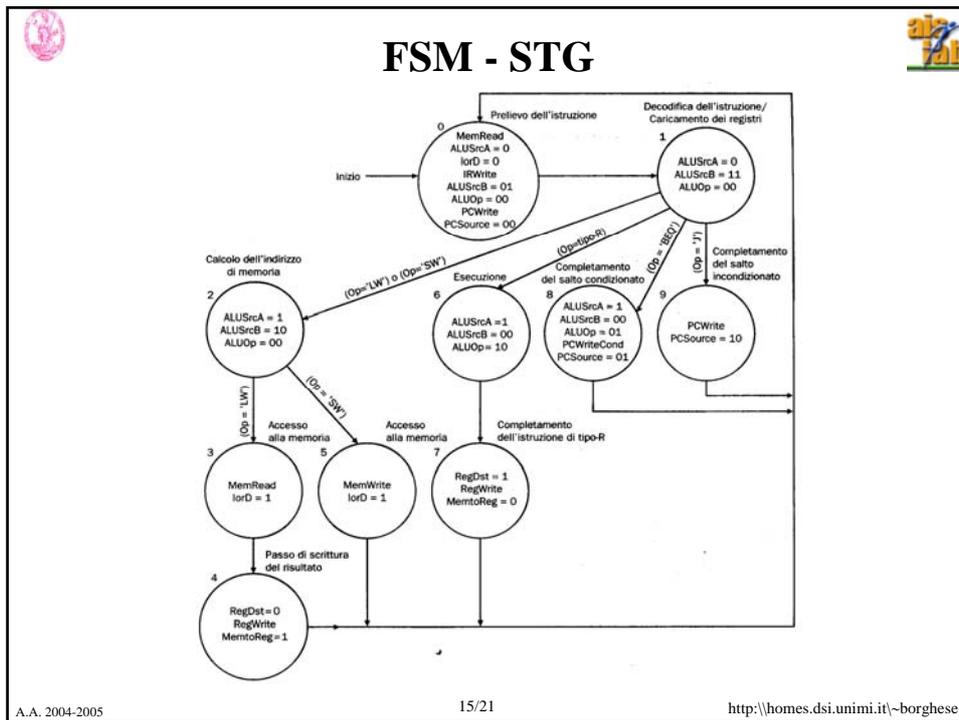
g(.): funzione di uscita: $Y = g(X)$ nelle macchine di Moore.

Per il buon funzionamento della macchina è previsto uno stato iniziale, al quale la macchina può essere portata mediante un comando di reset.



Fase di fetch e decodifica





FSM - STT

Stato (Passo di esecuzione)	Control Signals																	
	OpCode = R	OpCode = sw	OpCode = lw	OpCode = beq	OpCode = j	IrD	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
Calcolo - lw/sw - 2						X	0	0	0	1	10	00	X	0	0	0	X	X
Memoria - lw - 3						1	1	0	0	X	X	X	X	0	0	0	X	X
WB - lw - 4						X	0	1	0	X	X	X	X	0	0	1	0	1
Memoria - sw - 5						1	0	1	0	X	X	X	X	0	0	0	X	X
Calcolo - R - 6						X	0	0	0	1	00	10	X	0	0	0	X	X
WB - R - 7						X	0	0	0	X	X	X	X	0	0	1	1	0
Calcolo - beq - 8						X	0	0	0	1	00	01	01	0	1	0	X	X
Calcolo - j - 9						X	0	0	0	X	X	X	10	1	0	0	X	X

A.A. 2004-2005 16/21 http://homes.dsi.unimi.it/~borgnese



FSM - STT



Stato (Passo di esecuzione)	OpCode = R	OpCode = sw	OpCode = lw	OpCode = beq	OpCode = j	lOrd	MemRead	MemWrite	IRWrite	ALUScrA	ALUScrB	ALUs	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
Calcolo - lw/sw - 2	X	5	3	X	X	X	0	0	0	1	10	00	X	0	0	0	X	X
Memoria - lw - 3	X	X	4	X	X	1	1	0	0	X	X	X	X	0	0	0	X	X
WB - lw - 4	X	X	0	X	X	X	0	1	0	X	X	X	X	0	0	1	0	1
Memoria - sw - 5	X	0	X	X	X	1	0	1	0	X	X	X	X	0	0	0	X	X
Calcolo - R - 6	7	X	X	X	X	X	0	0	0	1	00	10	X	0	0	0	X	X
WB - R - 7	0	X	X	X	X	X	0	0	0	X	X	X	X	0	0	1	1	0
Calcolo - beq - 8	X	X	X	0	X	X	0	0	0	1	00	01	01	0	1	0	X	X
Calcolo - j - 9	X	X	X	X	0	X	0	0	0	X	X	X	10	1	0	0	X	X



FSM - sintesi della funzione di uscita



Stato $s_0 s_1 s_2 s_3$	lOrd	MemRead	MemWrite	IRWrite	ALUScrA	ALUScrB	ALUs	PCSource	PCWrite	PCWriteCond	RegWrite	RegDst	MemtoReg
Fase fetch - 0000	0	1	0	1	0	01	10	00	1	0	0	X	X
Decodifica - 0001	X	0	0	0	0	11	10	X	0	0	0	X	X
Calcolo - lw/sw - 0010	X	0	0	0	1	10	00	X	0	0	0	X	X
Memoria - lw - 0011	1	1	0	0	X	X	X	X	0	0	0	X	X
WB - lw - 0100	X	0	1	0	X	X	X	X	0	0	1	0	1
Memoria - sw - 0101	1	0	1	0	X	X	X	X	0	0	0	X	X
Calcolo - R - 0110	X	0	0	0	1	00	10	X	0	0	0	X	X
WB - R - 0111	X	0	0	0	X	X	X	X	0	0	1	1	0
Calcolo - beq - 1000	X	0	0	0	1	00	01	01	0	1	0	X	X
Calcolo - j - 1001	X	0	0	0	X	X	X	10	1	0	0	X	X

Esmpi:
 $Y(\text{RegWrite}) = (\text{Stato} == 7) \text{ OR } (\text{Stato} == 4)$
 $Y(\text{ALUScrA}) = (\text{Stato} == 8) \text{ OR } (\text{Stato} == 6)$
 $\text{OR } (\text{Stato} == 2)$



FSM – codifica della STT (stato futuro)



Passo esecuzione	OpCode = R 000000	OpCode = sw 101011	OpCode = lw 100011	OpCode = beq 000100	OpCode = j 000010
Fase fetch - 0000	0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1
Decodifica - 0001	0 1 1 0	0 0 1 0	0 0 1 0	1 0 0 0	1 0 0 1
Calcolo – lw/sw - 0010	X X X X	0 1 0 1	0 0 1 1	X X X X	X X X X
Memoria – lw - 0011	X X X X	X X X X	0 1 0 0	X X X X	X X X X
WB – lw - 0100	X X X X	X X X X	0 0 0 0	X X X X	X X X X
Memoria – sw – 0101	X X X X	0 0 0 0	X X X X	X X X X	X X X X
Calcolo – R - 0110	0 1 1 1	X X X X	X X X X	X X X X	X X X X
WB – R - 0111	0 0 0 0	X X X X	X X X X	X X X X	X X X X
Calcolo – beq .- 1000	X X X X	X X X X	X X X X	0 0 0 0	X X X X
Calcolo – j - 1001	X X X X	X X X X	X X X X	X X X X	0 0 0 0



FSM – circuito dello stato futuro



Esempio: $S_0(t+1) = (\overline{S_0} \overline{S_1} \overline{S_2} \overline{S_3}) + (S_0 \overline{S_1} \overline{S_2} \overline{S_3})(i_0 i_1 i_2 i_3 i_4 i_5) + (\overline{S_0} \overline{S_1} \overline{S_2} \overline{S_3})(i_0 i_1 i_2 i_4 i_5) + (\overline{S_0} \overline{S_1} \overline{S_2} \overline{S_3})$

Passo esecuzione	OpCode = R 000000	OpCode = sw 101011	OpCode = lw 100011	OpCode = beq 000100	OpCode = j 000010
Fase fetch - 0000	0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1
Decodifica - 0001	0 1 1 0	0 0 1 0	0 0 1 0	1 0 0 0	1 0 0 1
Calcolo – lw/sw - 0010	X X X X	0 1 0 1	0 0 1 1	X X X X	X X X X
Memoria – lw - 0011	X X X X	X X X X	0 1 0 0	X X X X	X X X X
WB – lw - 0100	X X X X	X X X X	0 0 0 0	X X X X	X X X X
Memoria – sw – 0101	X X X X	0 0 0 0	X X X X	X X X X	X X X X
Calcolo – R - 0110	0 1 1 1	X X X X	X X X X	X X X X	X X X X
WB – R - 0111	0 0 0 0	X X X X	X X X X	X X X X	X X X X
Calcolo – beq .- 1000	X X X X	X X X X	X X X X	0 0 0 0	X X X X
Calcolo – j - 1001	X X X X	X X X X	X X X X	X X X X	0 0 0 0



Sommario



I segnali di controllo della CPU multi-ciclo

Sintesi dell'Unità di Controllo come Macchina a Stati Finiti