

# Argomenti

---

---

- Processori microprogrammati
  - » Microistruzioni e microprogrammi
  - » Esempio di architettura microprogrammata
- Architetture CISC
- Architetture RISC
  - » Pipeline
  - » Architetture superscalari



# Processori microprogrammati

---

---

- Istruzioni macchina “complesse”
  - » L’esecuzione di un’istruzione macchina richiede generalmente un data path complesso cioè l’espletamento di più operazioni di trasferimento elementari (microoperazioni)
  - » Il codice operativo macchina non riesce a contenere tutte le informazioni di tempificazione ed abilitazione dei circuiti e dei registri coinvolti



# Esempio di architettura microprogrammata - Data Path

## ➤ Data Path

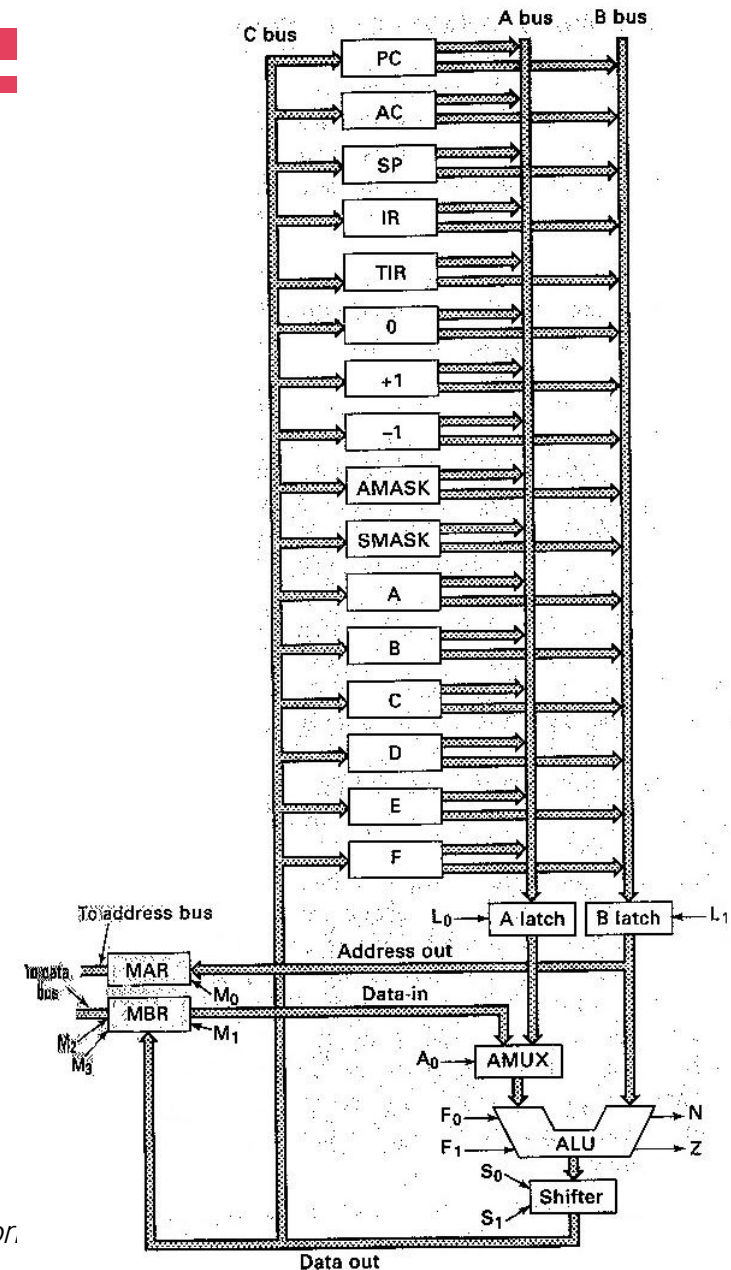
### » Registri generali

◆ A, B, C, D, E, F

### » Registri speciali

### » Bus interni

◆ A, B, C



# Esempio di architettura microprogrammata – Instruction set

## ➤ Livello Assembly

AN EXAMPLE MACROARCHITECTURE

Binary	Mnemonic	Instruction	Meaning
0000xxxxxxxxxxxx	LODD	Load direct	$ac := m[x]$
0001xxxxxxxxxxxx	STOD	Store direct	$m[x] := ac$
0010xxxxxxxxxxxx	ADDD	Add direct	$ac := ac + m[x]$
0011xxxxxxxxxxxx	SUBD	Subtract direct	$ac := ac - m[x]$
0100xxxxxxxxxxxx	JPOS	Jump positive	if $ac \geq 0$ then $pc := x$
0101xxxxxxxxxxxx	JZER	Jump zero	if $ac = 0$ then $pc := x$
0110xxxxxxxxxxxx	JUMP	Jump	$pc := x$
0111xxxxxxxxxxxx	LOCO	Load constant	$ac := x$ ( $0 \leq x \leq 4095$ )
1000xxxxxxxxxxxx	LODL	Load local	$ac := m[sp + x]$
1001xxxxxxxxxxxx	STOL	Store local	$m[x + sp] := ac$
1010xxxxxxxxxxxx	ADDL	Add local	$ac := ac + m[sp + x]$
1011xxxxxxxxxxxx	SUBL	Subtract local	$ac := ac - m[sp + x]$
1100xxxxxxxxxxxx	JNEG	Jump negative	if $ac < 0$ then $pc := x$
1101xxxxxxxxxxxx	JNZE	Jump nonzero	if $ac \neq 0$ then $pc := x$
1110xxxxxxxxxxxx	CALL	Call procedure	$sp := sp - 1; m[sp] := pc; pc := x$
1111000000000000	PSHI	Push indirect	$sp := sp - 1; m[sp] := m[ac]$
1111001000000000	POPI	Pop indirect	$m[ac] := m[sp]; sp := sp + 1$
1111010000000000	PUSH	Push onto stack	$sp := sp - 1; m[sp] := ac$
1111011000000000	POP	Pop from stack	$ac := m[sp]; sp := sp + 1$
1111100000000000	RETN	Return	$pc := m[sp]; sp := sp + 1$
1111101000000000	SWAP	Swap ac, sp	$tmp := ac; ac := sp; sp := tmp$
11111100yyyyyyyy	INSP	Increment sp	$sp := sp + y$ ( $0 \leq y \leq 255$ )
11111110yyyyyyyy	DESP	Decrement sp	$sp := sp - y$ ( $0 \leq y \leq 255$ )

xxxxxxxxxxxx is a 12-bit machine address; in column 4 it is called x.  
 yyyyyy as an 8-bit constant; in column 4 it is called y.

DIS

Fig. 4-14. The Mac-1 instruction set.



# Esempio di architettura microprogrammata - Esecuzione di un codice operativo

---

---

- **ADDD X           \*\* AC:=AC + M[x] \*\***
  
- **Fetch**
  - » **MAR:=PC**
  - » **Abilita l'operazione READ in memoria**
  - » **IR:=MBR**
  - » **PC:=PC+1**
  - » **Decodifica istruzione**



# Esempio di architettura microprogrammata - Esecuzione di un codice operativo (2)

---

---

- Operand Assembly
  - »  $MAR := IR.X$
  - » Abilita l'operazione READ in memoria
  
- Execute
  - »  $AC := AC + MBR$



# Microprogramma e microistruzioni

---

---

## ➤ Microistruzioni

- » Specifica una microoperazione
  - ◆ Calcolo o Trasferimento elementare
- » Ingloba le informazioni di tempificazione e di abilitazione dei circuiti e dei registri coinvolti nella microoperazione

## ➤ Microprogramma

- » Sequenza ordinata di microistruzioni che esegue le istruzioni macchina



# Supporto hardware

---

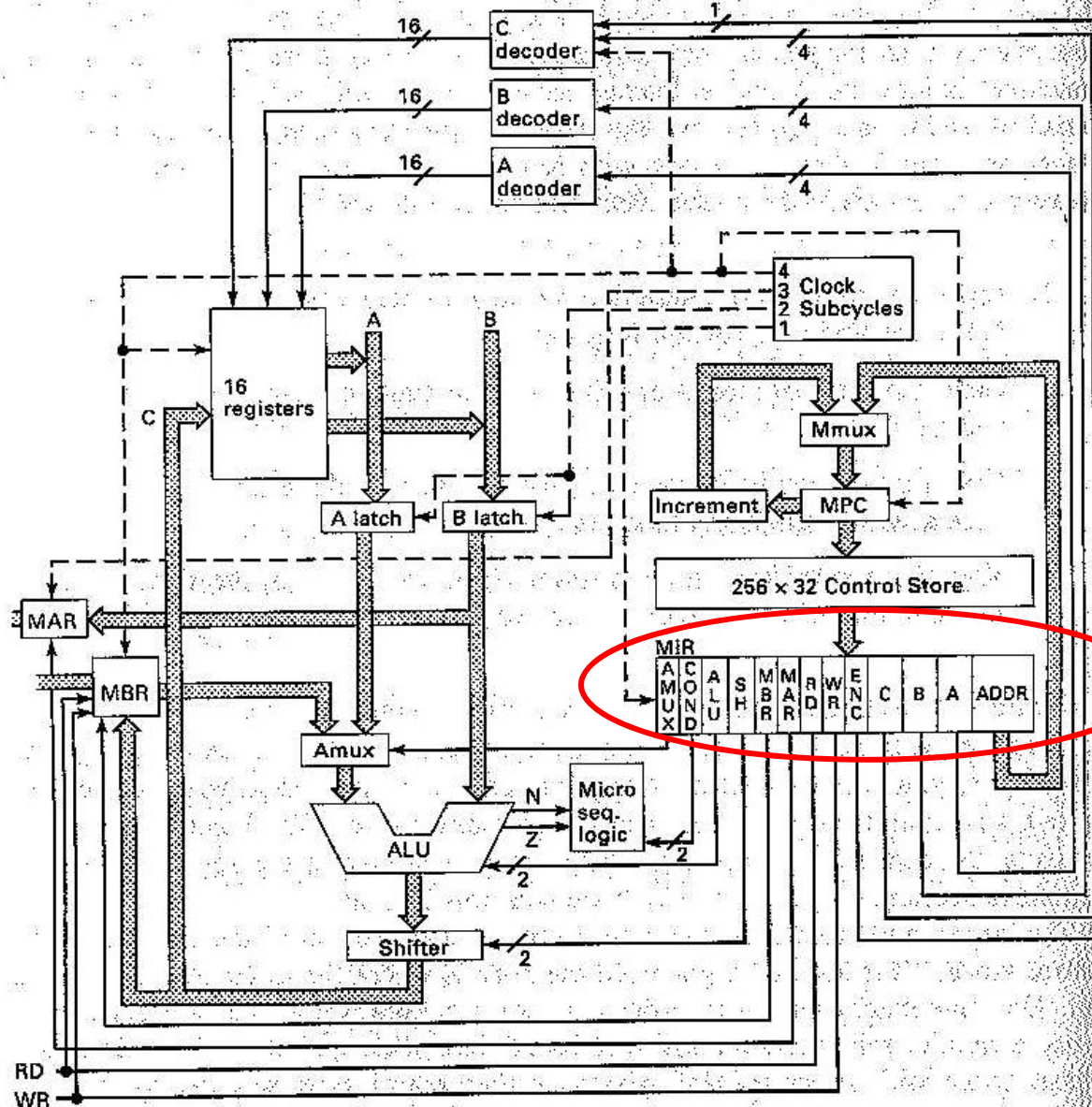
---

- Control storage (logica firmware)
  - » Memoria ROM per la memorizzazione dei microprogrammi
- Micro Instruction Register
  - » Registro per la memorizzazione della microistruzione corrente
- Micro Program Counter
  - » Registro per la memorizzazione dell'indirizzo della prossima microistruzione da eseguire





# Esempio di architettura microprogrammata

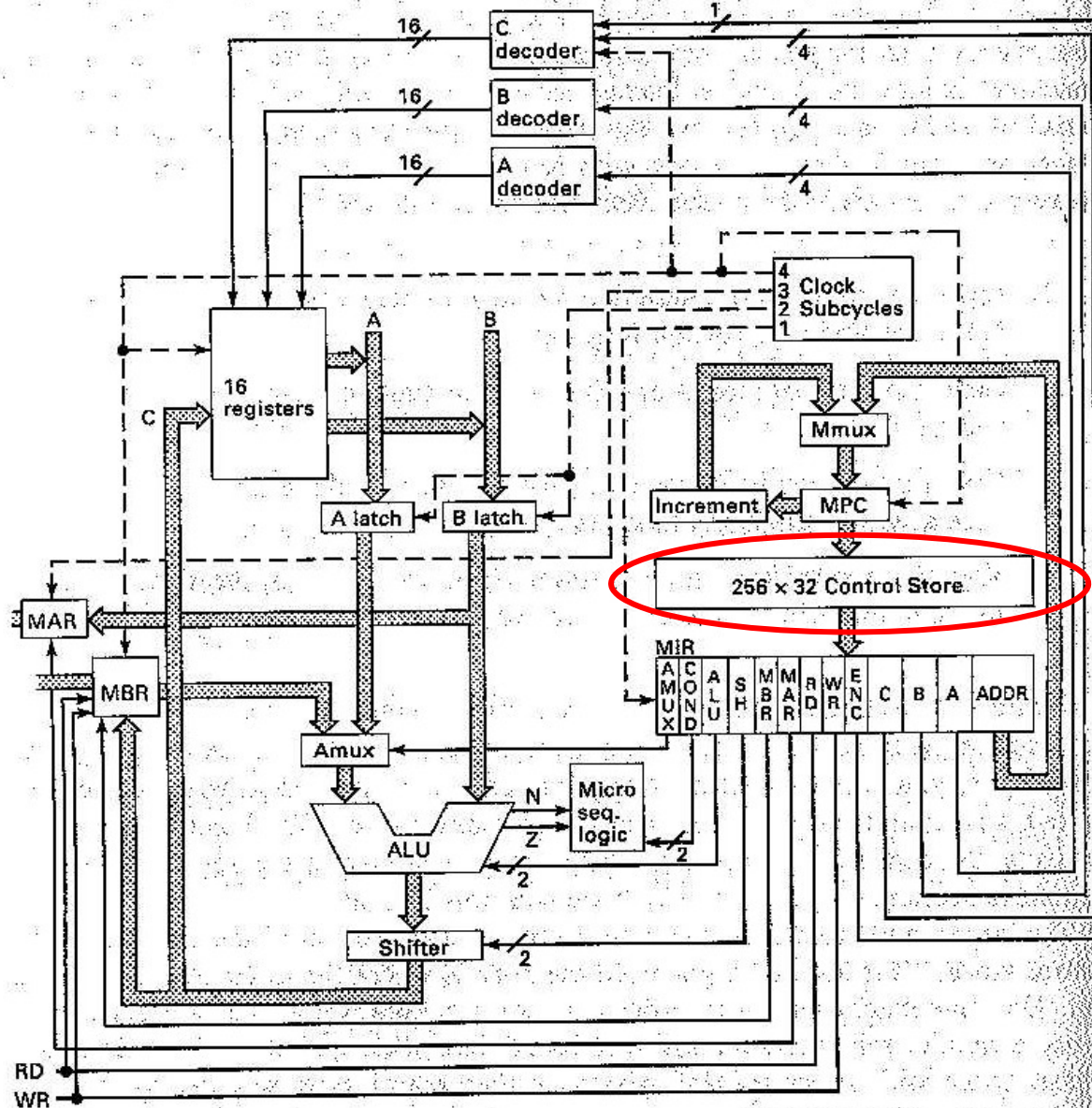


Micro Instruction Register

Fig. 4-10. The complete block diagram of our example microarchitecture.



# Esempio di architettura microprogrammata



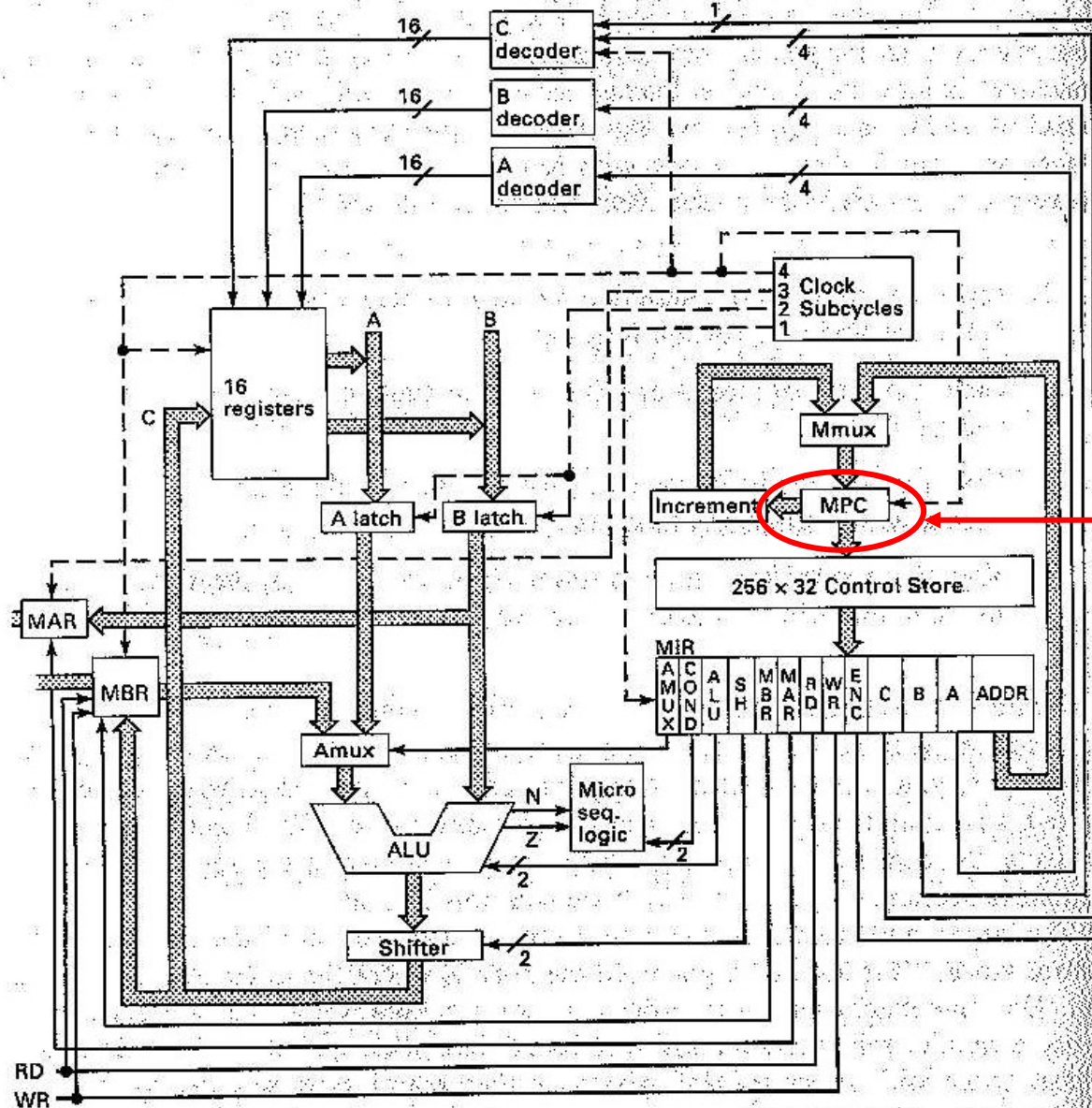
Memoria di controllo

Fig. 4-10. The complete block diagram of our example microarchitecture.





# Esempio di architettura microprogrammata



Micro Program Counter

Fig. 4-10. The complete block diagram of our example microarchitecture.



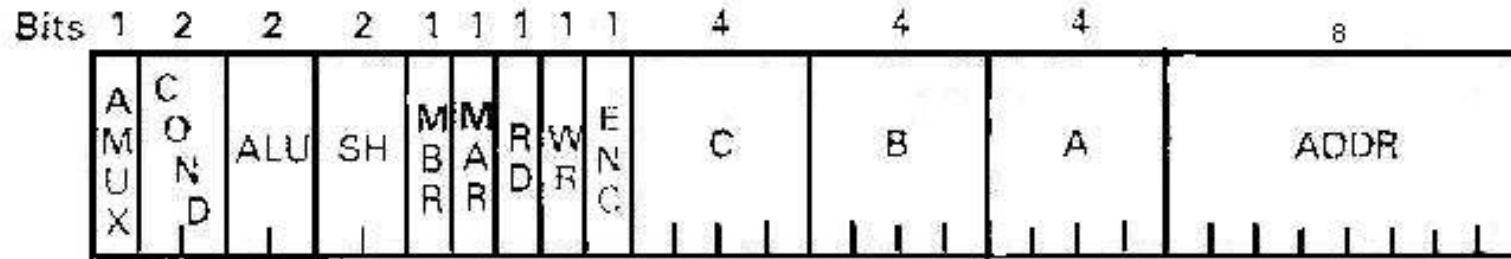
# Esempio di architettura microprogrammata – Formato Microistruzione

1000

00000000

00000000

00000000



<u>AMUX</u>	<u>COND</u>	<u>ALU</u>	<u>SH</u>	<u>MBR MAR, RD, WR, ENC</u>
0 = A latch	0 = No jump	0 = A + B	0 = No shift	0 = No
1 = MBR	1 = Jump if N = 1	1 = A AND B	1 = Shift right 1 bit	1 = Yes
	2 = Jump if Z = 1	2 = $\frac{A}{2}$	2 = Shift left 1 bit	
	3 = Jump always	3 = A	3 = (not used)	

Fig. 4-9. The microinstruction layout for controlling the data path of Fig. 4-8.



# Esempio di architettura microprogrammata – Tempificazione

---

---

- Clock interno a 4 fasi derivato dal clock esterno
  - » Fase 1 – Caricamento prossima microistruzione in MIR
  - » Fase 2 – Acquisizione dei valori dei bus A e B nei registri A-latch e B-latch
  - » Fase 3 – Ora che gli input all'ALU sono stabili si abilita l'ALU all'operazione
  - » Fase 4 – Ora che il risultato dell'operazione è stabile in Shifter si abilita la scrittura del risultato
- Esecuzione di una microistruzione per ogni ciclo di clock esterno
- L'esecuzione di un'istruzione macchina richiede un numero di cicli di clock pari al numero di microistruzioni necessarie



# Esempio di architettura microprogrammata - Codifica microistruzioni

Microoperazione	A	C											A
	M	O	A	M	M							E	D
	U	N	L	S	B	A	R	W	N				D
	X	D	U	H	R	R	D	D	C	C	B	A	R
MAR:= PC; RD	0	0	2	0	0	1	1	0	0	0	0	0	00
RD	0	0	2	0	0	0	1	0	0	0	0	0	00
IR:=MBR	1	0	2	0	0	0	0	0	1	3	0	0	00

⋮

⋮



# Esempio di architettura microprogrammata – Esecuzione delle microistruzioni

## ➤ Esempio IR:=MBR

### » Fase 1 – Caricamento MIR

A	C												A
M	O	A		M	M			E					D
U	N	L	S	B	A	R	W	N					D
X	D	U	H	R	R	D	D	C	C	B	A		R
1	0	2	0	0	0	0	0	1	3	0	0	00	MIR

- » Fase 2 – Caricamento dei registri A-latch e B-latch. In questo caso i valori acquisiti non verranno utilizzati per l'esecuzione della microistruzione
- » Fase 3 – Abilitazione ALU  
AMUX=1 seleziona il registro MBR  
ALU=2 lascia passare il dato senza effettuare modifiche
- » Fase 4 – Scrittura del risultato  
ENC = 1 forza la scrittura nei registri interni  
C = 3 seleziona il registro IR



# Architetture CISC e RISC

---

---

- CISC (Complex Instruction Set Computer)
  - » Utilizzo del *transistor budget* per massimizzare la taglia dell'instruction set
  - » Architettura microprogrammata
  - » Esempi: Intel X86, Pentium, P6
  
- RISC (Reduced Instruction Set Computer)
  - » Utilizzo del *transistor budget* per velocizzare un repertorio limitato di istruzioni
  - » Architettura non microprogrammata
  - » Esempi: MIPS RX000, SPARC, IBM PowerPC





# Pipeline

---

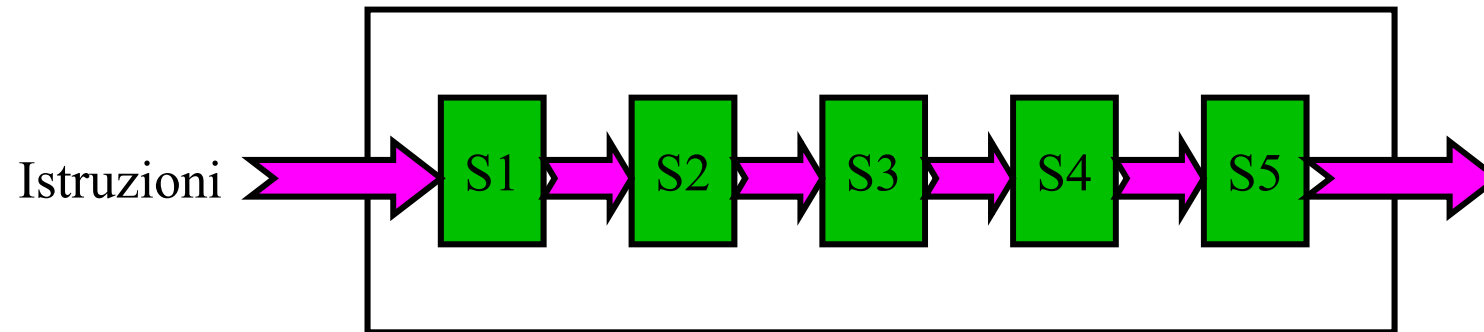
---

- scomposizione di una istruzione in stadi S1, S2, ... Sd stadi
- ogni stadio richiede un ciclo di clock
- sovrapposizione di stadi diversi di istruzioni diverse

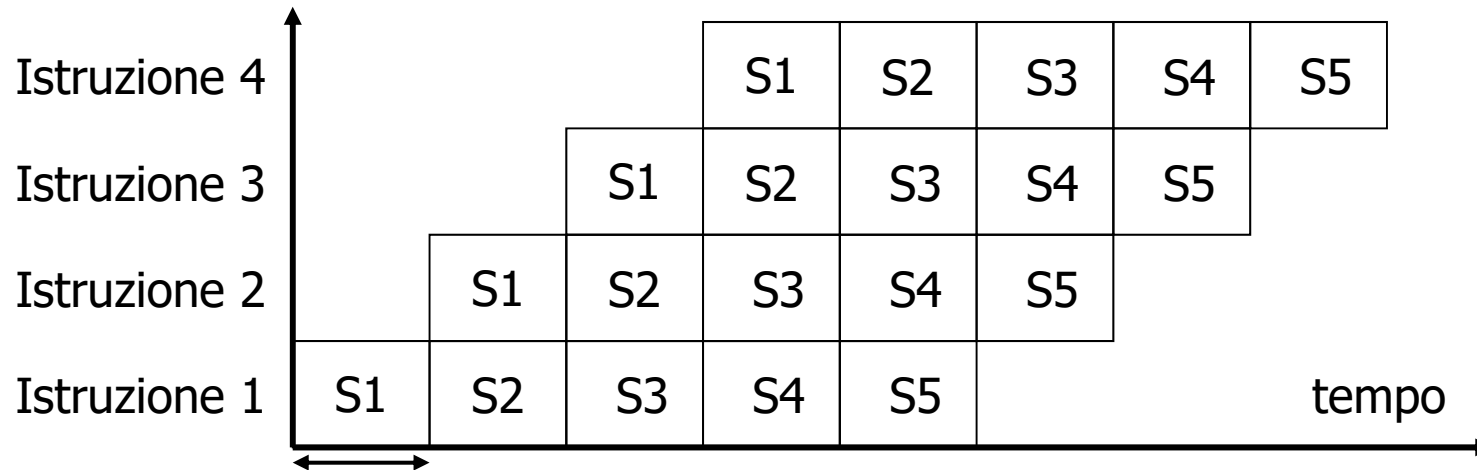


# Pipeline

- Esempio pipeline lineare a 5 stadi



- Tempificazione

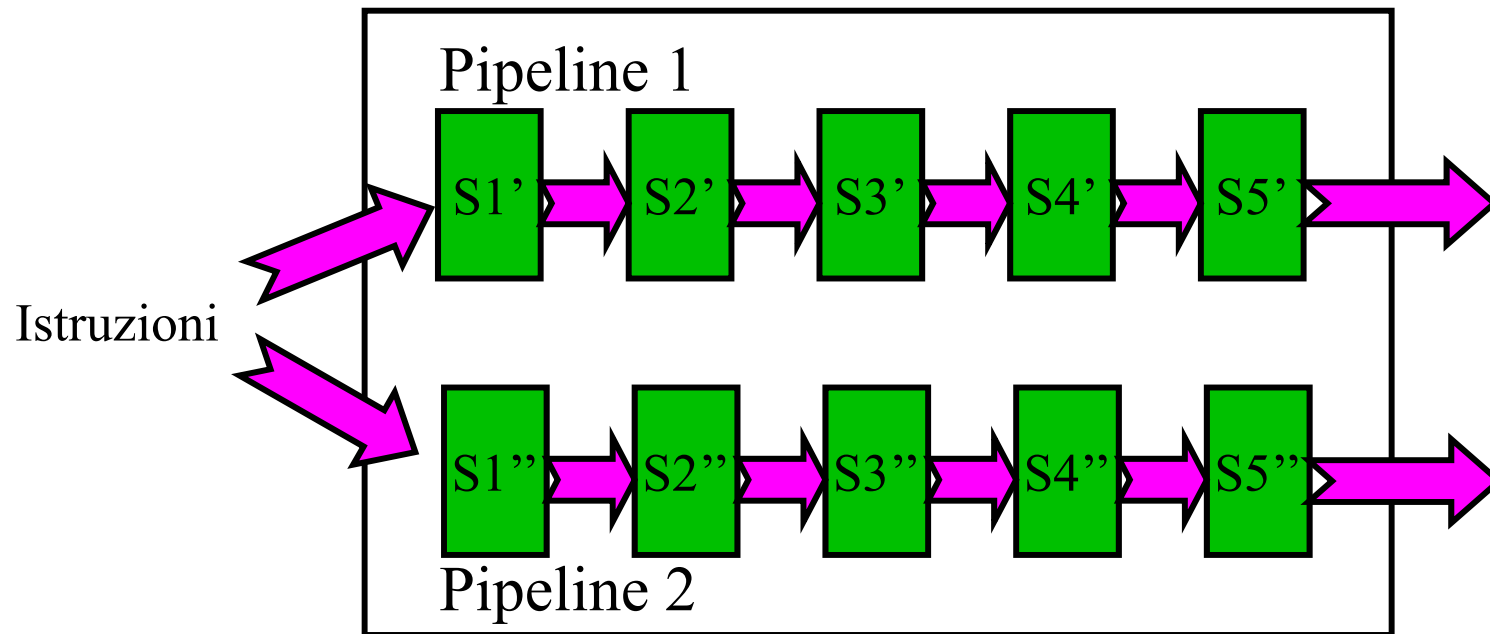


Ciclo di clock



# Superscalarità

- Capacità di una architettura di eseguire più di una istruzione per ciclo di clock, ad es., tramite parallelismo nella circuiteria



# Performance

---

---

- **MIPS (Million Instructions Per Second)**
  - » dipende dall'Instruction set (poco obiettivo)
- **FLOPS (FLoating-point Operations Per Second)**
  - » dipende dall'insieme di istruzioni floating-point offerte dall'architettura (più obiettivo rispetto a MIPS)
- **Benchmark:** programmi scritti in linguaggi ad alto livello rispetto ai quali si valutano le architetture. E' la misura più obiettiva e più usata



# Tecnologia

---

---

- **VLSI (Very Large Scale Integration )**
  - » tecnologia per realizzare circuiti integrati su **chip** di silicio
- **Microprocessore: Processore su un unico chip**
  - » Dimensione chip =  $O(1) \text{ cm}^2$
  - » No. transistor per chip =  $[10^7, 10^8]$

