

Calcolatori Elettronici

Classificazione dei calcolatori elettronici
Sistemi basati sull'architettura di Von Neumann
Sistemi basati sull'architettura Harward
Architettura dei calcolatori: definizioni
Evoluzione dell'architettura

Architettura della CPU
Ciclo di Istruzione
Esempio di data-path: il DLX
Unità di controllo cablata e microprogrammata

Classificazione dei calcolatori elettronici

Una classificazione obsoleta, basata su costi:

- Supercomputer > 5 M\$
- Mainframe 0.5-5 M\$
- Minicomputer 50-500 M\$
- Microcomputer K\$

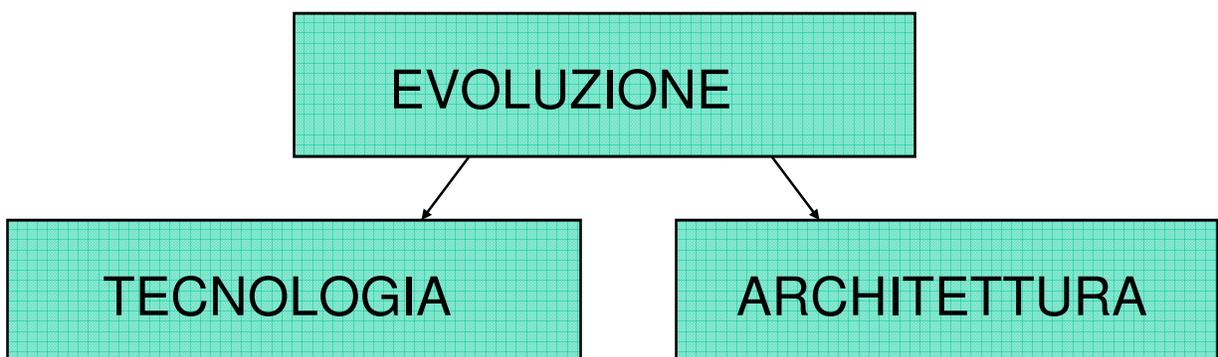
Oggi...

- Workstation vs. PC
- Server vs. client

I computer sono pervasivi:

- sistemi embedded
- Portable computer (palmari ecc...)
- Wearable computer? Primi prototipi

Le prestazioni aumentano del 25,30% l'anno per WS e PC



ARCHITETTURA

ARCHITETTURA DI UN CALCOLATORE?

I principali aspetti architettonici riguardano:

Architettura dell'instruction set

l'architettura come è visibile dal programmatore

Organizzazione:

gli aspetti di progetto di alto livello
(il sottosistema di memorie, la gestione dei bus
la struttura interna della CPU...)

Hardware

il progetto specifico di ogni macchina
(il progetto logico e le specifiche soluzioni)

Tecnologia (esempio FSB)

<u>CPU</u>	Frequency [MHz]	Transfer rate [MB/s]
<u>Intel 80386</u>	12-40	46-153
<u>Intel 80486</u>	16-50	61-191
<u>Intel Pentium</u>	50-66	381-503
<u>Intel Celeron</u>	66-200	503-6104
<u>Intel Pentium III</u>	100-133	762-1014
<u>AMD Athlon XP</u>	133-200	2029-3052
<u>Intel Pentium 4</u>	100-266.5	3051-8133
<u>Intel Pentium M</u>	100-133.25	3052-4066
<u>VIA C7</u>	100-200	3052-6103
<u>Intel Core 2</u>	133.25-400	4066-12207

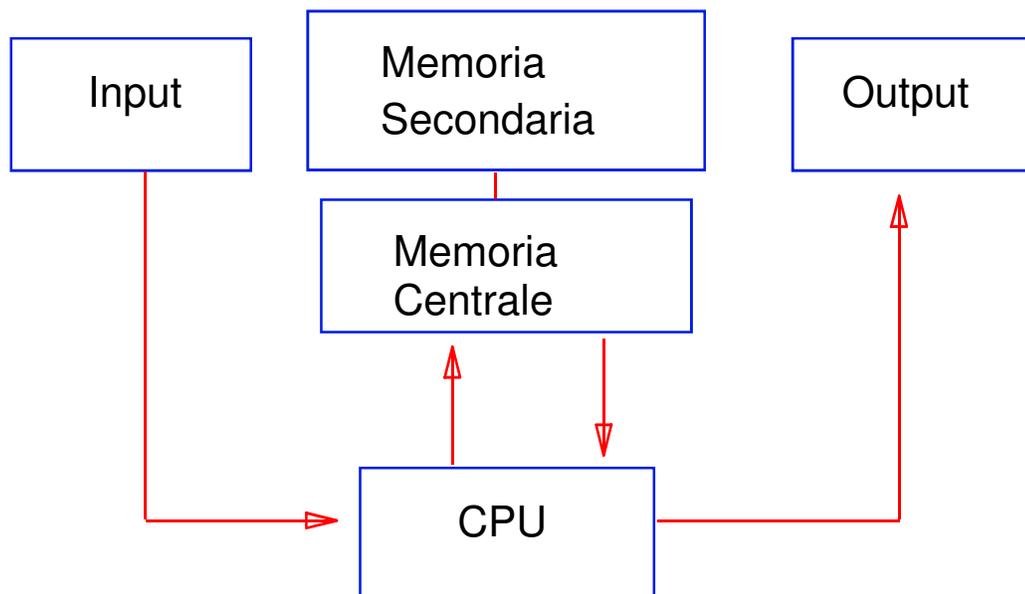
Front Side Bus rate

ARCHITETTURA DELLE CPU

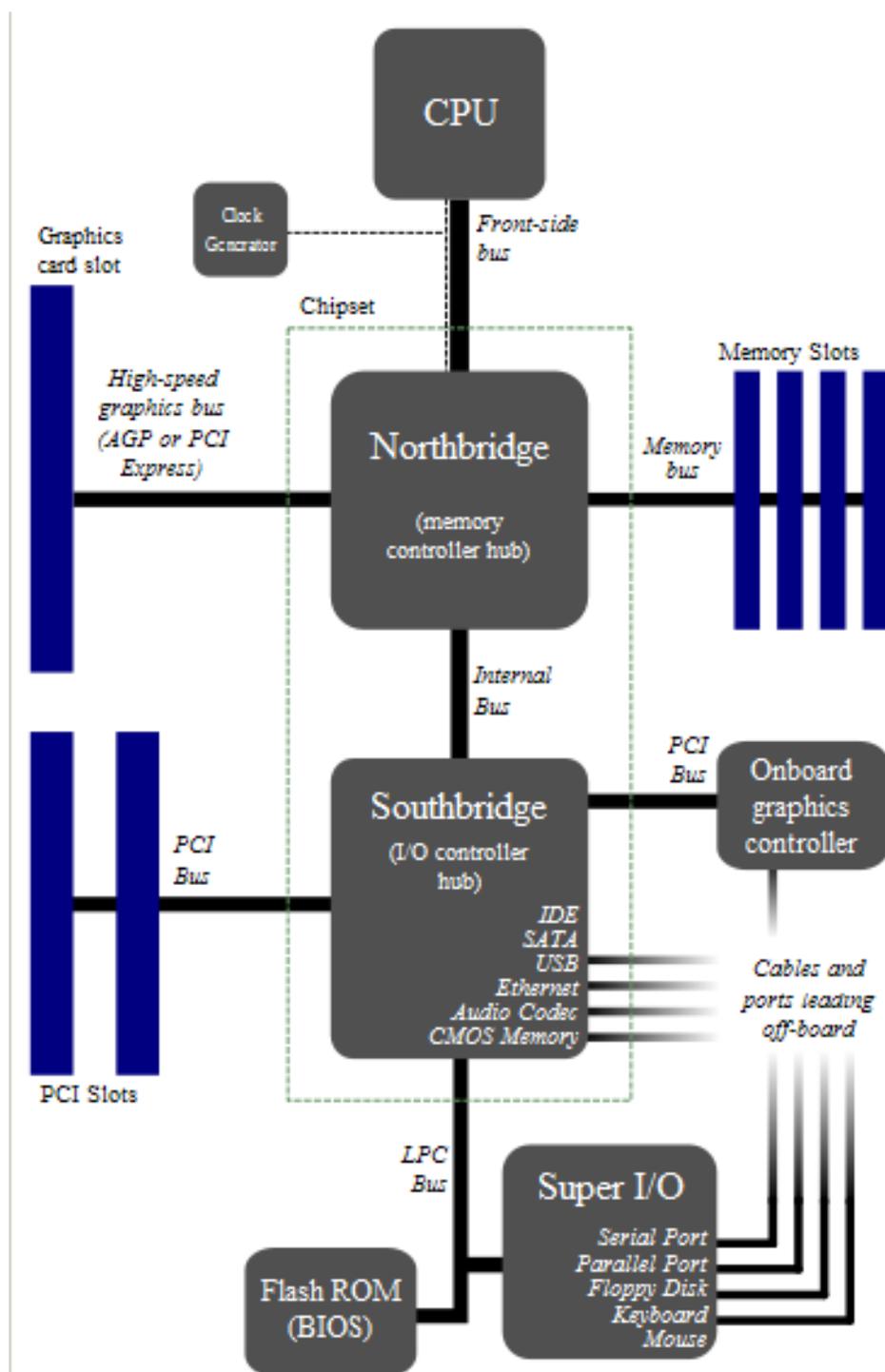
ARCHITETTURA DI VON NEUMANN (1946)

(STORED PROGRAM COMPUTER)

1. **MEMORIA** per contenere istruzioni e dati
2. **CPU** composta da una *unità di controllo* per eseguire il FETCH delle istruzioni e da una *unità di esecuzione* aritmetica che esegue l'istruzione sugli operandi
3. **Meccanismi di INPUT/OUTPUT** (con dispositivi periferici) per l'interfaccia con il mondo esterno



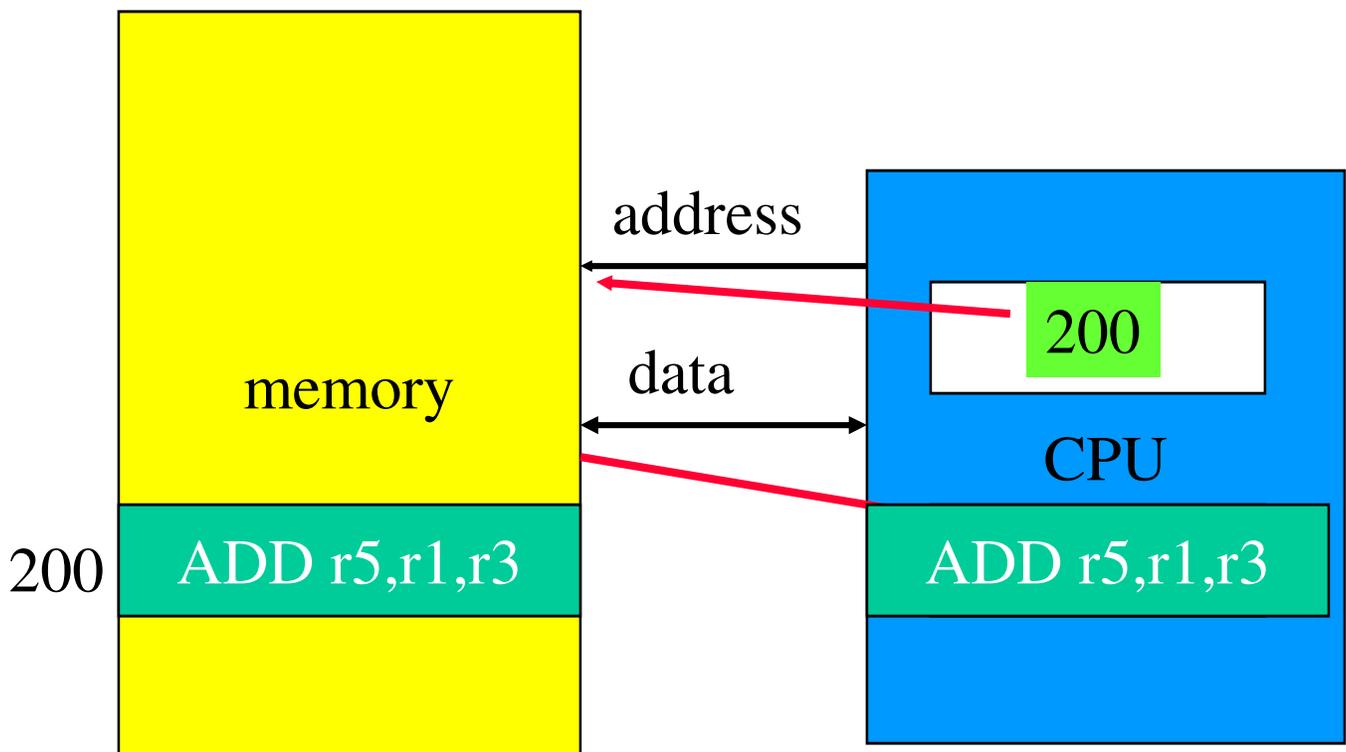
Architettura Pentium



ARCHITETTURA DELLE CPU

ARCHITETTURA VON NEUMANN

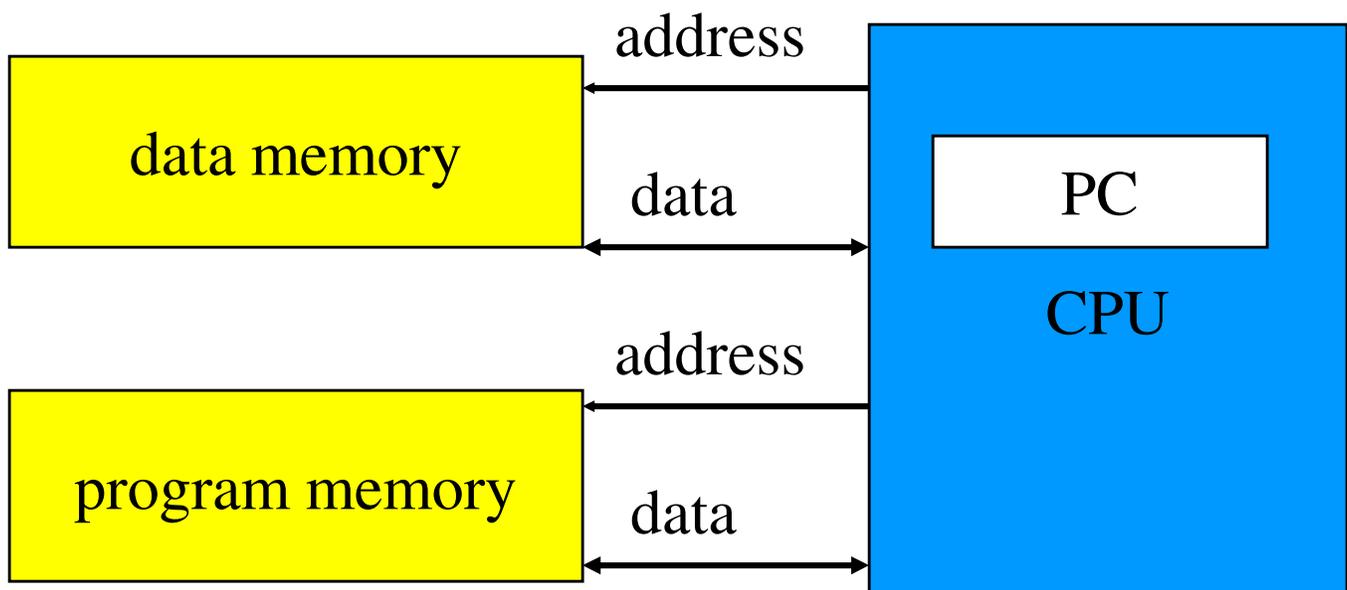
- La memoria contiene sia dati che istruzioni.
- La CPU fetcha istruzioni dalla memoria.
 - CPU e memorie separate identificano i computer programmabili.
- Quali registri di CPU servono: program counter (PC), instruction register (IR), general-purpose registers, etc.



ARCHITETTURA DELLE CPU

ARCHITETTURA HARWARD

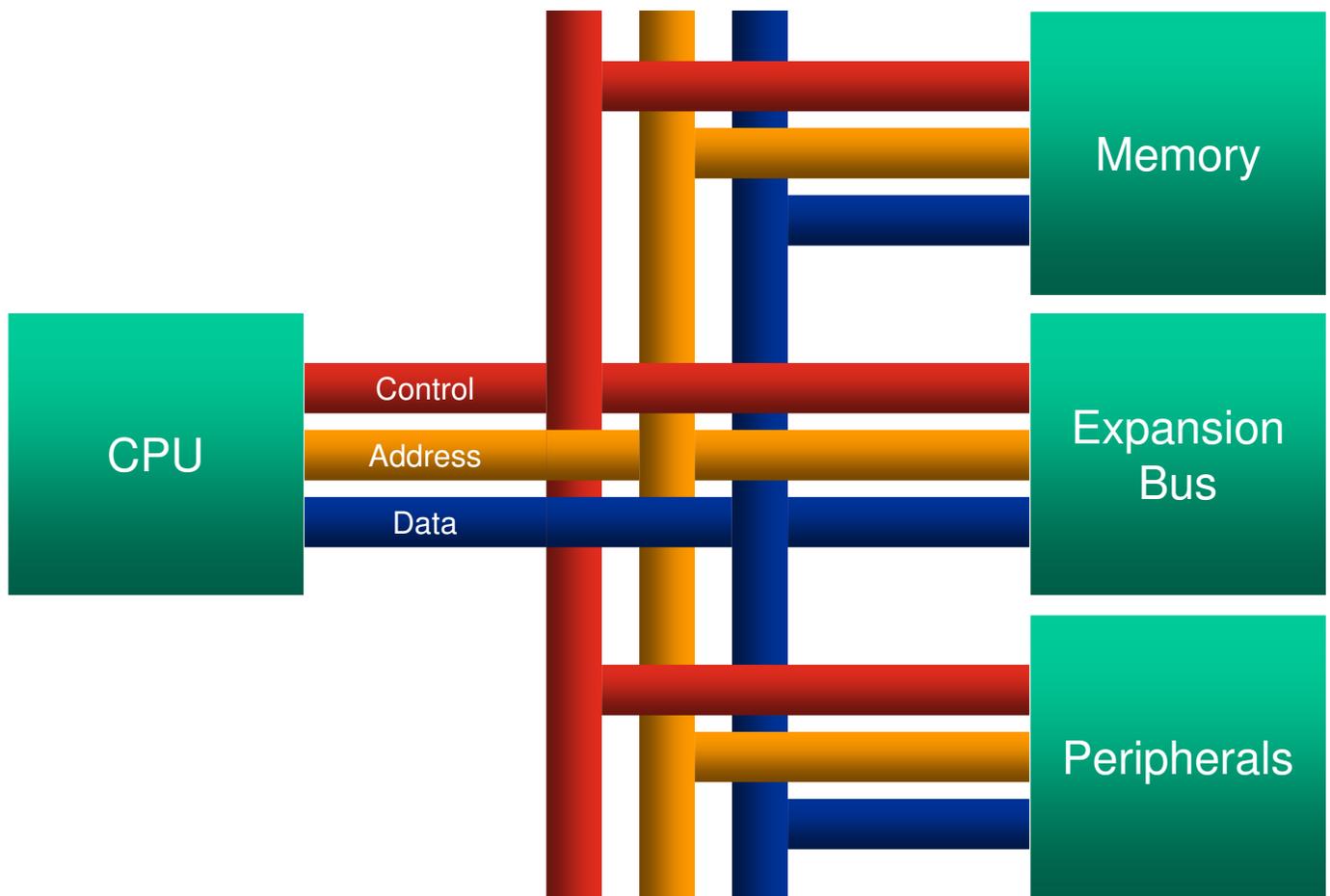
- Due diverse unità di memoria: una per il programma e una per i dati.
- La CPU esegue il fetch dalla memoria di programma.
- Es: Digital ALPHA è Workstation UNIX della Digital con processore RISC e architettura Harward



HARWARD vs VON NEUMANN

- Harvard non può utilizzare codice “self-modifying”.
- Harvard permette due fetch alla memoria simultanei (dati e programma).
- La maggior parte dei DSPs usa l’architettura Harvard per eseguire data streaming:
 - maggiore transfer rate verso le memorie;
 - Transfer rate maggiormente predicibile.
- Harvard però è una architettura più rigida
 - Dimensioni prefissate di memoria di programma
 - Dimensioni prefissate di memoria dati

Architettura di un computer



EVOLUZIONE

EVOLUZIONE DELLA CPU:

- aumento dell'integrazione e della frequenza di clock
- evoluzione dell'architettura interna
- parallelismo
- pipelining
- multithreading...

EVOLUZIONE DEL SISTEMA DI MEMORIE:

- diminuzione di tempi d'accesso e aumento del parallelismo (bus)
- gerarchie di memorie
 - cache
 - memoria virtuale

EVOLUZIONE DEI SISTEMI DI I/O:

- miglioramenti tecnologici...
- nuove interfacce
- nuove applicazioni (multimedia..)
- Periferiche sempre più "autonome" in grado di svolgere compiti complessi

MICROPROCESSORI

MICROPROCESSORI

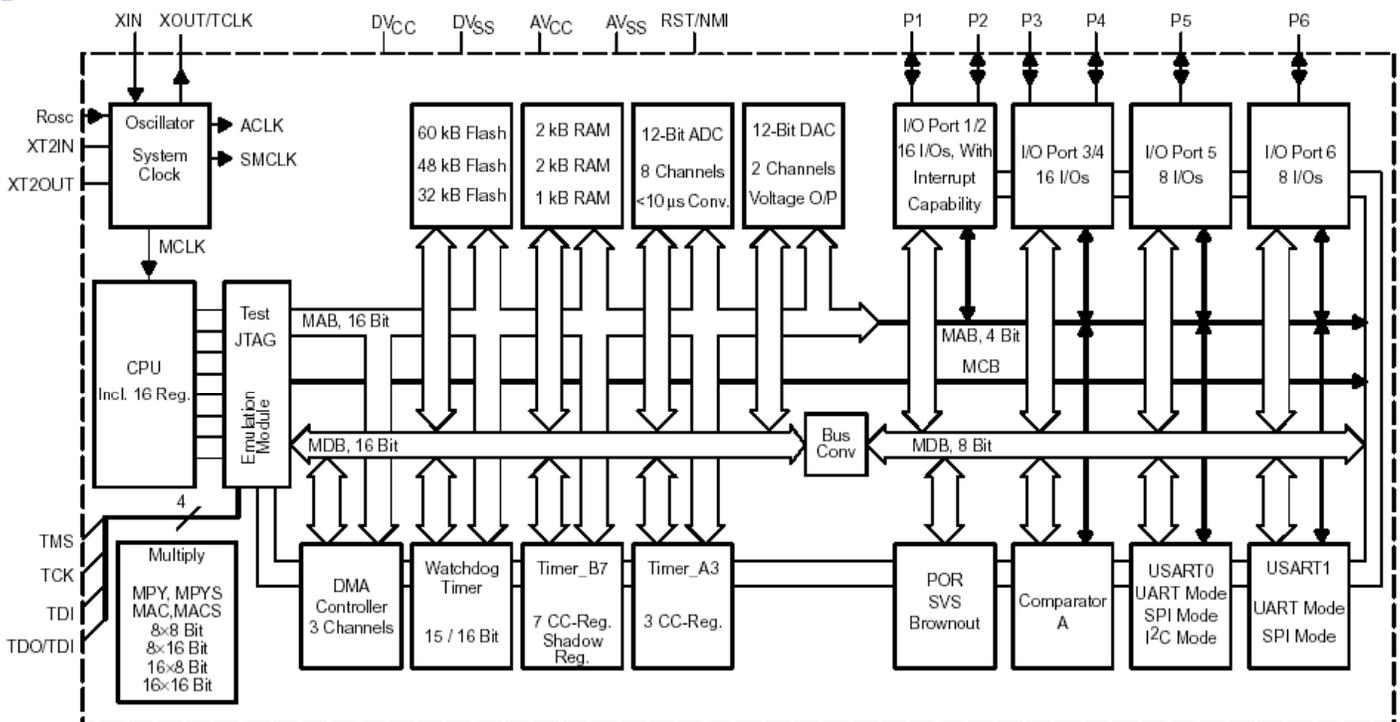
Per Personal
Computer

Per sistemi
embedded

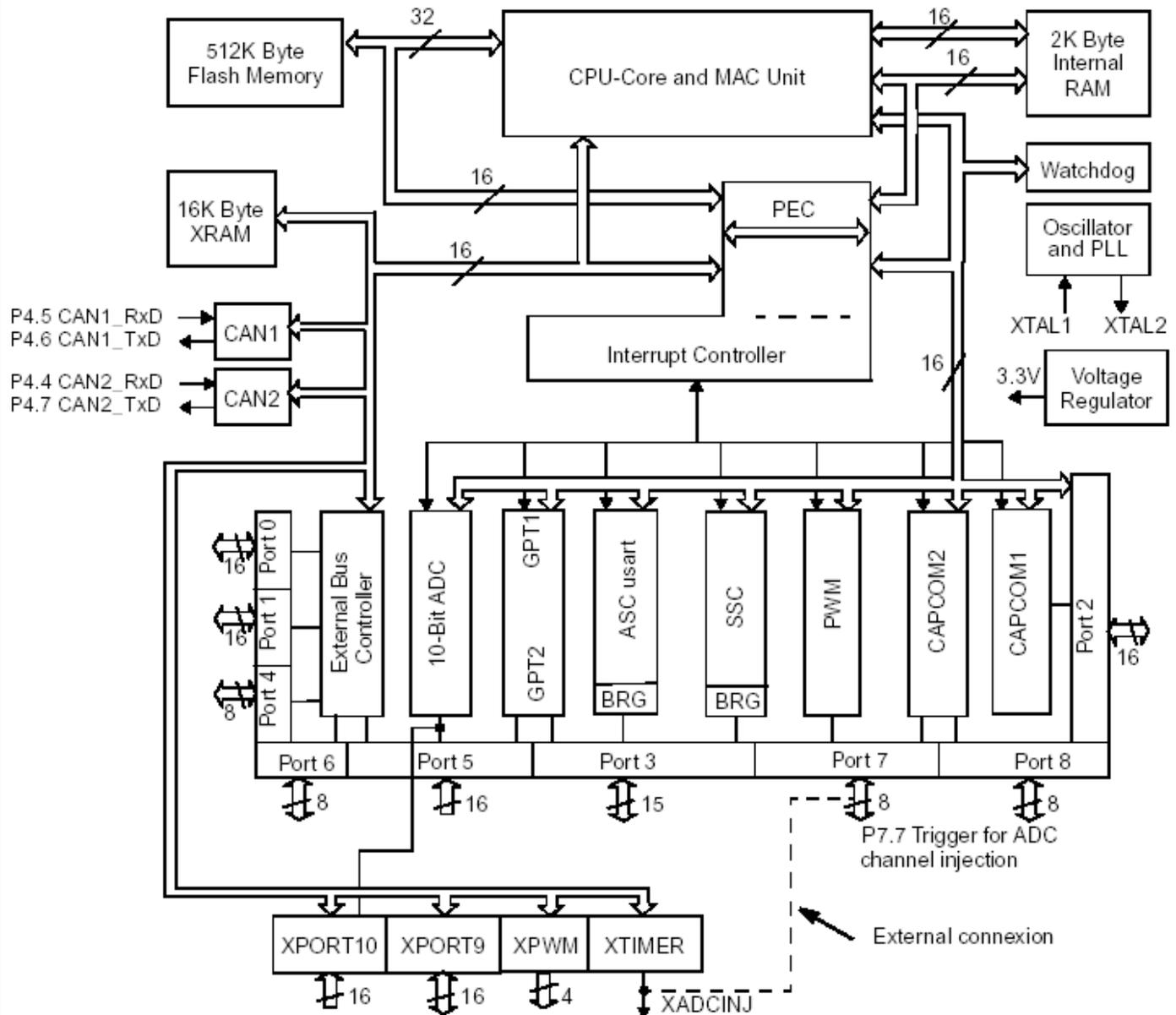
- ▲ Instruction Set standardizzati (tipo Intel IA-32)
- ▲ Chip set complessi
- ▲ Memory Management Unit (MMU)
- ▲ Unità floating point, MMX o simili (FPU)
- ▲ costi 75-800\$ e più

- ▲ Microcontrollori: integraz. di periferiche
- ▲ DSP: elevate prestazioni nell'elaborazione dei segnali
- ▲ non sempre MMU e FPU
- ▲ Media processors
- ▲ costi 5-50\$

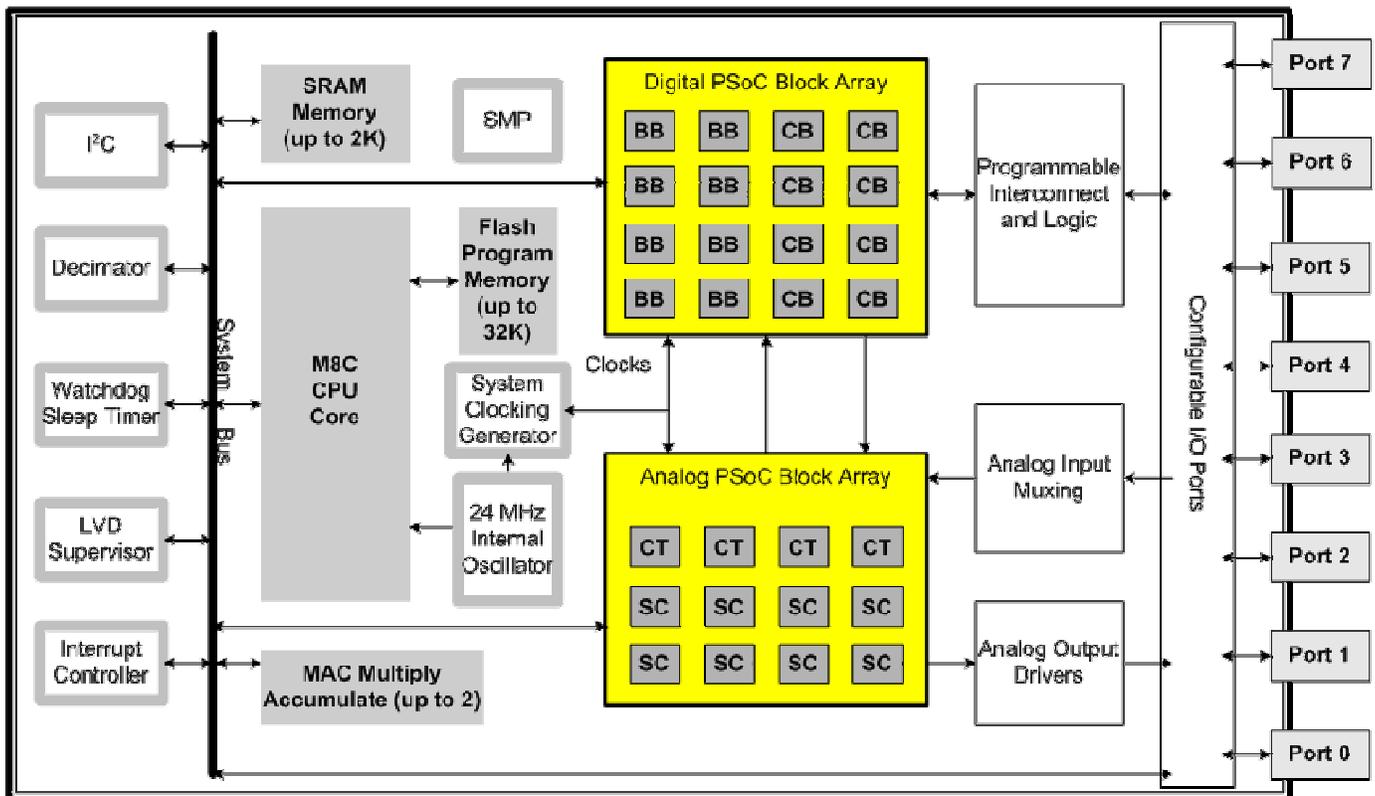
Esempi



Esempi

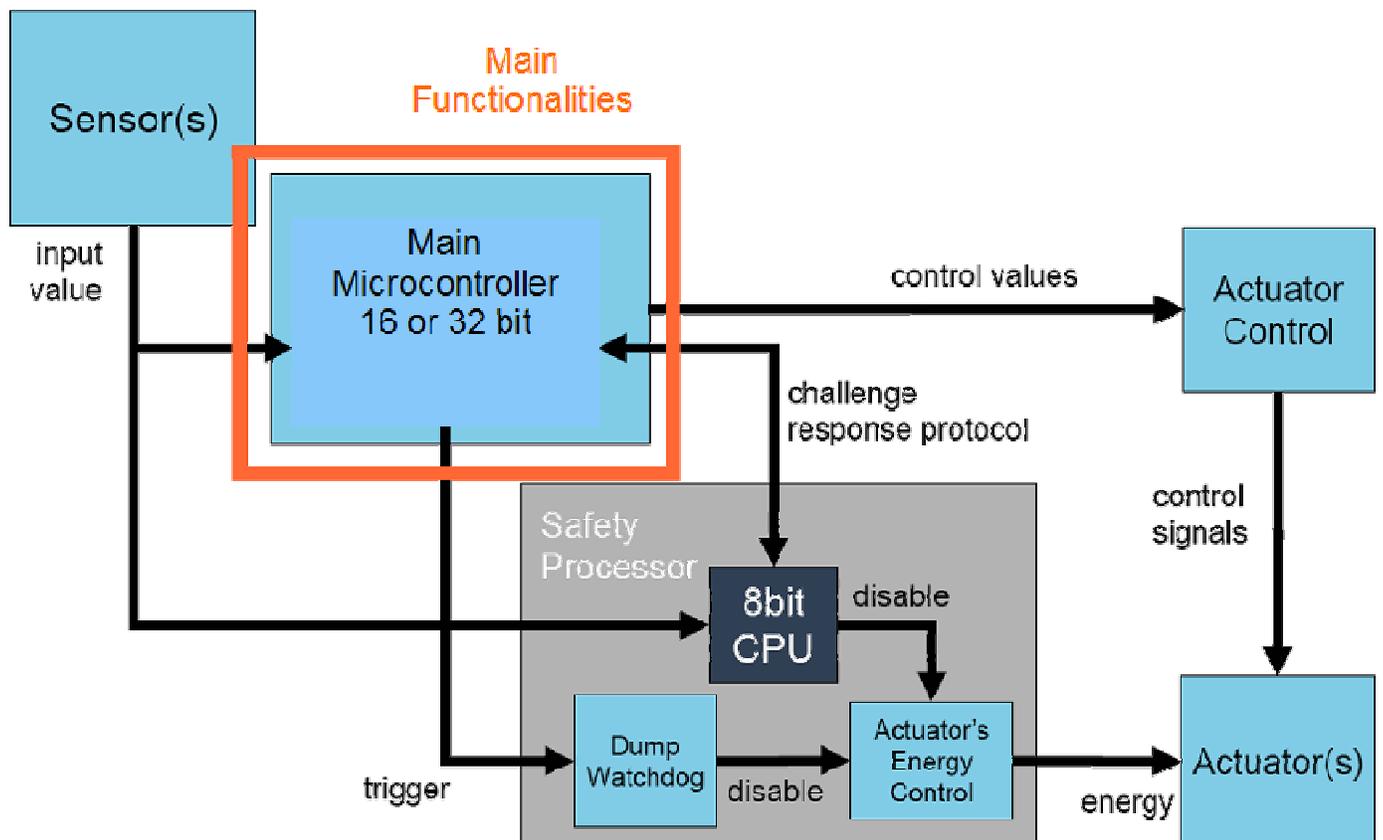


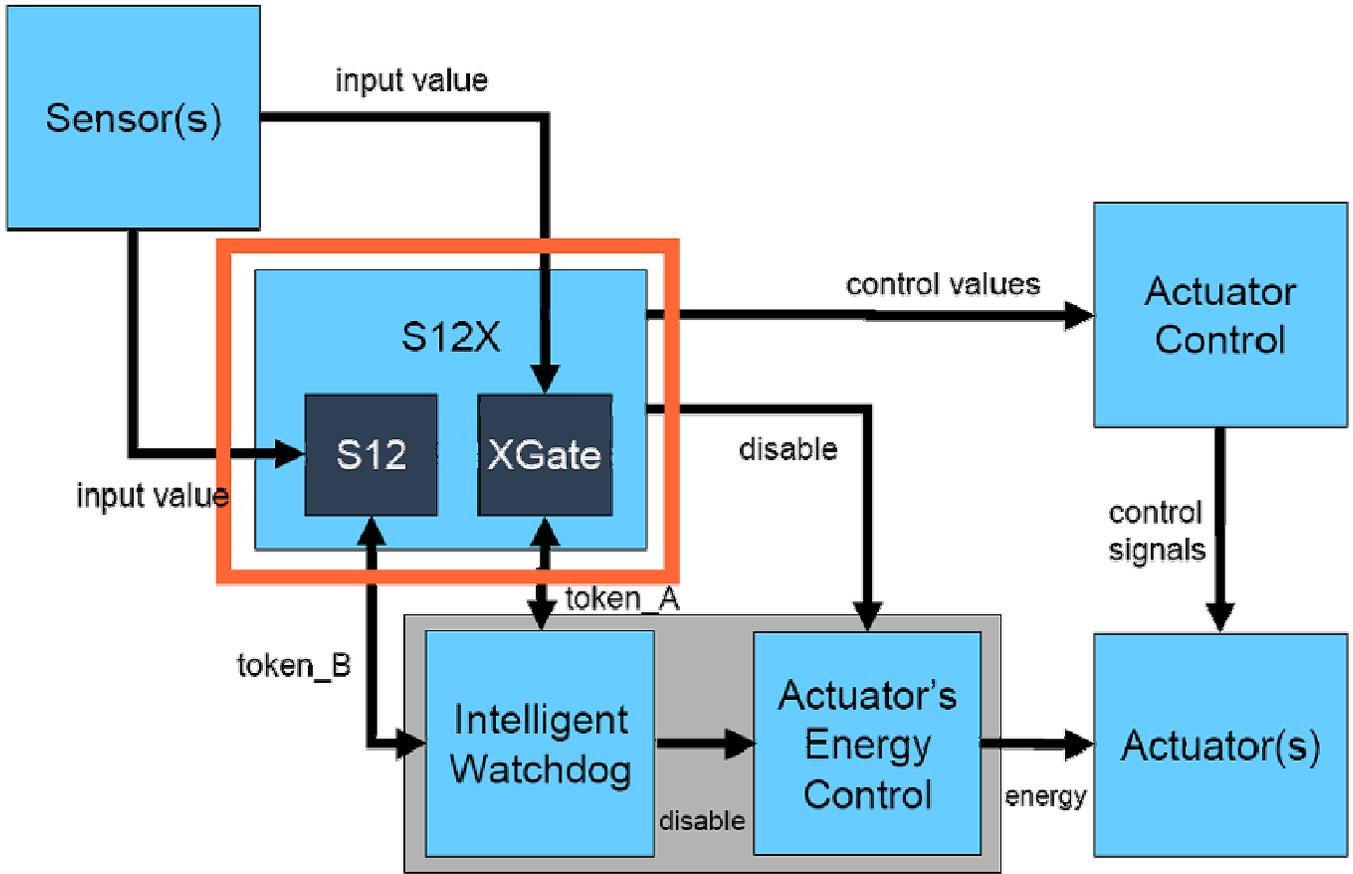
Esempi: PSoC



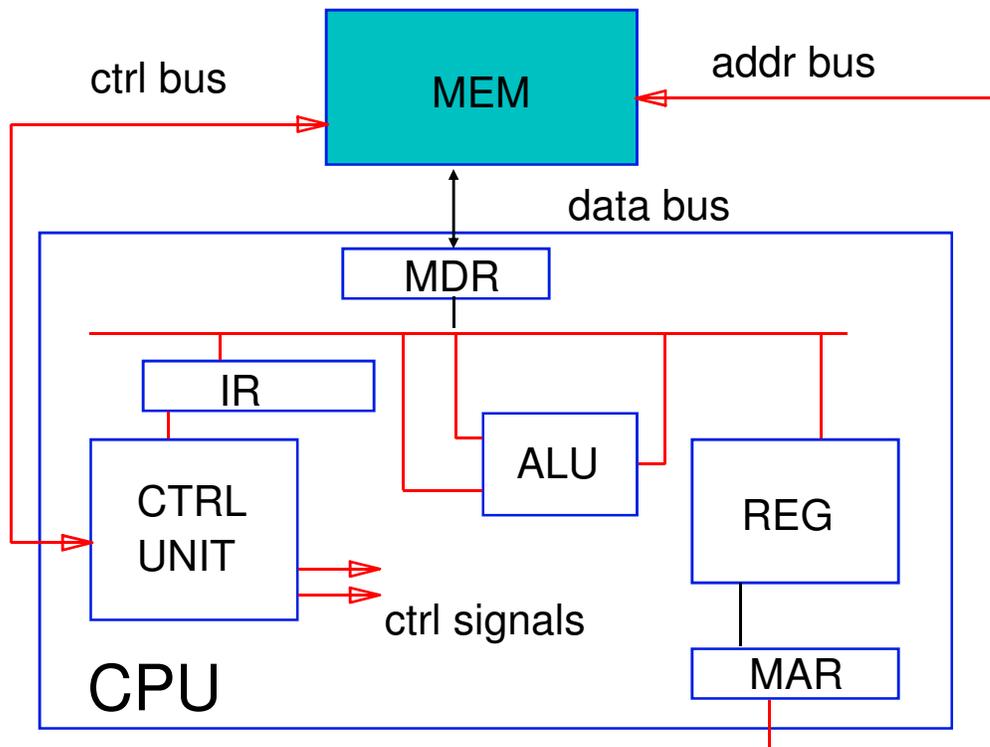
System On Chip

Esempi Sistemi Multicore



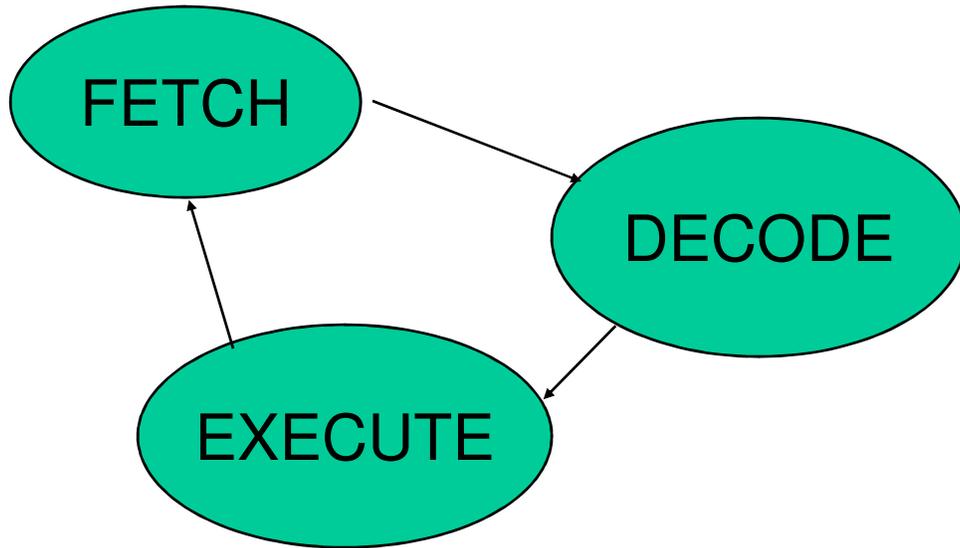


Architettura di semplice CPU



- 1) Control Unit: genera i segnali di controllo e di sincronizzazione
- 2) Data Path: ALU, registri, MDR...

FETCH ED EXECUTION



MAR \leftarrow PC
MDR \leftarrow M[MAR]
IR \leftarrow MDR fetch istr.

PC \leftarrow PC+n
<Decode opcode> decode

MAR \leftarrow IR(Oper_addr)
MDR \leftarrow M[MAR]
Reg \leftarrow MDR fetch oper.

Execute instr.

UNITA' DI CONTROLLO

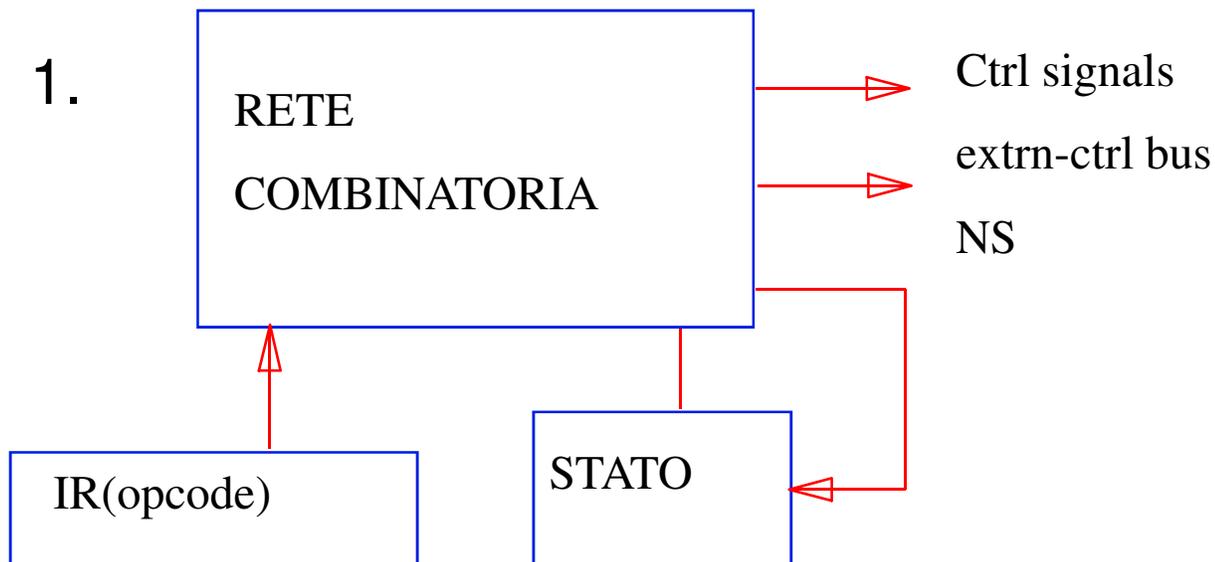
1. Controllo cablato o HARDWIRED
2. Controllo MICROPROGRAMMATO

La generazione della corretta sequenza di segnali di controllo per l'esecuzione di una data istruzione avviene da parte della Control Unit.

La sequenza generata è funzione:

- dell'istruzione (opcode, operandi, modi d'indirizzamento)
- dei flag
- di segnali di stato (stato attuale del sistema)

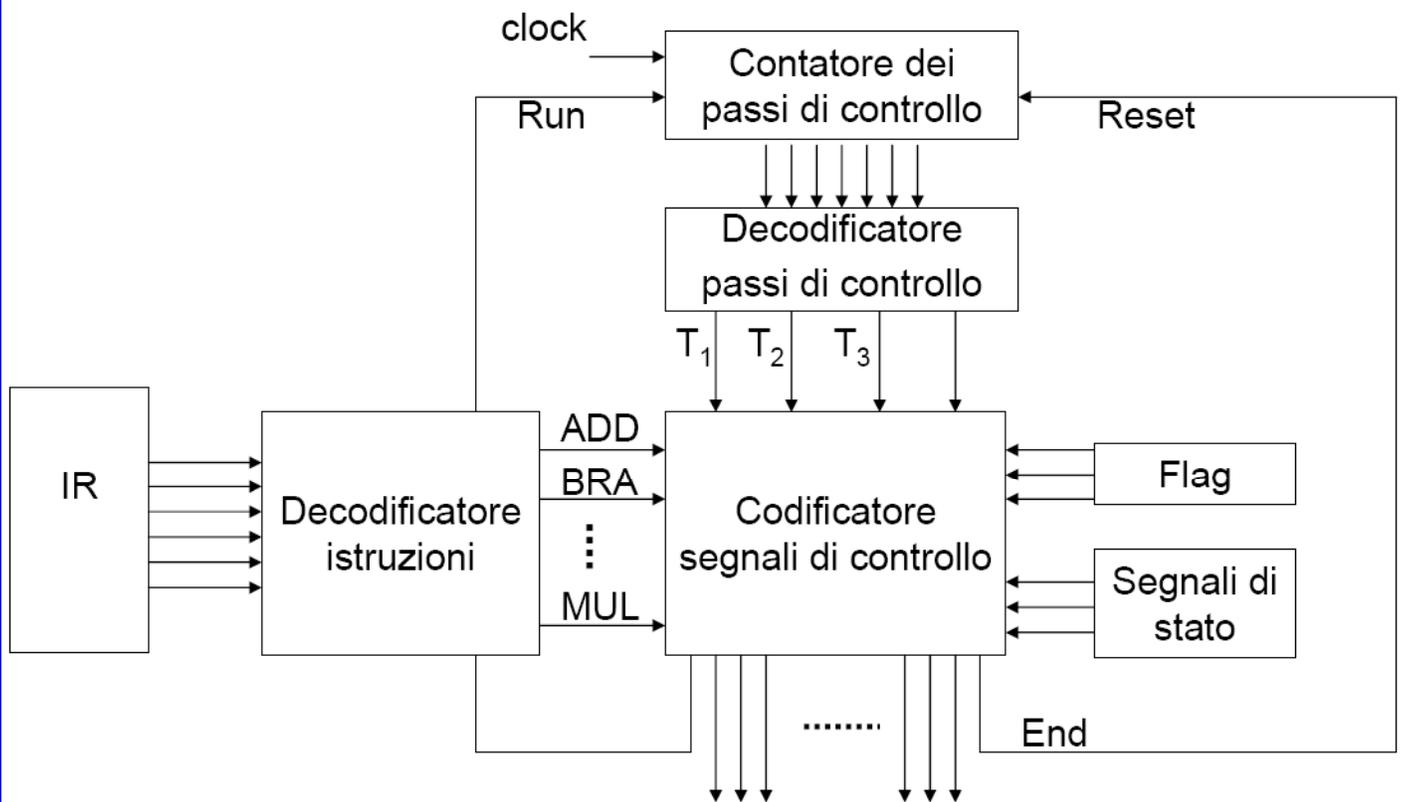
Controllo Hardwired o....



L'unità di controllo può essere realizzata con **una rete sequenziale** che ha:

- numero di stati coincidenti con i passi da realizzare per eseguire l'istruzione
- uscite coincidenti con i segnali di controllo e funzioni dello stato corrente, dell'istruzione, dei flag e di segnali di stato

....Controllo cablato



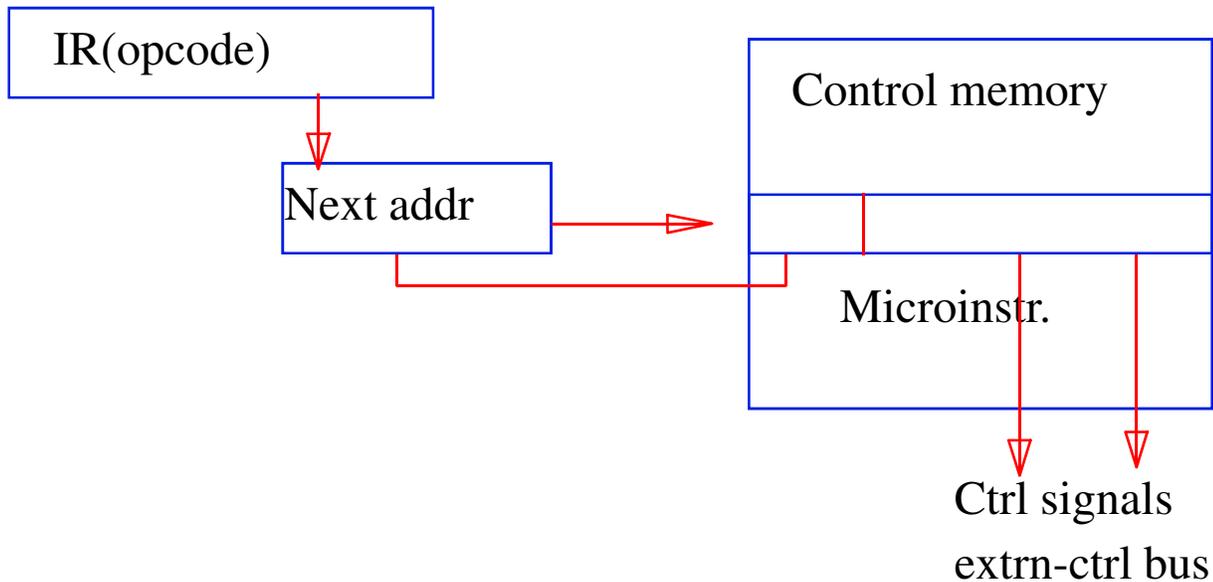
$$Z_{in} = T_1 + T_2 * ADD + T_5 * BRA + \dots$$

$$PC_{new} = T_2 + T_6 * N * BRA + PC * !BRA + \dots$$

$$End = T_7 * ADD + (T_4 * !N + T_6 * N) * BRA + \dots$$

Microprogrammato

2.



Gli insiemi dei segnali di controllo che vengono attivati in ogni passo di Controllo della macchina a stati del microprocessore possono essere organizzati in una struttura formata da una parola di controllo (control word), i cui singoli bit forniscono i valori dei segnali di controllo da generare per far progredire il sistema attraverso gli stati stessi.

Esempio

7	6	5	4	3	2	1	
0	0	0	0	0	1	0	PCin
0	0	0	0	0	0	1	Pcout
0	0	0	1	0	0	1	MARin
0	0	0	1	0	0	1	Read
0	0	0	0	0	0	0	MDRin
0	1	0	0	1	0	0	MDRout
0	0	0	0	1	0	0	IRin
0	0	1	0	0	0	0	Yin
0	0	0	0	0	0	1	Clear Y
0	0	0	0	0	0	1	Carry in
0	1	0	0	0	0	1	Add
0	1	0	0	0	0	1	Zin
1	0	0	0	0	1	0	Zout
1	0	0	0	0	0	0	R1in
0	0	1	0	0	0	0	R1out
0	0	0	0	0	0	0	R3in
0	0	0	1	0	0	0	R3out
0	0	1	0	0	1	0	WMFC
1	0	0	0	0	0	0	End

Il processore DLX

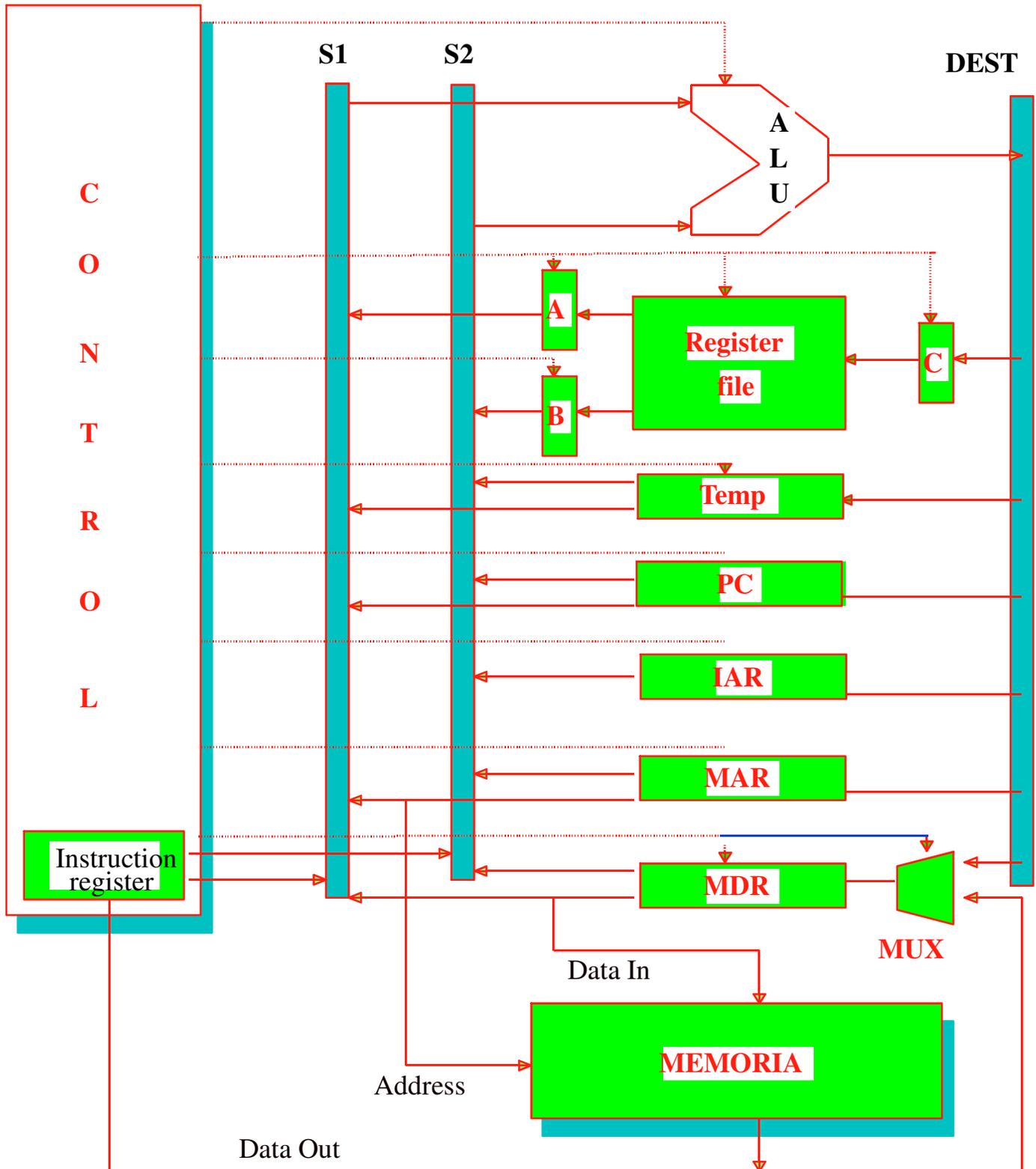
Esempio di architettura di CPU RISC:
il **DLX** (1990)

(unione di AMD 29K, DEC 3100, IBM 801,
Intel i860, MIPS, Motorola 68k, Sun SPARC)

caratteristiche:

- macchina LOAD/STORE
- General Purpose REGISTER FILE
composto da 32 registri
- INSTRUCTION SET semplice e “ridotto”
(RISC)
- istruzioni allineate a 32 bit
- architettura **pipeline**
- niente risorse dedicate allo STACK solo un
registro ACCUMULATORE (temporaneo)

Architettura DLX



Formato Istruzioni DLX

Format	Bits					
	31 - 26	25 - 21	20 - 16	15 - 11	10 - 6	5 - 0
R-type	0x0	Rs1	Rs2	Rd	<i>unused</i>	<i>opcode</i>
I-type	<i>opcode</i>	Rs1	Rd	<i>immediate</i>		
J-type	<i>opcode</i>	<i>value</i>				

DLX: ciclo di istruzione

DLX ciclo di istruzione

1) INSTRUCTION FETCH (**IF**)

$IR \leftarrow M[PC]$; $NPC \leftarrow PC + 4$

2) INSTRUCTION DECODE /REGISTER FETCH (**ID**)

$A \leftarrow R[IR6..10]$; $B \leftarrow R[IR11..15]$; $Imm \leftarrow R[IR16..31]$

3) EXECUTION/EFFECTIVE ADDRESS (**EX**)

- $ALUout \leftarrow A + immediate$; *Mem. ref.*
- $ALUout \leftarrow A \text{ op } (B \text{ or } immediate)$ *ALU op.*
- $ALUout \leftarrow NPC + immediate$; $cond \leftarrow (A \text{ op } 0)$ *branch-jmp*

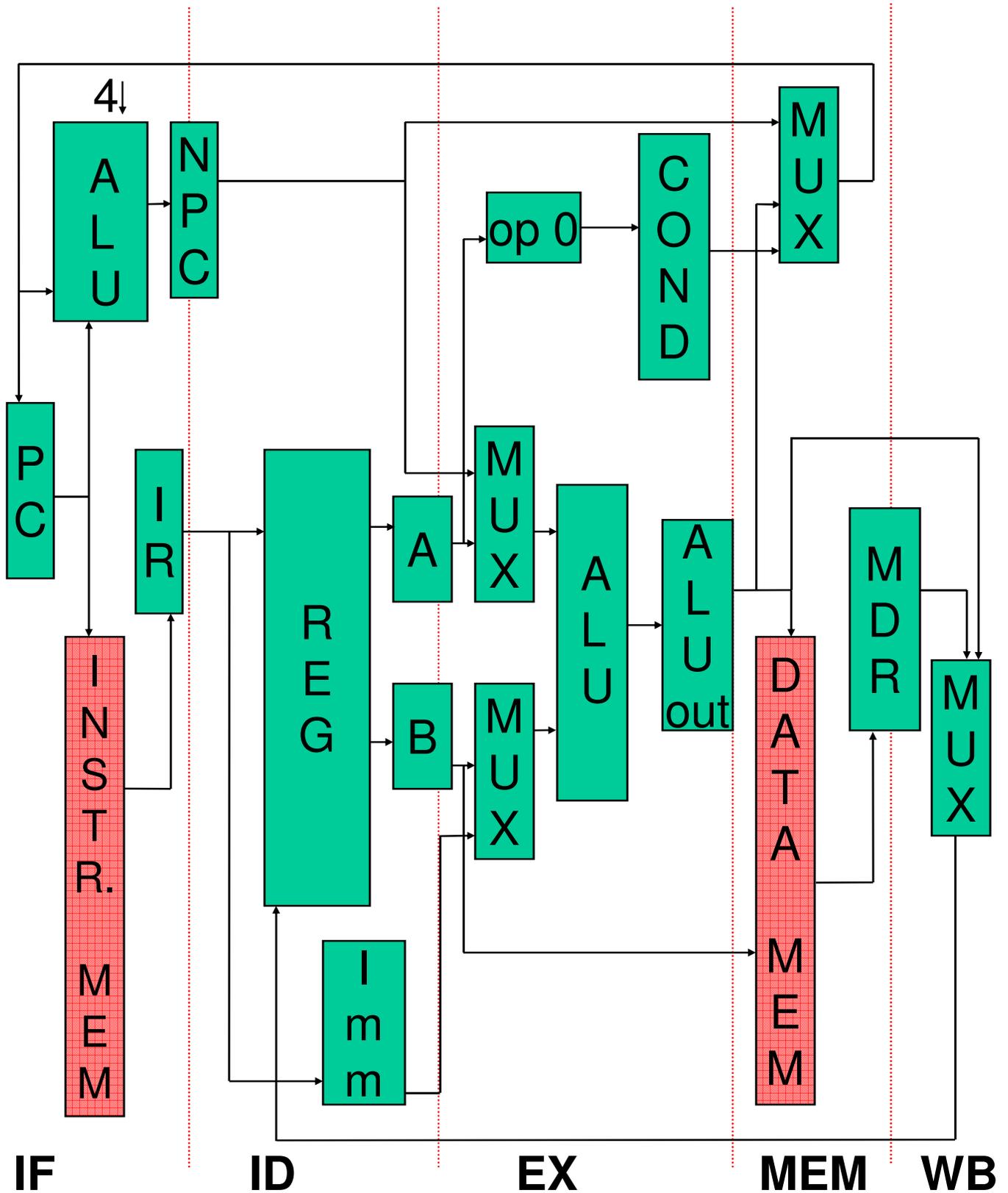
4) MEMORY ACCESS/ BRANCH END (**MEM**)

$MDR \leftarrow M[ALUout]$ or $M[ALUout] \leftarrow B$ *Mem ref.*
if (cond) $PC \leftarrow ALUout$ else $PC \leftarrow NPC$ *branch*

5) WRITE-BACK (**WB**)

$R[IR16..20] \leftarrow ALUout$ *Alu op.*
 $R[IR11..15] \leftarrow MDR$ *Load Mem ref.*

DLX DATAPATH



DLX DATAPATH

- Ciascuna fase è eseguita in un ciclo di clock (vincolo incrociato sul numero di livelli delle reti combinatorie della fase e la frequenza di clock)
- Osservazioni importanti sul data path:
 - ALUout è il registro con il risultato in uscita dalla ALU
 - PC è sostituito da:
 - se COND è vero, dal valore di ALUout (salti condizionali verificati, cioè TRUE)
 - se COND è falso, dal valore di NPC (salti condizionali non verificati, cioè FALSE, si prosegue con l'esecuzione sequenziale)
 - l'accesso alla memoria si svolge nella fase MEM, che dura un clock se il dato è in cache; in caso contrario, la fase viene prolungata