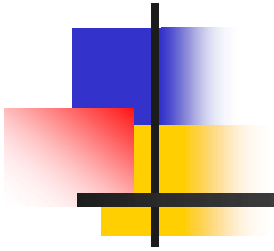


Calcolo Parallelo e Distribuito

Forme di Parallelismo – Classificazione di Flynn

Cenni sul calcolo distribuito

Docente: Prof. L. Marcellino

Tutor: Prof. P. De Luca

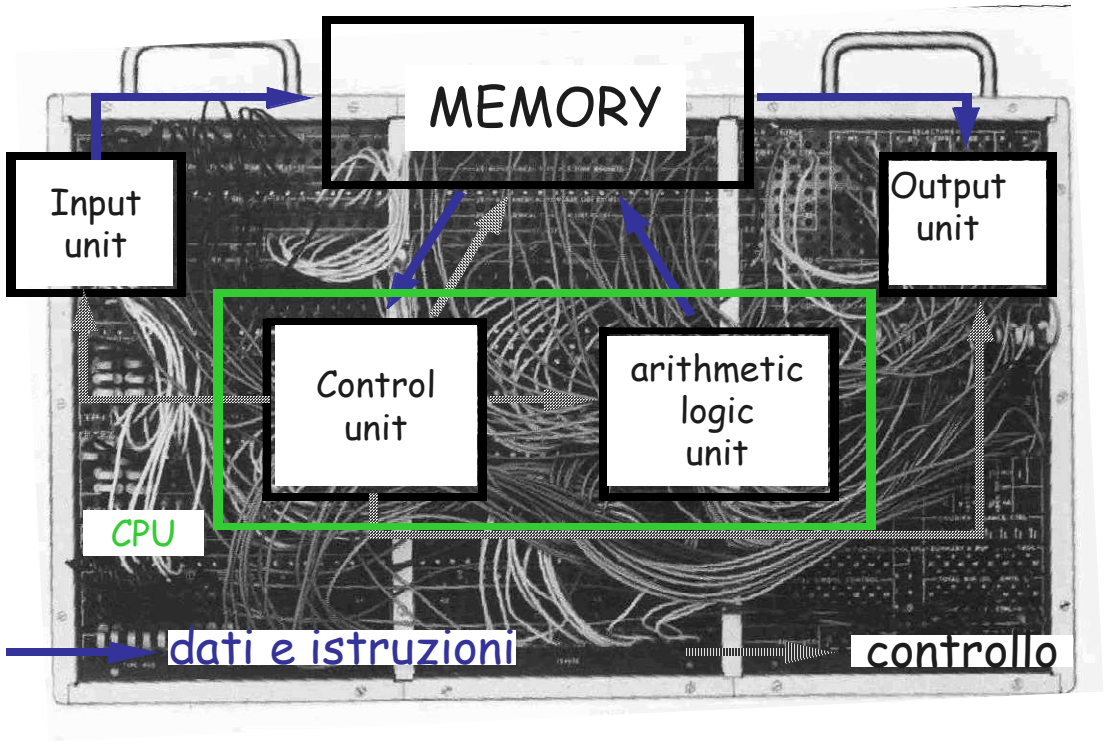
Calcolo Parallelo

Il **calcolo parallelo** è una evoluzione del calcolo seriale che tenta di emulare ciò che spesso avviene nel mondo naturale: molteplici eventi complessi e inter-correlati che avvengono nello stesso tempo.

L'idea del **calcolo parallelo** si basa sull'impiego simultaneo di risorse di calcolo multiple per risolvere un unico problema, spezzandolo in parti discrete elaborabili contemporaneamente, ovvero che possono essere eseguite in modo seriale su **differenti CPU**.

Questa è la macchina che tutti conosciamo

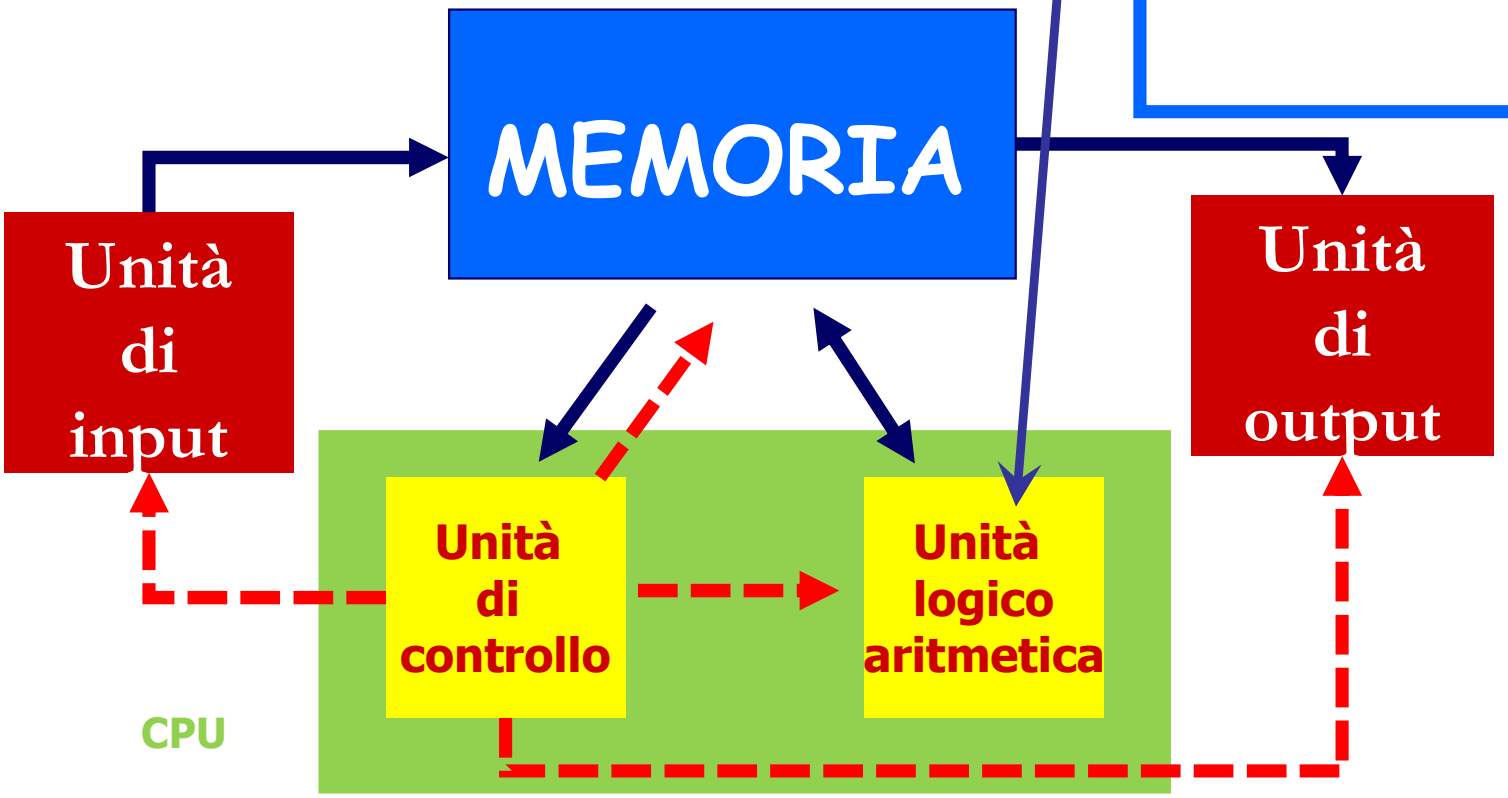
1945 John Von Neumann



parallelismo temporale

Macchina di Von Neumann

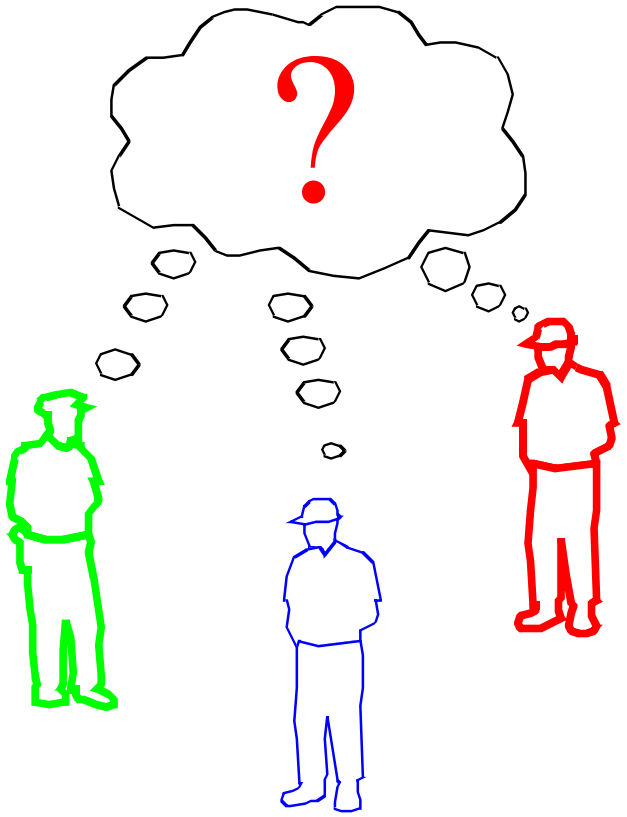
più unità funzionali all'interno di un'unica ALU



Dati e Istruzioni — controllo

Primo tipo di parallelismo *on-chip*

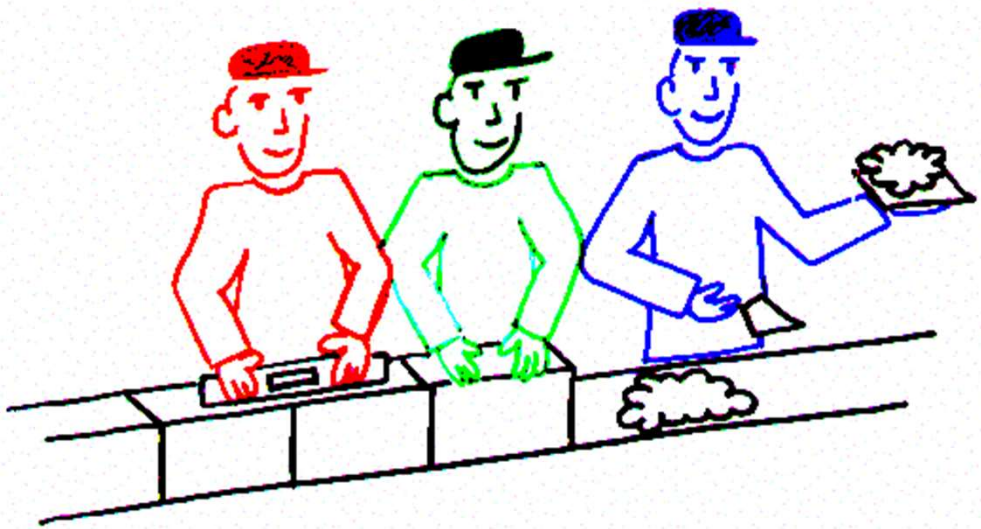
Esempio: costruzione di una casa



Primo tipo di parallelismo

Primo tipo di parallelismo

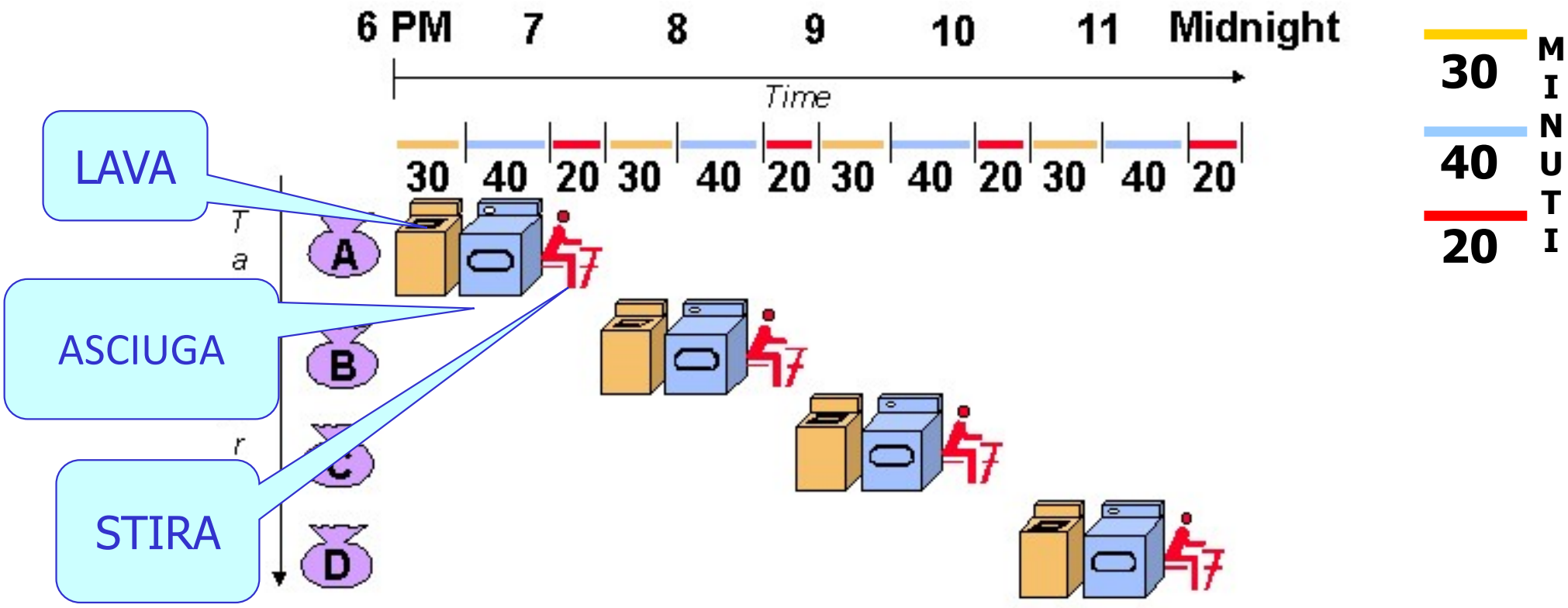
Tecnica della catena di montaggio (pipeline)



I tre operai eseguono contemporaneamente fasi successive dello stesso lavoro

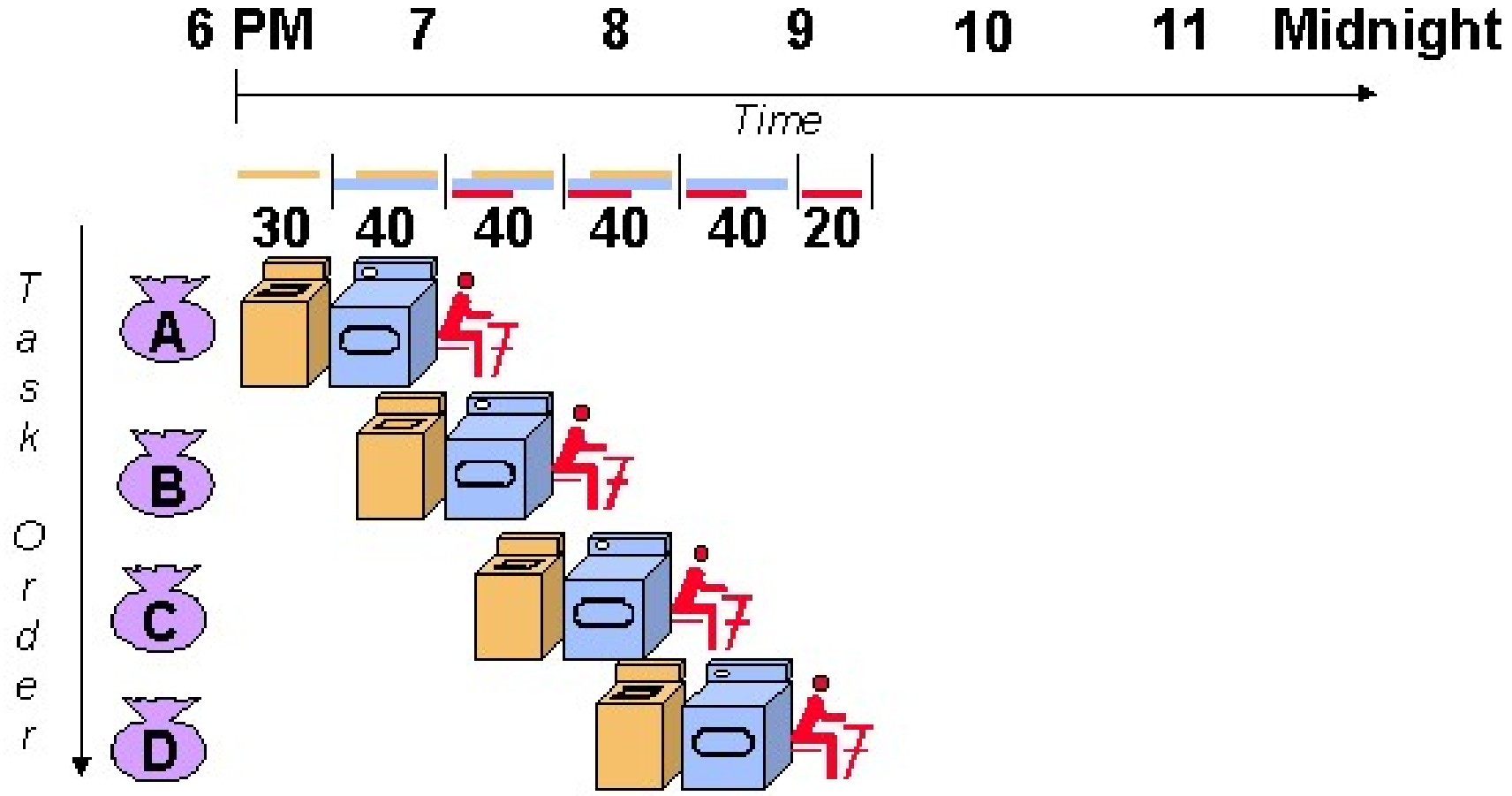
PARALLELISMO TEMPORALE

Esempio



Per completare il lavoro sono necessarie 6 ore!

Quanto tempo si risparmia?

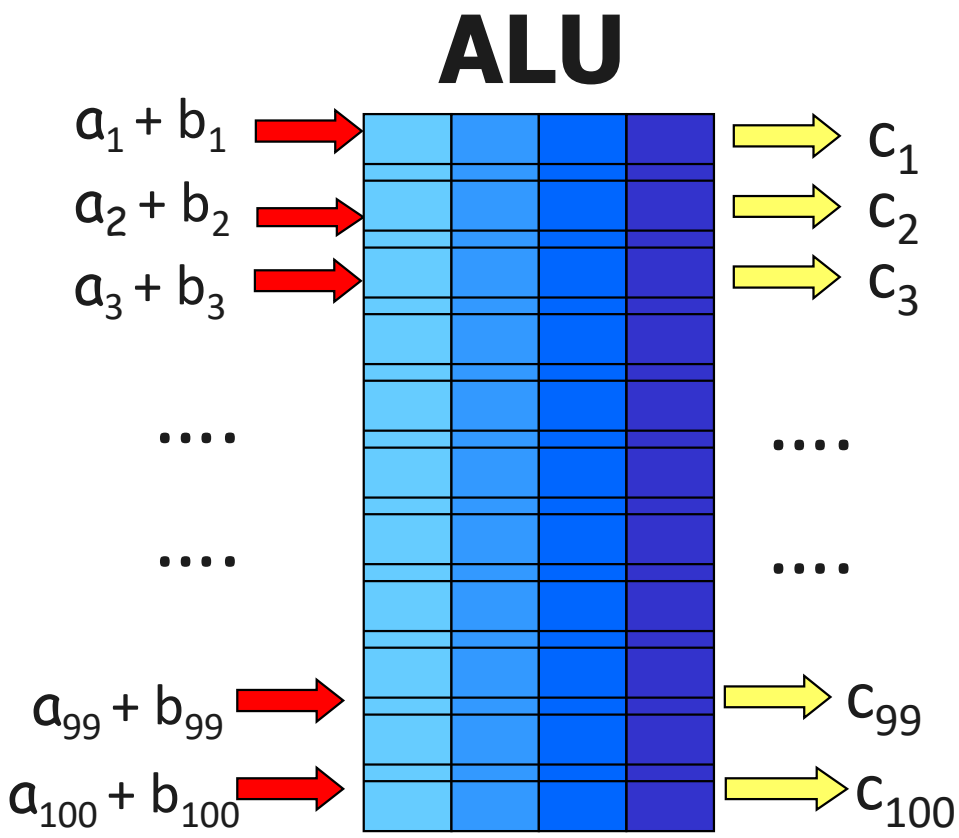


Con una pipeline bastano 3 ore !

Esempio: somma di 100 numeri floating point

$$c_i = a_i + b_i \qquad i=1,100$$

Unità tradizionale



Esempio: somma di 100 numeri floating point

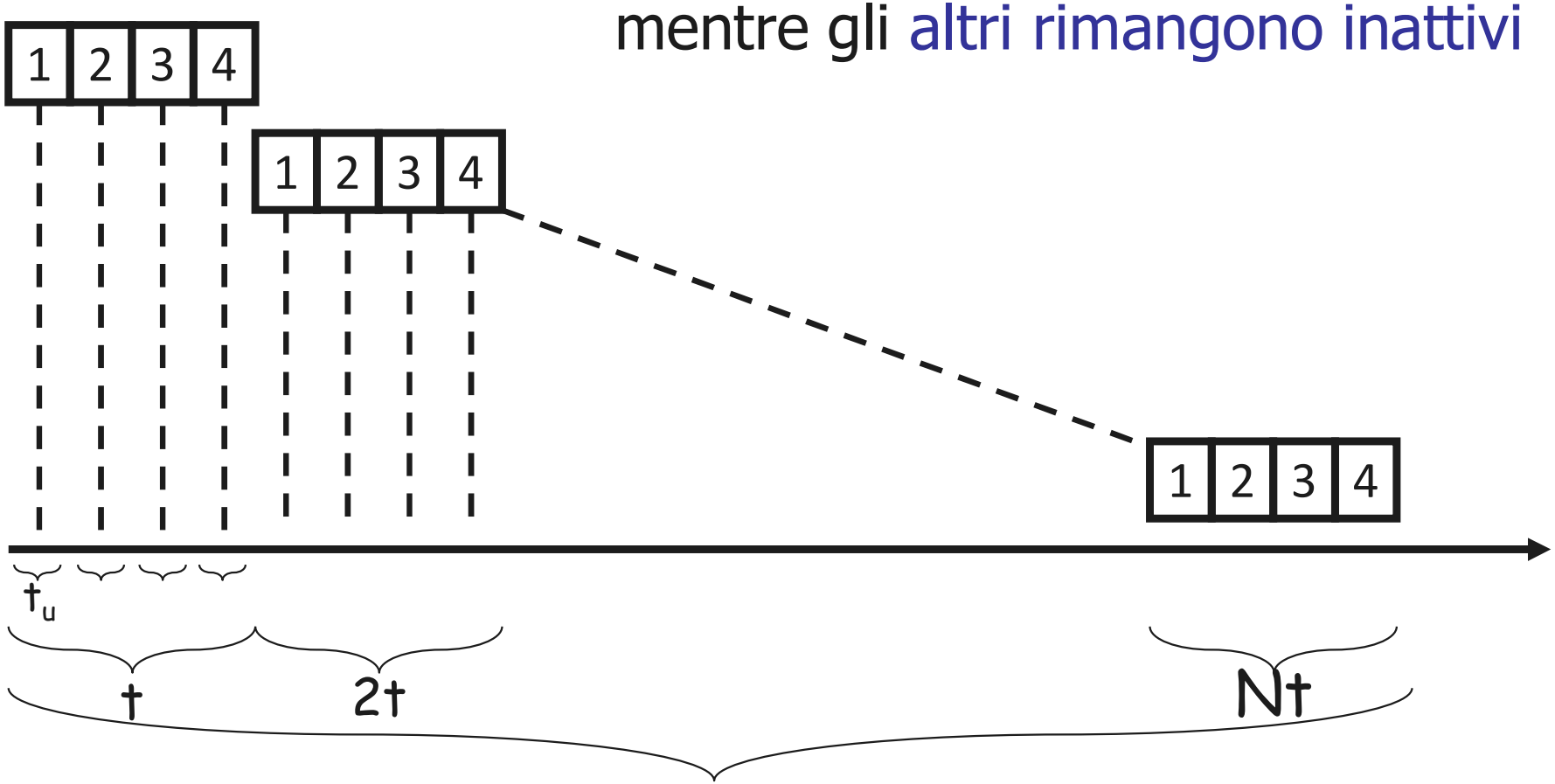
L'unità funzionale per l'addizione floating point è divisa in **4** segmenti

Ciascun segmento è preposto alla esecuzione di una fase dell'operazione, ad esempio
Per la somma:

CONFRONTO DEGLI ESPONENTI	FASE 1
SHIFT DELLA MANTISSA	FASE 2
SOMMA DELLE MANTISSE	FASE 3
NORMALIZZAZIONE	FASE 4

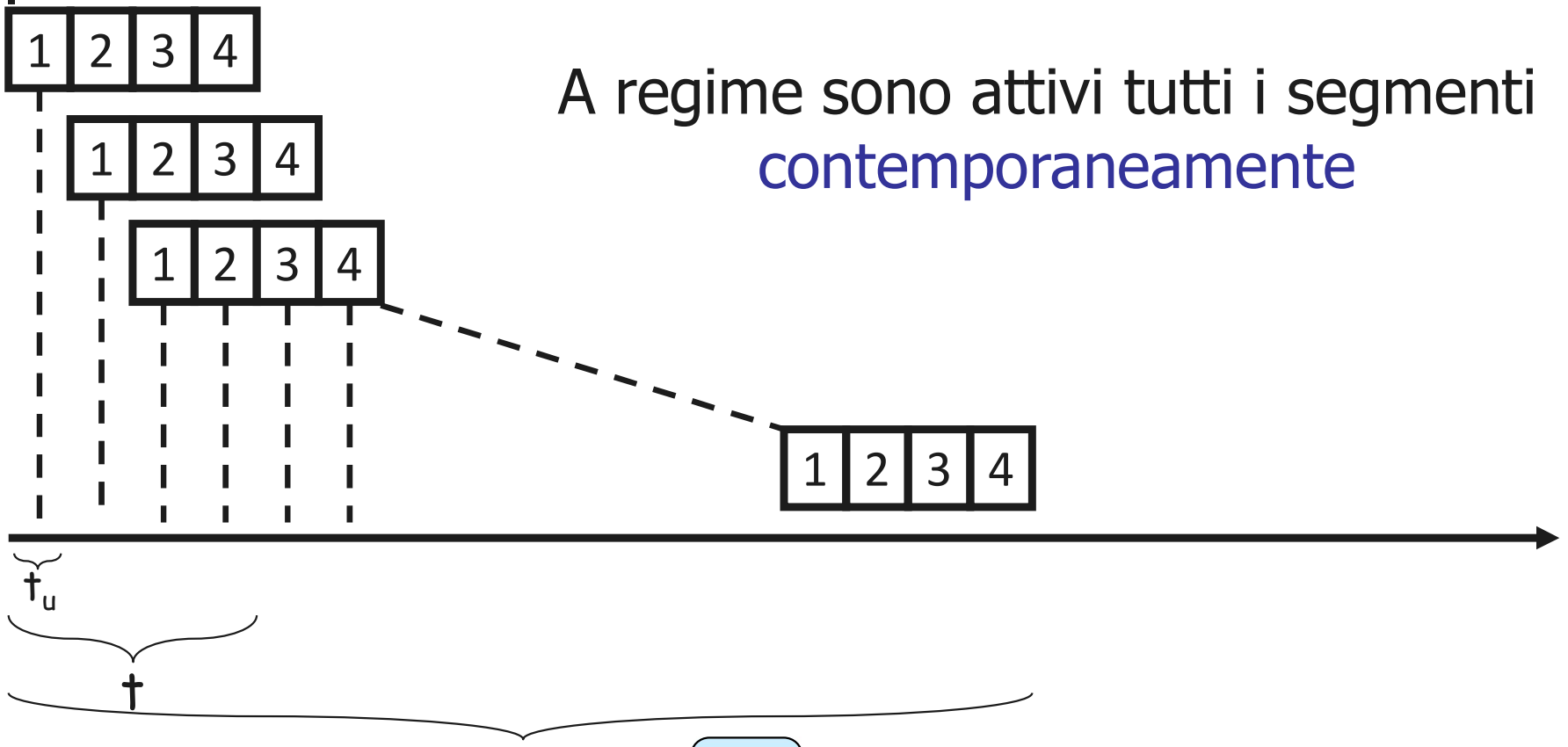
Esecuzione tradizionale di N somme f.p.

Solo un segmento alla volta è attivo
mentre gli altri rimangono inattivi



$$T = Nt = 4Nt_u$$

Esecuzione *pipelined* di N somme f.p.



A regime sono attivi tutti i segmenti contemporaneamente

$$T = [4+(N-1)]t_u < 4\ N\ t_u$$

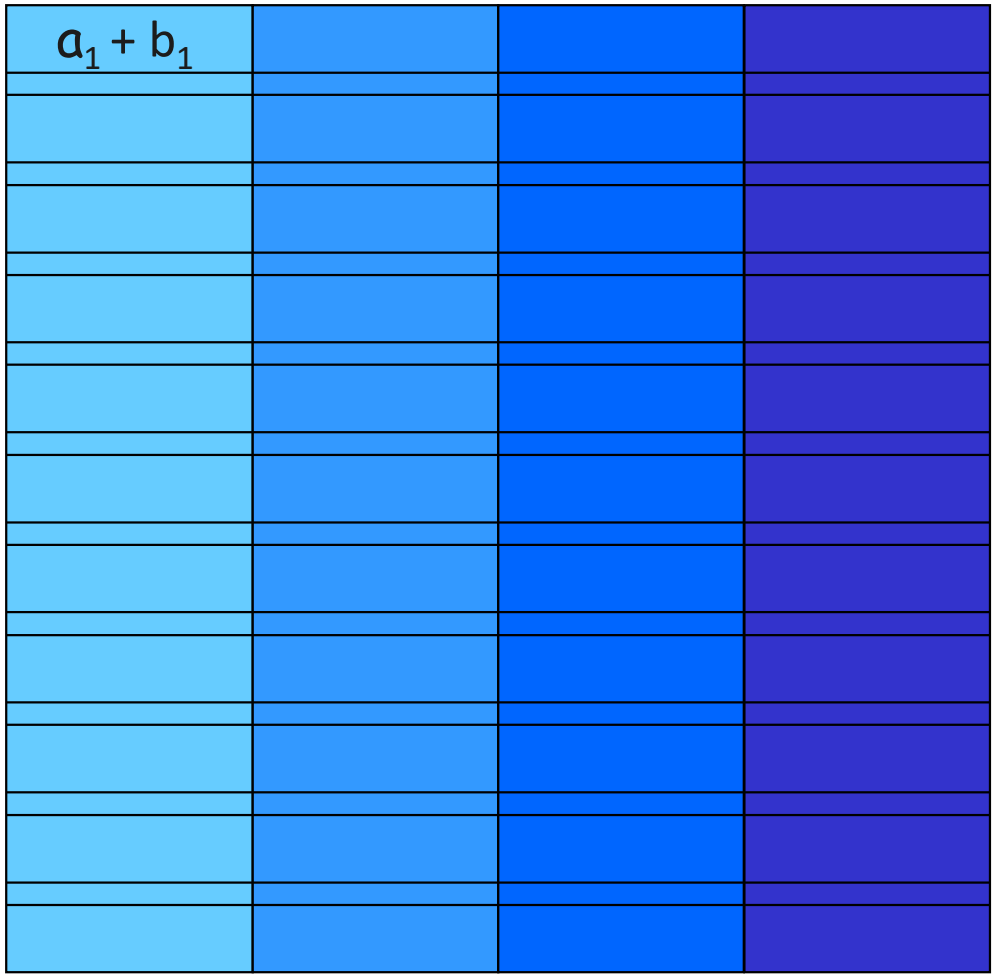
Numero di "segmenti" del pipe

Esempio: somma di 100 numeri floating point

$c_i = a_i + b_i \qquad i=1,100$

Unità pipelined

ALU



Hardware:

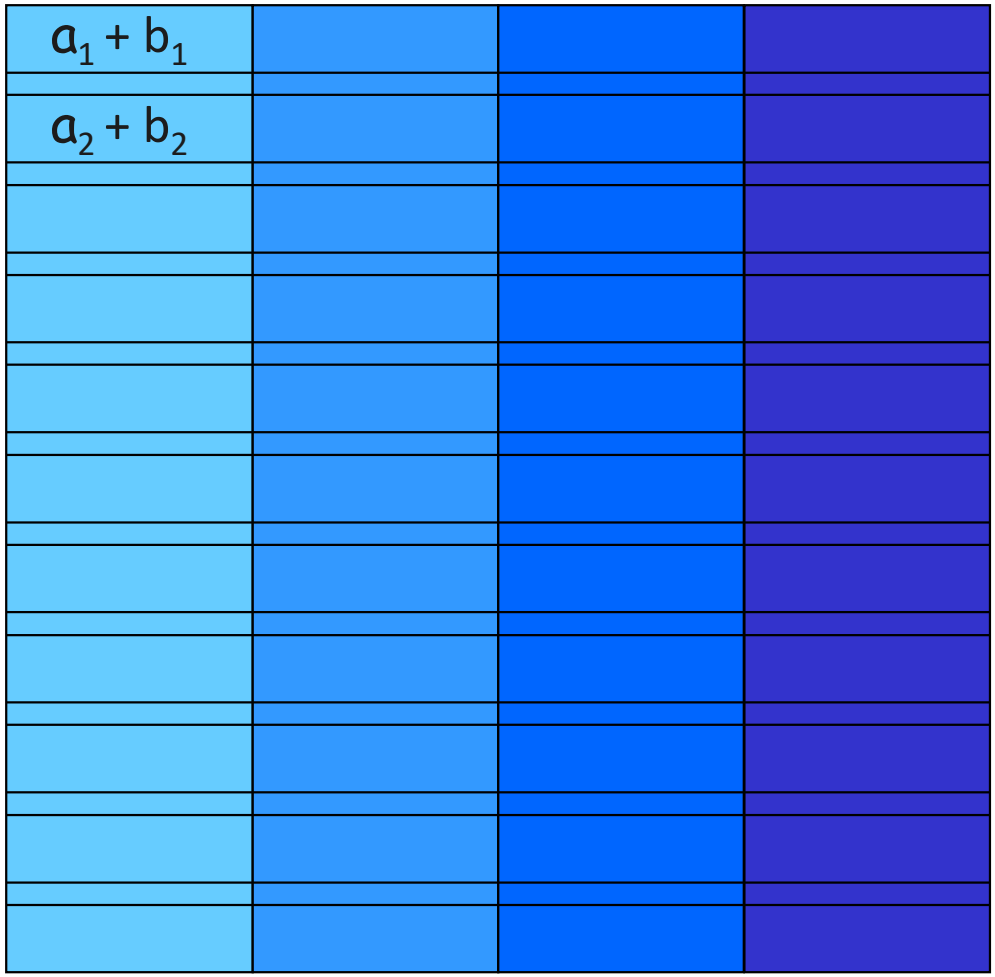
più unità funzionali all'interno di un'unica ALU

Esempio: somma di 100 numeri floating point

$c_i = a_i + b_i \qquad i=1,100$

Unità pipelined

ALU



Hardware:

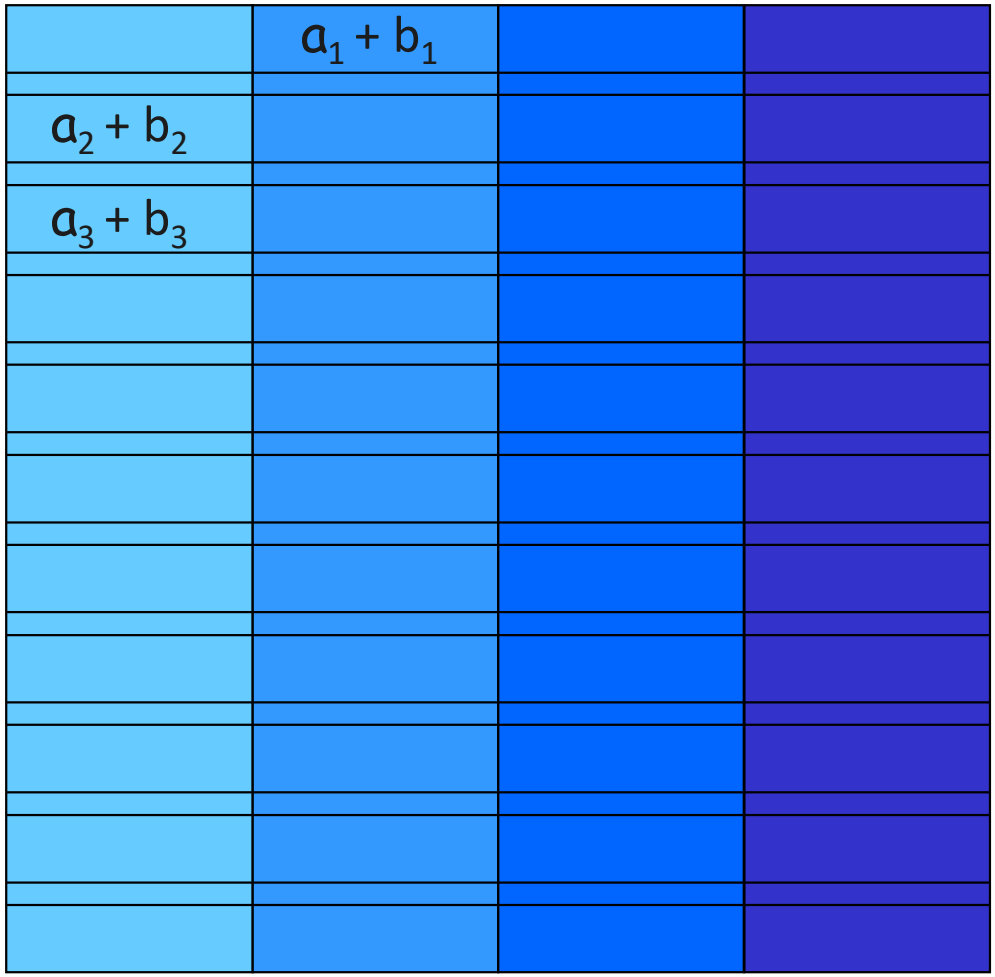
più unità funzionali all'interno di un'unica ALU

Esempio: somma di 100 numeri floating point

$c_i = a_i + b_i \qquad i=1,100$

Unità pipelined

ALU



Hardware:

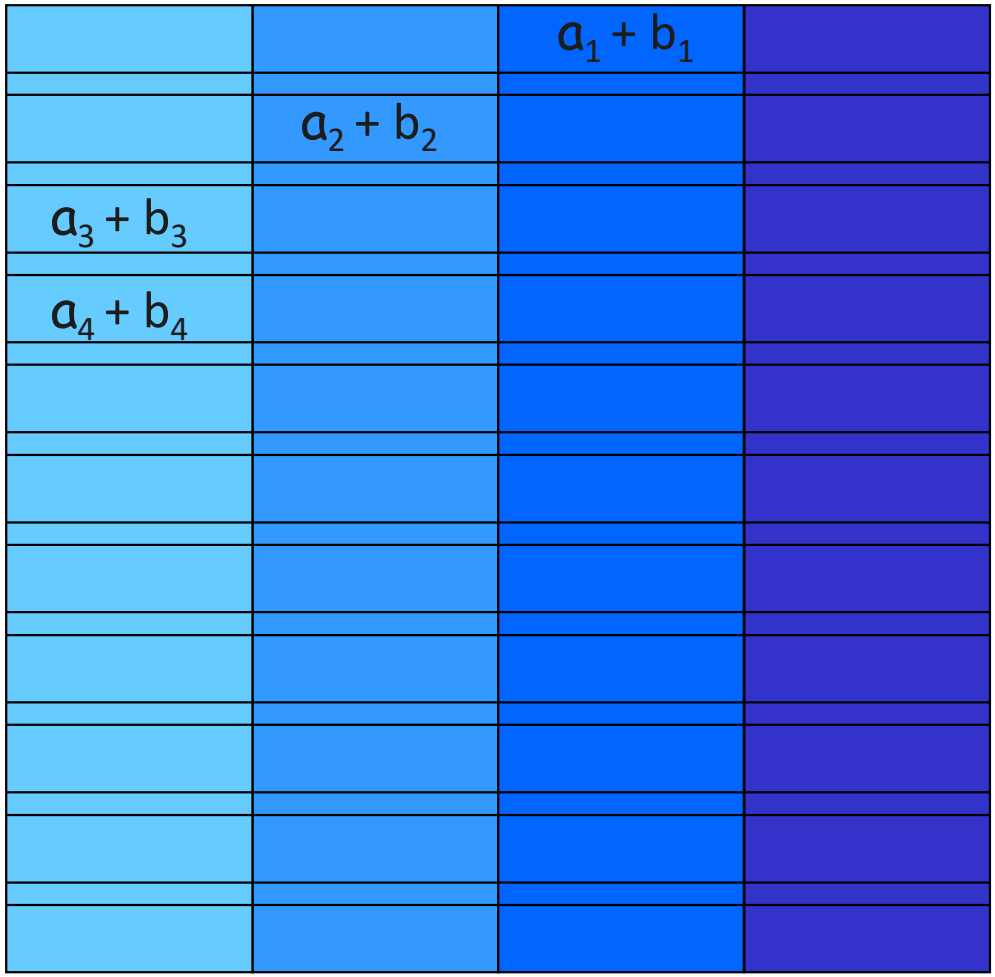
più unità funzionali all'interno di un'unica ALU

Esempio: somma di 100 numeri floating point

$c_i = a_i + b_i \qquad i=1,100$

Unità pipelined

ALU



Hardware:

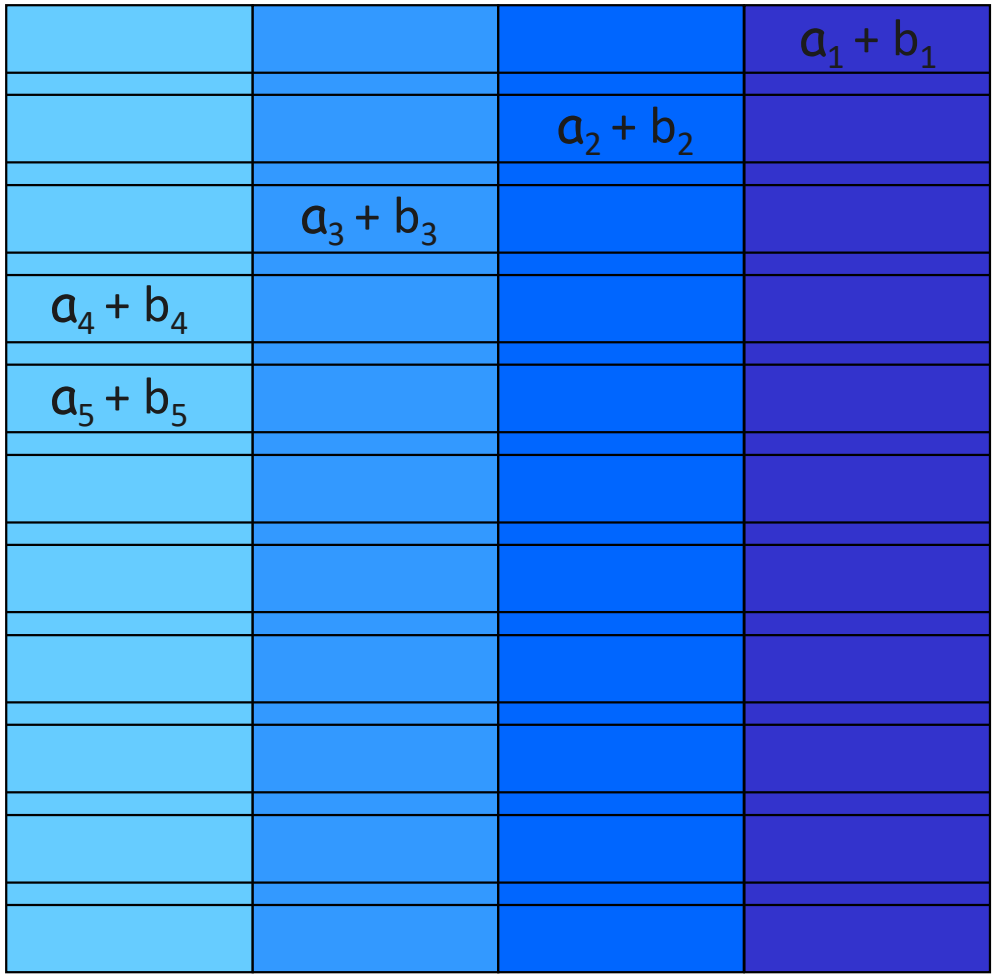
più unità funzionali all'interno di un'unica ALU

Esempio: somma di 100 numeri floating point

$c_i = a_i + b_i \qquad i=1,100$

Unità pipelined

ALU



Hardware:

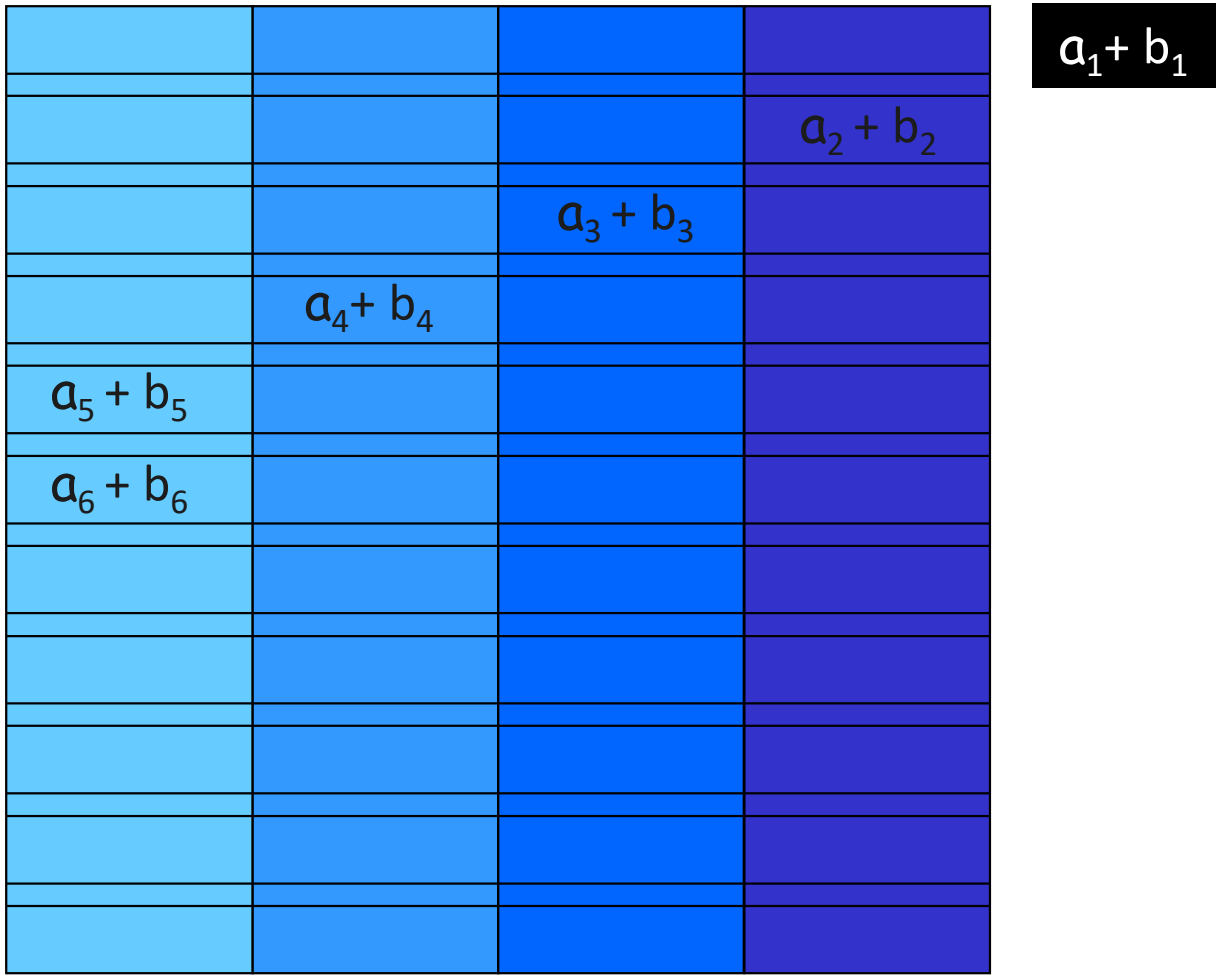
più unità funzionali all'interno di un'unica ALU

Esempio: somma di 100 numeri floating point

$c_i = a_i + b_i \qquad i=1,100$

Unità pipelined

ALU



Hardware:

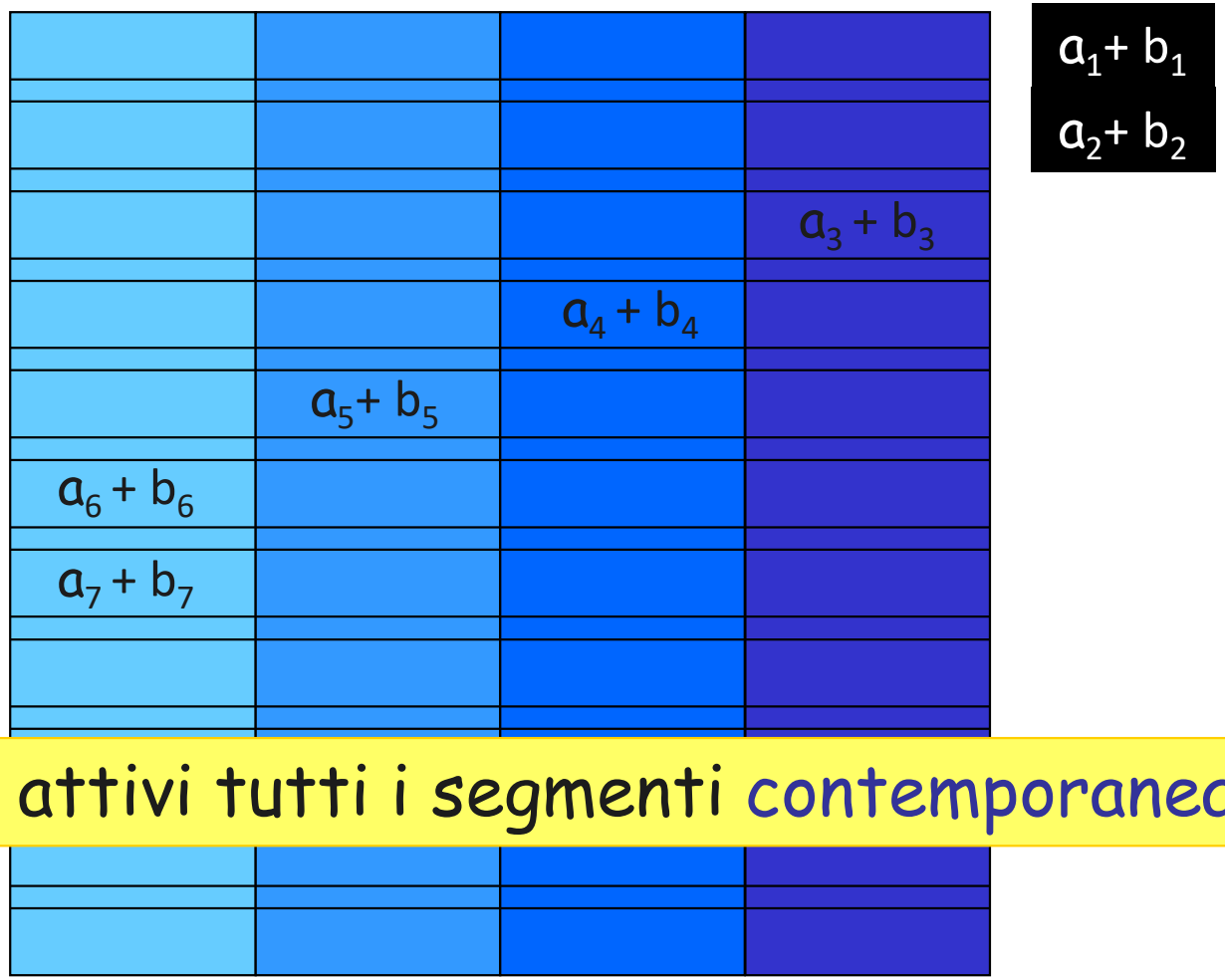
più unità funzionali all'interno di un'unica ALU

Esempio: somma di 100 numeri floating point

$c_i = a_i + b_i \qquad i=1,100$

Unità pipelined

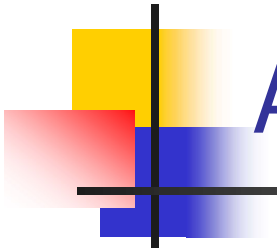
ALU



Hardware:

più unità funzionali all'interno di un'unica ALU

A regime sono attivi tutti i segmenti contemporaneamente



Architetture con unità funzionali *pipelined*

L'uso di unità funzionali pipelined è anche alla base dei

processori vettoriali

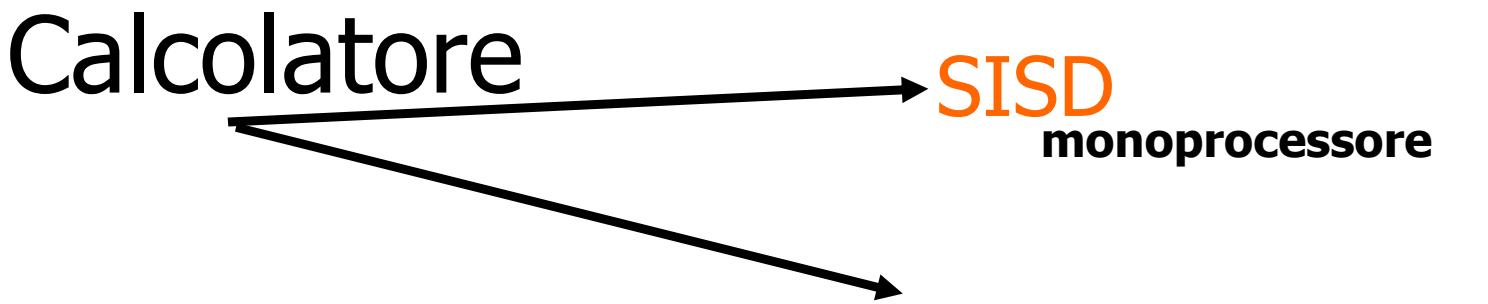
Capaci di

operare efficientemente su dati

strutturati sotto forma di vettori

Attualmente
tutti i microprocessori
utilizzano una struttura pipeline per migliorare le loro prestazioni.

Tassonomia di Flynn (dal 1965)

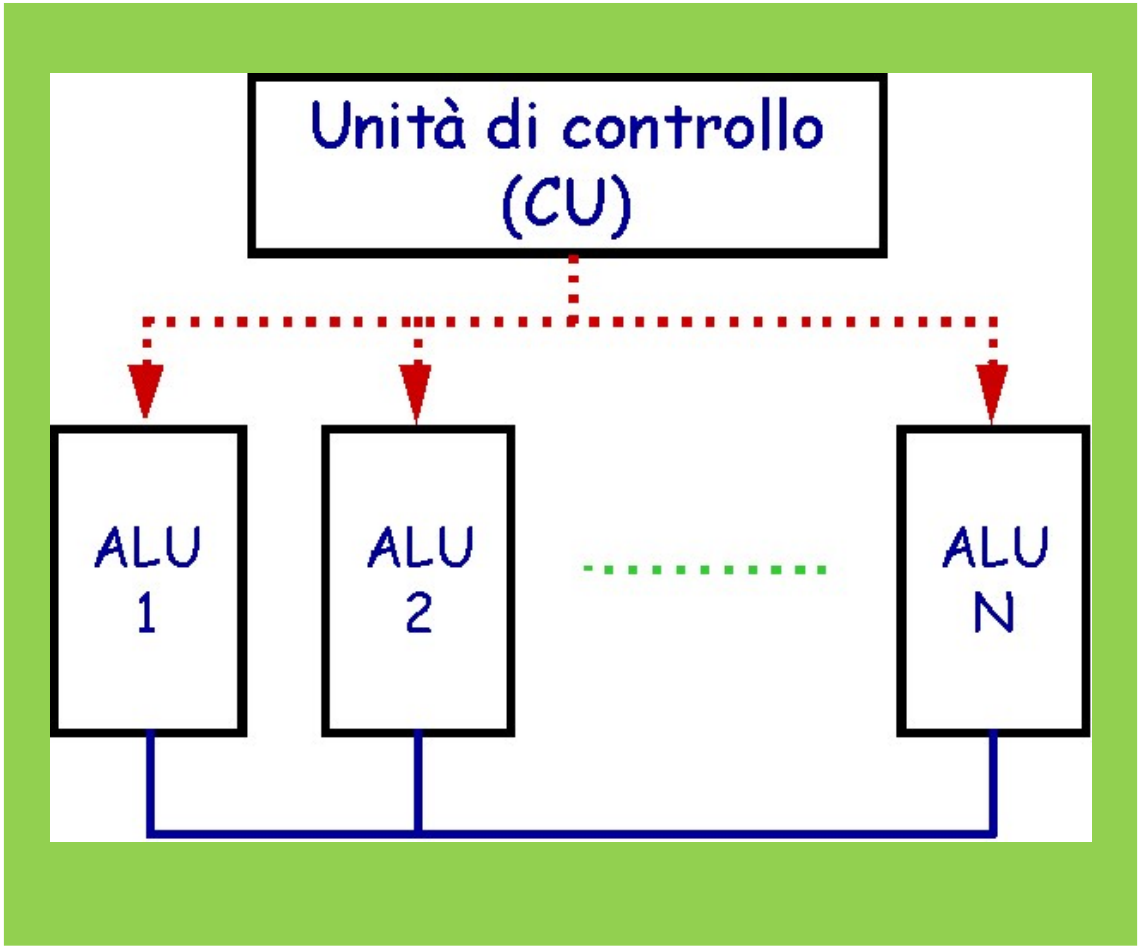


Calcolatori MISD

(Multiple Instruction Single Data)

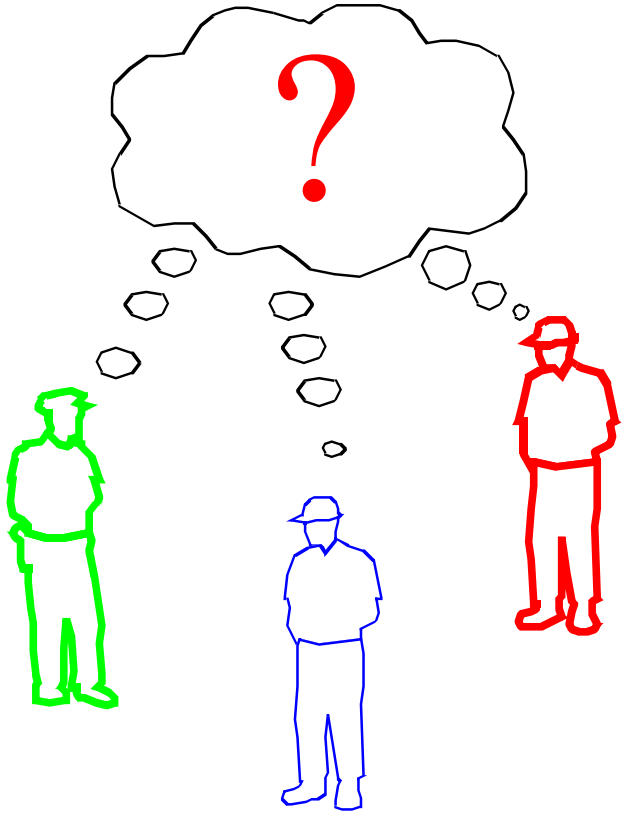
Parallelismo spaziale

Più unità aritmetico-logiche (ALU) operano sotto un comune controllo (CU) eseguendo in parallelo la ***stessa istruzione*** su ***dati diversi***



Dentro una singola CPU

Esempio: costruzione di una casa

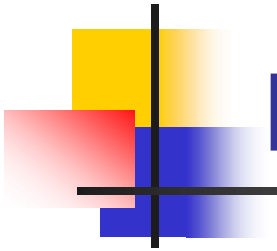


Secondo tipo di parallelismo

Secondo tipo di parallelismo

I tre operai eseguono contemporaneamente
la *stessa azione* su *mattoni diversi*

PARALLELISMO SPAZIALE

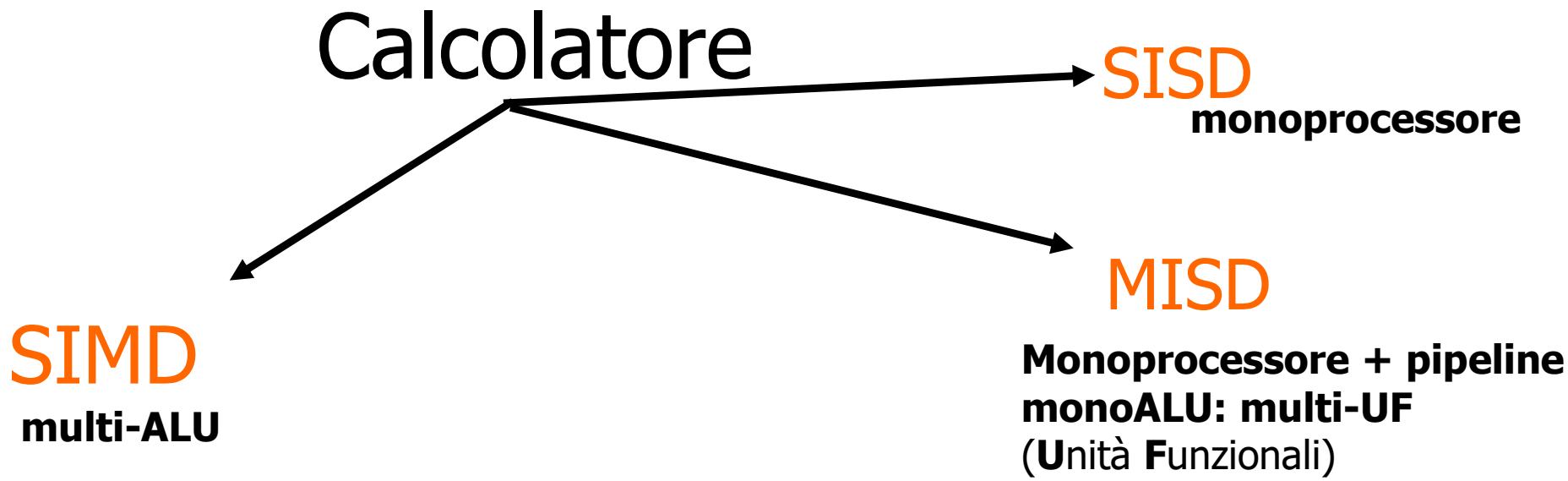


Parallelismo spaziale

Come è realizzato il
parallelismo spaziale in un calcolatore ?

Parallelismo nell'*unità operativa*
(più unità aritmetico-logiche)

Tassonomia di Flynn (dal 1965)



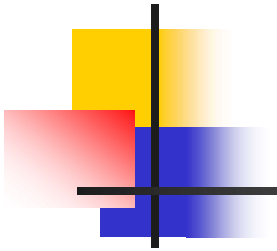
Calcolatori SIMD

(Single Instruction Multiple Data)

Terzo tipo di parallelismo

I tre operai eseguono contemporaneamente
azioni diverse su *parti diverse*

PARALLELISMO ASINCRONO



Parallelismo asincrono

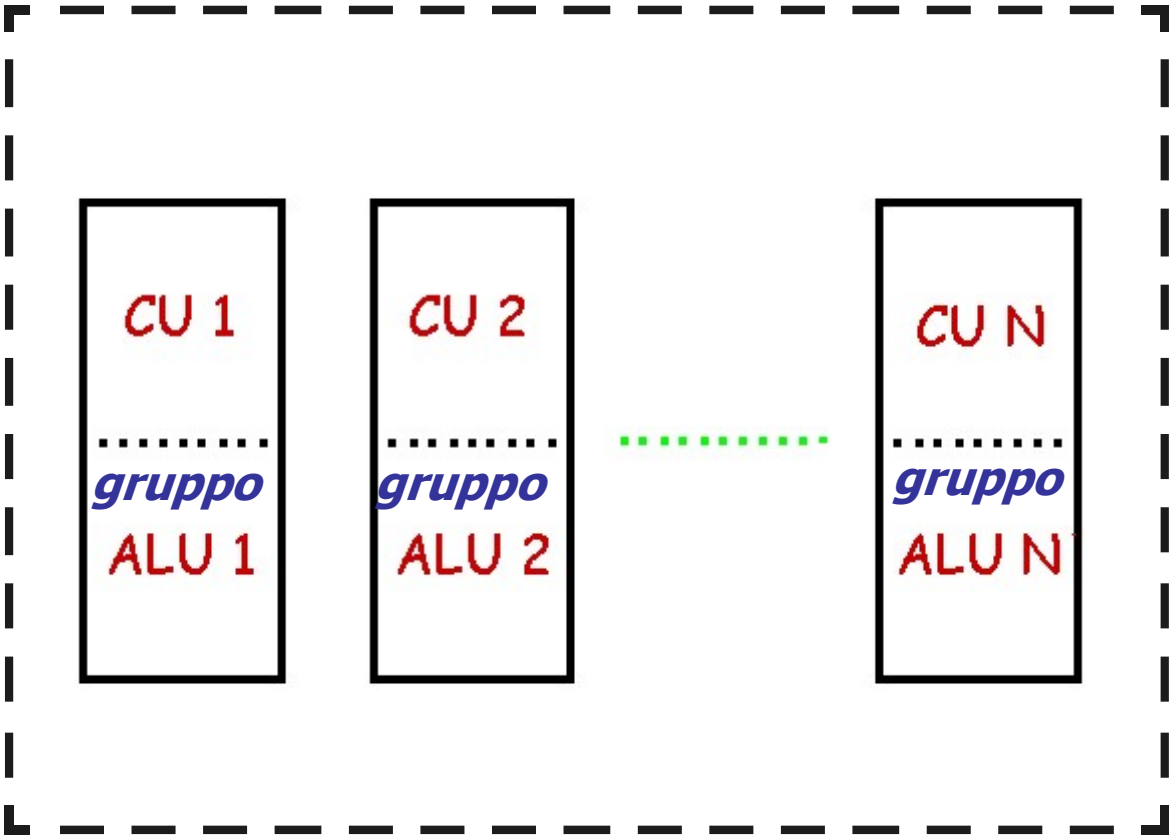
Differenti processori (CPU) cooperano eseguendo istruzioni diverse su dati diversi

Come è realizzato il parallelismo asincrono in un calcolatore ?

Parallelismo delle *CPU*
(più CPU = ALU+CU)

Parallelismo asincrono

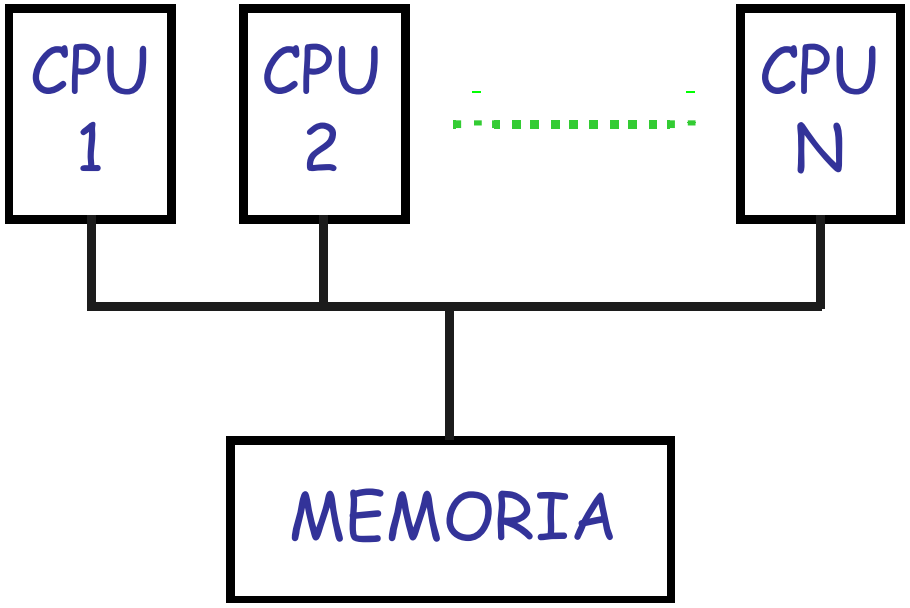
Più CPU (ALU + CU)
eseguono in parallelo
le ***istruzioni diverse***
su ***dati diversi***



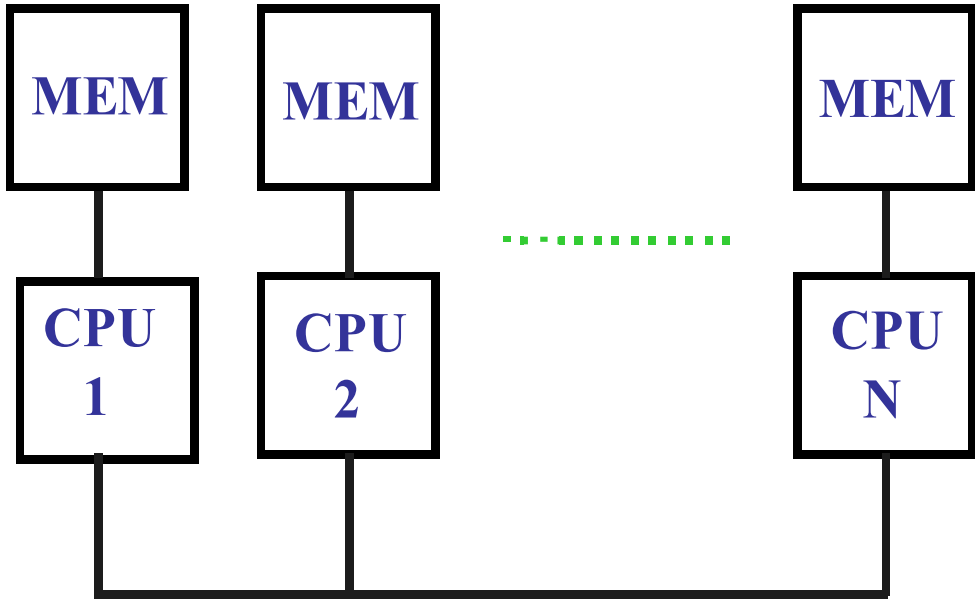
Calcolatori MIMD
(Multiple Instruction Multiple Data)

MIMD

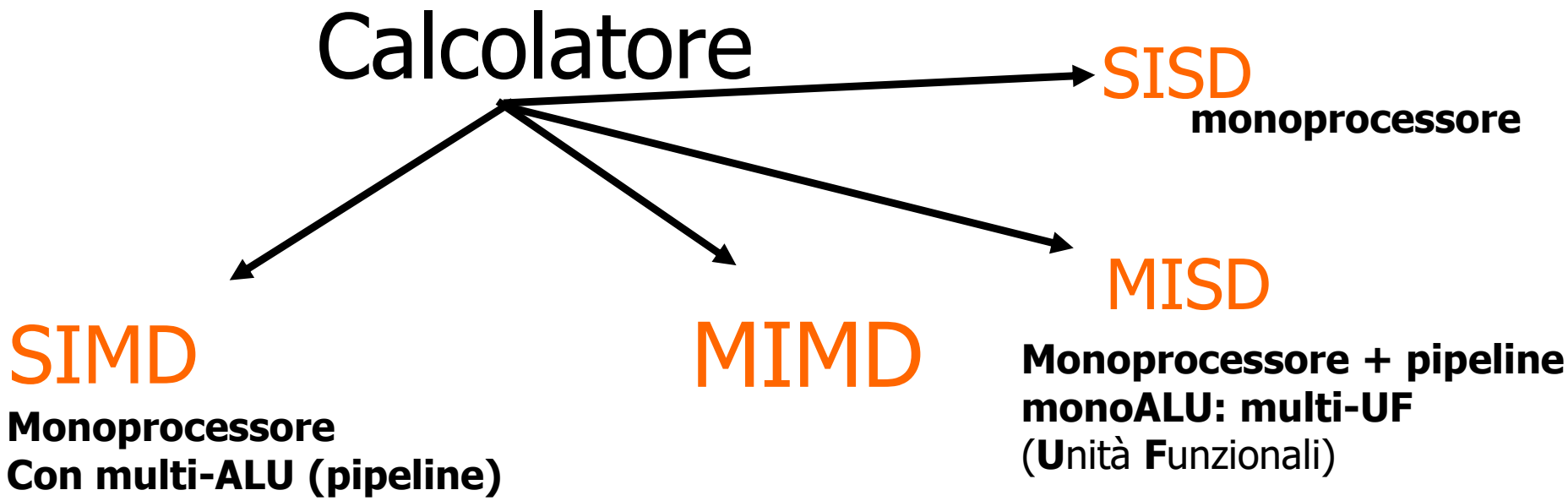
Calcolatori MIMD a memoria condivisa (shared-memory)

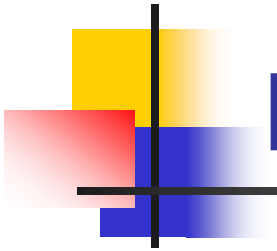


Calcolatori MIMD a memoria distribuita (distributed-memory)



Tassonomia di Flynn (dal 1965)



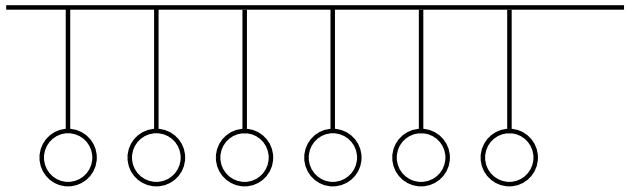


MIMD

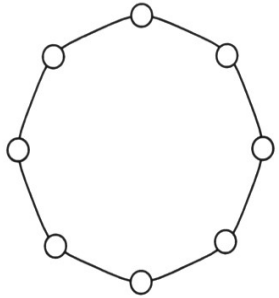
Come sono collegate
CPU e *memorie*
in un calcolatore
MIMD?

MIMD: collegamento CPU-memoria

Calcolatori MIMD a memoria distribuita

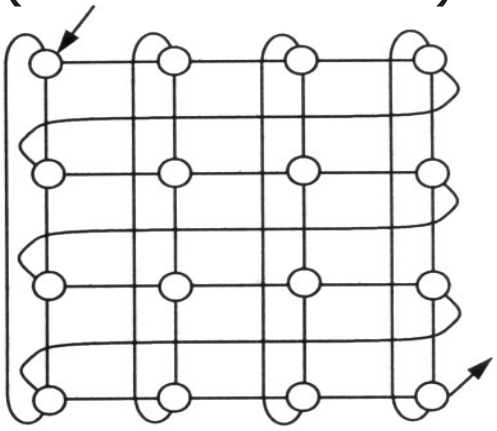


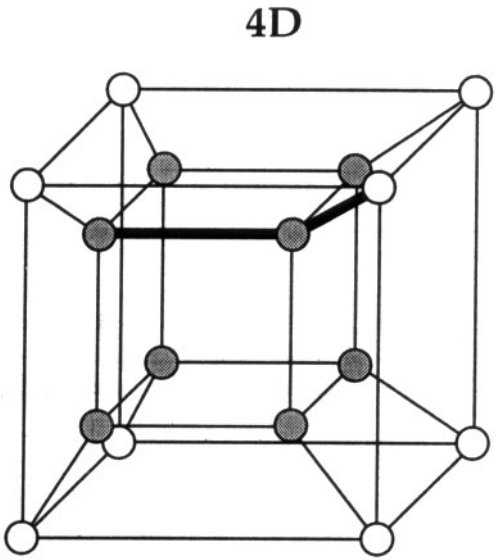
bus



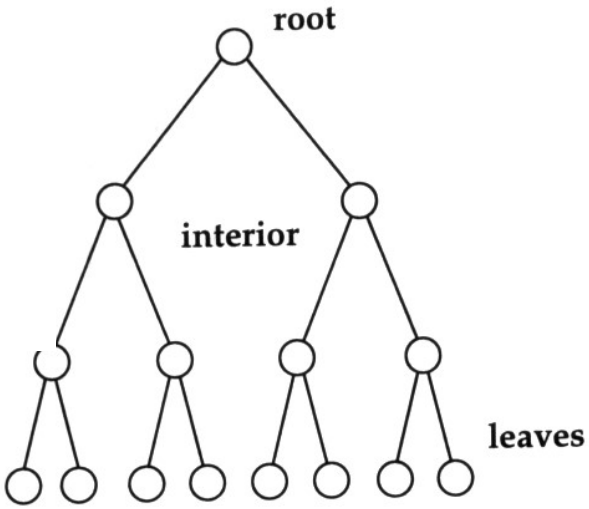
anello

reticolo 2d
(eventualm. *toro*)





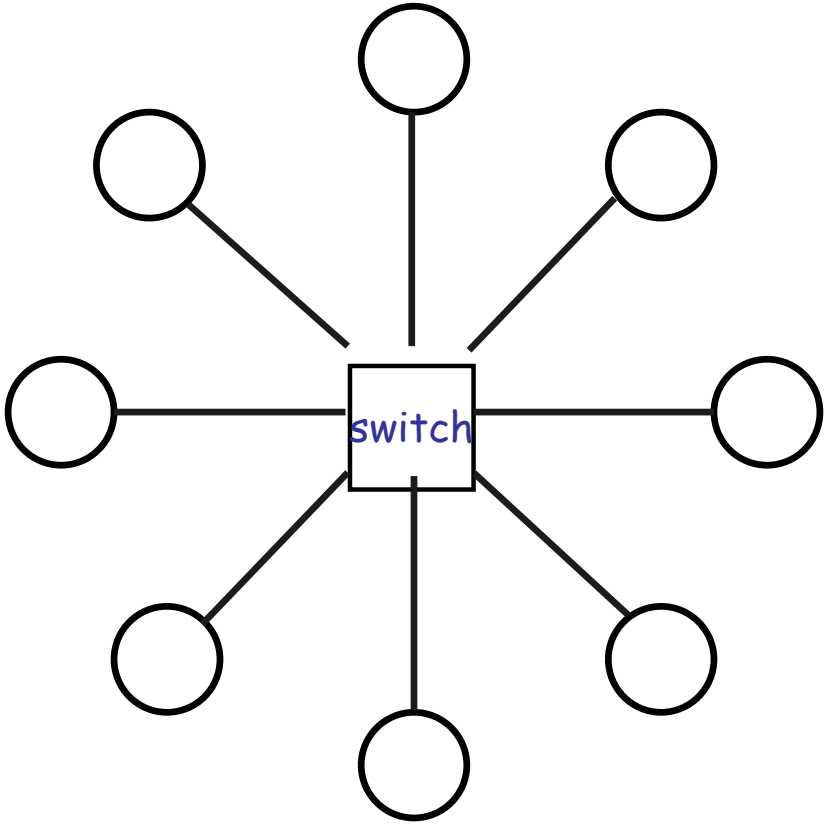
ipercubo



albero

MIMD: collegamento CPU-memoria

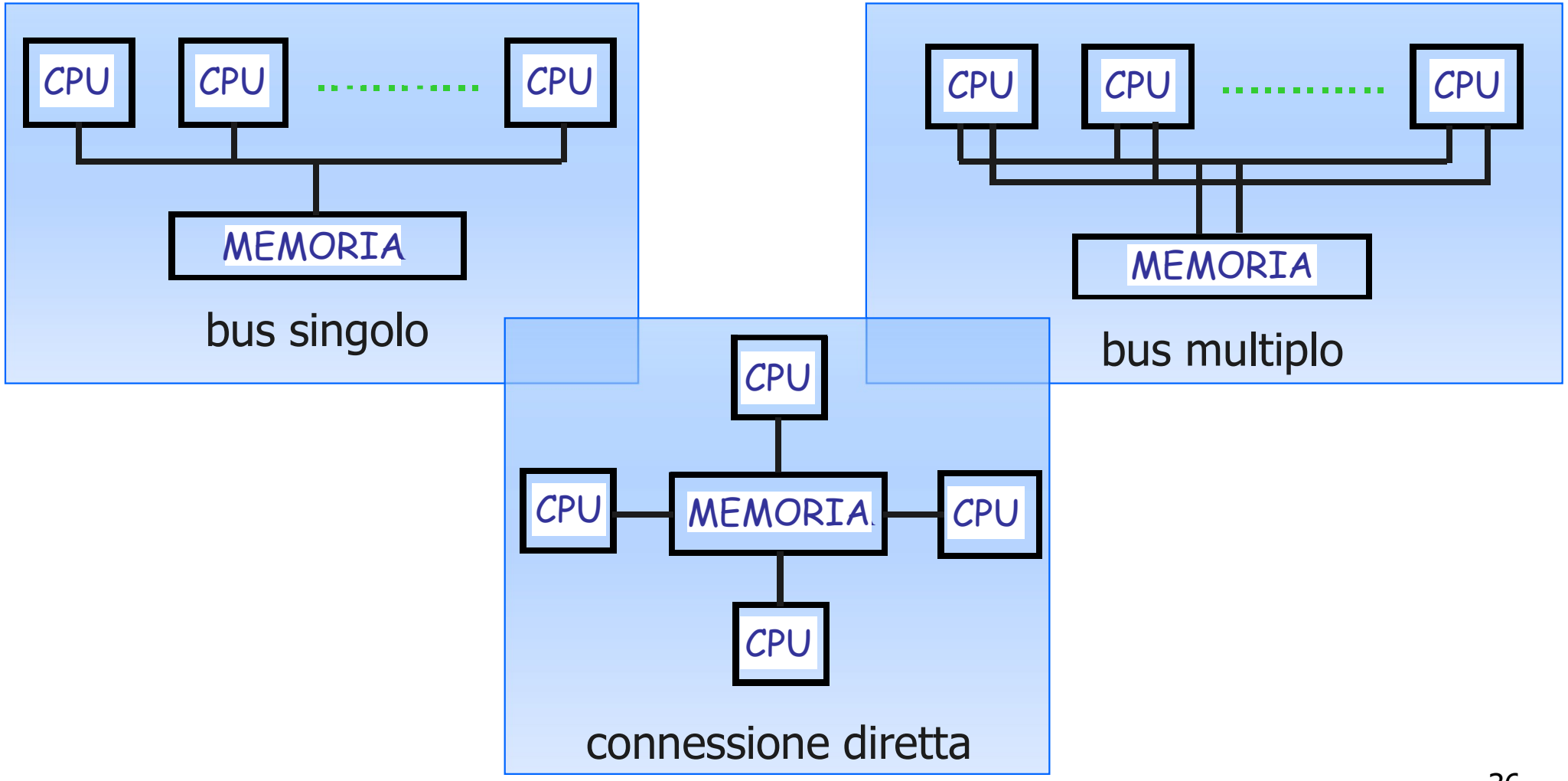
Calcolatori MIMD a memoria distribuita



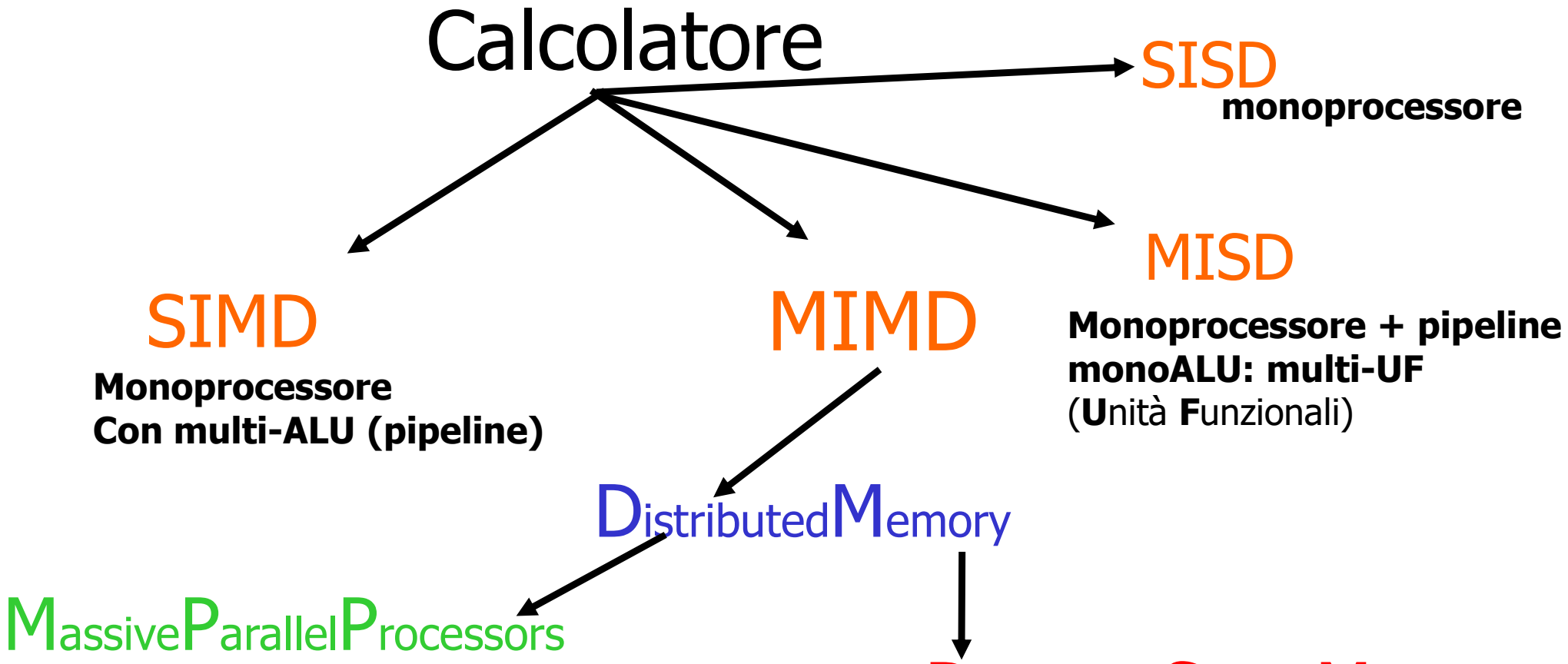
stella

MIMD: collegamento CPU-memoria

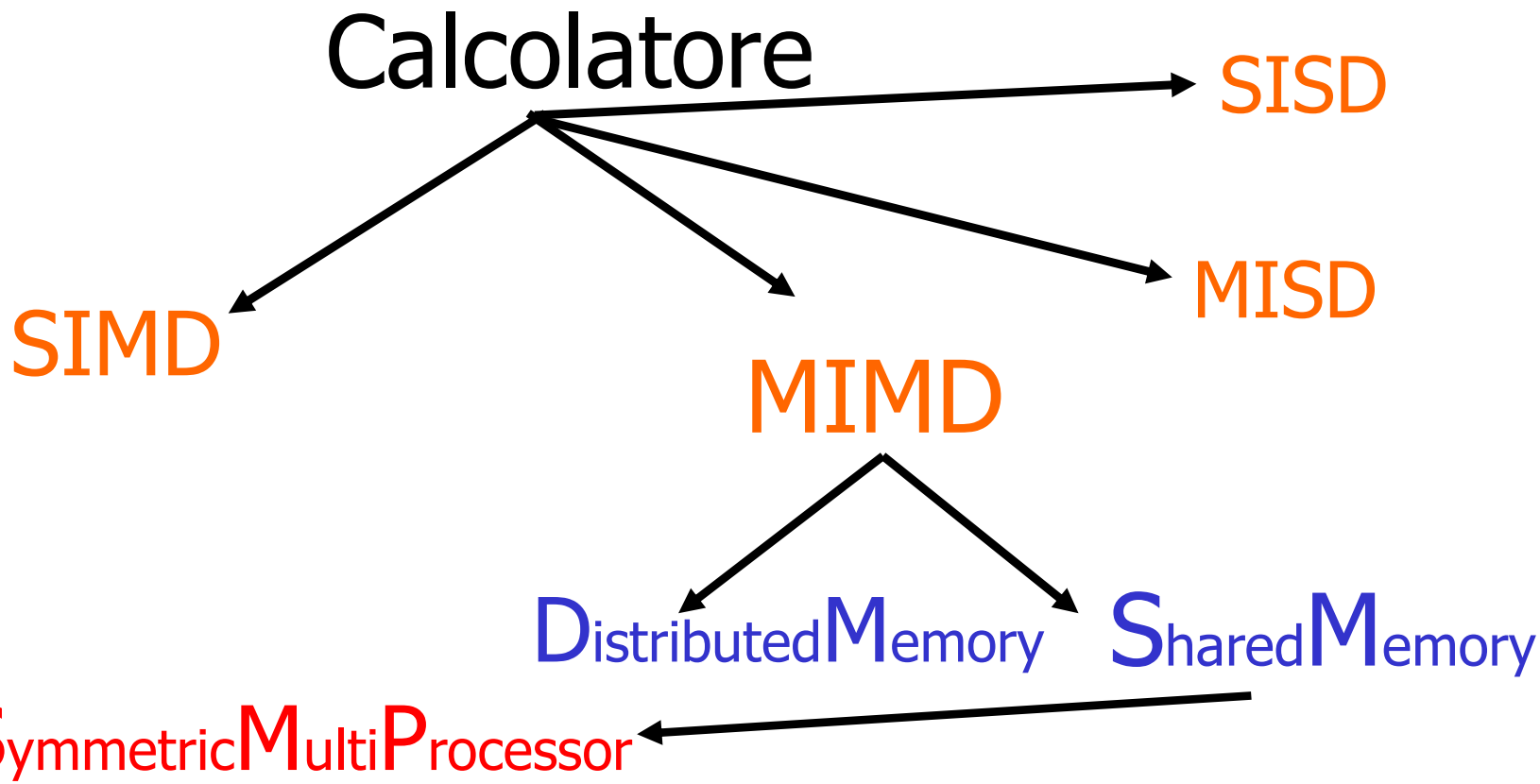
Calcolatori MIMD a memoria condivisa



Tassonomia di Flynn (dal 1965)

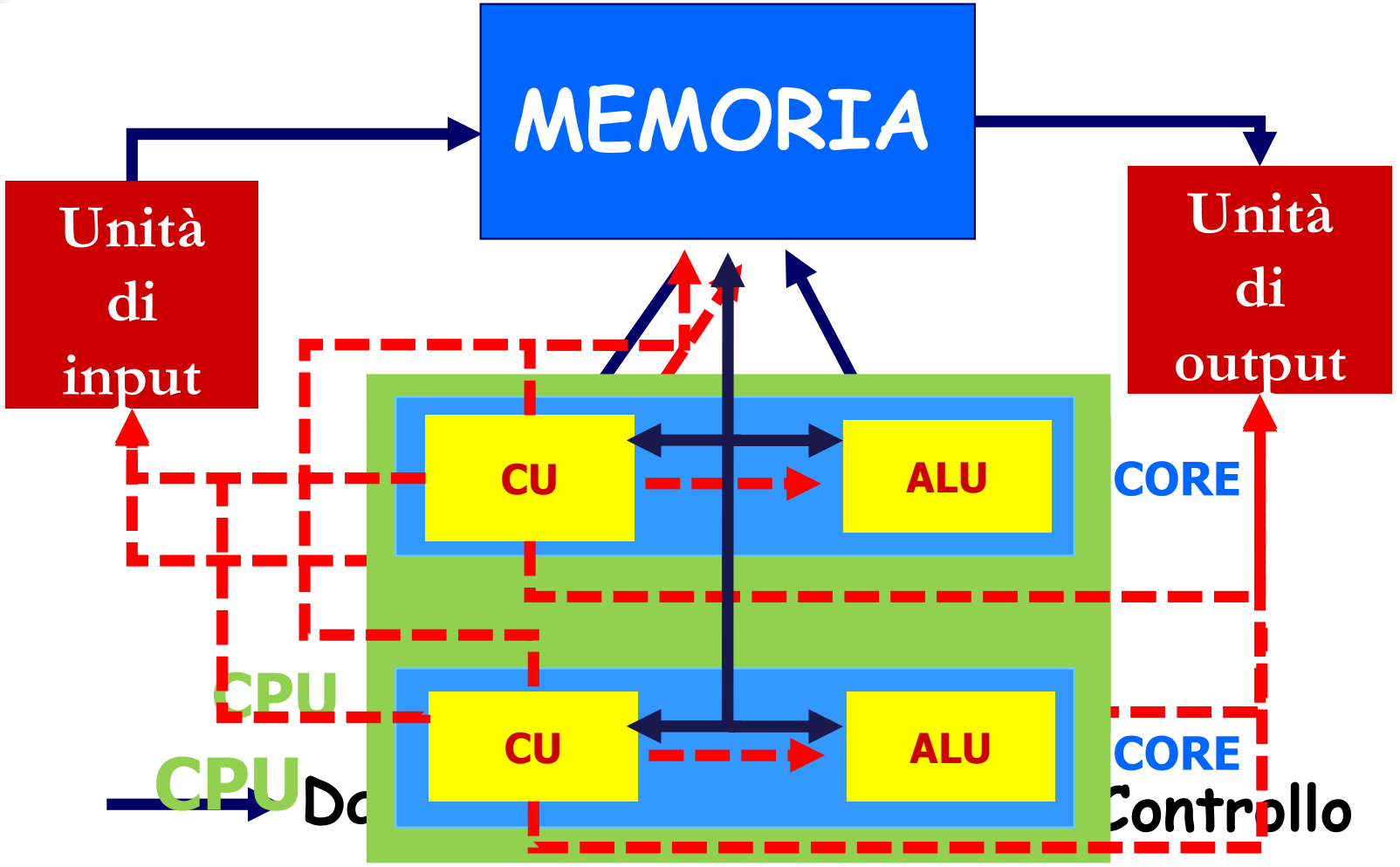


Tassonomia di Flynn (dal 1965)

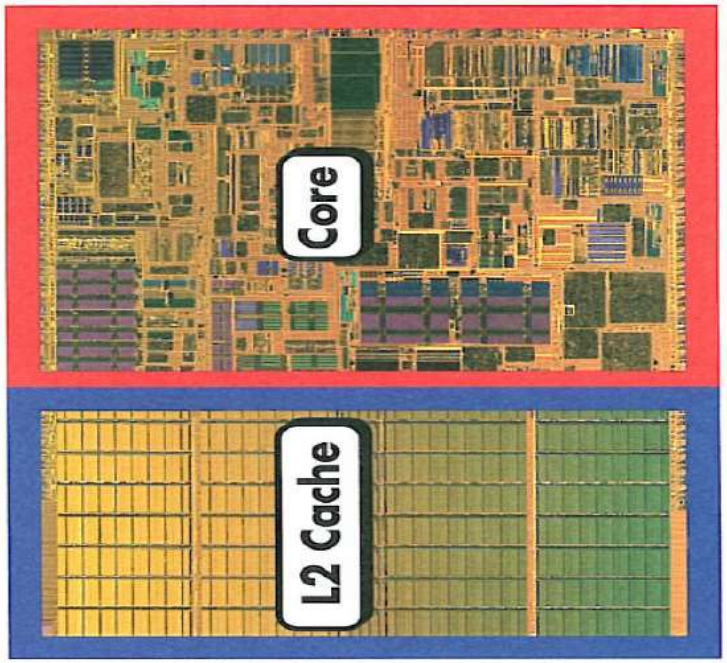


Termine utilizzato per un'architettura in cui due o più micro-processori **identici** sono connessi ad una singola memoria principale condivisa in un'unica macchina.
(**MULTICORE INTEL**)

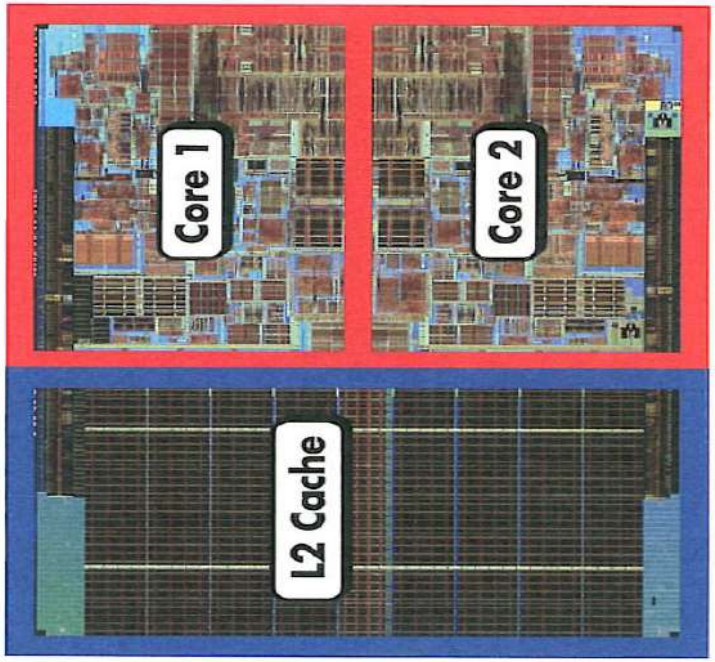
Multicore


MEMORIA
Unità di input
Unità di output
CU
ALU
CORE
CU
ALU
CORE
CPU
CPU
Controllo

Multicore: esempi



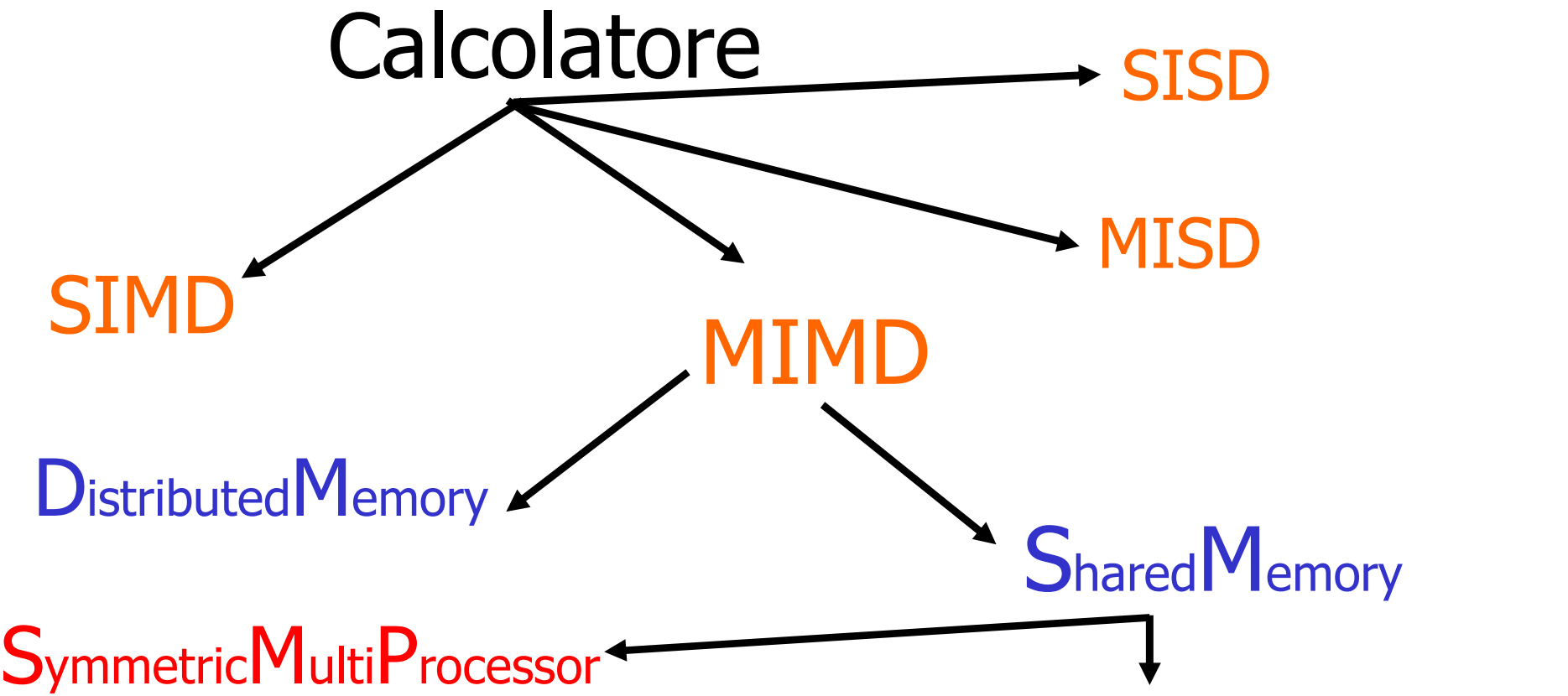
Processore Intel Pentium M (single core)



Processore Intel core duo

Secondo tipo di parallelismo *on-chip*

Tassonomia di Flynn (dal 1965)



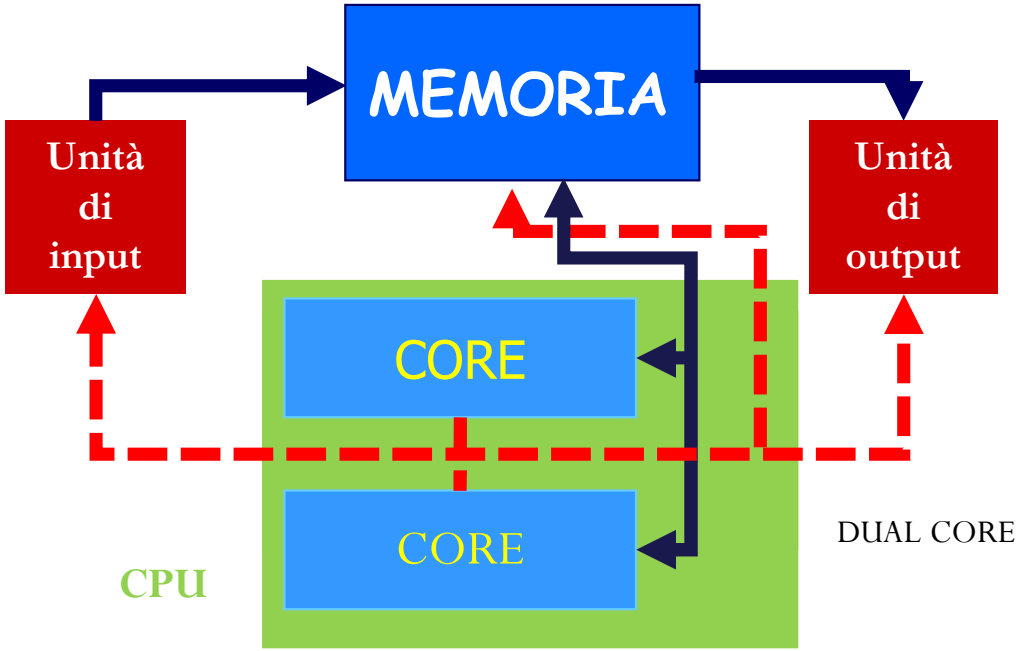
Termine utilizzato per un'architettura multiprocessore in cui due o più processori **identici** sono connessi ad una singola memoria principale condivisa. (**MULTICORE INTEL**)

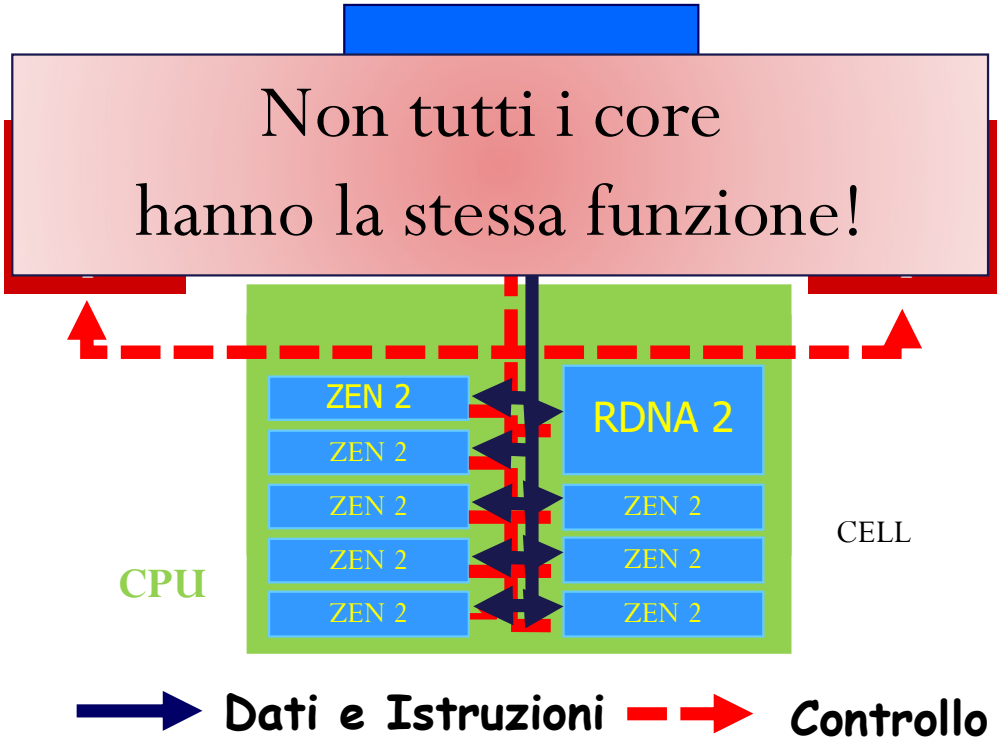
Molte CPU (o unita' processanti) con ruoli differenti in un'unica macchina (**cell IBM, GPU (Graphics Processing Unit), nVidia (TESLA**)

Esempio di calcolatore MIMD-ASMP

Sony (PlayStation 5) novembre 2020

- CPU: **8** Zen 2 Cores at 3.5GHz
- GPU: **1** RDNA 2 at 10.28 TFLOPs, 36 CUs at 2.23GHz





Graphics Processing Units (GPUs)

Dispositivi paralleli che ad oggi si sono guadagnate tanta fama, nascono nel **2000** per l'elaborazione grafica.

La loro architettura si e evoluta molto rapidamente e oggi rappresentano la **più sofisticata forma di macchina parallela.**

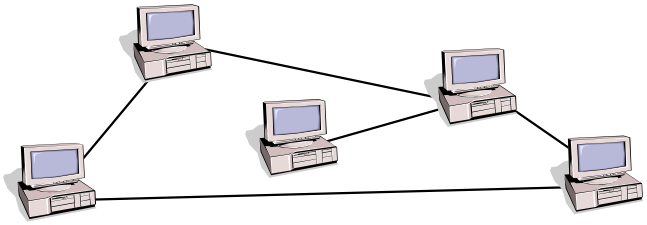
Oggi sono usate anche per il genaral purpose **GP-GPU** e sono argomento centrale del corso di High Performance Computing del CdL magistrale in Informatica Applicata (Machine Learning and Big Data)

Attualmente
tutti i microprocessori
utilizzano forme diverse di parallelismo, e nessun sistema in commercio si può definire puramente sequenziale.

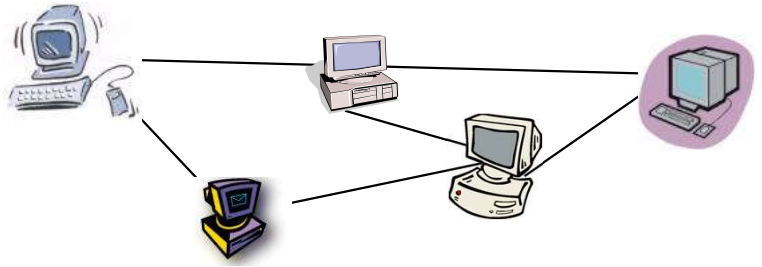
Architetture MIMD-DM

cluster multiprocessore

Cluster omogeneo



Cluster eterogeneo

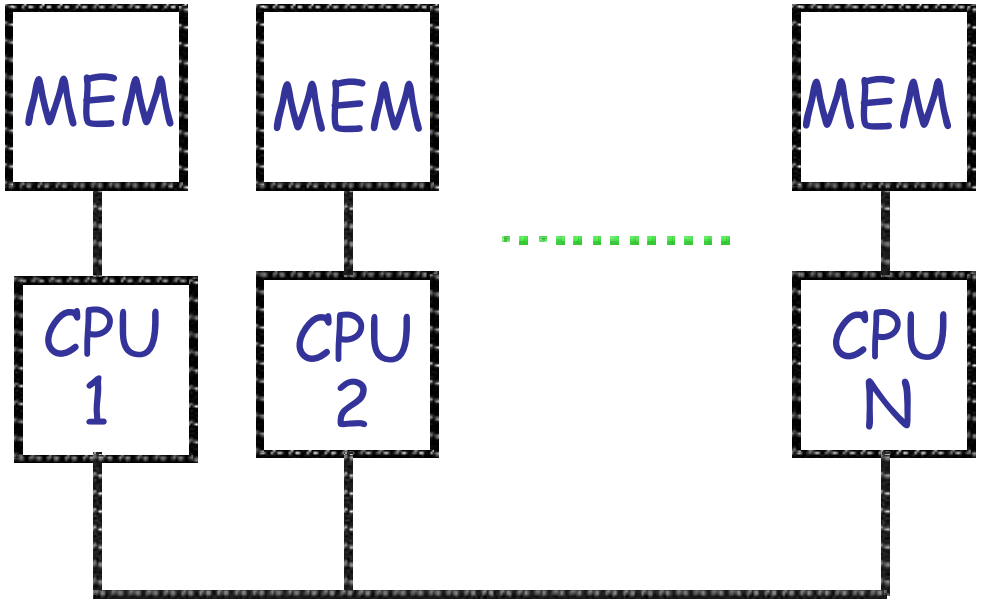


Cluster di processori ognuno con una propria memoria e una propria CPU, ovvero insiemi di calcolatori autonomi collegati tra loro attraverso reti di connessione di I/O, e quindi con connettori e cavi tipici di una rete standard.

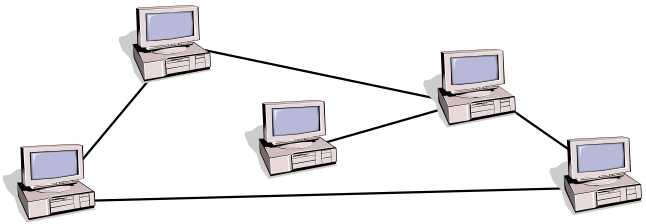
Ogni calcolatore ha una sua **copia distinta di sistema operativo**, il che fa **crescere i costi di amministrazione**, di contro le più **memorie connesse** possono essere utili in caso di **problemi di grandi dimensioni.**

Architetture MIMD-DM

cluster multiprocessore - strumenti



Cluster omogeneo



Message Passing Interface
MPI

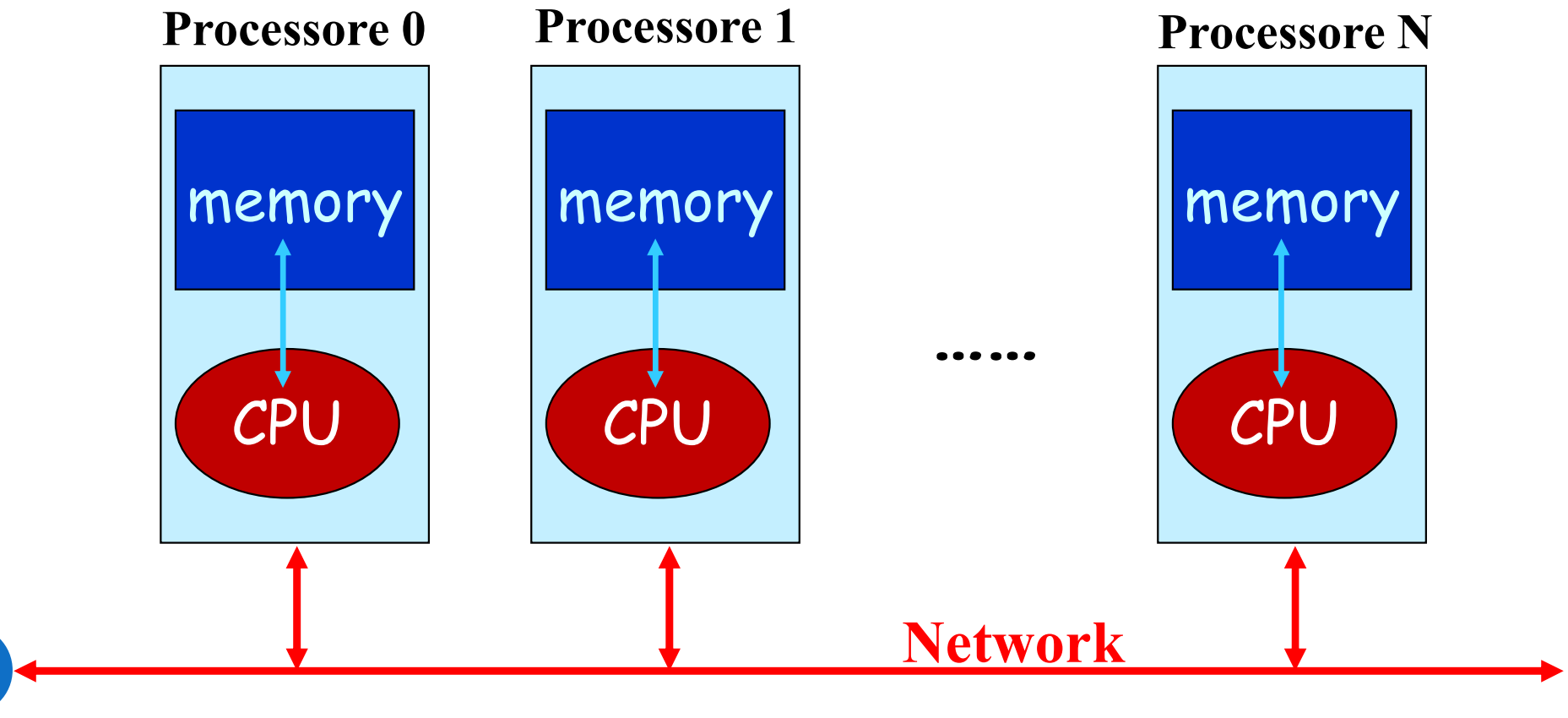
Necessità di organizzare le comunicazioni tra i processore

Architetture MIMD-DM

cluster multiprocessore - strumenti

Message Passing Interface
MPI

Ogni processore ha una **propria memoria** locale alla quale accede direttamente e può conoscere i dati in memoria di un altro processore o far conoscere i propri, attraverso il **trasferimento di dati**.



Architetture MIMD-DM

cluster multiprocessore - strumenti

La **libreria MPI** nasce nel **1991** e da allora molteplici sono le versioni proposte per renderla più *user friendly*.
Ad oggi, qualunque altra libreria per lo sviluppo di codice parallelo in ambiente MIMD-DM si poggia su MPI e sul suo paradigma di scambio messaggi:

- **PBLAS** (**P**arallel **B**asic **L**inear **A**lgebra **S**ubprograms), basata su **BLAS**

- **ScaLAPACK** (**Sca**lable **L**inear **A**lgebra **PACK**age), basata su **LAPACK**

- …
- …

- **PETSc** (**P**ortable, **E**xtensible **T**oolkit for **S**cientific **C**omputation)

Per la risoluzione numerica (in ambiente MIMD-DM) di
problemi modellati da equazioni differenziali (ODE-PDE)
ed in particolare per trattare sistemi di equazioni lineari e non lineari
che rappresentano il nucleo computazionale della discretizzazione di ODE e PDE.

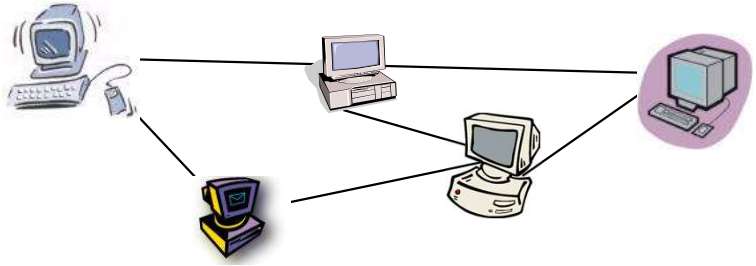
Architetture MIMD-DM

cluster multiprocessore - strumenti

Oggi il calcolo parallelo in ambiente distribuito può considerarsi un po' *agè* rispetto alle nuove architetture HPC, soprattutto per i tempi necessari al trasferimento dati!
Resta, comunque, di grande utilità per problemi caratterizzati da una grande quantità di dati…

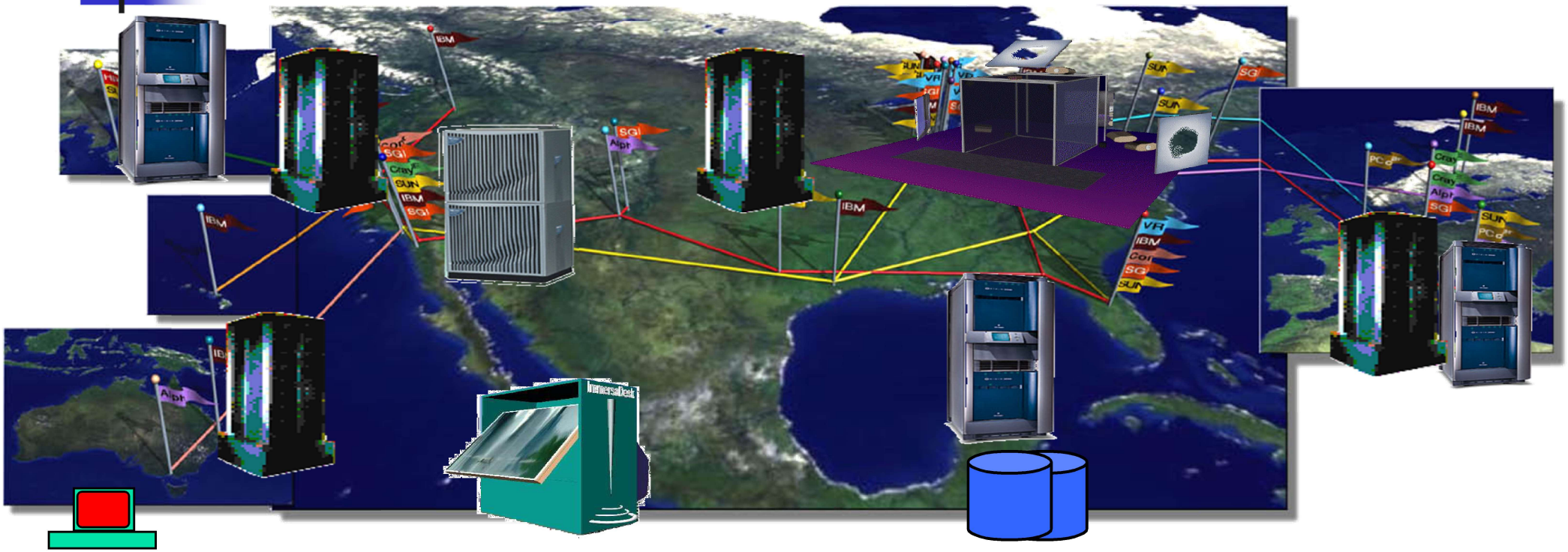
…ma soprattutto si declina in modo molto utile nelle nuove tecnologie del **GRID computing** e in particolare nel **CLOUD computing o storage**

Cluster eterogeneo

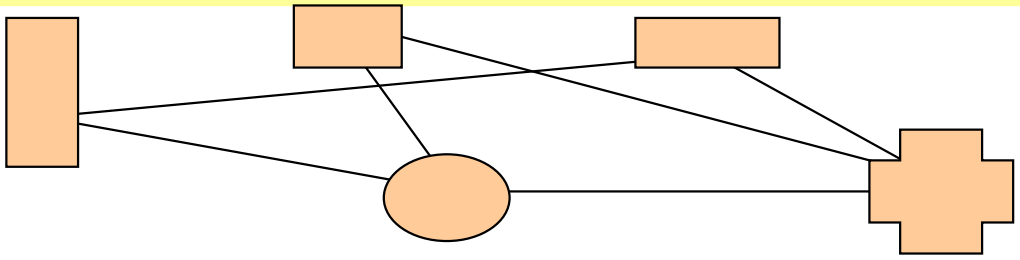


Ri-uso di risorse esistenti:
gli utenti possono acquisire e rilasciare
le risorse dinamicamente
in base alle esigenze o al carico offerto

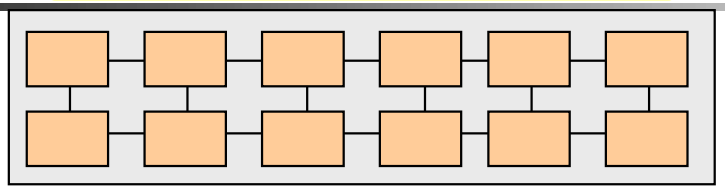
CALCOLATORE DISTRIBUITO



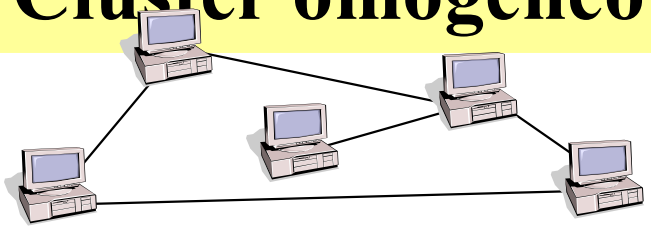
Calcolatore Distribuito (**Griglia**)



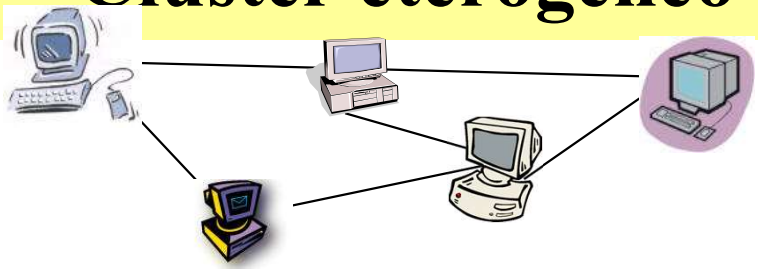
Calcolatore Parallelo



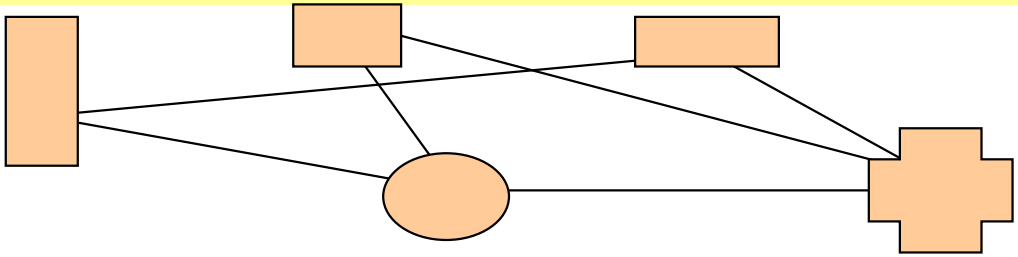
Cluster omogeneo



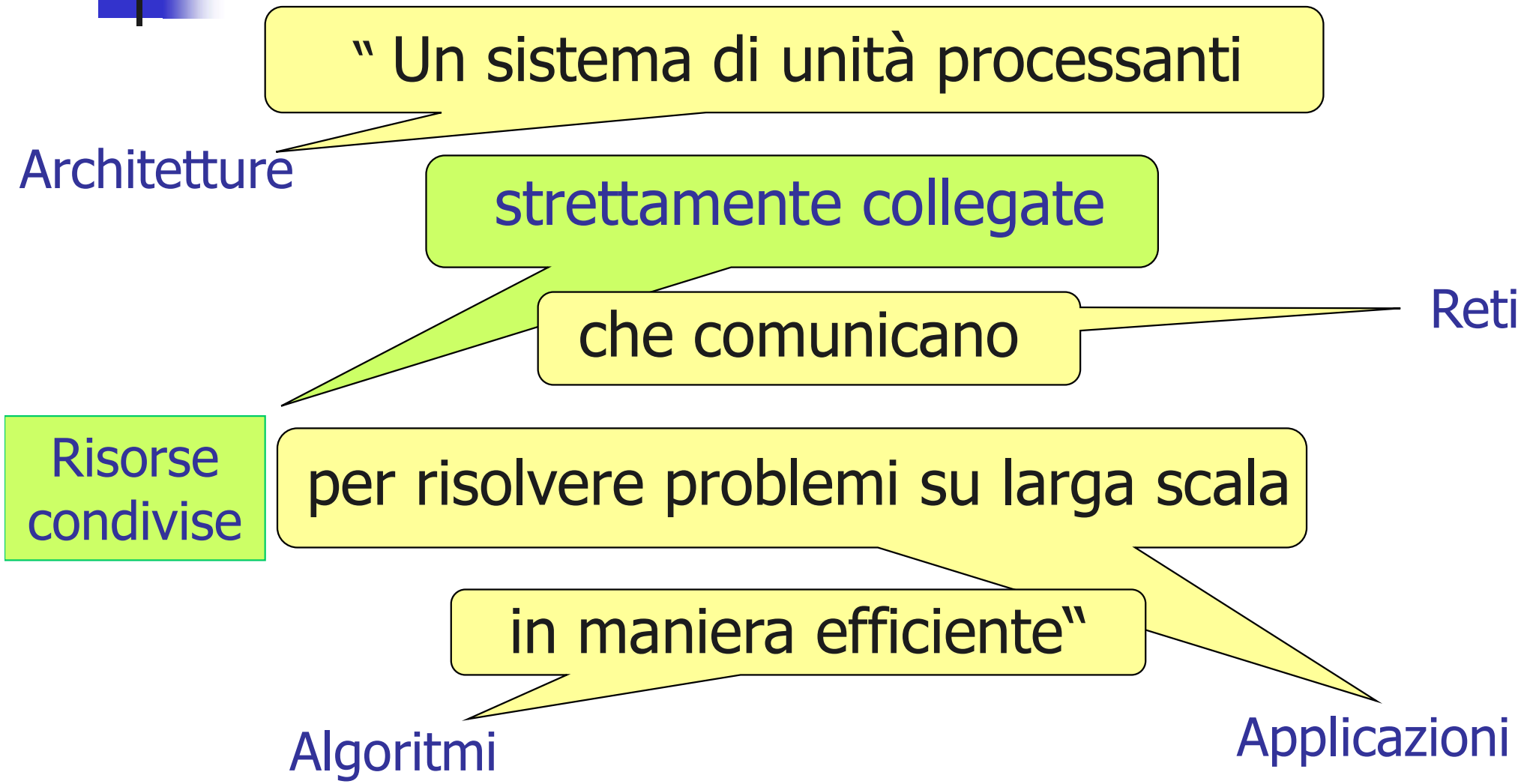
Cluster eterogeneo



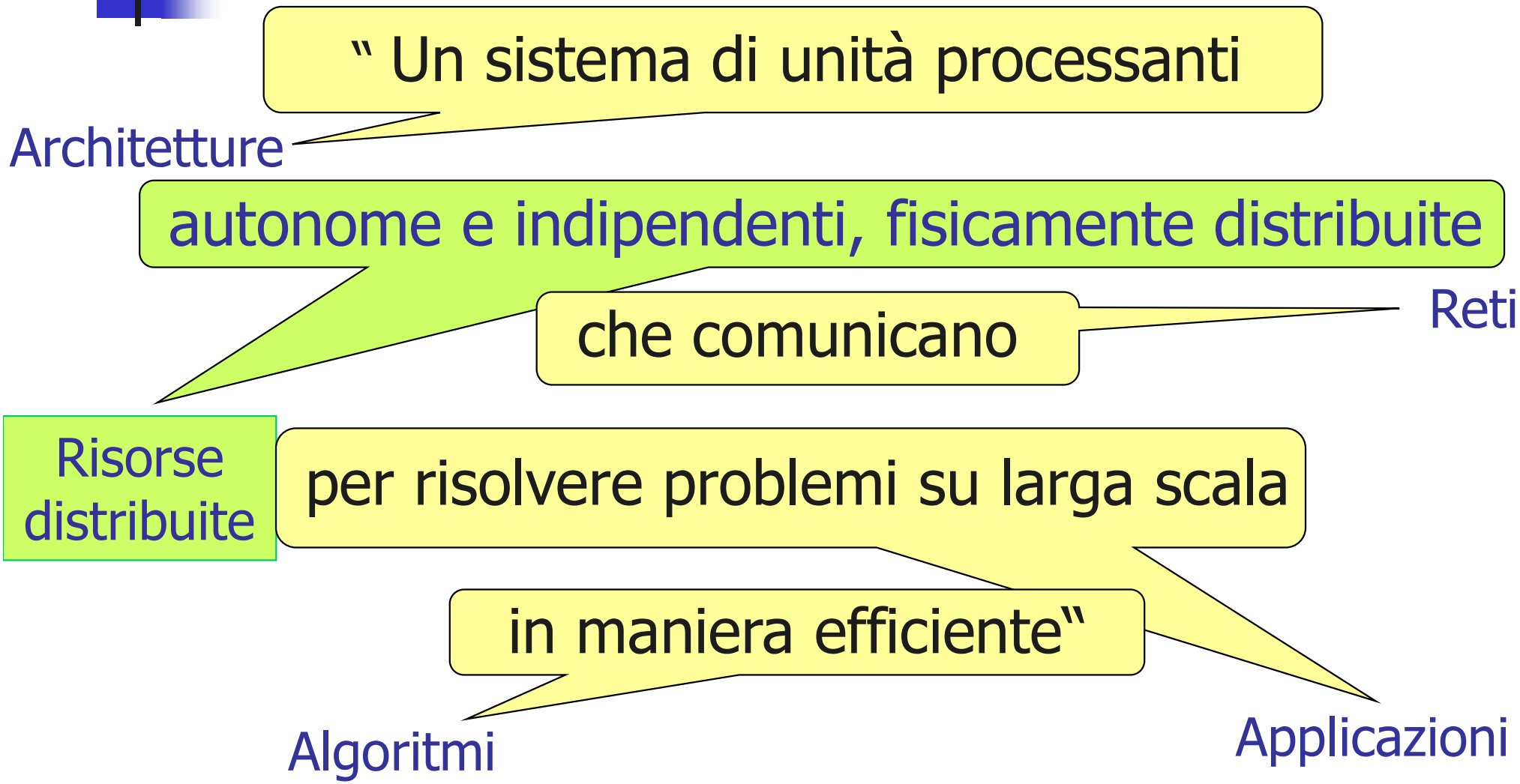
Calcolatore Distribuito (**Griglia**)



Che cosa è un ambiente di calcolo parallelo?



Che cosa è un ambiente di calcolo distribuito?



Parallelo vs distribuito (1)

Calcolatore parallelo

sistema di nodi collegati da switch specializzati e dedicati

(tightly coupled systems)

Sistema ad arch. distribuita

sistema di nodi collegati da reti geografiche

(loosely coupled systems)

La differenza è nella rete di connessione

Infrastruttura Cloud

Gli utenti possono acquisire e rilasciare le risorse dinamicamente in base alle esigenze o al carico offerto

Parallelo vs distribuito (2)

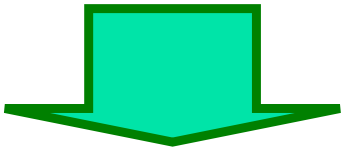
Calcolatore parallelo

Principale obiettivo:
Performance

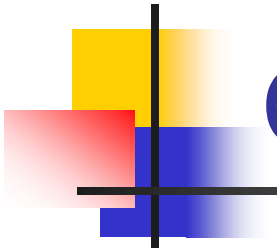
Risorse di calcolo omogenee

Sistema ad arch. distribuita

Principale obiettivo:
Ri-uso di risorse esistenti



Risorse di calcolo eterogenee



Cos’ è il calcolo parallelo/distribuito?

“in parallel computing we decompose into parts,

in distributed computing we assemble parts”

G.J. Fox, IEEE CiSE, 2002

“nel calcolo parallelo decomponiamo il problema,

nel calcolo distribuito assembliamo le risorse”

Calcolo parallelo… al fine di

Ridurre il tempo necessario alla risoluzione computazionale di un problema reale

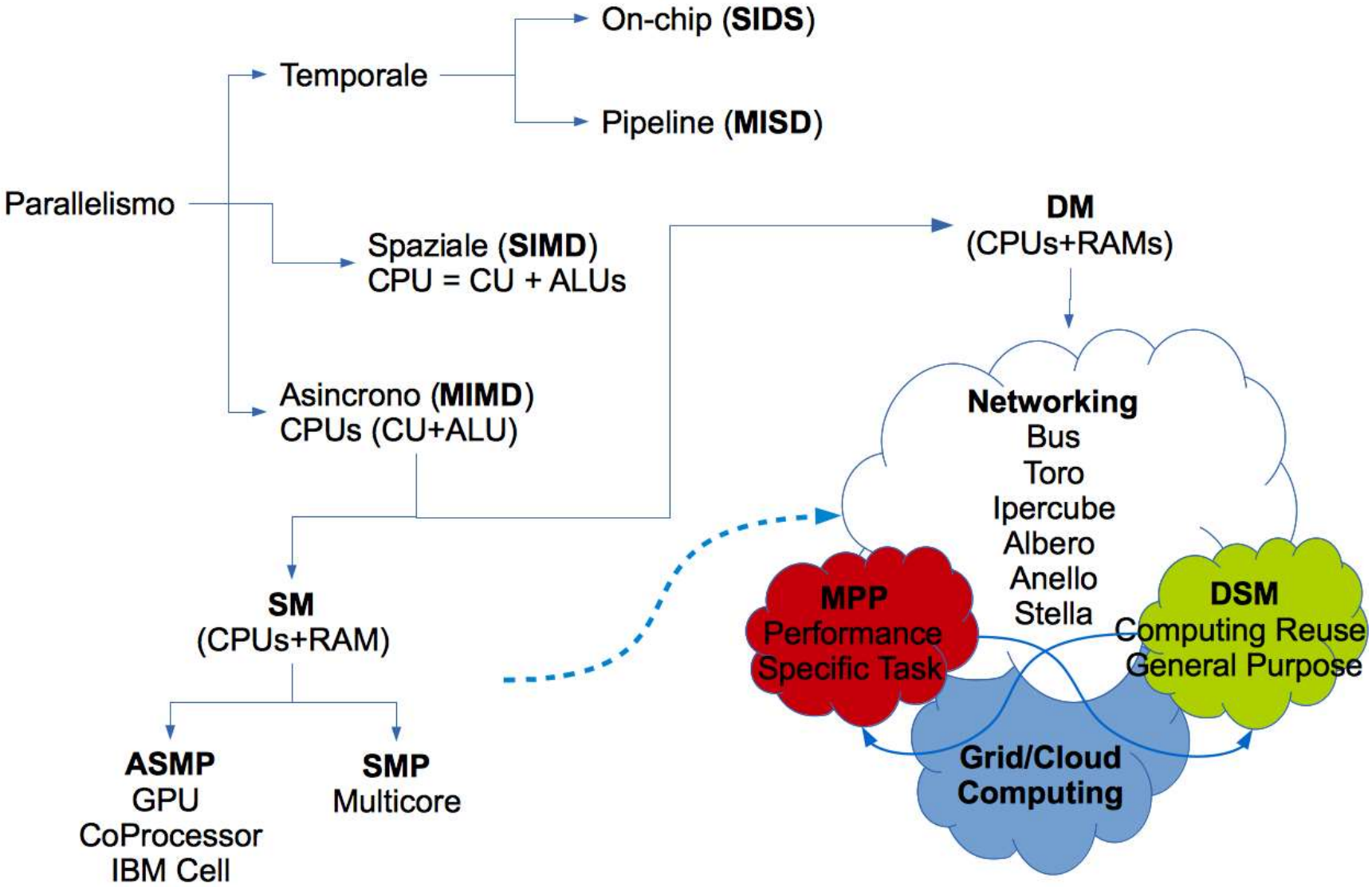
“ wall - clock” time

Calcolo distribuito, …al fine di

Riutilizzare “efficacemente” risorse hardware e software distribuite geograficamente sul territorio

“(ri)uso efficiente delle risorse”

Overview


On-chip (SIDS)
Temporale
Pipeline (MISD)
Parallelismo
Spaziale (SIMD)
CPU = CU + ALUs
Asincrono (MIMD)
CPUs (CU+ALU)
DM
(CPUs+RAMs)
Networking
Bus
Toro
Ipercube
Albero
Anello
Stella
SM
(CPUs+RAM)
MPP
Performance
Specific Task
DSM
Computing Reuse
General Purpose
Grid/Cloud
Computing
ASMP
GPU
CoProcessor
IBM Cell
SMP
Multicore