Programmazione Avanzata e parallela

Lezione 01

Luca Manzoni

Prerequisiti

In teoria li avete tutti

- Programmazione di base in C
  - Puntatori e allocazione dinamica di memoria
  - Strutture
  - Definire funzioni
- Programmazione di base in Python
  - Definire funzioni

Prerequisiti

In teoria li avete tutti

- Architetture e sistemi operativi
  - Funzionamento di base di una CPU
  - Programmazione multithread
- Algoritmi e strutture dati
  - Complessità di un algoritmo
  - Strutture dati di base (code, pile, alberi binari, etc.)

Argomenti del corso

Struttura generale

- Il corso è diviso in due parti principali:
  - Come l'architettura dei calcolatori moderni influenza le performance
  - Strutturare codice in modo che sia facilmente mantenibile e programmazione ad oggetti
- La prima parte farà uso del linguaggio C
- La seconda parte sarà invece incentrata su Python
- Vi sarà un momento "intermedio" riguardante come possiamo utilizzare alcune librerie "ad alte prestazioni" in Python

Argomenti del corso

Parte 1

- Gestire progetti in C:
  - File header, compilazione separate e librerie
  - Makefile per gestire la compilazione
- Struttura di un calcolatore moderno:
  - Pipeline, esecuzione out-of-order e branch prediction
  - Cache e gerarchia di memoria

Argomenti del corso

Parte 1

- Misurare le performance dei programmi
- Esempi di strutture dati che tengono conto della località di memoria
- Strutture dati per lavorare con la memoria di massa (e.g., B-trees)
- Istruzioni vettoriali (SIMD)
- Pattern di programmazione parallela con OpenMP
- Basi della struttura delle GPU

Argomenti del corso

Parte 2

- Chiamare C da Python
- Classi e oggetti in Python, ereditarietà e polimorfismo (recap)
- Metodi e metodi speciali
- Programmazione funzionale
- Introduzione a numpy e librerie per la programmazione parallela
- Seminari su altri linguaggi di programmazione

Esame

Struttura generale

- Esame in tre parti: progetto, scritto e orale
- Progetto: 50% del voto finale
- Scritto e orale: 50% del voto finale
- Lo scritto consiste in domande su tutti gli argomenti del corso
- Si accede all’orale con votazione positiva sia su progetto che su esame scritto
- Le domande dell’orale saranno su tutti gli argomenti del corso e sul progetto presentato

Esame

Progetto

- Il progetto è composto da due parti:
  - Una parte in C
  - Una parte in Python
- Progetto pubblicato all’inizio di novembre
- Il progetto deve essere svolto **individualmente**
- Il testo del progetto include istruzioni per la consegna (come consegnare e scadenze) che devono essere rispettate
- Il voto del progetto rimane valido per l’anno accademico*

*a meno che non risulti che il progetto non è stato svolto individualmente

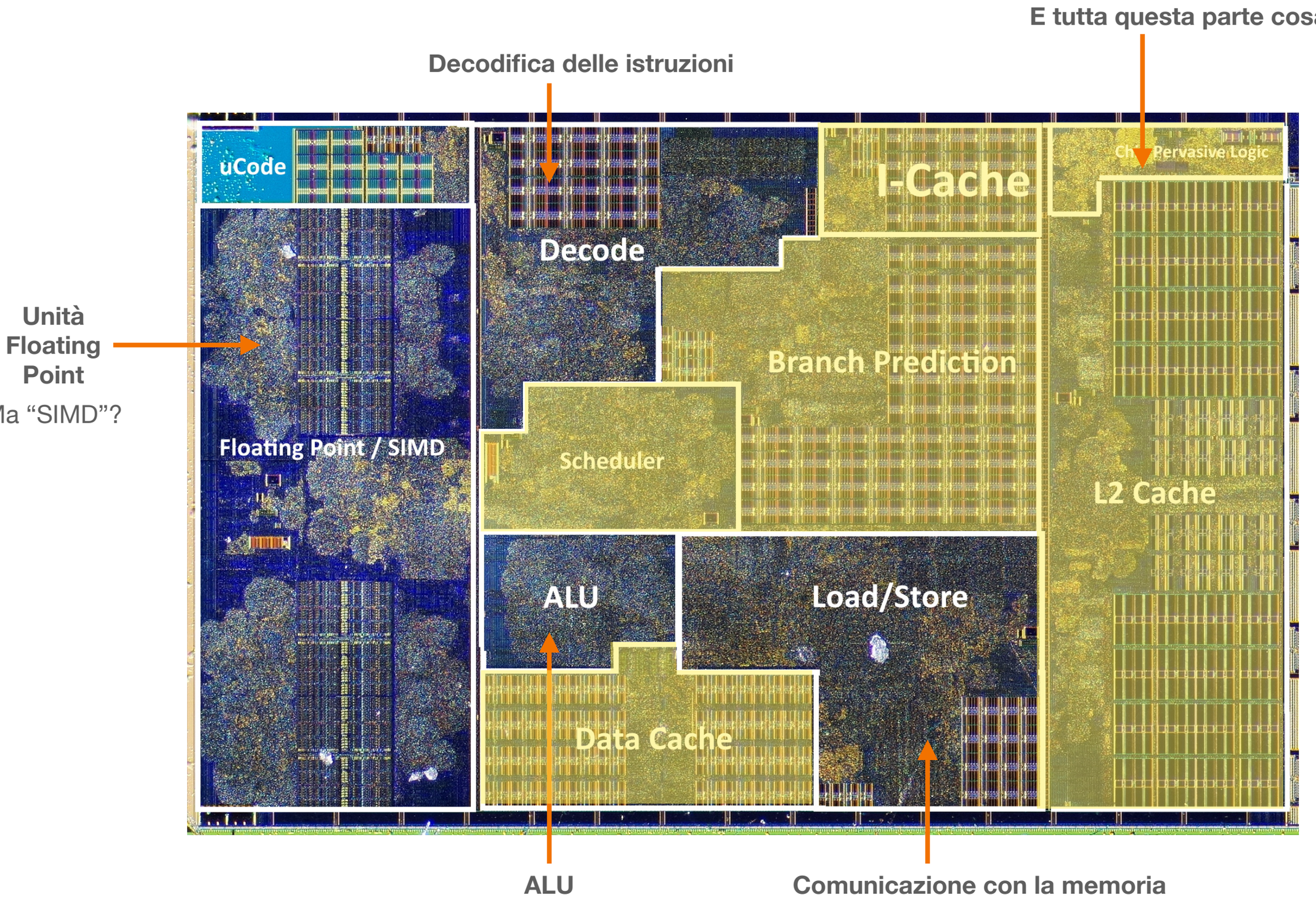
E ora qualche esempio

I calcolatori sono molto più complessi di come
avete visto ad architetture…

…e questo influenza le prestazioni
dei programmi che scriviamo

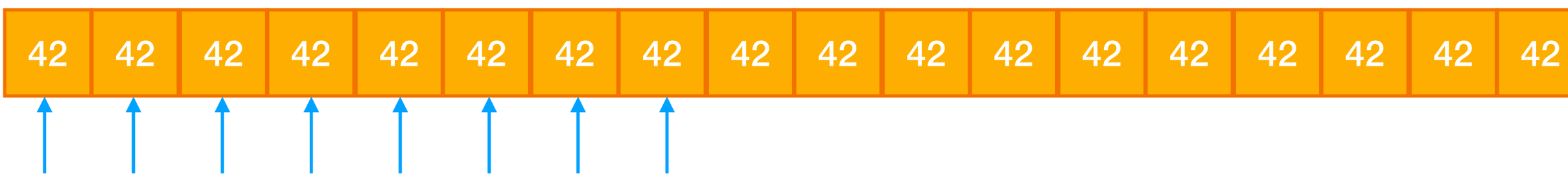
Struttura del calcolatore

Una vera CPU

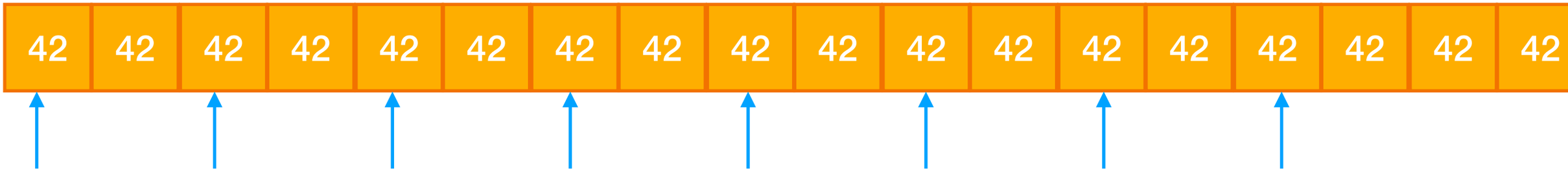


Accediamo alla memoria

E risultati inattesi



Sommiamo n valori con indici a distanza 1



Sommiamo n valori con indici a distanza 2

Sommiamo n valori con indici a distanza 4

Sommiamo n valori con indici a distanza 8

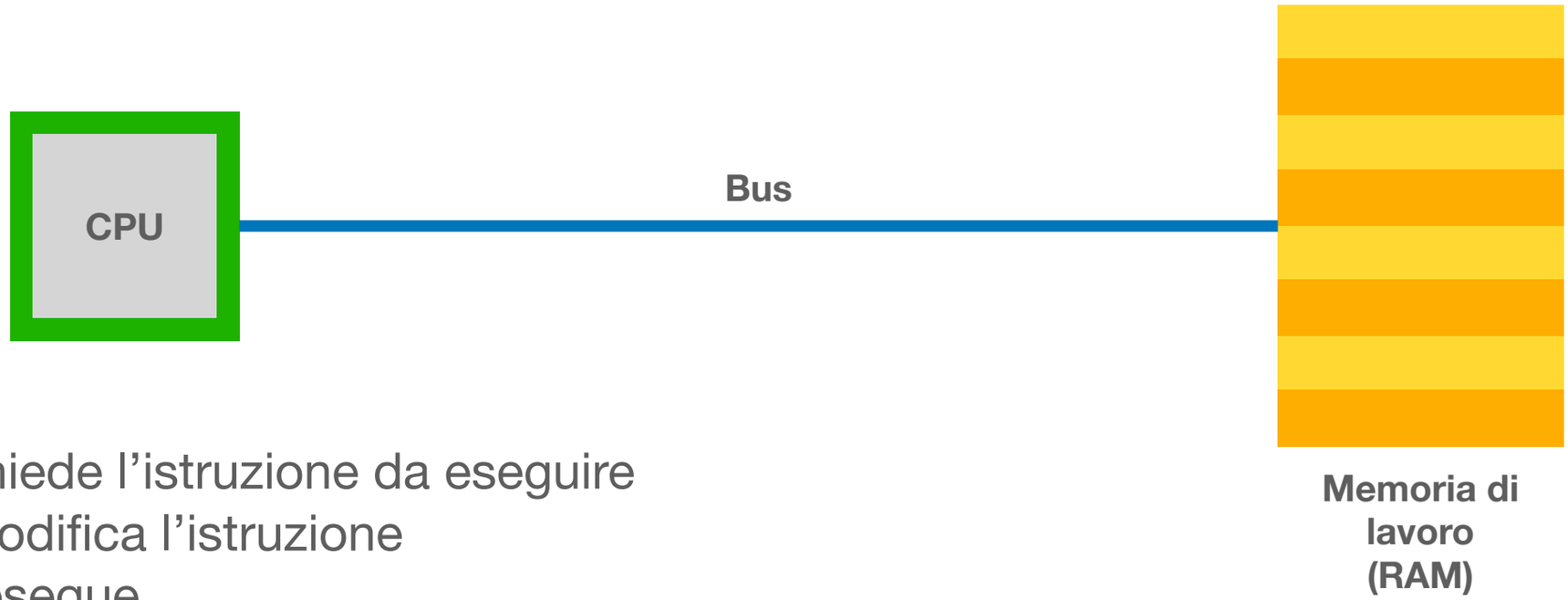
Sommiamo n valori con indici a distanza 16

$\vdots$

Ci aspettiamo tempi di esecuzione differenti?

Struttura del calcolatore

Un breve riassunto



- Richiede l’istruzione da eseguire
- Decodifica l’istruzione
- La esegue

- Contiene il programma e i dati

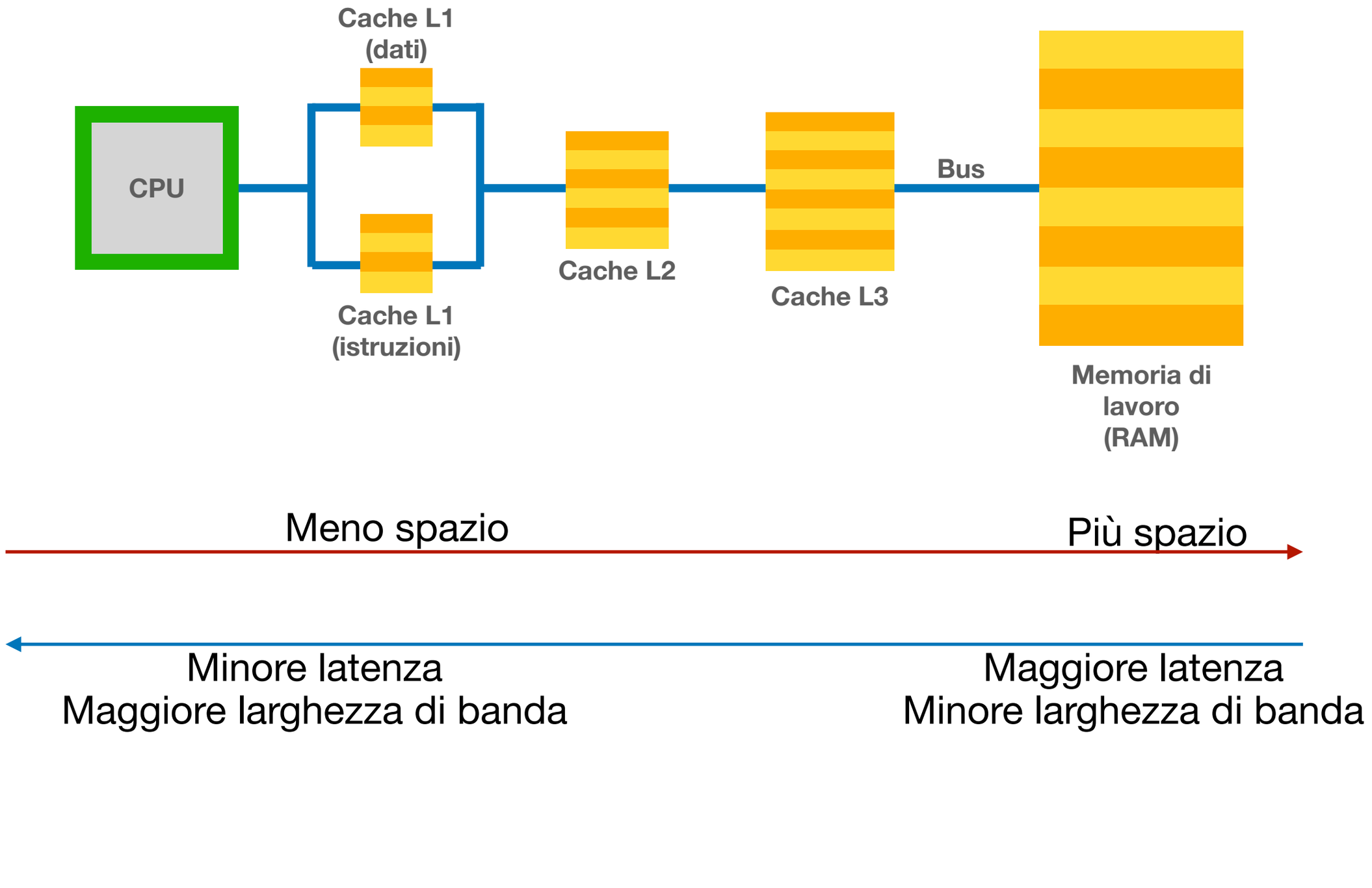
Struttura del calcolatore

Accesso alla memoria

- La memoria ha una latenza molto maggiore di “una istruzione”
- Se la CPU dovesse aspettare ogni volta che esegue una istruzione che accede alla memoria sarebbe lentissima
- C’è una serie di cache (L1, L2, L3 solitamente presenti) che hanno differenti tradeoff tra latenza, larghezza di banda e dimensioni
- I trasferimenti sono “a blocchi”, quindi accessi ad indirizzi vicini sono più rapidi

Struttura del calcolatore

Versione aggiornata



Sommiamo i valori sotto soglia

E risultati inattesi

42	56	3	34	27	89	75	63	99	12	42	55	51	32	23	12	1	7

Sommiamo tutti i valori minori di 50

1	3	7	12	12	23	27	32	34	42	42	51	55	56	63	75	89	99

Sommiamo tutti i valori minori di 50, ma il vettore è ordinato

Ci aspettiamo tempi di esecuzione differenti?

Struttura del calcolatore

Istruzioni multiple

- Le moderne CPU hanno più istruzioni in esecuzione contemporaneamente
- Se l'istruzioni successiva è nota posso (sotto certe condizioni) iniziare a eseguirla prima che quella attuale sia completata
- Ma se ho un salto condizionale?
- Faccio una previsione se il salto sarà preso o no e provo a eseguire l'istruzione più probabile (eventualmente annullando gli effetti se ho sbagliato)
- Più è facile la previsione più velocemente posso eseguire

Moltiplichiamo due matrici

Il fattore costante è importante

- Proviamo a moltiplicare due matrici con:
  - Codice Python
  - Codice C (con diverse opzioni del compilatore)
  - Codice da BLAS (una libreria) richiamata da Python
- Se anche i primi due algoritmi sono uguali in termini asintotici, ovvero $O(n^3)$, il fattore costante è importante
- La velocità deriva dall'algoritmo giusto (essenziale) e dall'implementazione che faccia uso della caratteristiche della macchina