Memoria e istruzione di assegnamento

Corso di Programmazione 1A, 2021-22
Felice Cardone

Struttura (semplificata) di un calcolatore

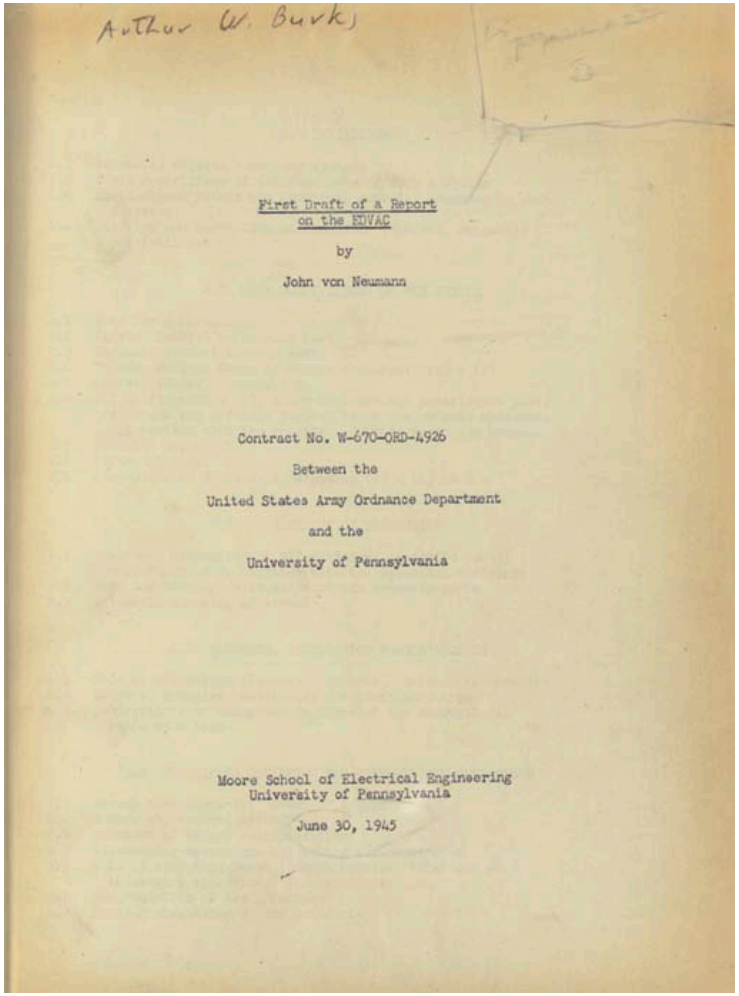
Si tratta della architettura suggerita da John von Neumann, contenuta nel documento:

First Draft of a Report on the EDVAC,

del 30 Giugno 1945

Per questo lo schema architetturale che viene descritto si chiama:

architettura di von Neumann



Arthur W. Burks

First Draft of a Report
on the EDVAC

by

John von Neumann

Contract No. W-670-ORD-4926

Between the

United States Army Ordnance Department

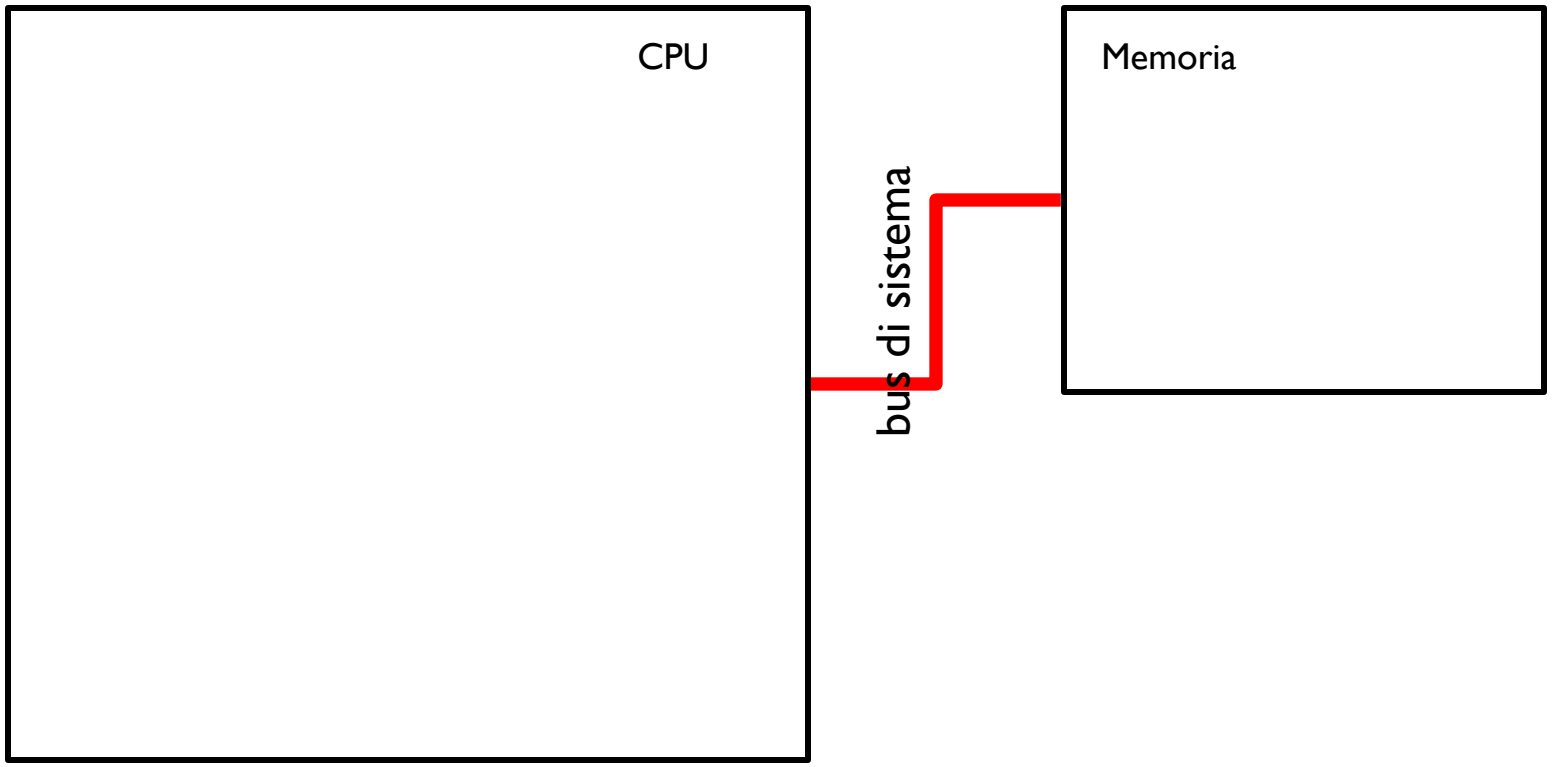
and the

University of Pennsylvania

Moore School of Electrical Engineering
University of Pennsylvania

June 30, 1945

Struttura (semplificata) di un calcolatore



Critica della architettura di von Neumann: Backus 1978

Can Programming Be Liberated from the von Neumann Style? A Functional Style and Its Algebra of Programs

John Backus
IBM Research Laboratory, San Jose

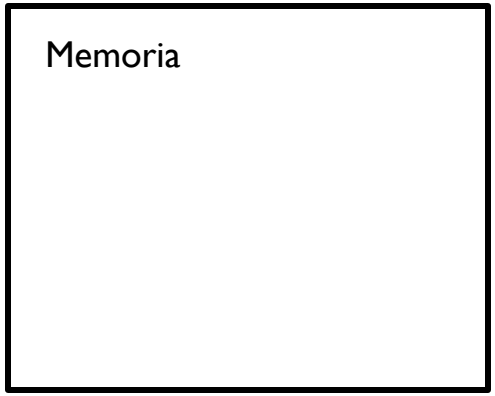
❝ In its simplest form a von Neumann computer has three parts:

- a central processing unit (or CPU),
- a store, and
- a connecting tube that can transmit a single word between the CPU and the store (and send an address to the store).

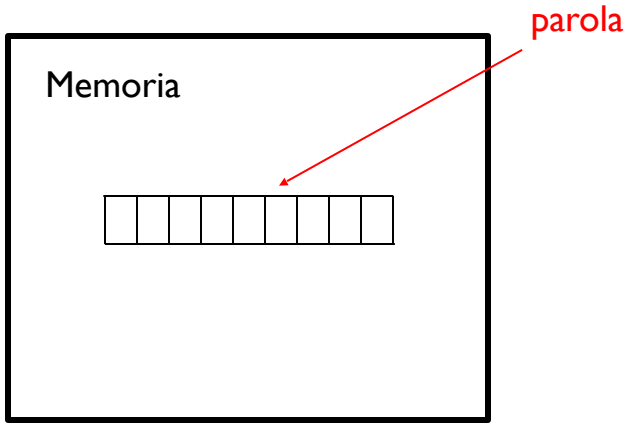
I propose to call this tube the von Neumann bottleneck. The task of a program is to change the contents of the store in some major way; when one considers that this task must be accomplished entirely by pumping single words back and forth through the von Neumann bottleneck, the reason for its name becomes clear.

Ironically, a large part of the traffic in the bottleneck is not useful data but merely names of data, as well as operations and data used only to compute such names. ❞

Struttura (semplificata) di un calcolatore: memoria

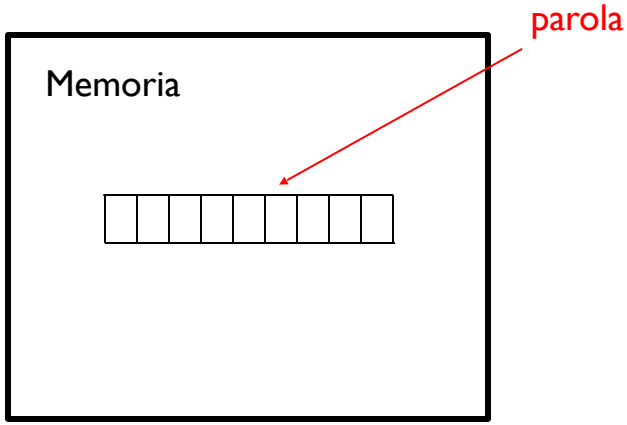


Struttura (semplificata) di un calcolatore: memoria



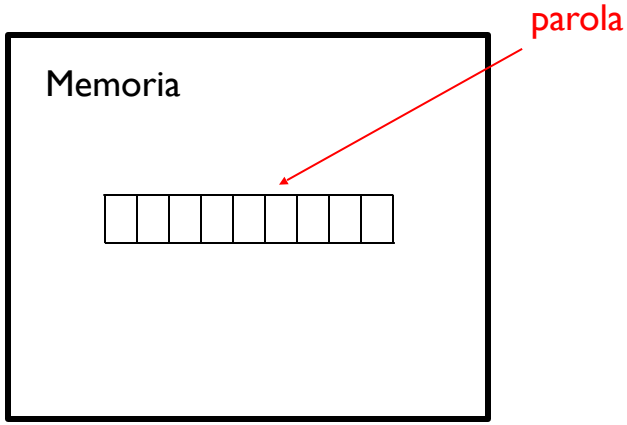
Struttura (semplificata) di un calcolatore: memoria

BOUILLON
CHARTIER
· PARIS ·

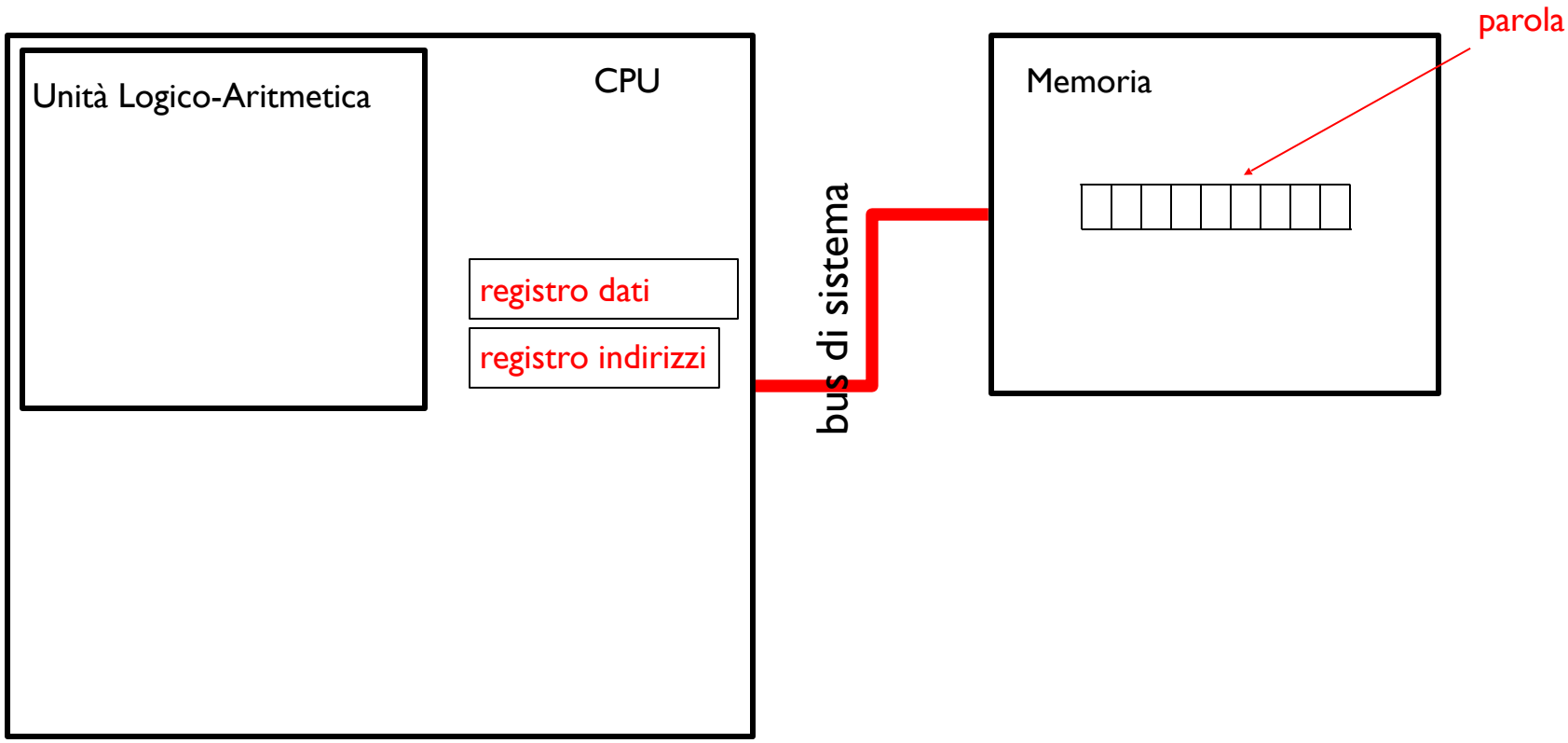


Struttura (semplificata) di un calcolatore: memoria

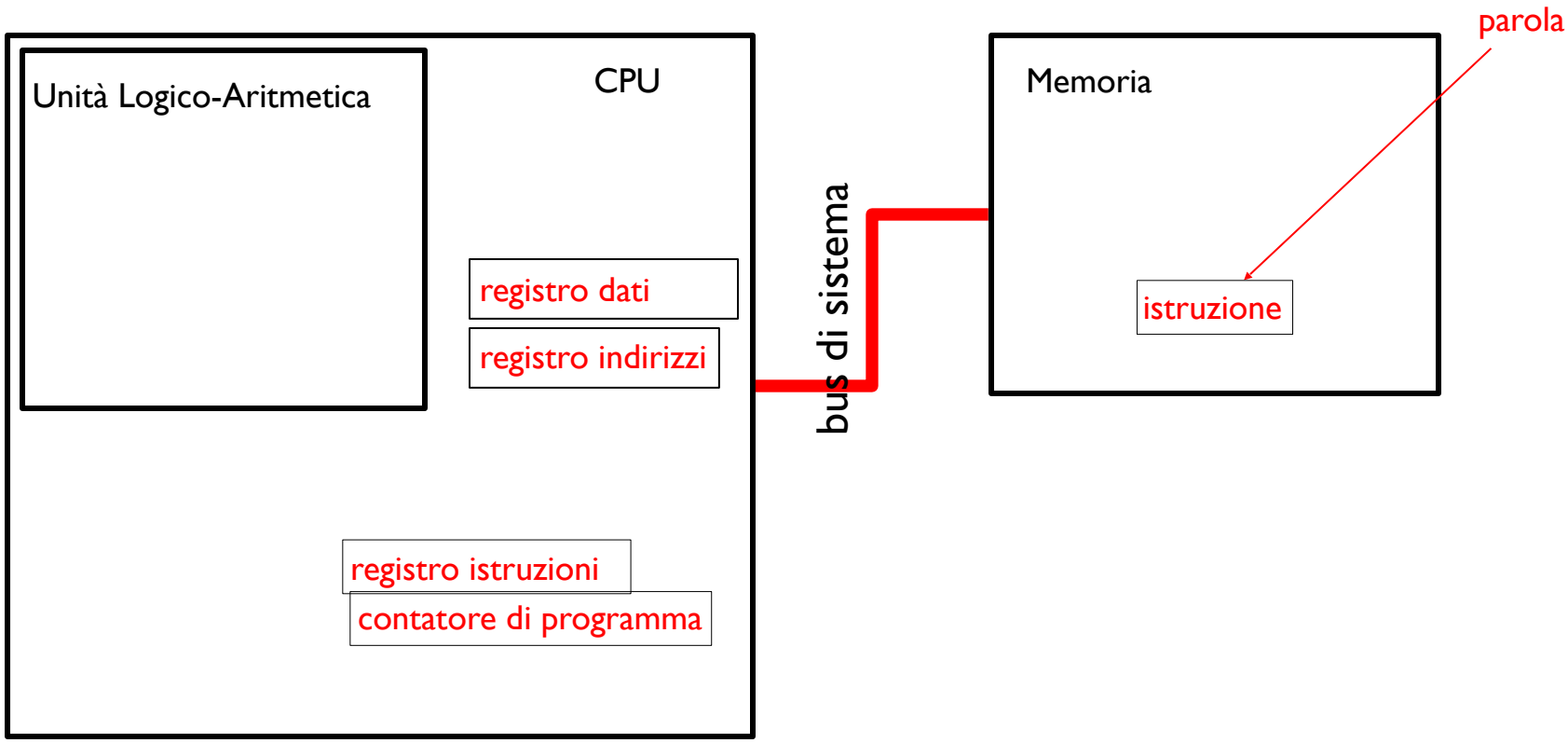
BOUILLON
CHARTIER
· PARIS ·



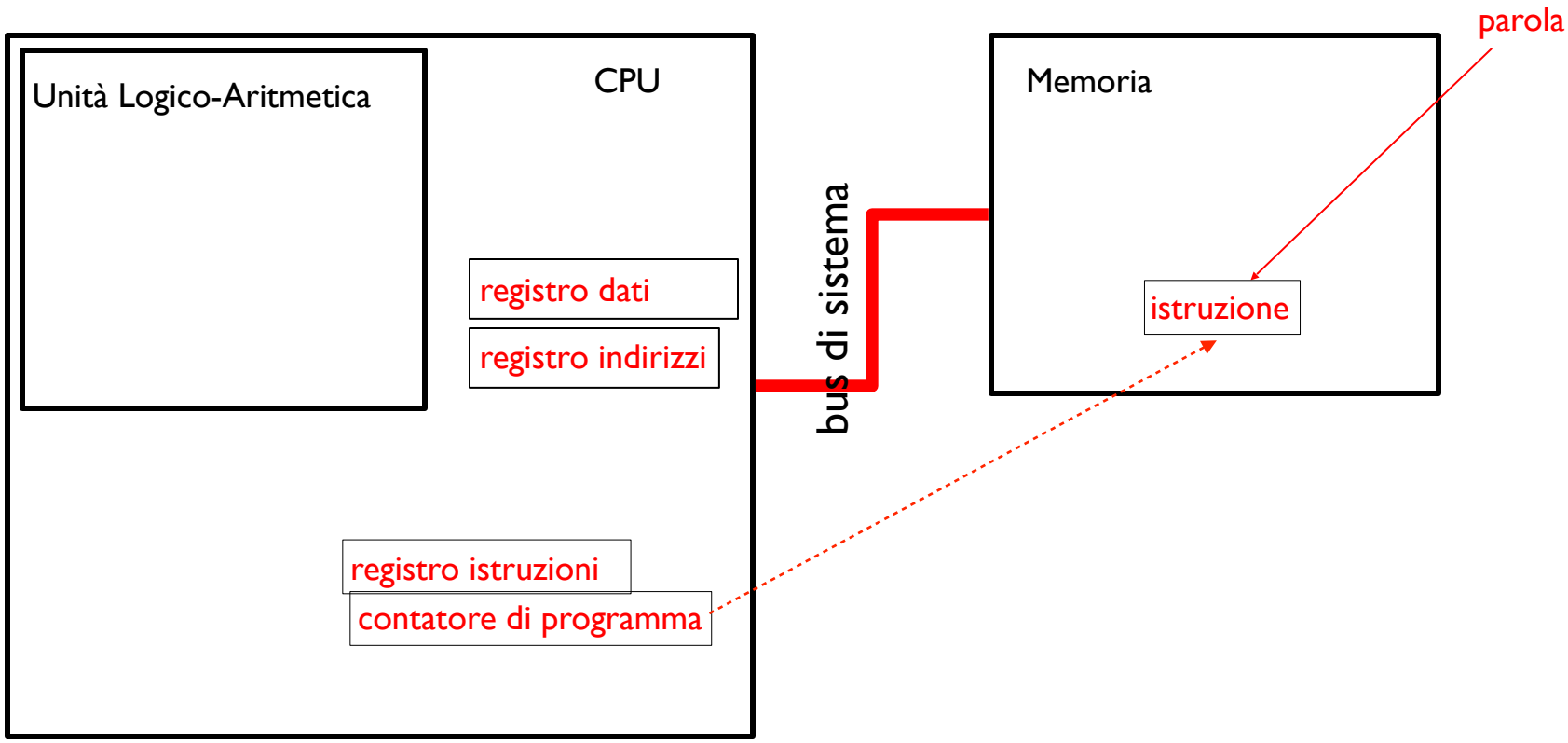
Struttura (semplificata) di un calcolatore



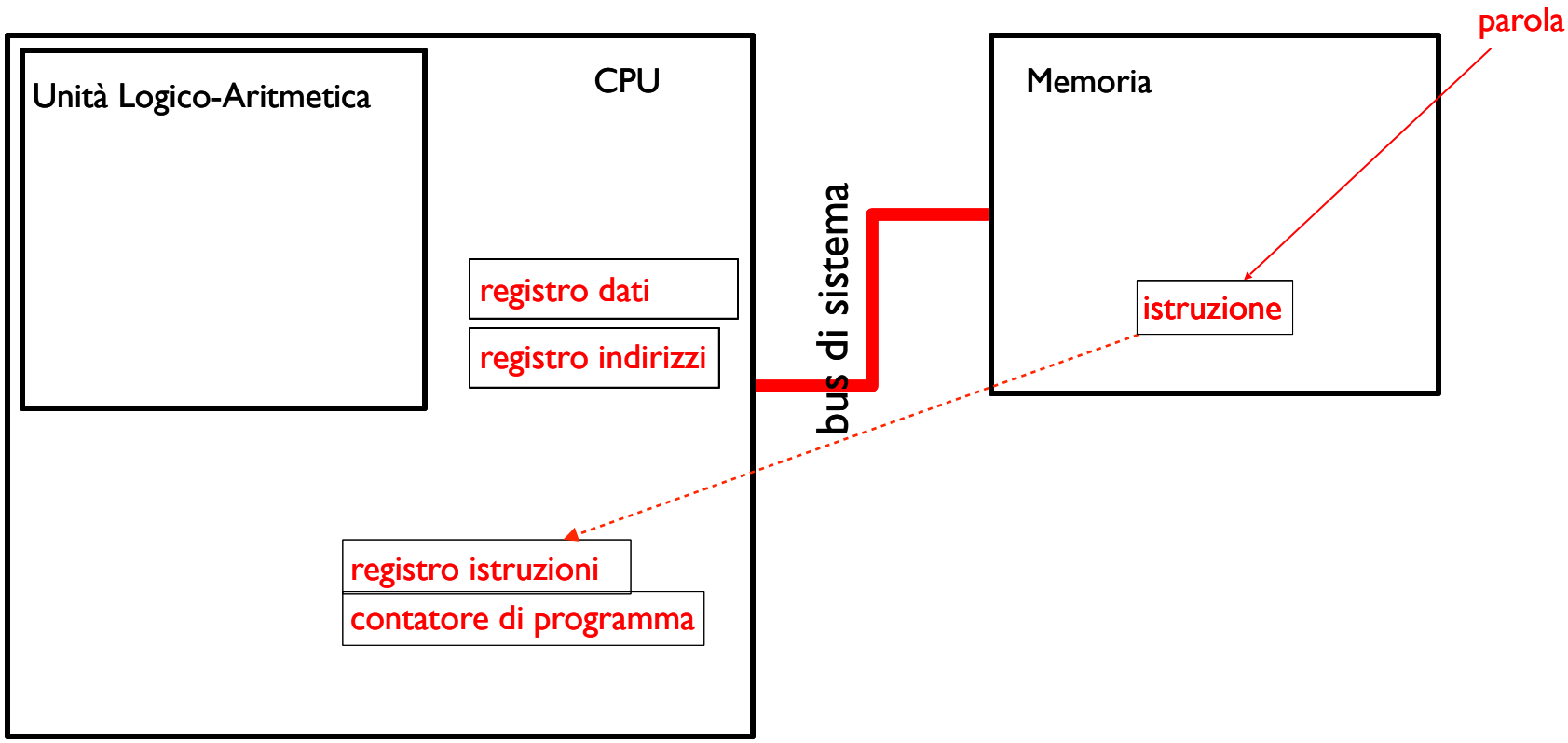
Struttura (semplificata) di un calcolatore



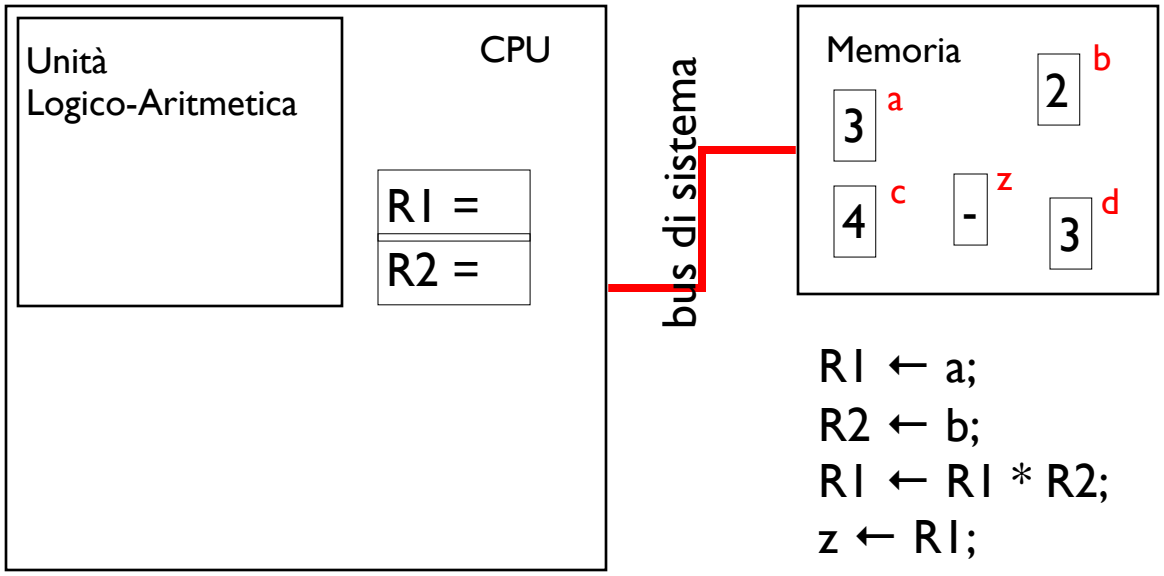
Il ciclo fetch-decode-execute



Il ciclo fetch-decode-execute

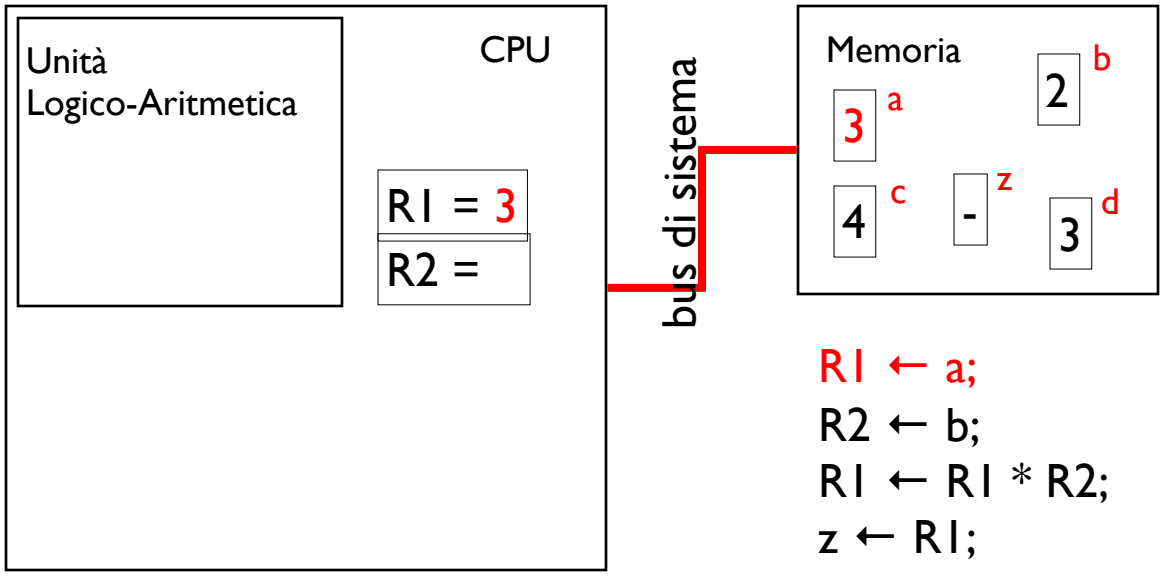


Esempio: valutazione dell'espressione 3 * 2 + 4 * 3



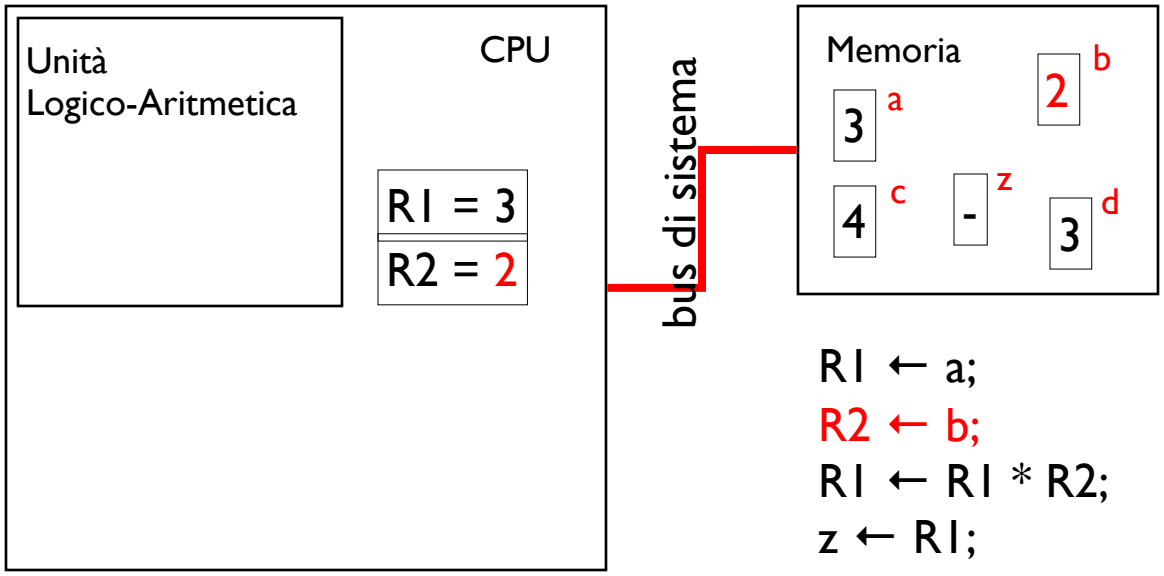
R1 ← a;
R2 ← b;
R1 ← R1 * R2;
z ← R1;
R1 ← c;
R2 ← d;
R1 ← R1 * R2;
R2 ← z;
R1 ← R1 + R2

Esempio: valutazione dell'espressione 3 * 2 + 4 * 3



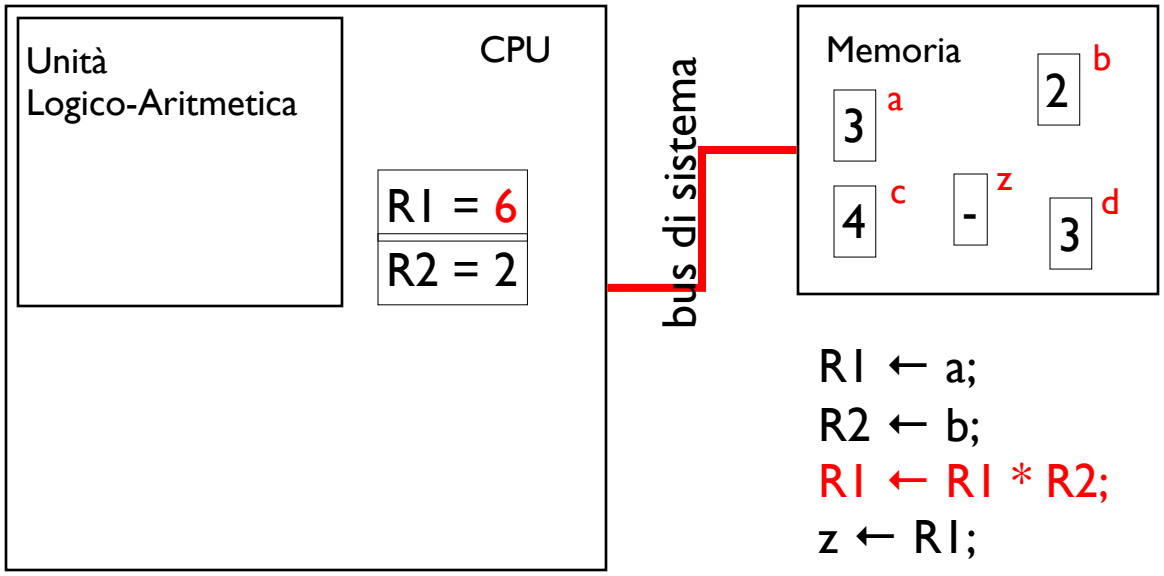
R1 ← a;
R2 ← b;
R1 ← R1 * R2;
z ← R1;
R1 ← c;
R2 ← d;
R1 ← R1 * R2;
R2 ← z;
R1 ← R1 + R2

Esempio: valutazione dell'espressione 3 * 2 + 4 * 3



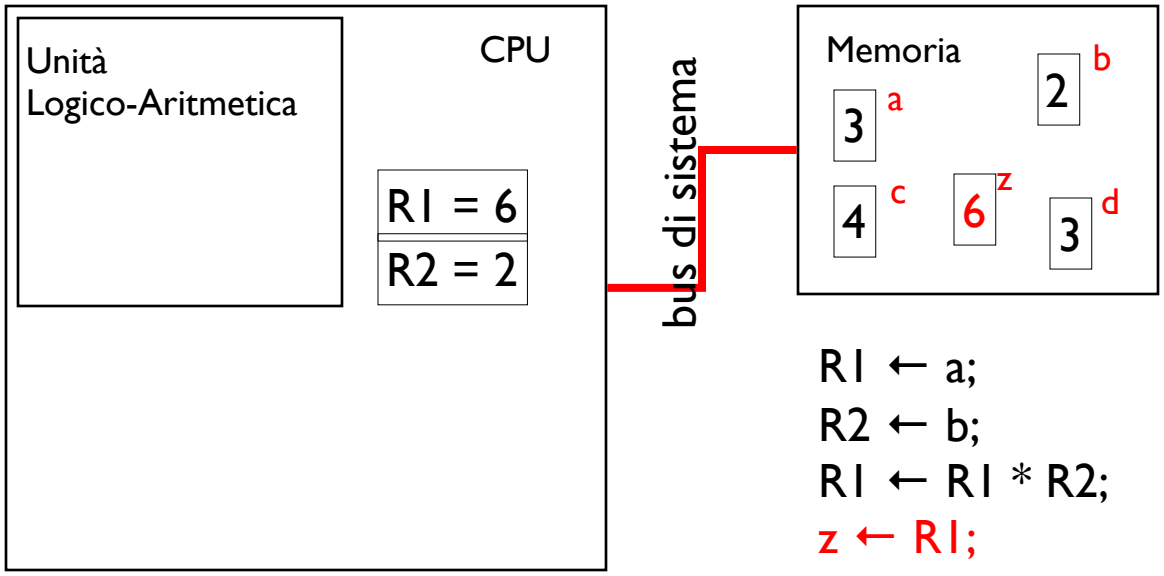
R1 ← a;
R2 ← b;
R1 ← R1 * R2;
z ← R1;
R1 ← c;
R2 ← d;
R1 ← R1 * R2;
R2 ← z;
R1 ← R1 + R2

Esempio: valutazione dell'espressione 3 * 2 + 4 * 3



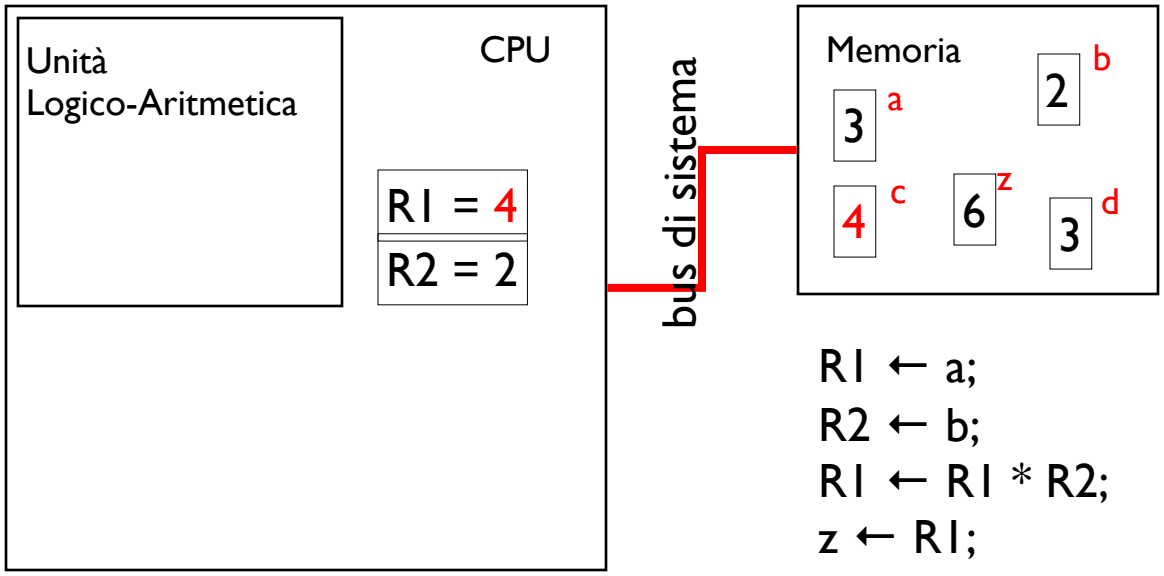
R1 ← a;
R2 ← b;
R1 ← R1 * R2;
z ← R1;
R1 ← c;
R2 ← d;
R1 ← R1 * R2;
R2 ← z;
R1 ← R1 + R2

Esempio: valutazione dell'espressione 3 * 2 + 4 * 3



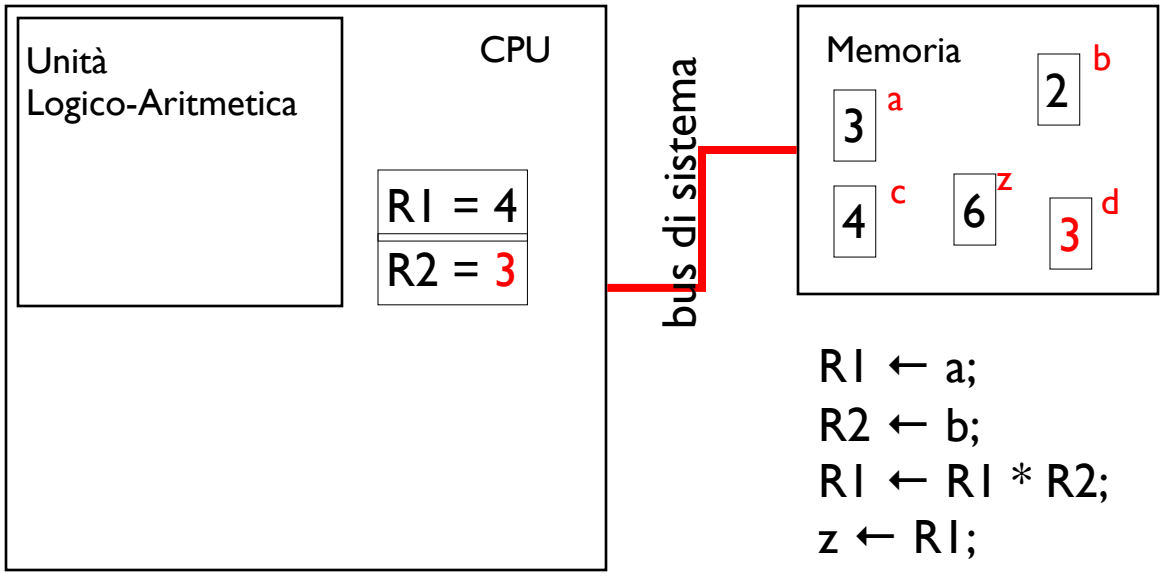
R1 ← a;
R2 ← b;
R1 ← R1 * R2;
z ← R1;
R1 ← c;
R2 ← d;
R1 ← R1 * R2;
R2 ← z;
R1 ← R1 + R2

Esempio: valutazione dell'espressione 3 * 2 + 4 * 3



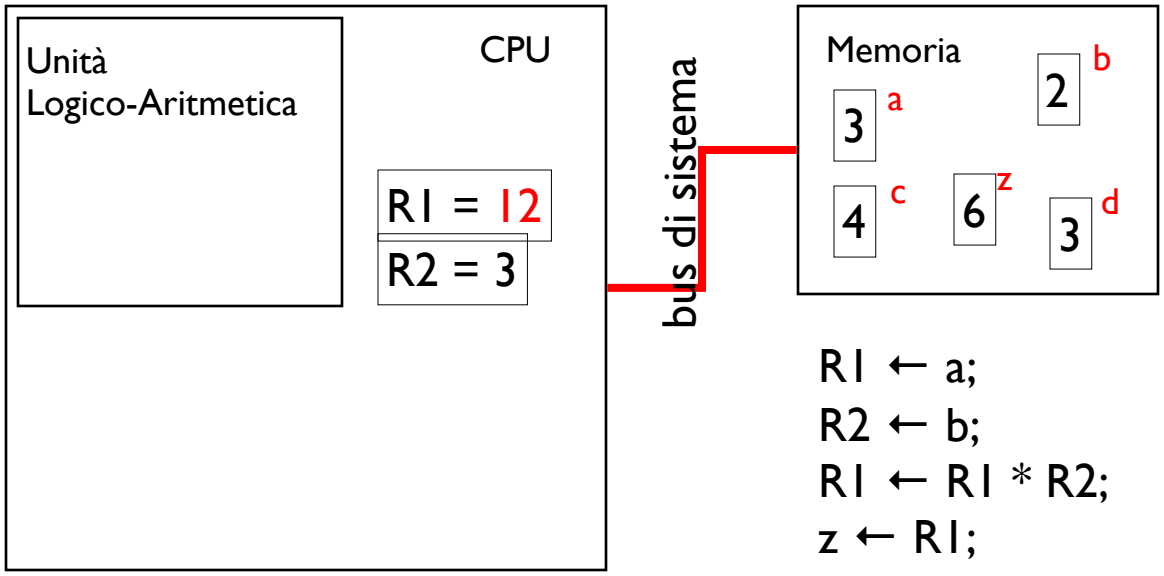
R1 ← a;
R2 ← b;
R1 ← R1 * R2;
z ← R1;
R1 ← c;
R2 ← d;
R1 ← R1 * R2;
R2 ← z;
R1 ← R1 + R2

Esempio: valutazione dell'espressione 3 * 2 + 4 * 3



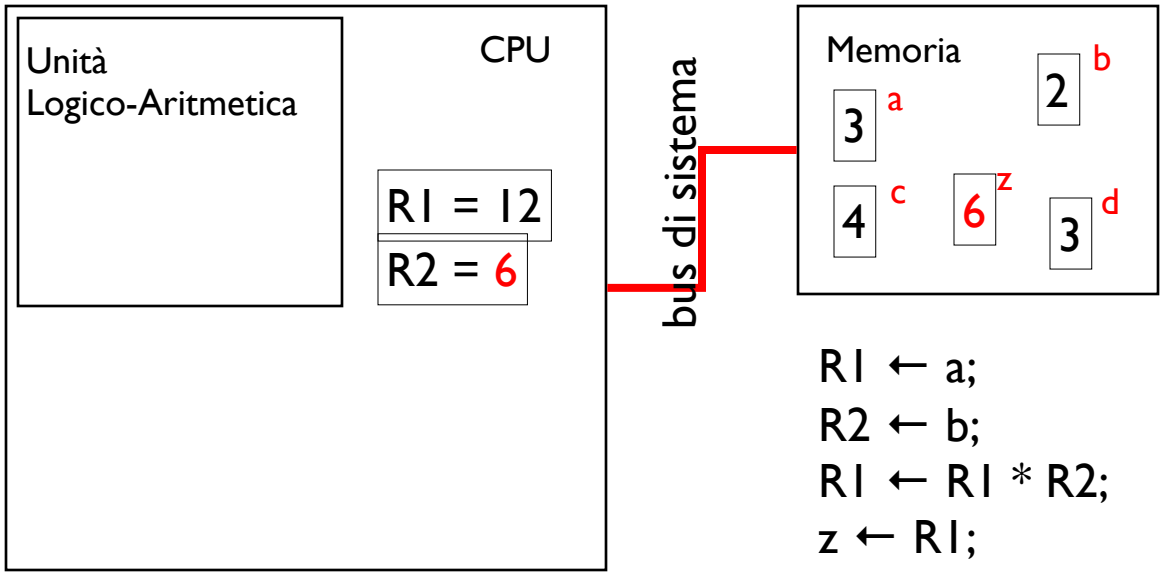
R1 ← a;
R2 ← b;
R1 ← R1 * R2;
z ← R1;
R1 ← c;
R2 ← d;
R1 ← R1 * R2;
R2 ← z;
R1 ← R1 + R2

Esempio: valutazione dell'espressione 3 * 2 + 4 * 3



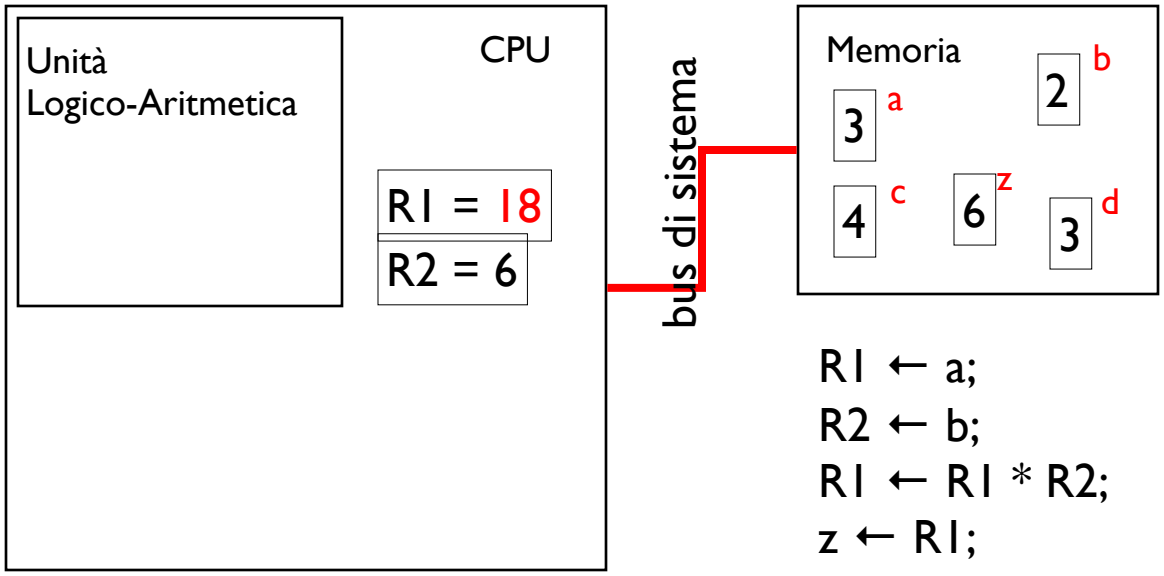

```
R1 ← a;
R2 ← b;
R1 ← R1 * R2;
z ← R1;
R1 ← c;
R2 ← d;
R1 ← R1 * R2;
R2 ← z;
R1 ← R1 + R2
```

Esempio: valutazione dell'espressione 3 * 2 + 4 * 3



R1 ← a;
R2 ← b;
R1 ← R1 * R2;
z ← R1;
R1 ← c;
R2 ← d;
R1 ← R1 * R2;
R2 ← z;
R1 ← R1 + R2

Esempio: valutazione dell'espressione 3 * 2 + 4 * 3



R1 ← a;
R2 ← b;
R1 ← R1 * R2;
z ← R1;
R1 ← c;
R2 ← d;
R1 ← R1 * R2;
R2 ← z;
R1 ← R1 + R2

Struttura (semplificata) di un programma

```
class <nome del programma> {
    public static void main (String[] args){
        <dichiarazioni di variabili>
        <istruzioni>
    }
}
```

Prima di essere utilizzata in un programma, una variabile deve essere dichiarata;

una dichiarazione di variabile può seguire, nel testo del programma, delle istruzioni

Esempio

```
class Saluto {
    public static void main (String[] args){
        System.out.println ("Ciao.");
        }
}
```

Questo programma è il contenuto di un file di testo chiamato

`Saluto.java`

compilato con il comando

`javac Saluto.java`

per generare il file bytecode `Saluto.class`

che viene mandato in esecuzione con il comando

`java Saluto`

Dichiarazioni di variabili: sintassi

Le variabili sono dichiarate assieme al loro tipo, per esempio

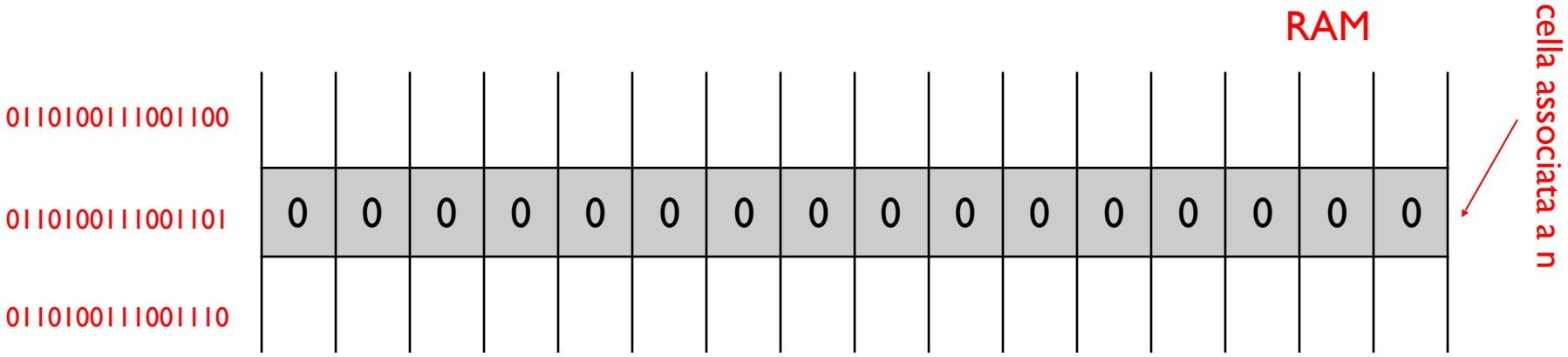
```
int n;

int i,j;   equivalente a int i; int j;

float pi;
```

Dal punto di vista sintattico, le variabili sono identificatori, sequenze di caratteri alfanumerici che non iniziano con una cifra.

Alcuni tipi primitivi	**Valori**
boolean	true, false
char	caratteri UNICODE
int	interi 32 bit
float	numeri in virgola mobile, 32 bit
double	numeri in virgola mobile, 64 bit

Dichiarazioni di variabili: semantica

```
class Esempio {
public static void main (String[] args){
      int n;
      double pi;
…     }
}
```



Dichiarazioni di variabili: semantica

Al momento della dichiarazione. ad una variabile viene associata una cella di memoria (o un gruppo di celle)

I tipi primitivi permettono al compilatore di calcolare quante celle di memoria allocare ad una variabile

Il valore di una variabile dichiarata ma non esplicitamente inizializzata deve essere considerato indefinito

Dichiarazioni di variabili: semantica

Al momento della dichiarazione. ad una variabile viene associata una cella di memoria (o un gruppo di celle)

I tipi primitivi permettono al compilatore di calcolare quante celle di memoria allocare ad una variabile

Il valore di una variabile dichiarata ma non esplicitamente inizializzata deve essere considerato indefinito

Morale: non usare variabili senza dichiararle ed inizializzarle.

Variabili: riepilogo

Una variabile è caratterizzata da:

1. un nome (che è un identificatore);
2. una dimensione (determinata dal tipo con il quale è stata dichiarata);
3. un valore (del tipo della variabile).

Su una variabile sono disponibili le seguenti operazioni:

1. dichiarazione
2. lettura
3. scrittura

La dichiarazione introduce un nome ed un tipo (e, di conseguenza, una dimensione) per la variabile;

La scrittura associa un valore alla variabile;

La lettura usa (senza modificare) il valore della variabile.

Istruzione di assegnamento

Formato generale

<variabile> = <espressione>;

dove <espressione> ha lo stesso tipo di <variabile>

Esecuzione di un'istruzione di assegnamento x = E

1. si valuta l'espressione E; se la valutazione termina si ottiene un valore v;
2. il valore v viene scritto nella cella di memoria (eventualmente più di una) associata alla variabile x al momento della sua dichiarazione.

Osservazione

Sulla sinistra di un assegnamento, una variabile indica una cella di memoria;

sulla destra di un assegnamento, una variabile rappresenta il valore scritto (al momento della valutazione) nella cella associata.

Espressioni di tipo primitivo (anteprima)

Espressioni di tipo numerico

se E,E' sono espressioni di tipo int, anche

E + E'	(somma)
E * E'	(prodotto)
E / E'	(divisione)
E % E'	(resto della divisione)

sono espressioni di tipo int.

Espressioni di tipo boolean

se E,E' sono espressioni di tipo int, allora

E = E'	==
E > E'	>
E < E'	<
E ! E'	!=
E > E'	>=
E < E'	<=

sono espressioni di tipo boolean.

Semantica delle istruzioni di assegnamento: esempio

Per eseguire l'istruzione

x = (-3 + 5) * 4;

avendo dichiarato int x:

(1) Valutare l'espressione (-3 + 5) * 4:
valutare l'espressione -3 + 5
valutare l'espressione -3,
valutare l'espressione 3, ottenendo valore 3
ottenendo valore -3
valutare l'espressione 5, ottenendo valore 5
ottenendo valore 2
valutare l'espressione 4, ottenendo valore 4
ottenendo valore 8

(2) scrivere il valore 8 nella cella di memoria associata a x al momento della sua dichiarazione.

Semantica delle istruzioni di assegnamento: esempio

Per valutare l'espressione

```
x = (-3 + 5) * 4;
```

avendo dichiarato int x:

(1) Valutare l'espressione (-3 + 5) * 4:
valutare l'espressione -3 + 5
valutare l'espressione -3,
valutare l'espressione 3, ottenendo valore 3
ottenendo valore -3
valutare l'espressione 5, ottenendo valore 5
ottenendo valore 2
valutare l'espressione 4, ottenendo valore 4
ottenendo valore 8

(2) scrivere il valore 8 nella cella di memoria associata a x al momento della sua dichiarazione.

Una notazione per la valutazione di espressioni

$$\dfrac{\dfrac{\dfrac{3 \rightarrow 3}{\quad} \qquad \dfrac{-3 \rightarrow -3 \qquad 5 \rightarrow 5}{\quad}}{(-3 + 5) \rightarrow 2} \qquad \dfrac{}{4 \rightarrow 4}}{(-3 + 5) * 4 \rightarrow 8}$$

Una notazione per la valutazione di espressioni

$$\frac{\dfrac{3 \rightarrow 3 \qquad \dfrac{-3 \rightarrow -3 \qquad 5 \rightarrow 5}{(-3 + 5) \rightarrow 2} \qquad 4 \rightarrow 4}{(-3 + 5) * 4 \rightarrow 8}$$

Strutture di controllo