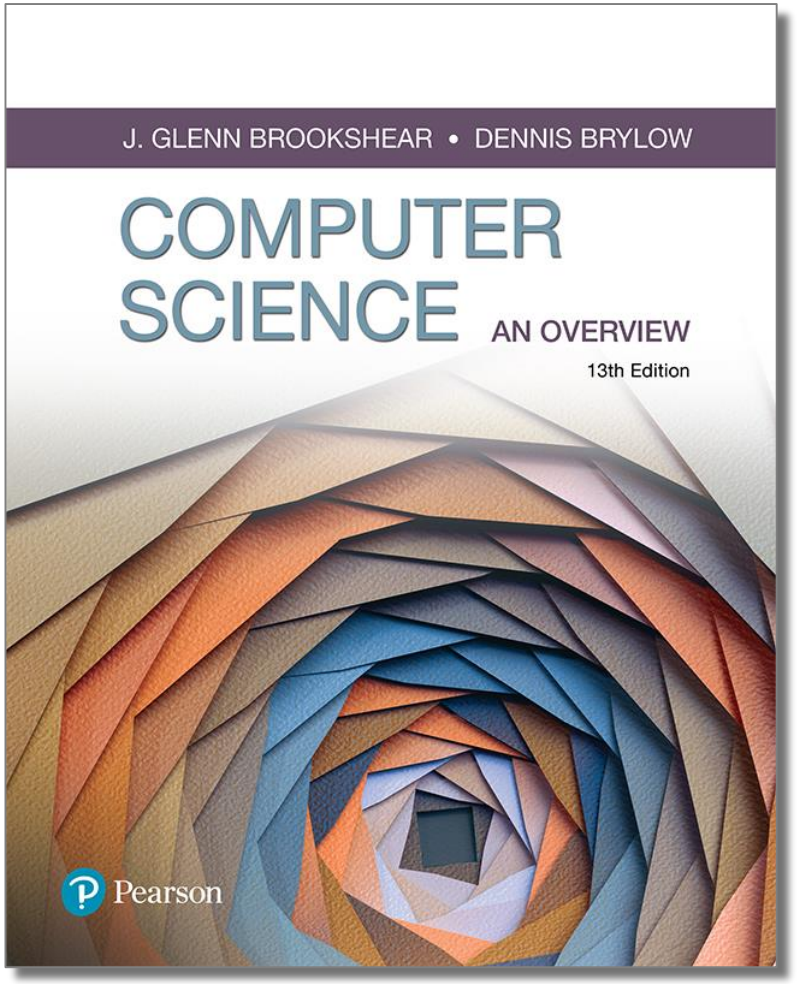
Computer Science An Overview

13th Edition



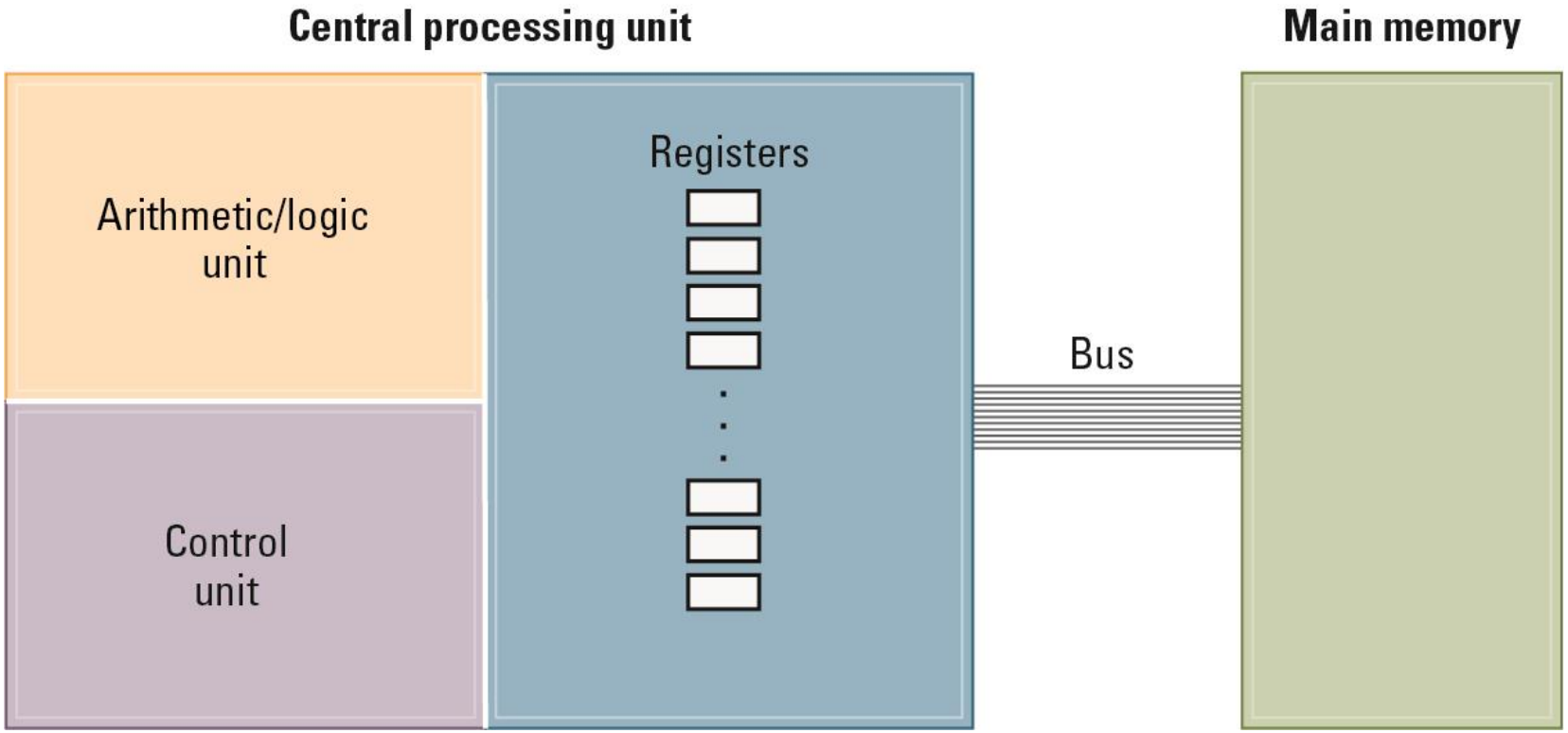
Chapter 2

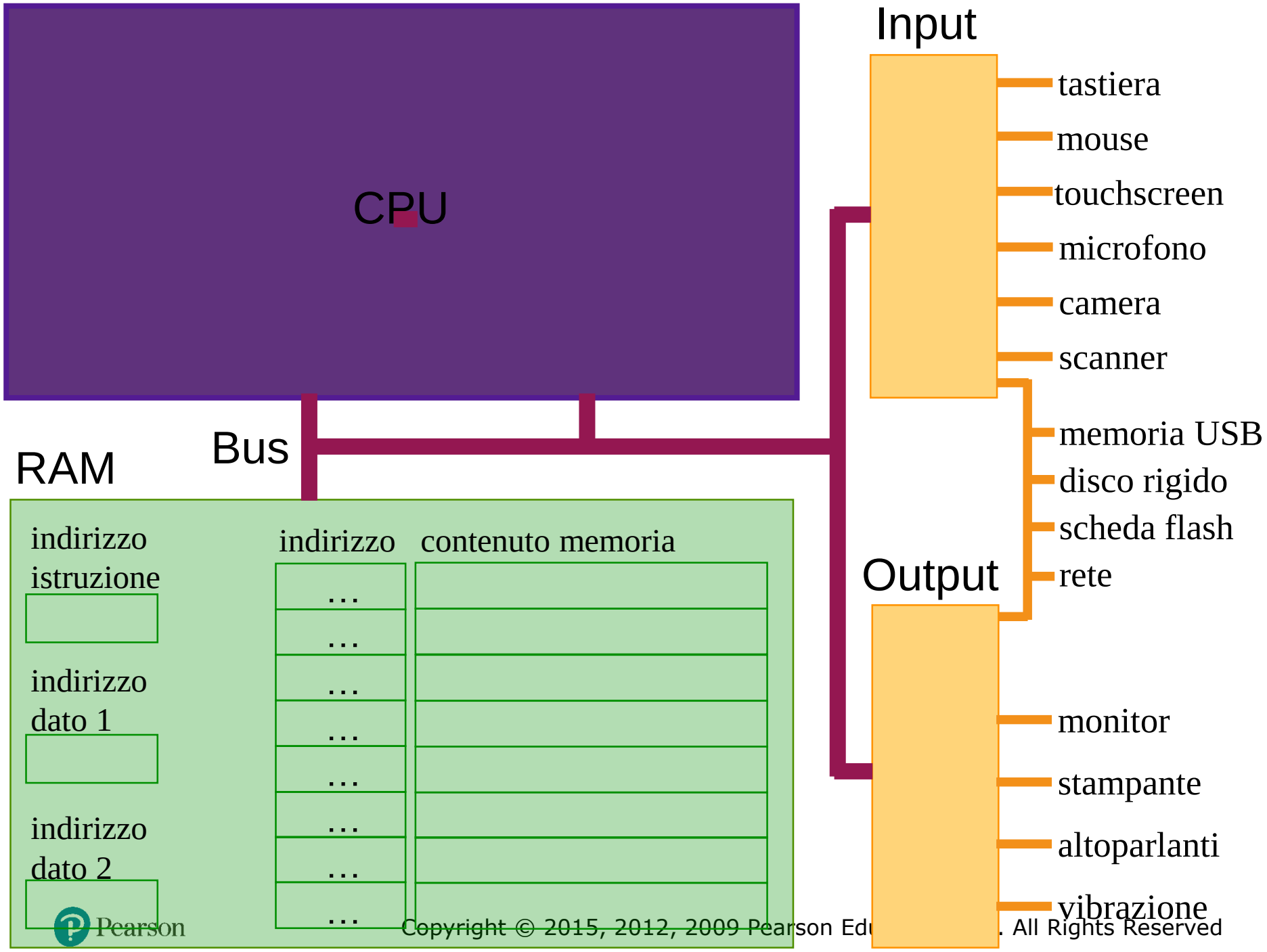
Data Manipulation

Architettura del computer

- Central Processing Unit (CPU)
  - Arithmetic/Logic Unit
  - Control Unit
  - Register Unit
    - General purpose registers
    - Special purpose registers
- Bus
- Main Memory

Diagramma a blocchi





Stored Program Concept

Un programma può essere codificato come sequenza di bit e caricato nella memoria principale.

Dalla memoria principale la Control Unit può estrarre, decodificare ed eseguire istruzioni.

Linguaggio macchina

- **Machine instruction:** istruzione codificata come sequenza di bit e riconoscibile dalla CPU
- **Machine language:** insieme di tutte le istruzioni riconoscibili da una macchina

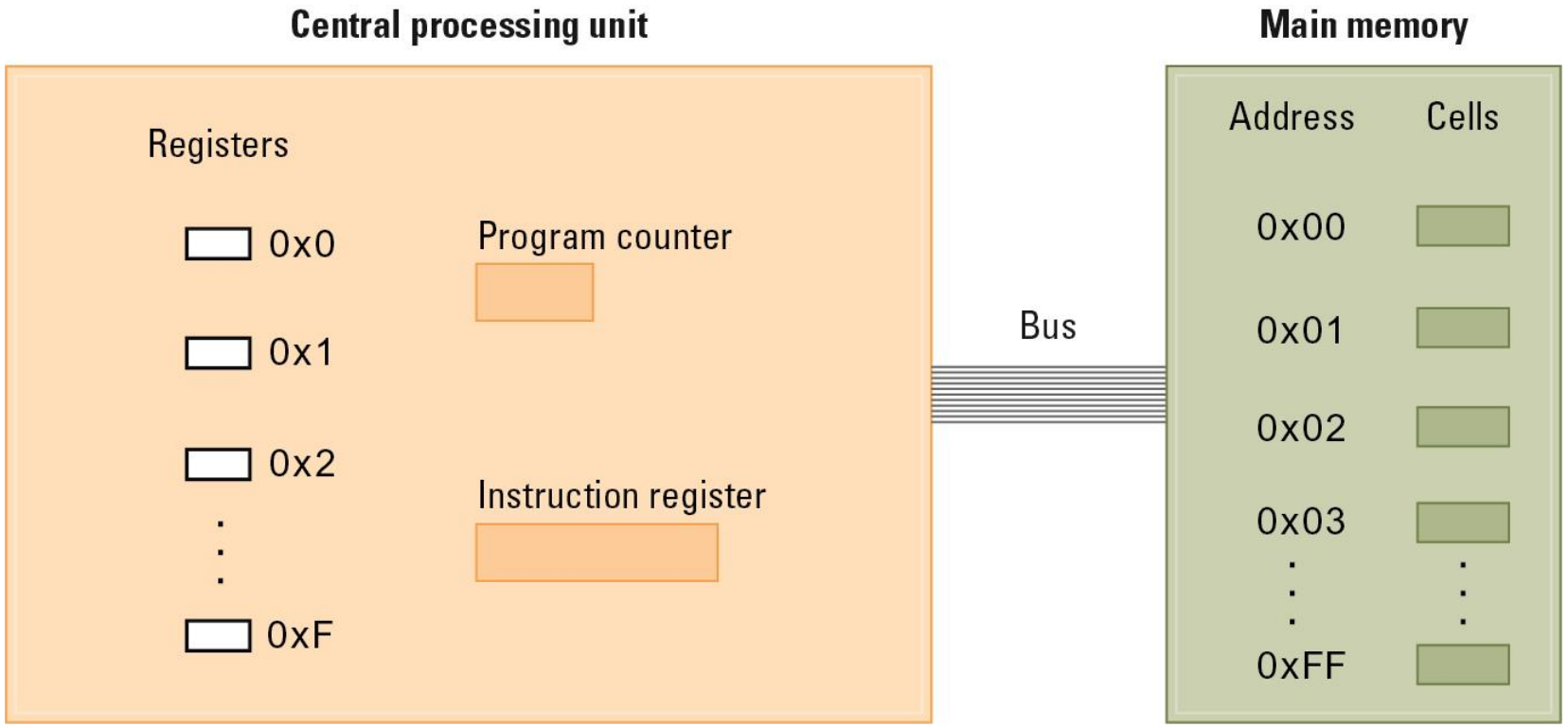
Machine Language

- Reduced Instruction Set Computing (RISC)
  - Poche, semplici, efficienti e veloci istruzioni
  - Ex.: PowerPC from Apple/IBM/Motorola and ARM
- Complex Instruction Set Computing (CISC)
  - Molte, complesse e potenti istruzioni
  - Ex.: Intel

Machine Instruction

- Data Transfer: copia dati da una locazione ad un'altra (e.g. LOAD, STORE)
- Arithmetic/Logic: operazioni sulle sequenze di bit (e.g. +, -, *, /, AND, OR, SHIFT, ROTATE)
- Control: dirige l'esecuzione del programma (e.g. JUMP, BRANCH)

Esempio di architettura



Istruzione macchina: elementi

- **Op-code:** specifica l'operazione da eseguire
- **Operand:** fornisce informazioni più dettagliate in riferimento all'operazione
  - L'interpretazione degli operandi varia in base all'op-code.

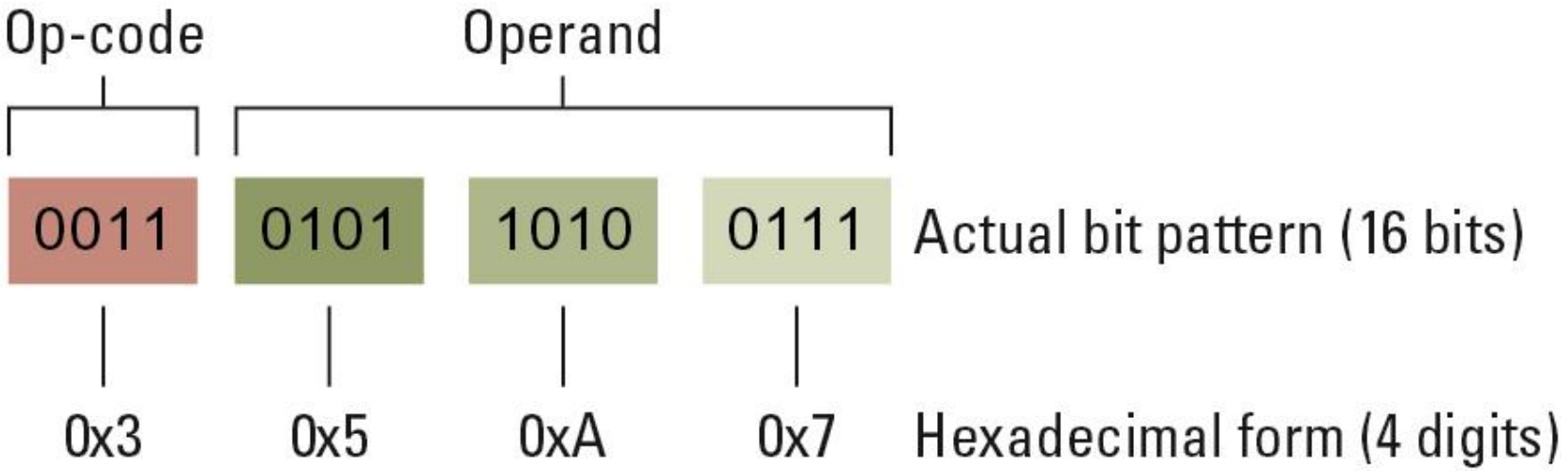
Esecuzione di un programma

- Controllata da:
  - Instruction register
    - Mantiene l'istruzione corrente
  - Program counter
    - Mantiene l'indirizzo della prossima istruzione

- Ciclo macchina: (ripete 3 step)
  - Fetch, Decode, Execute

Linguaggio macchina Vole

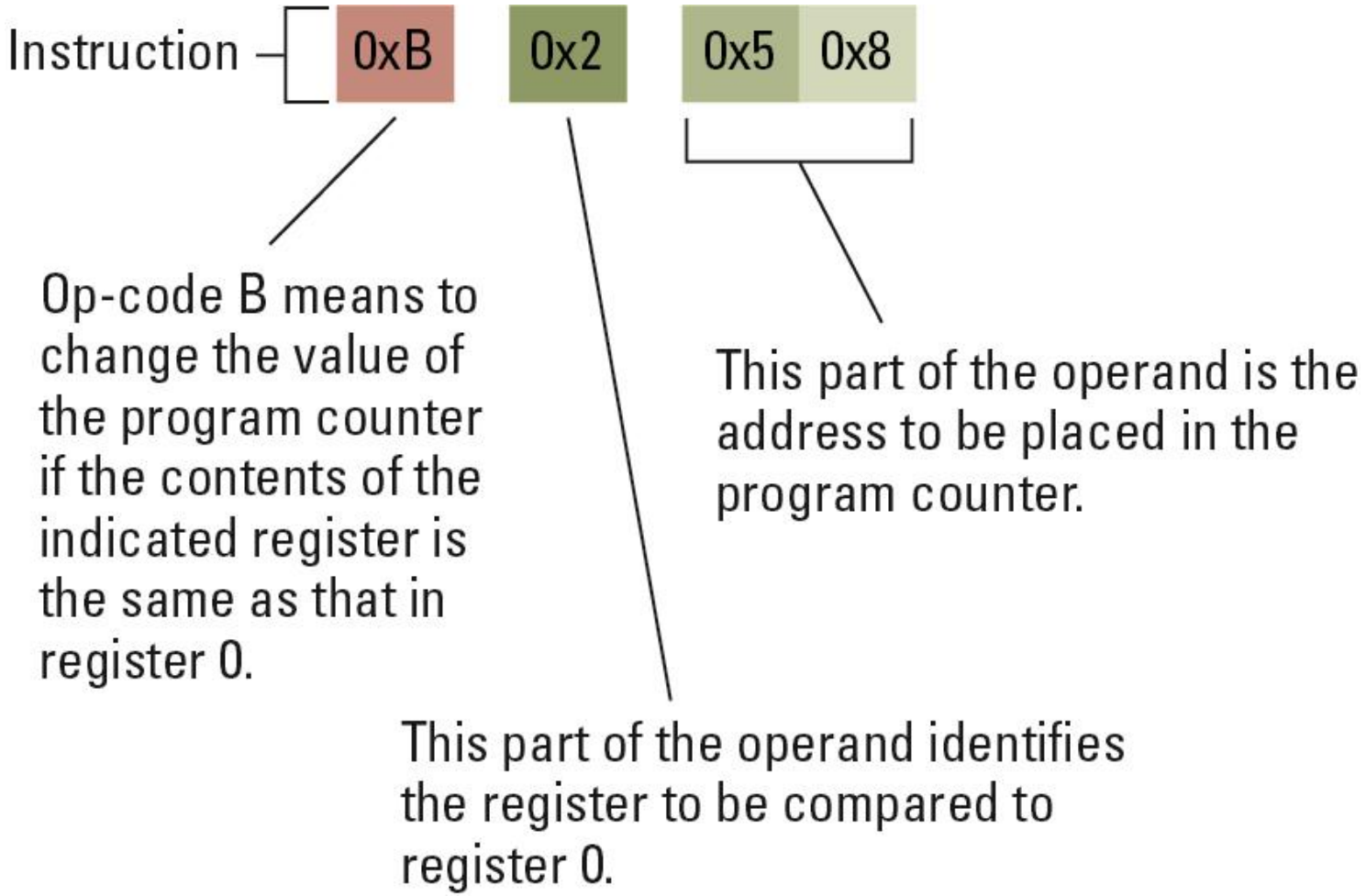
Istruzioni macchina: 2 byte



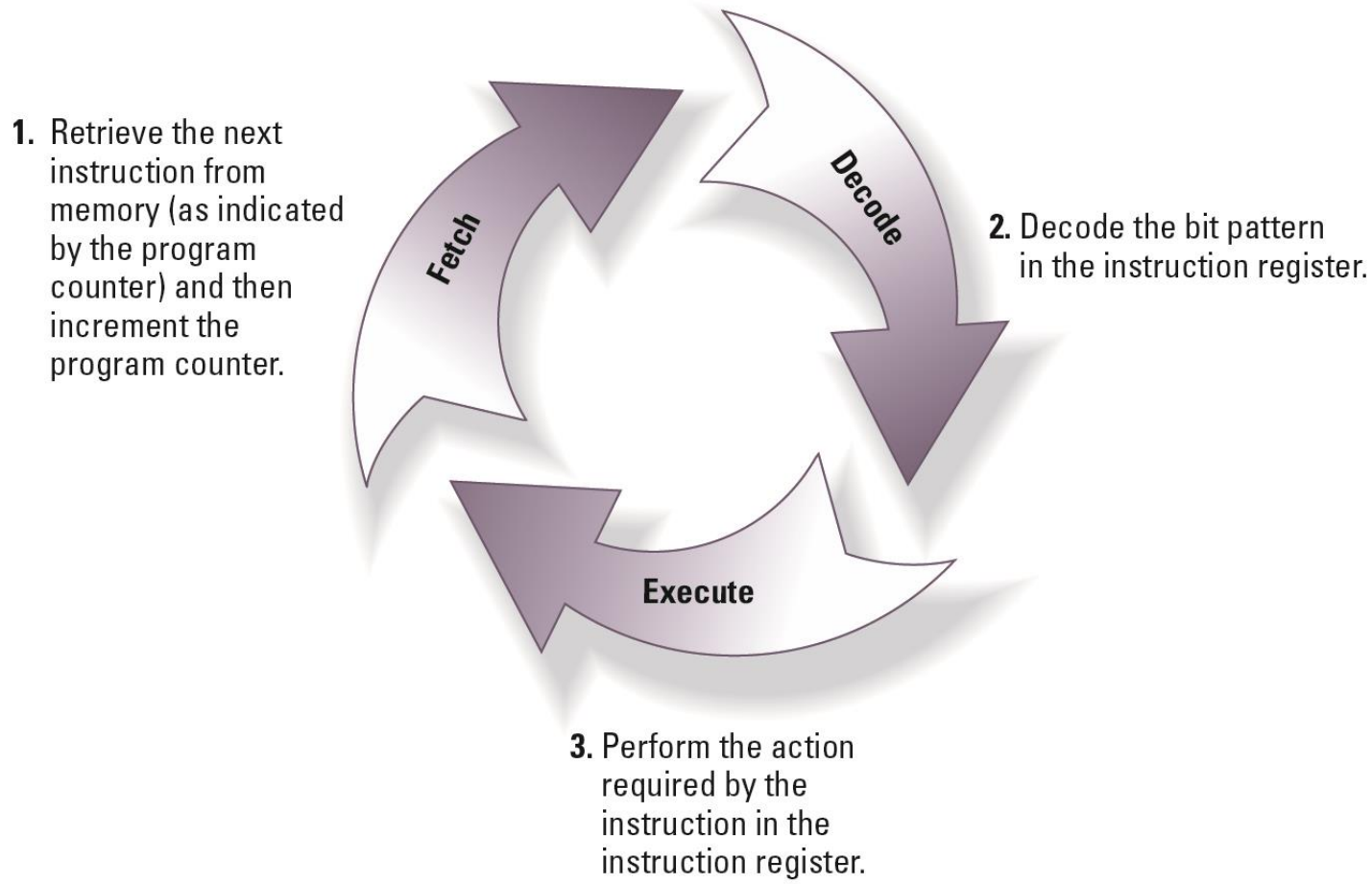
VOLE Instructions

Op-code	Operand	Description
1	RXY	LOAD the register R with the bit pattern found in the memory cell whose address is XY. Example: 14A3 would cause the contents of the memory cell located at address A3 to be placed in register 4.
2	RXY	LOAD the register R with the bit pattern XY. Example: 20A3 would cause the value A3 to be placed in register 0.
3	RXY	STORE the bit pattern found in register R in the memory cell whose address is XY. Example: 35B1 would cause the contents of register 5 to be placed in the memory cell whose address is B1.
4	0RS	MOVE the bit pattern found in register R to register S. Example: 40A4 would cause the contents of register A to be copied into register 4.
5	RST	ADD the bit patterns in registers S and T as though they were two's complement representations and leave the result in register R. Example: 5726 would cause the binary values in registers 2 and 6 to be added and the sum placed in register 7.
6	RST	ADD the bit patterns in registers S and T as though they represented values in floating-point notation and leave the floating-point result in register R. Example: 634E would cause the values in registers 4 and E to be added as floating-point values and the result to be placed in register 3.
7	RST	OR the bit patterns in registers S and T and place the result in register R. Example: 7CB4 would cause the result of ORing the contents of registers B and 4 to be placed in register C.
8	RST	AND the bit patterns in register S and T and place the result in register R. Example: 8045 would cause the result of ANDing the contents of registers 4 and 5 to be placed in register 0.
9	RST	EXCLUSIVE OR the bit patterns in registers S and T and place the result in register R. Example: 95F3 would cause the result of EXCLUSIVE ORing the contents of registers F and 3 to be placed in register 5.
A	R0X	ROTATE the bit pattern in register R one bit to the right X times. Each time place the bit that started at the low-order end at the high-order end. Example: A403 would cause the contents of register 4 to be rotated 3 bits to the right in a circular fashion.
B	RXY	JUMP to the instruction located in the memory cell at address XY if the bit pattern in register R is equal to the bit pattern in register number 0. Otherwise, continue with the normal sequence of execution. (The jump is implemented by copying XY into the program counter during the execute phase.) Example: B43C would first compare the contents of register 4 with the contents of register 0. If the two were equal, the pattern 3C would be placed in the program counter so that the next instruction executed would be the one located at that memory address. Otherwise, nothing would be done and program execution would continue in its normal sequence.
C	000	HALT execution. Example: C000 would cause program execution to stop.

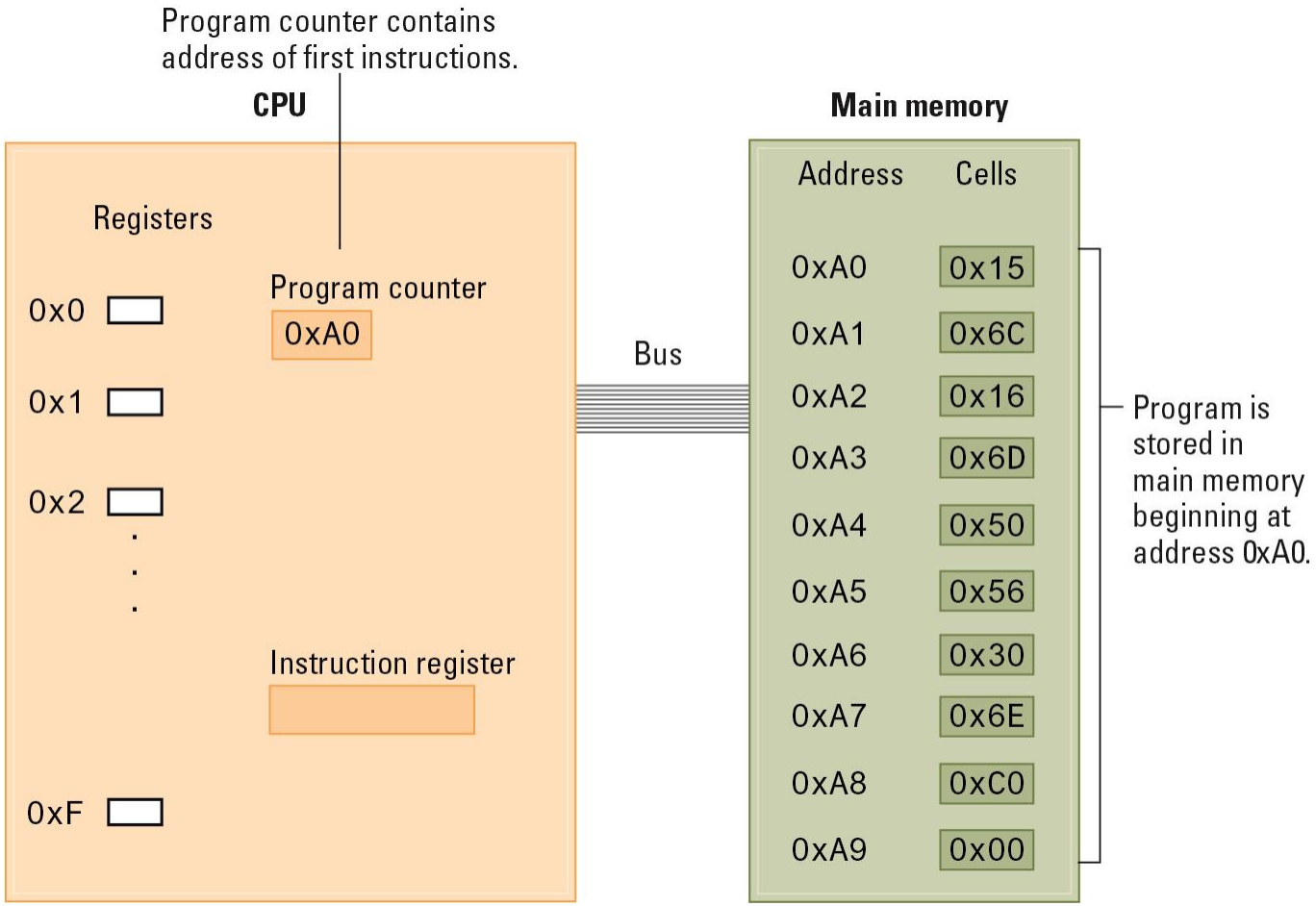
Esempio: decodificare l'istruzione 0x35A7



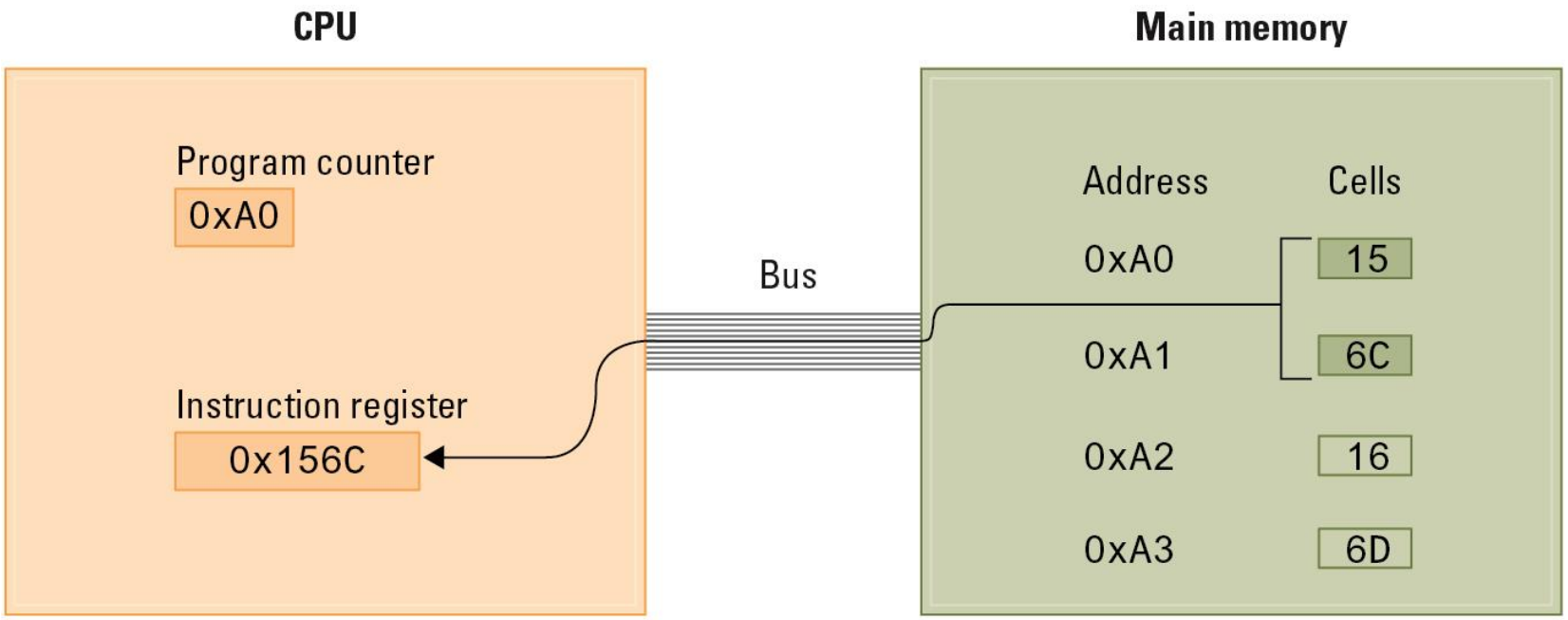
Il ciclo macchina



Programma caricato nella memoria principale

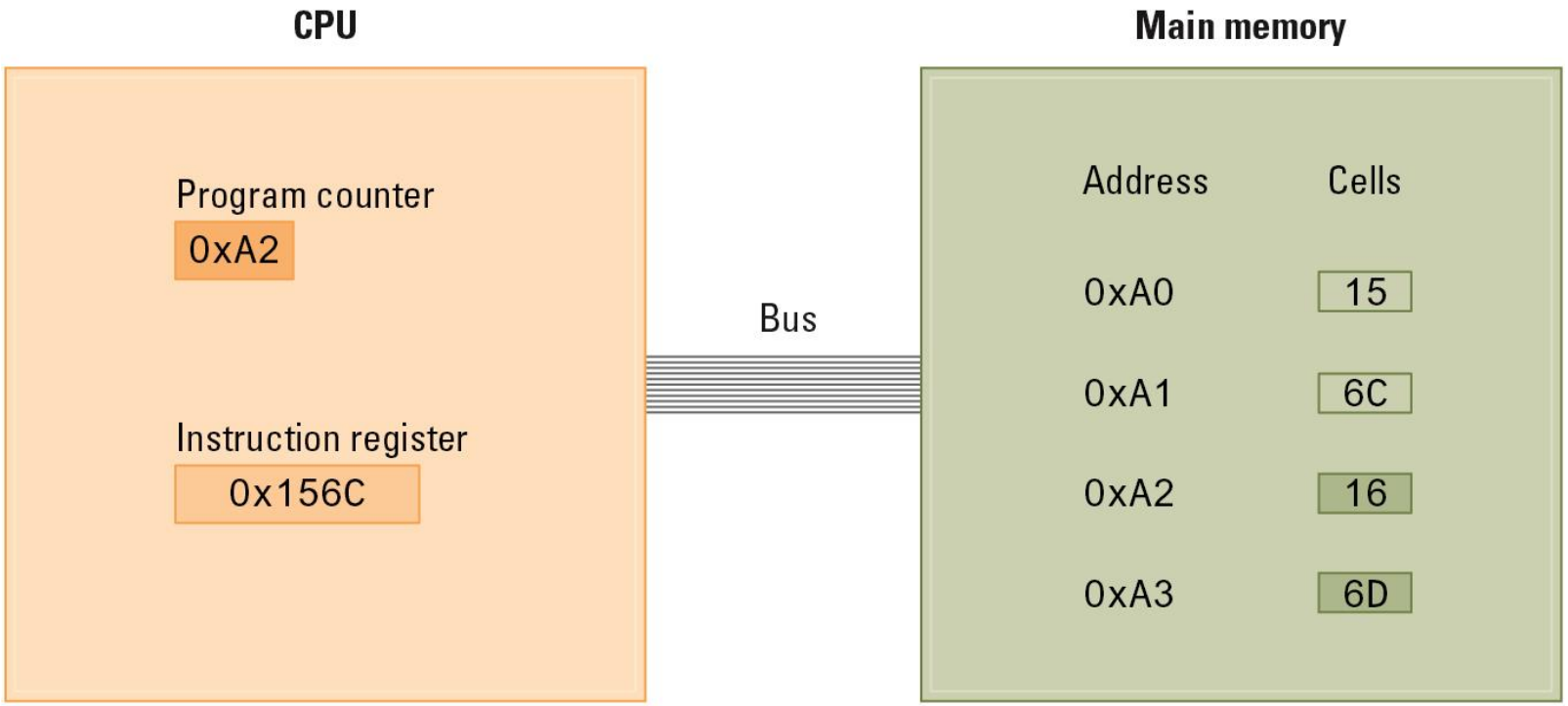


Fase di fetch (I)



a. At the beginning of the fetch step, the instruction starting at address 0xA0 is retrieved from memory and placed in the instruction register.

Fase di fetch (II)



b. Then the program counter is incremented so that it points to the next instruction.

Arithmetic Logic Unit (ALU)

- Logic Operations:
  - AND, OR, XOR
- Rotation and Shift Operations:
  - circular shift, logical shift, arithmetic shift

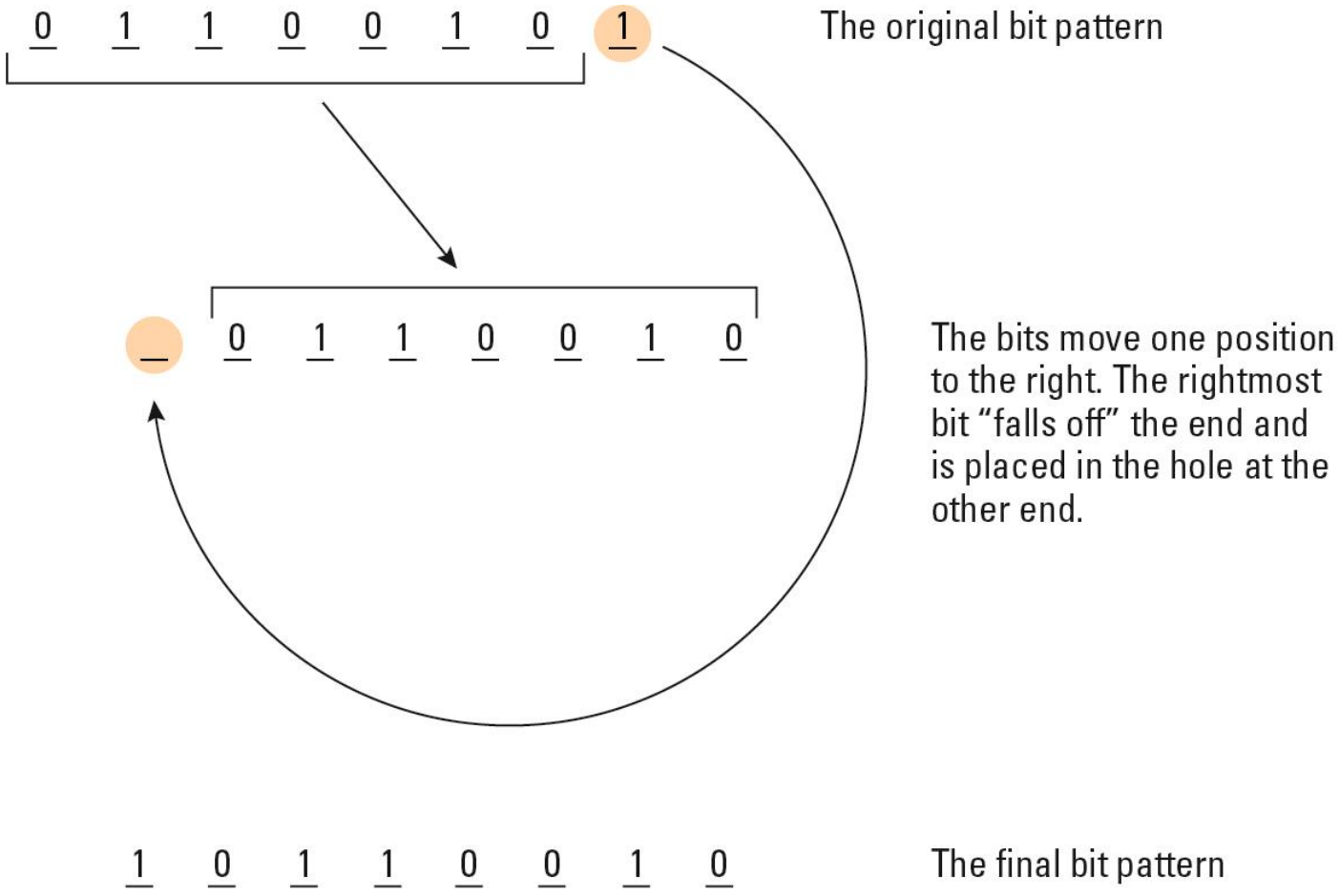
Operazione di shift

0 1 0 1 0 0 1 1

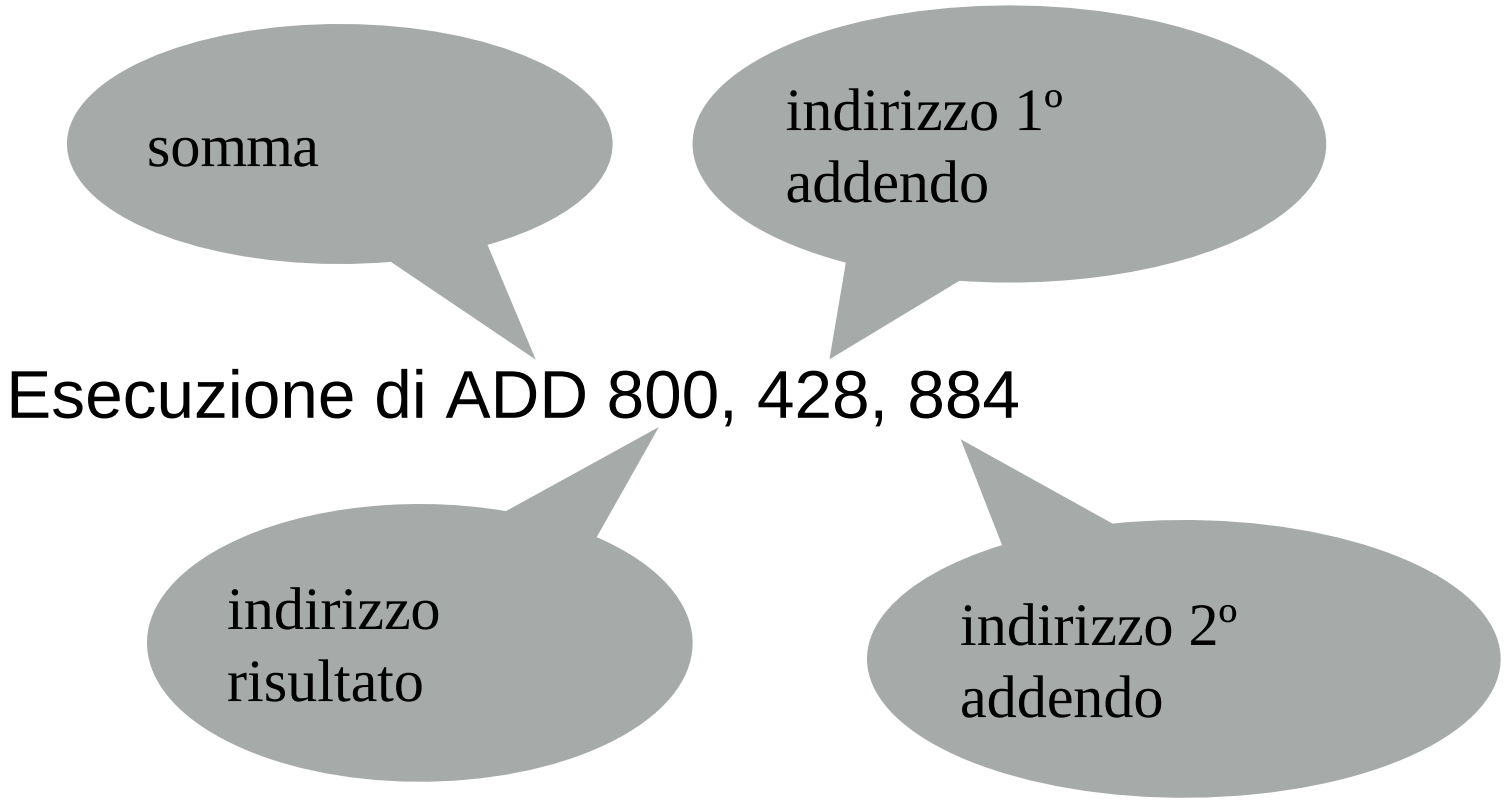
Shift a sinistra:

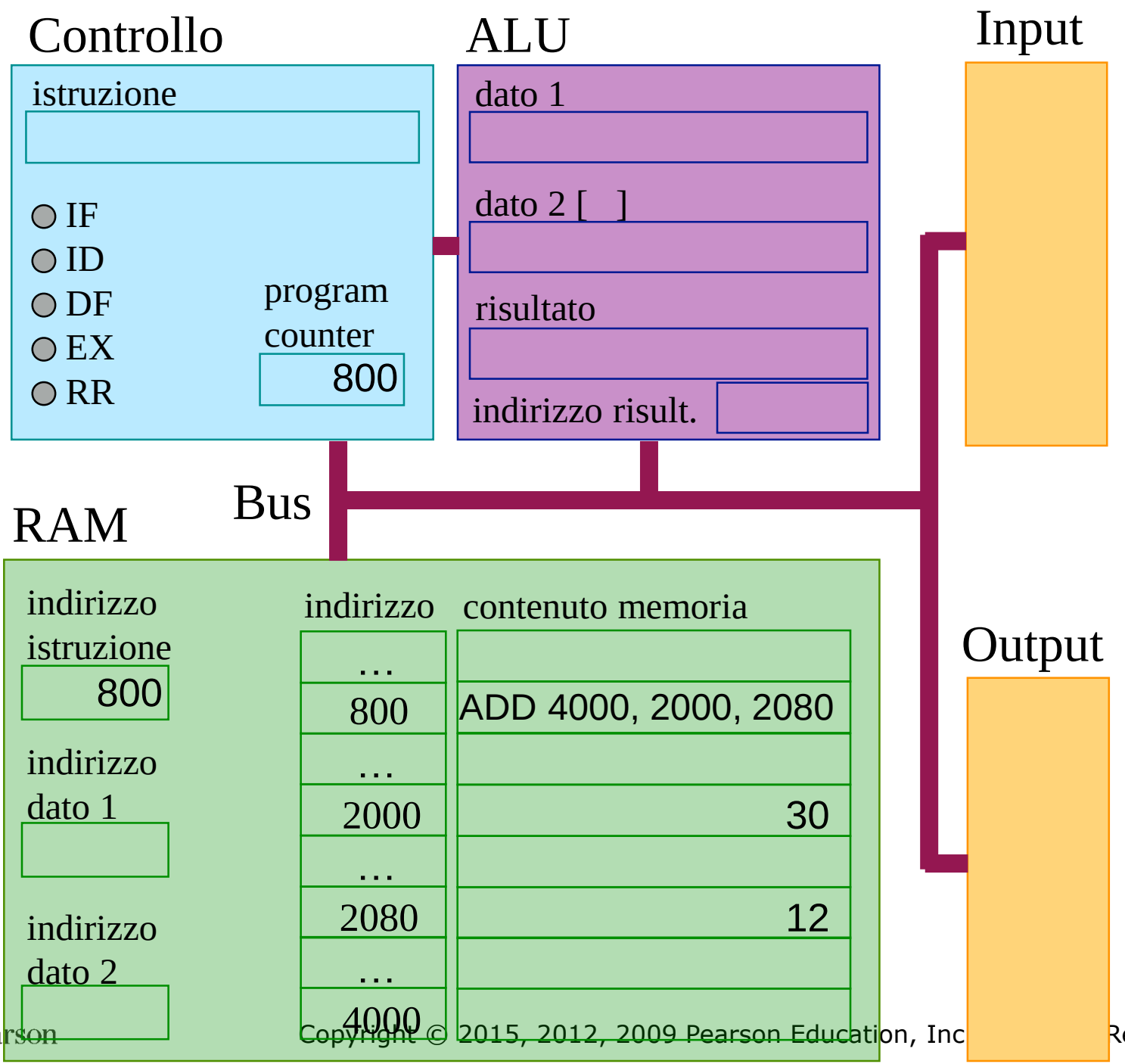
1 0 1 0 0 1 1 0

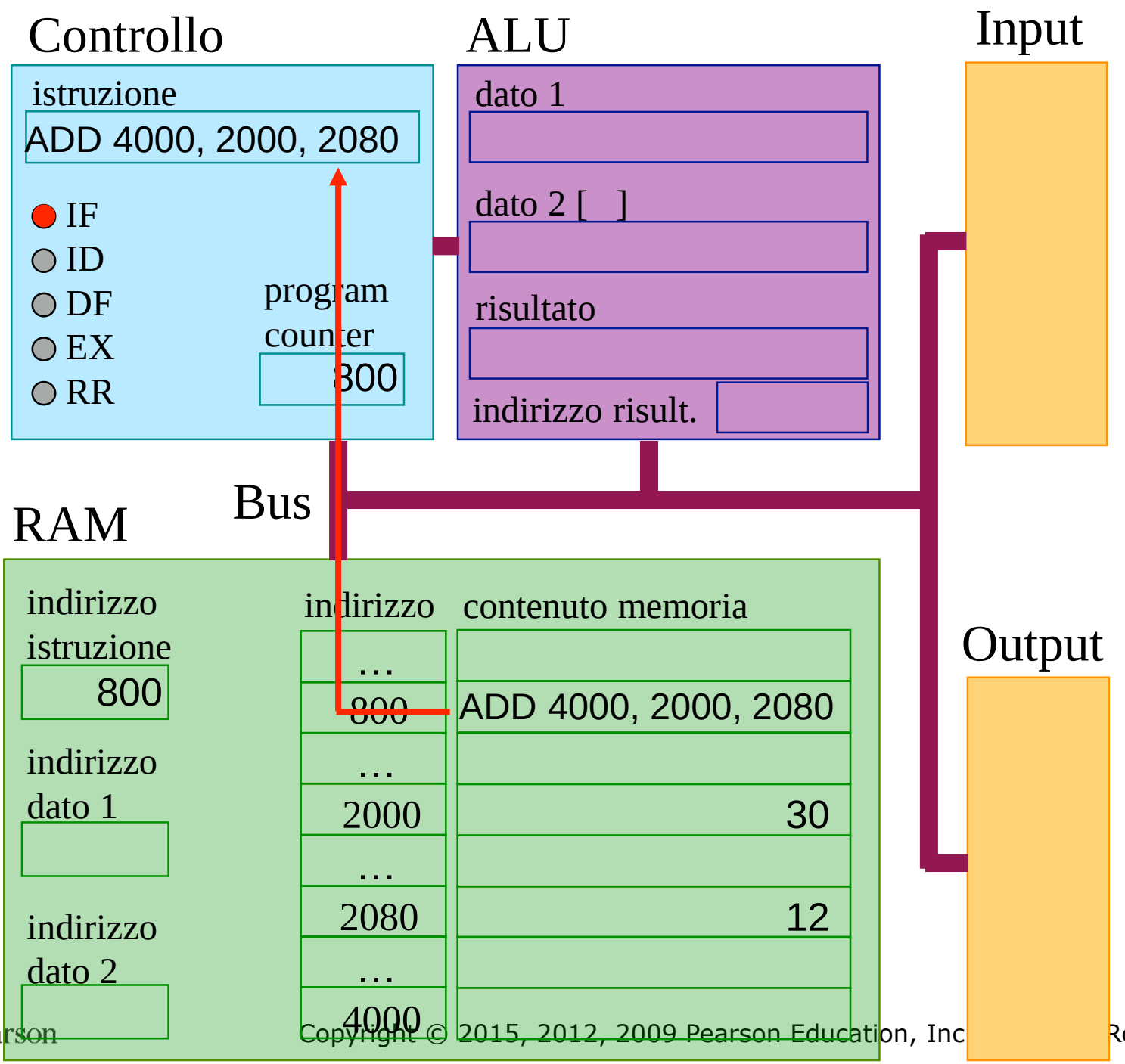
Operazione Rotate

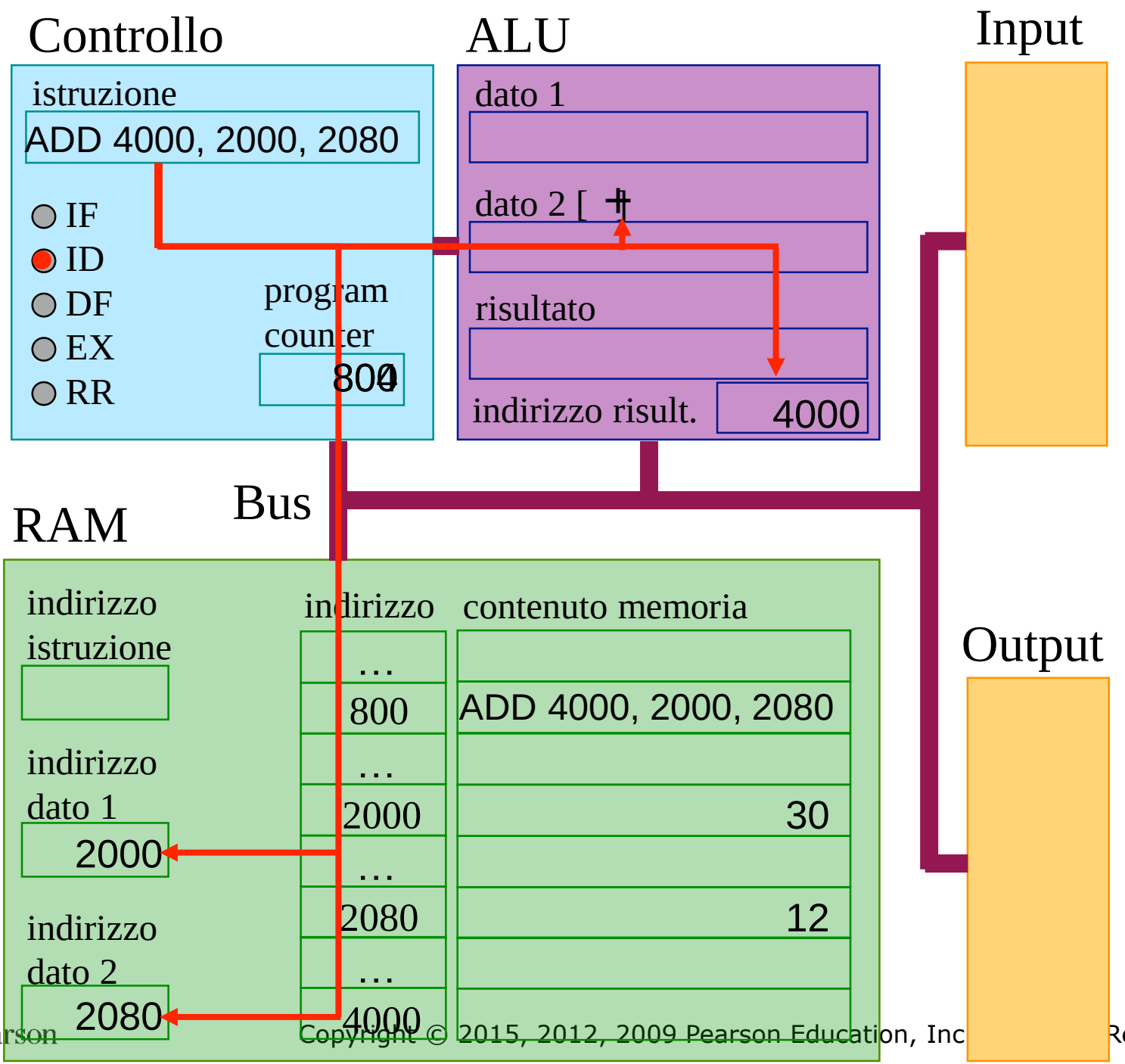


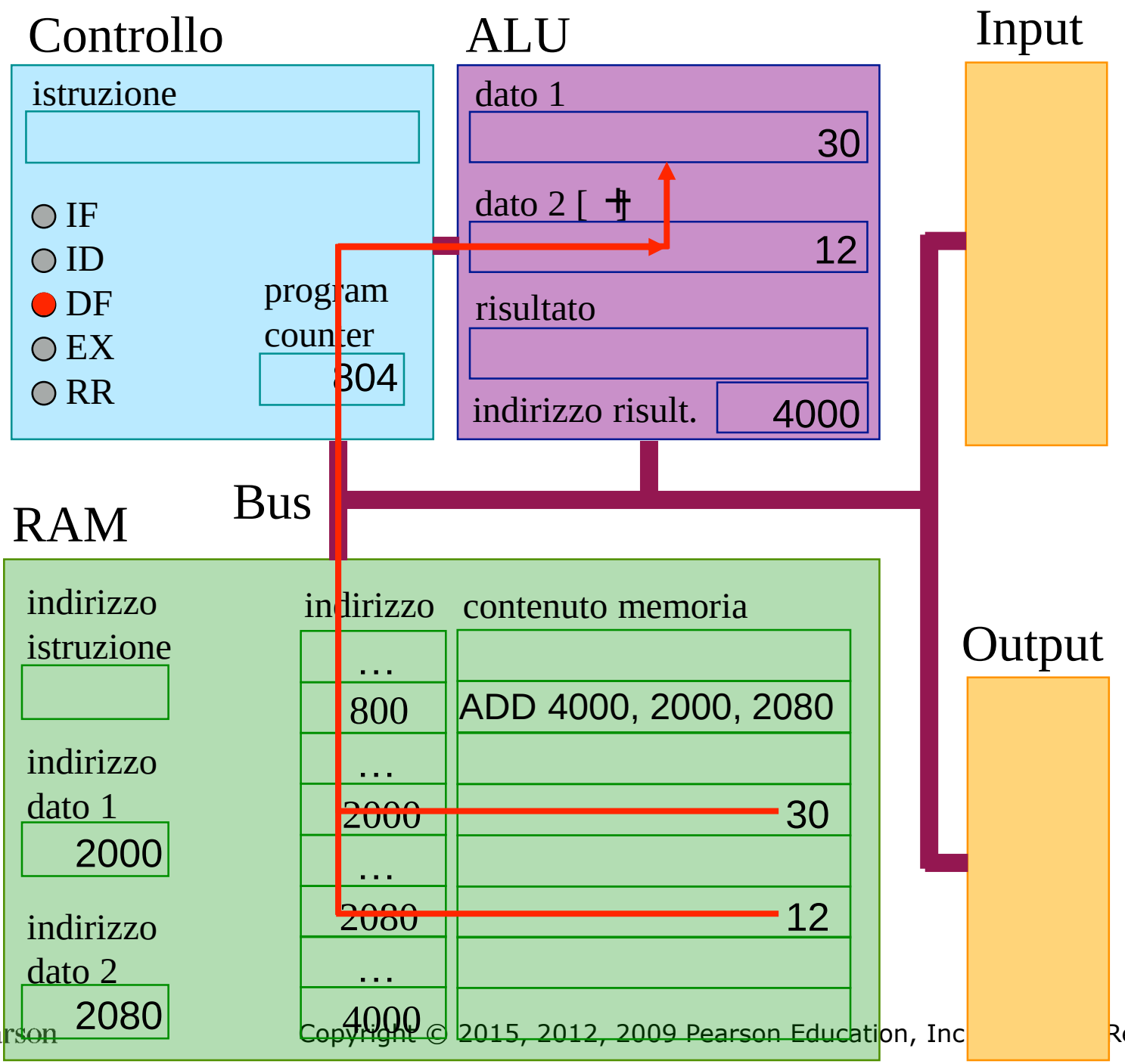
ADD 800, 428, 884

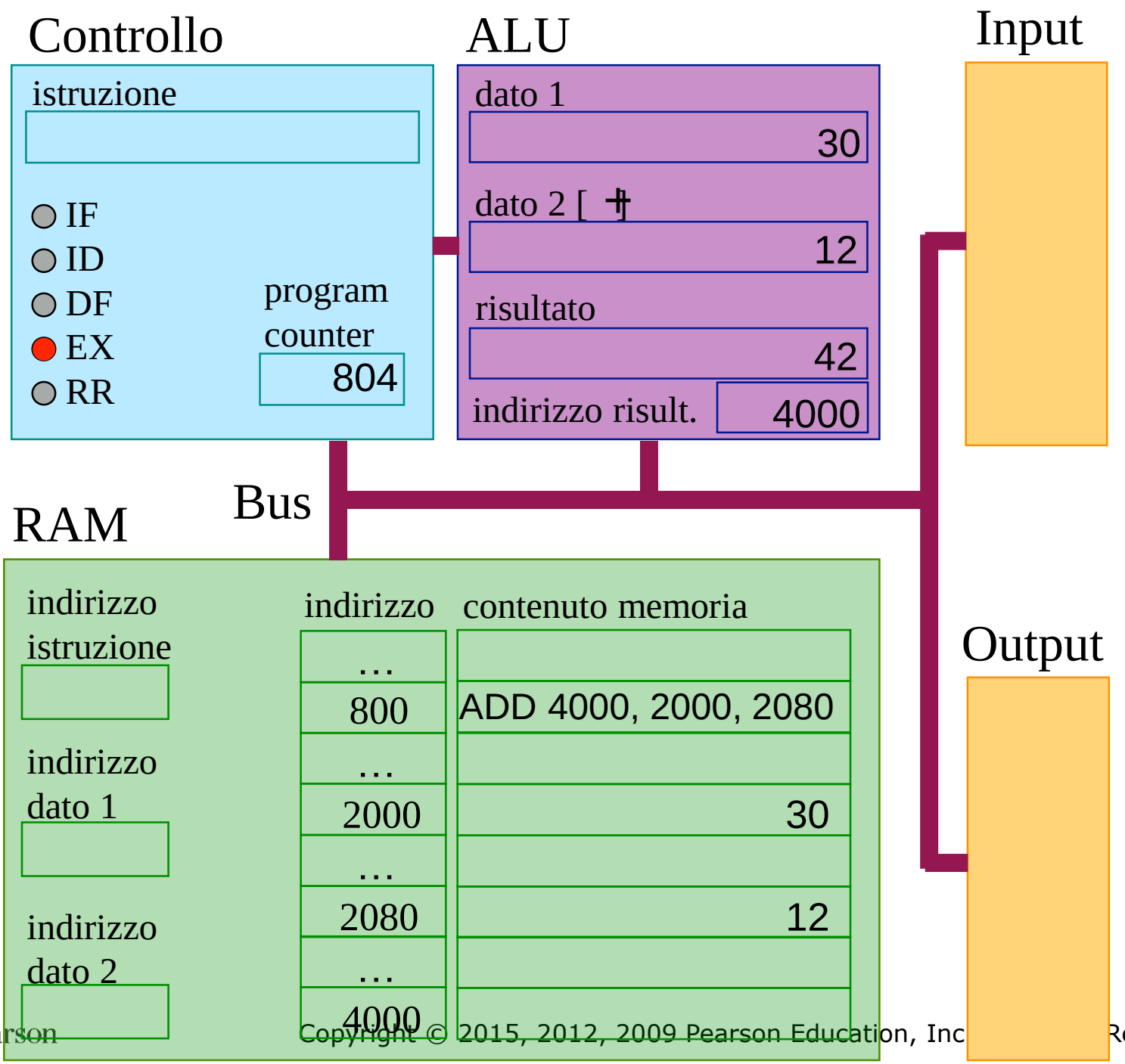


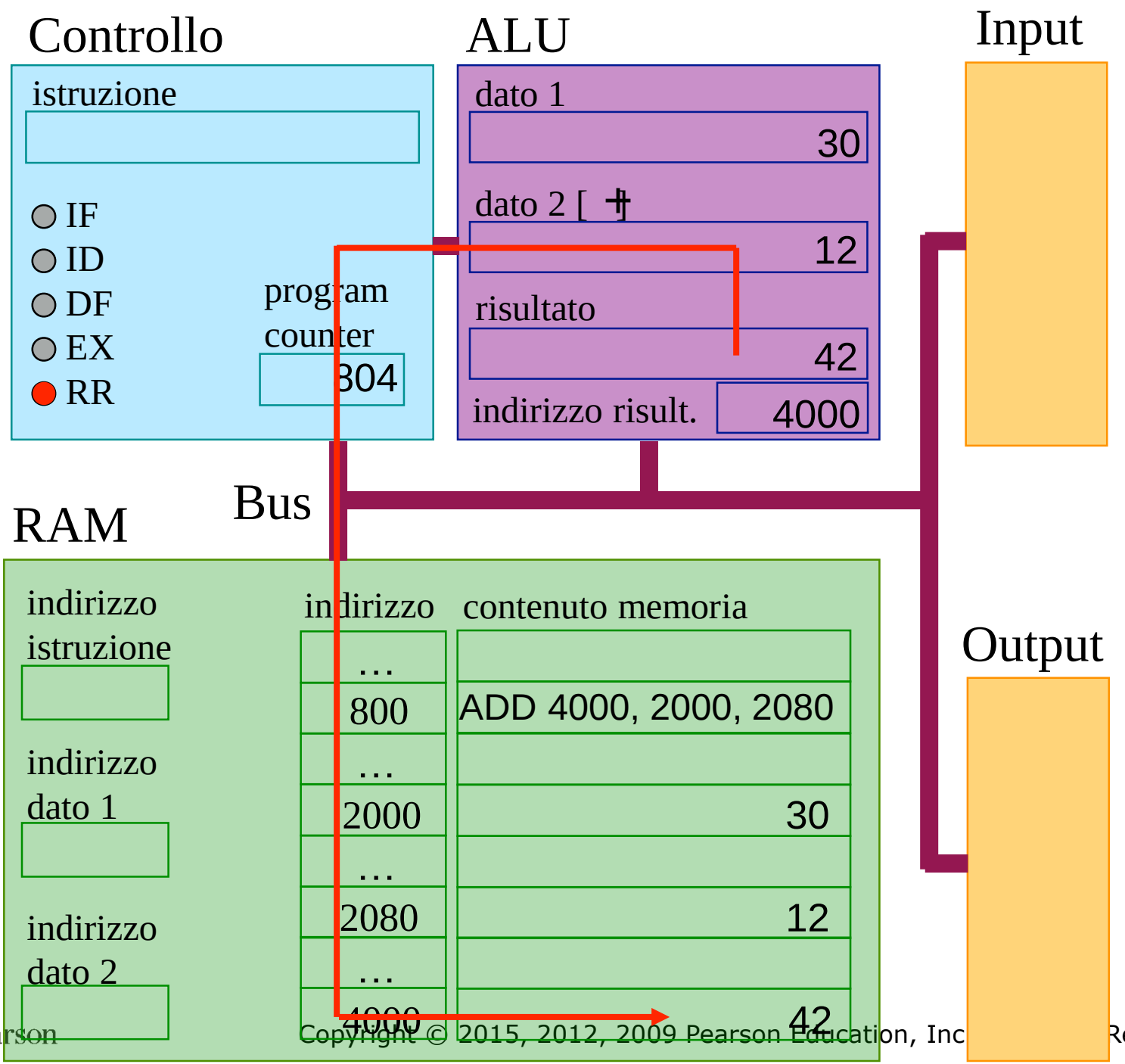








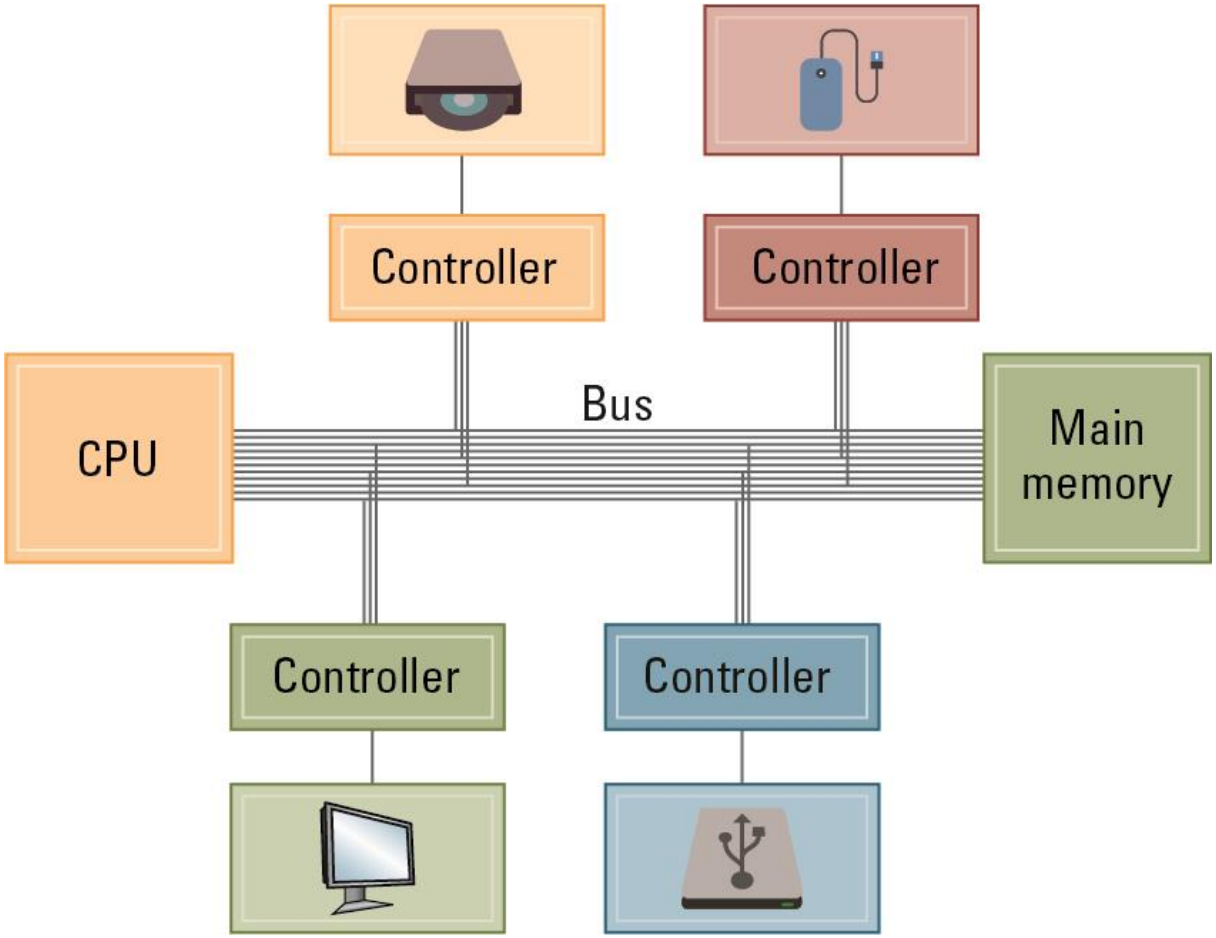




Comunicazione con altri dispositivi (I)

- **Controller:** gestisce la comunicazione tra computer e altri dispositivi
  - Specializzati (by type of device)
  - General purpose
    - Universal Serial Bus (USB)
    - High Definition Media Interface (HDMI)
- **Port:** punto di connessione device-computer

Controllers



Comunicazione con altri dispositivi (II)

- **Direct memory access (DMA):** capacità dei controller di accedere alla memoria principale
  - **Collo di bottiglia di Von Neumann :** avviene quando la CPU ed I controllers competono per l'accesso al bus
- **Handshaking:** processo di coordinamento durante il trasferimento dei dati tra computer e periferica

Comunicazione seriale e parallela

- **Popular Communication Media**
  - **Parallel Communication**
  - **Serial Communication**

Data Communication Rates

- Measurement units
  - bps: bits per second
  - Kbps: Kilo-bps (1,000 bps)
  - Mbps: Mega-bps (1,000,000 bps)
  - Gbps: Giga-bps (1,000,000,000 bps)
- Bandwidth: Maximum available rate

Altre architetture

- Tecnologie per aumentare il tasso di trasmissione:
  - Pipelining: sovrappone gli step del ciclo macchina
  - Elaborazione parallela: usa più processori simultaneamente
    - SISD: Single Instruction, Single Data
      - No parallel processing
    - MIMD: Multiple Instruction, Multiple Data
      - Different programs, different data
    - SIMD: Single Instruction, Multiple Data
      - Same program, different data