

Architettura di un Elaboratore

Fabio Massimo Zanzotto

Ricapitoliamo puntate precedenti

- Cosa abbiamo a disposizione:
 - Concetto di algoritmo (con eventuale parametrizzazione)
 - Rappresentazione dell'informazione
- Cosa abbiamo capito:
 - Dato un algoritmo e una buona rappresentazione dei "dati" che deve trattare possiamo costruire una macchina che risolve il nostro problema o esegue un compito
- Cosa manca?
 - Come passiamo ad un risolutore generale di problemi?

F. M. Zanzotto, R. Bacci

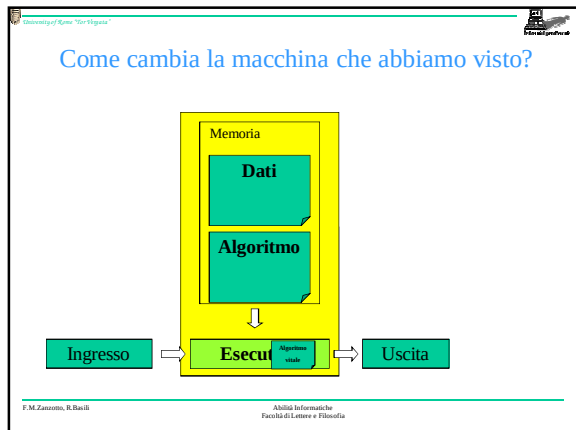
Abilità Informatiche
Facoltà di Lettere e Filosofia

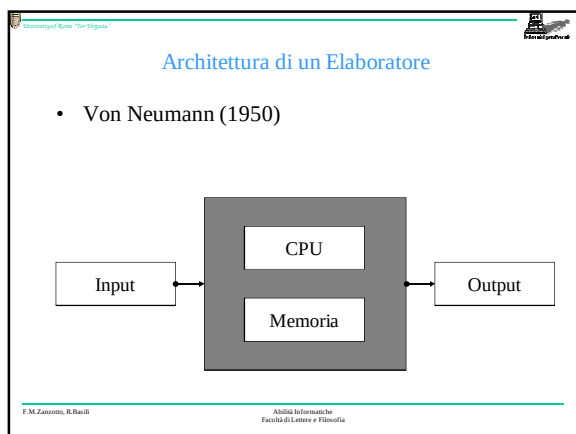
Idee Fondamentali

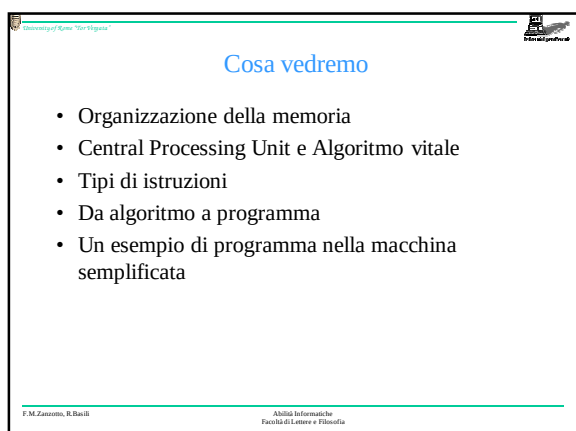
- **Algoritmo Vitale:** Si può pensare ad un algoritmo "vitale" che abbia come scopo quello di eseguire algoritmi
- **Dati e algoritmi:** Un algoritmo scritto con una certa codifica può essere "dato" di un altro algoritmo

F. M. Zanzotto, R. Bacci

Abilità Informatiche
Facoltà di Lettere e Filosofia







Organizzazione della Memoria

Memoria

Domande fondamentali

- Cos'è una **memoria**?
- Quando una memoria è utilizzabile?

Al solito, osserviamo il mondo!!!

Cos'è una memoria?

Guardiamo le “*Memorie Esterne*”
del genere umano

Noialtri come trasmettiamo e manteniamo la
conoscenza?

- Lapidi
- Fogli di carta
- Libri
- Biblioteche

Cosa hanno in comune
queste cose?

University of Rome "Tor Vergata"

Facoltà di Lettere e Filosofia

Cos'è una memoria?

- Prima risposta:
Uno **spazio definito** in cui è possibile mettere delle informazioni che poi possono essere reperite.

F.M. Zaccaro, R. Bacci

Abilità Informatiche
Facoltà di Lettere e Filosofia

University of Rome "Tor Vergata"

Facoltà di Lettere e Filosofia

Quando una memoria è utilizzabile?

- Uno spazio definito: *il libro di testo*
- Reperire informazioni:
 - Risposta mia: *"Reperite dal libro tutto quello che viene detto a lezione e studiatelo!!"*
 - Risposta che vorreste voi:

Gentile professore,
Volevo domandarle se vi sono alcune parti o capitoli dei libri di testo da lei assegnateci che possono essere non studiati.
Grazie molte.

F.M. Zaccaro, R. Bacci

Abilità Informatiche
Facoltà di Lettere e Filosofia

University of Rome "Tor Vergata"

Facoltà di Lettere e Filosofia

Quando una memoria è utilizzabile?

- Quando possiamo trovare l'informazione non parlando del contenuto.**

E' sempre possibile?

Obiettivo: **si estragga la 5 parola dei testi seguenti**

Il gatto mangia la coda al topo.

Il gatto mangiandogli la coda, fa male al topo.

F.M. Zaccaro, R. Bacci

Abilità Informatiche
Facoltà di Lettere e Filosofia

Quando una memoria è utilizzabile?

F.M. Zaccaro, R. Bacci

Abilità Informatiche
 Facoltà di Lettere e Filosofia

Cos'è una memoria?

- Seconda risposta:
 Uno **spazio definito e diviso in parti uguali (partizionato)** in cui è possibile mettere delle informazioni che poi possono essere reperite.

F.M. Zaccaro, R. Bacci

Abilità Informatiche
 Facoltà di Lettere e Filosofia

Quando una memoria è utilizzabile?

F.M. Zaccaro, R. Bacci

Abilità Informatiche
 Facoltà di Lettere e Filosofia





Cos'è una memoria?

- Terza e ultima risposta:
 Uno **spazio definito e diviso in parti uguali (partizionato) le cui parti hanno un nome detto indirizzo** in cui è possibile mettere delle informazioni che poi possono essere reperite.

Risposta finale

Uno spazio definito, partizionato e indirizzabile in cui è possibile mettere informazioni e reperirle attraverso un indirizzo

F.M. Zaccaro, R. Bacci

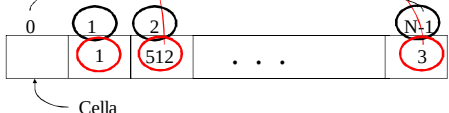
Abilità Informatiche
 Facoltà di Lettere e Filosofia





Organizzazione della memoria

- Unità atomica, bit
- Cella (o Parola)
- Valori
- Indirizzi



F.M. Zaccaro, R. Bacci

Abilità Informatiche
 Facoltà di Lettere e Filosofia





Organizzazione della memoria

- Proprietà
 - Dimensione, o Spazio di Indirizzamento
 - i.e. numero delle celle N
 - Tempo d'accesso
 - i.e. intervallo di trasferimento da/a memoria

F.M. Zaccaro, R. Bacci

Abilità Informatiche
 Facoltà di Lettere e Filosofia

Diagram illustrating the organization of memory (Organizzazione della memoria) for a 32-bit word, showing the distribution of bits across four 8-bit bytes (m-1, m-2, m-3, 3, 2, 1, 0).

The diagram shows a grid of 32 bits (8 rows by 4 columns) representing the memory organization. The columns are labeled m-1, m-2, m-3, 3, 2, 1, 0. The rows are labeled 0000000, 0000001, 0000010, ..., 0010011, ..., 1111111. The bits are distributed as follows:

- Row 0000000: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0000001: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0000010: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0000011: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0000100: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0000101: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0000110: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0000111: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0001000: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0001001: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0001010: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0001011: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0001100: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0001101: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0001110: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0001111: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0010000: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0010001: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0010010: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0010011: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0010100: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0010101: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0010110: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0010111: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0011000: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0011001: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0011010: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0011011: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0011100: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0011101: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0011110: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0011111: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0100000: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0100001: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0100010: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0100011: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0100100: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0100101: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0100110: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0100111: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0101000: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0101001: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0101010: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0101011: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0101100: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0101101: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0101110: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0101111: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0110000: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0110001: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0110010: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0110011: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0110100: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0110101: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0110110: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0110111: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0111000: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0111001: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0111010: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0111011: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0111100: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0111101: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 0111110: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 0111111: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 1000000: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 1000001: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 1000010: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 1000011: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 1000100: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 1000101: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 1000110: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 1000111: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=1
- Row 1001000: m-1=0, m-2=0, m-3=0, 3=0, 2=0, 1=0, 0=0
- Row 1001001: m-1=0, m-2=0

[illegible]




Organizzazione della memoria

- Per una memoria di dimensione K (cioe' K celle disponibili) ho bisogno di un numero n di bit di indirizzamento tale che

$$2^n \geq K$$
- Per evitare ridondanze K e' scelto tale che

$$2^n = K$$
- 1 **byte** = 8 **bit**
- 1 **kilobyte** = 1 **Kbyte** = 2^{10} bytes = 1,024 bytes
- 1 **megabyte** = 1 **Mbyte** = 2^{20} bytes 1,000,000 bytes

F.M.Zanussi, R.Belli

Abilit Informatiche
 Facoltà di Lettere e Filosofia

[illegible]

University of Rome "Tor Vergata"

Operazioni nella memoria

MILANO
MILANO

F.M. Zanone, R. Basili

Abilità Informatiche
Facoltà di Lettere e Filosofia





Operazioni della Memoria

- *Lettura di un dato (fetch)*
da Memoria a CPU
- *Scrittura di un dato (store)*
da CPU a Memoria
- Canali di Comunicazione (*bus*)
 - Dati
 - Indirizzi
 - Controllo (Lettura/Scrittura)

F.M. Zaccaro, R. Bacci

Abilità Informatiche
 Facoltà di Lettere e Filosofia





Operazioni della Memoria

- *Nella Lettura*
La CPU mette a disposizione un **indirizzo**
- *Nella Scrittura*
La CPU mette a disposizione un **indirizzo** ed un **valore**

F.M. Zaccaro, R. Bacci

Abilità Informatiche
 Facoltà di Lettere e Filosofia





Operazioni della Memoria

- Due registri gestiscono il traffico:
 - *Memory Address Register (MAR)*
 - contiene l'indirizzo della cella di memoria da cui eseguire la lettura/scrittura
 - *Memory Buffer Register (MBR)*
 - Scrittura: contiene il valore da memorizzare
 $M(MAR) \leftarrow MBR$
 - Lettura: riceve il valore della cella di indirizzo MAR
 $M(MAR) \rightarrow MBR$

F.M. Zaccaro, R. Bacci

Abilità Informatiche
 Facoltà di Lettere e Filosofia



University of Rome "Tor Vergata"



Facoltà di Lettere e Filosofia

Abilità Informatiche

Gerarchia della Memoria

- Registri
 - Veloci, Costosi
- RAM (Random Access Memory)
 - Meno veloce, meno costosa
- Memoria Secondaria
 - Lenta, bassissimi costi
- Cache
 - tra RAM e registri

F.M. Zaccaro, R. Bacci

Abilità Informatiche
Facoltà di Lettere e Filosofia

CPU e Algoritmo vitale



University of Rome "Tor Vergata"



Facoltà di Lettere e Filosofia

Abilità Informatiche

Il Processore Centrale (CPU)

... cosa deve fare?

Ciclo Vitale:

- Ciclo Operativo (*Instruction Cycle*)
 - Localizza la istruzione successiva *I*
 - Carica l'istruzione *I*
 - Decodifica *I*
 - Esegue *I*

F.M. Zaccaro, R. Bacci

Abilità Informatiche
Facoltà di Lettere e Filosofia

University of Rome "Tor Vergata"

Facoltà di Lettere e Filosofia

Il Processore Centrale (CPU)

L'Algoritmo:

- Finché ($I \neq \text{HALT}$) {
 - Localizza I
 - Carica I
 - Decodifica I
 - Esegui I

University of Rome "Tor Vergata"

Facoltà di Lettere e Filosofia

Il Processore Centrale (CPU)

- Elementi Costitutivi
 - ALU (Arithmetic-Logic Unit)
 - Opera sui dati
 - Operazioni aritmetiche, relazioni logiche, trasmissione
 - Control Unit (CU o UC)
 - Gestisce il ciclo operativo
 - Eventuali condizioni di errore (per es, overflow)
 - Interruzioni asincrone

University of Rome "Tor Vergata"

Facoltà di Lettere e Filosofia

Il Processore Centrale (CPU)

- Architettura




Un primo algoritmo

Razionalizziamo

- 1) Dati i due numeri A e B
- 2) Si metta in A ciò che si ottiene facendo $A + 1$
- 3) Si metta in B ciò che si ottiene facendo $B - 1$
- 4) Se B non è uguale a 0
 allora si torni al passo 2)
 altrimenti A contiene la somma tra l'originale A e l'originale B

F.M. Zaccaro, R. Bacci
Abilità Informatiche
Facoltà di Lettere e Filosofia




Il Processore Centrale (CPU)

- Esecuzione delle Istruzioni
 - Decodifica Operazione
 - Le operazioni possibili corrispondono a rappresentazioni, dette *codici operativi*
 - Ogni operazione determina anche il tipo dei suoi operandi (Operation Mode)
 - Gli operandi hanno una loro rappresentazione
 - Costanti (valori diretti)
 - Registri (accumulatori)
 - Indirizzi in memoria

F.M. Zaccaro, R. Bacci
Abilità Informatiche
Facoltà di Lettere e Filosofia




Il Processore Centrale (CPU)

- Esecuzione delle Istruzioni
 - Decodifica Operazione
 - Una istruzione occupa lo spazio necessario a codificare tutte le seguenti informazioni
 - Codice Operativo + Modo
 - Valori diretti *
 - Registro/i associato alla istruzione *
 - Indirizzo associato alla istruzione *

* sta per opzionale cioè 'dipendente da Codice operativo e Modo

F.M. Zaccaro, R. Bacci
Abilità Informatiche
Facoltà di Lettere e Filosofia

Il Processore Centrale (CPU)

- Localizzazione delle Istruzioni
 - Program Counter (PC)
 - E' il registro in cui in ogni istante e' memorizzato l'indirizzo della prossima istruzione
 - Instruction Register (IR)
 - E' il registro che contiene la istruzione in fase di esecuzione

F.M. Zaccaro, R. Bacci

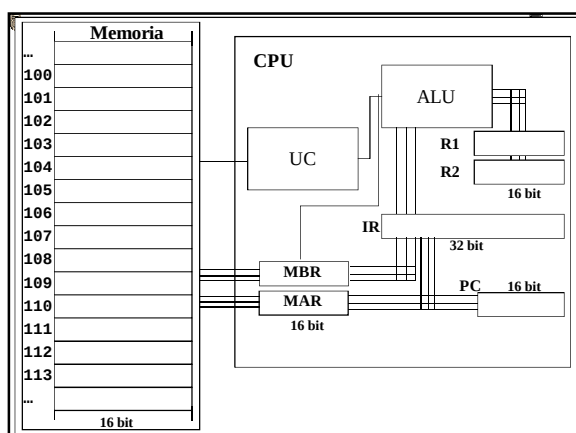
Abilità Informatiche
 Facoltà di Lettere e Filosofia

Il Processore Centrale (CPU)

- Ciclo di fetch della istruzione
 - $MAR \leftarrow PC$
 - $MBR \leftarrow M(MAR)$ % lettura istruzione
 - $IR \leftarrow MBR$
 - $PC \leftarrow PC + s$ % s e' la taglia della istruzione

F.M. Zaccaro, R. Bacci

Abilità Informatiche
 Facoltà di Lettere e Filosofia



Classi di Istruzioni

Il Processore Centrale (CPU)

- Classi di Istruzioni tipiche:
 - Istruzioni di assegnazione/modifica
 - Istruzioni di salto e di controllo
 - Istruzioni di I/O

Classi di Istruzioni

- Istruzioni di assegnazione/modifica MOV
Formalizzazione in linguaggio naturale

Es.

- Assegna ad A il valore 2
- $A = 2$ o meglio $A \leftarrow 2$
- Sia 0 il valore iniziale di I
- $I \leftarrow 0$
- $A \leftarrow A + 1$

Classi di Istruzioni

- Istruzioni di assegnazione/modifica

Es. <Date due variabili A e B>, Scambio di valori tra A e B

- Scrivi in Temp il valore corrente di A
- Scrivi in A il valore corrente di B
- Scrivi in B il valore corrente di Temp

- Temp ← A
- A ← B
- B ← Temp

F.M. Zaccaro, R. Bacci

Abilità Informatiche
 Facoltà di Lettere e Filosofia

Classi di Istruzioni

- Istruzioni di controllo (1)

Es.
Se (A>B) allora il MASSIMO e' A
altrimenti il MASSIMO e' B

Se (A>B) allora

MASSIMO ← A

altrimenti

MASSIMO ← B

F.M. Zaccaro, R. Bacci

Abilità Informatiche
 Facoltà di Lettere e Filosofia

Classi di Istruzioni

- Istruzioni di controllo (1)

Prevedono:

- Una condizione (a valori booleani)
(i.e. $f : D \rightarrow \{0,1\}$)
- Una modifica della sequenza delle istruzioni

F.M. Zaccaro, R. Bacci

Abilità Informatiche
 Facoltà di Lettere e Filosofia





Istruzione di Salto

- Salto Condizionato :
 - Frase lecita: **J<COND> <etichetta> (<etichetta>**
e' l'unico operando)
 - JMP nessuna
 - JG <cond> = il confronto ha dato esito >
 - JGE <cond> = il confronto ha dato esito ≥
 - JL <cond> = il confronto ha dato esito <
 - JLE <cond> = il confronto ha dato esito ≤
 - JEQ <cond> = il confronto ha dato esito =
 - JNZ <cond> = il confronto ha dato esito ≠

F.M. Zaccaro, R. Bacci

Abilità Informatiche
 Facoltà di Lettere e Filosofia





Istruzioni di I/O

- Gestiscono l'ingresso dei dati
(es. Leggi il valore di x,
nel calcolo di una f(x))
- Provvedono alla trasmissione in uscita dei dati
(es. y ≤ f(x)
Stampa il valore di y su schermo)

F.M. Zaccaro, R. Bacci

Abilità Informatiche
 Facoltà di Lettere e Filosofia





Il Linguaggio L definito

Istruzioni di assegnazione/modifica

MOV <OP_A> <OP_B>

Istruzioni di salto e di controllo

JMP <OP_A>, JGE<OP_A>, ...

Istruzioni di I/O

Istruzioni matematiche (Operatori sono i registri):

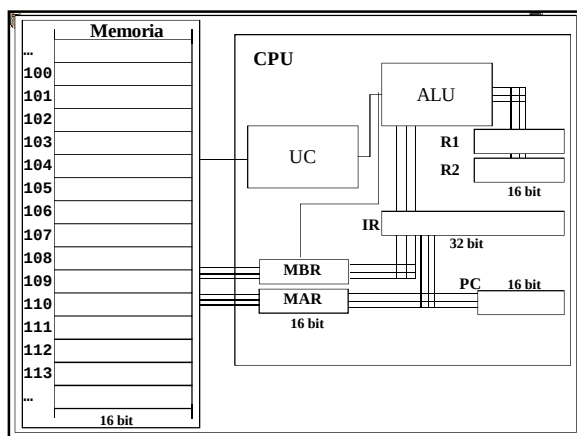
ADD<OP_A> <OP_B>

CMP <OP_A> <OP_B> : compara <OP_A> <OP_B>

F.M. Zaccaro, R. Bacci

Abilità Informatiche
 Facoltà di Lettere e Filosofia

Codificare le istruzioni



Codificare le istruzioni

MOV AX, B
Cod Operativo

--	--	--	--

Registro Indirizzo

ADD AX, BX
Cod Operativo

--	--	--	--	--

Registro Registro

F. M. Zanone, R. Ricci Analisi Informatiche Facoltà di Lettere e Filosofia

Da Algoritmo a Programma

Da Algoritmo a Programma

Problema → Algoritmo → Programma

F.M. Zanetti, R. Bacci

Abilità Informatiche
Facoltà di Lettere e Filosofia

Cominciamo a programmare

Problema: Determinare il massimo di N numeri letti in ingresso

$i \leftarrow 0, \max \leftarrow 0$

Finche' $i < N$

<Leggi X>

Se $X > \max$ allora $\max \leftarrow X$

$i \leftarrow i + 1$

<Stampa max>

Problema → Algoritmo

F.M. Zanetti, R. Bacci

Abilità Informatiche
Facoltà di Lettere e Filosofia





Cominciamo a programmare

Problema: Calcolare la divisione A/B intera

```

Quoz ← 0
<Leggi A>
<Leggi B>
Finche' B <= A
    A ← A - B
    Quoz ← Quoz + 1
<Stampa Quoz>

```

Problema → Algoritmo

F.M. Zaccaro, R. Bacci

Abilità Informatiche
 Facoltà di Lettere e Filosofia





Cominciamo a programmare

Problema: Determinare il massimo tra due numeri A e B

- <Leggi A>
- <Leggi B>
- 3.1 Se A > B allora il
 - 3.1.1 MASSIMO e' A
 - 3.2 altrimenti
 - 3.2.1 il MASSIMO e' B
- <Scrivi Massimo>

Problema → Algoritmo

F.M. Zaccaro, R. Bacci

Abilità Informatiche
 Facoltà di Lettere e Filosofia





Da Algoritmo a Programma

Problema: Determinare il massimo tra due numeri A e B
(sia dato il **Linguaggio L**)

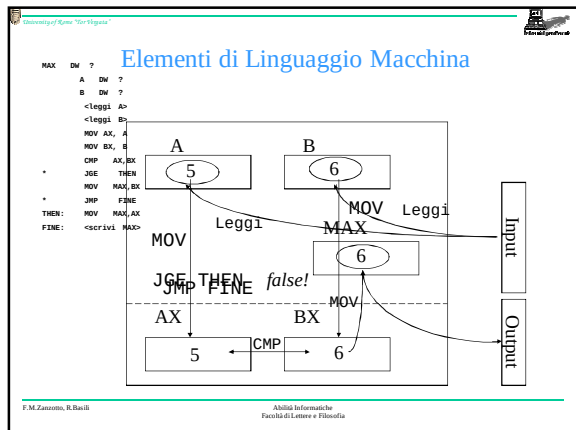
```

MOV AX, A
MOV BX, B
CMP AX, BX
JGE THEN
MOV MAX, BX
JMP FINE
THEN: MOV MAX, AX
FINE: <scrivi MAX>

```

F.M. Zaccaro, R. Bacci

Abilità Informatiche
 Facoltà di Lettere e Filosofia



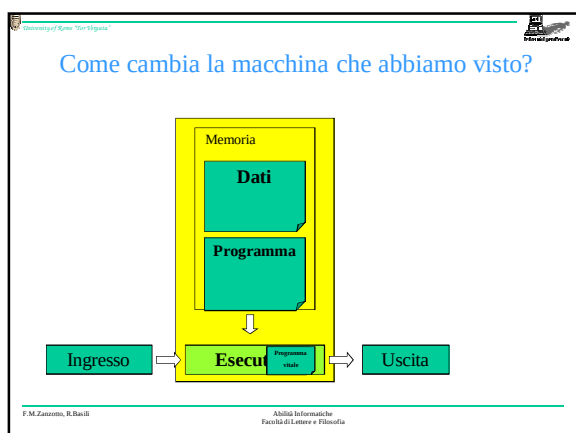
Osserviamo il concetto introdotto

Da *Algoritmo* a *Programma*:

- Una macchina ha un insieme di istruzioni che può eseguire
- Le istruzioni, come i dati, sono scritti secondo una codifica
- Data una macchina, un *programma* è la traduzione di un algoritmo in una sequenza di istruzioni che la macchina può eseguire

Attenzione: Una macchina può essere anche “virtuale” (ovvero non realizzata direttamente da una macchina fisica)... conservate il concetto!

F.M. Zanoni, R. Bacci



Un esempio di programma nella macchina semplificata

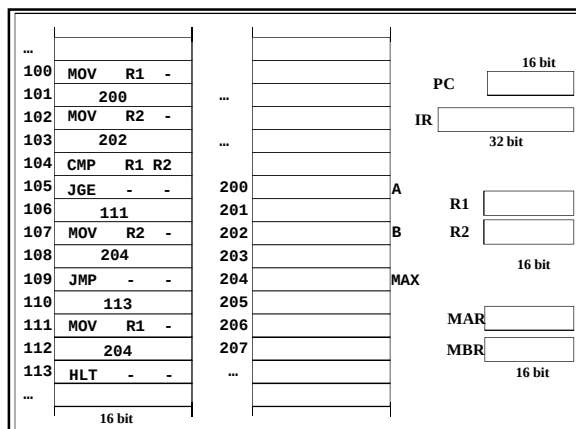
Linguaggio Macchina

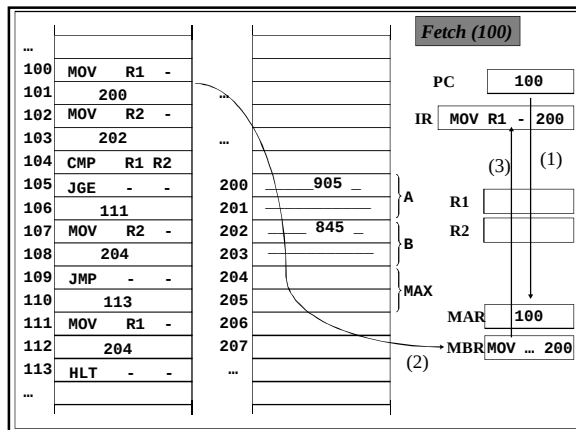
- Esempio: Massimo tra due numeri in Assembler

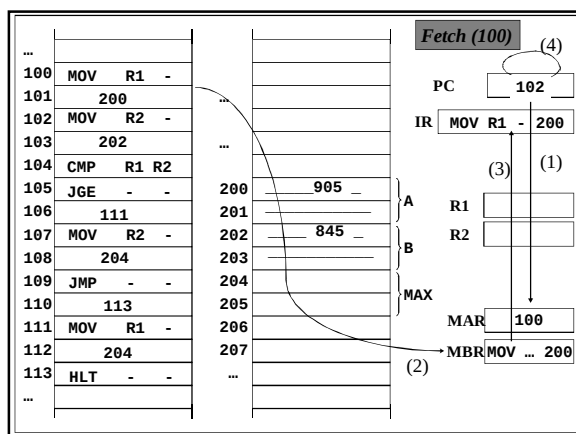
```

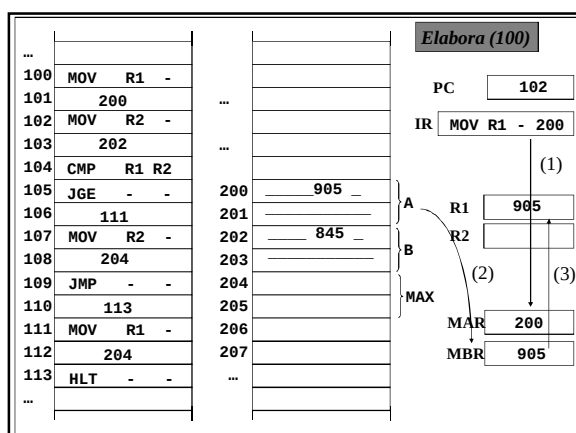
MAX  DW  ?
A     DW  ?
B     DW  ?
...
MOV  AX, A
MOV  BX, B
CMP  AX, BX
JGE  THEN
MOV  MAX, BX
JMP  FINE
THEN MOV  MAX, AX
FINE HLT

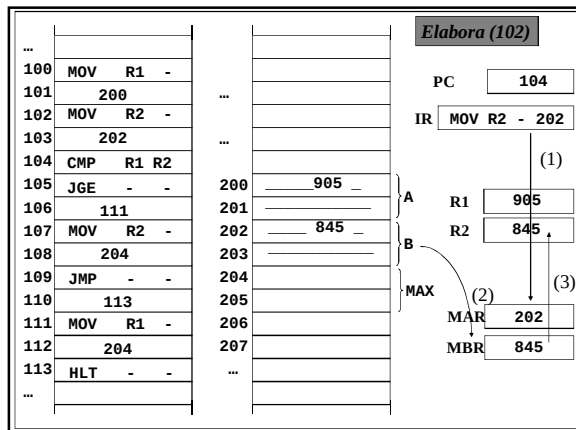
```

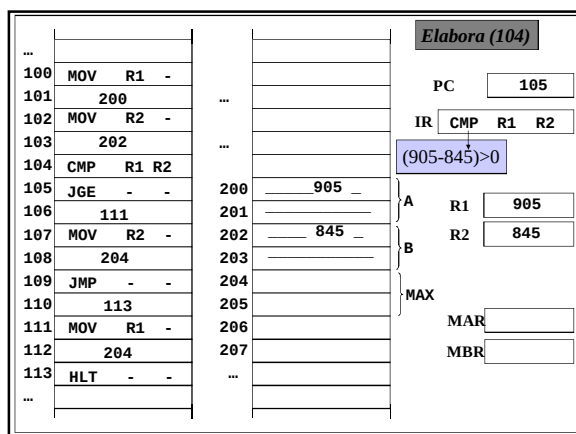


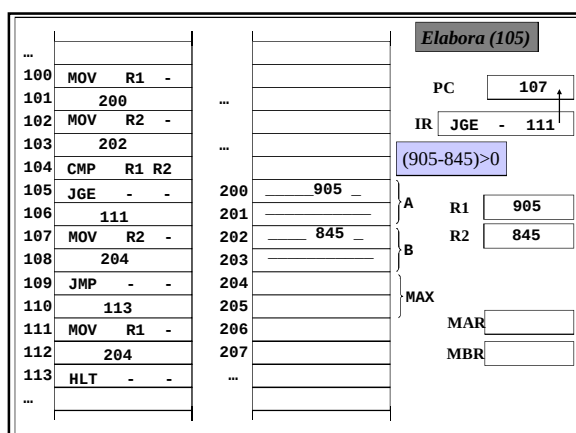


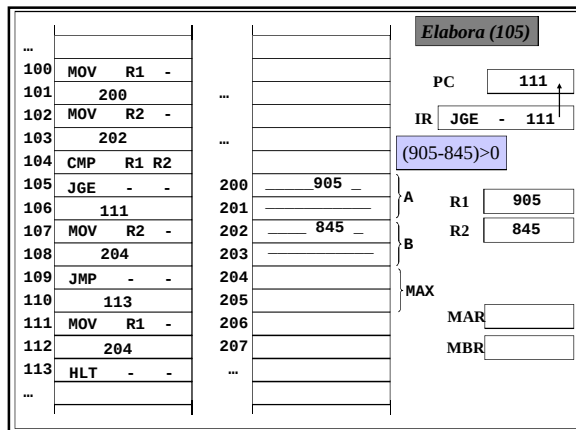


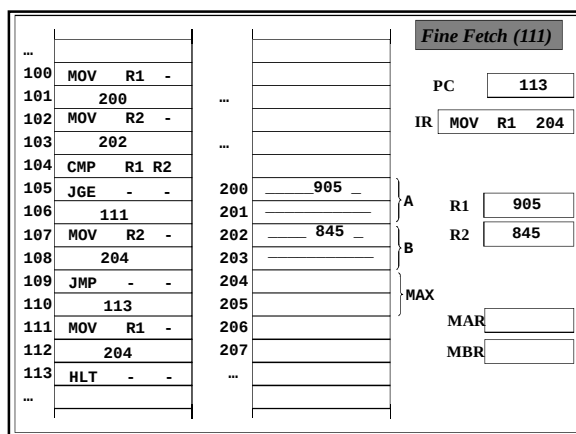


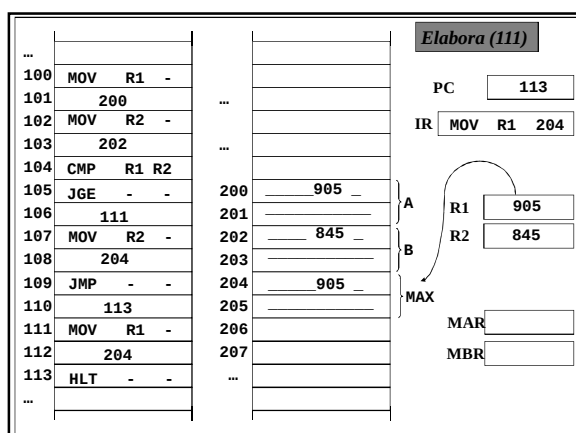












University of Rome "Tor Vergata"

Facoltà di Lettere e Filosofia

Ricapitaliamo

- Idee Fondamentali
 - **Algoritmo Vitale:** Si può pensare ad un algoritmo “vitale” che abbia come scopo quello di eseguire algoritmi
 - **Dati e algoritmi:** Un algoritmo scritto con una certa codifica può essere “dato” di un altro algoritmo
- Abbiamo visto:
 - Macchina di Von Neumann
 - Memoria
 - Central Processing Unit
 - Concetto di Programma
 - Un esempio di esecuzione di programma

F.M. Zaccaro, R. BasiliAbilità Informatiche
Facoltà di Lettere e Filosofia
