

# Algoritmi e la macchina di von Neumann

Stefano Lodi

Insegnamento di Informatica

*Alma Mater Studiorum Università di Bologna*

`stefano.lodi@unibo.it`

# Algoritmi

## Algoritmi

Requisiti per gli algoritmi

Esempio di flow-chart

Osservazioni

Ripetizioni nel flow-chart

Un esecutore meccanico astratto

Un algoritmo eseguibile da  $M1$

Macchina di von Neumann

Caratteristiche della Macchina di von Neumann

Controllo della sequenza di esecuzione

Linearizzazione

Un modello di esecutore CPU

Istruzioni assemblative

Istruzioni assemblative

Linguaggi assemblativi nelle CPU attuali

Traduzione di istruzioni complesse

Prodotto di interi in linguaggio assemblativo

Prodotto di interi in linguaggio assemblativo

Prodotto di interi in linguaggio assemblativo

## ■ Un'idea fondamentale della informatica e della matematica **algoritmo**

- ◆ nell'antichità venivano già usati algoritmi per il calcolo delle operazioni aritmetiche
- ◆ Il termine deriva dal nome Al-Khuwarizmi, matematico arabo
- ◆ Un algoritmo può essere fornito a una persona o macchina (opportunamente codificato), cioè un *calcolatore elettronico*, per essere eseguito allo scopo di *compiere un lavoro*
- ◆ esempio: un semplice algoritmo adatto a un esecutore umano
  1. inserire tre monete da 20 centesimi
  2. tirare il cassetto corrispondente al tipo di caramelle desiderato
  3. estrarre il pacchetto
  4. richiudere il cassetto
  5. ritirare il resto di 10 centesimi

# Requisiti per gli algoritmi

Algoritmi

Requisiti per gli algoritmi

Esempio di flow-chart

Osservazioni

Ripetizioni nel flow-chart

Un esecutore meccanico astratto

Un algoritmo eseguibile da  $M1$

Macchina di von Neumann

Caratteristiche della Macchina di von Neumann

Controllo della sequenza di esecuzione

Linearizzazione

Un modello di esecutore CPU

Istruzioni assemblative

Istruzioni assemblative

Linguaggi assemblativi nelle CPU attuali

Traduzione di istruzioni complesse

Prodotto di interi in linguaggio assemblativo

Prodotto di interi in linguaggio assemblativo

Prodotto di interi in linguaggio assemblativo

Prodotto di interi in linguaggio assemblativo

assemblativo

- Un esecutore umano è in grado di risolvere spesso ambiguità e imprevisti
- Un calcolatore elettronico non possiede tali capacità
- Pertanto è necessario imporre il rispetto di alcune condizioni nella formulazione di algoritmi per un calcolatore
  - ◆ Le istruzioni non possono essere ambigue
  - ◆ Le istruzioni devono essere eseguibili in tempo finito
  - ◆ La terminazione dell'esecuzione dopo un numero finito di passi deve essere garantita, in qualunque situazione iniziale
- Un formalismo grafico intuitivo per descrivere algoritmi è quello dei *diagrammi a blocchi (flow-chart)*, introdotto negli anni '60
- Le istruzioni e la loro successione è indicata attraverso blocchi di forma diversa collegati da linee orientate
  - ◆ L'esagono è un *blocco di controllo*, cioè specifica il prossimo blocco da eseguire a seconda del risultato di un test
  - ◆ Il rettangolo è un *blocco azione*, che specifica una operazione o azione da eseguire
  - ◆ Ellissi sono i blocchi *iniziale* e *finale*, che specificano dove iniziare e quando terminare

# Esempio di flow-chart

Algoritmi

Requisiti per gli algoritmi

Esempio di flow-chart

Osservazioni

Ripetizioni nel flow-chart

Un esecutore meccanico astratto

Un algoritmo eseguibile da  $M1$

Macchina di von Neumann

Caratteristiche della Macchina di von Neumann

Controllo della sequenza di esecuzione

Linearizzazione

Un modello di esecutore CPU

Istruzioni assemblative

Istruzioni assemblative

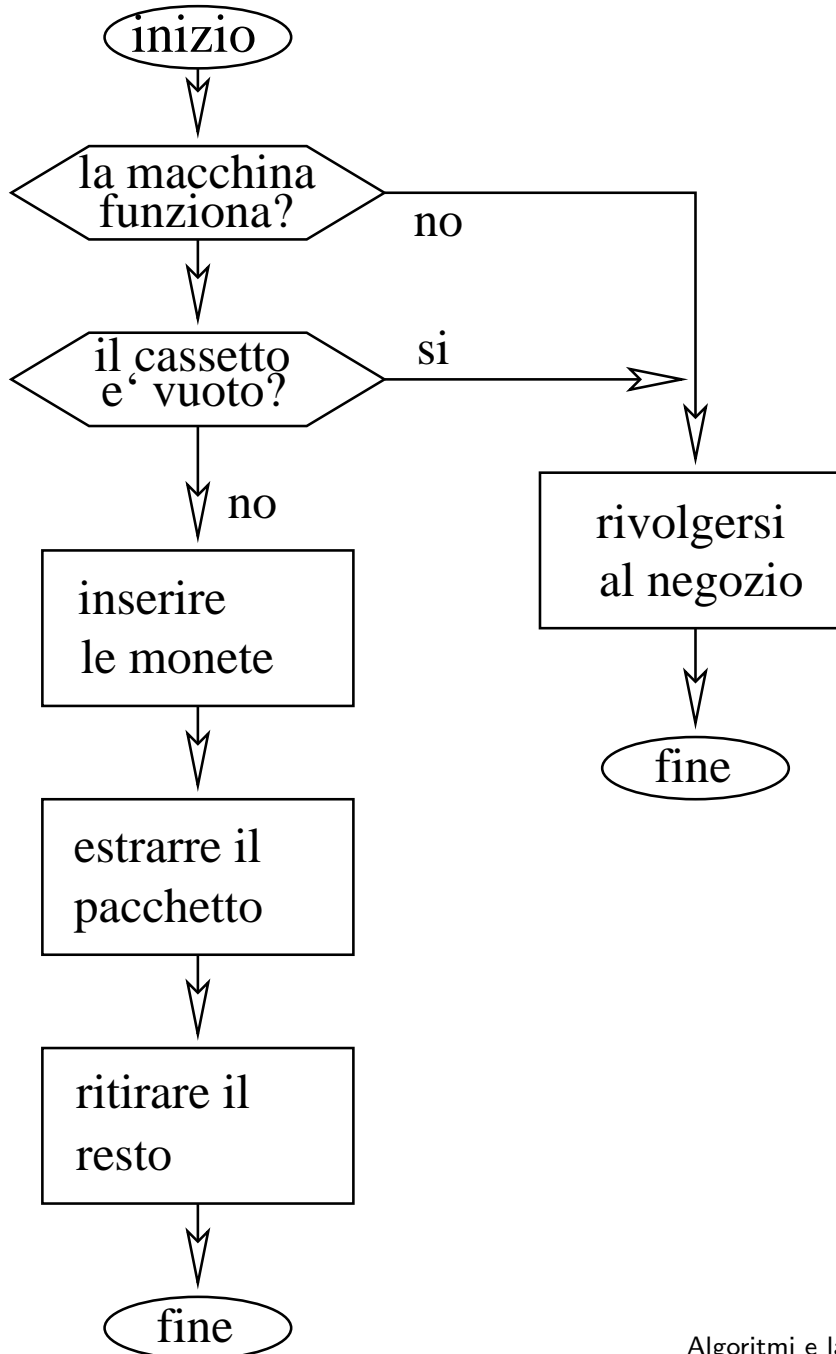
Linguaggi assemblativi nelle CPU attuali

Traduzione di istruzioni complesse

Prodotto di interi in linguaggio assemblativo

Prodotto di interi in linguaggio assemblativo

Prodotto di interi in linguaggio assemblativo



# Osservazioni

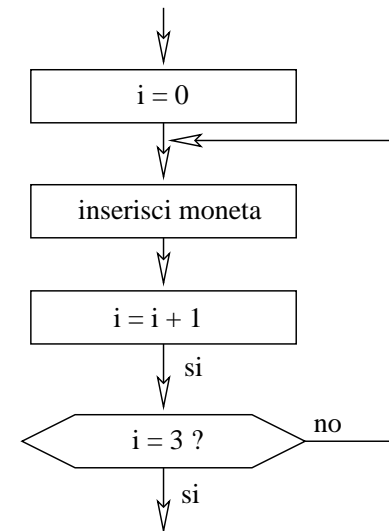
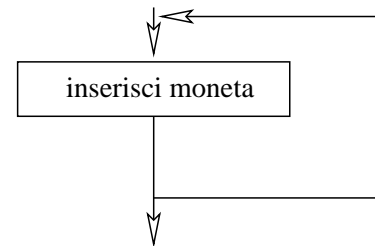
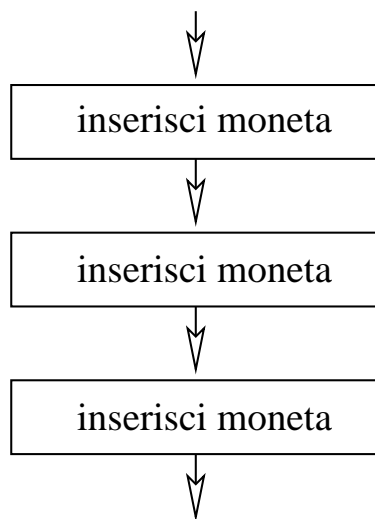
Algoritmi  
Requisiti per gli algoritmi  
Esempio di flow-chart  
Osservazioni  
Ripetizioni nel flow-chart  
Un esecutore meccanico astratto  
Un algoritmo eseguibile da  $M1$   
Macchina di von Neumann  
Caratteristiche della Macchina di von Neumann  
Controllo della sequenza di esecuzione  
Linearizzazione  
Un modello di esecutore CPU  
Istruzioni assemblative  
Istruzioni assemblative  
Linguaggi assemblativi nelle CPU attuali  
Traduzione di istruzioni complesse  
Prodotto di interi in linguaggio assemblativo  
Prodotto di interi in linguaggio assemblativo  
Prodotto di interi in linguaggio assemblativo

- Sono presenti due blocchi di controllo: verificano se il distributore funziona e se le caramelle volute sono presenti.
  - ◆ Sono necessari per eliminare due ambiguità insite nell'algoritmo per l'esecutore umano
- I blocchi di controllo rendono il diagramma bidimensionale, prospettando due vie d'uscita dal blocco, per proseguire nell'esecuzione.
  - ◆ Se la condizione è vera, si deve uscire lungo la linea con il si
  - ◆ Se la condizione è falsa, si deve uscire lungo la linea con il no
- Dunque, i blocchi di controlli possono cambiare la successione delle istruzioni effettivamente eseguite, secondo le condizioni esterne
- Il blocco "inserire le monete" non è elementare

# Ripetizioni nel flow-chart

Algoritmi  
Requisiti per gli algoritmi  
Esempio di flow-chart  
Osservazioni  
Ripetizioni nel flow-chart  
Un esecutore meccanico astratto  
Un algoritmo eseguibile da  $M1$   
Macchina di von Neumann  
Caratteristiche della Macchina di von Neumann  
Controllo della sequenza di esecuzione  
Linearizzazione  
Un modello di esecutore CPU  
Istruzioni assemblative  
Istruzioni assemblative  
Linguaggi assemblativi nelle CPU attuali  
Traduzione di istruzioni complesse  
Prodotto di interi in linguaggio assemblativo  
Prodotto di interi in linguaggio assemblativo  
Prodotto di interi in linguaggio assemblativo

- Consideriamo tre soluzioni per il blocco non elementare dell'esempio
  - ◆ La prima è corretta ma non è una tecnica di programmazione soddisfacente in generale
    - (Cosa accadrebbe se il flow-chart dovesse compiere un lavoro in cui il numero di ripetizioni richieste fosse 100000?)
  - ◆ La seconda non è ammissibile
    - l'esecuzione non terminerebbe mai
  - ◆ La terza è pienamente soddisfacente



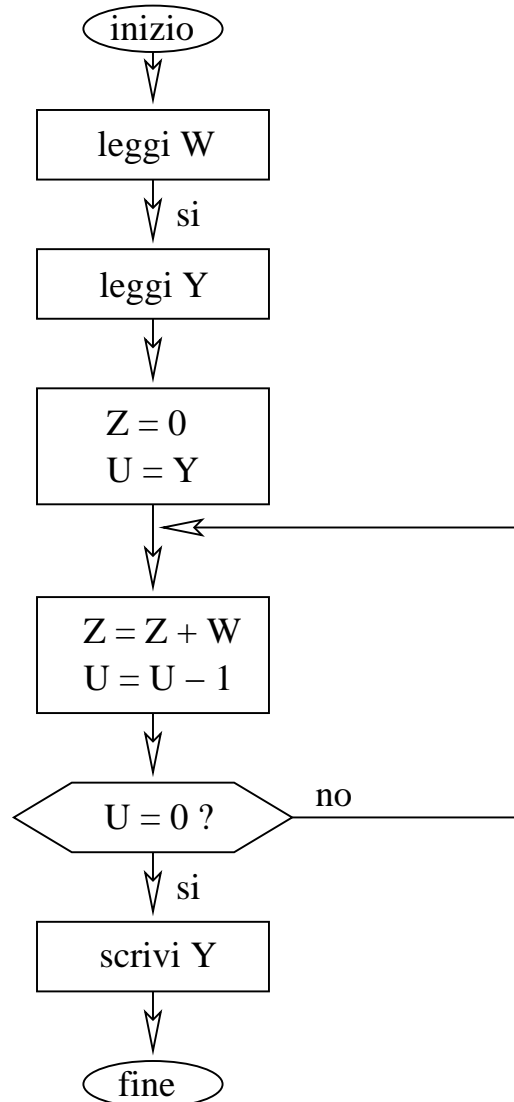
# Un esecutore meccanico astratto

Algoritmi  
Requisiti per gli algoritmi  
Esempio di flow-chart  
Osservazioni  
Ripetizioni nel flow-chart  
Un esecutore meccanico astratto  
Un algoritmo eseguibile da  $M1$   
Macchina di von Neumann  
Caratteristiche della Macchina di von Neumann  
Controllo della sequenza di esecuzione  
Linearizzazione  
Un modello di esecutore CPU  
Istruzioni assemblative  
Istruzioni assemblative  
Linguaggi assemblativi nelle CPU attuali  
Traduzione di istruzioni complesse  
Prodotto di interi in linguaggio assemblativo  
Prodotto di interi in linguaggio assemblativo  
Prodotto di interi in linguaggio assemblativo

- Supponiamo di possedere un esecutore  $M1$  e di volere scrivere algoritmi per esso
- In primo luogo occorre sapere quali sono le sue capacità
- **Ipotesi su  $M1$** 
  1. Comprende un linguaggio simbolico
  2. Può acquisire numeri interi in ingresso e produrre interi in uscita
  3. Può eseguire solo
    - ◆ somme e sottrazioni
    - ◆ verifiche di uguaglianza a zero
    - ◆ verifiche di non uguaglianza a zero

# Un algoritmo eseguibile da $M1$

Algoritmi  
Requisiti per gli algoritmi  
Esempio di flow-chart  
Osservazioni  
Ripetizioni nel flow-chart  
Un esecutore meccanico astratto  
Un algoritmo eseguibile da  $M1$   
Macchina di von Neumann  
Caratteristiche della Macchina di von Neumann  
Controllo della sequenza di esecuzione  
Linearizzazione  
Un modello di esecutore CPU  
Istruzioni assemblative  
Istruzioni assemblative  
Linguaggi assemblativi nelle CPU attuali  
Traduzione di istruzioni complesse  
Prodotto di interi in linguaggio assemblativo  
Prodotto di interi in linguaggio assemblativo  
Prodotto di interi in linguaggio assemblativo

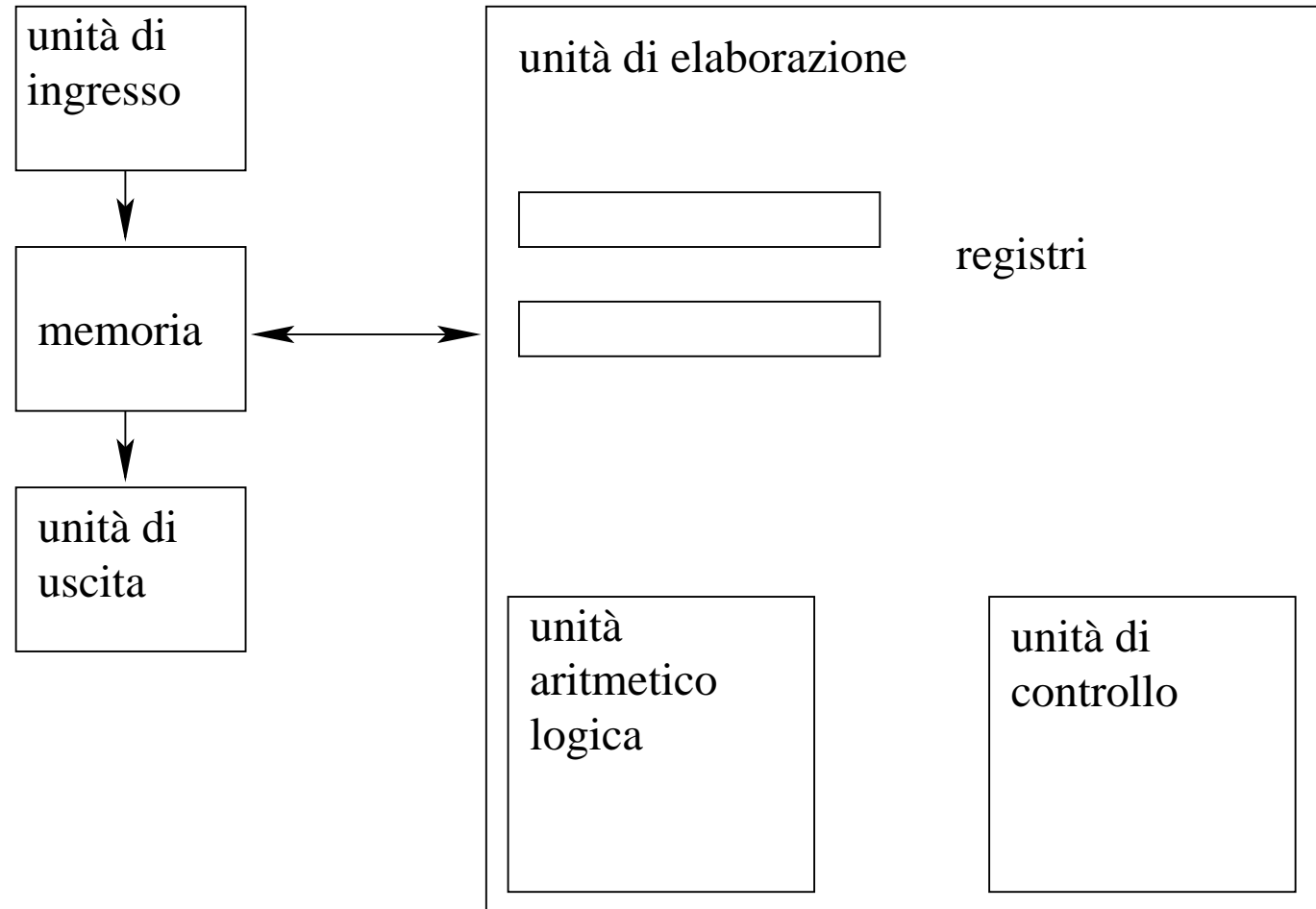


```
program prodotto_interi
var Y,W,Z,U: integer
inizio
  leggi W;
  leggi Y;
  Z = 0;
  U = Y;
  ripeti
    Z = Z+W;
    U = U-1;
  fino a quando U <> 0;
  scrivi Z;
fine
```



# Macchina di von Neumann

- *M1* ha capacità simili alla *macchina di von Neumann*, progettata da von Neumann, Burke e Goldstine a Princeton nel 1947



Algoritmi  
Requisiti per gli algoritmi  
Esempio di flow-chart  
Osservazioni  
Ripetizioni nel flow-chart  
Un esecutore meccanico astratto  
Un algoritmo eseguibile da *M1*  
Macchina di von Neumann  
Caratteristiche della Macchina di von Neumann  
Controllo della sequenza di esecuzione  
Linearizzazione  
Un modello di esecutore CPU  
Istruzioni assemblative  
Istruzioni assemblative  
Linguaggi assemblativi nelle CPU attuali  
Traduzione di istruzioni complesse  
Prodotto di interi in linguaggio assemblativo  
Prodotto di interi in linguaggio assemblativo  
Prodotto di interi in linguaggio assemblativo

# Caratteristiche della Macchina di von Neumann

- La macchina di von Neumann si compone di

**Memoria** Conserva la codifica di dati e programmi ed è costituita da una sequenza di celle identiche, univocamente numerate

**Unità di elaborazione** (*Central Processing Unit*, **CPU**) Si compone di

**Unità aritmetico-logica** (*Arithmetic logic unit*, **ALU**) Esegue le istruzioni aritmetiche (somme, sottrazioni, confronti con zero) e logiche (congiunzione, disgiunzione, negazione) ,

**Registri** Durante l'esecuzione, sono preposti a contenere l'istruzione in esecuzione e gli operandi per le operazioni effettuate dalla ALU, la posizione

**Unità di controllo** Governa la sequenza in cui le istruzioni sono eseguite

**Unità di ingresso e uscita** Colloquiano con l'esterno, trasferendo i contenuti di celle di memoria

**bus di comunicazione** Canali di comunicazione che permettono il trasferimento di dati tra le varie unità

- La architettura della macchina permette l'esecuzione di programmi diversi *senza modificare la macchina stessa o la sua configurazione*

Algoritmi  
Requisiti per gli algoritmi  
Esempio di flow-chart  
Osservazioni  
Ripetizioni nel flow-chart  
Un esecutore meccanico astratto  
Un algoritmo eseguibile da  $M1$   
Macchina di von Neumann  
Caratteristiche della Macchina di von Neumann  
Controllo della sequenza di esecuzione  
Linearizzazione  
Un modello di esecutore CPU  
Istruzioni assemblative  
Istruzioni assemblative  
Linguaggi assemblativi nelle CPU attuali  
Traduzione di istruzioni complesse  
Prodotto di interi in linguaggio assemblativo  
Prodotto di interi in linguaggio assemblativo  
Prodotto di interi in linguaggio assemblativo

# Controllo della sequenza di esecuzione

Algoritmi  
Requisiti per gli algoritmi  
Esempio di flow-chart  
Osservazioni  
Ripetizioni nel flow-chart  
Un esecutore meccanico astratto  
Un algoritmo eseguibile da  $M1$   
Macchina di von Neumann  
Caratteristiche della Macchina di von Neumann  
Controllo della sequenza di esecuzione  
Linearizzazione  
Un modello di esecutore CPU  
Istruzioni assemblative  
Istruzioni assemblative  
Linguaggi assemblativi nelle CPU attuali  
Traduzione di istruzioni complesse  
Prodotto di interi in linguaggio assemblativo  
Prodotto di interi in linguaggio assemblativo  
Prodotto di interi in linguaggio assemblativo

- Con quale principio la unità di controllo governa la sequenza di esecuzione?
- Le istruzioni sono eseguite nell'ordine in cui si trovano memorizzate nella memoria, con un'eccezione:
  - ◆ Se l'istruzione è un'istruzione di *salto condizionato*, l'unità di controllo eseguirà come prossima istruzione quella specificata nell'istruzione di salto, a seconda del risultato di un test specificato sempre dall'istruzione di salto
- Programmi come `prodotto_interi` possono essere riscritti in forma compatibile con il funzionamento dell'unità di controllo
- Infatti, i programmi finora visti possono essere **linearizzati**

# Linearizzazione

Algoritmi  
Requisiti per gli algoritmi  
Esempio di flow-chart  
Osservazioni  
Ripetizioni nel flow-chart  
Un esecutore meccanico astratto  
Un algoritmo eseguibile da  $M1$   
Macchina di von Neumann  
Caratteristiche della Macchina di von Neumann  
Controllo della sequenza di esecuzione  
Linearizzazione  
Un modello di esecutore CPU  
Istruzioni assemblative  
Istruzioni assemblative  
Linguaggi assemblativi nelle CPU attuali  
Traduzione di istruzioni complesse  
Prodotto di interi in linguaggio assemblativo  
Prodotto di interi in linguaggio assemblativo  
Prodotto di interi in linguaggio assemblativo

- Nel linguaggio compreso da  $M1$ , un'istruzione di salto condizionato può essere definita utilizzando una *etichetta* simbolica associata a una istruzione del programma
  - ◆ L'associazione si realizza **scrivendo l'etichetta a sinistra dell'istruzione**
- Associamo a  $Z = Z+W$  l'etichetta PROD

```
program prodotto_interi
var Y,W,Z,U: integer
inizio
    leggi W
    leggi Y
    Z = 0
    U = Y
PROD Z = Z+W
    U = U-1
    se U <> 0 salta a PROD
    scrivi Z
fine
```

# Un modello di esecutore CPU

Algoritmi  
Requisiti per gli algoritmi  
Esempio di flow-chart  
Osservazioni  
Ripetizioni nel flow-chart  
Un esecutore meccanico astratto  
Un algoritmo eseguibile da  $M1$   
Macchina di von Neumann  
Caratteristiche della Macchina di von Neumann  
Controllo della sequenza di esecuzione  
Linearizzazione  
Un modello di esecutore CPU  
Istruzioni assemblative  
Istruzioni assemblative  
Linguaggi assemblativi nelle CPU attuali  
Traduzione di istruzioni complesse  
Prodotto di interi in linguaggio assemblativo  
Prodotto di interi in linguaggio assemblativo  
Prodotto di interi in linguaggio assemblativo

- Il linguaggio compreso è il *linguaggio assemblativo*
- Istruzioni tipiche del linguaggio assemblativo sono il caricamento di dati dalla memoria ai registri, esecuzione di operazioni sui registri, salti condizionati, trasferimento di dati tra l'esterno e la memoria
- Il linguaggio assemblativo **non** è ad alto livello
  - ◆ Un'istruzione in linguaggio ad alto livello come il linguaggio compreso da  $M1$  corrisponde a piú istruzioni in linguaggio assemblativo

# Istruzioni assemblative

Algoritmi  
Requisiti per gli algoritmi  
Esempio di flow-chart  
Osservazioni  
Ripetizioni nel flow-chart  
Un esecutore meccanico astratto  
Un algoritmo eseguibile da  $M1$   
Macchina di von Neumann  
Caratteristiche della Macchina di von Neumann  
Controllo della sequenza di esecuzione  
Linearizzazione  
Un modello di esecutore CPU  
Istruzioni assemblative  
Istruzioni assemblative  
Linguaggi assemblativi nelle CPU attuali  
Traduzione di istruzioni complesse  
Prodotto di interi in linguaggio assemblativo  
Prodotto di interi in linguaggio assemblativo  
Prodotto di interi in linguaggio assemblativo

caricamento di un operando	LOAD in R1 A LOAD in R2 A	copia nel registro R1 o R2 il contenuto della cella di memoria identificata da A
operazioni aritmetiche sui registri	ADD, SUB	ADD (SUB) somma (sottrae) il contenuto del registro R2 al registro R1 e memorizza il risultato in R1
test (salto condizionato)	JUMP to A	se $R1 \neq 0$ , la prossima istruzione da eseguire dovrà essere caricata dalla cella associata al nome simbolico A
memorizzazione dei registri	STORE R1 to A STORE R2 to A	memorizza il contenuto del registro R1 o R2 nella cella di memoria associata al nome simbolico A

# Istruzioni assemblative

Algoritmi  
Requisiti per gli algoritmi  
Esempio di flow-chart  
Osservazioni  
Ripetizioni nel flow-chart  
Un esecutore meccanico astratto  
Un algoritmo eseguibile da  $M1$   
Macchina di von Neumann  
Caratteristiche della Macchina di von Neumann  
Controllo della sequenza di esecuzione  
Linearizzazione  
Un modello di esecutore CPU  
Istruzioni assemblative  
Istruzioni assemblative  
Linguaggi assemblativi nelle CPU attuali  
Traduzione di istruzioni complesse  
Prodotto di interi in linguaggio assemblativo  
Prodotto di interi in linguaggio assemblativo  
Prodotto di interi in linguaggio assemblativo

acquisizione di un dato dall'esterno	READ A	leggi un dato e memorizzalo nella cella di memoria identificata dalla lettera A
invio di un dato all'esterno	WRITE A	scrivi il valore contenuto nella cella identificata dal nome simbolico A
Arresta l'esecuzione	STOP	arresta l'esecuzione del programma

# Linguaggi assemblativi nelle CPU attuali

- Algoritmi
- Requisiti per gli algoritmi
- Esempio di flow-chart
- Osservazioni
- Ripetizioni nel flow-chart
- Un esecutore meccanico astratto
- Un algoritmo eseguibile da  $M1$
- Macchina di von Neumann
- Caratteristiche della Macchina di von Neumann
- Controllo della sequenza di esecuzione
- Linearizzazione
- Un modello di esecutore CPU
- Istruzioni assemblative
- Istruzioni assemblative
- Linguaggi assemblativi nelle CPU attuali
- Traduzione di istruzioni complesse
- Prodotto di interi in linguaggio assemblativo
- Prodotto di interi in linguaggio assemblativo
- Prodotto di interi in linguaggio assemblativo

- Le CPU reali memorizzano i programmi in forma di sequenze di numeri binari
- Ogni produttore rende disponibili
  - ◆ Le descrizioni complete delle istruzioni in codice binario e delle loro funzioni; l'insieme è detto *linguaggio assoluto*, o *linguaggio macchina*
  - ◆ La descrizione di un linguaggio assemblativo specifico
  - ◆ Ogni istruzione di linguaggio assemblativo corrisponde una istruzione in linguaggio macchina
- Alcune CPU molto diffuse condividono gran parte dei propri linguaggi assemblativi, e quindi macchina
  - ◆ Es. Intel Pentium; AMD K6, Duron, Athlon, Sempron



# Traduzione di istruzioni complesse

Algoritmi  
Requisiti per gli algoritmi  
Esempio di flow-chart  
Osservazioni  
Ripetizioni nel flow-chart  
Un esecutore meccanico astratto  
Un algoritmo eseguibile da  $M1$   
Macchina di von Neumann  
Caratteristiche della Macchina di von Neumann  
Controllo della sequenza di esecuzione  
Linearizzazione  
Un modello di esecutore CPU  
Istruzioni assemblative  
Istruzioni assemblative  
Linguaggi assemblativi nelle CPU attuali  
Traduzione di istruzioni complesse  
Prodotto di interi in linguaggio assemblativo  
Prodotto di interi in linguaggio assemblativo  
Prodotto di interi in linguaggio assemblativo

- Non vi è corrispondenza biunivoca tra le istruzioni che compaiono nel programma di prodotto di interi espresso in linguaggio ad alto livello e le istruzioni assemblative
- Le istruzioni di un linguaggio ad alto livello sono più complesse e corrispondono a più istruzioni assemblative
- Ad esempio, l'istruzione  $Z = Z + W$  richiede in linguaggio assemblativo
  - ◆ due accessi in lettura alla memoria per copiare i contenuti delle celle corrispondenti a  $Z$  e  $W$  in due registri
  - ◆ la somma tra i registri
  - ◆ un accesso in scrittura per copiare il contenuto del registro risultato nella cella di memoria corrispondente a  $Z$

Istruzione ad alto livello	Sequenza di istruzioni assemblative
$Z = Z + W$	LOAD in R1 Z LOAD in R2 W ADD STORE R1 to Z

# Prodotto di interi in linguaggio assemblativo

Algoritmi  
 Requisiti per gli algoritmi  
 Esempio di flow-chart  
 Osservazioni  
 Ripetizioni nel flow-chart  
 Un esecutore meccanico astratto  
 Un algoritmo eseguibile da  $M1$   
 Macchina di von Neumann  
 Caratteristiche della Macchina di von Neumann  
 Controllo della sequenza di esecuzione  
 Linearizzazione  
 Un modello di esecutore CPU  
 Istruzioni assemblative  
 Istruzioni assemblative  
 Linguaggi assemblativi nelle CPU attuali  
 Traduzione di istruzioni complesse  
 Prodotto di interi in linguaggio assemblativo  
 Prodotto di interi in linguaggio assemblativo  
 Prodotto di interi in linguaggio assemblativo

riga	ind. simb.	istruzione	significato	R1	R2	W	Y	Z	U
0		READ W	leggi W	-	-	3	-	-	-
1		READ Y	leggi Y	-	-	3	2	-	-
2		LOAD in R1 ZERO	$Z = 0$	0	-	3	2	-	-
3		STORE R1 to Z	$Z = 0$	0	-	3	2	0	-
4		LOAD in R1 Y	$U = Y$	2	-	3	2	0	-
5		STORE R1 to U	$U = Y$	2	-	3	2	0	2
6	PROD	LOAD IN R1 Z	$Z = Z+W$	0	-	3	2	0	2
7		LOAD IN R2 W	$Z = Z+W$	0	3	3	2	0	2
8		ADD	$Z = Z+W$	3	3	3	2	0	2
9		STORE R1 to Z	$Z = Z+W$	3	3	3	2	3	2
10		LOAD IN R1 U	$U = U-1$	2	3	3	2	3	2
11		LOAD IN R2 UNO	$U = U-1$	2	1	3	2	3	2
12		SUB	$U = U-1$	1	1	3	2	3	2
13		STORE R1 to U	$U = U-1$	1	1	3	2	3	1

# Prodotto di interi in linguaggio assemblativo

Algoritmi  
 Requisiti per gli algoritmi  
 Esempio di flow-chart  
 Osservazioni  
 Ripetizioni nel flow-chart  
 Un esecutore meccanico astratto  
 Un algoritmo eseguibile da  $M1$   
 Macchina di von Neumann  
 Caratteristiche della Macchina di von Neumann  
 Controllo della sequenza di esecuzione  
 Linearizzazione  
 Un modello di esecutore CPU  
 Istruzioni assemblative  
 Istruzioni assemblative  
 Linguaggi assemblativi nelle CPU attuali  
 Traduzione di istruzioni complesse  
 Prodotto di interi in linguaggio assemblativo  
 Prodotto di interi in linguaggio assemblativo  
 Prodotto di interi in linguaggio assemblativo

riga	ind. simb.	istruzione	significato	R1	R2	W	Y	Z	U
14		LOAD IN R1 U	se $U \neq 0$ salta a PROD	1	1	3	2	3	1
15		JUMP to PROD	se $U \neq 0$ salta a PROD	1	1	3	2	3	1
6	PROD	LOAD in R1 Z	$Z = Z+W$	3	1	3	2	3	1
7		LOAD in R2 W	$Z = Z+W$	3	3	3	2	3	1
8		ADD	$Z = Z+W$	6	3	3	2	3	1
9		STORE R1 to Z	$Z = Z+W$	6	3	3	2	6	1
10		LOAD in R1 U	$U = U-1$	1	3	3	2	6	1
11		LOAD in R2 UNO	$U = U-1$	1	1	3	2	6	1
12		SUB	$U = U-1$	0	1	3	2	6	1
13		STORE R1 to U	$U = U-1$	0	1	3	2	6	0
14		LOAD in R1 U	se $U \neq 0$ salta a PROD	0	1	3	2	6	0
15		JUMP to PROD	se $U \neq 0$ salta a PROD	0	1	3	2	6	0

# Prodotto di interi in linguaggio assembleativo

Algoritmi  
Requisiti per gli algoritmi  
Esempio di flow-chart  
Osservazioni  
Ripetizioni nel flow-chart  
Un esecutore meccanico astratto  
Un algoritmo eseguibile da  $M1$   
Macchina di von Neumann  
Caratteristiche della Macchina di von Neumann  
Controllo della sequenza di esecuzione  
Linearizzazione  
Un modello di esecutore CPU  
Istruzioni assembleative  
Istruzioni assembleative  
Linguaggi assembleativi nelle CPU attuali  
Traduzione di istruzioni complesse  
Prodotto di interi in linguaggio assembleativo  
Prodotto di interi in linguaggio assembleativo  
Prodotto di interi in linguaggio assembleativo

riga	ind. simb.	istruzione	significato	R1	R2	W	Y	Z	U
16		WRITE Z		1	1	3	2	3	1
17		STOP		1	1	3	2	3	1