



CAPITOLO 6:

Il Little Man Computer

The Architecture of Computer Hardware, Systems Software & Networking: An Information Technology Approach

5th Edition, Irv Englander

John Wiley and Sons ©2013

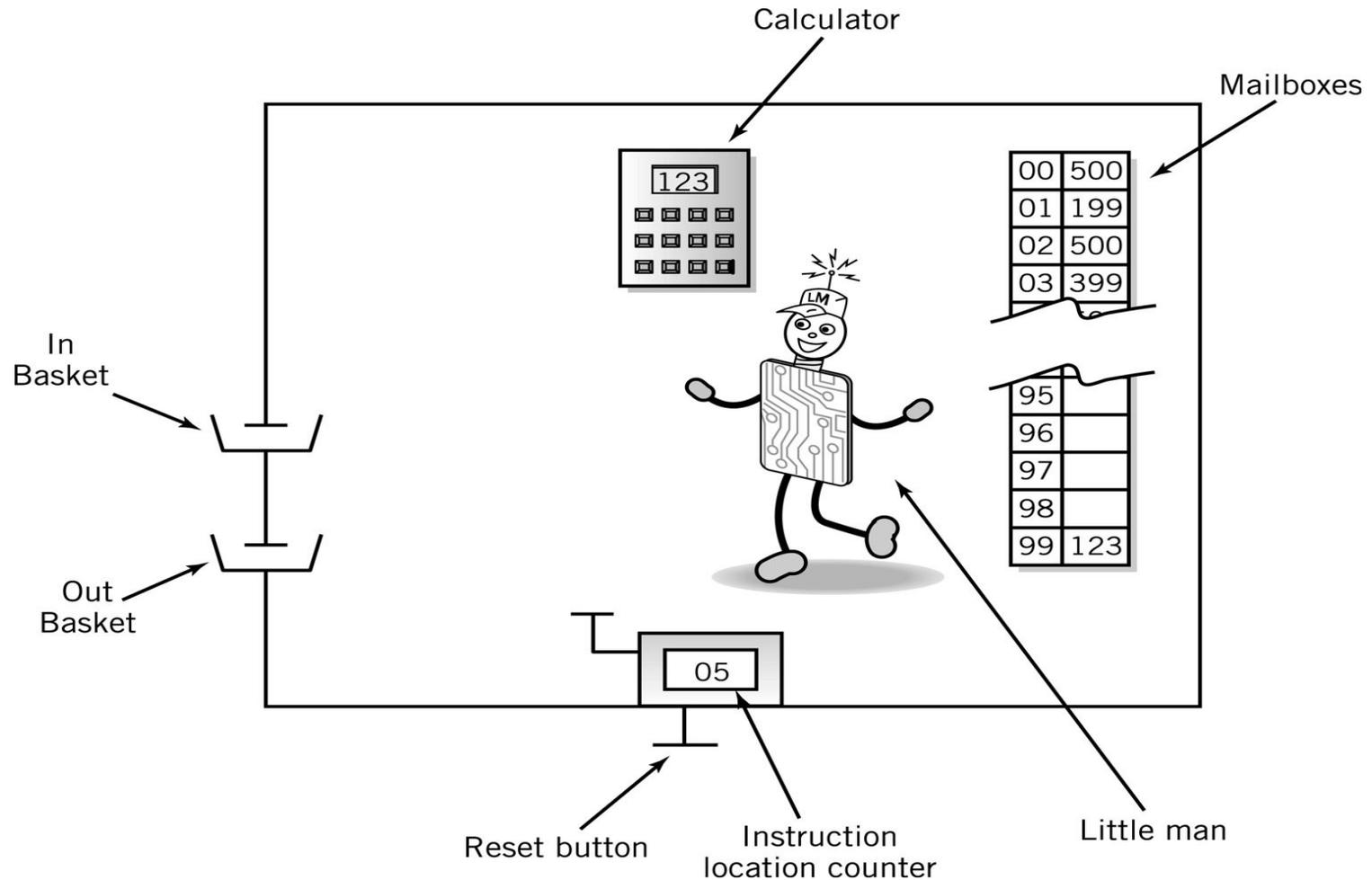
Diapositive realizzate da Angela Clark, University of South Alabama

Diapositive per la 4° edizione realizzate da Wilson Wong, Bentley University

Diapositive per il CLEII tradotte e adattate da Gianluca Amato, Università CH-PE



II Little Man Computer





Mailbox: indirizzi / contenuto

- Gli indirizzi sono consecutivi, partono da 00 e terminano a 99
- Il contenuto può essere
 - Un dato, ovvero un numero a tre cifre, o
 - Una istruzione

Indirizzo	Contenuto
00	500
01	345



Contenuto: istruzioni

- Op code
 - In LMC è rappresentato da un'unica cifra
 - Codice operativo
- Operando
 - In LMC, è rappresentato dalle due cifre che seguono l'opcode
 - È l'indirizzo di una locazione di memoria

Istruzione

Op code	Operando
3	25

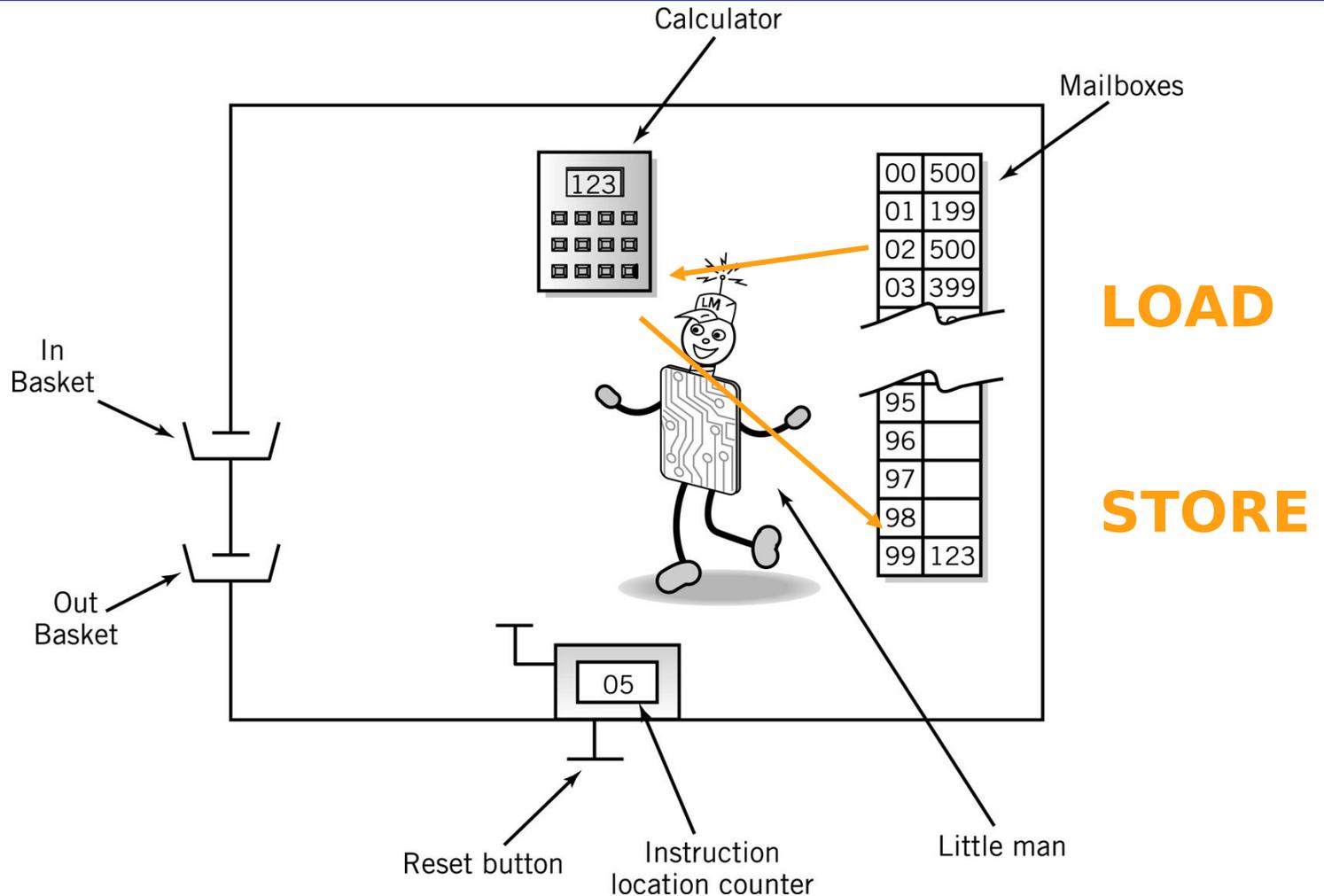


Insieme di istruzioni

Aritmetica	1xx	ADD
	2xx	SUBTRACT
Spostamento dati	3xx	STORE
	5xx	LOAD
Input/Output	901	INPUT
	902	OUTPUT
Controllo macchina	000	HALT



LMC Spostamento dati





Spostamento dati

- Tra mailbox e calcolatrice

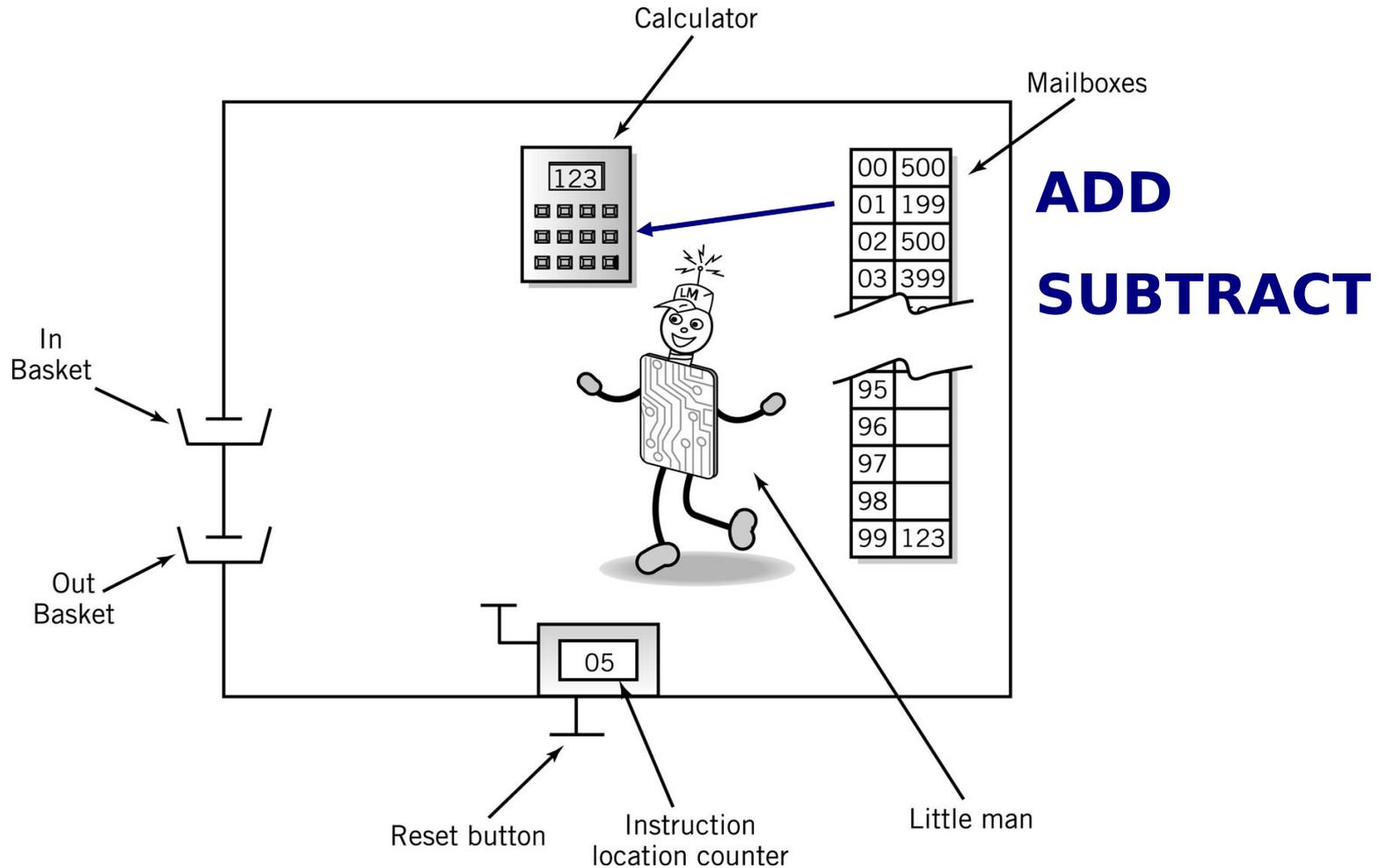
STA (store
accumulator)

LDA (load
accumulator)

Content	
Op Code	Operand (address)
3	XX
5	XX



LMC Istruzioni aritmetiche





Istruzioni aritmetiche

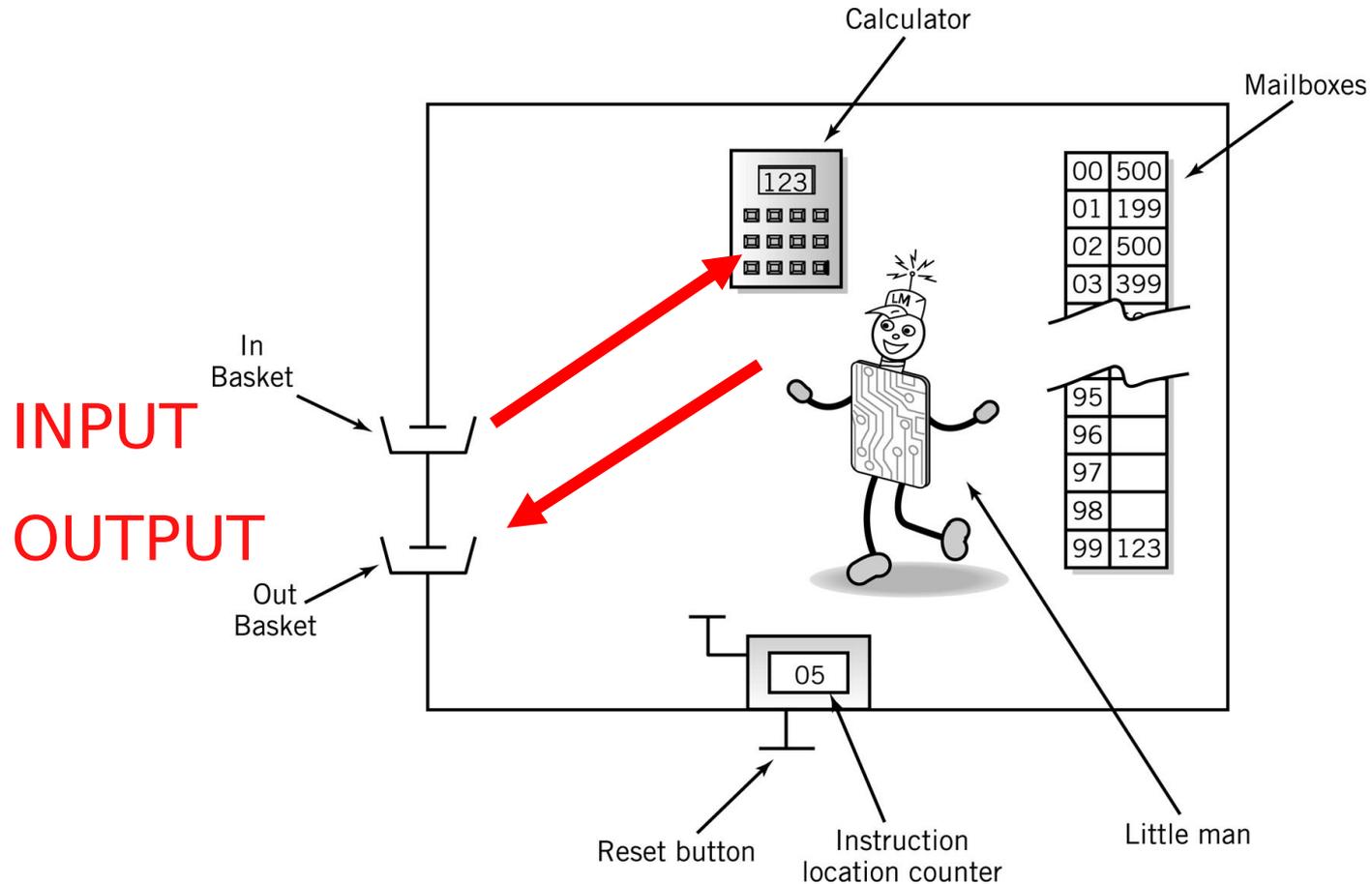
- Legge la mailbox
- Esegue l'istruzione sulla calcolatrice

Content

	Op Code	Operand (address)
ADD (add)	1	XX
SUB (subtract)	2	XX



LMC Input/Output





Input/Output

- Sposta dati tra la calcolatrice e i cestini

Content

	Op Code	Operand (address)
INP (input)	9	01
OUT (output)	9	02



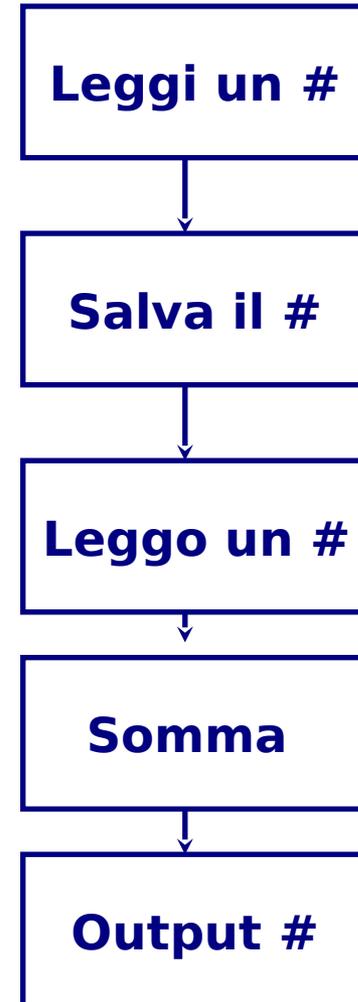
Contenuto: dati

- Fisicamente identiche alle celle che contengono le istruzioni
- Localizzate al di fuori della sequenza di istruzioni



Esempio: somma di 2 numeri

- Assumiamo che i dati siano memorizzati nelle celle con indirizzo > 90
- Scriviamo le istruzioni





Somma di 2 numeri

Mailbox	Code	Instruction Description
00	901	;input 1° numero
01	399	;memorizza dato
02	901	;input 2° numero
03	199	;somma 1° # al 2° #
04	902	;output risultato
05	000	;fermati
99	000	;dato



Linguaggio assembly

- *Codici mnemonici* (piccole sequenze di caratteri) rappresentano istruzioni
 - Corrispondenza 1 a 1 tra istruzioni del linguaggio assembly e istruzioni del linguaggio macchina
 - Specifico per ogni CPU (o famiglia di CPU)
- La traduzione da assembly a linguaggio macchina è svolta dall'assemblatore.



Somma di 2 numeri in assembly

Mailbox	Mnemonic	Instruction Description
00	INP	;input 1° numero
01	STA 99	;memorizza dato
02	INP	;input 2° numero
03	ADD 99	;somma 1° # al 2° #
04	OUT	;output risultato
05	HLT	;fermati
99	DAT 000	;dato



Locazioni per i dati

- È possibile riservare una locazione per i dati con la **pseudo-istruzione** DAT
 - DAT non è una istruzione, non viene eseguita dalla CPU, ma è riconosciuta dal programma assemblatore
 - serve solo a dare un nome ad una locazione di memoria e a metterci dentro un valore PRIMA che il programma inizi l'esecuzione



Controllo di flusso

- Salto (esecuzione di una istruzione fuori sequenza)
 - Cambia l'indirizzo del contatore di programma

Content

BRA (Jump)

BRZ (Branch on 0)

BRP (Branch on +)

Op Code	Operand (address)
6	XX
7	XX
8	XX



Insieme di istruzioni rivisto

Aritmetica	1xx	ADD
	2xx	SUB
Spostamento dati	3xx	STORE
	5xx	LOAD
BRA	6xx	JUMP
BRZ	7xx	BRANCH ON 0
BRP	8xx	BRANCH ON +
Input/Output	901	INPUT
	902	OUTPUT
Controllo macchina	000	HALT

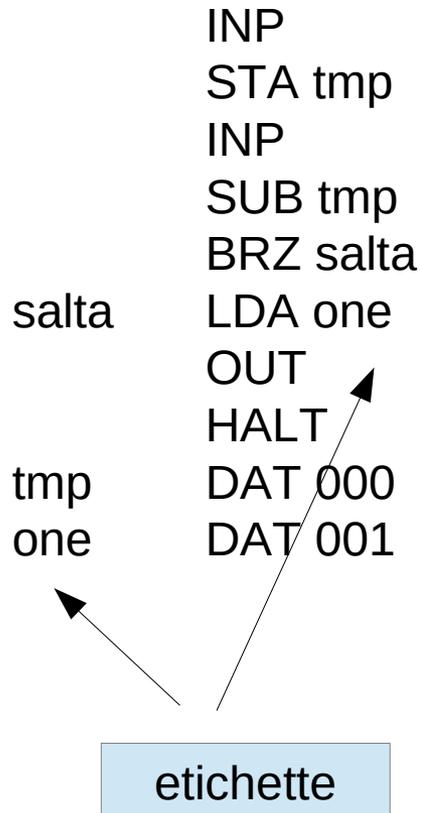


Trova la differenza positiva di 2 numeri

00	INP	901	
01	STA 10	310	
02	INP	901	
03	STA 11	311	
04	SUB 10	210	
05	BRP 08	808	;test
06	LDA 10	510	;se negativo, cambia ordine
07	SUB 11	211	
08	OUT	902	;stampa risultato e
09	HLT	000	;fermati
10	DAT	000	;usato per i dati
11	DAT	000	;usato per i dati



Traccia di esecuzione



PC prima	PC dopo	A	tmp	one
--	0	0	0	1
0	1	23	0	1
1	2	23	23	1
2	3	12	23	1
3	4	-9	23	1
4	5	-9	23	1
5	6	1	23	1
6	7	1	23	1
7	--	1	23	1

Inputs: 23 12

Outputs: 1



Programma con un ciclo

Dato un input n , restituisce $0 + 1 + 2 + \dots + n$

```

                INP
ciclo           STA count
                LDA somma
                ADD count
                STA somma
                LDA count
                SUB one
                BRZ fine
                BRA ciclo
fine           LDA somma
                OUT
                HALT
somma          DAT 000
count          DAT 000
one            DAT 001
```

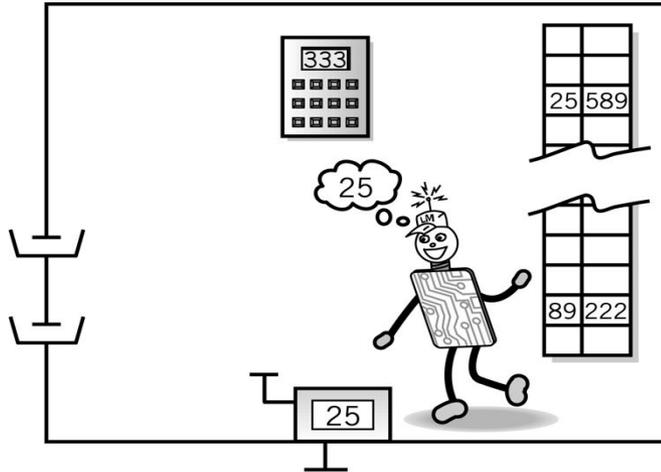


Ciclo delle istruzioni

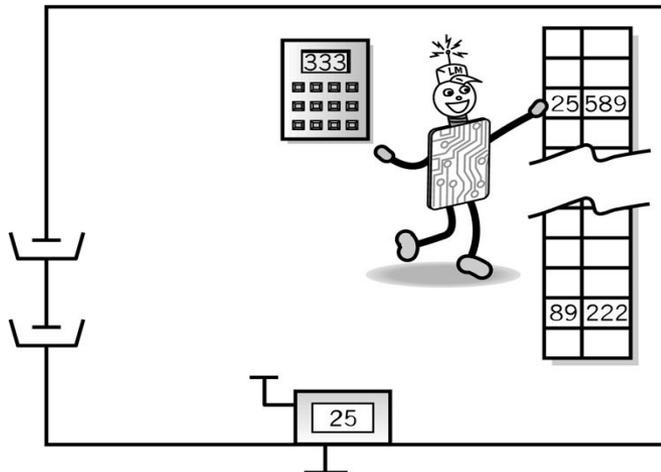
- *Fetch*: Il Little Man determina quale istruzione deve eseguire
- *Execute*: Il Little Man esegue l'istruzione
 - Il lavoro da fare dipende dall'istruzione letta



La fase di fetch



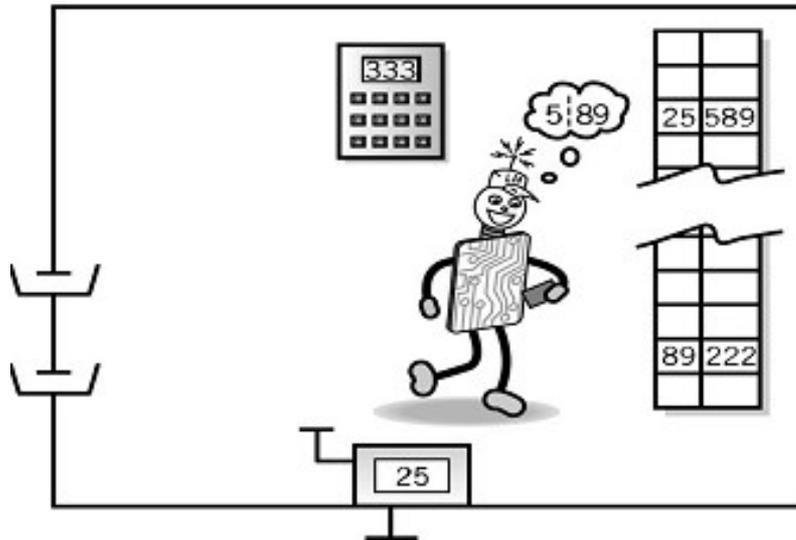
1. Il Little Man legge l'indirizzo dal location counter



2. Si dirige alla casella il cui indirizzo è contenuto nel location counter



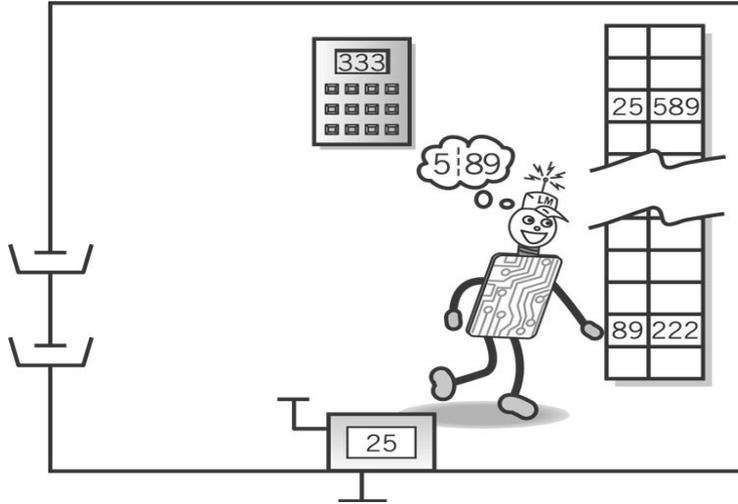
Fetch, continua



3. E legge il numero presente sul foglio di carta (rimette a posto il foglio di carta nel caso ne abbia bisogno di nuovo

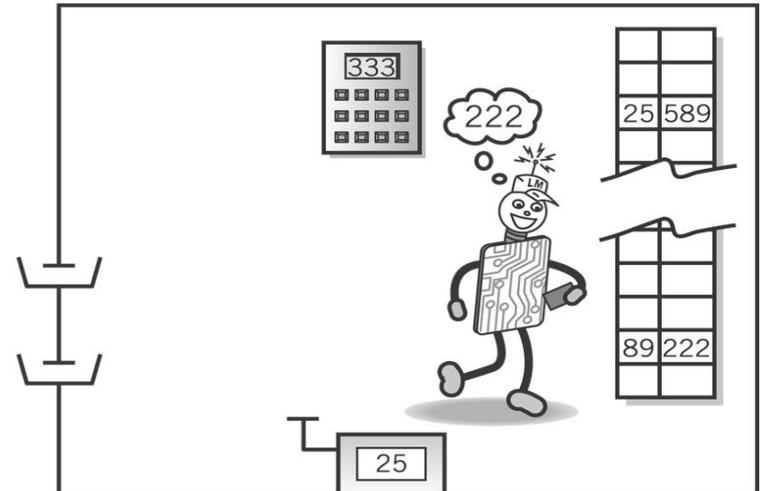


La fase di execute



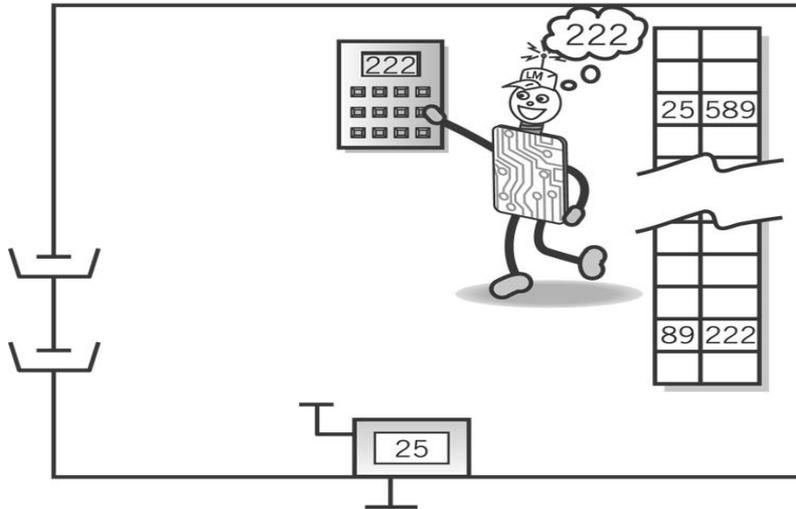
1. Il Little Man va all'indirizzo della mailbox specificato nella istruzione appena letta

2. Legge il numero in quella cella (e rimette il foglio a posto, nel caso serva più tardi)



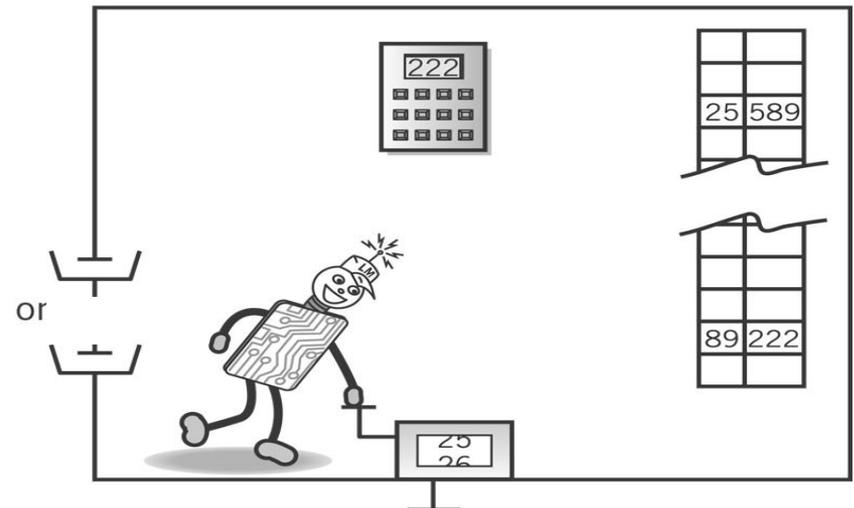


Execute, cont.



3. Va alla calcolatrice e digita il numero appena letto

4. Va al location counter e preme il pulsante, preparandosi così al successivo ciclo di fetch/execute





Architettura di von Neumann (1945)

- Concetto di programma memorizzato
- La memoria è indirizzata in maniera lineare
- La memoria è indirizzata in maniera indipendente dal contenuto



Copyright 2013 John Wiley & Sons

All rights reserved. Reproduction or translation of this work beyond that permitted in section 117 of the 1976 United States Copyright Act without express permission of the copyright owner is unlawful. Request for further information should be addressed to the Permissions Department, John Wiley & Sons, Inc. The purchaser may make back-up copies for his/her own use only and not for distribution or resale. The Publisher assumes no responsibility for errors, omissions, or damages caused by the use of these programs or from the use of the information contained herein.”