Chap 2: Data processing

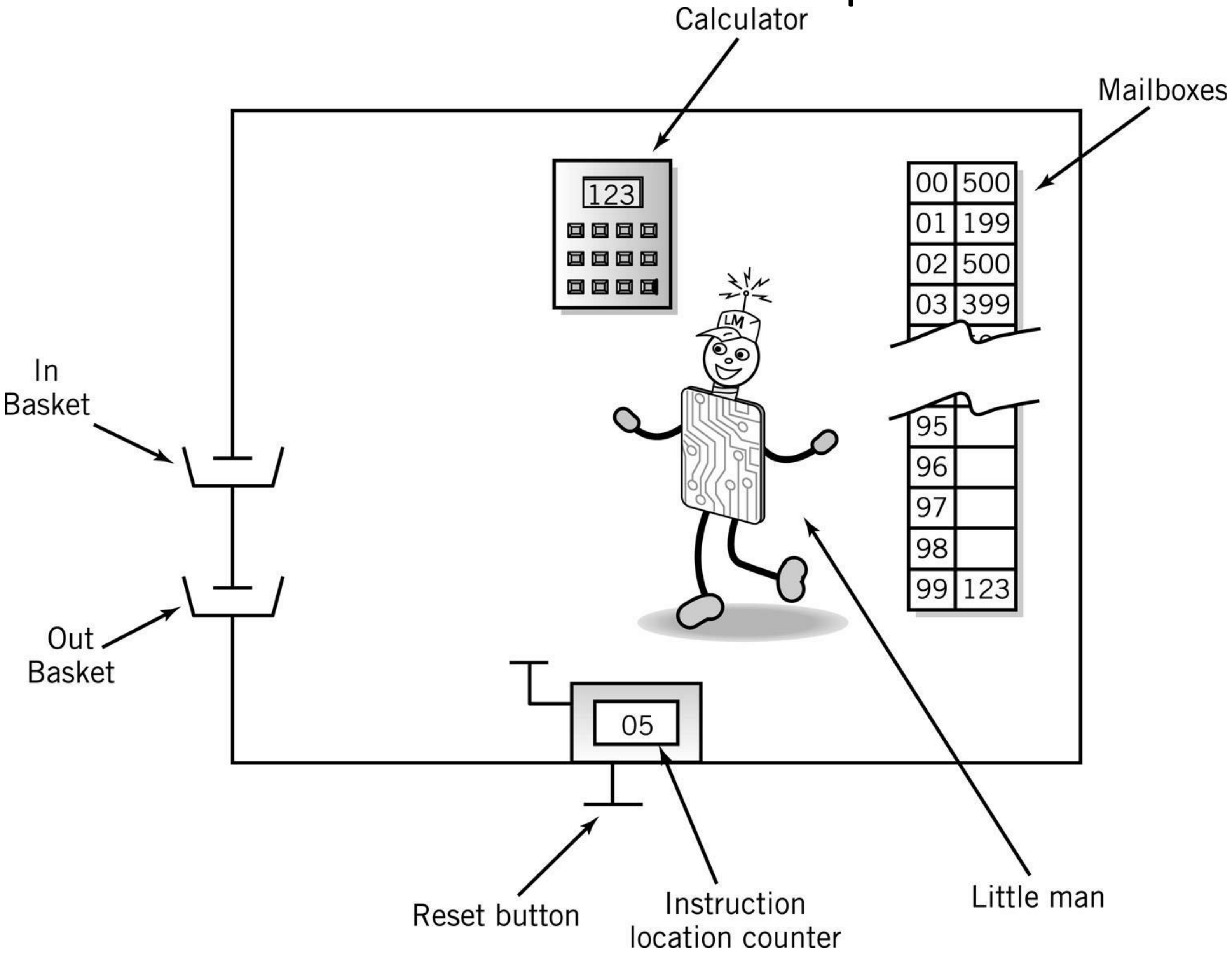
1. Numerical system
2. Data Representation and data Formats
3. Data processing systems

Link to the course material: https://drive.google.com/open?id=0BxS73bgp4UGdMktGakVRZ2pMamc

2.3 Data processing systems

Link to the course material: https://drive.google.com/open?id=0BxS73bgp4UGdMktGakVRZ2pMamc

Little man computer



Mailboxes: Address vs. Content

- **Addresses** are consecutive starting at 00 and ending at 99
- **Content** may be:

    - **Data**, a *three digit number*, or

    - **Instructions**

Address	Content

Content: Instructions

Op code: In LMC, represented by a *single digit*

Operation code

Arbitrary mnemonic

Operand: In LMC, represented by *two digits following* the op code

- Object to be manipulated

  - **Data** or
  - Address of data

Address	**Content**	
	Op code	Operand

Assembly Language

- Specific to a CPU
- 1 to 1 correspondence between assembly language instruction and binary (machine) language instruction
- *Mnemonics*(short character sequence) represent instructions
- Used when programmer needs precise control over hardware, e.g., device drivers

Instruction Set

Arithmetic

1xx	ADD
2xx	SUB

Data Movement

3xx	STORE
5xx	LOAD

Input/Output

901	INPUT
902	Output

Machine Control (coffee break)

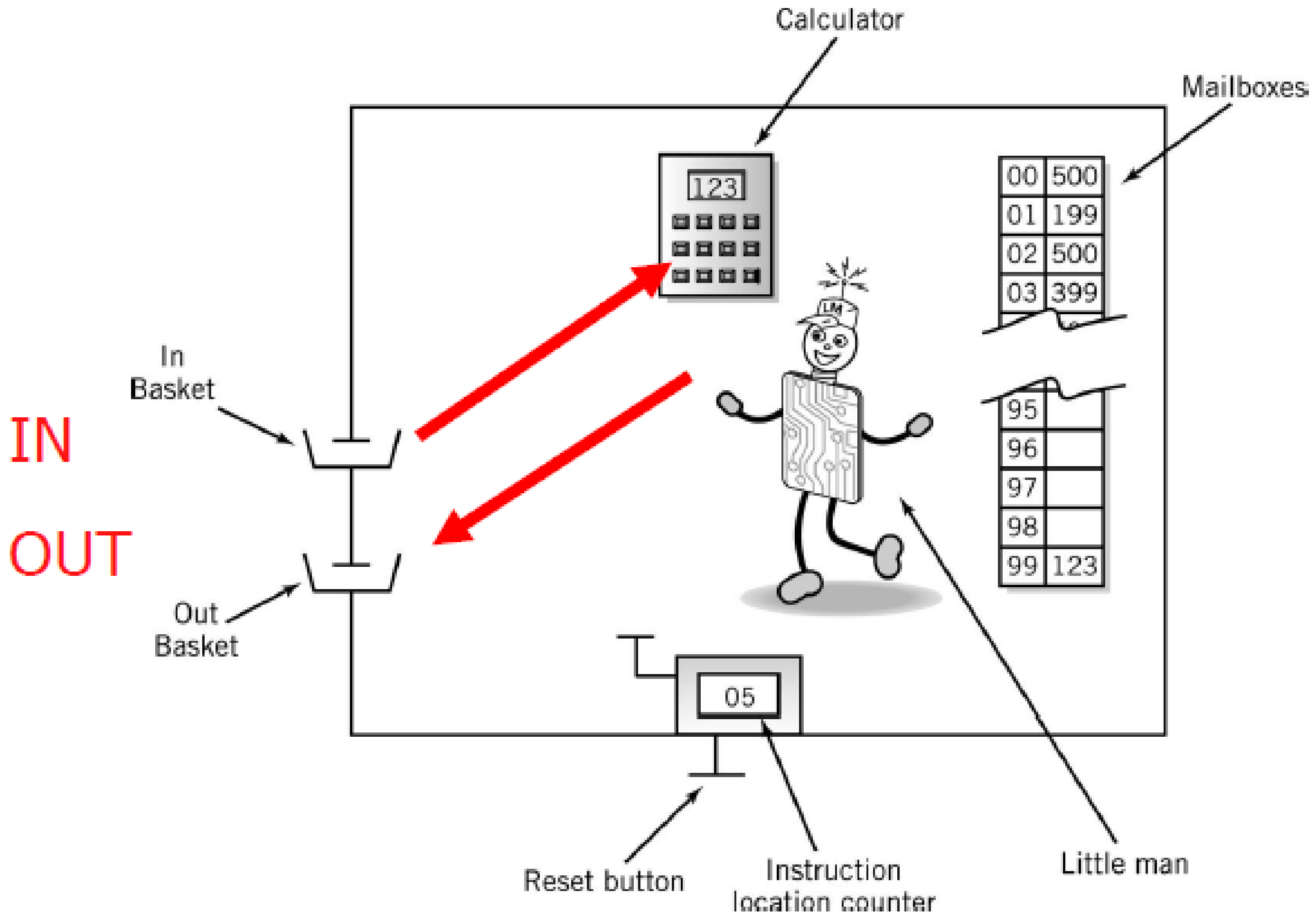
000	HALT
	COB

Input/Output

Move data between calculator and in/out baskets

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

LMC Input/Output

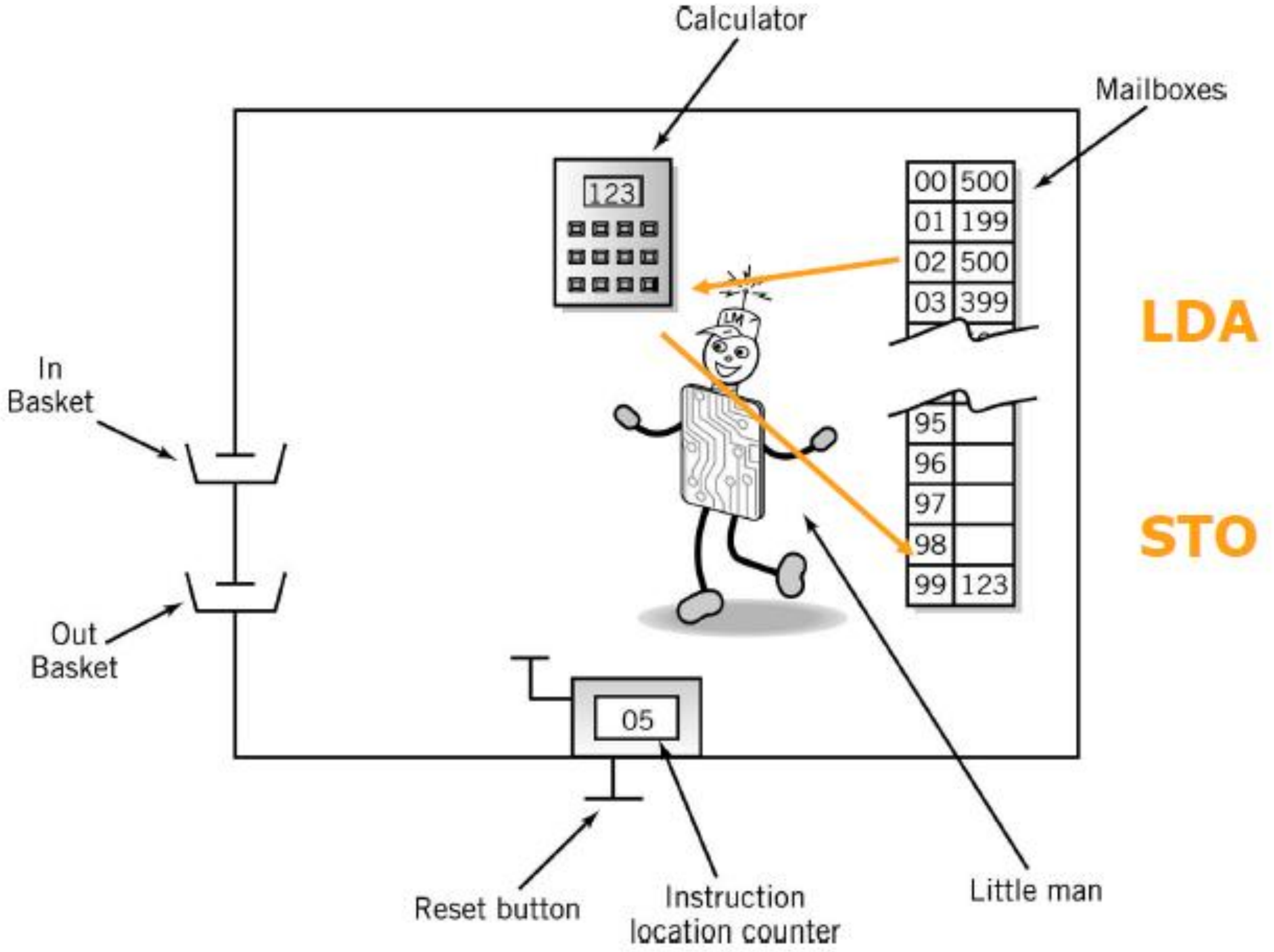


Internal Data Movement

Between mailbox and calculator

	Content	
	Op Code	Operand (address)
STO (store)	3	xx
LDA (load)	5	xx

LMC Internal Data Movement



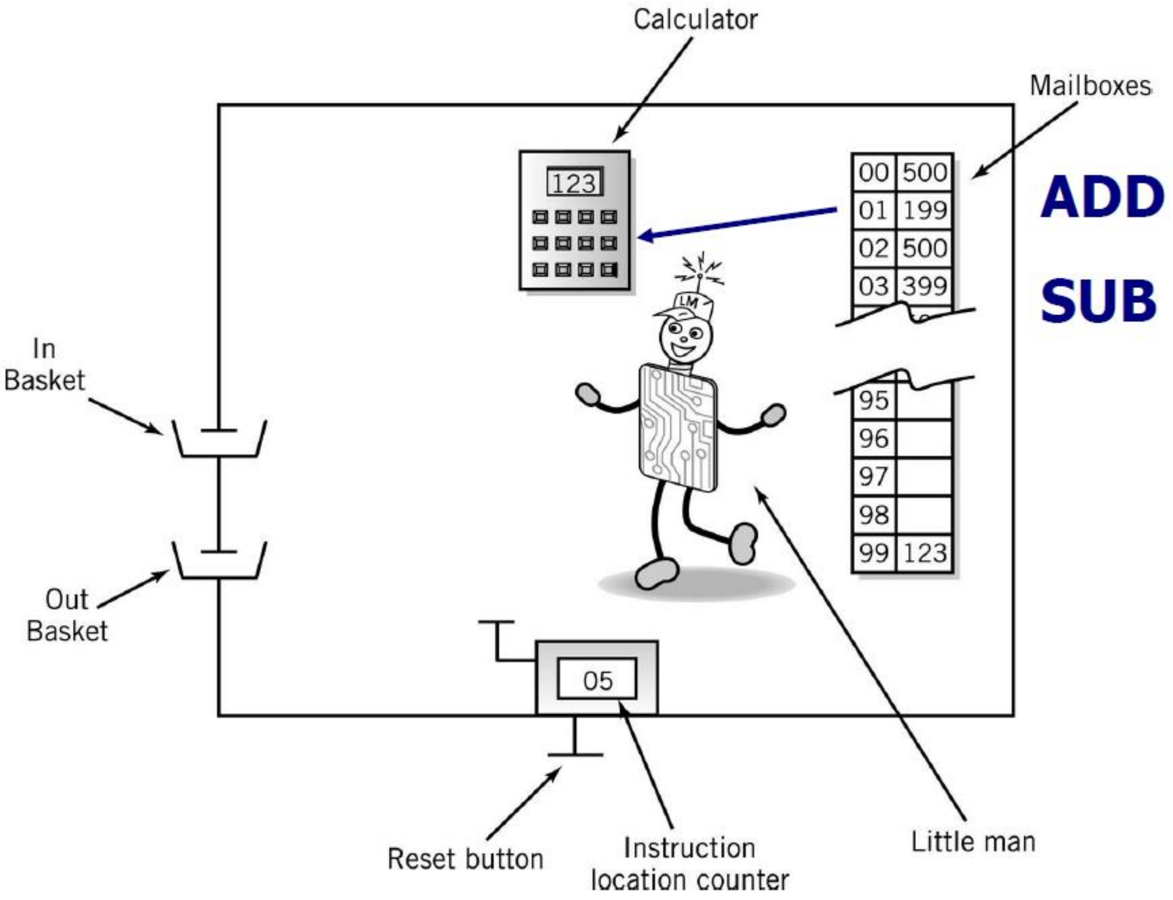
Arithmetic Instructions

Read mailbox

Perform operation in the calculator

	Content	
	Op Code	Operand (address)
ADD	1	xx
SUB	2	xx

LMC Arithmetic Instructions



00	500
01	199
02	500
03	399
95	
96	
97	
98	
99	123

ADD

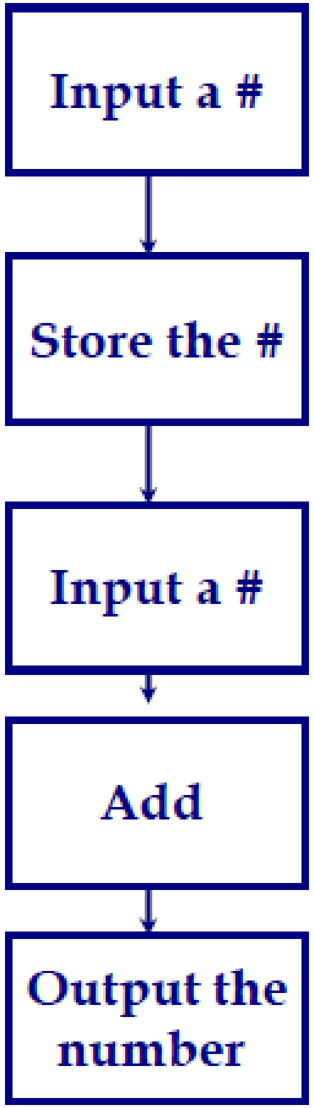
SUB

Data storage location

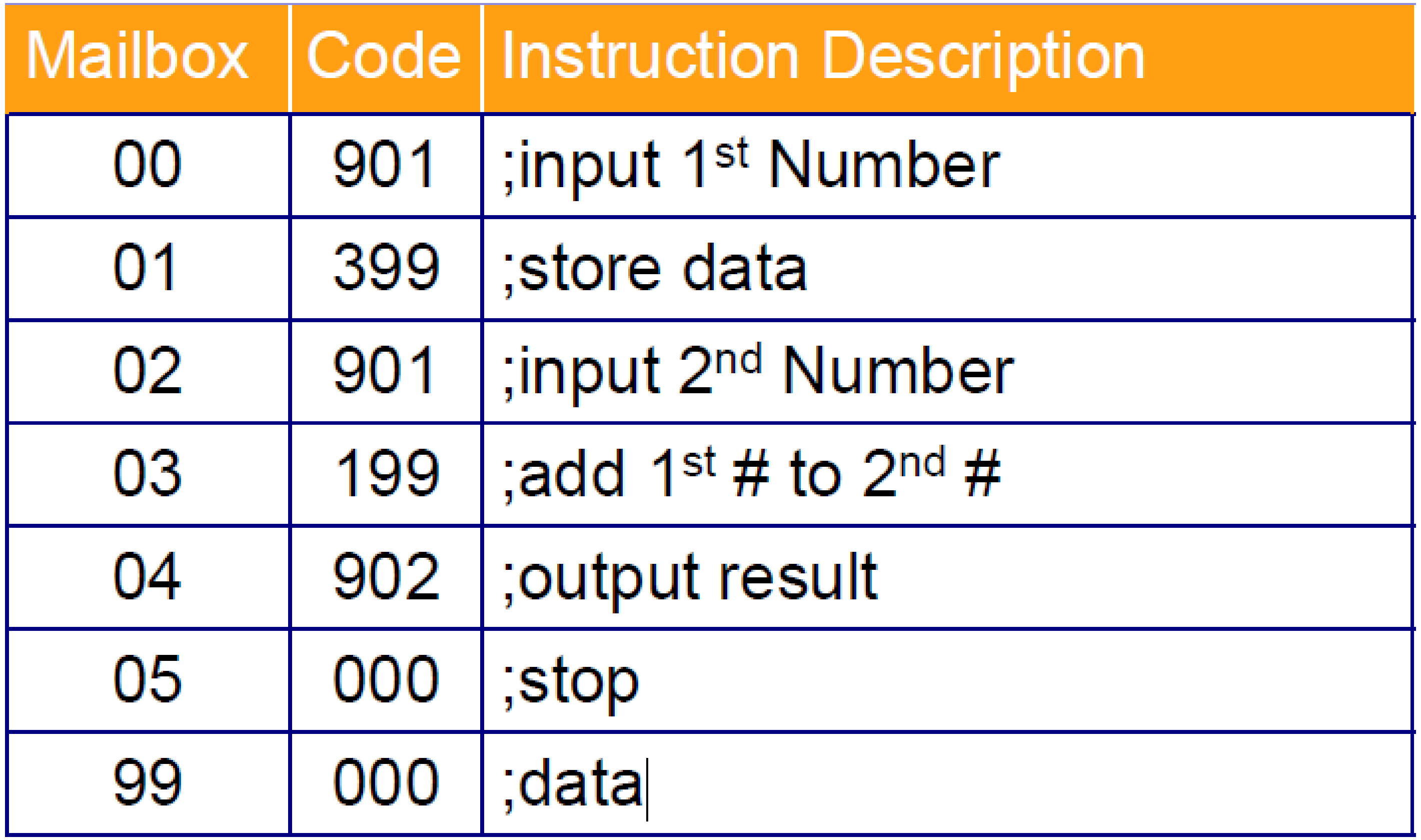
- Physically identical to instruction mailbox
- Not located in instruction sequence
- Identified by *DAT* mnemonic

Simple Program: Add 2 Numbers

- Assume data is stored in mailboxes with addresses >90
- Write instructions

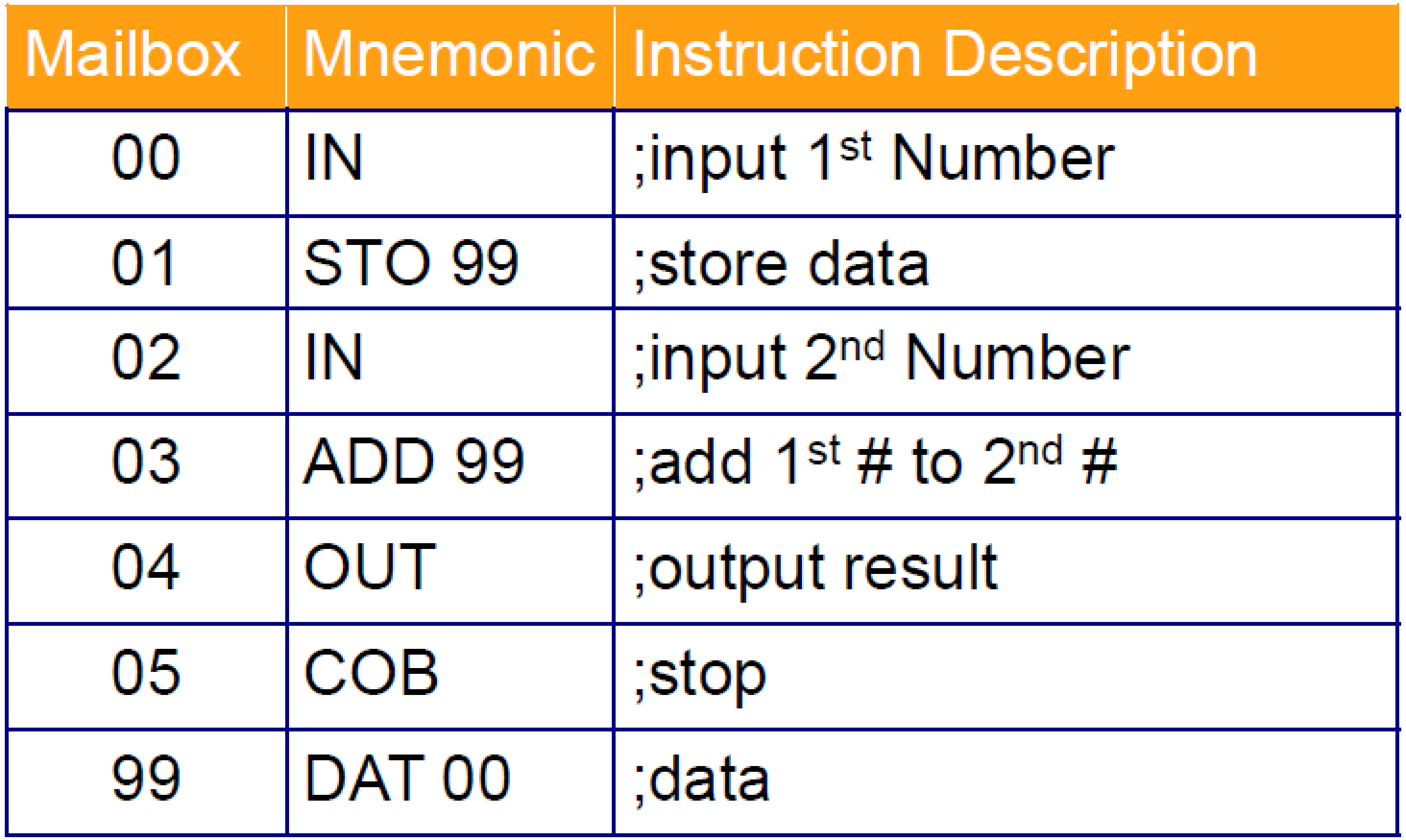


Program to Add 2 Numbers



Mailbox	Code	Instruction Description
00	901	;input 1st Number
01	399	;store data
02	901	;input 2nd Number
03	199	;add 1st # to 2nd #
04	902	;output result
05	000	;stop
99	000	;data

Program to Add 2 Numbers : Using Mnemonics



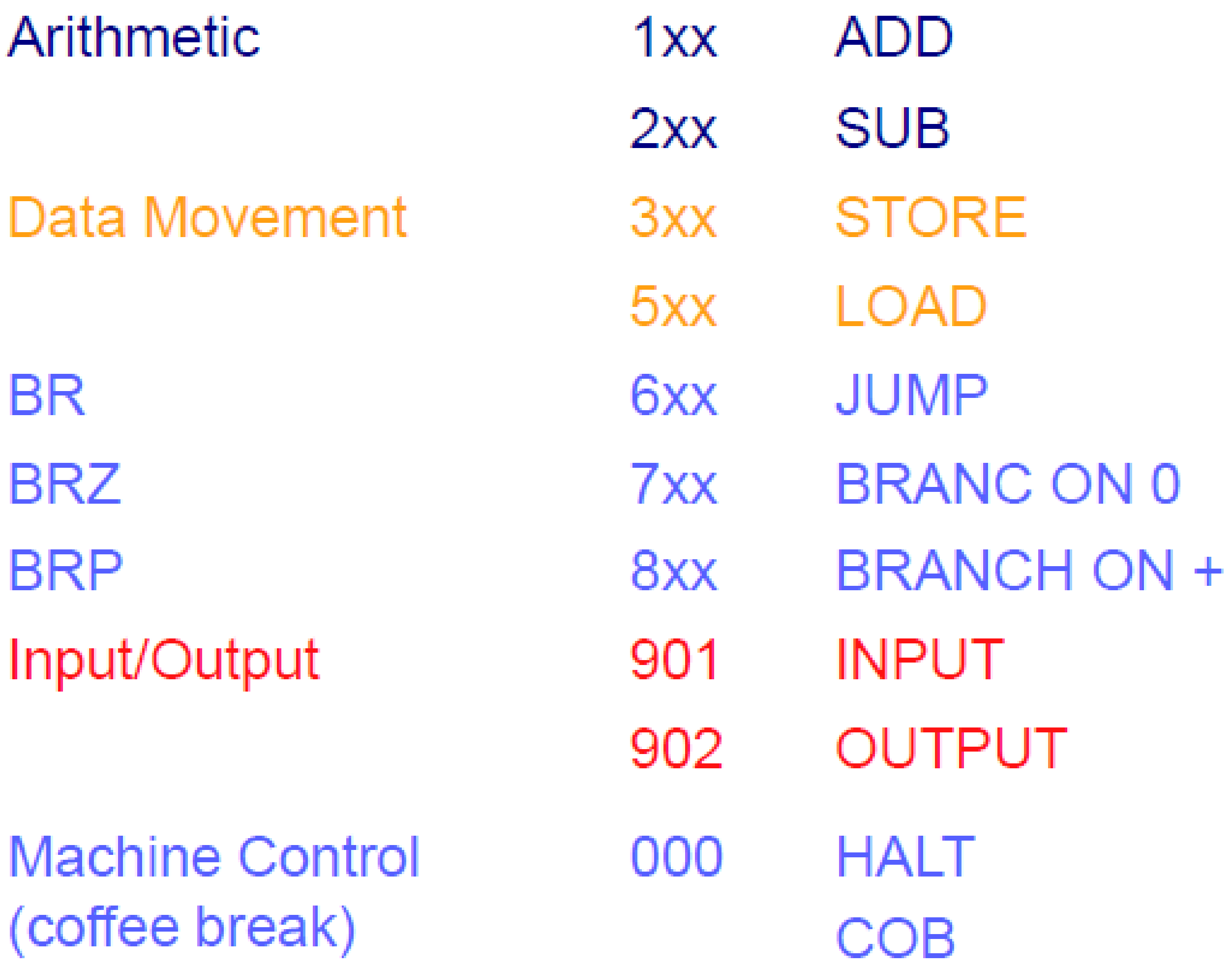
Mailbox	Mnemonic	Instruction Description
00	IN	;input 1st Number
01	STO 99	;store data
02	IN	;input 2nd Number
03	ADD 99	;add 1st # to 2nd #
04	OUT	;output result
05	COB	;stop
99	DAT 00	;data

Program Control

- Branching (executing an instruction out of sequence)
  - Changes the address in the counter
- Halt

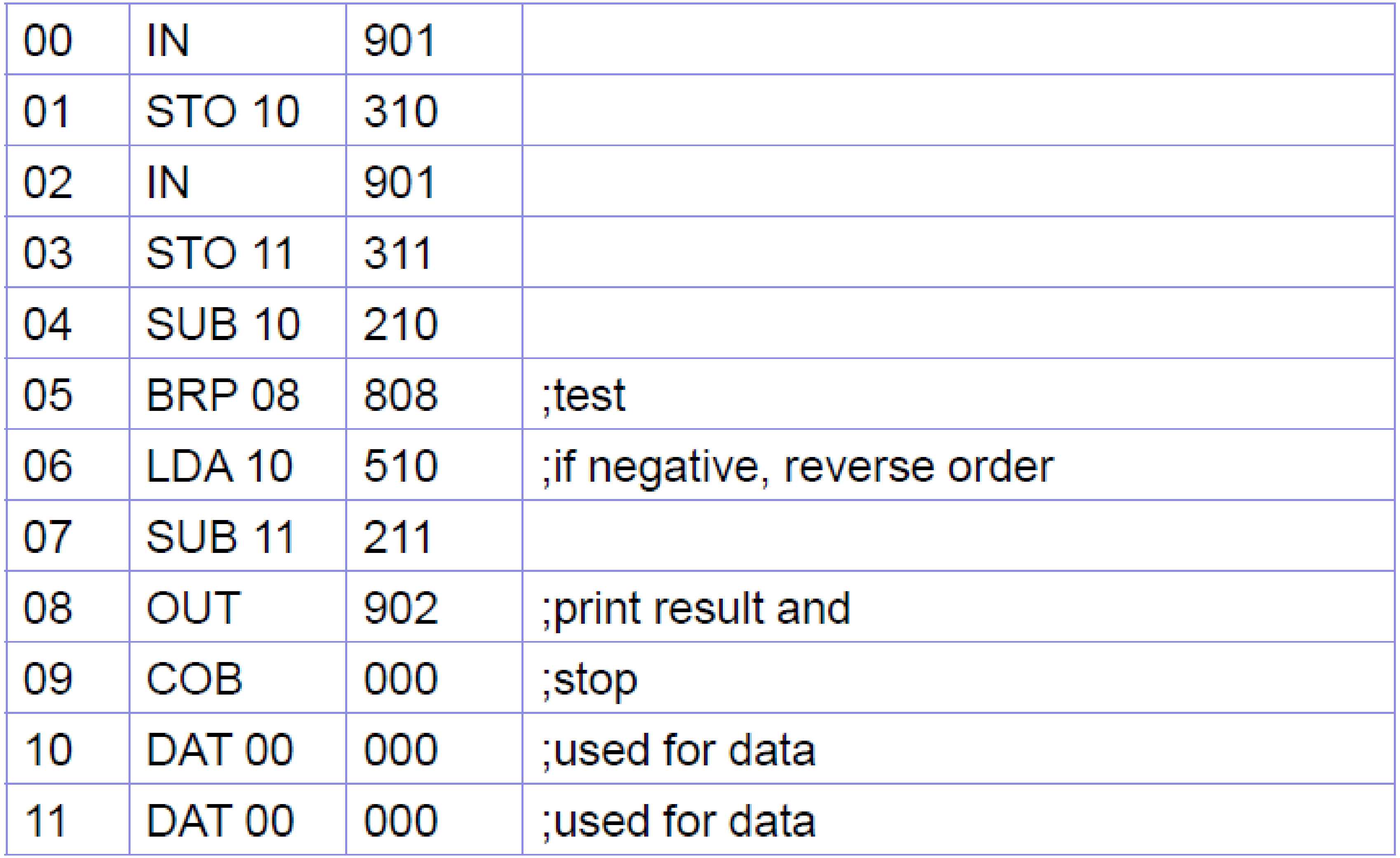
	Content	
	Op Code	**Operand (address)**
BR(Jump)	6	xx
BRZ(Branch on 0)	7	xx
BRP(Branch on +)	8	xx
COB(stop)	0	(ignore)

LMC Instruction Set



Arithmetic	1xx	ADD
	2xx	SUB
Data Movement	3xx	STORE
	5xx	LOAD
BR	6xx	JUMP
BRZ	7xx	BRANC ON 0
BRP	8xx	BRANCH ON +
Input/Output	901	INPUT
	902	OUTPUT
Machine Control (coffee break)	000	HALT COB

Find Positive Difference of 2 Numbers



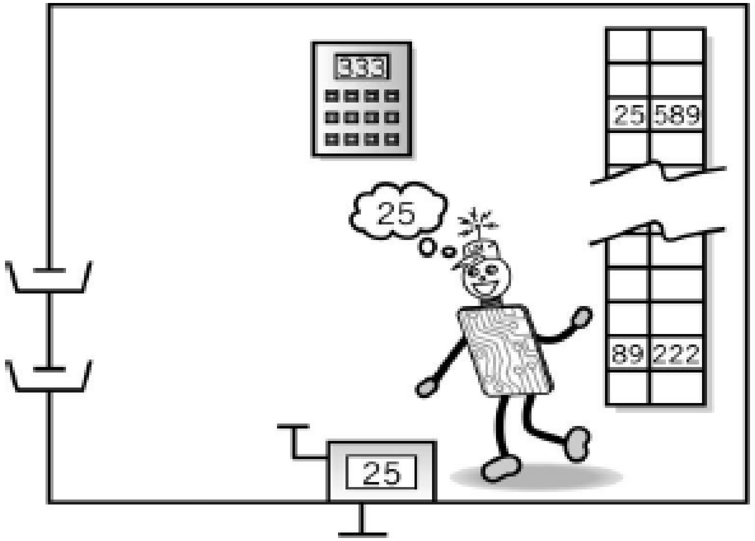
00	IN	901	
01	STO 10	310	
02	IN	901	
03	STO 11	311	
04	SUB 10	210	
05	BRP 08	808	;test
06	LDA 10	510	;if negative, reverse order
07	SUB 11	211	
08	OUT	902	;print result and
09	COB	000	;stop
10	DAT 00	000	;used for data
11	DAT 00	000	;used for data

Instruction Cycle

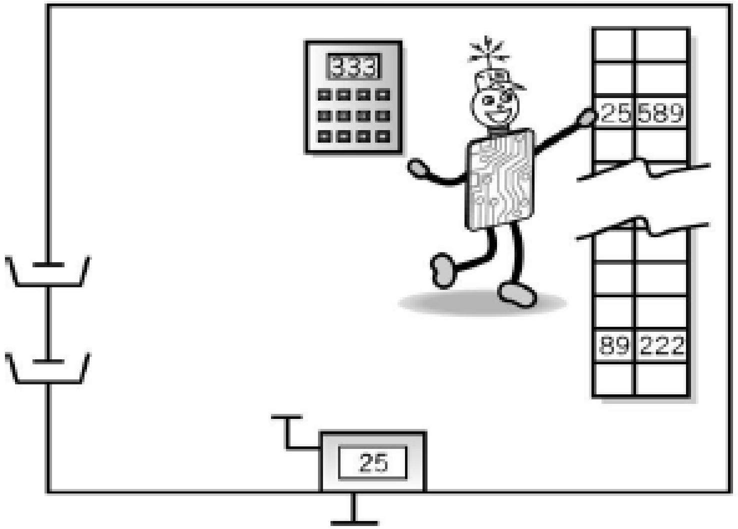
- *Fetch*: Little Man finds out what instruction he is to execute
- *Execute*: Little Man performs the work.

Fetch Portion of Fetch and Execute Cycle

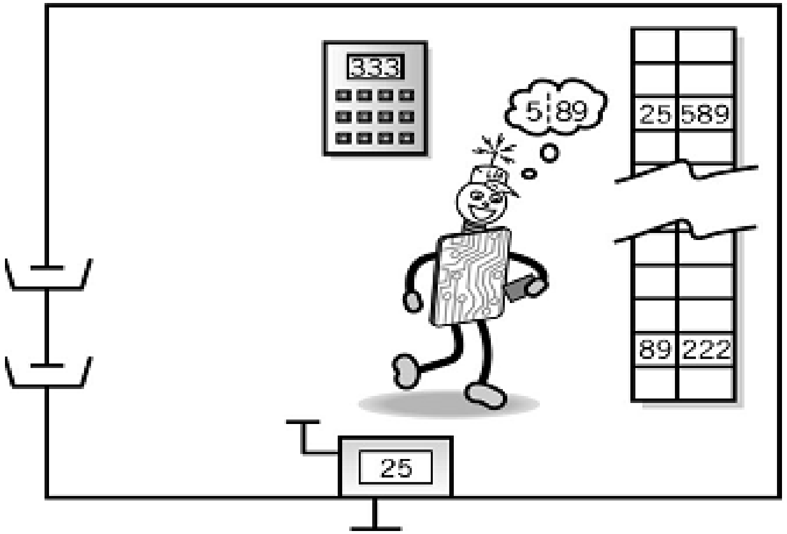
1.Little Man reads the address from the location counter



2.He walks over to the mailbox that corresponds to the location counter

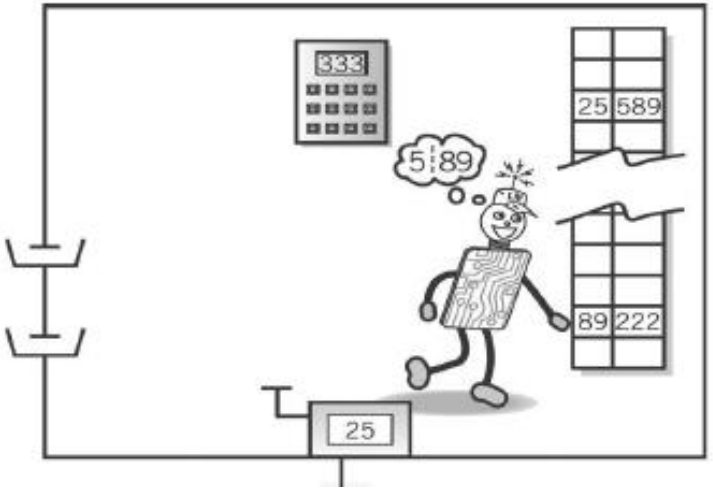


3.And reads the number on the slip of paper (he puts the slip back in case he needs to read it again later)

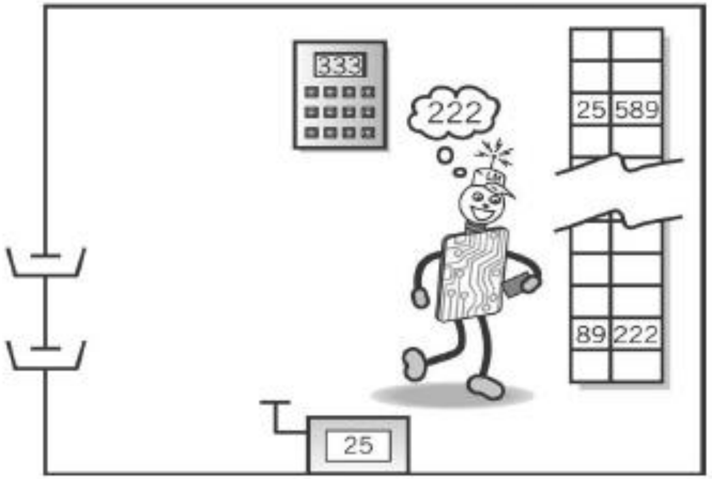


Execute Portion

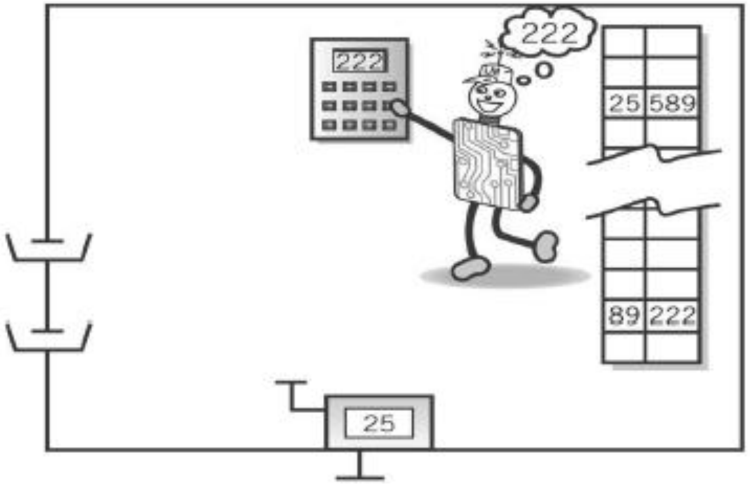
1. The Little Man goes to the mailbox address specified in the instruction he just fetched.



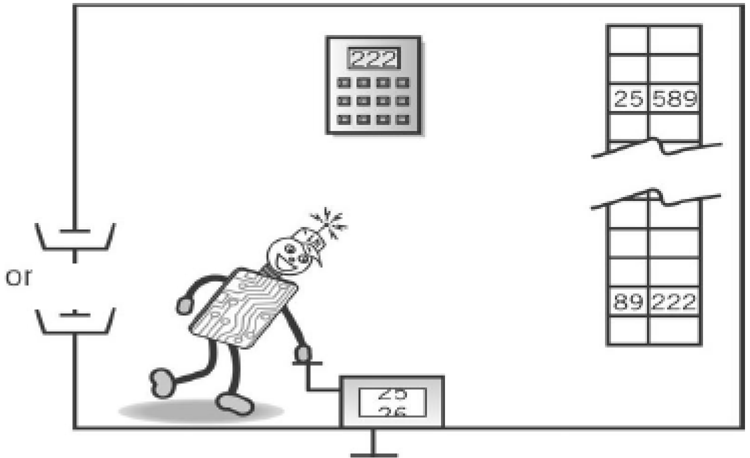
2.He reads the number in that mailbox (he remembers to replace it in case he needs it later).



3.He walks over to the calculator and punches the number in.



4.He walks over to the location counter and clicks it, which gets him ready to fetch the next instruction.



von Neumann Architecture(1945)

- Stored program concept
- Memory is addressed linearly
- Memory is addressed without regard to content