



# The Little Man Computer

---

- **Reading: Chapter 6**



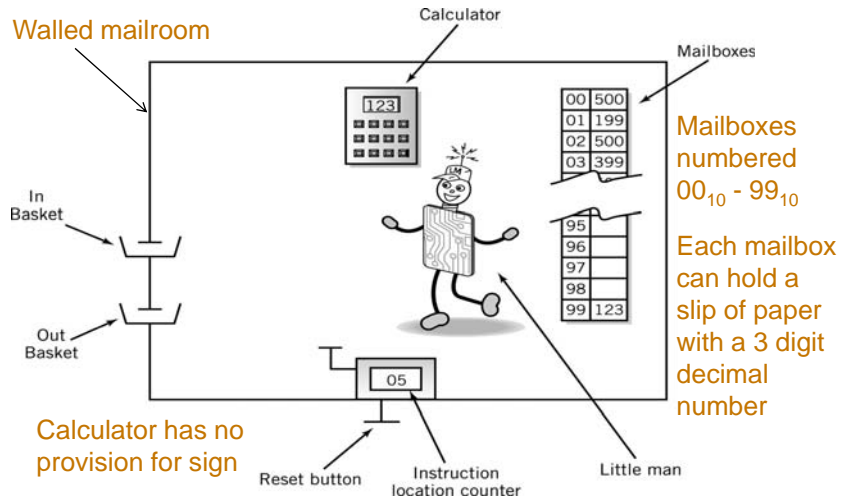
# Objectives

---

- Understand the basic operations of a computer by first understanding the operation of a simple, mechanical, version of a computer



# The Little Man Computer



Copyright 2010-2011 John Wiley & Sons, Inc. & Robert F. Kelly

6-3



# 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

Copyright 2010-2011 John Wiley & Sons, Inc. & Robert F. Kelly

6-4



## 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 (note mailboxes have a 2-digit address)

Address	Content	
	Op code	Operand

Copyright 2010-2011 John Wiley & Sons, Inc. & Robert F. Kelly

6-5



## Magic!

- Load program into memory
- Put data into In Basket
- Wait for result

Copyright 2010-2011 John Wiley & Sons, Inc. & Robert F. Kelly

6-6



# 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

Copyright 2010-2011 John Wiley & Sons, Inc. & Robert F. Kelly

6-7



# LMC Instruction Set

Arithmetic	1xx	ADD	Adds 3-digit number in mailbox to number already in calculator
	2xx	SUB	
Data Movement	3xx	STORE	Store 3 digit contents of calculator into mailbox
	5xx	LOAD	Enter 3 digit contents of mailbox in calculator
Input/Output	901	INPUT	Inputs number in basket into calculator
	902	OUTPUT	Writes number in calculator on paper and places in out basket
Machine Control (coffee break)	000	HALT	Stops little man
		COB	

Copyright 2010-2011 John Wiley & Sons, Inc. & Robert F. Kelly

6-8



# Input/Output

- Move data between calculator and in/out baskets

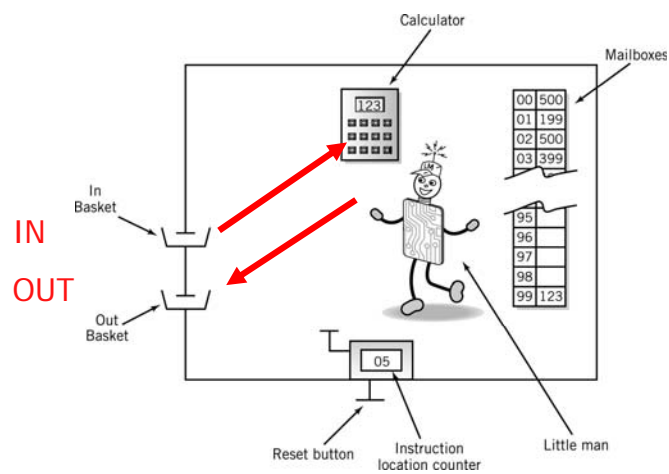
	Content	
	Op Code	Operand (address)
IN (input)	9	01
OUT (output)	9	02

Copyright 2010-2011 John Wiley & Sons, Inc. & Robert F. Kelly

6-9



# LMC Input/Output



Copyright 2010-2011 John Wiley & Sons, Inc. & Robert F. Kelly

6-10



# Internal Data Movement

- Between mailbox and calculator

STO  
(store)  
LDA (load)

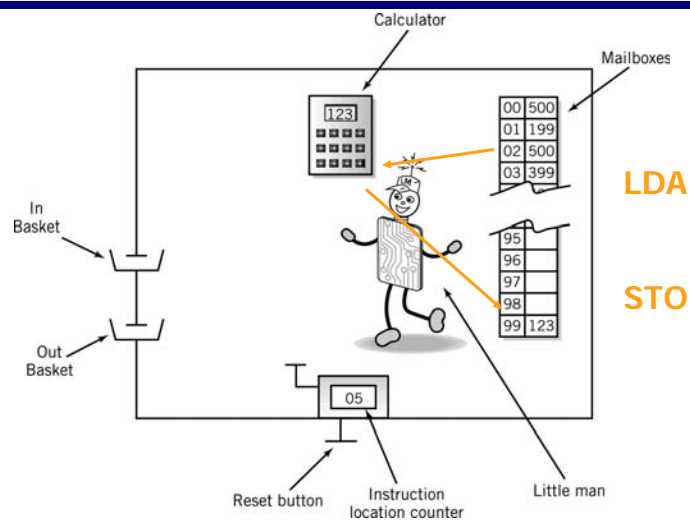
Content	
Op Code	Operand (address)
3	XX
5	XX

Copyright 2010-2011 John Wiley & Sons, Inc. & Robert F. Kelly

6-11



# LMC Internal Data



Copyright 2010-2011 John Wiley & Sons, Inc. & Robert F. Kelly

6-12



# Arithmetic Instructions

- Read mailbox
- Perform operation in the calculator

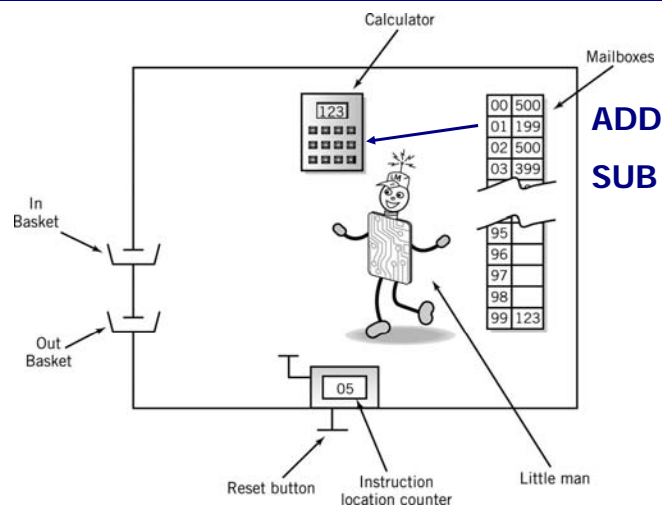
	Content	
	Op Code	Operand (address)
ADD	1	XX
SUB	2	XX

Copyright 2010-2011 John Wiley & Sons, Inc. & Robert F. Kelly

6-13



# LMC Arithmetic Instructions



Copyright 2010-2011 John Wiley & Sons, Inc. & Robert F. Kelly

6-14



## Data storage location

---

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

Copyright 2010-2011 John Wiley & Sons, Inc. & Robert F. Kelly

6-15



## Little Man Operations

---

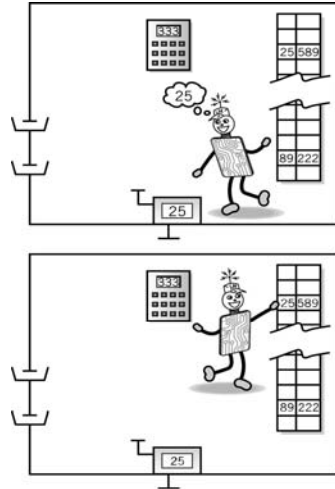
- Assume data is stored in mailboxes with addresses >90
- Instructions are stored in mailboxes starting at mailbox 00
- Little man starts by looking at the instruction counter and executing the instruction found in the associated mailbox
- When an instruction is complete, little man will increment the counter

Copyright 2010-2011 John Wiley & Sons, Inc. & Robert F. Kelly

6-16



## Fetch Portion of Fetch and Execute Cycle



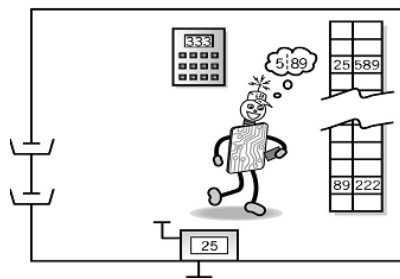
1. Little Man reads the address from the location counter
2. He walks over to the mailbox that corresponds to the location counter

Copyright 2010-2011 John Wiley & Sons, Inc. & Robert F. Kelly

6-17



## Fetch, cont.



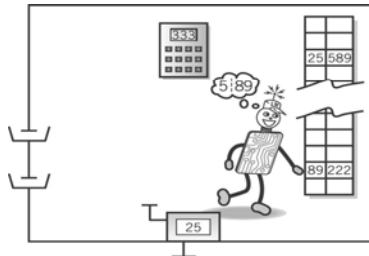
3. And reads the number on the slip of paper (he puts the slip back in case he needs to read it again later)

Copyright 2010-2011 John Wiley & Sons, Inc. & Robert F. Kelly

6-18

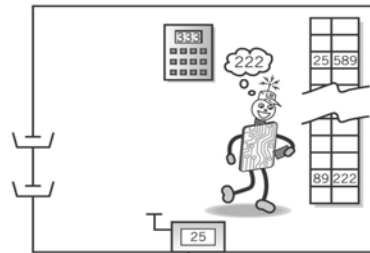


## Execute Portion



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).

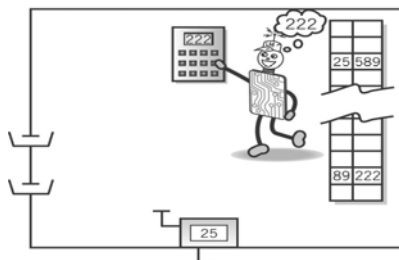


Copyright 2010-2011 John Wiley & Sons, Inc. & Robert F. Kelly

6-19

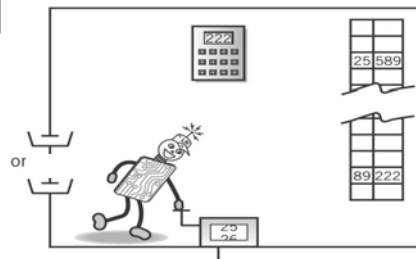


## Execute, cont.



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.



Copyright 2010-2011 John Wiley & Sons, Inc. & Robert F. Kelly

6-20

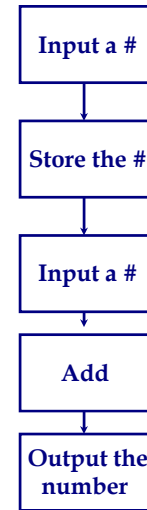


## Simple Program: Add 2 Numbers

- User places 2 numbers in the in basket
- Sum of the 2 numbers will appear in the out basket

Let's get volunteers to act out the operation of this program

1. One student to place 2 numbers in in-box and examine output
2. One student simulate mailboxes (e.g., read an instruction and execute it)
3. One student to retrieve (i.e., state) the contents of a mailbox
4. One student to simulate the calculator
5. One student to simulate the Instruction Ctr



Copyright 2010-2011 John Wiley & Sons, Inc. & Robert F. Kelly

6-21



## Program to Add 2 Numbers

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

Copyright 2010-2011 John Wiley & Sons, Inc. & Robert F. Kelly

6-22



## Program to Add 2 Numbers: Using Mnemonics

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

Copyright 2010-2011 John Wiley & Sons, Inc. & Robert F. Kelly

6-23



## Extended Instruction Set

- Branch Unconditional-6 (Jump) – Little man changes the instruction counter to the number shown in the address digits
- Branch on Zero-7 – If the value of the instruction counter is zero, little man will modify the instruction counter to the number shown in the address digits
- Branch on Positive-8 – If the value of the instruction counter is positive, little man will modify the instruction counter to the number shown in the address digits

Copyright 2010-2011 John Wiley & Sons, Inc. & Robert F. Kelly

6-24



## Extensions - Program Control

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

- Halt

BR (Jump)

BRZ (Branch on 0)

BRP (Branch on +)

COB (stop)

Content	
Op Code	Operand (address)
6	xx
7	xx
8	xx
0	(ignore)

Copyright 2010-2011 John Wiley & Sons, Inc. & Robert F. Kelly

6-25



## 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

Copyright 2010-2011 John Wiley & Sons, Inc. & Robert F. Kelly

6-26



## Find Positive Difference of 2 Numbers

00	IN	901	
01	STO 10	310	Let's simulate this program
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

Copyright 2010-2011 John Wiley & Sons, Inc. & Robert F. Kelly

6-27



## Instruction Cycle

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

Copyright 2010-2011 John Wiley & Sons, Inc. & Robert F. Kelly

6-28



## von Neumann Architecture (1945)

---

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