CMPT 422

Computer Organization and Architecture

Lecture Six

"The Little Man Computer"

Spring 2017

D. Cenk Erdil

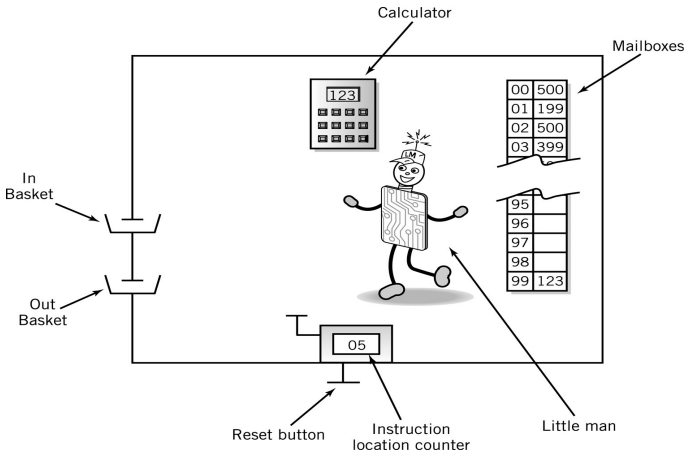
Outline

- Chapter Six - The Little Man Computer
  - Mailboxes
  - Instruction Set
  - Input/Output
  - Arithmetic Instructions
  - Data Storage
  - Sample Programs
  - Program Control
  - Fetch-Execute Cycle
- Arduino - Serial-Controlled Mood Lamp
- Call it a day!

Lecture Schedule

Wk	Dates	Class	Laboratory	References		Course Objectives
1	1/19	Computers and Systems		Chapter 1		1, 3
2	1/23 & 1/26	An Introduction to Systems Concepts and Architecture	Get to Know Your Tools and LED Flasher	Chapter 2	Arduino 01	1, 2, 3, 7
3	1/30 & 2/2	Number Systems	Spaceship Interface & SOS Morse Code Signaller	Chapter 3	Arduino 02	1, 6, 7, 8
4	2/6 & 2/9	Data Formats	Love-o-Meter & Traffic Lights	Chapter 4	Arduino 03	1, 6, 7, 8, 9
5	2/13 & 2/16	Representing Numerical Data	Color Mixing Lamp & LED Chase Effect	Chapter 5	Arduino 04	1, 6, 7, 8, 9
6	2/20 & 2/23	The Little Man Computer	**Serial-Controlled Mood Lamp**	Chapter 6	TBA	1, 8, 9
7	2/27 & 3/2	Midterm Recap & Exam on 3/2 in-class		Chapters 1-6		All of the above
8	3/6 & 3/9	The CPU and Memory	Shift Register 8-Bit Binary Counter	Chapter 7	TBA	2, 3, 4, 7. 8, 9
		SPRING RECESS				
9	3/20 & 3/23	CPU and Memory: Design, Enhancement & Implementation	Dual 8-Bit Binary Counters	Chapter 8	TBA	2, 3, 4, 7, 8
10	3/27 & 3/30	Input/Output	LED Dot Matrix	Chapter 9	TBA	1, 6, 7, 8
11	4/3 & 4/6	Computer Peripherals	Make your own design	Chapter 10		1, 2, 3
12	4/10 & 4/13	Modern Computer Systems	Make your own design	Chapter 11		1, 2, 3, 5, 9
13	4/17 & 4/20	Presentations				
14	4/24 & 4/27	Presentations/Extra/Backup				7, 10
15	5/1 & 5/4	Presentations/Extra/Final Recap				7, 10
	5/11	Final Exam at 1pm		Chapters 7-11		All of the above

The 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

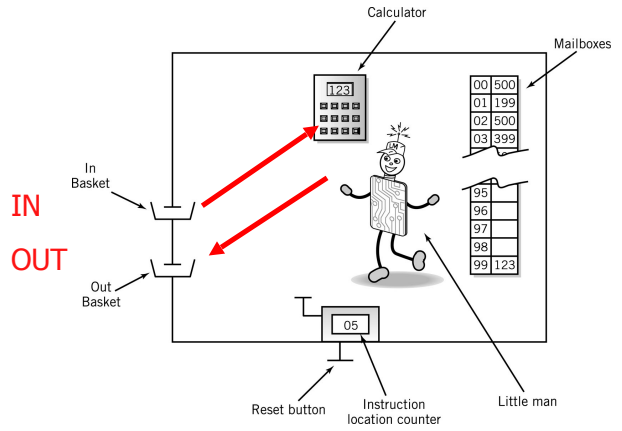
Arithmetic	1xx	ADD
	2xx	SUB
Data Movement	3xx	STORE
	5xx	LOAD
Input/Output	901	INPUT
	902	Output
Machine Control (coffee break)	000	HALT COB

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

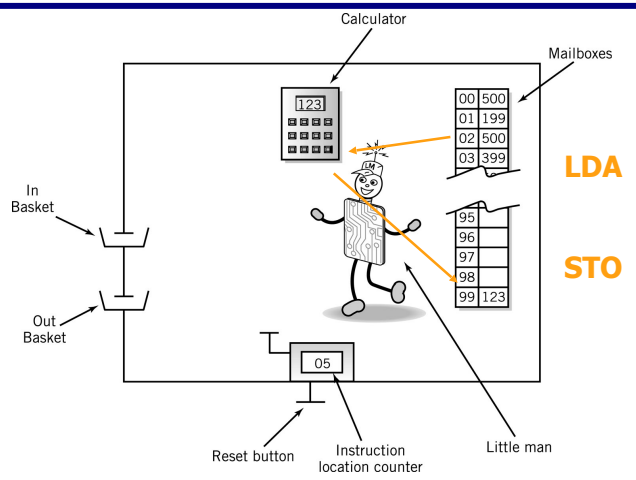


Internal Data Movement

- Between mailbox and calculator

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

LMC Internal Data

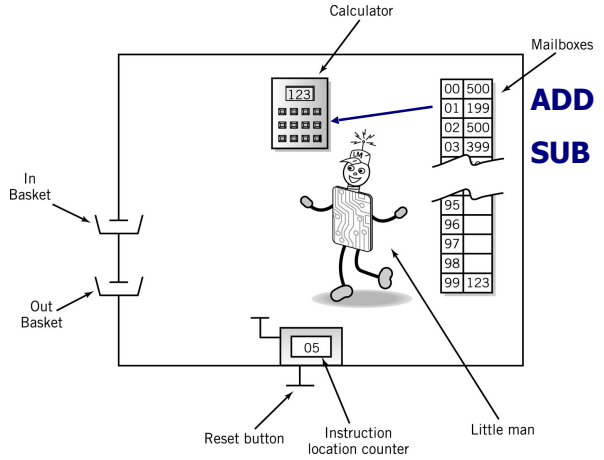


Arithmetic Instructions

- Read mailbox
- Perform operation in the calculator

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

LMC Arithmetic Instructions

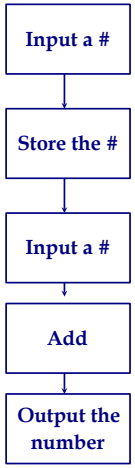


Data storage location

- 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	BRANCH ON 0
BRP	8xx	BRANCH ON +
Input/Output	901	INPUT
	902	OUTPUT
Machine Control (coffee break)	000	HALT COB

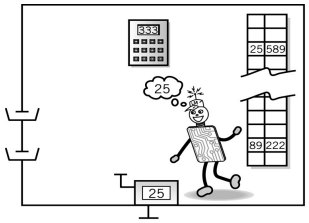
Find Positive Difference of 2 Numbers

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

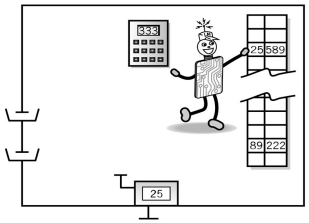
Instruction Cycle

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

Fetch Portion of Fetch and Execute Cycle

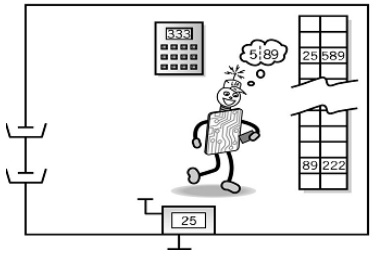


1. Little Man reads the address from the location counter



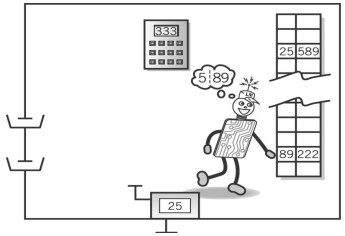
2. He walks over to the mailbox that corresponds to the location counter

Fetch, cont.



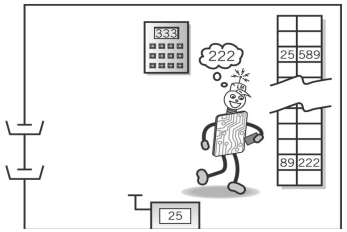
3. And reads the number on the slip of paper (he puts the slip back in case he needs to read it again later)

Execute Portion

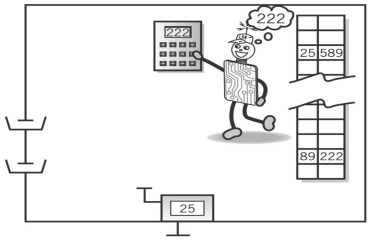


1. The Little Man goes to the mailbox address specified in the instruction he just fetched

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



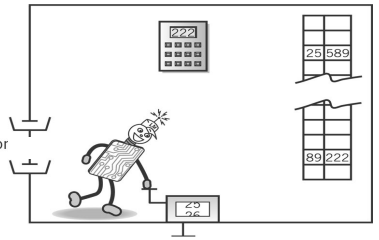
Execute, cont.



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

or



von Neumann Architecture (1945)

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

Copyright 2013 John Wiley & Sons