

Unit-3 Programming the Basic Computer

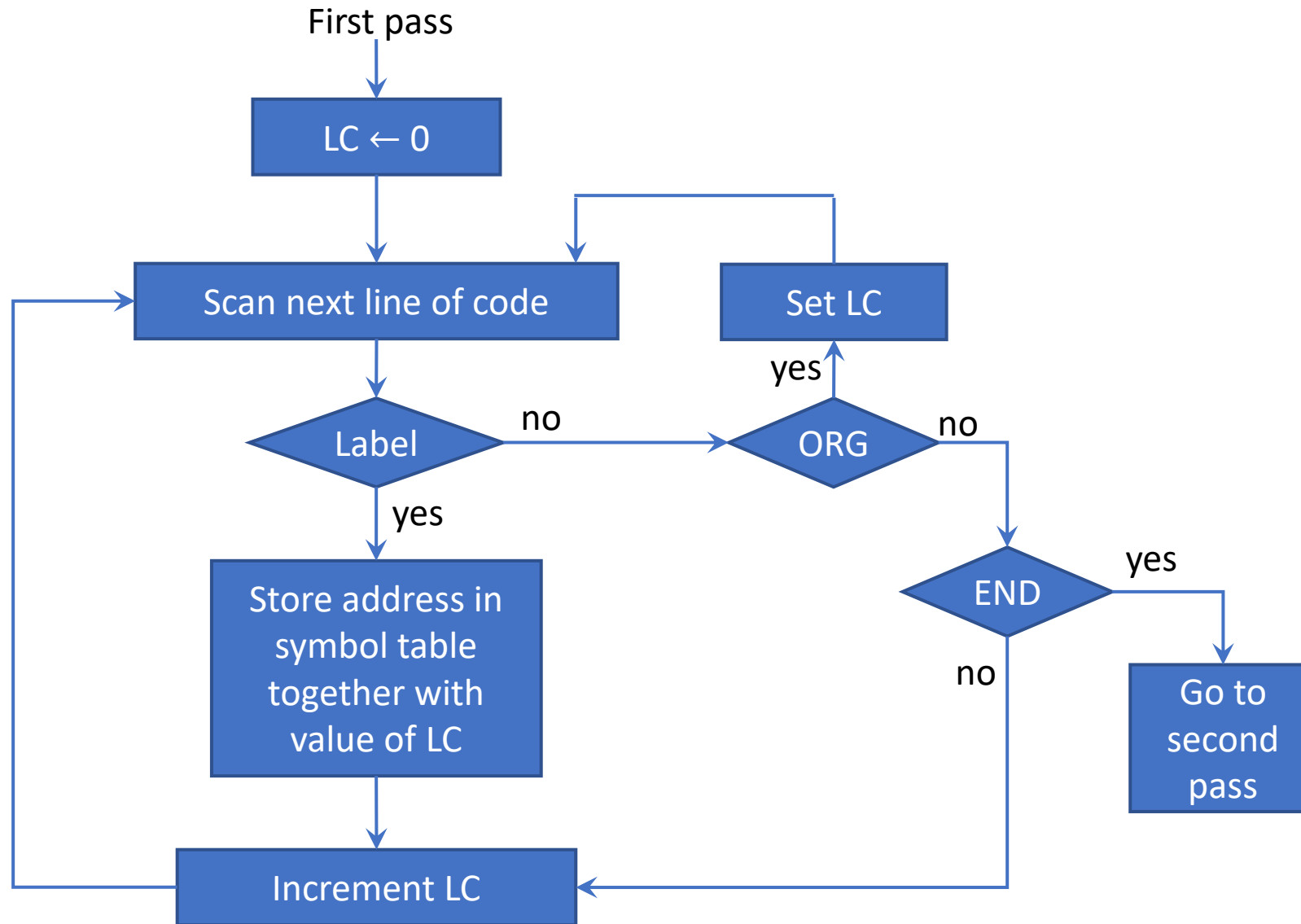
Dr. Ram Kishan Dewangan

Assistant Professor

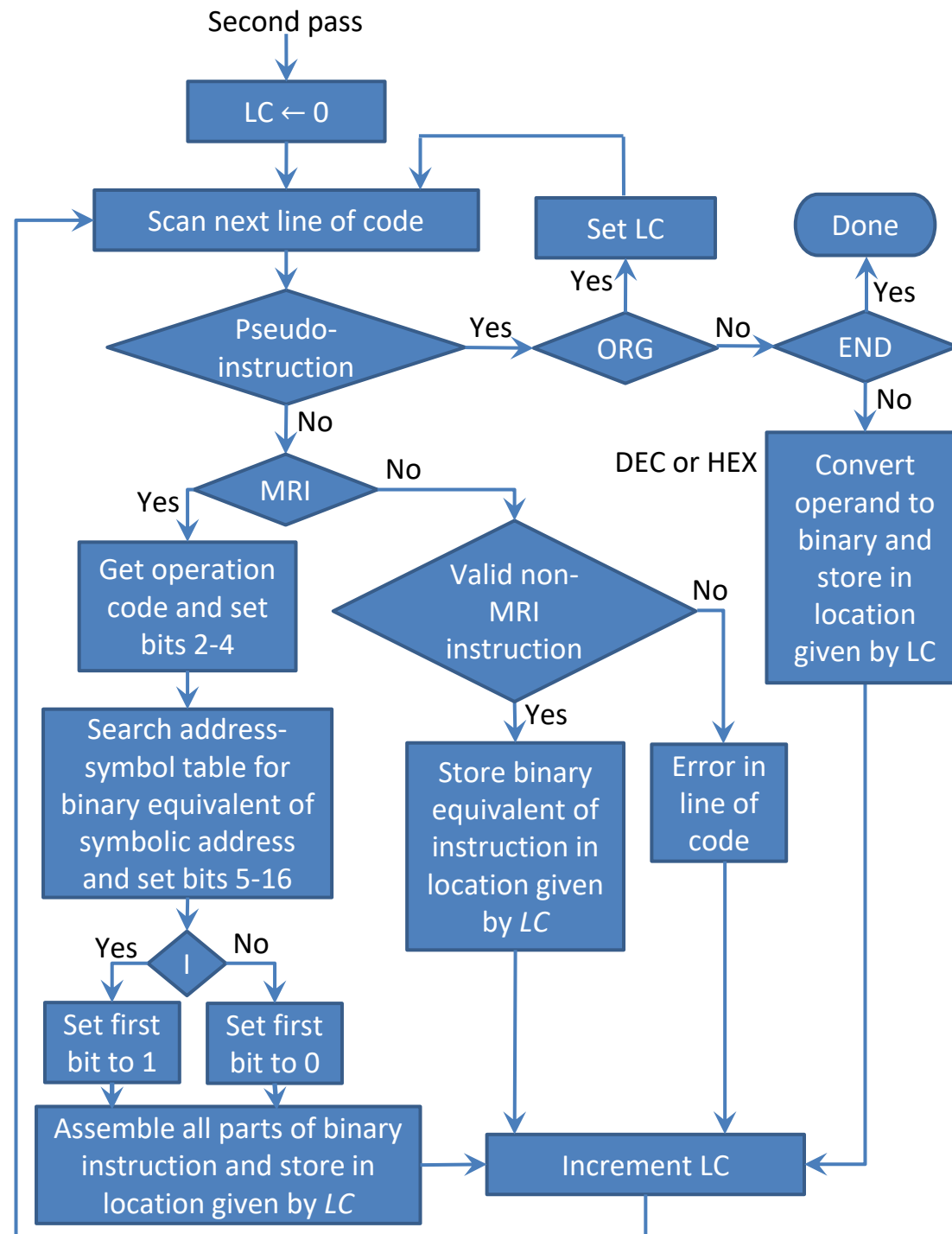
Topics to be covered

- Introduction
- Machine Language
- Assembly Language
- Assembler
- Program loops
- Programming Arithmetic and logic operations
- Subroutines
- I-O Programming

First Pass of an assembler



Second Pass of an assembler



Program Loops

Program Loops

- A *program loop* is a sequence of instructions that are executed many times, each time with a different set of data.
- A system program that translates a program written in a high-level programming language to a machine language program is called a *compiler*.

A.L.P. to Add 100 Numbers

1	ORG 100	/Origin of program is HEX 100
2	LDA ADS	/Load first address of operands
3	STA PTR	/Store in pointer
4	LDA NBR	/Load minus 100
5	STA CTR	/Store in counter
6	CLA	/Clear accumulator
7	LOP, ADD PTR I	/Add an operand to AC
8	ISZ PTR	/Increment pointer
9	ISZ CTR	/Increment counter
10	BUN LOP	/Repeat loop again
11	STA SUM	/Store sum
12	HLT	/Halt

A.L.P. to Add 100 Numbers

13	ADS, HEX 150	/First address of operands
14	PTR, HEX 0	/This location reserved for pointer
15	NBR, DEC -100	/Constant to initialized counter
16	CTR, HEX 0	/This location reserved for a counter
17	SUM, HEX 0	/Sum is store here
18	ORG 150	/Origin of operands is HEX 150
19	DEC 75	/First operand
.		
.		
.		
118	DEC 23	/Last operand
119	END	/End of symbolic program

A.L.P. to clear the contents of hex locations 500 to 5FF with 0

1		ORG 100	/Origin of program is HEX 100
2		LDA ADS	/Load first address of operands
3		STA PTR	/Store in pointer
4		LDA NBR	/Load minus 255
5		STA CTR	/Store in counter
6		CLA	/Clear accumulator
7	LOP,	STA PTR I	/Store zero to location pointed by PTR
8		ISZ PTR	/Increment pointer
9		ISZ CTR	/Increment counter
10		BUN LOP	/Repeat loop again
11		HLT	/Halt
12	ADS,	HEX 500	/First address of operands
13	PTR,	HEX 0	/This location reserved for pointer
14	NBR,	DEC -255	/Constant to initialize counter
15	CTR,	HEX 0	/This location reserved for a counter
16		END	/End of symbolic program

Programming Arithmetic and Logic Operation

A.L.P. to Add Two Double-Precision Numbers

1	ORG 100	/Origin of program is HEX 100
2	LDA AL	/Load A low
3	ADD BL	/Add B low, carry in E
4	STA CL	/Store in C low
5	CLA	/Clear AC
6	CIL	/Circulate to bring carry into AC(16)
7	ADD AH	/Add A high and carry
8	ADD BH	/Add B high
9	STA CH	/Store in C high
10	HLT	/Halt

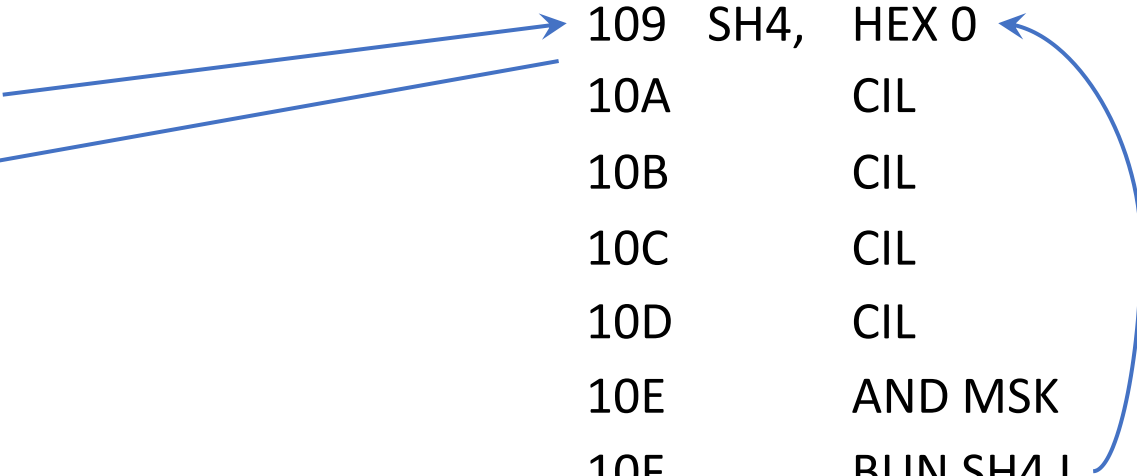
Subroutines

Subroutine

- A set of common instructions that can be used in a program many times is called a *subroutine*.
- Each time that a subroutine is used in the main part of the program, a branch is executed to the beginning of the subroutine.
- After the subroutine has been executed, a branch is made back to the main program.
- A subroutine consists of a self contained sequence of instructions that carries a given task.

A.L.P. to demonstrate Subroutine

		ORG 100		
100	LDA X		109	SH4, HEX 0
101	BSA SH4		10A	CIL
102	STA X		10B	CIL
103	LDA Y		10C	CIL
104	BSA SH4		10D	CIL
105	STA Y		10E	AND MSK
106	HLT		10F	BUN SH4 I
107	X, HEX 1234		110	MSK, HEX FFF0
108	Y, HEX 4321			END



I-O

Programming

A.L.P. to input one character

1		ORG 100	/Origin of program is HEX 100
2	CIF,	SKI	/Check input flag
3		BUN CIF	/Flag = 0, branch to check again
4		INP	/Flag = 1, input character
5		OUT	/Print character
6		STA CHR	/Store character
7		HLT	
8	CHR,	-	/Store character here
9		END	

A.L.P. to output one character

1	ORG 100	/Origin of program is HEX 100
2	LDA CHR	/Load character into AC
3	COF, SKO	/Check output flag
4	BUN COF	/Flag = 0, branch to check again
5	OUT	/Flag = 1, output character
6	HLT	
7	CHR, HEX 0057	/Character is "W"
8	END	

Thank You