RAMA UNIVERSITY

www.ramauniversity.ac.in

FACULTY OF ENGINEERING & TECHNOLOGY

CSPS-106 Computer Organization

Lecture-13

Mr. Dilip Kumar J Saini
Assistant Professor
Computer Science & Engineering

OUTLINE

- MAJOR COMPONENTS OF CPU
- GENERAL REGISTER ORGANIZATION
- OPERATION OF CONTROL UNIT
- ALU CONTROL
- MEMORY STACK ORGANIZATION

MAJOR COMPONENTS OF CPU

Storage Components:

Registers

Flip-flops

Execution (Processing) Components:

Arithmetic Logic Unit (ALU):

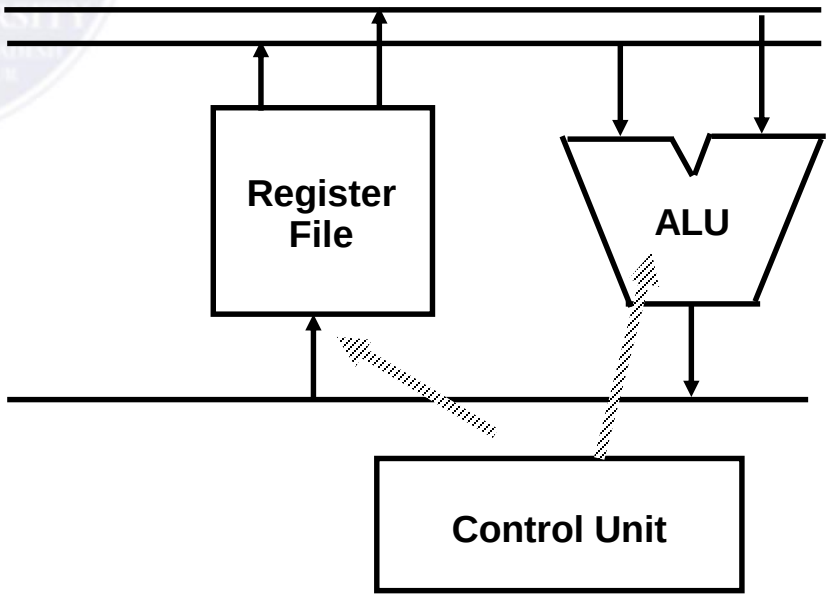
Arithmetic calculations, Logical computations, Shifts/Rotates

Transfer Components:

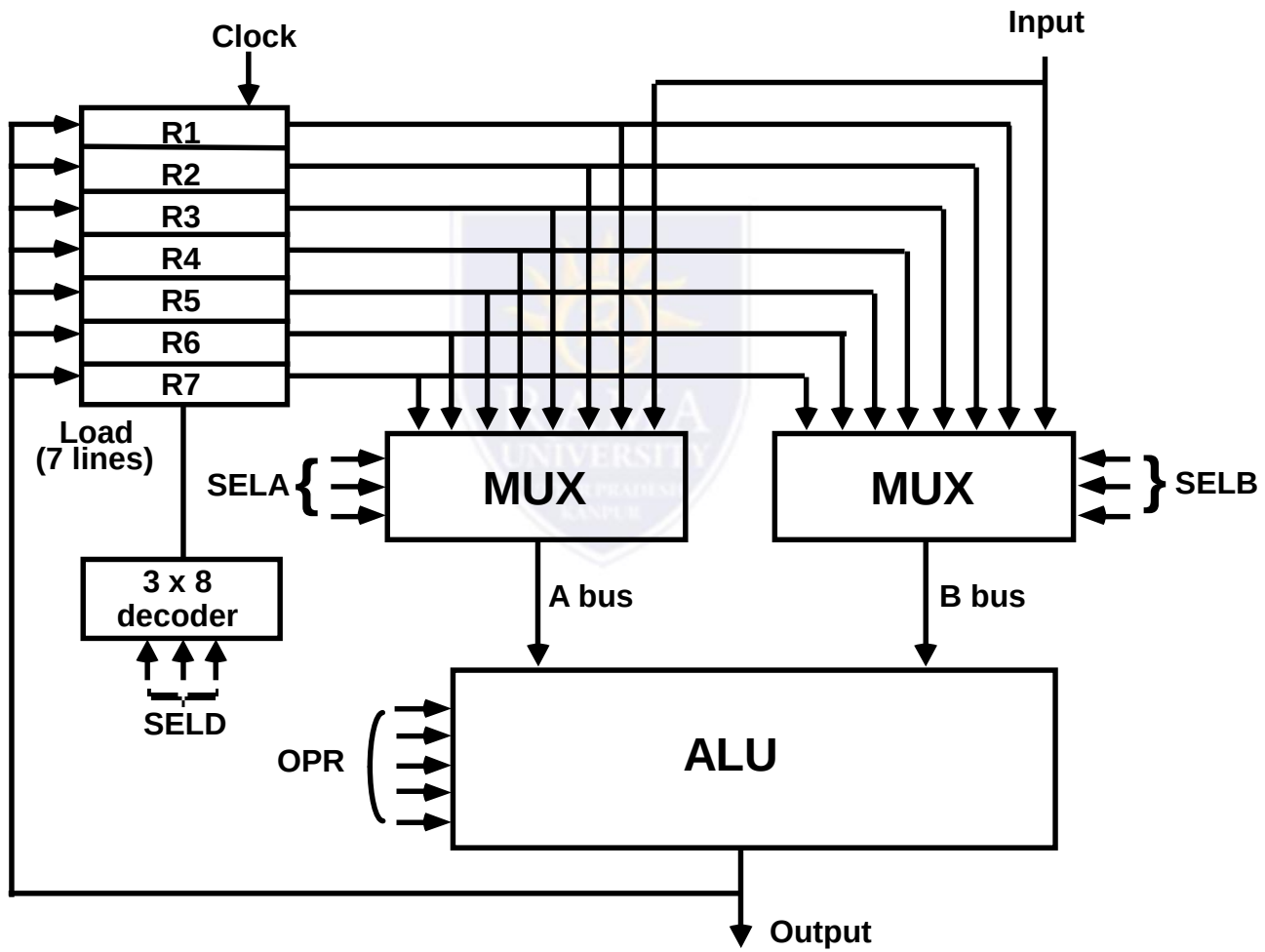
Bus

Control Components:

Control Unit



GENERAL REGISTER ORGANIZATION



OPERATION OF CONTROL UNIT

The control unit directs the information flow through ALU by:

- Selecting various *Components* in the system

- Selecting the *Function* of ALU

Example: R1 <- R2 + R3

[1] MUX A selector (SELA): BUS A ← R2

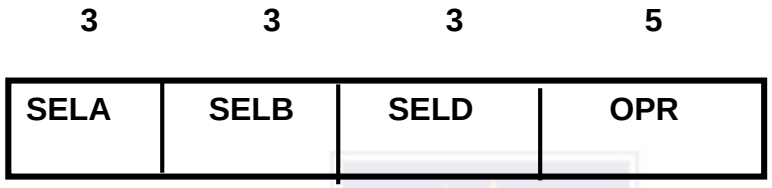
[2] MUX B selector (SELB): BUS B ← R3

[3] ALU operation selector (OPR): ALU to ADD

[4] Decoder destination selector (SELD): R1 ← Out Bus

OPERATION OF CONTROL UNIT

Control Word



Encoding of register selection fields

Binary Code	SELA	SELB	SELD
000	Input	Input	None
001	R1	R1	R1
010	R2	R2	R2
011	R3	R3	R3
100	R4	R4	R4
101	R5	R5	R5
110	R6	R6	R6
111	R7	R7	R7

ALU CONTROL

Encoding of ALU operations

OPR Select	Operation	Symbol
00000	Transfer A	TSFA
00001	Increment A	INCA
00010	ADD A + B	ADD
00101	Subtract A - B	SUB
00110	Decrement A	DECA
01000	AND A and B	AND
01010	OR A and B	OR
01100	XOR A and B	XOR
01110	Complement A	COMA
10000	Shift right A	SHRA
11000	Shift left A	SHLA

Examples of ALU Microoperations

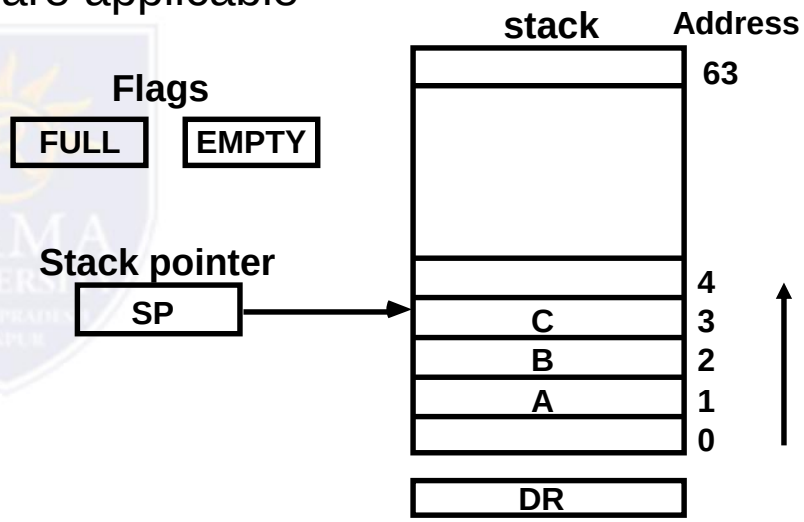
	Symbolic Designation				
Microoperation	SELA	SELB	SELD	OPR	Control Word
R1 ← R2 - R3	R2	R3	R1	SUB	010 011 001 00101
R4 ← R4 ∨ R5	R4	R5	R4	OR	100 101 100 01010
R6 ← R6 + 1	R6	-	R6	INCA	110 000 110 00001
R7 ← R1	R1	-	R7	TSFA	001 000 111 00000
Output ← R2	R2	-	None	TSFA	010 000 000 00000
Output ← Input	Input	-	None	TSFA	000 000 000 00000
R4 ← shl R4	R4	-	R4	SHLA	100 000 100 11000
R5 ← 0	R5	R5	R5	XOR	101 101 101 01100

REGISTER STACK ORGANIZATION

Stack

- Very useful feature for nested subroutines, nested loops control
- Also efficient for arithmetic expression evaluation
- Storage which can be accessed in LIFO
- Pointer: SP
- Only PUSH and POP operations are applicable

Register Stack



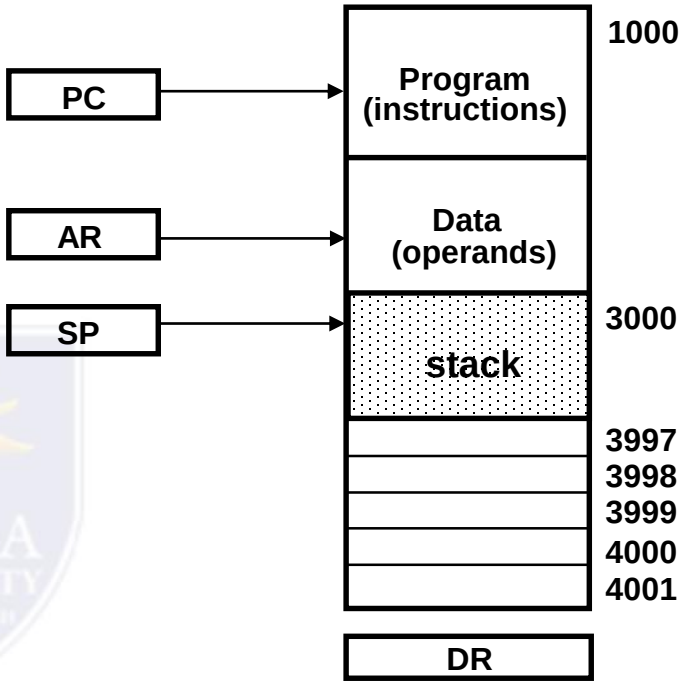
Push, Pop operations

/* Initially, SP = 0, EMPTY = 1, FULL = 0 */

PUSH	**POP**
SP ← SP + 1	**DR ← M[SP]**
M[SP] ← DR	**SP ← SP - 1**
If (SP = 0) then (FULL ← 1)	**If (SP = 0) then (EMPTY ← 1)**
EMPTY ← 0	**FULL ← 0**

MEMORY STACK ORGANIZATION

Memory with Program, Data, and Stack Segments



- A portion of memory is used as a stack with a processor register as a stack pointer

- PUSH: SP ← SP - 1

  M[SP] ← DR

- POP: DR ← M[SP]

  SP ← SP + 1

- Most computers do not provide hardware to check stack overflow (full stack) or underflow(empty stack)

Multiple Choice Question

MUTIPLE CHOICE QUESTIONS:

Sr no	Question	Option A	Option B	OptionC	OptionD
1	is the first step in the evolution of programming languages:	machine language	assembly language	code language	none of these
2	Mnemonic refers to:	Instructions	Code	Symbolic codes	Assembler
3	Mnemonic represent	Operation codes	Strings	Address	None of these
4	To represent addresses in assembly language weuse:	String characters	Array	structures	Enum
5	Assembler works to convert assembly language program into machine language:	Before the computer can execute it	After the computer can execute it	In between execution	None of these

REFERENCES

- http://www.engppt.com/search/label/Computer%20Organization%20and%20Architecture
- http://www.engppt.com/search/label/Computer%20Architecture%20ppt