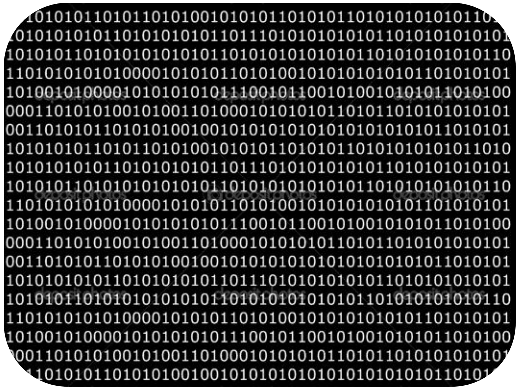
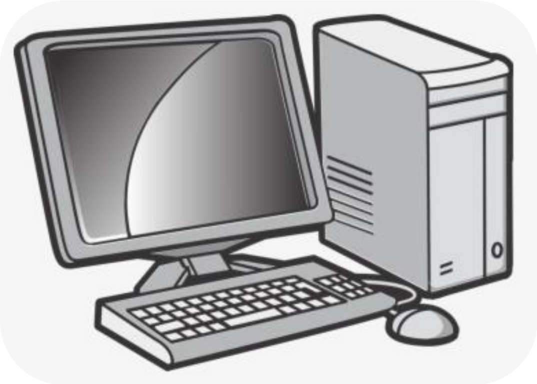
AMIRAJ

COLLEGE OF ENGINEERING & TECHNOLOGY

Unit-5
Central Processing Unit





Subject:- COA
Code:-3140707

Prepared by:
Asst. Prof. S Y JOSHI
(CSE Department,ACET)

AMIRAJ
COLLEGE OF ENGINEERING & TECHNOLOGY

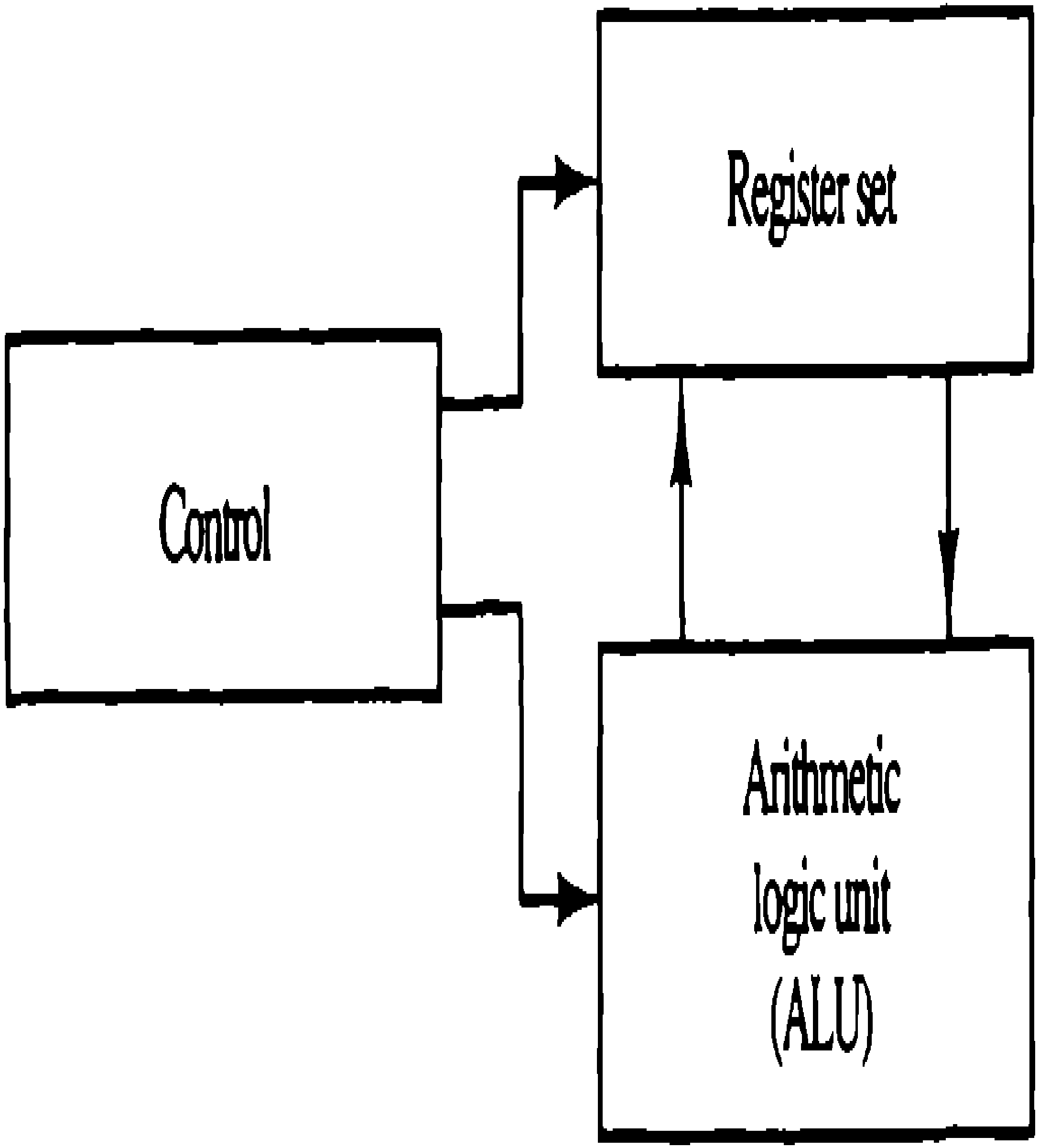
Central Processing Unit

Topics to be covered

- Introduction
- General Register Organization
- Stack Organization
- InstructionFormats
- Addressing Modes
- Data Transfer and Manipulation
- Program Control
- Reduced instruction Set Computer-RISC

CPU

- The part of the computer that performs the bulk of data-processing operations is called the central processing unit and is referred to as the CPU.
- The CPU is made up of three major parts



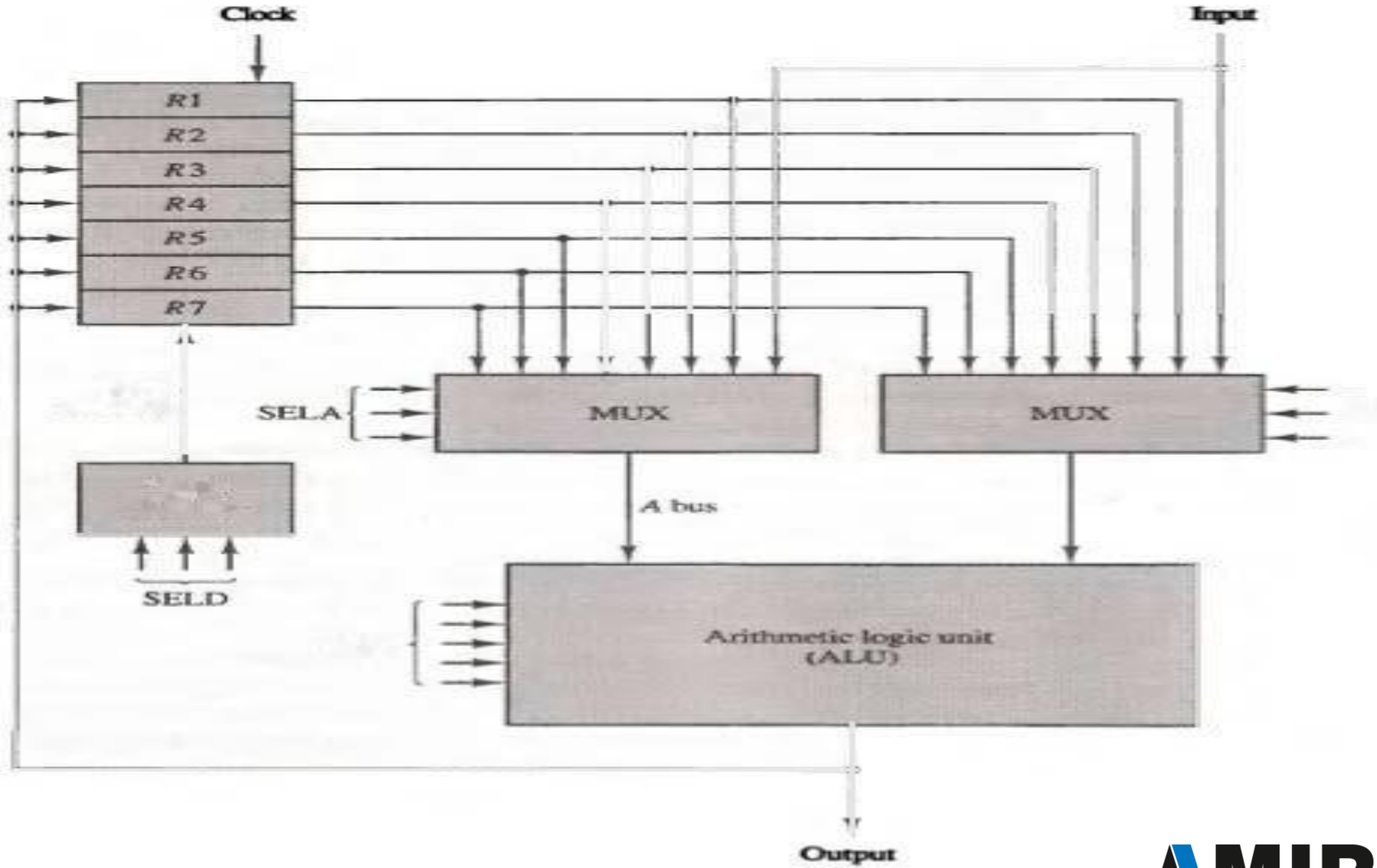
General Register Organization

- Memory locations are needed for storing pointers, counters, return addresses, temporary results, and partial products during multiplication.
- Having to refer to memory locations for such applications is time consuming because memory access is the most time-consuming operation; it is more convenient and more efficient to store these intermediate values in processor registers
- When a large number of registers are included in the CPU, it is most efficient to connect them through a common bus system.

General Register Organization

bus system

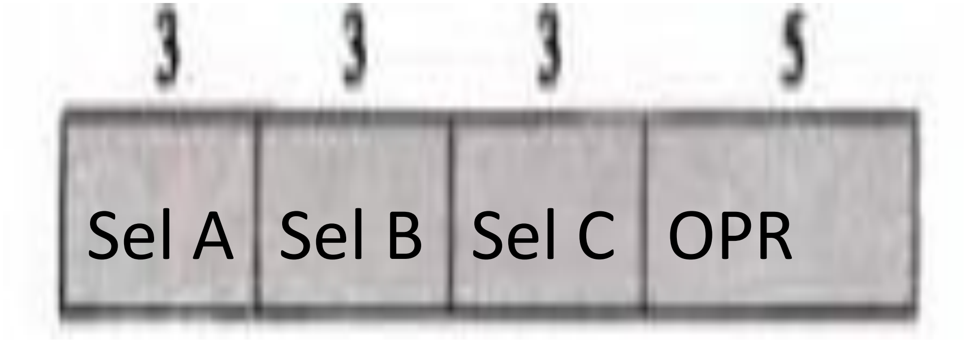
SECTION 8-2 General Register Organization 243



(a) Block diagram

General Register Organization

There are 14 binary selection inputs in the unit, and their combined value specifies a control word. The 14-bit control word is defined in Fig. It consists of four fields.



3	3	3	5
Sel A	Sel B	Sel C	OPR

•The three bits of SELA select a source register for the A input of the ALU.

• The three bits of SELB select a register for the B input of the ALU.

•The three bits of SELD select a destination register using the decoder and its seven load outputs.

•The five bits of OPR select one of the operations in the ALU.

•The 14-bit control word when applied to the selection inputs specify a particular micro-operation.

General Register Organization

For example, to perform the operation

R 1 <-- R2 + R3

the control must provide binary selection variables to the following selector inputs:

1. MUX A selector (SELA): to place the content of R2 into bus A
2. MUX B selector (SELB): to place the content of R3 into bus B
3. ALU operation selector (OPR): to provide the arithmetic addition A + B
4. Decoder destination selector (SELD): to transfer the content o f the output bus into R1.

General Register Organization

- The encoding of the register selections is specified in Table-1

Table-1 Encoding of Register Selection Fields

Code	SELA	SELB	SELD
000	Input	Input	None
001	**R 1**	**R 1**	**R 1**
010	R2	R2	R2
011	**R3**	**R3**	**R3**
100	R4	R4	R4
101	**R5**	**R5**	**R5**
110	R6	R6	R6
1 1 1	**R7**	**R7**	**R7**

General Register Organization

- The ALU provides arithmetic and logic operations. In addition, the CPU must provide shift operations. The shifter may be placed in the input of the ALU to provide a preshift capability, or at the output of the ALU to provide postshifting capability. In some cases, the shift operations are included with the ALU The OPR field has five bits and each operation is designated with a symbolic name.

TABLE-2 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**

General Register Organization

Example of Micro-operations:

control word of 14 bits is needed to specify a Micro-operation in the CPU. The control word for a given micro operation can be derived from the selection variables.

For example, the subtract Micro operation given by the statement

$$R1 \leftarrow R2 - R3$$

specifies R2 for the A input of the ALU, R3 for the B input of the ALU, R1 for the destination register, and an ALU operation to subtract A – B.

The binary control word for the subtract Microoperation is obtained as follows:

Field:	SELA	SELB	SELD	OPR
Symbol:	R2	R3	R1	SUB
Control word:	010	011	001	00101

General Register Organization

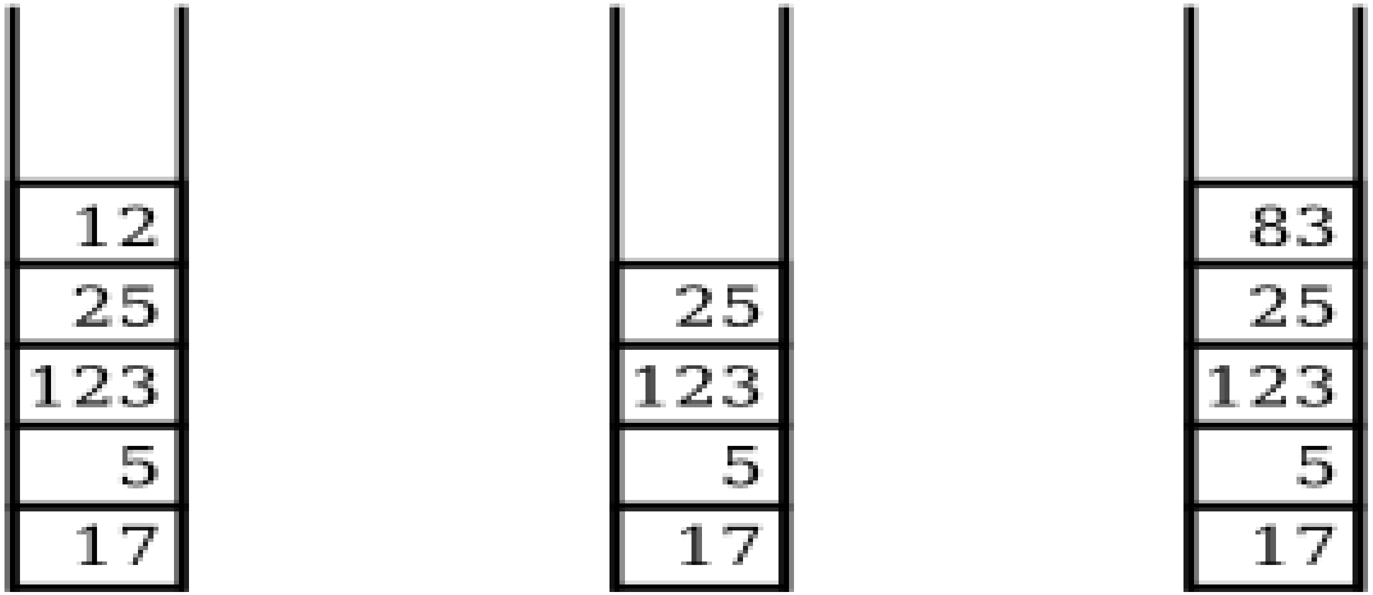
- The most efficient way to generate control words with a large number of bits is to store them in a memory unit.
- A memory unit that stores control words is referred to as a control memory. By reading consecutive control words from memory, it is possible to initiate the desired sequence of microoperations for the CPU.
- This type of control is referred to as microprogrammed controL.
- The binary control word for the CPU will come from the outputs of the control memory marked "micro-ops."

Stack Organization

- A stack is a storage device that stores information in such a manner that the item stored last is the first item retrieved . Stack is the last-in, first-out (LIFO) list. The stack in digital computers is essentially a memory unit with an address register that can count only (after an initial value is loaded into it).
- The register that holds the address for the stack is called a stack pointer (SP) because its value always points at the top item in the stack.
- The two operations of a stack are the insertion and deletion of items.
- The operation of insertion is called **push** (or push-down) because it can be thought of as the result of pushing a new item on top.
- The operation of deletion is called **pop** (or pop-up) because it can be thought of as the result of removing one item so that the stack pops up.
- These operations are simulated by incrementing or decrementing the stack pointer register.

Stack Organization
PUSH-POP

In a stack, all operations take place at the "top" of the stack. The "push" operation adds an item to the top of the stack. The "pop" operation removes the item on the top of the stack and returns it.

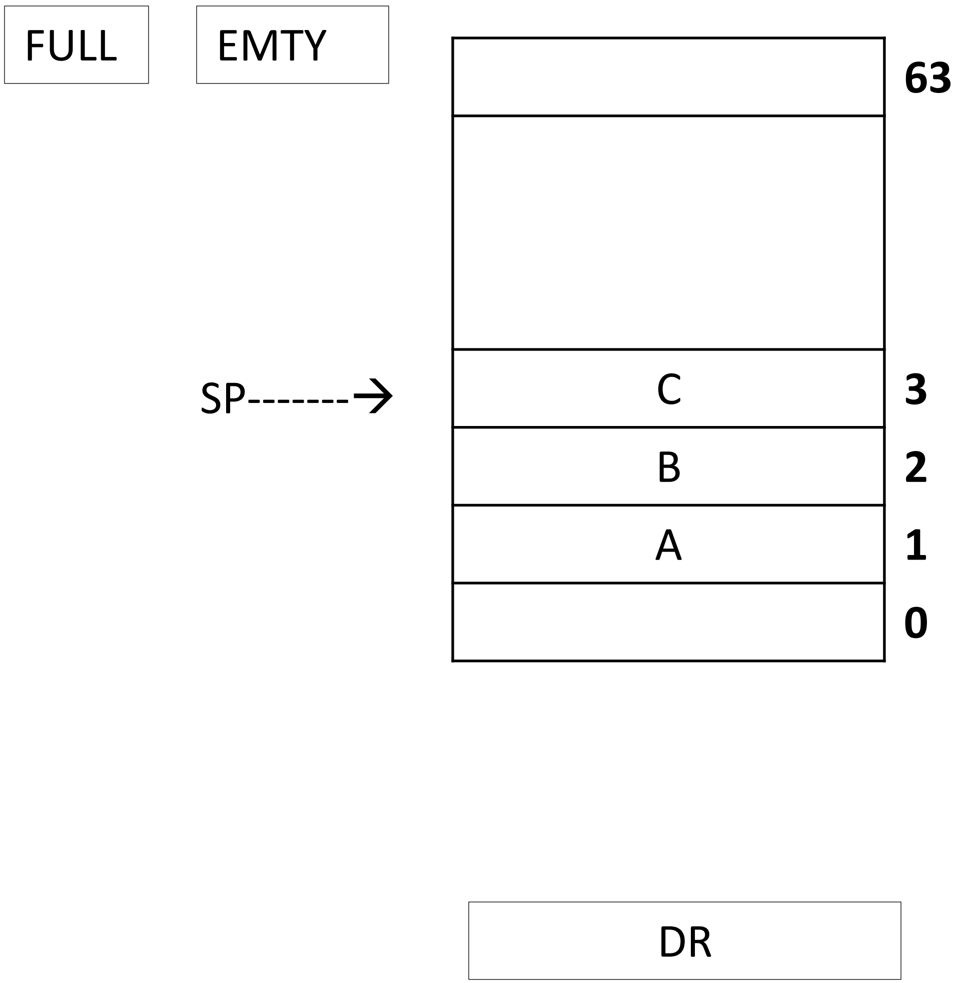


Original stack. After pop(). After push(83).

Stack Organization

- A stack can be placed in a portion of a large memory or it can be organized as a collection of a finite number of memory words or registers.
- The stack pointer register SP contains a binary number whose value is equal to the address of the word that is currently on top of the stack.
- Three items are placed in the stack: A, B, and C, in that order. Item C is on top of the stack so that the content of SP is now 3.
- To remove the top item, the stack is **popped** by reading the memory word at address 3 and decrementing the content of SP . Item B is now on top of the stack since SP holds address 2.
- To insert a new item, the stack is **pushed** by incrementing SP and writing a word in the next-higher location in the stack. Note that item C has been read out but not physically removed.

B.D OF 64-WORD STACK



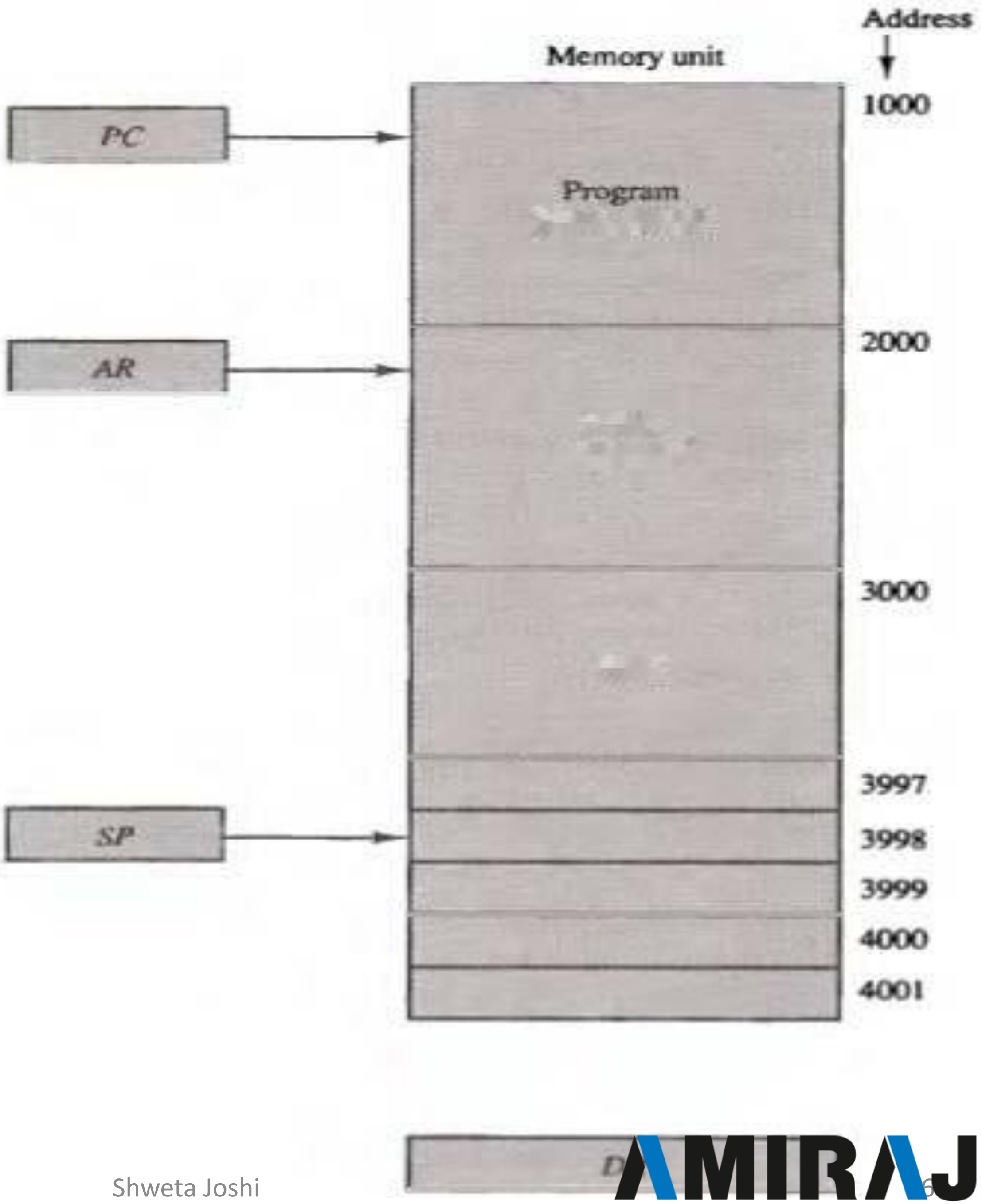
AMIRAJ
COLLEGE OF ENGINEERING & TECHNOLOGY

Stack Organization

- A stack can exist as a stand-alone unit as in Fig. or can be implemented in random-access memory attached to a CPU.
- The implementation of a stack in the CPU is done by assigning a portion of memory to a stack operation and using a processor register as a stack pointer.
- Figure 4 shows a portion of computer memory partitioned into three segments: program, data, and stack.
- The program counter PC points at the address of the next instruction in the program.
- The address register **AR** points at an array of data.
- The stack pointer **SP** points at the top of the stack.
- The three registers are connected to a common address bus, and either one can provide an address for memory.
- PC is used during the fetch phase to read an instruction.
- AR is used during the execute phase to read an operand.
- SP is used to push or pop items into or from the stack

STACK ORGANIZATION

As shown in Fig, the initial value of SP is 4001 and the stack grows with decreasing addresses.
Thus the first item stored in the stack is at address 4000 , the second item is stored at address 3999, and the last address that can
be used for the stack Is 3000.



Instruction Formats

- The bits of the instruction are divided into groups called fields. The most common fields found in instruction formats are:

1. An operation code held that specifies the operation to be performed.
2. An address field that designates a memory address or a processor register.
3. A mode field that specifies the way the operand or the effective address is determined.

- Computers may have instructions of several different lengths containing varying number of addresses. The number of address fields in the instruction format of a computer depends on the internal organization of its registers.

Instruction Formats

We will evaluate the arithmetic statement X = (A + B) * (C + D)

- **Three Address instruction**

ADD R1, A, B

ADD R2, C,D

MUL X,R1,R2

- **Two Address instruction**

MOV Rl, A

ADD Rl, B

MOV R2, C

ADD R2, D

MUL Rl , R2

MOV X, Rl

- **One address instruction**

LOAD A

ADD B

STORE T

LOAD C

ADD D

MUL T

STORE X

- **Zero Address instruction**

PUSH A

PUSH B

ADD

PUSH C

PUSH D

ADD

MUL

POP X

Instruction Formats

- RISC INSTRUCTIONS
- The instruction set of a typical RISC processor is restricted to the use of load and store instructions when communicating between memory and CPU.

ADDRESSING MODES

- Computers use addressing mode techniques for the purpose of accommodating one or both of the provisions:
- 1. To give programming versatility to the user by providing such facilities as pointers to memory, counters for loop control, indexing of data and program relocation.
- 2. To reduce the number of bits in the addressing field of the instruction.

Addressing Modes

- Implied Mode
- Immediate Mode
- Register Mode
- Register Indirect Mode
- Auto increment/Auto Decrement Mode
- Direct Mode
- Indirect Mode
- Relative Address Mode
- Indexed Addressing Mode
- Base Register Addressing Mode

Data Transfer and Manipulation

Data Transfer

Load	LD
Store	ST
Move	MOV
Exchange	XCH
Input	IN
Output	OUT
Push	PUSH
Pop	POP

Data Manipulation

1.Arithmetic instructions

Increment	INC
Decrement	DEC
Add	ADD
Subtract	SUB
Multiply	MUL
Divide	DIV
Addwithcarry	ADDC
Subtractborrow	SUBB
Negate (2‘s)	NEG

2. Logical and bit manipulation instructions

Clear	CLR
Complement	COM
AND	AND
OR	OR
Exclusive-OR	XOR
Clear carry	CLRC
Set carry	SETC
Complement carry	COMC
Enable interrupt	EI
Disable interrupt	DI

3. Shift instructions

Logical shift right	SHR
Logical shift left	SHL
Arithmetic shift right	SHRA
Arithmetic shift left	SHLA
Rotate right	ROR
Rotate left	ROL
Rotate rightthrucarry	RORC
Rotate leftthrucarry	ROLC

Program Control

Branch	BR
Jump	JMP
Skip	SKP
Call	CALL
Return	RET
Compare	CMP
(by subtraction)	
Test (by ANDing)	TST

RISC

The concept of RISC architecture involves an attempt to reduce execution time by simplifying the instruction set of the computer.

The major characteristics of a RISC processor are:

1. Relatively few instructions
2. Relatively few addressing modes
3. Memory access limited to load and store instructions
4. All operations done within the registers of the CPU
5. Fixed-length, easily decoded instruction format
6. Single-cycle instruction execution
7. Hardwired rather than micro programmed control

References

- **Images , descriptive Tables , from Computer System Architecture, Morris Mano, 3rd edition Prentice Hall**

- **Note: These pdf/ppt notes are for purpose of teaching aids to classroom/online sessions study, and in no case imply for GTU syllabus or GTU exam. For GTU syllabus or exam related preparation, one may, however will need to attend college/online lectures and refer books given by GTU in their official syllabus.**