

COMPUTER ARCHITECTURE AND  
ORGANISATION ASSIGNMENT

Name: J. Sadhana  
ECE-e  
15B81A04 DT

Q) Explain the following basic operational registers of a computer.

- A) Program Counter (PC)
- B) Memory Address Register (MAR)
- C) Memory Data or Buffer Register (MDR or MBR)
- D) Instruction Register (IR)

A)  
a) Program Counter (PC)

It holds the address of next instruction that is to be executed. It is called fetching memory (read instruction from memory)

b) Memory Address Register (MAR)

It contains the address of memory location which is to be accessed.

MAR is unidirectional i.e. address is transferred from MAR to memory. It is one of the specialised registers.

c) Memory Data or Buffer Register (MDR or MBR)

It contains the data that is to be written

or read by the memory

It is bidirectional i.e. read or written into memory.

d) Instruction Register (IR).

It holds the instruction that is to be executed currently.

MAR, MDR and IR are together called specialised registers.

⑩ Convert the  $(125)_{10}$  into the following codes

(a) Binary coded Decimal (BCD)

(b) Excess 3 code      (c) Gray code

to  
A)

$$(125)_{10} = 0000\ 0001\ 0010\ 0101 \rightarrow \text{BCD}$$

$$= 0011\ 0100\ 0101\ 1000 \rightarrow \text{Excess 3 code}$$

Gray code

$$\begin{array}{ccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \swarrow & \searrow & \swarrow & \searrow & \swarrow & \searrow & \swarrow \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{array}$$

$$(25)_{10} = 0000\ 0001\ 1100\ 0110 \rightarrow \text{Gray code}$$

⑪ Derive the truth for a 3 bit parity generator and 4 bit parity checker using an EVEN parity bit

(2)

Parity bit generation

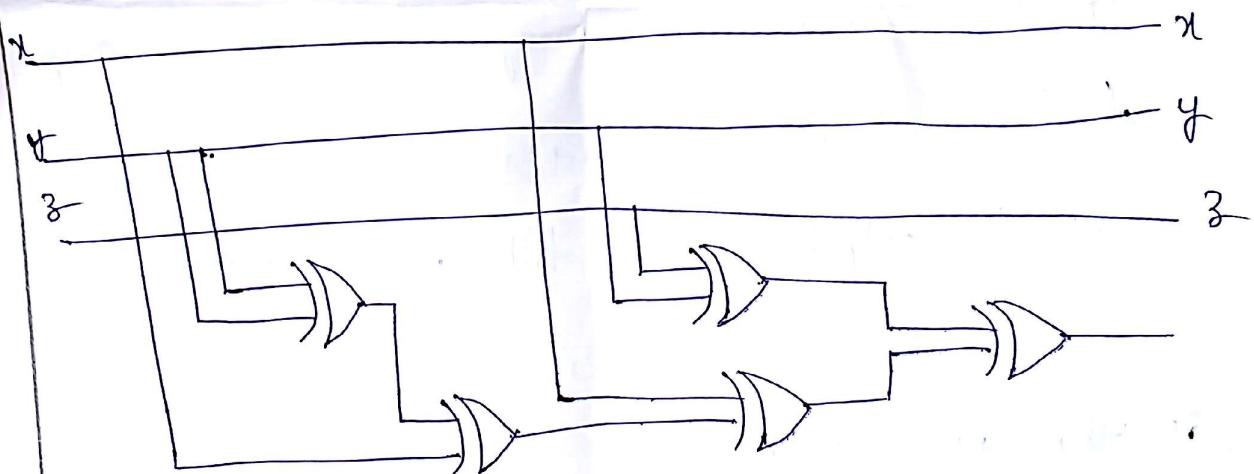
Message      P(even)

xyz

0 0 0	0
0 0 1	1
0 1 0	1
0 1 1	0
1 0 0	1
1 0 1	0
1 1 0	0
1 1 1	1

Source

Destination



Parity generator  
transmitter side

Parity checker  
Receiver side

Fig Error detection with even parity bit

Output = 0 indicates no errors and if contains even no. of 1's

Output = 1 indicates error and if contains even no. of 1's

## UNIT-II

- Q) Design and explain 4-bit adder/subtractor.

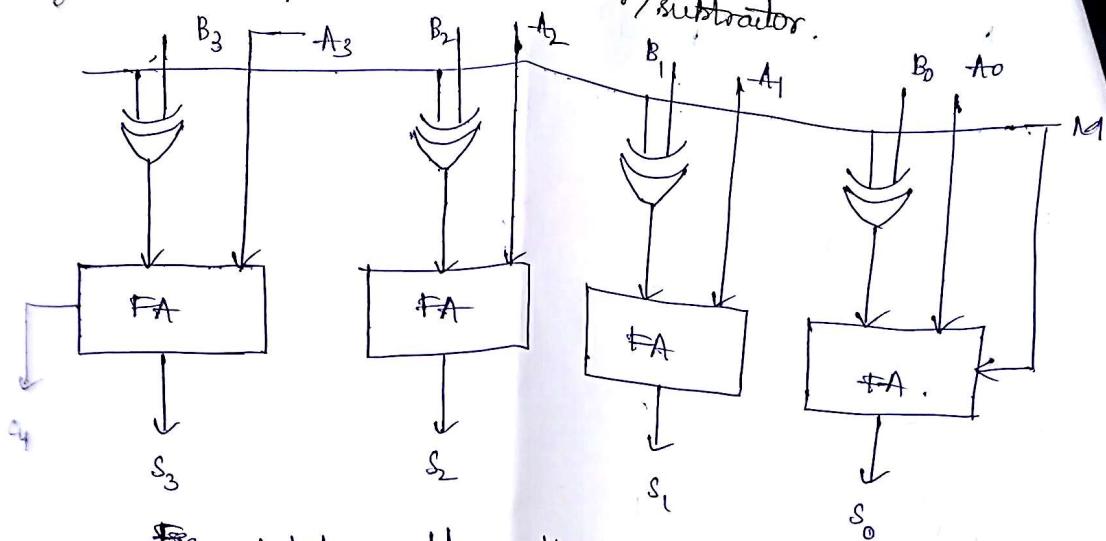


Fig: 4-bit adder-subtractor.

If  $M=0$ , Input carry carry  $C_0=0$  and the output of first full adder  $= A_0 + B_0$

Analogically output from all full adders  $= A+B$

$\therefore$  It acts like adder.

When  $M=1$ ,  $C_0=1$

The output is  $A+B+1$

(i)  $A-B$  for unsigned no's when  $A \geq B$

(ii) and 2's complement of  $B-A$  if  $A < B$

(iii) For signed numbers, if  $A+B$  when there is overflow.

The above circuit acts as both adder and subtractor.

(3)

design and explain one stage of arithmetic and logic shift unit.

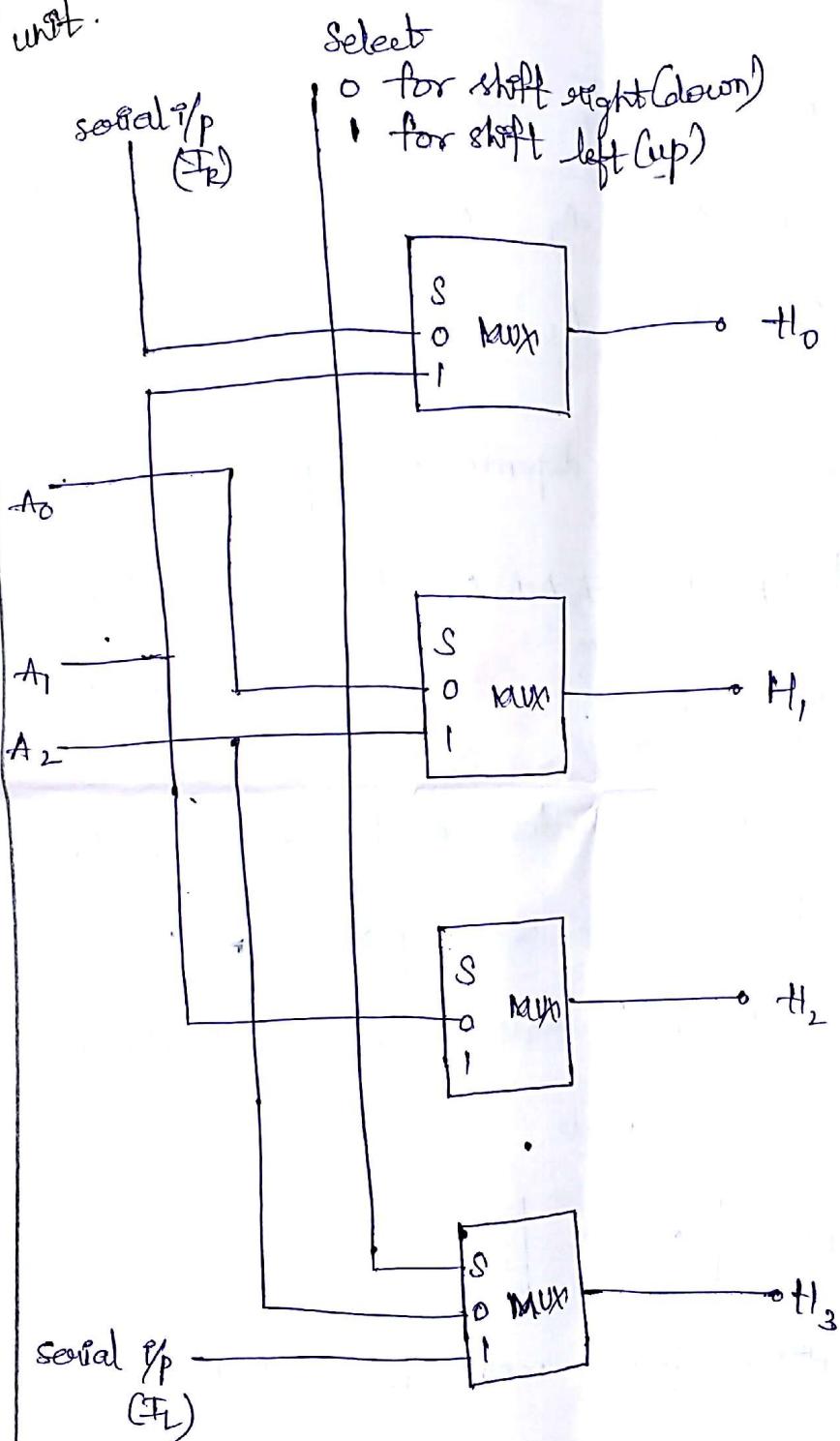


Fig. 4-bit combinational circuit shifter for both logical and arithmetic shift

$S_R \rightarrow$  serial i/p for shift right

$S_L \rightarrow$  serial i/p for shift left

Function Table:

<u>select</u>	<u>output</u>			
s	$t_0 \ t_1 \ t_2 \ t_3 \dots$			
0	$I_R \ A_0 \ A_1 \ A_2$			
1	$A_1 \ A_2 \ A_3 \ I_L$			

The values of  $I_R$  and  $I_L$  depends on operation (logie or arithmetic). Actual if is  $A_0 A_1 A_2$ . It is shifted to right/left depending on 's'.

- (3) Explain different addressing modes with one example.  
 a) Addressing mode specifies the way operand or effective address is determined. The mode field (in instruction) indicates the addressing mode.

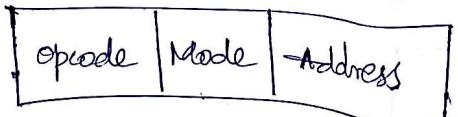


Fig Instruction format with mode field.  
 There are different addressing modes.

i) Implied mode:

In this mode, the definition of instruction specifies the location of operand.

∴ In implied mode, address field does not exist.

### 2) Immediate mode:

Operand is specified in instruction in immediate mode.

### 3) Register mode:

In this mode, the address of the operand is contained in the register specified in the instruction.

### 4) Register indirect mode:

In this mode, the address of the operand is contained in the register specified in the instruction.

### 5) Autoincrement or Autodecrement mode:-

It comes under indirect mode.

In autodecrement mode, register is decremented after the execution of instruction.

In autodecrement mode, register is decremented prior(before) to the execution of instruction.

### 6) Direct Address mode:

In this mode, the address part of the instruction specifies effective address.

### 7) Indirect Address mode:

→ In this mode, address part of the instruction indicates

address of effective address ie address of the instruction gives effective address.

#### 8) Relative Address mode:

→ In this mode, effective address is given by adding the content of the PC to the address part of the instruction.

#### 9) Indexed addressing mode:

→ In this mode, effective address is given by adding the content of the index register to the address part of the instruction.

#### 10) Base Register Addressing mode:

→ In this mode, effective address is given by adding the content of base register to the address part of the instruction

#### Example for addressing modes

Initially  $PC = 200$

Instruction at 200  
is fetched.

$PC = 200$

$R1 = 400$

$AR = 100$

$AGA$

$AC$

Memory field	
200	Load to AC/ Mode
201	Address = 300
202	next instruction
399	400 450
400	700
500	800
600	900
702	325
800	300

(5)

R1 → Processor register

XR → Index register.

<u>Addressing Mode</u>	<u>Effective Address</u>	<u>Content of Ae</u>
Direct Address	500	800
Immediate operand	201	500
Indirect Address	800	300
Relative address.	702	325
Indexed address	600	900
Register direct	-	400
Register indirect	400	700
Autoincrement	400	700
Autodecrement	399	450.

(6)

Write short notes on the following.

a) Stack organisation.

b) Instruction formats

c) RISC

A)

Stack organisation: Stack is a storage device which follows last in first out.

→ Register stack is a stand alone type

Memory stack will be part of RAM.

SP → stack pointer is always used to indicate the top most filled location of the stack.

→ FULL and EMPTY are the flip-flops.

FULL will be set when stack is full

EMPTY " " " " " " " empty

→ Initially 1, 2, ..., 63 locations are filled and then 0 is filled. ∵ first item is taken from 0, then 63, 62, ..., 1.

### b) Instruction formats

Instruction format consists of mode, operational code and address fields. Address fields contains address of operand or processor register.

### c) RISC

Reduced Instruction Set Computer (RISC)

It contains

- 1) Relatively few instructions
- 2) Relatively few addressing modes
- 3) Memory access limited to load and store instruction  
(no manipulation)
- 4) All operations are done within the registers of CPU

(B)

- Fixed length, easily decoded instruction format.
- Single cycle instruction execution
- It uses hardwired rather than microprogram control.

(B) Evaluate the arithmetic statement  $x = (A+B)*(C+D)$  using zero, one, two and three address instructions.

(A) Three address Instructions:

$$\text{ADD } R_1, A, B \quad R_1 \leftarrow M[A] + M[B]$$

$$\text{ADD } R_2, C, D \quad R_2 \leftarrow M[C] + M[D]$$

$$\text{MUL } X, R_1, R_2 \quad M[X] \leftarrow R_1 * R_2$$

Two Address Instructions:

$$\text{MOV } R_1, A \quad R_1 \leftarrow M[A]$$

$$\text{ADD } R_1, B \quad R_1 \leftarrow R_1 + M[B]$$

$$\text{MOV } R_2, C \quad R_2 \leftarrow M[C]$$

$$\text{ADD } R_2, D \quad R_2 \leftarrow R_2 + M[D]$$

$$\text{MUL } R_1, R_2 \quad R_1 \leftarrow R_1 * R_2$$

$$\text{Mov } X, R_1 \quad M[X] \leftarrow R_1$$

1 Address Instructions (Stack)

$$\text{LOAD } A \quad AC \leftarrow M[A]$$

$$\text{ADD } B \quad AC \leftarrow AC + M[B]$$

$$\text{STORE } T \quad M[T] \leftarrow AC$$

$$\text{LOAD } C \quad AC \leftarrow M[C]$$

$$\text{ADD } D \quad AC \leftarrow AC + M[D]$$

MUL T

$$Ac \leftarrow Ac * M[T]$$

Zero-address

Instructions:

PUSH A

$$TOS \leftarrow A$$

PUSH B

$$TOS \leftarrow B$$

ADD

PUSH C

$$TOS \leftarrow A+B$$

PUSH D

$$TOS \leftarrow C$$

ADD

MUL

$$TOS \leftarrow C+D$$

POP X

$$TOS \leftarrow (A+B) * (C+D)$$

$$M[X] \leftarrow TOS$$

(+) ④

Explain the following operations in detail with example.

- a) Basic microoperations      b) Arithmetic shift      c) Circular shift
- d) Rotate shift.

(+) ⑤

a) Basic Microoperations:

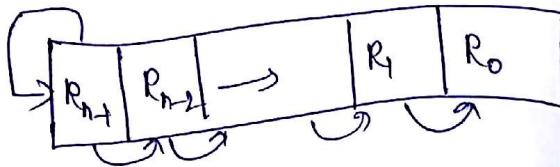
They are defined as the operations performed on data stored in registers. They are: Arithmetic, logical and data transfer operations.

Eq!  $R_2 \leftarrow R_1, R_3 \leftarrow R_1 + R_2, R_3 \leftarrow R_1 - R_2, R_2 \leftarrow \bar{R}, R_2 \leftarrow \bar{R}_2 + 1,$   
 $R_3 \leftarrow R_1 + \bar{R}_2 + 1$

b) Arithmetic shift:

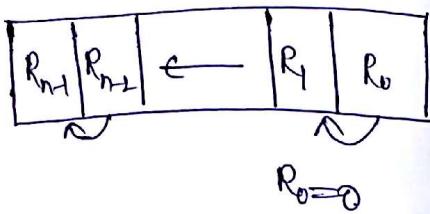
It is used to shift the signed binary numbers to left or right.

### (i) Arithmetic shift right



It divides the signed binary numbers by 2.

### (ii) Arithmetic shift left:



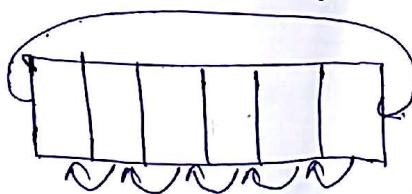
After ~~right~~ shift left operation, if the sign changes, overflow occurs.  $V_s = R_{n+1} \oplus R_{n+2}$

$V_s = 0 \Rightarrow$  If  $R_{n+1} = R_{n+2} \rightarrow$  No overflow.

### (c) Circular shift:

The serial o/p of shift register is connected to the serial i/p of shift register.

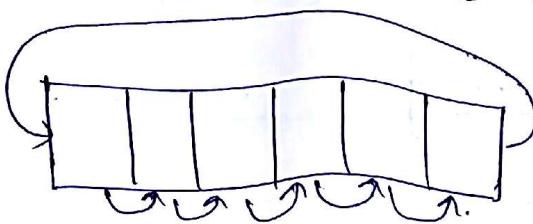
Ex: 1)



→ circular shift left

MSB bit is transferred to LSB

2)



→ circular shift right

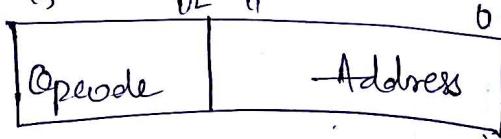
In circular shift right operation LSB is transferred to MSB.

⑩ Draw instruction format of a computer (16 bits) and Microinstruction format of a computer (30 bits). Explain diff. fields.

#### Instruction Format:

It consists of 16 bits.

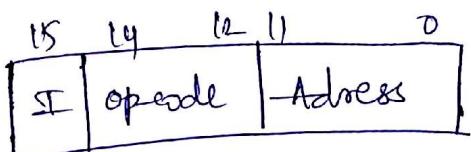
It consists of operation code and Address field



#### Instruction Format.

There are 3 types of instructions.

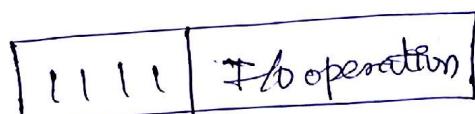
- 1) Memory reference
- 2) Register Reference
- 3) Input-output instruction.



a) Memory reference instruction



b) Register reference instruction.



c) Input-Output instruction.

→ Memory is referred in case of memory reference instruction in order to get operand. Bit I indicates whether it is direct addressing mode or indirect addressing mode.

3 bits are used to specify the type of operation.

Address of operand is called as effective address.

→ Register reference instruction: For this instruction, opcode is 111 I=0. 4 bits are used to specify the operation to be performed on Accumulator and test is also performed on accumulator.

→ Input-Output Instruction:

If opcode = 111, it indicates I/O instructions and I=1

It defines the type of I/O operation and test performed.

## Micro instruction Format

3	3	3	2	2	1
$F_1$	$F_2$	$F_3$	CD	BR	AD

$F_1, F_2, F_3$  : Microoperation fields  $\rightarrow$  specify microoperations

CD : Condition for branching

BR : Branch field (Type of branch)

AD : Address field.