Unit - 4 : Central Processing Unit.

1 Explain General Register Organization with block diagram.

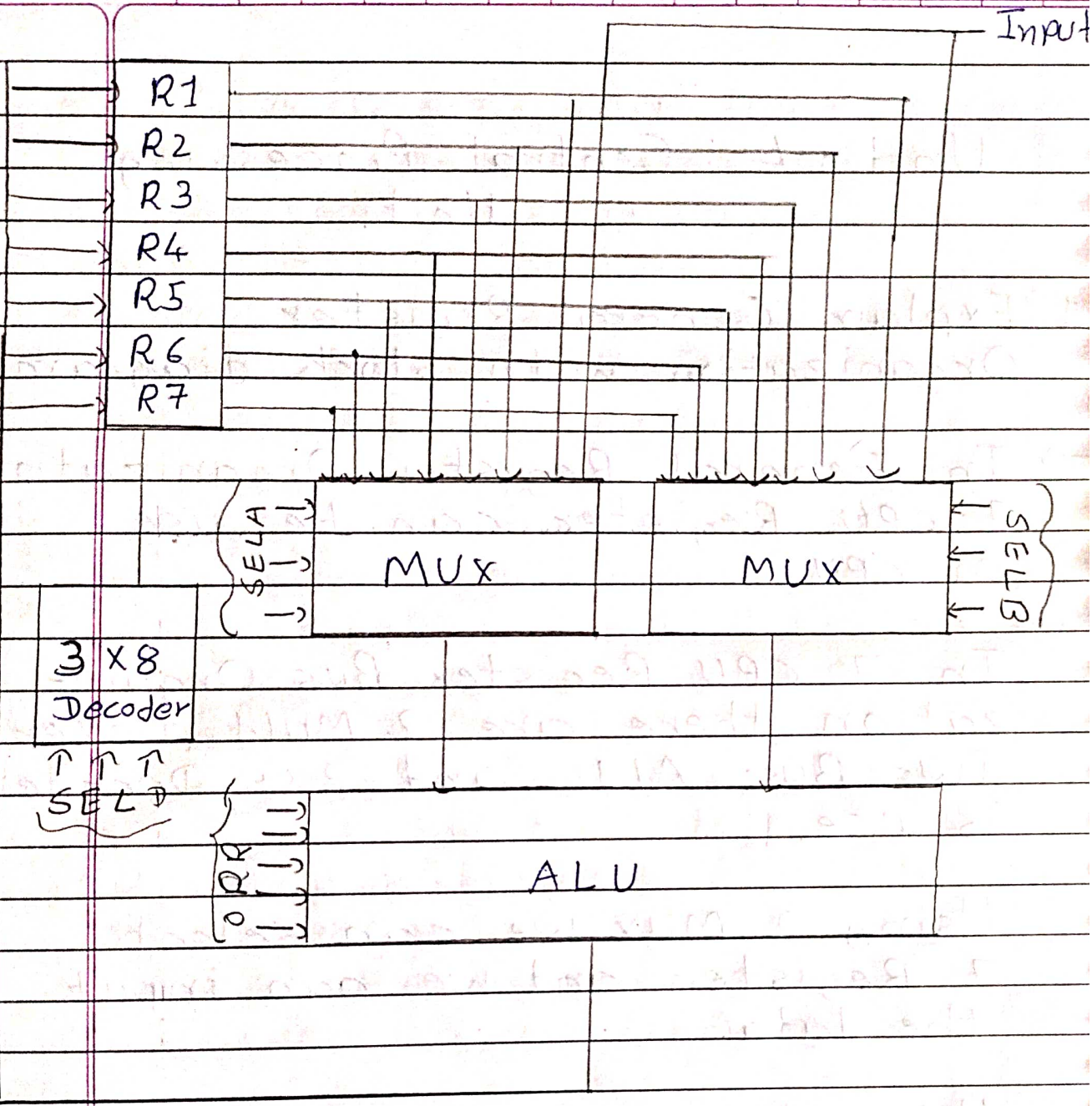
=> In General Register Organization 7 CPU Register can be use in CPU.

In 7 CPU Register Bus Organization there are 2 Multiplexer, Two Bus, ALU and 3x8 Decoder is use.

Using 2 Mux we can select 7 Register and we can input the data.

Using two Bus we can transfer the data in the ALU.

ALU can perform all the Micro-operation and Transfer output using 3x8 Decoder into the register.



3	3	3	5
SELA	SELB	SELD	OPR

Instruction.

2 Explain Control Word format For different Micro-operation.

=> For Control the General Register Oraganization we have to use this word.

ORR	Operation
00000	Transfer A
00001	Increment A
00010	ADD A + ADD B
00101	Sub A - Sub B
00110	Decrement B
01000	AND A and B
01010	OR A and B
01100	XOR A and B
01110	Complement A
10000	Shift right A
11000	Shift Left A

(* Explain Table in own word.

Page No.
Date : / /

3 Draw and Explain Block Diagram of 64 - word stack with Full and Empty Flag.

=> Stack is use to nested subroutines and nested interrupt services.

Stack is follow LIFO structure.

For stack we have to follow SP Pointer.

For Stack Oraganization we have to use Push and PoP operation.

There are two types of Stack Oragani zation.

1) Register stack Organization

2) Memory stack Organization.

1 Register stack Organization.

For Stack Organization, Pointer SP = 0 , Flag Empty = 1 and Full = 0.

		6
		5
		4
SP →	C	3
	B	2
	A	1
		0

Push Operation :

SP ← SP + 1
M[SP] ← DR
IF (SP = 0) then (Full ← 1)
Empty ← 0

Pop Operation :

DR ← M[SP]
SP ← SP - 1
IF (SP = 0) then (Empty ← 1)
Full ← 0

Page No.
Date : / /

4 Explain Stack organization using Push and Pop instructions.

=> (* Write Que-3 Register Stack Oraganization Operation. and add this Reaming part)

For Memory Stack Organization,

Push Operation:

$$SP \leftarrow SP - 1$$
$$M[SP] \leftarrow DR$$

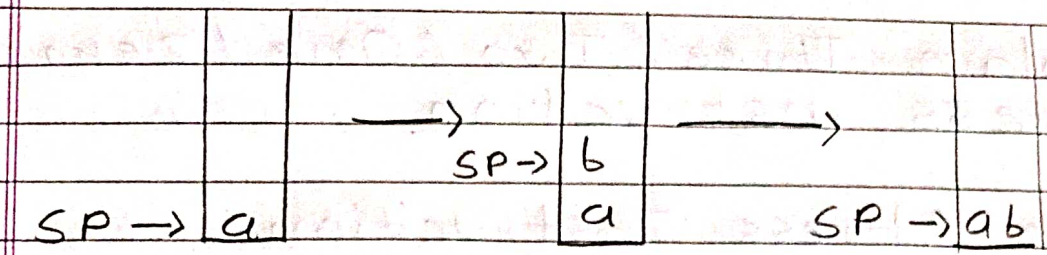
Pop Operation:

$$DR \leftarrow M[SP]$$
$$SP \leftarrow SP + 1$$

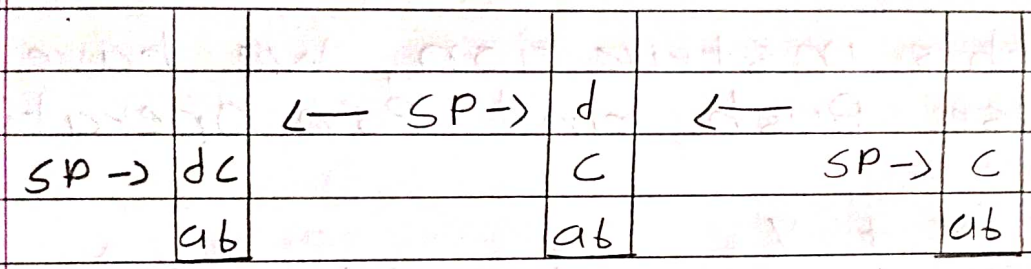
5 Use stack for solving Arithmetic expression using Reverse Polish Notation.

=> Ex. $(a * b) + (c * d)$

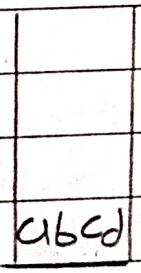
=>



Input a Input b Input *



Input * Input d Input c



input +

6 Explain Three / Two / One / Zero address instruction.

=> Zero Address Instruction:

In this address instruction computer does not use an address field.

In this instruction, we have to use Push and Pop operation.

Ex. A = X + Y

```
Push X
Push Y
ADD
POP A
```

=> One Address Instruction:

In this address instruction, line contain only one address.

In this instruction one address can be a register name or Memory.

Ex. LDA 2005 // Addition
ADD X

=> Two address instruction :

In this instruction, instruction contain only two address field.

Ex. A = X + Y

MOV R_1, X
ADD R_1, Y

=> Three address instruction :

In this instruction, instruction contain three address field.

Three address instruction reduse the size of program.

Ex. X = A + B

ADD R_1, A, B

7 Explain different Addressing Mode of instruction.

=> There are eight addressing mode of instruction.

1 Implied Mode:

In implied addressing mode the operand is specified in the instruction itself.

Zero Address instruction is use for implied mode.

Ex. Push A

2 Immediate Mode:

In Immediate addressing mode the present in the instruction.

One Address instruction is use for immediate mode.

Ex. MOV A, 35H

Gurukrupa Xerox Services

3 Register Mode:

In this addressing mode, instruction contain register address.

Designated operand need to be in a register.

Ex. MOV A, B

4 Register Indirect Mode:

In this instruction, instruction specifies a register which contains the memory address of Program.

Ex. LDA 2005

Instruction does not contain the operand but contain address of operand.

5 Autoincrement or Autodecrement Mode:

When the address in the register is used to access memory, the

value in the register is increment and decrement by 1 automatically.

6 Direct Address Mode:

Instruction specifies the memory address which can be used directly to access the memory.

In this address mode, operand can be access directly to the memory.

Ex. LDA 2050H

7 Indirect Address Mode:

Instruction specifies the address of a memory location that contain a address of the operand.

In this address mode, operand can not be access directly to the memory.

8 Relative Address Mode:

The address fields of an instruction specifies the part of the address.

There are three type of Relative address mode.

(i) PC Relative Address Mode:

EA = PC + IR(address)

(ii) Indexed Relative Address Mode:

EA = IX + IR(address)

(iii) Base Register Address Mode:

EA = BAR + IR(address)

(BAR = Base address Register)

8 Explain Direct, Register, Indirect and Register indirect addressing mode.

=) Write - Que - 7 - Addressing Mode

Q Explain Data Transfer Instruction with example.

=> This are the different Data Transfer Instruction.

1 Load : Transfer the data from memory to ass accmulator.

Ex. LDA 2000H

2 Store : Transfer the data from accmulator to memory.

Ex. STA 2000H

3 Move : Transfer the data from Register to memory.

Ex. MOV B, C

4 Exchange : Swaps the data either between two register or memory and register.

5 Input : Transfer data between the register and the input terminal.

6 Output : Transfer data between the register and output termina

7 Push and Pop : Transfer data between the register and memory stack.

Ex. Push B
Pop A

10 Explain Arithmetic Instruction with example.

=> This are the basic arithmetic instruction.

1 ADD : Add the two data.

Ex. add B

2 Subtract : Subtract two data.

Ex. SUB A

3 Add with Carry : Add the two data with carry.

Ex. ADDC B

4 Subtract with Borrow : Subtract the two data with its borrow.

Ex. SUBB B

5 Increment : Increment the data.

Ex. INC

6 Decrement : Decrement the data.

Ex. DEC

7 Multiply : Multiply two data.

Ex MUL B

8 Divide : Divide two data.

Ex. DIV B

11 Explain Logical Instruction with example.

=> This are the Basic logical instruction.

1 Clear : Clear the Register or Memory.

Ex. CLR

2 Complement : Complement the data.

Ex. COM

3 AND : Perform AND operation on data.

Ex. AND B

4 OR : Perform OR operation on data.

Ex. OR B

5 Exclusive-OR : Perform XOR operation on data.

Ex. XOR B

6 Clear Carry : Clear the carry flag

Ex. CLRC

7 Set carry : Set the carry in the flag.

Ex. SETC

8 Complement carry : Complement the carry flag.

Ex. COMC

9 Enable Interrupt : Enable the interruput in program.

Ex. EI

10 Disable Interrupt : Disable the interrupt in program.

Ex. DI

12 Explain Shift Instruction with its example.

=> This are the basic shift instruction.

1 Logical Shift Right : Perform logical shift right operation.

Ex. SHR B

2 Logical Shift Left : Perform logical shift left operation.

Ex. SHL B

3 Arithmetic Shift Right : Perform arithmetic shift right operation

Ex. SHRA B

4 Arithmetic Shift Left : Perform arithmetic shift Left operation

Ex. SHLA C

5 Rotate Right : Perform rotate right operation.

Ex. RoR C

6 Rotate Left : Perform rotate left operation.

Ex. ROL C

7 Rotate Right thru carry : Perform Rotate operation with carry flag.

Ex. RORC D

8 Rotate Left thru carry : Perform Rotate operation with carry flag.

Ex. ROLC D.

13 Explain Program Control Instruction.

=> This are the Basic Program Control instruction.

1 Branch : Branch instruction is use to Perform different condintional branch instruction.

2 Jump : Jump instruction update the program counter and execute another location inside the program.

3 Skip : Skip instruction is used to skip next instruction in the program.

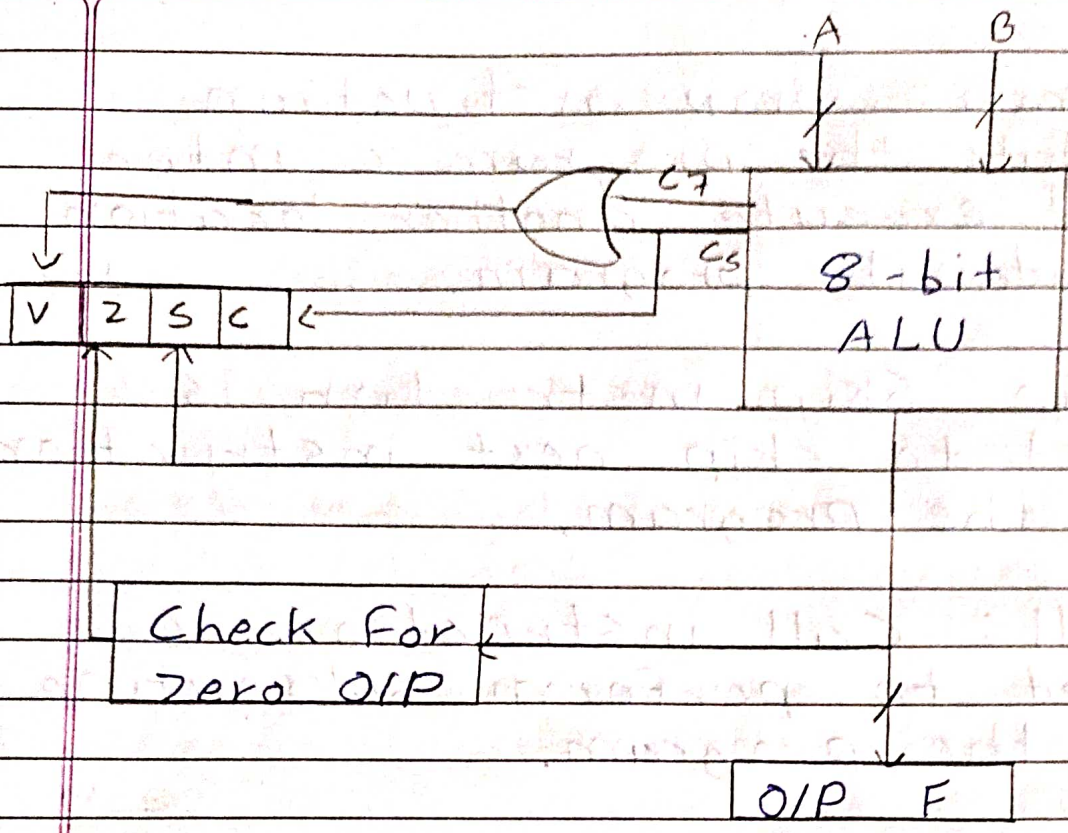
4 Call : Call instruction is used to perform subroutine in the program.

5 Return : Return instruction is returns control from a subroutine back to the calling program.

6 Compare : Compare instruction is used to compare two value in the program.

14 Explain Status Register bits using example.

=> The bits of the status register are modified according to the operation performed in ALU.



These are the Status Register Flags

1 C (Ccarry) : If carry out from ALU than set to 1

2 S (Ssign) : The MSB bit of the ALU's output.

3 Z (Zzero) : If ALU's all output are zero than set to 1.

4 V (Overflow) : If there is overflow than set to 1.

15 Define Interrupt with its type.

=> An interrupt is a signal to stop to execution of CPU.

Interrupt signal can be an internal or external signal.

When CPU get any IO operation than interrupt stop the execution of program.

There are two type of Interrupt

1) Hardware Interrupt

2) Software Interrupt.

1 Hardware Interrupt:

Hardware Interrupt is also called external interrupt.

If the signal is generated by external devices than Hardware interrupt is accures.

Ex. If we click a mouse to do some action this clicking

the mouse generates signal that is given to the processor to do action such interrupt are called Hardware interrupt.

2 Software Interrupt :

Software Interrupt is also called internal interrupt.

Internal interrupt arise where illegal instruction are use.

Software interrupt are initiated by the executing an instruction.

16 Explain Difference between RISC and CISC.

	RISC	CISC
1	It is a Reduced Instruction set Computer.	It is a Complex Instruction set Computer.
2	Execution time is Short	Execution time is Large.
3	Require more memory register.	Require less memory register.
4	Fixed format instruction.	Variable format instruction.
5	Pipline is simple.	Pipline is difficult.
6	Simple decoding instruction.	Complex decoding instruction.
7	Less Cycle Time	More Cycle Time.
8	Size of Program is Long.	Size of Program is small.