

Practice Q/B - 1 SolutionObjective Type Question

- Q.1 The CME and CLE instructions needed in the basic computer in order to set the E flip-flop to 1.
- Q.2 Computer Organization is concerned with the way the hardware component operates and the way they are connected together to form the computer system.
- Q.3 The third state in a tri-state buffer is High-Impedance state (where circuit behaves as open-circuit).
- Q.4 In a basic computer, the type of instruction are memory-reference, register-reference, and I/O-reference instructions.
- Q.5 In memory-reference instruction, the operation bits specify from 000 (D₀) to 110 (D₆).
- Q.6 The devices that provide the means for a computer to communicate with the user or other computers are referred to as: I/O.
- Q.7 The bit sequence 0010 is serially entered (from right-most bit first) into a 4-bit parallel out shift register that is initially clear. What are the Q outputs after two clock pulses?
- Sol. Bit = 0010
- Initial = 0000
- At 1st clock Insert right most bit of i.e. '0' \Rightarrow 0000 (last bit)
- At 2nd clock Insert next right most bit ie. '1' \Rightarrow 1000 (")
- So after two clock pulses Q output = 1000.
- Q.8 Function of Program Counter is to Hold the address of instruction.
- Q.9 Which of the computer register holds the instruction code? Instruction Register.
- Q.10 The symbolic description of transfer from one register to another register is known as Register Transfer Language (RTL).

Short Question

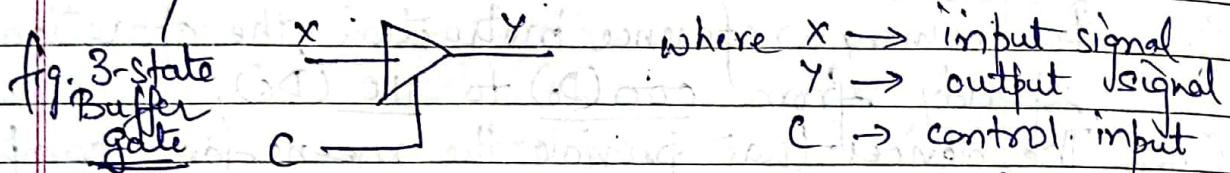
Q.1 What do you mean by register transfer language (RTL)?

Sol. The symbolic notation used to describe the micro-operation transfer among registers is called a register transfer language (RTL).

Q.2 Draw a diagram of a bus system using 3-state buffer and a decoder.

Sol. Take an example of a digital computer system having 4 registers and each register is of 2-bit.

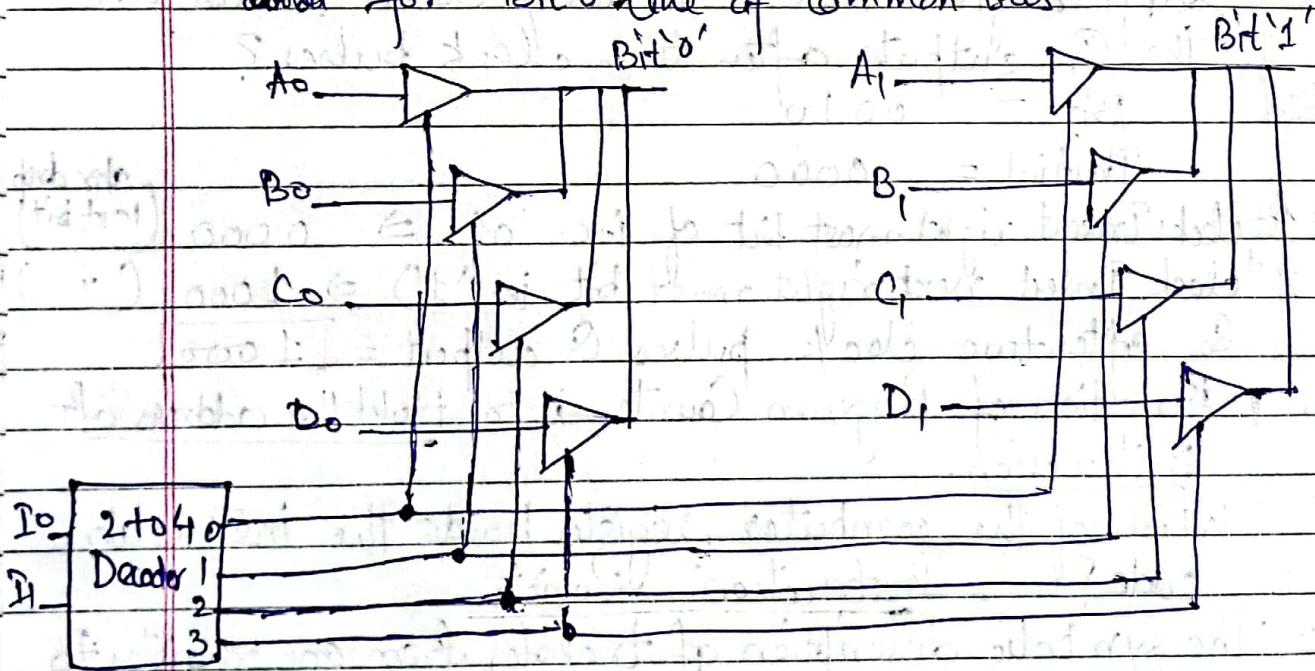
Now, we draw diagram of bus system for this system.



If $C = 0$ (The buffer gate moves to High-Impedance state)

If $C = 1$ (The buffer gate performs normal)

As we have 4 registers, so we require 4 buffer gates and for Bit '0' line of common bus.



→ This block of gate buffer set will keep on repeating as much as times as no. of bits in each register. But, the decoder will remain only one.

I_1, I_0	Register Select
0 0	Reg. A
0 1	Reg. B
1 0	Reg. C
1 1	Reg. D

Q.3 Explain $P : R_3 \leftarrow R_5$ statement.
Sol: If ($P=1$) then transfer the contents of R_5 to register R_3 .

Q.4 Draw the block diagram of control unit of basic computer and explain.

Sol: The control unit of basic computer is as follows:

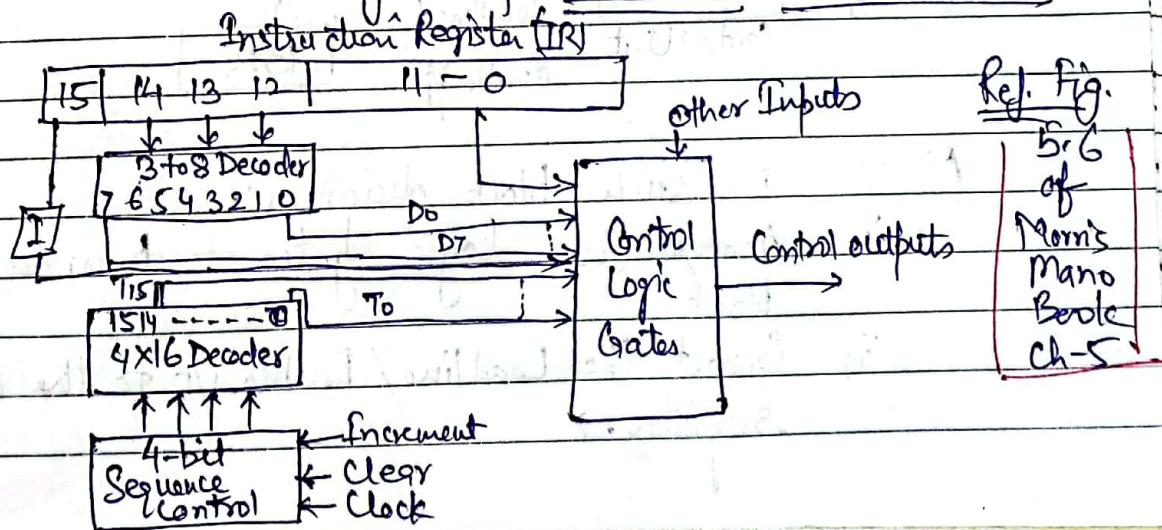
The timing signals for all registers in the basic computer is controlled by master clock generator.

These clock pulses are applied to all flip-flops and registers in the system.

There are two types of control organization:

1. Hardwired Control Unit (Logic is implemented through decoders, & other digital circuits)
2. Micro-programmed Control Unit (control info. is stored in control mem. in form of control words)

Below is the diagram of Hardwired Control Unit



Explanation

The Control Unit consists of 2 decoders, a sequence counter, and a no. of control logic gates.

An instruction is fetched from the memory and is placed in instruction register (IR) where it divides into 3 parts:

IR (0-11) \rightarrow applied to Control logic gates

IR (12-14) \rightarrow applied to 3×8 decoder as op-code

The decoder generates 8 outputs from D₀ to D₇

IR (15) \rightarrow transferred to 1 flip-flop

The 4-bit sequence counter generates timing signals from T₀ - T₁₅ that are used to synchronise with the help of a 4x16 decoder.

The sequence counter (SC) has 3 i/p's

i) Increment (INR)

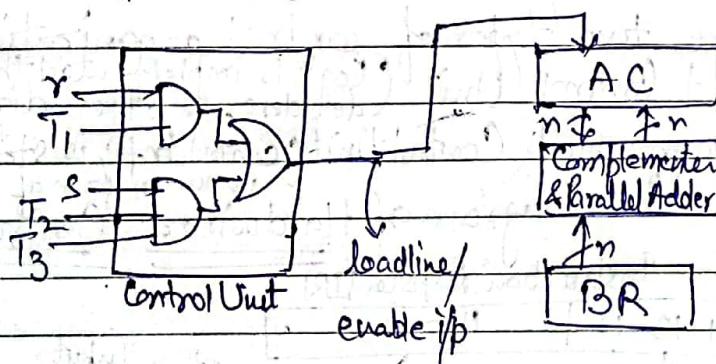
ii) Clear (CLR)

iii) Clock (CLK.)

Q.5 Draw the block diagram for the hardware that implements the following statement:

$$\text{or } T_1 + S T_2 T_3 : AC \leftarrow AC - BR$$

Sol.



Note: for such block diagram:

i) ~~Always~~ Use logic gates to draw control unit block

ii) Connect the Loadline/ Enable i/p to the destination register.

Q.6

Design a 4-bit combination circuit 'incrementer' using four full adder.

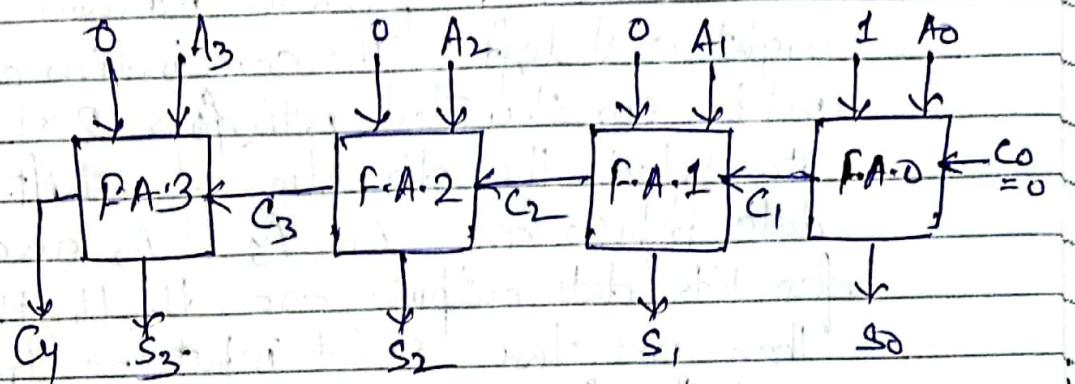
Sol:

Increment micro-operation $\Rightarrow A + 1$

where A word is of 4-bit and 1 is also has 4-bit representation as 0001

$$\text{i.e. } A_3 A_2 A_1 A_0 + 0001$$

A Full Adder has 3 input variables and 2 output variables. To add 4-bit, we have to use 4-bit parallel adder.



Q.7 List various registers with their function required for basic computer function.

Sol.	Register	Symbol	Size	function
1.	Data Register	DR	16-bit	Holds the operand/data to be processed
2.	Address Register	AR	16-bit	Holds the address of the operand
3.	Instruction Register	IR	16-bit	Holds the instruction code
4.	Temporary Register	TR	16-bit	Holds the temporary word
5.	Accumulator	AC	16-bit	Processor register
6.	Program Counter	PC	12-bit	Holds the addr. of the instruction
7.	Input Register	INPR	8-bit	Holds an 8-bit I/p character
8.	Output Register	OUTR	8-bit	Holds an 8-bit o/p character

Q.8 Tabulate various shift micro-operation and design a 4-bit combination shifter circuit.

- Sol. There are three type of shift micro-operations
- Logical Shift
 - Circular Shift
 - Arithmetic Shift

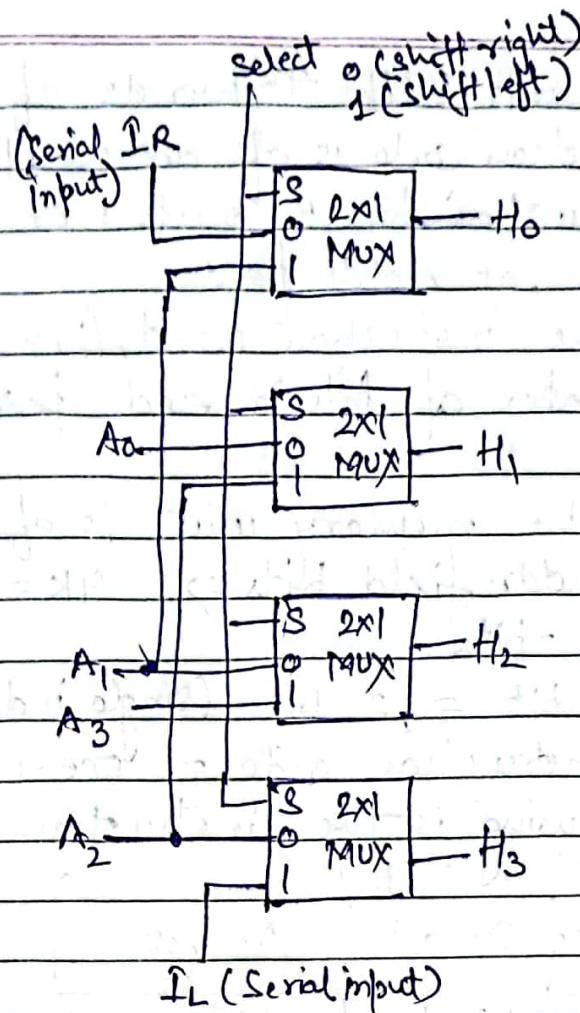
The shifter circuit is a bidirectional shift register with parallel load. Information can be transferred to the register in parallel and then shifted to the right or left. The combination circuit shifter can be constructed with four 2×1 Multiplexer to select the direction of shift. The four bit data inputs are A_0, A_1, A_2 , and A_3 and the four bits data outputs are H_0, H_1, H_2 , and H_3 .

There are two serial inputs $\rightarrow I_R$ for shift right and I_L for shift left.

When selection input (S) of multiplexer is 0, the data is shifted right $\rightarrow I_R, A_0, A_1, A_2$. and if $S = 1$ then data is shifted left $\rightarrow A_1, A_2, A_3, I_L$.

Functional Table

Select (S)	Outputs			
	H_0	H_1	H_2	H_3
0	I_R	A_0	A_1	A_2
1	A_1	A_2	A_3	I_L



Q.9 Initial $\Rightarrow R = 10001101$

Sol. In sequence, we perform following functions

↳ it means we have to perform ops on the previous 4-op. result.

i) Logical Shift left $\Rightarrow (\leftarrow)$ $\boxed{10001101}$ $I_L = 0$

ii) Circular Shift Right $\Rightarrow (\circlearrowright)$ $\boxed{00001101}$

iii) Logical Shift Right $\Rightarrow (\rightarrow)$ $I_R = 0$ $\boxed{00000110}$

iv) Circular Shift Left $\Rightarrow (\circlearrowleft)$ $\boxed{00001100}$

Q.10

Memory unit with 4K words of 16-bit each.

Instruction code is of one word.

The instruction has 3 parts: an indirect bit, an opcode, an addr. part.

Draw the instruction word format and indicate the number of bits in each part.

Sol.

Since, the memory unit is of 4K words =

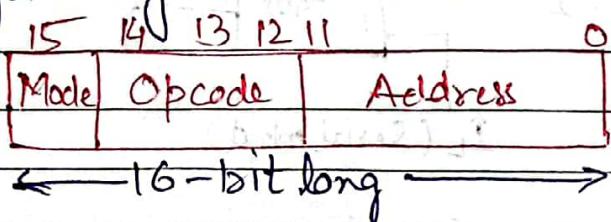
$$\text{So, addr field bits} \Rightarrow 4K = 2^2 \times 2^{10} = 2^{12}$$

Addr. bits = 12 - bits

Mode bit = 1-bit (single indirect mode)

Total instruction code = one-word = 16-bit

The following is the instruction format



The opcode field $\Rightarrow 16 - (12+1) = 16-13 = 3\text{-bits}$

Q.11

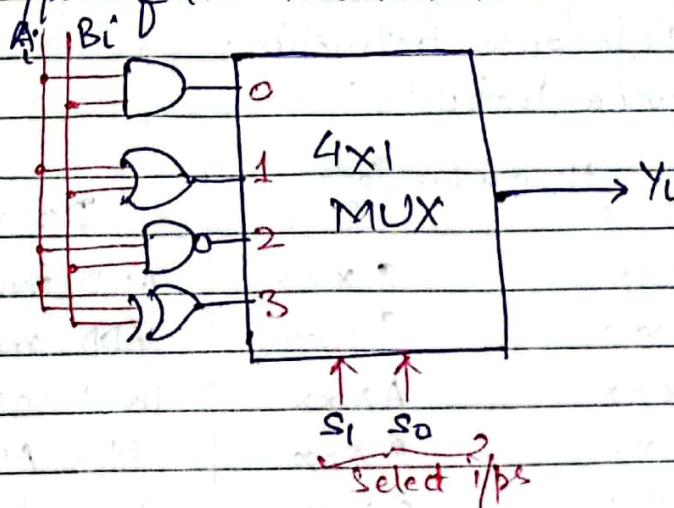
Describe the hardware implementation of logic micro-operations. Draw the diagram of one-stage of logic circuit used with AND, OR, NAND and XOR gates.

Sol.

Hardware implementation of logic micro-operation requires that logic gates be cascaded for each bit or pair of bits in the registers to perform the required logic function. Although, there are general 16 - microoperations. But we can design with given gates.

One - stage of a circuit that generates the four basic / given logic de-ops. It consists of four gates: AND, OR, NAND and XOR and a

multiplexer. Each of the four logic m-ops is generated through a gate that performs the required logic. The o/p's of the gates are applied to the data i/p's of the multiplexer. Since, 4 gates, the size of MUX will be 4×1 and therefore, the select lines is 2 (S_1, S_0) to select one of the data i/p's of the MUX and directs its value to the o/p.



S_1, S_0	Logic M-Op Select
0 0	AND
0 1	OR
1 0	NAND
1 1	EX-OR

Q.12 Draw a diagram of bus system for 4 registers, each of 8-bit using 3-state buffer and a decoder.

Sol: Refer Q.2 for this

Only thing is that referred diagram is for 4-registers and 2-bit each size.

⇒ for this question 12, we need to repeat the buffer set for 6 more times.

that is Bit '0' line, Bit '1' line, Bit '2' line --- Bit '7' line.

Q.13 Define Computer Instruction and list different type of instructions.

A computer instruction is a binary code that specifies a sequence of micro-operations for the computer.

There are three types of instructions:

- i) Memory Reference Instruction →

I	Opcode	Addr.
(000=110)		
- ii) Register Reference Instruction →

0	111	Reg Operation
---	-----	---------------
- iii) I/O Reference Instruction →

I	111	I/O Operation
---	-----	---------------

	Symbol	I = 0	I = 1	Description
Memory Reference	D0 AND	0xxx	8xxx	AND mem-word to AC
	D1 ADD	1xxx	9xxx	ADD mem-word to AC
	D2 LDA	2xxx	Axxx	Load mem-word to AC
	D3 STA	3xxx	Bxxx	Store AC content to mem.
	D4 BUN	4xxx	Cxxx	Branch unconditionally
	D5 BSA	5xxx	Dxxx	Branch & Save Return Addr.
	D6 ISZ	6xxx	Exxx	Increment and Skip if zero
Register Reference	CLA	7800		Clear AC
	CLE	7400		Clear E
	CMA	7200		Complement AC
	CME	7100		Complement E
	CIR	7080		Circulate right AC & E
	CIL	7040		Circulate left AC & E
	INC	7020		Increment AC
	SPA	7010		Skip if AC positive i.e. AC(15)=0
	SNA	7008		Skip if AC negative i.e. AC(15)=1
	SZA	7004		Skip if AC = 0
	SZF	7000		Skip if E = 0
	HLT	7001		Halt computer

INP	F800	Input a character to AC
OUT	F400	Output character from AC
SRI	F200	Skip on input flag
SKO	F100	Skip on output flag
ION	F080	Interrupt ON
IOF	F040	Interrupt OFF

Q.14 Differentiate between hardwired and micro-programmed control unit.

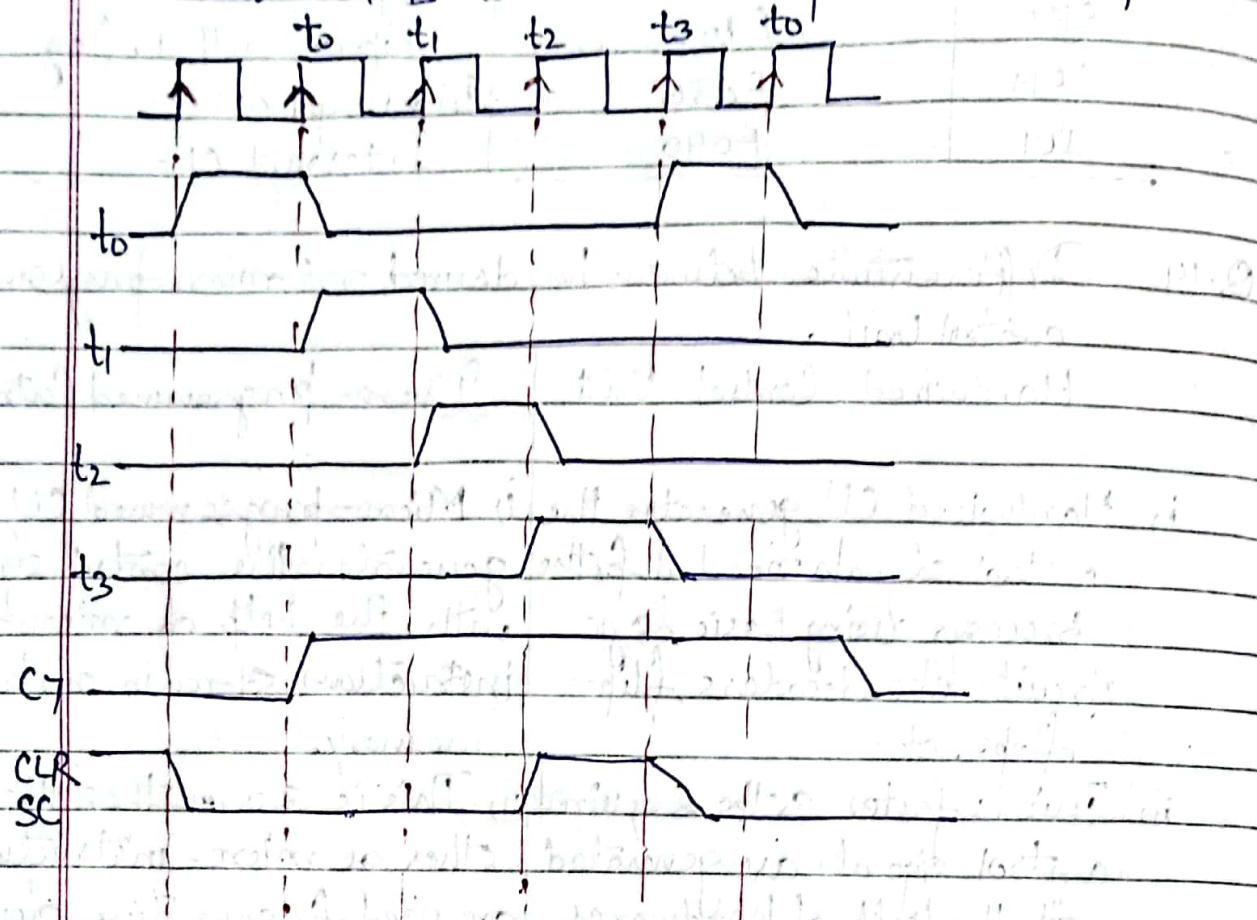
Sol. Hardwired Control Unit Micro-programmed Control Unit

- i) Hardwired CU generates the control signals needed for the processor using basic logic circuits like decoders, flip-flops, etc.
- ii) This is faster as the required control signals are generated with the help of hardware.
- iii) Difficult to modify as the control signals that need to be generated are hard-wired.
- iv) It is costlier as it is realized in terms of logic gates.
- v) Used in Reduced Instruction Set Computer (RISC).
- vi) Only limited no. of instructions are used due to hardware implementation.
- i) Micro-programmed CU generates the control signals with the help of micro-instructions stored in control memory.
- ii) This is slower than the other as micro-instructions are used for generating signals.
- iii) Easy to modify as modification need to be done only at the instruction level.
- iv) It is less expensive as only micro-instructions are required to generate control signals.
- v) Used in Complex Instruction Set Computer (CISC).
- vi) Control signals for many instructions can be generated.

Q.15 Draw the timing diagram for :

$C_7 t_3$; $SC \leftarrow 0$

where C_7 is activated with positive clock pulse.



Initially, the CLR i/p (clear) input of SC is active. The first positive transition of the clock clears SC to 0 which in turns the timing signal T_0 out of the decoder. T_0 is active during one clock cycle. SC is incremented with every positive clock transition unless its CLR i/p is active.

The last three waveforms show how SC is cleared when $C_7 t_3 = 1$.

When timing cycle t_3 is active, the control fn. $C_7 t_3$ becomes active. This signal is applied to the CLR i/p of SC. On next positive clock transition the counter is cleared to '0'.

Long Questions

Q.1 What are micro-operations? What are its various types? Illustrate the implementation of each category of u-op through its block diagram(s).

Sol: The operations executed on data stored in registers are called micro-operations. Examples of micro-operations are shift, count, clear, and load.

The various types of Micro-operations are:

1. Arithmetic u-op.
2. Logical u-op
3. Shift u-op

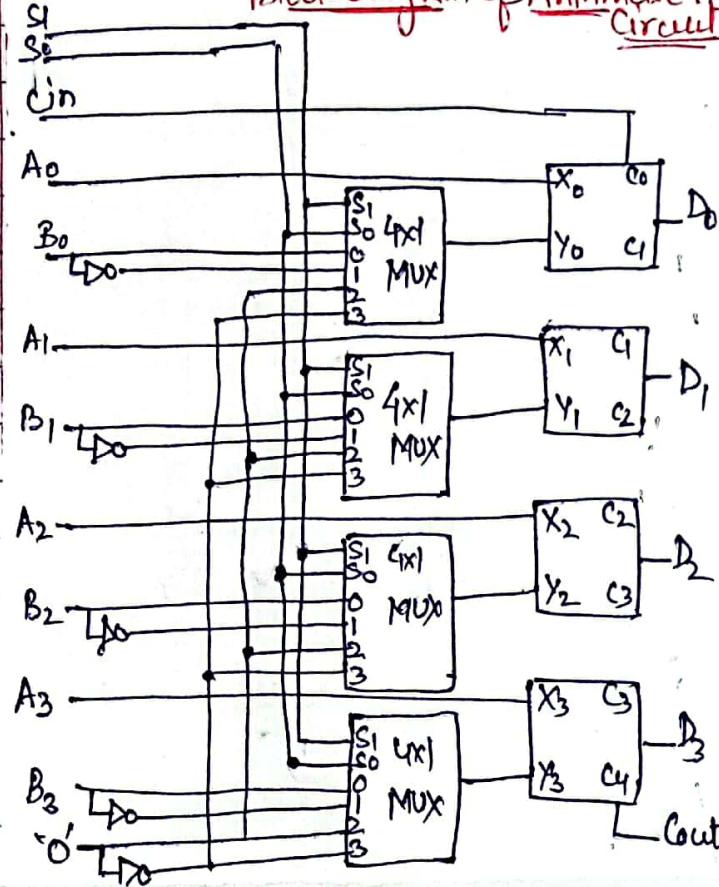
For Hardware Implementation of each category

1. 4-bit Arithmetic Circuit

It has 4 full adders (4-bit Parallel Adder) and four multiplexers for choosing different operations. There are two 4-bit i/p A and B and 4-bit o/p D which is o/p of $X + Y + \text{carry bit}$.

$$D = A + Y + C_{in}$$

Block diagram of Arithmetic Circuit



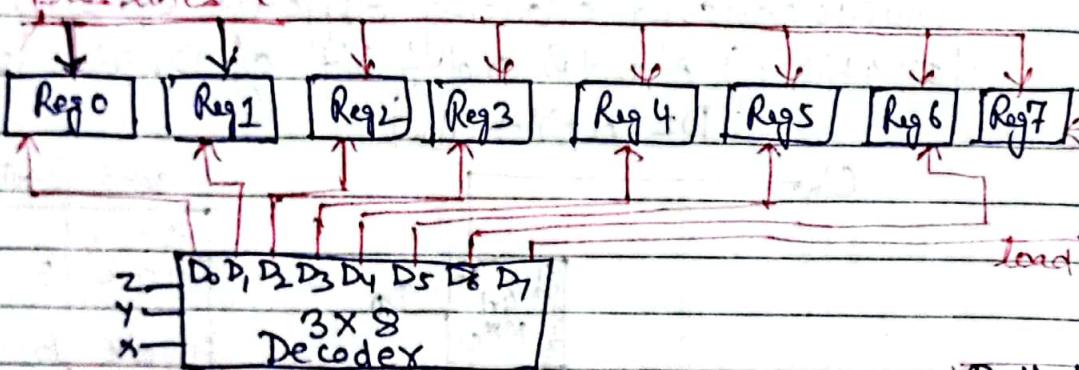
Select	Input	Output & u-op
S_1, S_0, C_{in}	Y	$O/p \Rightarrow D = A + Y + C_{in}$
0 0 0	B	$A + B + C_{in}$ (Addition w/o carry)
0 0 1	B	$A + B + 1$ (Addition with carry)
0 1 0	\bar{B}	$A + \bar{B} + 0$ (Subtraction with borrow)
0 1 1	\bar{B}	$A + \bar{B} + 1$ (Subtraction)
1 0 0	0	$A + 0 + 0$ (Transfer A)
1 0 1	0	$A + 0 + 1$ (Increment)
1 1 0	1	$A + 1 + 0$ (Decrement)
1 1 1	1	$A + 1 + 1$ (Transfer A)

2. for logical H/W implementation
 Refer → Q.11 solution

3. for Shift Left H/W implementation
 Refer → Q.8 solution

Q.2 What is bus? Design a bus system capable of transmitting data from any register from a group of registers (32-bit each) to any other register in a group of 8 registers (32-bit each). Illustrate the logic through its block diagram.

from
 Bus Lines (32-bit Common Bus System)



xyz	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇	Destination Register Select
000	1	0	0	0	0	0	0	0	Reg ₀
001	0	1	0	0	0	0	0	0	Reg ₁
010	0	0	1	0	0	0	0	0	Reg ₂
011	0	0	0	1	0	0	0	0	Reg ₃
100	0	0	0	0	1	0	0	0	Reg ₄
101	0	0	0	0	0	1	0	0	Reg ₅
110	0	0	0	0	0	0	1	0	Reg ₆
111	0	0	0	0	0	0	0	1	Reg ₇

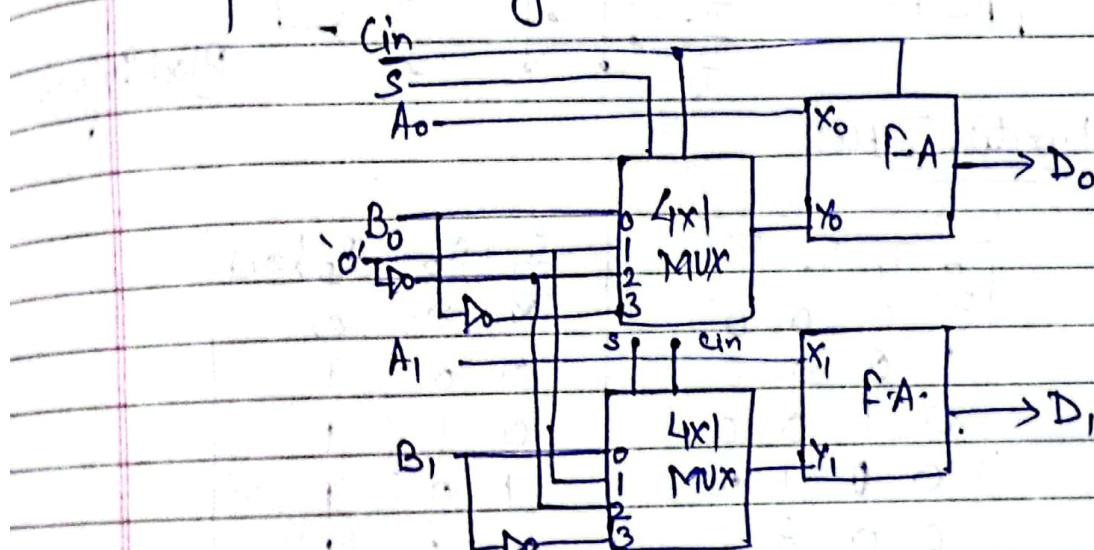
Q.3

S	$Cin = 0$	$Cin = 1$
0	$D = A + B$	$D = A + 1$
1	$D = A - 1$	$D = A + \bar{B} + 1$

S	Cin	Y	Micro-operations
0	0	B	$A + B$ (Addition w/o Carry)
0	1	0	$A + 1$ (Increment)
1	0	1	$A - 1$ (Decrement)
1	1	\bar{B}	$A + \bar{B} + 1$ (Subtraction)

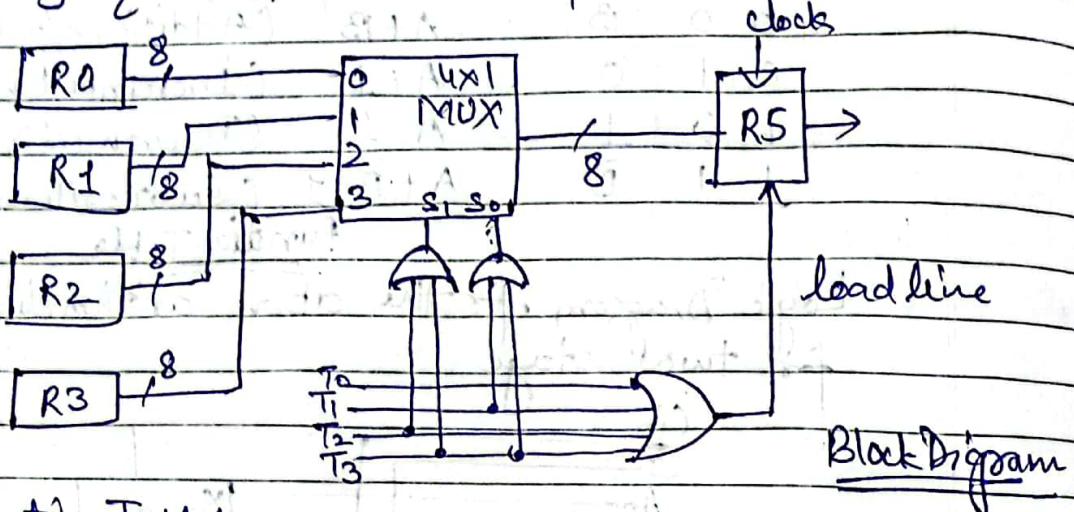
function Table

Logic Diagram for the above arithmetic circuit
for two stage



Here, for choice of 4 different arithmetic operations, we need to add second i/p B of the full adder as $B, 0, 1, \bar{B}$ to first i/p A . To select these 4 choices we will use 4x1 Multiplexer where 2 selected lines will be S and Cin and to depict this above is the function table.

Q4. ~~Sol.~~ R_0, R_1, R_2, R_3 are connected to 4×1 MUX and the o/p of MUX is connected to i/p of R_5 . This transfer from either of the 4 register into R_5 will take place only when loadline of R_5 will get HIGH. Following eg. ① ② ③ are the expressions of select lines S_1, S_0 .



Function Table:

To T_0, T_1, T_2, T_3	S_1, S_0, R_5 load
0 0 0 0	x x 0
1 0 0 0	0 0 1
0 1 0 0	0 1 1
0 0 1 0	1 0 1
0 0 0 1	1 1 1

$$S_0 = T_1 + T_3 \quad \text{--- (1)}$$

$$S_1 = T_2 + T_3 \quad \text{--- (2)}$$

$$R_5 \text{ load} = T_0 + T_1 + T_2 + T_3 \quad \text{--- (3)}$$

The explanation of (1), (2) & (3) is as follows.

S_1 is 1, when either T_2 is high OR T_3 is high.

S_0 is 1, when either T_1 is high OR T_3 is high.

The loadline of R_5 will be high if any of the signals from T_0 to T_3 gets HIGH.

Using the function table, we can draw the above block diagram.

Q.5 What are the various phases of instruction cycle? Give the u-op. of fetch & decode phases. How the first two register transfer statements are implemented.

Soln The four phase of instruction cycle are:

1. Fetch the instruction from the memory;
2. Decode the instruction.
3. Calculate the effective address of the operand (optional)
4. Execute the instruction.

Initially, $SC \leftarrow 0$
 To: $AR \leftarrow PC$
 $T_1: IR \leftarrow M[AR], PC \leftarrow PC + 1$

fetch
Phase
1c-ops.

Decode $[T_2: I \leftarrow IR(15), \text{Decode} \leftarrow IR(12-14), AR \leftarrow IR(0-1)]$

Phase

u-op: If Decode o/p is $(D_0 - D_6)$, it is a memory reference instruction and if $I = 0$, it is direct addressing and if $I = 1$, it is indirect addressing mode.

If Decode o/p is D_7 , it is either register reference instruction or I/O reference instruction. Now, if $I = 0$, it is register reference and if $I = 1$, it is I/O reference.

This is how, decode phase determines the type of the instruction.

Q.6 Tabulate various memory reference instructions?
Explain BUN and BSA.

Soln. There are 7 different memory reference instructions

1	Opcode	Addr..	Opcode = 000 to 110 Do to D6
---	--------	--------	---------------------------------

Opcode	Symbol	I=0	I=1	Hexadecimal code	Description
D0	AND	0 xxx	8 xxx		AND memory word to AC
D1	ADD	1 xxx	9 xxx		Add mem. word to AC
D2	LDA	2 xxx	A xxx		Load mem. word to AC
D3	STA	3 xxx	B xxx		Store content of AC in mem.
D4	BUN	4 xxx	C xxx		Branch unconditionally
D5	BSA	5 xxx	D xxx		Branch & Save Return Address
D6	ISZ	6 xxx	E xxx		Increment & Skip if Zero

BUN : Branch Unconditionally

This instruction transfers the program to the instr. specified by the effective addr. This allows programmer to specify an instruction ~~out of sequence~~.

D₄-Ty : PC \leftarrow AR, SC \leftarrow 0

BSA : Branch and Save Return Address

This instruction is useful for branching to a portion of the program called a subroutine or procedure. When executed, this instruction stores the addr. of next instr. in sequence into mem. loc. specified by effective address. The EA+1 then transferred to PC to serve as the addr. of 1st instr. in subroutine.

D₅-Ty : M[AR] \leftarrow PC, AR \leftarrow AR+1

D₅-Ts : PC \leftarrow AR, SC \leftarrow 0

Q.7

Provide hexa decimal 16-bit instruction code and explain what an instruction going to perform.

- a. 0001 0000 0010 0100
- b. 1011 0001 0010 0100
- c. 0111 1000 0000 0000
- d. 0111 0000 1000 0000
- e. 1111 1000 0000 0000

Soln.

a. $\begin{array}{cccccc} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ D_1 & D_2 & D_3 & D_4 & D_5 & D_6 & D_7 \end{array} \Rightarrow (1024)H$

Addr.

Opcode $\Rightarrow 001$ (D_1) , it is mem. ref. instr. of direct addr. mode

\Rightarrow Add content of mem. loc. $024H$ to AC

b. $\begin{array}{cccccc} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ D_1 & D_2 & D_3 & D_4 & D_5 & D_6 & D_7 \end{array} \Rightarrow (B124)H$

Addr.

Opcode $\Rightarrow 011$ (D_3) , it is mem. ref. instr. of indirect addr.

\Rightarrow Store the content of AC to mem. loc. pointed by mem. loc. $(124)H$

c. $\begin{array}{cccccc} 0 & 1 & 1 & 1 & 1000 & 0000 0000 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ D_1 & D_2 & D_3 & D_4 & D_5 & D_6 \end{array} \Rightarrow (7800)H$

 $B_{11} = 1$

Opcode is D_7 (0111) and $I=0$, it is Register Ref. Instr.

$\Rightarrow B_{11} = 1 \Rightarrow$ CLA i.e., Clear the AC.

d. $\begin{array}{cccccc} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ D_1 & D_2 & D_3 & D_4 & D_5 & D_6 & D_7 \end{array} \Rightarrow (7080)H$

Opcode is D_7 and $I=0$, it is Register Ref. Instr.

$\Rightarrow B_7=1 \Rightarrow$ CIR i.e. Circulate Right the content of AC

e. $\begin{array}{cccccc} 1 & 1 & 1 & 1 & 1000 & 0000 0000 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ D_1 & D_2 & D_3 & D_4 & D_5 & D_6 \end{array} \Rightarrow (F800)H$

Opcode is D_7 and $I=1$, it is I/O Ref. Instr. $\Rightarrow B_{11} = 1 \Rightarrow$ Input a character