Assignment-I [CE and CS]

1. Differentiate between computer organization and computer architecture. Describe different types of ROMs.
2. Design a 5-bit arithmetic circuit that is capable of performing addition, subtraction, increment, decrement, and transfer operations. Show the function table for this arithmetic circuit.
3. Describe the Instruction cycle of Load Accumulator (LDA) instruction with the necessary Register transfer language. Assume direct addressing is used in instruction to load the data.
4. Discuss shift micro-operations with examples. Draw the block diagram of the parity generator and parity checker for 3-bit data using odd parity.
5. Design an arithmetic circuit with one selection variable S and two n–bit data inputs A and B. The circuit generates the following four arithmetic operations in conjunction with the input carry C_{in}. Draw the logic diagram of the first 2 MSBs.

S	**C_{in} = 0**	**C_{in} = 1**
1	D= A – 1 (Decrement)	D = A+ B' + 1 (Subtract)
0	D = A+B (Add)	D = A+ 1 (Increment)

6. What are the advantages of signed 2's complement representation over signed magnitude representation and signed 1's complement representation? Discuss 2421 code as self-complementary code

7. Differentiate between von Neumann and Harvard architecture. Describe how read and write operations are performed in Random Access Memory with a suitable block diagram.
8. What is the drawback of BCD? How it can be overcome by certain weighted code, describe it with example. Perform the arithmetic operation $(-82)_{10} + (-5)_{10}$ using signed-2's complement representation.

9. A 6-bit register contains the binary value 110010. What is the register value after the arithmetic shift right?
10. Draw the block diagram and describe the operation for the hardware that implements the following statements:

$$x + yz: \quad AR \leftarrow AR + BR$$

11. Where AR and BR are two n-bit registers and x, y, and z are control variables. Include the logic gates for the control function.
12. Describe two different methods to identify overflow conditions while performing an addition operation with a suitable example. Represent the number 8621 in a) Excess 3-code b) 2421 code.
13. Design a 5-bit adder-subtractor circuit. Describe four different applications of logical micro-operations
14. Give two different reasons for using Addressing Modes by computers. Explain three major types of interrupts that cause a break in the regular execution of a program.
15. Explain the Instruction cycle of a basic computer with a suitable flow chart without considering the interrupt cycle.

16. What is the difference between a hardwired control unit and a microprogrammed control unit?
17. Discuss the purpose of floating point representation. In what way it is different than fixed point representation?
18. Discuss selective set, selective complement, insert, and mask micro-operations as logical instructions.
19. Discuss the working mechanism of BSA instruction concerning fetch, decode, and execute cycle. In what way it is different than interrupt?
20. With necessary figures and RTL, discuss the Fetch, Decode, and Execute cycle of ADD and ISZ instruction.
21. For the given questions use the Control functions and Micro-operations for the Basic Computer given on the next page.
    - Derive the control gates for the write input of the memory in the basic computer.
    - Show the complete logic of the interrupt flip flop 'R' in the basic computer. Use a JK flip-flop and minimize the number of gates.
22. Concerning the **Timing and Control Unit** shown in Chapter 5(Central Processing Unit), discuss the fetch and decode phase of 1 AND 235 instruction.

Fetch	$R'T_0$:	$AR \leftarrow PC$
	$R'T_1$:	$IR \leftarrow M[AR]$, $PC \leftarrow PC + 1$
Decode	$R'T_2$:	$D_0, \ldots, D_7 \leftarrow$ Decode $IR(12\text{–}14)$, $AR \leftarrow IR(0\text{–}11)$, $I \leftarrow IR(15)$
Indirect	$D_7'IT_3$:	$AR \leftarrow M[AR]$
Interrupt:		
$T_0'T_1'T_2'(IEN)(FGI + FGO)$:		$R \leftarrow 1$
	RT_0:	$AR \leftarrow 0$, $TR \leftarrow PC$
	RT_1:	$M[AR] \leftarrow TR$, $PC \leftarrow 0$
	RT_2:	$PC \leftarrow PC + 1$, $IEN \leftarrow 0$, $R \leftarrow 0$, $SC \leftarrow 0$
Memory-reference:		
AND	D_0T_4:	$DR \leftarrow M[AR]$
	D_0T_5:	$AC \leftarrow AC \wedge DR$, $SC \leftarrow 0$
ADD	D_1T_4:	$DR \leftarrow M[AR]$
	D_1T_5:	$AC \leftarrow AC + DR$, $E \leftarrow C_{out}$, $SC \leftarrow 0$
LDA	D_2T_4:	$DR \leftarrow M[AR]$
	D_2T_5:	$AC \leftarrow DR$, $SC \leftarrow 0$
STA	D_3T_4:	$M[AR] \leftarrow AC$, $SC \leftarrow 0$
BUN	D_4T_4:	$PC \leftarrow AR$, $SC \leftarrow 0$
BSA	D_5T_4:	$M[AR] \leftarrow PC$, $AR \leftarrow AR + 1$
	D_5T_5:	$PC \leftarrow AR$, $SC \leftarrow 0$
ISZ	D_6T_4:	$DR \leftarrow M[AR]$
	D_6T_5:	$DR \leftarrow DR + 1$
	D_6T_6:	$M[AR] \leftarrow DR$, if $(DR = 0)$ then $(PC \leftarrow PC + 1)$, $SC \leftarrow 0$
Register-reference:		
	$D_7I'T_3 = r$ (common to all register-reference instructions)	
	$IR(i) = B_i$ $(i = 0, 1, 2, \ldots, 11)$	
	r:	$SC \leftarrow 0$
CLA	rB_{11}:	$AC \leftarrow 0$
CLE	rB_{10}:	$E \leftarrow 0$
CMA	rB_9:	$AC \leftarrow \overline{AC}$
CME	rB_8:	$E \leftarrow \overline{E}$
CIR	rB_7:	$AC \leftarrow$ shr AC, $AC(15) \leftarrow E$, $E \leftarrow AC(0)$
CIL	rB_6:	$AC \leftarrow$ shl AC, $AC(0) \leftarrow E$, $E \leftarrow AC(15)$
INC	rB_5:	$AC \leftarrow AC + 1$
SPA	rB_4:	If $(AC(15) = 0)$ then $(PC \leftarrow PC + 1)$
SNA	rB_3:	If $(AC(15) = 1)$ then $(PC \leftarrow PC + 1)$
SZA	rB_2:	If $(AC = 0)$ then $PC \leftarrow PC + 1$
SZE	rB_1:	If $(E = 0)$ then $(PC \leftarrow PC + 1)$
HLT	rB_0:	$S \leftarrow 0$
Input-output:		
	$D_7IT_3 = p$ (common to all input–output instructions)	
	$IR(i) = B_i$ $(i = 6, 7, 8, 9, 10, 11)$	
	p:	$SC \leftarrow 0$
INP	pB_{11}:	$AC(0\text{–}7) \leftarrow INPR$, $FGI \leftarrow 0$
OUT	pB_{10}:	$OUTR \leftarrow AC(0\text{–}7)$, $FGO \leftarrow 0$
SKI	pB_9:	If $(FGI = 1)$ then $(PC \leftarrow PC + 1)$
SKO	pB_8:	If $(FGO = 1)$ then $(PC \leftarrow PC + 1)$
ION	pB_7:	$IEN \leftarrow 1$
IOF	pB_6:	$IEN \leftarrow 0$