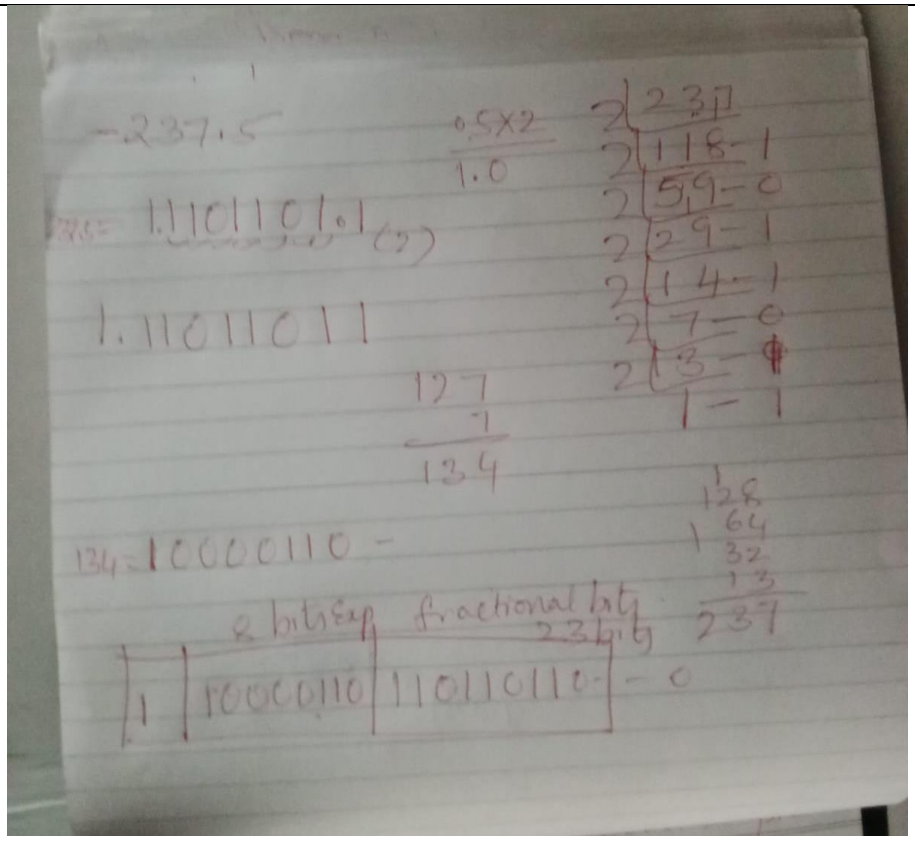
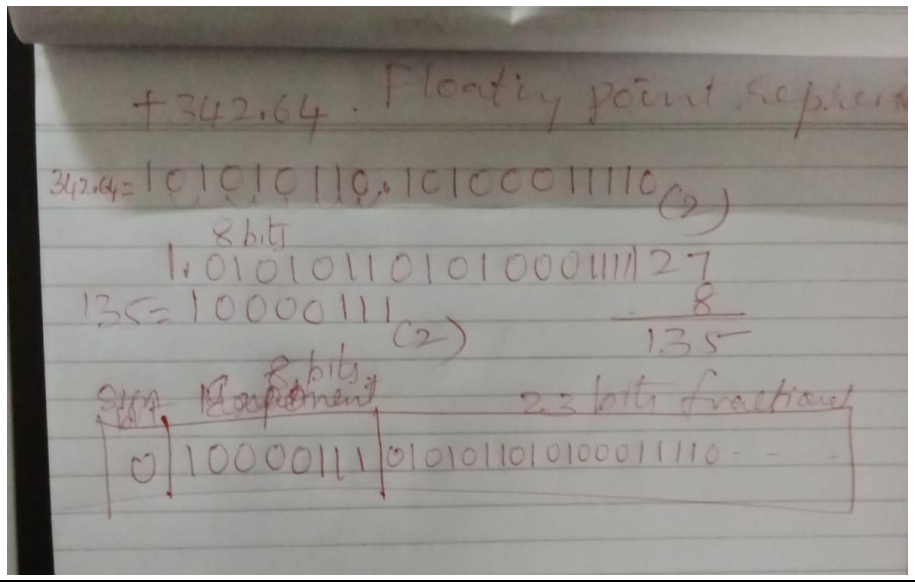
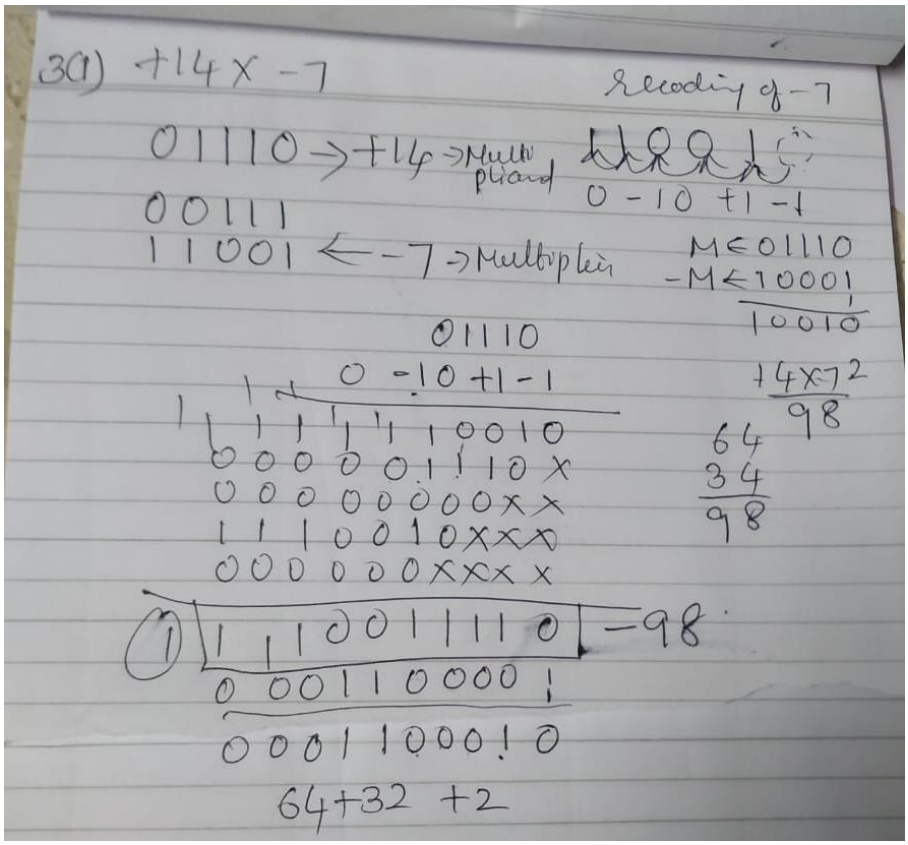
	R V College of Engineering **Department of Computer Science and Engineering** **CIE - I : (Test1) Question Paper**		
Course: (Code)	**APPLIED DIGITAL LOGIC DESIGN AND COMPUTER ORGANIZATION (CS234AI)**	**Semester :**	**3rd BE**
Date : 9th Jan 2024	**Duration :** 90 minutes	**Staff :**	**KB/PSB/MH/DD**
Name :	**USN** :	**Section :**	**A/B/C/D/CD/CY**

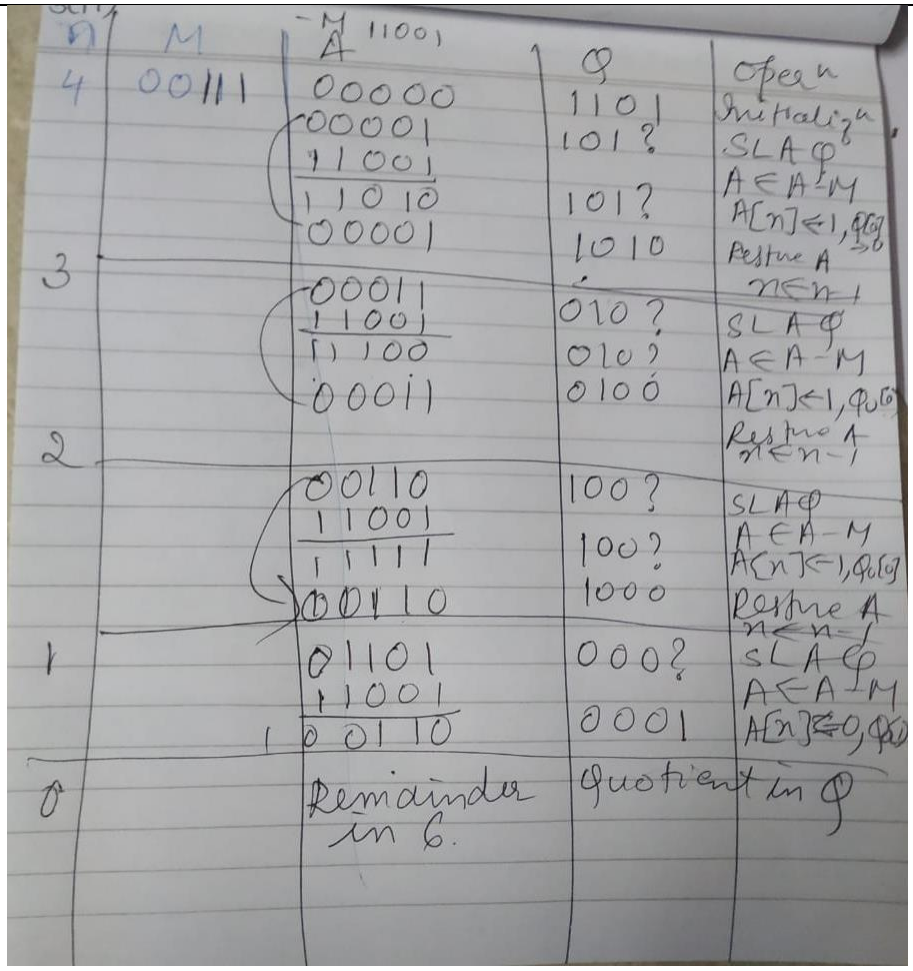
Sl. no	**Answer all the questions**	**Marks**	**L1-L6**	**CO**
1	i) Simplify the Boolean function using QM Method. F (P, Q, R, S) = ∑ m (0, 1, 2, 4, 5, 7, 8, 9,14,15) 2 Quads and 3 Pairs. P'R'+Q'R'+P'Q'S'+PQR+QRS or P'QS ii) Simplify the following Boolean expressions using K maps. F (w,x,y,z) = ∑ m (0,1,4,5,8,9,14,15) + dc(11,12) 2 Quads and 1 Pair. W'Y'+X'Y'+ WXY	10	L2	CO1
2	Represent the following numbers, as indicated below. i) 40 and -32 in 8 bit signed magnitude representation 40=00101000 -32=10100000 ii) 9 and 18 in BCD representation 9 - 1001 18 - 0001 1000 iii) 0 and 4 in Excess 3 representation 0 - 0011 4 - 0111 iv) -237.5 in 32 bit (single precision) floating point number representation	10	L2	CO1



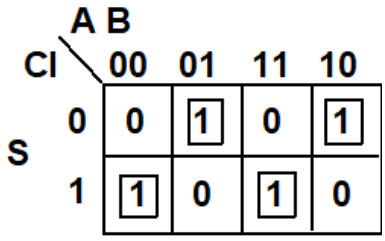
v) +342.64 in 32 bit (single precision floating point representation)

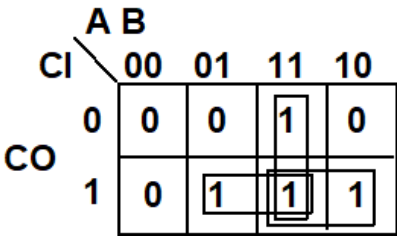


3	i) Using booths algorithm, multiply the numbers: +14 x -7  ii) Using Restoring Division Algorithm, divide: +13 / +7	10	L3	CO1

	 M stores the divisor, Q stores the dividend			
4	Design 1bit Parallel Adder (write Truth Table, KMap Simplification and final expressions) and using 1bit Parallel Adders design a 4bit parallel(ripple) adder/subtractor circuit, with suitable control input to select addition/subtraction. Describe the working of the circuit. Using the above design, Write the answer for the following cases of addition /subtraction. a. X=1001 Y=1101 ADD/SUB = 0 b. X=0011, Y=1010 ADD/SUB = 1 Justify with suitable example, the above circuit works for signed 4 bit numbers, clearly indicating the meaning of Overflow.	10	L3	CO2

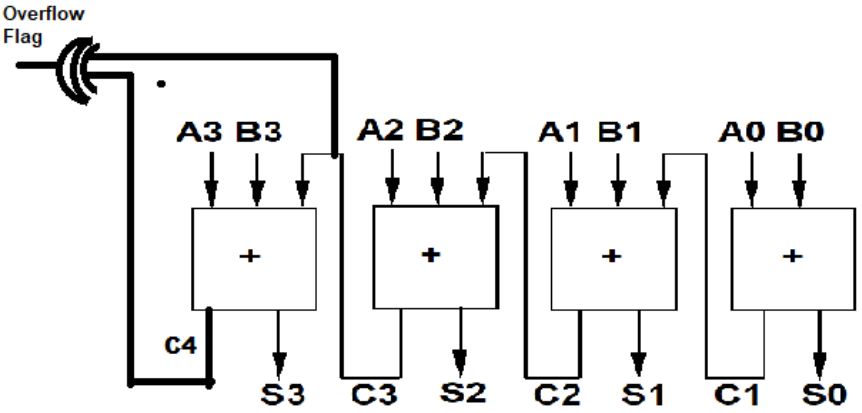
A	B	Cl	S	CO
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

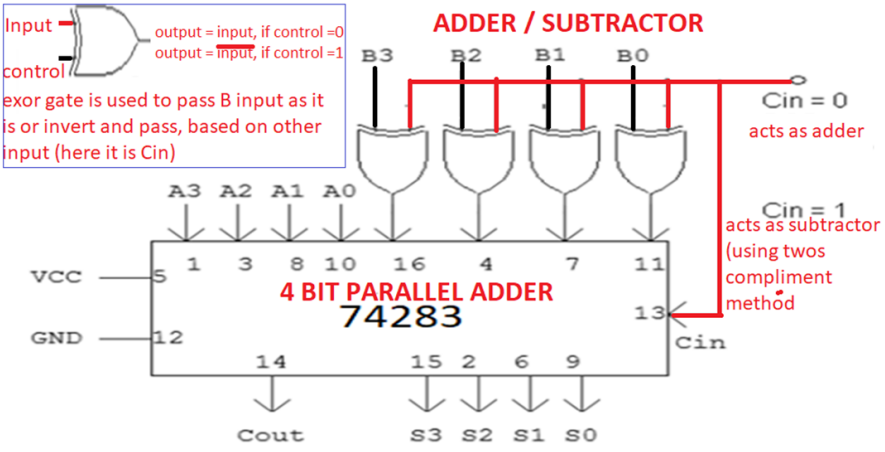




S = Cl xor A xor B

CO = B Cl + A Cl + A B = Cl (A + B) + A B





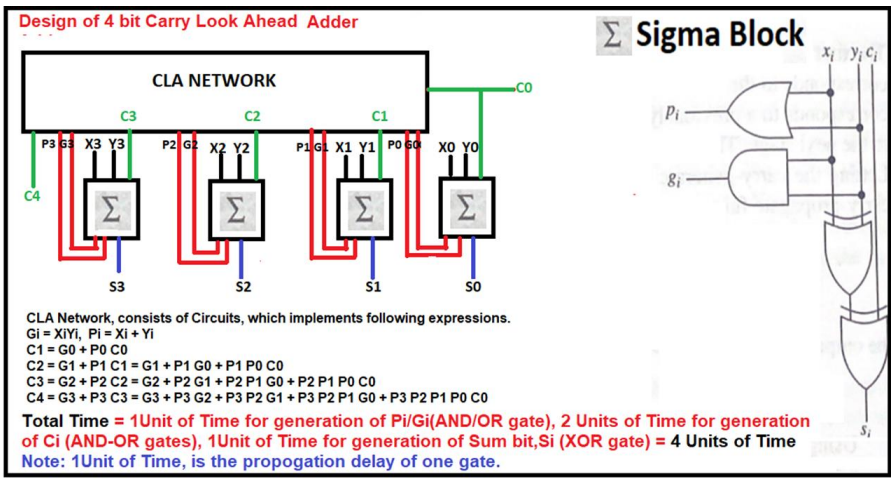
X=1001 Y=1101 ADD/SUB = 0

Ans: cy =1 sum = 0110

X=0011, Y=1010 ADD/SUB = 1

Ans: cy=0, sum = 1001

	Signed Addition/Subtraction: The same circuit is used for signed addition / subtraction. Example: Let us add two signed numbers (4bit) +5 and -3, +5 -> 0101 -3 -> 1101 Ans-> 0010, Overflow Flag->0 Overflow flag can be realized/designed using one ExOr gate. Overflow is the ExOr of the carry coming from last bit and previous to last bit (i.e ExOr of carry generated from D2 and D3 bits). Overflow flag indicates, the answer generated is exceeded the capacity of holding/range permitted. (For 4 bit adders, answer should be in the range -> -8 to +7).			
5	Discuss the merits of CLA adder over normal ripple adders. Derive the Carry equations, C1 to C4 for 4-bit CLA adder. Draw the circuit diagram, clearly indicating the circuits and all the connections. Suggest time calculations, for designing 16 bit adder using Ripple adder, CLA adder, cascading CLA adders and any other methods Total Time Required for N bit adder using Ripple parallel adder = N * 2, Takes too long - need to investigate FASTER adders!. The Carry Look Ahead Adders, CLA adders, overcomes the disadvantage of taking more units of time. The principle behind design of CLA adders, Carry input at each stage is direct function of operand bits. Thus the idea of having SUM and CARRY outputs of the ith stage be a function of Xi,Yi and Ci is replaced by having these outputs be a function of Xi,Yi and C0. Let us write S and Cout expressions for full adder, in terms of inputs X and Y and Cin, for 'i' th bit **Si = Xi .xor. Yi .xor. Ci** **Ci+1 = Xi Yi + Xi Ci + Yi Ci** **= Xi Yi + Ci (Xi + Yi)** **= Xi Yi + Ci (Xi .or. Yi)** **Let us represent, XiYi by Gi (also called as Carry generate function) &** **Xi.or.Yi by Pi (also called carry propagate function)** **= Gi + Ci Pi** **Now, lets write the these expressions, for C1,C2,C3 and C4** **C1 = G0 + P0 C0** **C2 = G1 + P1 C1 = G1 + P1 G0 + P1 P0 C0** **C3 = G2 + P2 C2 = G2 + P2 G1 + P2 P1 G0 + P2 P1 P0 C0** **C4 = G3 + P3 C3 = G3 + P3 G2 + P3 P2 G1 + P3 P2 P1 G0 + P3 P2 P1 P0 C**	10	L3/L4	CO2



Let us explore time requirements for 16 bit adder:

Using Ripple Adder - 16 x 2 = 32 Units of Time (Gate Delays)

Using CLA principle - 4 Units of Time (but circuit will be very complex, increase number of gates and its inputs.

Using Cascading of 4 bit CLA adders (4 numbers): 2 + (n/2), when n is the number of bits = 2 + 16/2 = 10 Units of Time

Course Outcomes: After completing the course, the students will be able to:-	
CO 1	Apply design requirements for digital systems and Computer organization
CO 2	Analyze the models used for designing various Ccombinational and Sequential circuits
CO 3	Develop applications of synchronous sequential networks using flip flops, registers and counters
CO 4	Design optimized modern processors and memories for given specifications
CO 5	Investigate techniques of digital system design for building industry relevant real-world systems using electronic components and modern tools

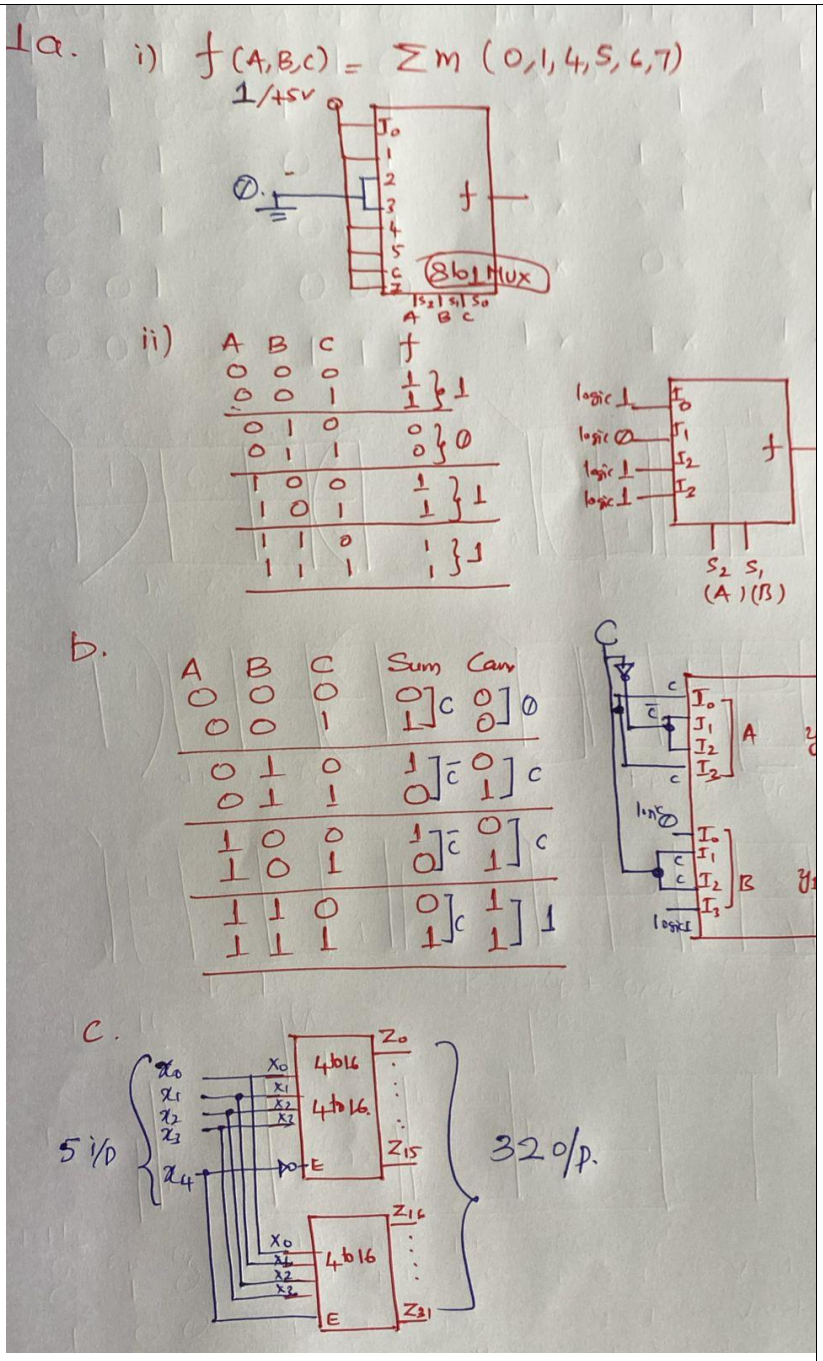
Course Outcomes:

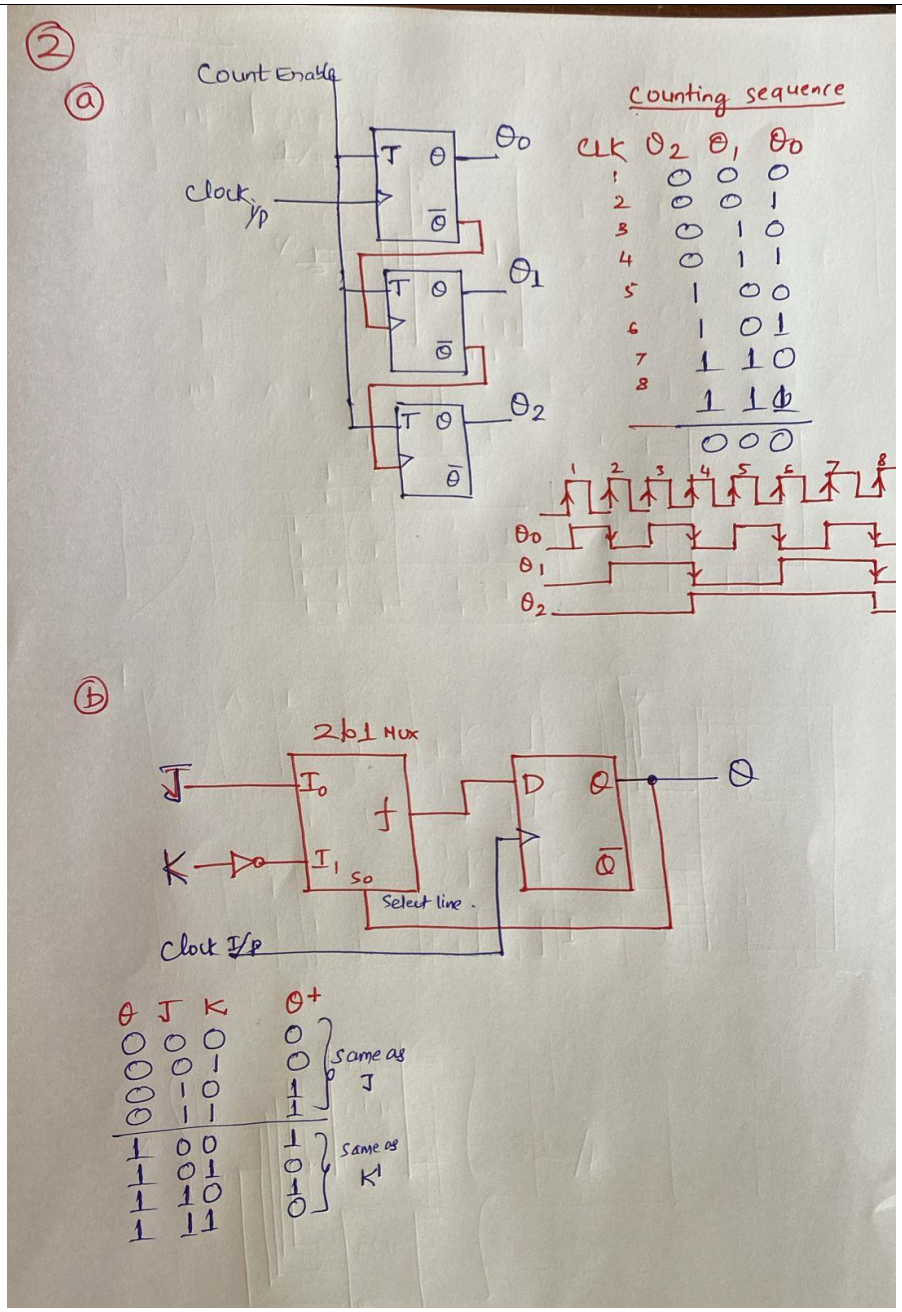
Marks Distribution	Particulars		CO1	CO2	CO 3	CO4	CO5	L1	L2	L3	L 4	L 5	L6
	Test	Max Marks	30	20					20	25	5		

	R V College of Engineering **Department of Computer Science and Engineering** **CIE - II : (Test2) Question Paper**	
Course: (Code)	**APPLIED DIGITAL LOGIC DESIGN AND COMPUTER ORGANIZATION (CS234AI)**	**Semester : 3rd BE**

Date : 22nd Feb 2024	**Duration :** 90 minutes	**Staff :**	**KB/PSB/MH/DD**
Name :	**USN** :	**Section :**	**A/B/C/D/CD/CY**

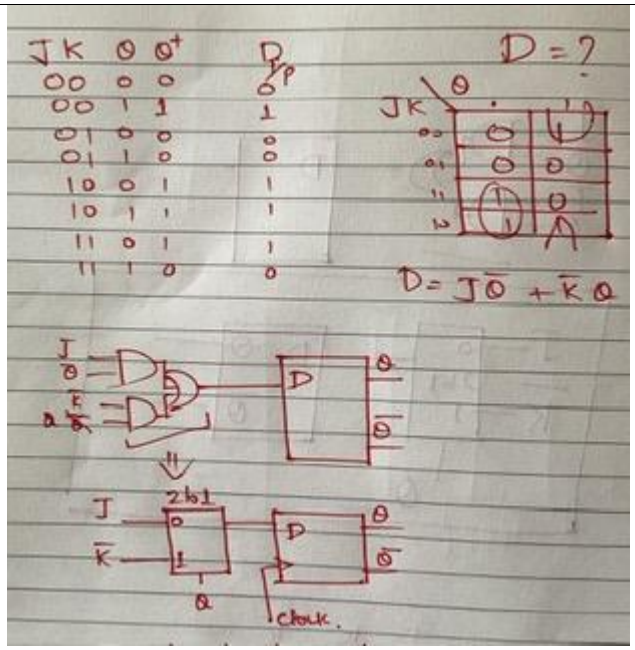
Sl. no	**Answer all the questions**	**Marks**	**L1-L6**	**CO**
1	a. Implement f (A, B, C) = ∑ m (0, 1, 4, 5, 6, 7) using (i) 8:1 MUX with a, b, c as select lines. (ii) 4:1 MUX with a, b as select lines. b. Design a digital circuit for full adder using Dual 4:1 Multiplexer IC (Ex.74153)with necessary gates. c. Construct 5-to-32-line decoder from two 4 to 16-line decoder.	4+ 4+ 2	L3	CO2

				
2	a. Design 3 bit Asynchronous Up Counter, using positive edge triggered T flip-flops b. Design JK flip flop using a D flip-flop, a 2 to 1 mux and an inverter if required and show all the steps.	4+ 6	L4	CO2

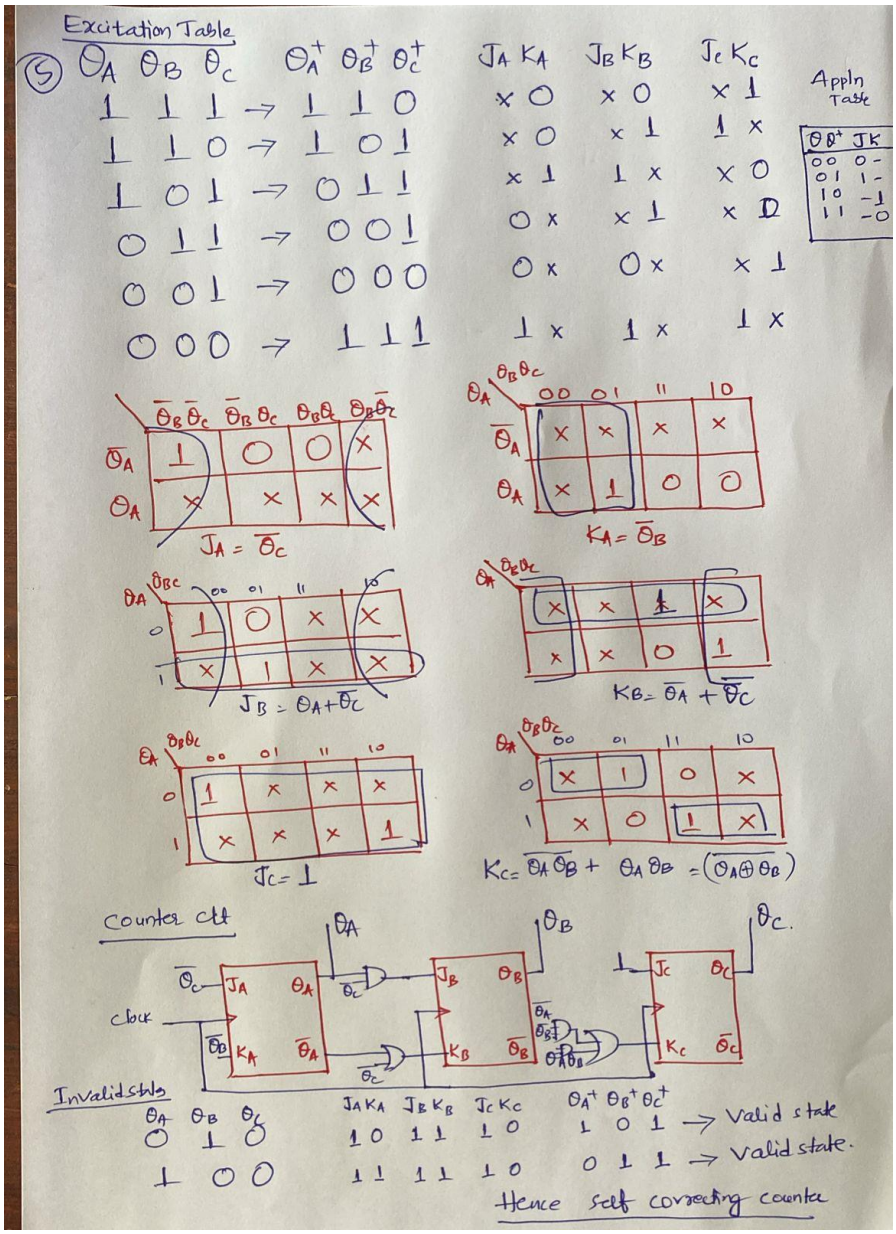
	② ⓐ 			

ⓑ

Q	J	K	Q+	
0	0	0	0	same as J
0	0	1	0	
0	1	0	1	
0	1	1	1	
1	0	0	1	same as $\bar{K}$
1	0	1	0	
1	1	0	1	
1	1	1	0	

<table>
<tr><td></td><td> 
</td><td></td><td></td><td></td></tr>
<tr><td>3</td><td>a. Design a Master Slave JK Latch/FlipFlop using only Nand gates. Show the timing diagram and function table.
b. Explain with proper timing diagrams 0’s and 1’s catching
Refer Notes</td><td>6+
4</td><td>L3</td><td>CO3</td></tr>
<tr><td>4</td><td>Design a 4 bit Universal Shift Register, Which supports following modes of operation-

<table><tr><td>S1</td><td>S0</td><td>Operations</td></tr><tr><td>0</td><td>0</td><td>Hold</td></tr><tr><td>0</td><td>1</td><td>Shift Right Data</td></tr><tr><td>1</td><td>0</td><td>Circular Shift Left Data</td></tr><tr><td>1</td><td>1</td><td>Parallel Load the Data</td></tr></table>
Justify how the shift registers can be used as counters (Ring and Johnson). (4 M)
ring counter-n flipflops , n counts
johnson counter-n flipflops, 2n counts</td><td>6+
4</td><td>L3</td><td>CO3</td></tr>
<tr><td>5</td><td>Design Synchronous Counter using JK Positive Edge Triggered Flip Flops, which produces the following sequence: 111,110,101,011,001, 000.
Verify the above design as a self correcting counter or not.</td><td>8+
2</td><td>L4</td><td>CO3</td></tr>
</table>

Course Outcomes: After completing the course, the students will be able to:-	
CO 1	Apply design requirements for digital systems and Computer organization
CO 2	Analyze the models used for designing various Combinational and Sequential circuits
CO 3	Develop applications of synchronous sequential networks using flip flops, registers and counters
CO 4	Design optimized modern processors and memories for given specifications
CO 5	Investigate techniques of digital system design for building industry relevant real-world systems using electronic components and modern tools

Course Outcomes:

Marks Distribution	Particulars		CO1	CO2	CO 3	CO4	CO5	L1	L2	L3	L 4	L 5	L6
	Test	Max Marks		20	30					40	10		

	R V College of Engineering **Department of Computer Science and Engineering** **Improvement Test - Question Paper**		
Course: (Code)	**APPLIED DIGITAL LOGIC DESIGN AND COMPUTER ORGANIZATION (CS234AI)**	**Semester :**	**3rd BE**
Date : 20nd Mar 2024	**Duration :** 90 minutes	**Staff :**	**KB/PSB/MH/DD**
Name :	**USN** :	**Section :**	**A/B/C/D/CD/CY**

Sl. no	**Answer all the questions**	**M**	**L1-L6**	**CO**
1	A. Define Addressing Modes and List all the Generic Addressing Modes supported by the processor instruction set, with an example instruction.	6	L2	CO1
	B. Write Assembly Language Code snippet, to add N numbers, stored in the memory. Store the answer in the memory. Draw the memory layout and clearly indicate with comments, all the assembler directives used in the program.	4		

Name	Example	Assembler syntax	Addressing function
Immediate	EXAMPLES MOV #20,R0 move the value 20, to R0	#Value	Operand=Value
Register	MOV R1,R0 move the contents of R1 to R0	Ri	EA=Ri
Absolute (Direct)	MOV SUM,R2 add the number stored in memory location "SUM" TO R2 and store the answer in R2	LOC	EA=LOC
Indirect	ADD (R2),R0 add the number stored in memory location, whose address is stored in R2, with R0 and store ans in R0. R2 acts as pointer	(Ri)	EA=[Ri]
		(LOC)	EA=[LOC]
Index	MOV 4(R2),R0 move the contents of memory location, whose address is computed by addng 4 and contents of R2, to the register R0	X(Ri)	EA=[Ri]+X
Base with index	ADD (R0,R1),R2 add the contents of memory location, whose address is computed by adding the contents of R0 and R1, with R2 and store the answer in R2	(Ri, Rj)	EA=[Ri]+[Rj]
Base with index and offset		X(Ri, Rj) ADD 10(R0,R1),R2	EA=[Ri]+[Rj]+X
Relative	BNE LOOP (BRANCH NOT EQUAL) program jumps/branches to the location, LOOP, whose address is computed by adding the contents of PC with the offset value i.e distance of LOOP relative to PC	X(PC)	EA=[PC]+X
Autoincrement	ADD (R2)+,R0 same as Indirect, but here pointer is incremented, after the operation	(Ri)+	EA=[Ri]; Increment Ri
Autodecrement	ADD -(R2),R0 same as Indirect, but here pointer is decremented, before the operation	-(Ri)	Decrement Ri; EA=[Ri]

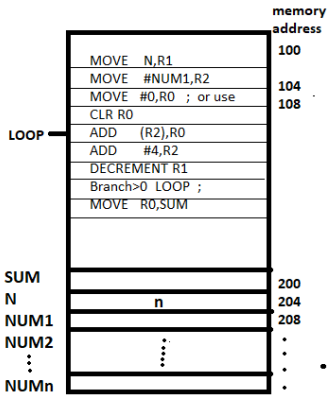
```
ORIGIN 100

MOVE   N,R1
MOVE   #NUM1,R2
MOVE   #0,R0  ;  or use CLR R0
ADD    (R2),R0
ADD    #4,R2
DECREMENT R1
Branch>0  LOOP  ; or use  BGT
LOOP
MOVE   R0,SUM

ORIGIN 200
SUM:  RESERVE   4
   N:  DATAWORD 150
NUM1: RESERVE 600
END
```

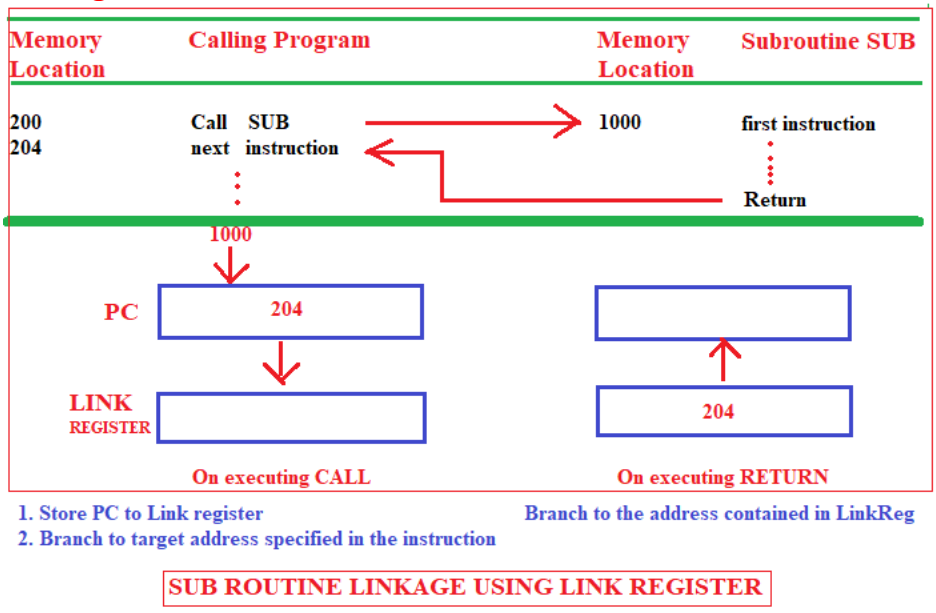
Asembler Directives Used: ORIGIN,RESERVE,DATAWORD,END

Note: Not required to write program in the memory layout



Note: write comments to the program and directives

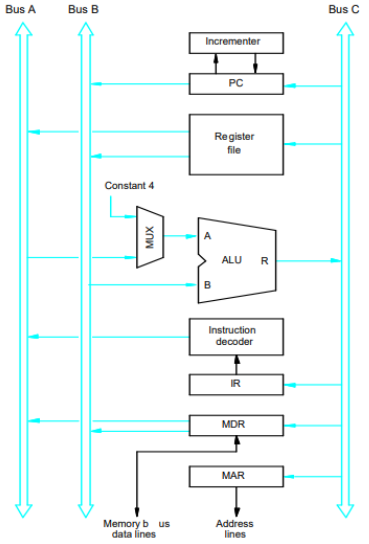
2	A. Write Assembly Language Code snippets using Three, Two and One address instructions. Assume the processor supports ADD, SUM, MULT, MOV, LOAD/STORE instructions:
	$Y= X^3 + 5X - 45$
	3 Address Instructions
	MOV X,Y
	MULT Y,Y,Y
	MULT X,Y,Y
	MULT #5,X,X
	ADD X,Y,Y
	SUB Y,#45,Y ; assumed opnd1-opnd2= opnd3, (it can be other way also)
	
	2 Address Instructions
		6
		
		4
			L3	CO5

	MOV X,Y
	MULT Y,Y
	MULT X,Y
	MULT #5,X
	ADD X,Y
	SUB #45,Y
	
	One Address Instruction
	LOAD X
	MULT X
	MULT X
	ADD X
	ADD X
	ADD X
	ADD X
	ADD X
	SUB #45
	Note: write comments to the program
	
	B. Demonstrate with code snippet, Machine Language/Assembly Language support to "Subroutine Linkage".
	 
	
	SUB ROUTINE LINKAGE USING LINK REGISTER
	Note: Briefly explain the diagram			

3	Discuss the different Read Only Memory Types and their features.	5 5	L3	CO4

	▪ Read-Only Memory:
	  ▪ Data are written into a ROM when it is manufactured.
	▪ Programmable Read-Only Memory (PROM):
	  ▪ Allow the data to be loaded by a user.
	  ▪ Process of inserting the data is irreversible.
	  ▪ Storing information specific to a user in a ROM is expensive.
	  ▪ Providing programming capability to a user may be better.
	▪ Erasable Programmable Read-Only Memory (EPROM):
	  ▪ Stored data to be erased and new data to be loaded.
	  ▪ Flexibility, useful during the development phase of digital systems.
	  ▪ Erasable, reprogrammable ROM.
	  ▪ Erasure requires exposing the ROM to UV light.
	▪ Electrically Erasable Programmable Read-Only Memory (EEPROM):
	  ▪ To erase the contents of EPROMs, they have to be exposed to ultraviolet light.
	  ▪ Physically removed from the circuit.
	  ▪ EEPROMs the contents can be stored and erased electrically.
	▪ Flash memory:
	  ▪ Has similar approach to EEPROM.
	  ▪ Read the contents of a single cell, but write the contents of an entire block of cells.
	  ▪ Flash devices have greater density.
	  Higher capacity and low storage cost per bit.
	  ▪ Power consumption of flash memory is very low, making it attractive for use in equipment that is battery-driven.
	  ▪ Single flash chips are not sufficiently large, so larger memory modules are implemented using flash cards and flash drives.
	
	Design 512x16 RAM Chip, using 256x8 RAM Chips. Show the Steps.
	Size of each chip = 256 x 8
	Total chips = (512x16) / (256 x 8) = 4 chips
	No. of Rows = 512/256 = 2 No. Of Cols = 2
	Decoder size = 1 to 2
	No. address bits for full size (512 x 16) = 9
	No. address bits for each chip (256 x 8) = 8
	
	Note: Draw the diagram			

4	A. Program contains 200 instructions, out of that 40% of instructions requires 1 clock cycles and remaining requires 2 clock cycles for execution. Find the total time required to execute the program running on a 1 MHz machine. T = N x S / R	5 5	L4	CO4

	(80 x 1 + 120 x 2) / 1 M = 320 Micro seconds Note: explain the terms of the equation B. Draw the block diagram of the Processor with Three Bus Datapath, highlighting the advantages over the Single Bus Processor Design. **Multiple-Bus Organization (3 Bus)**  • Allow the contents of two different registers to be accessed simultaneously and have their contents placed on buses A and B. • Allow the data on bus C to be loaded into a third register during the same clock cycle. • Incrementer unit. • ALU simply passes one of ts two input operands unmodified to bus C → control signal: R=A or R=B Minimum or Less no. of clock cycles are required to complete the operation. Ex: ADD R1,R2,R7 can be completed in the single clock cycle. ALU receives the data from two buses, and the result is stored in to the required register in the same clock cycle. No temporary registers like Y and Z required.			
5	Write the control sequence for Single-Bus and Three-Bus Processor Datapath for execution of the following instruction: ADD R0, R1, R7. Instruction is stored in the memory. Give the comments for every sequence. Control Sequence for Single Bus Instruction. This instruction adds the contents of memory location pointed by R3 to R1, and store the answer in R1, Executing this instruction requires the following actions, 1. Fetch the instruction, 2. Fetch the first operand (the contents of the memory location pointed to by R3), 3. Perform the addition, 4. Load the result into R1. Control sequence is indicated below	5 5	L3	CO4

	1. PCout,MARin,Read,Select4,Add,Zin (place PC contents on MAR, start Read operation, use ALU to ADD 4 to PC contents) 2. Zout,PCin,Yin,WMFC (store the new value to PC, i.e old +4, wait for Instruction arrival from memory) 3. MDRout,IRin (instruction read from memory is moved to IR) 4. R3out,MARin,Read (move R3, which contains adds, to MAR, initiate memory Read operation) 5. R1out,Yin,WMFC (wait for the data to come from memory, at the same time, move other operand in R1 to Y, as bus free) 6. MDRout,SelectY,Add,Zin (move memory data from MDR and add to Y, and store in Z) 7. Zout,R1in,End (move the answer from Z to R1) **Wrie control sequence for Add R4,R5,R6 for 3busorganization** Step Action 1 PC_{out}, R=B, MAR_{in}, Read, IncPC 2 WMF C 3 MDR_{outB}, R=B, IR_{in} 4 $R4_{outA}$, $R5_{outB}$, SelectA, Add, $R6_{in}$, End			

Course Outcomes: After completing the course, the students will be able to:-	
CO 1	Apply design requirements for digital systems and Computer organization
CO 2	Analyze the models used for designing various Combinational and Sequential circuits
CO 3	Develop applications of synchronous sequential networks using flip flops, registers and counters
CO 4	Design optimized modern processors and memories for given specifications
CO 5	Investigate techniques of digital system design for building industry relevant real-world systems using electronic components and modern tools

Course Outcomes:

Marks Distribution	Particulars		CO1	CO2	CO3	CO4	CO5	L1	L2	L3	L4	L5	L6
	Test	Marks	10			30	10		10	35	5		

	R V College of Engineering Department of Computer Science and Engineering CIE - I : (ReTest) Question Paper		
Course: (Code)	APPLIED DIGITAL LOGIC DESIGN AND COMPUTER ORGANIZATION (CS234AI)		Semester : 3rd BE
Date : 27th March 2024	Duration : 90 minutes	Staff :	KB/PSB/MH/DD
Name :	USN :	Section :	A/B/C/D/CD/CY

Sl. no	Answer all the questions	Marks	L1-L6	CO
1	Design a digital circuit for full adder and full subtractor using 4:1 Multiplexers. Draw the truth table and demonstrate the circuit with all the connections.	10	L3	CO2
2	i) Using booths algorithm, multiply the numbers: +8 x -5 ii) Using Restoring Division Algorithm, divide: +11 / +3	10	L2	CO1
3	Design 4 bit Parallel and 4bit CLA adders. Draw the circuits for both type of adders.	10	L3	CO2
4	-Simplify the Boolean function into Sum of products (SOP) and Products of sum (POS) form. f (A, B, C, D) = ∑ m (0, 1, 2, 5, 8, 9, 10) -Simplify the following Boolean expressions using K maps. F (w, x, y, z) = ∑ m (0,1,4,5,8,9,14,15) + dc(11,12)	10	L3	CO1
5	Design a single bit comparator, by writing the truth table and realize the circuits. Indicate the circuits for designing n bit comparator using cascading one bit comparator	10	L4	CO2

Course Outcomes: After completing the course, the students will be able to:-	
CO 1	Apply design requirements for digital systems and Computer organization
CO 2	Analyze the models used for designing various Ccombinational and Sequential circuits
CO 3	Develop applications of synchronous sequential networks using flip flops, registers and counters
CO 4	Design optimized modern processors and memories for given specifications
CO 5	Investigate techniques of digital system design for building industry relevant real-world systems using electronic components and modern tools

Course Outcomes:

Marks Distribution	Particulars		CO1	CO2	CO3	CO4	CO5	L1	L2	L3	L4	L5	L6
	Test	Max Marks	30	20	**	**	**	**	10	30	10	**	**

CS234AI

USN

RV COLLEGE OF ENGINEERING®

(An Autonomous Institution Affiliated to VTU)

III Semester B. E. Examinations April/May-2024

Computer Science and Engineering

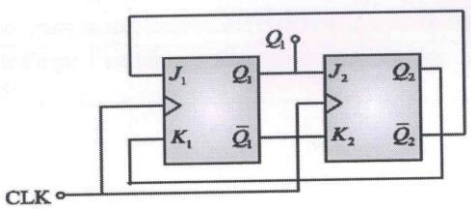
APPLIED DIGITAL LOGIC DESIGN AND COMPUTER ORGANIZATION

Time: 03 Hours ***Maximum Marks: 100***

Instructions to candidates:

1. Answer all questions from Part A. Part A questions should be answered in first three pages of the answer book only.
2. Answer FIVE full questions from Part B. In Part B question number 2 is compulsory. Answer any one full question from 3 and 4, 5 and 6, 7 and 8, 9 and 10.

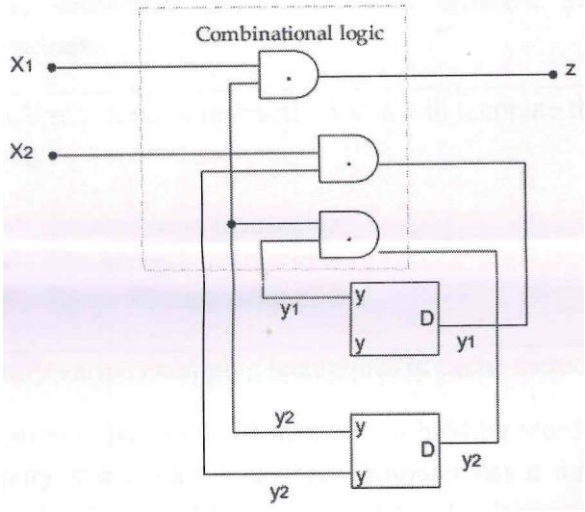
PART-A

1	1.1	Represent the number 263.3 in $32bit$ floating point representation.	02
	1.2	Identify the logic gate utilized in the following applications: a) Two way switch b) To access a bank locker c) Duplicate key to open the door of a house d) Detection of $1bit$ errors in digital communication	02
	1.3	Represent the following expression using only 2 input $NAND$ gates: $(AB + A'B')(CD' + C'D)$	02
	1.4	Design a 32to1 multiplexer, using lower order multiplexers.	02
	1.5	Realize (conversion) the following: JK flip flop using SR flip flop.	02
	1.6	Multiply +13 with −6 using bit pair recoding.	02
	1.7	Registers $R4$ and $R5$ contain the decimal numbers 2000 and 3000 before each of the following addressing modes is used to access a memory operand. What is the effective address (EA) in each case? a) Add $12(R4), R5$ b) Move $(R4, R5), R0$	02
	1.8	Illustrate with examples any two assembler directives.	02
	1.9	The outputs of the two flipflops $Q1, Q2$ in the Fig. 1.9 are initialized to 0,0. Give the sequence generated at $Q1$ upon the application of clock signal.  Fig. 1.9	02
	1.10	A program contains 1000 instruction, out of that 25% of instructions requires 4 clock cycles, 40% instruction required 5clock cycles and remaining required 3 clock cycles for execution. Find the total time required to execute the program running on a $1GHz$ machine.	02

PART-B

2	a	A small company has 100shares of stock and each share entities its owner to vote at a stockholders meeting. Mr. A owns 10 shares, Mr. Bowns 20shares, Mr. C owns 30 shares and Mr. D owns 40shares. A two-thirds majority is required in order to pass a measure at a stockholders meeting, each of the four stockholders has a switch which he or she closes to vote 'yes' for all of his or her shares and opens to vote $'no'$. A switching circuit is to be designed to turn on a light if the measure passes i) Derive a truth table for the output function Z ii) Write the minterm expansion for Z and simplify using K map iii) Design two level gate logic diagram using the simplified SoP.	06
	b	Simplify the Boolean function using QM Method. F (P, Q, R, S) = ∑ m (0, 1,2, 4, 5,7,8, 9,14,15)	06
	c	Using Restoring Division Algorithm, divide: +13 / +7	04
3	a	Discuss the merits of CLA adder over normal ripple adders. Derive the Carry equations, C1 to C4 for 4-bit CLA adder. Draw the circuit diagram, clearly indicating the circuits and all the connections.	06
	b	Derive the characteristic equation and excitation table for SR and JK flip flops.	04
	c	Explicate $0's$ and $1's$ catching problem present in JK flip flop. How can this problem be resolved?	06
		OR	
4	a	A positive edge triggered Dflip flop is connected to a positive edge triggered JK flip flop. The Qoutput of the D flip flop is connected to both the J and K inputs of the JK flip flop, while the Qoutput of the JK flipflop is connected to the input of the D flip flop. Initially, the output of the D flip flop is set to logic one and and the output of the JK flip flop is cleared. What is the bit sequence generated at the Qoutput of the JK flip flop when the flip flops are connected to a free running common clock? Both the flip flops have non zero propagation delays.	04
	b	Design JK flip flop using a D flip-flop, a 2 to 1 mux and an inverter if required and show all the steps.	06
	c	Design a $4-bit$Universal shift register using positive edge triggered D flip- flops to operate as indicted in the table below.	06

$ModeSelect$		$RegisterOperation$
$A1$	$A0$	
0	0	$Complementcontents$
0	1	$CircularShift$
1	0	$Shift$
1	1	$load$

5	a	Analyze the following circuit in Fig. 5.a by deriving the excitation equations, transition equations, excitation table, transition table and state diagram.  Fig. 5.a	10
	b	Design Synchronous Counter using JK Positive Edge Triggered Flip Flops, which produces the following sequence: 111,110,101,011,001, 000.	06
		OR	
6	a	Give any four differences between mealy and moore models. Recognize the sequence 0110/1001 using mealy model.	08
	b	For the state table given in Table 6.b, eliminate the redundant states using Row elimination implication table method. Show the reduced state table and state transition diagram.	

$PresentState$	$NextState$		$Output$
	$x=0$	$x=1$	
S_0	S_1	S_2	0
S_1	S_3	S_5	1
S_2	S_5	S_4	0
S_3	S_1	S_6	1
S_4	S_5	S_2	0
S_5	S_4	S_3	1
S_6	S_5	S_6	0

08

7	a	List and explain the basic steps needed to execute the machine instruction $ADDLOCA,R0$ in terms of transfers between the components of processor, memory and some control commands.	04
	b	List and explain any five addressing modes with examples. A processor has 40 distinct instructions and 24general purpose registers.A32 – bit instruction word has an opcode, two registers operands and an immediate operand. Give the number of bits available for the immediate operand field.	08
	c	Describe the factors that affect the performance of the computer? Explain any two ways in which performance can be improved.	04
		OR	
8	a	Write an ALP Program to add set of N numbers.	05
	b	Define subroutine linkage. Explain different mechanism of passing parameters to subroutines.	05

	c	Write a sequence of instructions that will compute the value of $y = x^2 + 3x + 6$ for a given x using: i) Three-address instructions ii) Two-address instructions iii) Zero-address instructions	06
9	a	Identify various mapping techniques in cache memory along with necessary diagrams. A computer has an $8GB$ memory with $64bit$ word sizes. ($1word = 4bytes$). Each block of memory stores 16 words. The computer has a direct-mapped cache of 128 blocks. The computer uses word level addressing. What is the address format?	08
	b	Write the control sequence, with comments for the execution of the following instructions, using Single bus and Three bus data paths. $ADD(R5), R3$.	08
		OR	
10	a	Examine all the memory types available under RAM and ROM, which are used in building embedded systems and Personal computers. Design $2M \times 32$ memory chip using $512k \times 8$ memory chip along with external connections.	08
	b	Illustrate with neat diagrams, the working of hardwired control unit	08

Signature of Scrutinizer **Signature of Chairman**

Name: **Name:**

18CS35

USN | | | | | | | | | | |

RV COLLEGE OF ENGINEERING®

(An Autonomous Institution affiliated to VTU)

III Semester B. E. Supplementary Examinations Feb / Mar - 2024

Computer Science and Engineering

FOUNDATIONS OF COMPUTER SYSTEMS DESIGN

Time: 03 Hours *Maximum Marks: 100*

Instructions to candidates:

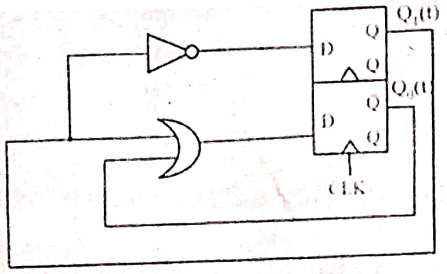
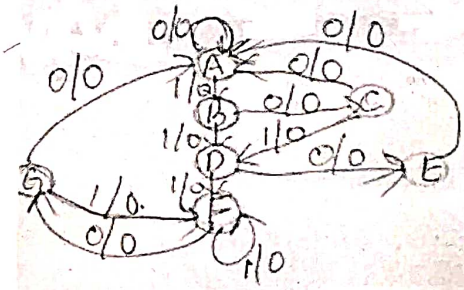
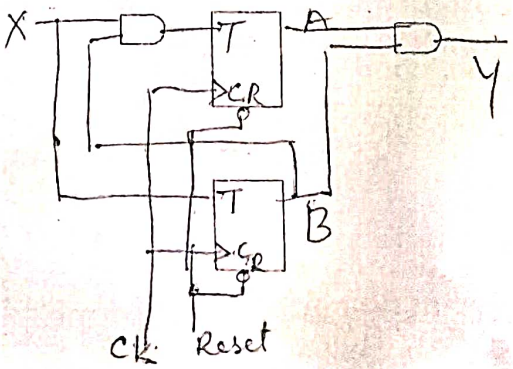
1. Answer all questions from Part A. Part A questions should be answered in first three pages of the answer book only.
2. Answer FIVE full questions from Part B. In Part B question number 2, 7 and 8 are compulsory. Answer any one full question from 3 and 4 & one full question from 5 and 6

PART-A

1	1.1	Using Booth's algorithm, multiply 13 & −11	02
	1.2	The following bit pattern represents a floating-point number in *IEEE* 754 single precision format 1 10000011 10100000000000000000000 What is the value of the number in decimal form?	02
	1.3	With an example, show the big endian and little endian format.	02
	1.4	Represent −10 using sign and magnitude and 1's complement representation.	02
	1.5	If two full adders and half adders are implemented using gates, then for the addition of two, 17 bit numbers (using minimum no. of gates), compute the number of half adders and full adders required.	02
	1.6	Implement the following expression using only one 8 to 1 mux using a,b,c as address $F(a,b,c,d) = \Sigma m(0,1,3,5,6,8,9,11,13,14)$.	02
	1.7	The minimized form of the logical expression $(A'B'C' + A'BC' + A'BC + A'BC')$ is _____.	02
	1.8	Realize 3 to 8 line decoder using 2 to 4 line decoders.	02
	1.9	What is nested subroutine?	02
	1.10	Give methods to reduce memory access time.	02

PART-B

2	a	Simplify the following expression using k-map and realize the simplified expression using *NAND* gates. $F(A,B,C,D) = \Sigma m(0,1,2,4,6,7,9,15) + dc(3,12)$	06
	b	Using Restoring division algorithm, divide the number 8|3	06
	c	Realize a single circuit for adder and subtractor using 4 bit parallel adder.	04
3	a	Write the function table and derive the characteristic equations for the following: i) *SR* flip-flop ii) *JK* flip-flop iii) *D* flip-flop iv) *T* flip-flop	08

	b	Realize using *NAND* gates the Master Slave *JK* flip flop and explain. With relevant timing diagrams, explain 0's and 1's catching.	08
		OR	
4	a	Design a mod 16 Asynchronous counter using negative edge triggered JK flip-flops.	04
	b	Design a mod 5 ring counter and mod 12 Johnson counter, write the truth table.	06
	c	Design a mod 6 synchronous up counter using *D* flip flops.	06
5	a	Specify the advantages of Mealy model over Moore model. Detect the sequence 1101 using Mealy model.	06
	b	Analyze the following circuit and write the state equation for the fig 5b.  Figure 5b	04
	c	Explain the three State assignment rules? Show the state table and state assignment table using binary sequence for the fig 5c.  Figure 5c	06
		OR	
6	a	Analyze the following circuit and write the excitation equations, derive transition equations, compute excitation table, transition table and draw the state diagram for the fig 6a. Is this a mealy or a moore model and why?  Fig 6a	10

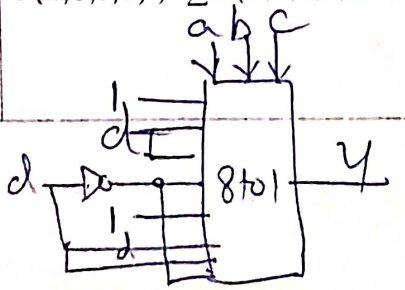
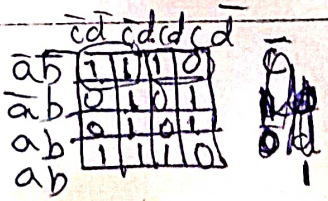
	b	What are the advantages of state reduction? Using row elimination or implication methods eliminate the redundant states and draw the reduced state diagram with the state Table 7b given below. Table 7b *Present state* | *Next State* $X=0$ | *Next State* $X=1$ | *Output Y* A | B | C | 1 B | D | C | 0 C | F | E | 0 D | E | B | 1 E | B | C | 1 F | C | E | 0 G | F | G | 0	06
7	a	What is an instruction code? Explain in detail any five addressing modes with examples.	08
	b	Explain the instruction cycle with a neat diagram and also mention the registered involved in this cycle.	04
	c	What is safe push and safe pop?	04
8	a	Mention the applications of cache. Discuss direct, associative and set associate cache mapping techniques with suitable examples.	06
	b	Illustrate with a neat diagram, the hardwired control with its advantages.	05
	C	Define the Static $RAM(SRAM)$. Design 128×16 – bit RAM using 128×4 – bit RAM. Also explain the working of $SRAM$ cell.	05

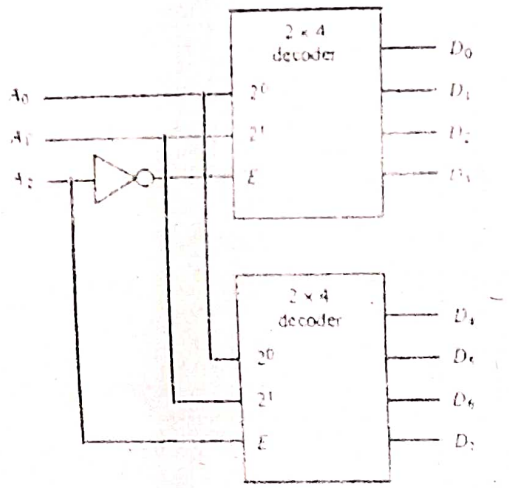
UG

RV College of Engineering®, Bengaluru-560 059
Autonomous Institution affiliated to Visvesvaraya Technological University

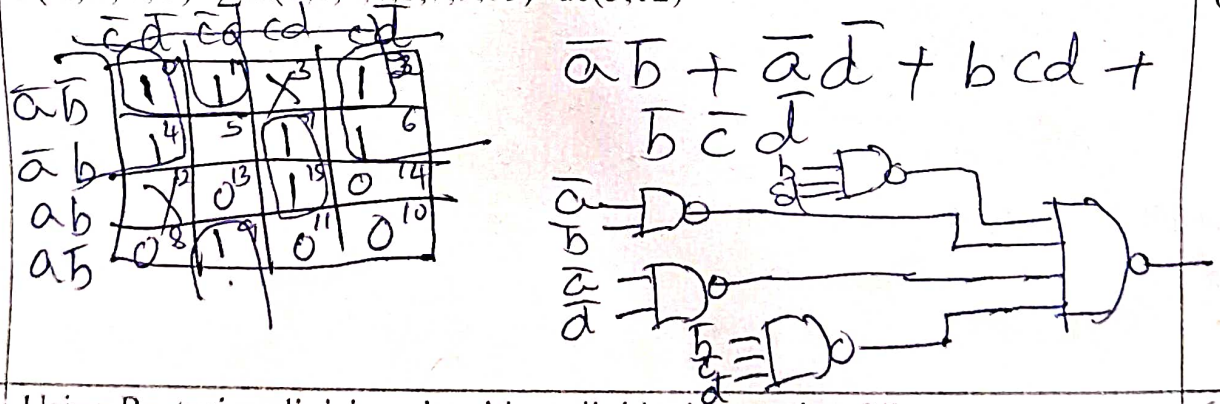
Scheme and solution

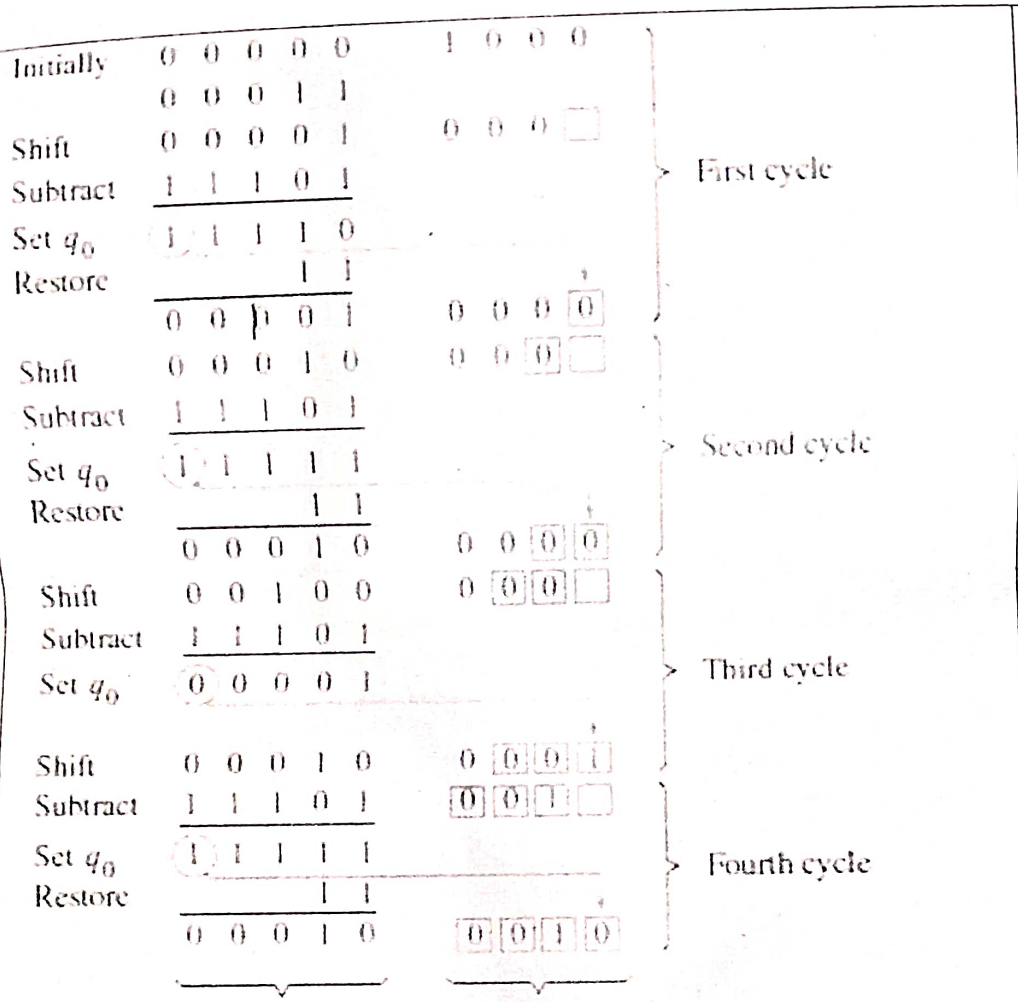
COURSE CODE:18CS35 | COURSE TITLE:Foundations of Computer System Design

Question No	Answer	Marks
1.1	booth's algorithm , multiply +13 & - 11 +13 = 01101 -11 = 10101 -13 = 10011 Recoding Method 10101 -1 +1 -1 +1 -1 01101 -1 +1 -1 +1 -1 10 10 01 11 10011 00000 11010 11111 00110 00011 01100 11001 10000 1000111110001	2
1.2	The value of the number in decimal form is	2
1.3	Big-endian stores the most significant bytes first, whereas little-endian stores the least significant bytes first.	2
1.4	-10 in sign and magnitude number is 1 0001 0000 1` complement of 00001010 is 11110101	2
1.5	1.16	2
1.6	**8 to 1 mux.** **F(a,b,c,d)**)=∑m(0,1,3,5,6,8,9,11,13,14)   	2

1.7	**A'C'+BC'+A'B**	2
1.8	**3 to 8 line decoder using 2 to 4 line decoders.** 	2
1.9	Nested subroutines allow you to direct program flow from the main program to a subroutine, and then to another subroutine. Nested subroutines make complex programming easier and program operation faster, because the programmer does not have to continually return from one subroutine to enter another	2
1.10	**to reduce memory access time:** **The two basic ways of ensuring this are:** **□ Reducing the number of access to memory via:** **o Checking for location of content** **o Using specialized algorithms** **□ Using very fast memory access devices**	2

Part B

Q. No.	Questions	Marks
2a	F(A,B,C,D)=∑m(0,1,2,4,6,7,9,15)+dc(3,12) 	6
B	Using Restoring division algorithm, divide the number 8/3.	6



c

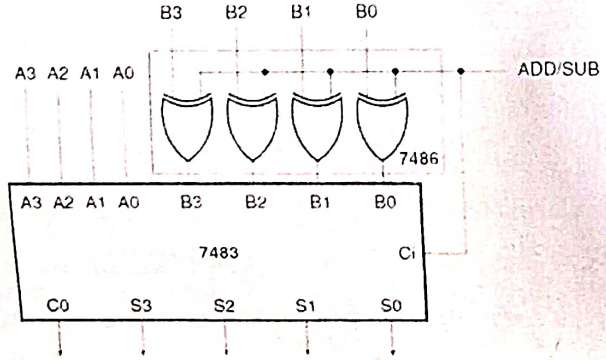


Figure 16.7 Adder–subtractor circuit (example 16.8)

3a function table and the characteristic equations for — 8

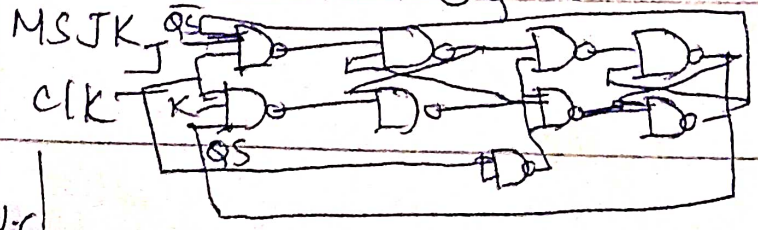
i) SR flip-flop = $S + \bar{R}Q$
ii) JK flip flop = $J\bar{Q} + \bar{K}Q$
iii) D flip-flop = D
iv) T flip-flop = $T \oplus Q$

1 m

SR	Q
00	N.C
01	0
10	1
11	f.s

JK	Q
00	N.C
01	0
10	1
11	$\bar{Q}$

b — 8



Qs – 2m

$\bar{Q}s$ Explⁿ 2m

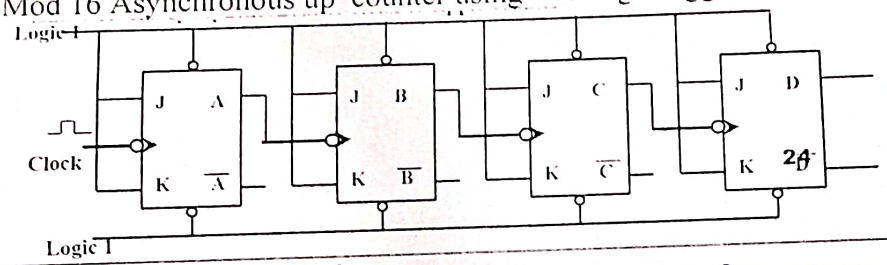
D	Q	T	Q
0	0	0	N.C
1	1	1	$\bar{Q}$

0's Catching – 2m

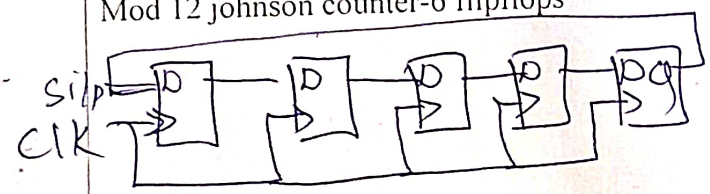
1's Catching – 2m

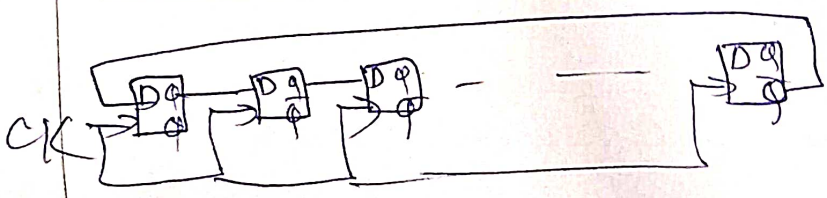
OR

4a Mod 16 Asynchronous up counter using -ve edge triggered T flip-flops. 4



b Mod 5 ring counter-5 flipflops
Mod 12 johnson counter-6 flipflops 6

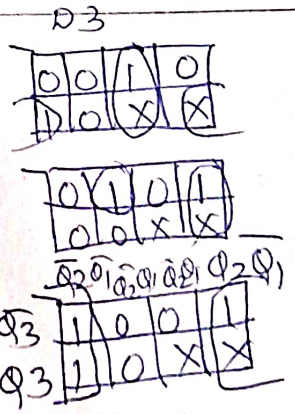




00000
10000
01000
00100
00010
00001

c mod 6, Synchronous up counter using D flip flops. 6

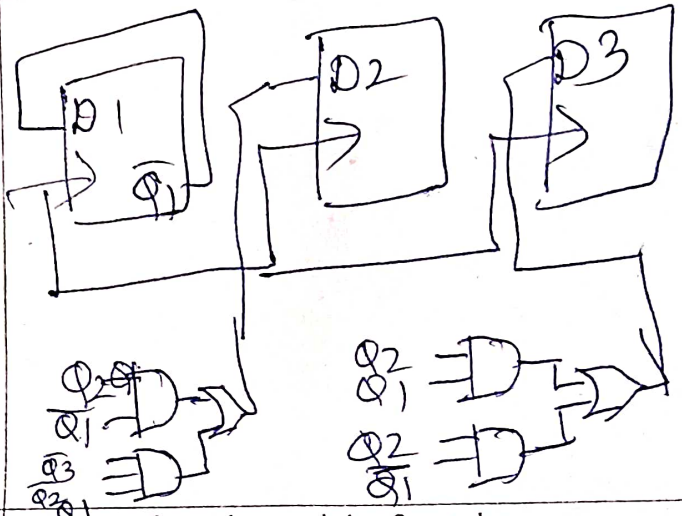
Q_3 Q_2 Q_1	Q_3^+ Q_2^+ Q_1^+	D_3 D_2 D_1
0 0 0	0 0 1	0 0 1
0 0 1	0 1 0	0 1 0
0 1 0	0 1 1	0 1 1
0 1 1	1 0 0	1 0 0
1 0 0	1 0 1	1 0 1
1 0 1	0 0 0	0 0 0
1 1 0	X X X	X X X
1 1 1	X X X	X X X



$D_1 = \bar{Q}_1$

$D_2 = Q_2\bar{Q}_1 + \bar{Q}_3Q_2Q_1$

$D_3 = Q_2Q_1 + Q_2\bar{Q}_1$



5a	Advs of mealy models -3 marks 1101 is the sequence detected.-3 marks	6
b	Q1(t+1)=Q1(t)' , Q0(t+1)=Q0(t)+Q1(t)	4

6

Present state	Next state		Output	
	X=0	X=1	X=0	X=1
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	(g=e)	f	0	1

−2m

State assignment rules:4 M

OR

6a	Excitation equations T A = B x T B = x y = A B Transition equations	10

$$A=(\overline{Bx})A+(Bx)\bar{A}$$

$$=A\bar{B}+A\bar{x}+\bar{A}Bx$$

$$B=x\oplus B$$

Table 5-5
State Table for Sequential Circuit with T Flip-Flops

Present State		Input	Next State		Output
A	B	x	A	B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	1

b

Advantages of state reduction.-optimization of circuits, more economical, lessened hardware, increased speed. — 1

6

Present State	Next State		Output Y
	X = 0	X = 1	
A	B	C	1
B	D	C	0
C	C	A	0
D	A	B	1
G	C	G	0

— 3 m

Table Reduced state table

$$A=(\overline{Bx})A+(Bx)\overline{A}$$

$$=A\overline{B}+A\overline{x}+\overline{A}Bx$$

$$B=x\oplus B$$

Table 5-5
State Table for Sequential Circuit with T Flip-Flops

Present State		Input	Next State		Output
A	B	x	A	B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	1

b

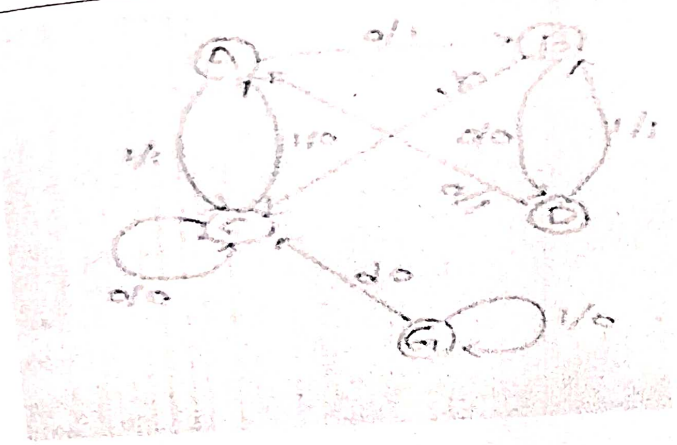
Advantages of state reduction.-optimization of circuits, more economical, lessened hardware, increased speed. — 1

6

Present State	Next State		Output Y
	X = 0	X = 1	
A	B	C	1
B	D	C	0
C	C	A	0
D	A	B	1
G	C	G	0

— 3 m

Table 5.8 Reduced state table



−2m

Fig20. Reduced state diagram

7a	Implied mode In this mode, the instruction contains an indirect definition of the operand. An example of an implied mode instruction is CMA (complement accumulator). Here, the operand (complement) is implicitly specified in the instruction. Immediate addressing mode In this mode, the instruction contains both the opcode and the operand. It can be said that an instruction that uses the immediate addressing mode contains an operand field in place of an address field. The operation, as well as the operands, are mentioned in the instruction. An example of immediate addressing mode instruction is ADD 10. Here, ADD, which is the operation, and 10, which is the operand, are specified. Register mode In this mode, the instruction specifies a register. This register stores the operand. An example of register mode instruction is: AC = AC + [R] This will add the operand stored at register R to the operand stored in the accumulator. Register indirect mode In this mode, the instruction specifies a register. This register stores the effective address of the operand. An instruction that uses register indirect addressing mode is:	2+6

AC = AC + [[R]]

Here, the contents which reside in the memory location specified by the register R will be added to the contents of the accumulator.

Direct addressing mode

In this mode, the instruction specifies an address. This address is the address of the operand. An example of a direct addressing mode instruction is: AC = AC + [X]

This will add the operand stored at address X with the operand stored in the accumulator. This mode is also referred to as absolute addressing mode.

Indirect addressing mode

In this mode, the instruction specifies an address. The memory location specified by the address contains the address of the operand. An example of an indirect addressing mode instruction is:

AC = AC + [[X]]

This will add the operand stored at the address specified by the memory location X with the contents of the accumulator.

Auto-increment/decrement mode

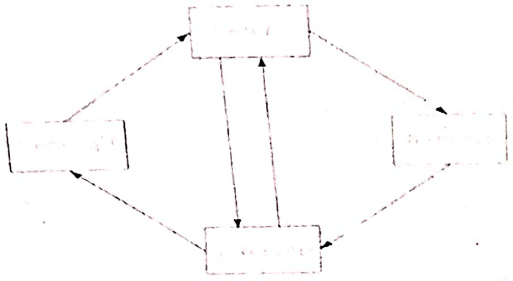
In this mode, the instruction specifies a register which points to a memory address that contains the operand. However, after the address stored in the register is accessed, the address is incremented or decremented, as specified. The next operand is found by the new value stored in the register.

Relative address mode

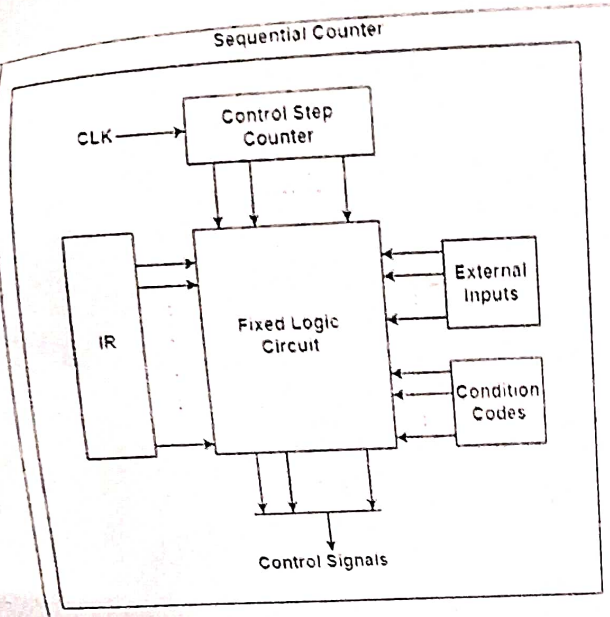
In this mode, the contents of the address field are added to the constant stored in the program counter. The result of the addition gives the address of the operand. For example, suppose the address field contains 850, and the program counter contains 20, then the operand will be at memory location 850 + 20 = 870.

Indexed addressing mode

In this mode, the address of the operand is determined by adding the

	contents of the address field and the contents of the index register. Base register addressing mode In this mode, the address of the operand is determined by adding the contents of the address field and the contents of the base register.	
b	 The Instruction Cycle The instruction cycle helps the CPU perform its primary job of expecting tasks. The instruction cycle comprises three main stages and is also addressed as the fetch-decode-execute cycle or fetch-execute cycle because of the steps involved. The stages are as follows: **Fetch stage** **Decode stage** **Execute stage** The following procedures are included in each instruction cycle: The CPU reads the effective address from memory if the instruction has an indirect address The register retrieves instructions from memory. It is capable of carrying out the command Decoding is the primary task assigned The loop continues carrying out the task repeatedly until a halt condition is met. Till then, the cycle keeps on restarting. MAR, MDR, IR, PC, GPR registers are involved	4
c	safe push and safe pop. The address of the element at the top of the stack is continuously	4

	influenced by the stack pointer. It has the ability to add or remove elements from the stack. Push operation refers to insertion operations and pop operation to deletion operations.	
8a	**Direct Mapping:** Each block from main memory has only one possible place in the cache organization in this technique. For example : every block i of the main memory can be mapped to block j of the cache using the formula : **j = i modulo m** Where : i = main memory block number j = cache block number m = number of blocks in the cache The address here is divided into 3 fields : Tag, Block & Word. **Associative Mapping:** Here the mapping of the main memory block can be done with any of the cache block. The memory address has only 2 fields here: word & tag. This technique is called as fully associative cache mapping. ***Example:*** If we have a fully associative mapped cache of 8 KB size with block size = 128 bytes and say, the size of main memory is = 64 KB. Then: Number of bits for the physical address = 16 bits (as memory size = 64 KB = $2^6 \times 2^{10} = 2^{16}$) Number of bits in block offset = 7 bits (as block size = 128 bytes = 2^7) No of tag bits = Number of bits for the physical address – Number of bits in block offset = 16-7 = 9 bits No of cache Blocks = Cache size/block size = 8 KB / 128 Bytes = 8×1024 Bytes/128 Bytes = 2^6 blocks. **3. Set – Associative Mapping:** It is the combination of advantages of both direct & associative mapping. Here, the cache consists of a number sets, each of which consists of a number of blocks. The relationships are : **n = w * L** **i = j modulo w** where i : cache set number j : main memory block number n : number of blocks in the cache w : number of sets L : number of lines in each set	6
b	Illustrate with a neat diagram, the hardwired control. A hardwired control is a mechanism of producing control signals using **Finite State Machines (FSM)** appropriately. It is designed as a **sequential logic circuit**. The final circuit is constructed by physically connecting the components such as **gates**, **flip flops**, and drums. Hence, it is named a hardwired controller.	5



c	Design 128×8 RAM using 128×4 RAM chips. SRAM consists of flip-flops. bistable circuits composed of four to six transistors. Once a flip-flop stores a bit, it keeps that value until the opposite value is stored in it. SRAM is used primarily for small amounts of memory called registers in a computer's CPU and for fast "cache" memory. Design 128×16 -bit RAM using 128×4 -bit RAM Solution: p = 128 / 128 = 1; q = 16 / 4 = 4 . •Therefore, p×q = 1×4 = 4 memory chips of size 128×4 are required to construct 128×16 bit RAM	5

128×16

4

−2

−3 marks

Design − 1m