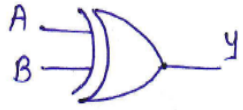
WINTER – 2018 EXAMINATION
MODEL ANSWER

Subject: Digital Techniques & Microprocessor **Subject Code:** 22323

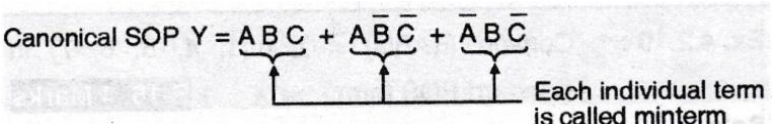
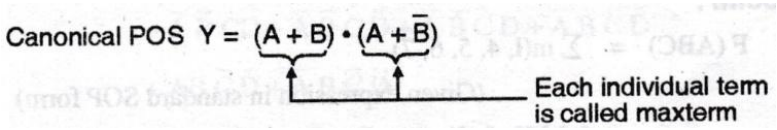
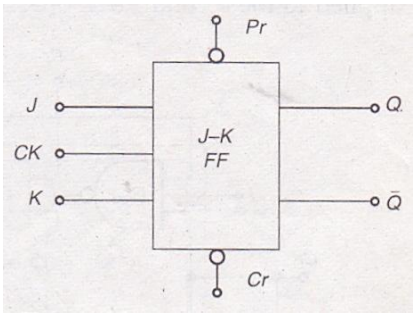
Important Instructions to examiners:

1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills).
4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.
7) For programming language papers, credit may be given to any other program based on equivalent concept.

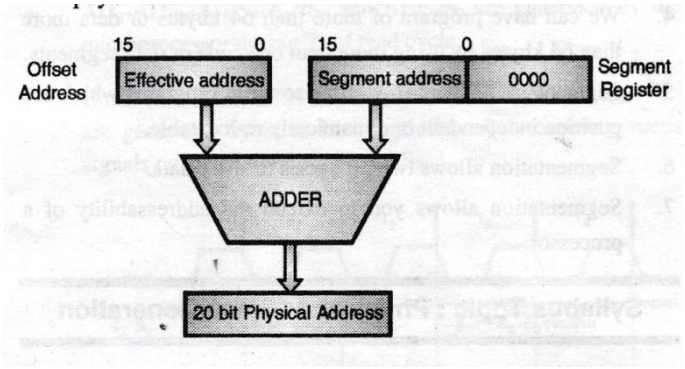
Q. No.	Sub Q.N.	Answer	Marking Scheme
1.	**a) Ans.**	**Attempt any FIVE of the following:** **Draw symbol and write truth table of EX-OR gate.** **Symbol**  **Truth Table** Truth Table for two input EX-OR gate. A logical gate whose output is one when odd number of inputs are one, for any other condition output is low. Inputs: A, B — Output: Y 0 0 0 1 0 1 0 1 1 1 1 0	**10** **2M** ***Symbol 1M*** ***Truth Table 1M***

WINTER – 2018 EXAMINATION
MODEL ANSWER

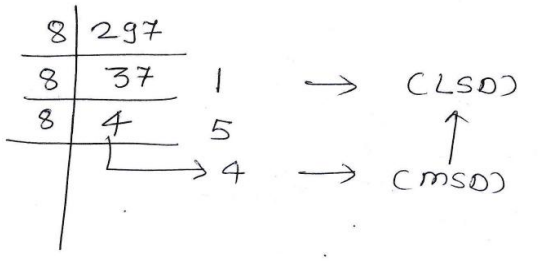
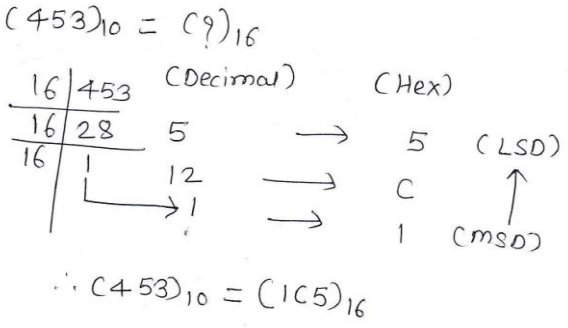
Subject: Digital Techniques & Microprocessor **Subject Code:** 22323

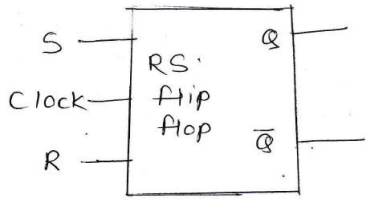
	b) **Ans.**	**Define terms "Minterm" and "Maxterm" with proper example of each.** **Minterm:** Each individual term in the canonical SOP form is called as Minterm. Example:  **Maxterm:** Each individual term in the canonical POS form is called as Maxterm. Example:  	***2M*** ***Each Definition with example 1M*** ***Each Definition with example 1M***
	c) **Ans.**	**Draw symbol of JK flip-flop and write its truth table.** **Symbol**  **Truth Table:** Inputs: J_n, K_n; Output: Q_{n+1} 0 0 Q_n 0 1 0 1 0 1 1 1 $\overline{Q_n}$	**2M** ***Symbol 1M*** ***Truth Table 1M***

	d) Ans.	**State importance of pipelining in 8086 microprocessor** • In pipelining, while the current instruction is executing, next instruction is fetched using a queue. • Pipelining enables many instructions to be executed at the same time. • It allows execution to be done in fewer cycles. • Speed up the execution speed of the processor. • More efficient use of processor.	***2M*** ***Any two points 2M***
	e) Ans.	**Give any four applications of digital circuits.** Applications of digital circuits i) Object Counter ii) Dancing Lights iii) Scrolling Notice board iv) Multiplexer v) Digital Computers vi) Washing machines, Television vii) Digital Calculators viii) Military Systems ix) Medical Equipments x) Mobile Phones xi) Radar navigation and guiding systems xii) Microprocessors	***2M*** ***Any relevant four applications 2M***
	f) **Ans.**	**Define the following terms –** **(i) Physical Address** **(ii) Effective Address** **(i) Physical Address** ***(Note: Diagram is Optional)*** Physical: The address given by BIU is 20 bit called as physical address. It is the actual address of the memory location accessed by the microprocessor. It is calculated as	***2M*** ***Each definition 1M***

		 **(ii) Effective Address** Effective Address: Effective address or the offset address is the offset for a memory operand. It is an unassigned 16 bit number that gives the operand's distance in bytes from the beginning of the segment.	
	g) **Ans**	**Choose instruction for following situations:** **(i) Addition of 16 bit Hex. No with carry** **(ii) Division of 8 bit No.** **(iii) Rotate content of BL by 4 bit.** **(iv) Perform logical AND operation of AX and BX** **(i) Addition of 16 bit Hex. No with carry** ***(Note any other relevant registers shall also be considered)*** ADC Destination 16, Source 16 **OR** ADC AX, BX **OR** ADC AX, 4500H **(ii) Division of 8 bit No.** ***(Note any other relevant registers shall also be considered)*** DIV SOURCE **OR** DIV BL	***2M*** ***Each instruction ½ M***

		(iii)Rotate content of BL by 4 bit.
		MOV CL,04H
		ROR BL, CL
		OR
		MOV CL, 04H
		ROL BL, CL
		
		(iv) Perform logical AND operation of AX and BX
		AND AX,BX	

2.	**a)** **Ans.**	**Attempt any THREE of the following:** **Convert following decimal to octal and Hexadecimal** **i) $(297)_{10}$ = ( )$_8$** **ii) $(453)_{10}$ = ( )$_{16}$** **(i) $(297)_{10}$ = ( )$_8$**  $\therefore (297)_{10} = (451)_8$ **(ii) $(453)_{10}$ = ( )$_{16}$**  	**12** **4M** ***Each conversion 2M***

	b) **Ans.**	**Convert the given minterm into standard POS form.** $Y(A,B,CD) = (\bar{A}.BC) + (B.\bar{C}\,\bar{D}) + (\bar{A}\,\bar{B})$ ***Note: Solution is given by considering*** $Y(A,B,CD)$ ***as*** $Y(A,B,C,D)$ $Y(A,B,C,D) = \bar{A}BC + B\bar{C}\bar{D} + \bar{A}\bar{B}$ Converting into standard SOP form, $Y(A,B,C,D) = \bar{A}BC(D+\bar{D}) + (A+\bar{A})B\bar{C}\bar{D} + \bar{A}\bar{B}(C+\bar{C})(D+\bar{D})$ $= \bar{A}BCD + \bar{A}BC\bar{D} + AB\bar{C}\bar{D} + \bar{A}B\bar{C}\bar{D} + \bar{A}\bar{B}CD + \bar{A}BC\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}\bar{C}\bar{D}$ $= \sum m(7,6,12,4,3,2,1,0)$ $= \sum m(0,1,2,3,4,6,7,12)$ $= \prod M(5,8,9,10,11,13,14,15)$ ... Standard Pos form $= (A+\bar{B}+C+\bar{D})(\bar{A}+B+C+D)(\bar{A}+B+C+\bar{D})$ $(\bar{A}+B+\bar{C}+D)(\bar{A}+B+\bar{C}+\bar{D})(\bar{A}+\bar{B}+C+\bar{D})$ $(\bar{A}+\bar{B}+\bar{C}+D)(\bar{A}+\bar{B}+\bar{C}+\bar{D})$.	***4M*** ***Standard SOP form 2M*** ***Conversion to Standard POS 2M***
	c) **Ans.**	**Draw symbol and write truth table for the following flip flop and give one application of each.** **i) Clocked R-S flip flop** **ii) T- flip flop** **(i) Clocked R-S flip flop** **Symbol**  	***4M*** ***Symbol ½ M***

Truth Table

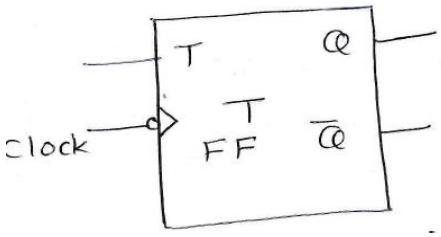
Clock	S	R	Q_{n+1}	$\overline{Q}_{n+1}$	Remark
0	X	X	Q_n	$\overline{Q}_n$	No change
1	0	0	Q_n	$\overline{Q}_n$	No Change
1	0	1	0	1	Reset
1	1	0	1	0	Set
1	1	1	Race	Race	Avoid

Truth table 1M

Application:
i) Clocked RS flip-flop can be used in sequential circuits.
ii) It can be used to design counters.
iii) It can be used as a latch in digital circuits.

Any one Application ½ M

(ii) T- flip flop
Symbol

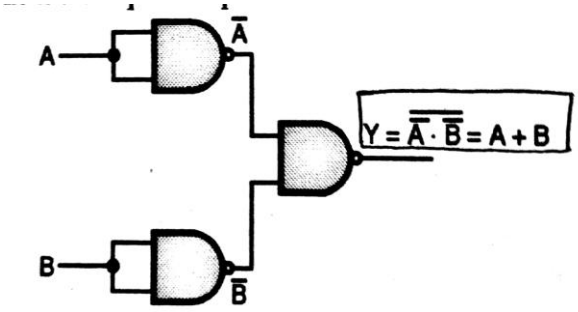


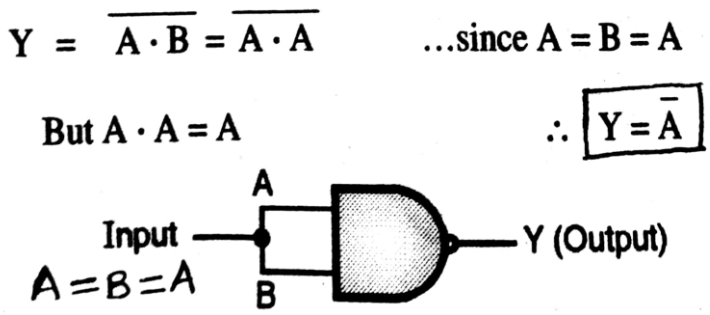
Symbol ½ M

Truth Table

T	Q_{n+1}
0	Q_n
1	$\overline{Q}_n$

Truth table 1M

		Application: i) Used to design counters in digital circuits. ii) Can be used in frequency divider circuits.	***Any one Application ½ M***
	d) Ans.	**Prove $A(\bar{A}+C)(\bar{A}B+C)(\bar{A}BC+\bar{C})=0$** ***Note: Any other relevant laws applied shall be considered while obtaining the correct answer.*** L·H·S. $= A(\bar{A}+C)(\bar{A}B+C)(\bar{A}BC+\bar{C})$ $=(A\bar{A}+AC)(\bar{A}B+C)(\bar{A}BC+\bar{C})$ $=(0+AC)(\bar{A}B+C)(\bar{A}BC+\bar{C})$ $(\because A\bar{A}=0)$ $=(A\bar{A}BC+AC)(\bar{A}BC+\bar{C})$ $(\because CC=C)$ $=(0+AC)(\bar{A}BC+\bar{C})$ $(\because A\bar{A}=0)$ $=A\bar{A}BC+AC\bar{C}$ $(\because CC=C)$ $=0+0$ $(\because A\bar{A}=0 \text{ and } C\bar{C}=0)$ $=0$ $=$ R·H·S· Hence Proved	***4M*** ***Correct solution 4M***
3	**a) Ans.**	**Attempt any <u>THREE</u> of the following:** **Implement OR gate and NOT gate using "Universal NAND gate". Write expressions for both.** 1. "OR" gate using "Universal NAND" gate:  	**12** **4M** ***Output Expression 1M*** ***Circuit Diagram 1M***

		2. "NOT" gate using "Universal NAND" gate: 	***Output Expression 1M*** ***Circuit Diagram 1M***
	b) **Ans**	**Explain following instructions for 8 bit and 16 bit data.** **(i) PUSH** **(ii) DAA** **(iii) IDJV** **(iv) XOR** ***Note: Any other relevant registers shall also be considered in the example/explanation.*** **(i) PUSH** Format: **PUSH source** This instruction decrements the SP (Stack Pointer) register (by 2) and copies the word specified by source to the location at the top of the stack. Here, Source can be a 16-bit general purpose register, segment register or memory location. Example- PUSH AX **OR** **PUSH AX** This instruction decrements the stack pointer by 2 and copies the 16 bit data from AX register to the stack segment where the stack pointer then points.	**4M** ***Explaination of each ½ M*** ***Example for each case ½ M***

(ii) DAA

DAA stands for Decimal Adjust Accumulator AL after BCD Addition

Explanation:

This instruction is used to make sure the result of adding two packed BCD numbers is adjusted to be a correct BCD number. The result of the addition must be in AL for DAA instruction to work correctly. If the lower nibble in AL after addition is > 9 or Auxiliary Carry Flag is set, then add 6 to lower nibble of AL. If the upper nibble in AL is > 9H or Carry Flag is set, and then add 6 to upper nibble of AL.

Example: - (Any Same Type of Example)

if AL=99 BCD and BL=99 BCD
Then ADD AL, BL

```
   1001 1001 = AL= 99 BCD
+  1001 1001 = BL = 99 BCD
----------------------------------------
    0011 0010 = AL =32 H and CF=1, AF=1
```

After the execution of DAA instruction, the result is CF = 1

```
   0011 0010 =AL =32 H AH =1
+  0110 0110
-------------------------
   1001 1000 =AL =98 in BCD
```

same type example for 16 bit can be considered.

OR

DAA instruction is used to convert the sum of two packed BCD numbers in the register AL into a correct BCD number.

Example :

```
MOV AL, 23H
MOV BL, 47H
ADD AL, BL
DAA
```

After the execution of the above instructions, the result in AL = 70H.

(iii) IDJV
(NOTE: CONSIDER THE GIVEN INSTRUCTION AS IDIV):
Syntax : IDIV source
It divides a signed word in AX by an signed byte in source during 16/8 division. Also it is used to divide a signed double word in DX,AX by an signed word in source during 16/8 division.
operation:

a. if the source is byte then

AL ⟵ AL/signed 8 bit source
AH ⟵ AL MOD signed 8 bit source

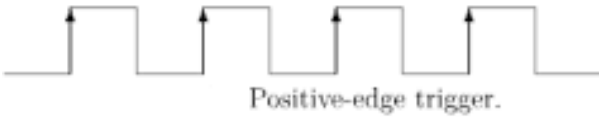
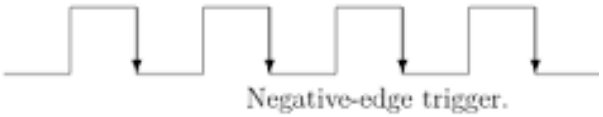
b. if the source is word then

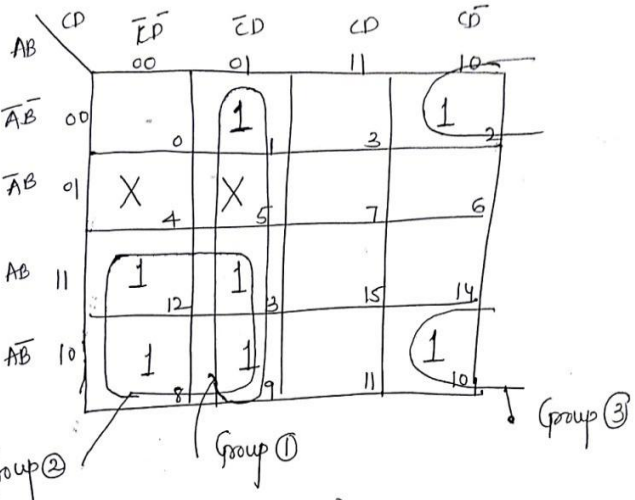
AX ⟵ DX,AX/signed 16 bit source
DX ⟵ DX,AX MOD signed 16 bit source

OR

IDIV BL
This instruction is used to divide signed word in AX register by signed byte in BL register. The quotient after division will be stored in AL register, whereas the remainder is stored in AH register.
IDIV BX
This instruction is used to divide signed double word in DX,AX register by signed word in BX register. The signed 16 bit quotient will be stored in AX register, whereas the signed 16 bit remainder is stored in AH register.

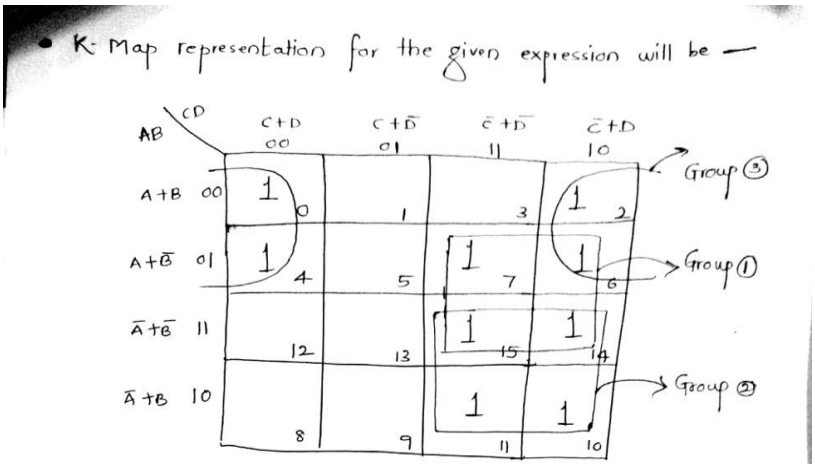
(iv) **XOR** − Used to perform Exclusive-OR operation over each bit in a byte/word with the corresponding bit in another byte/word.
Syntax: XOR Destination, Source
Example:
For 8bit data:
XOR AL, BL
This instruction performs Exclusive-OR bit by bit at AL with BL and the result is stored in AL..
For 16bit data:
XOR AX, BX
This instruction performs Exclusive-OR bit by bit word at AX with word in BX and the result is stored in AX.

	c) **Ans.**	**Draw waves for positive and negative triggering with proper labeling. Identify two situations where these triggering can be used?** ***Note: Any additional relevant point related to triggering situation shall be considered***  Positive-edge trigger.  Negative-edge trigger. 1. Edge triggering can be used in flipflops as clock input. 2. It is used in counters circuits. 3. They can be used in shift registers 4. They can be used to synchronous data.	**4M** ***Diagram 2M*** ***Any relevant situation where triggering is used 2M***
	d) **Ans**	**Simplify Y=F(A, B,CD)** **= ∑m (1, 2, 8, 9, 10, 12, 13) + d(4,5)** **Using K-map and write expression** ***Note: Solution is given considering Y=F(A, B,CD) as Y= F(A, B,C,D)***	**4M**

		• K-map representation for the given expression will be –  • To find equation (expression) – Group ① (QUAD) ⇒ $\bar{C}D$ Group ② (QUAD) ⇒ $A\bar{C}$ Group ③ (PAIR) ⇒ $\bar{B}C\bar{D}$ Therefore, The Required expression is, $f(A,B,C,D) = \bar{C}D + A\bar{C} + \bar{B}C\bar{D}$	***Correct K-map 2M*** ***Correct equation 2M***
4	**a)**	**Attempt any <u>THREE</u> of the following** **Suggest "Two instruction" for each of the following addressing modes.** **(i) Register Addressing Mode.** **(ii) Direct Addressing Mode** **(iii) Based Indexed Addressing Mode** **(iv) Immediate Addressing Mode.**	**12** **4M**

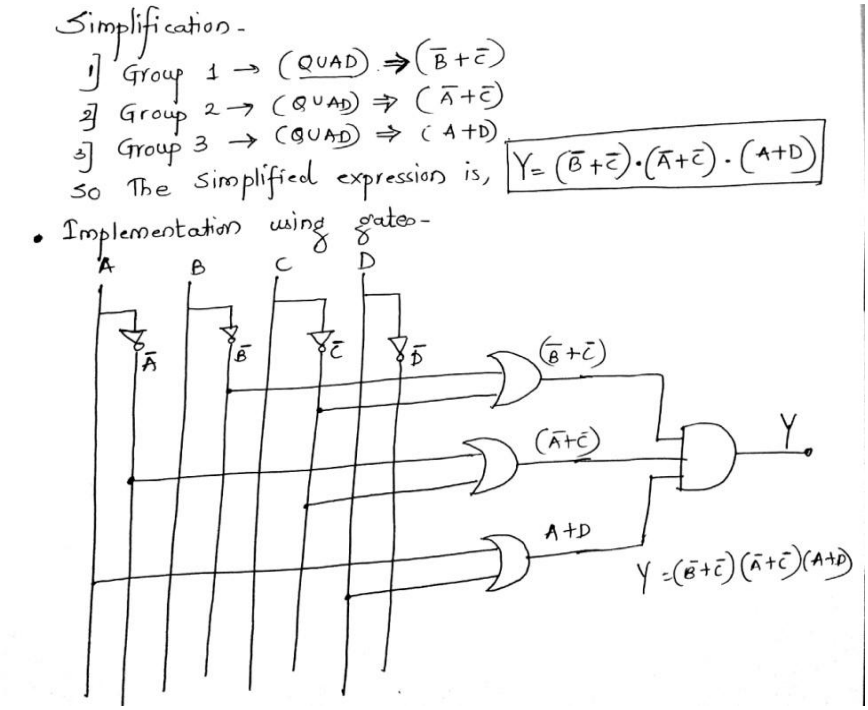
WINTER – 2018 EXAMINATION
MODEL ANSWER

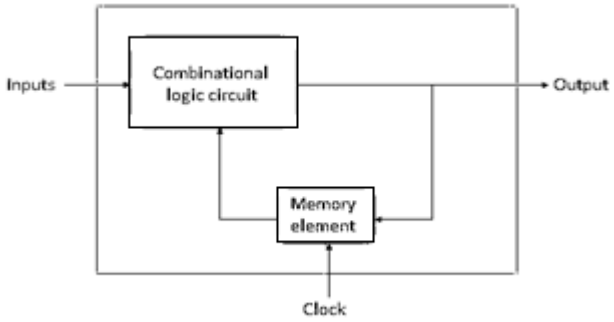
Subject: Digital Techniques & Microprocessor **Subject Code:** 22323

	Ans	i) **Register Addressing Mode:** a. MOV AX, CX b. AND AL, BL c. ROR AL, CL ii) **Direct addressing mode:** a.MOV AL, [3000H] b. AND AX,[8000H] c.INC [4712H] iii) **Based indexed Addressing mode:** 1.MOV AX, [BX][SI] 2.ADD AL, [BX][DI] 3.MOV AX, [BX+SI] iv) **Immediate addressing mode:** 1.MOV AL, 46H 2. MOV BX, 1234H 3. MOV DX, 0040H	***Consider any two instruction, each instruction ½ M***
	b) **Ans.**	**Minimize the expression and draw logic circuit using basic gates.** **F (A,B,CD) = πm {0, 2, 4, 6, 7, 10, 11, 14, 15}** ***Note: Solution is given considering Y=F(A, B,CD) as Y= F(A, B,C,D)*** 	***4M*** ***Correct K-Map 2M***

WINTER – 2018 EXAMINATION
MODEL ANSWER

Subject: Digital Techniques & Microprocessor **Subject Code:** 22323

			***Simplification 1M*** ***Logic diagram 1M***
	c) **Ans.**	**Compare combinational and sequential circuits. Draw block diagram of sequential circuit and describe the function of each block** **PARAMETERS** | **COMBINATIONAL CIRCUIT** | **SEQUENTIAL CIRCUIT** **Definition** | The output at any instant of time depends upon the input present at that instant of time. | The output at any instance of time depends upon the present input as well as past input and output. **Need of Memory** | No memory element required in the ckt | Memory element required to stored bit **Need of clock** | Clock input not necessary | Clock input necessary **Examples** | E.g. Adders, Subtractors ,Code converters, comparators etc. | E.g. Flip flop, Shift registers, counters etc, **Applications** | Used to simplify Boolean expressions, k-map , Truth table | Used in counters & registers	**4M** ***Any 2 differences 2M***



1. Sequential logic circuits are those, whose output depends not only on the present value of the input but also on previous values of the input signal.
2. Sequential circuit can be considered as combinational circuit with feedback circuit.
3. Sequential circuit uses a memory element like flip – flops as feedback circuit in order to store past values.

Block diagram 1M

Explanation 1M

d) **i) Differentiate between RISC and CISC processor (Three point)**
ii) Compare 8086 and 80586 (Pentium)(3 points)

4M

Ans **i) Differentiate between RISC and CISC processor (Three point)**

Sr. No	PARAMETER	RISC PROCESSOPR	CISC PROCESSOR
1.	Instruction set	Few instructions	More instructions
2.	Data types	Few data types	More data types
3.	Addressing mode	Few Addressing modes	More Addressing modes
4.	Registers	Large number of general purpose registers	Small number of general purpose registers & special purpose registers.
5.	Architecture type	Load/store architecture	No load/store architecture
6.	Operation	Single- cycle	Multi-cycle
7.	Design	Hardwired control	Micro-coded
8.	Instruction Set format	Fixed length	Variable length

Any three points 2M

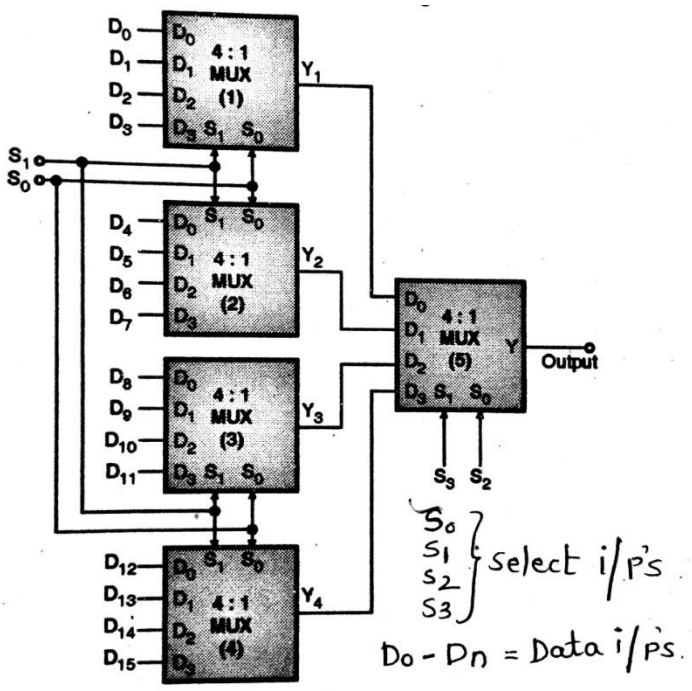
ii) Compare 8086 and 80586 (Pentium)(3 points)

SR. NO	PARAMETER	8086	80586 (Pentium)
1.	Data Bus	16 bit	64 bit
2	Address Bus	20 bit	32 bit
3	Physical memory	1 MB	4 GB
4	Register size	16 bit	32 bit
5	Voltage required	5 V	3.3 V
6	Clock type	1x	3x
7	Pipelining	Yes	Yes

Any three points 2M

e) **Ans.**

Draw 16:1 multiplexer using 4:1 multiplexers "ONLY" with proper labels.



4M

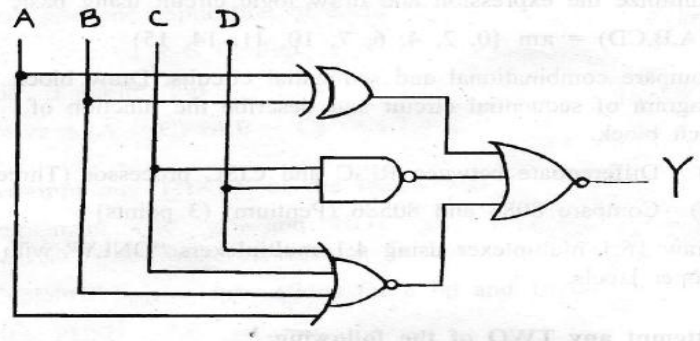
Correct Diagram 3M

Proper Labeling 1M

5		**Attempt any <u>TWO</u> of the following:**	**12**
	a)	**Write algorithm and 8086 assembly language program to find average salary of five employees of "SILICON Systems". Assume 4 digit salary of each employee. Also write output.**	**6M**
	Ans.	***Note: Any other correct logic shall be considered.***	
		ALGORITHM	***Algorithm 2M***
		1. START 2. DEFINE ARRAY SALARY OF 5 NUMBERS EACH 4 DIGIT IN DATA SEGMENT 3. DEFINE VARIABLE AVG TO STORE RESULT IN DATA SEGMENT 4. MOVE DATA IN AX 5. MOVE DATA FROM AX TO DS 6. MOVE NUM1 TO CX TO SET COUNTER 7. LOAD ADDRESS OF ARRAY SALARY TO BX 8. MOVE 0000H TO AX 9. ADD CONTENTS OF MEMORY POINTED BY BX TO AX 10. IF NO CARRY, GOTO STEP 12 11. INCREMENT DX REGISTER 12. INCREMENT BX TWICE TO POINT TO NEXT NUMBER 13. DECREMENT COUNTER CX; IF NOT ZERO GOTO STEP 9 14. DIVIDE THE SUM BY NUM1 15. STORE THE RESULT AX INTO AVG 16. END	
		PROGRAM	***Program 3M***

		DATA SEGMENT
		SALARY DW 4000H,5000H,6000H,7000H,8000H
		NUM1 DW 05H
		AVG DW ?
		DATA ENDS
		CODE SEGMENT
		ASSUME DS:DATA, CS:CODE
		START:
		MOV AX,DATA
		MOV DS,AX
		MOV CX,NUM1
		MOV BX, OFFSET SALARY
		MOV AX,0000H	

		L1: ADD AX, [BX]
		JNC NEXT
		INC DX
		NEXT: INC BX
		INC BX
		LOOP L1
		DIV NUM1
		MOV AVG,AX
		MOV AH,4CH
		INT 21H
		CODE ENDS
		END START
		
		Output
		AVG=6000H
			Output 1M

	b)	**Refer Fig No. 1 and write truth table and output "Y", write expression at output of gates. Redraw the Fig. No. 1."** Truth Table Inputs / Output A B C D Y 0 0 0 0 ⋮ ⋮ ⋮ ⋮ 1 1 1 1  **Fig No.1**	**6M**

WINTER – 2018 EXAMINATION
MODEL ANSWER

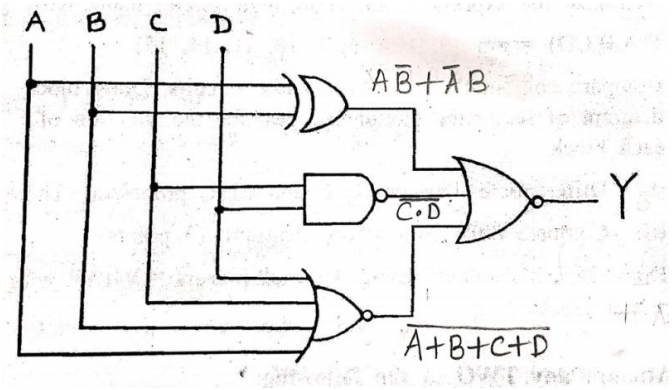
Subject: Digital Techniques & Microprocessor **Subject Code:** 22323

Ans

Truth Table

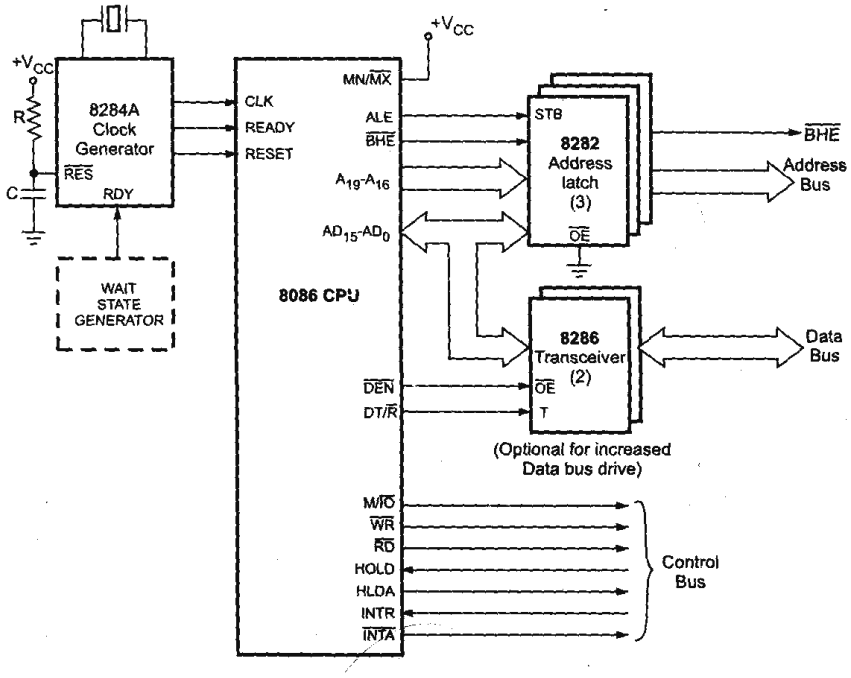
Truth Table				
Input				Output
A	B	C	D	Y
0	0	0	0	**0**
0	0	0	1	**0**
0	0	1	0	**0**
0	0	1	1	**1**
0	1	0	0	**0**
0	1	0	1	**0**
0	1	1	0	**0**
0	1	1	1	**0**
1	0	0	0	**0**
1	0	0	1	**0**
1	0	1	0	**0**
1	0	1	1	**0**
1	1	0	0	**0**
1	1	0	1	**0**
1	1	1	0	**0**
1	1	1	1	**1**

Truth Table 3M



$$Y = \overline{(A\bar{B}+\bar{A}B) + (\overline{C \cdot D}) + (\overline{A+B+C+D})}$$

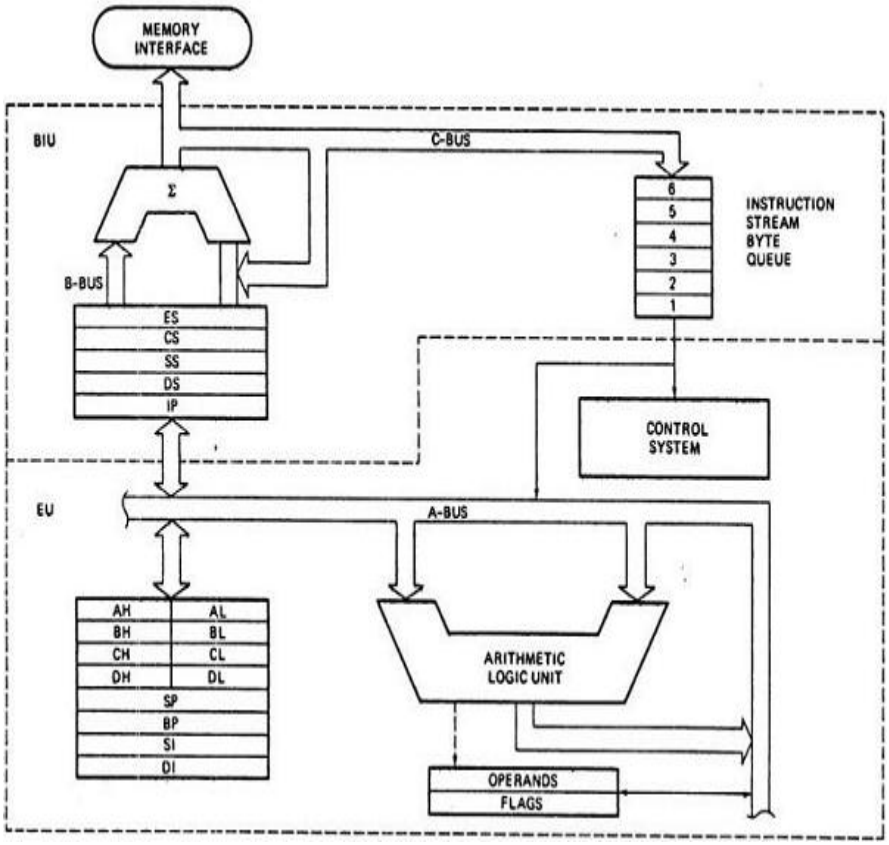
Expression at output of gates 3M

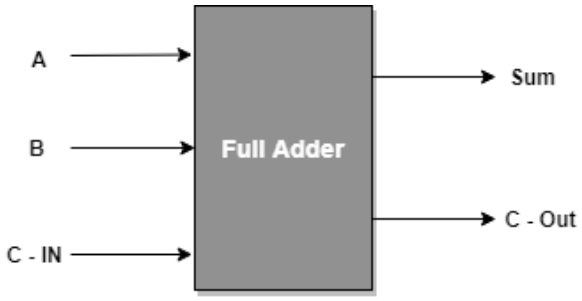
	c) **Ans.**	**Draw minimum mode configuration of 8086 and explain the function of each block.**  When MN/$\overline{MX}$ = 1 or connected to +V_{CC} as shown in the figure, the 8086 microprocessor operates in minimum mode system. • In this mode, the microprocessor chip itself gives out all the control signals. This is a single processor mode. • **The 8284 clock generator** in the system is used to generate the CLK and to synchronize some external signals with the system clock. It is also used to generate RESET and READY signal through wait state generator. • **Three 8282 address latches** are used for separating the valid address from the multiplexed address/data signals and the controlled by the ALE signal generated by 8086. • **Two 8286 Transceivers** are the bi-directional buffers. They are required to separate the valid data from the time multiplexed address/data signal. This is controlled by two signals, DEN & DT/$\overline{R}$.	**6M** ***Diagram 3M*** ***Explanation 3M***

WINTER – 2018 EXAMINATION
MODEL ANSWER

Subject: Digital Techniques & Microprocessor **Subject Code:** 22323

6		**Attempt any <u>TWO</u> of the following:**	**12**
	a)	**Draw architectural block diagram of 8086 microprocessor and describe the function of each block.**	**6M**
	Ans	***Note: Any other relevant diagram shall be considered.***	
		Internal architecture of Intel 8086:	***Explanation of blocks 3M***
		Intel 8086 is a 16 bit integer processor. It has 16-bit data bus and 20-bit address bus. The internal architecture of Intel 8086 is divided into two units, 1. Bus Interface Unit (BIU) 2. Execution Unit (EU). Bus Interface Unit (BIU) **Memory Interface:** The Bus Interface Unit (BIU) generates the 20-bit physical memory address and provides the interface with external memory (ROM/RAM). 8086 has a single memory interface. **Instruction Byte queue:** To speed up the execution, 6-bytes of instruction are fetched in advance and kept in a 6- byte Instruction Queue while other instructions are being executed in the Execution Unit (EU). **Segment registers:** There are four 16-bit segment registers, viz., the code segment (CS), the stack segment (SS), the extra segment (ES), and the data segment (DS). The processor uses CS segment for all accesses to instructions referenced by instruction pointer (IP) register. **Adder**: 8086's BIU produces the 20-bit physical memory address by combining a 16-bit segment address with a 16-bit offset address using the adder circuit. **2. Execution Unit:** **Control unit:** The instructions fetched by BIU in the instruction byte queue are decoded under the control of timing and control signals. **Arithmetic and Logic Unit (ALU) :** Execution unit has a 16 bit ALU, which performs arithmetic & logic operations. **General purpose register unit:** All general registers of the 8086 microprocessor can be used for arithmetic and logic operations. The general registers are: Accumulator register AL (8 bit), AX (AL & AH for 16 bit), Base register, Count register, Data register , Stack Pointer (SP), Base Pointer (BP), Source Index (SI), Destination Index (DI). **Flags**: is a 16-bit register containing 9 1-bit flags: Overflow Flag	

		(OF), Direction Flag (DF), Interrupt-enable Flag (IF), Single-step Flag (TF), Sign Flag (SF), Zero Flag (ZF), Auxiliary carry Flag (AF), Parity Flag (PF), Carry Flag (CF).  	***Block Diagram 3M***
	b) **Ans.**	**Design full adder using K-MAP and draw logic circuit using basic gates and write truth table.** ***Note : In logic diagram, instead of basic gates, Exclusive –OR (EXOR) gates shall be considered.*** Full Adder is the adder which adds three inputs and produces two outputs. The first two inputs are A and B and the third input is an input carry as C_{in}. The output carry is designated as C_{out} and the normal output is designated as S which is SUM. A full adder logic is designed in such a manner that can take eight inputs together to create a byte-wide adder and cascade the carry bit from one adder to the another.	**6M**



Truth Table

Input			Output	
A	B	Cin	Sum	Carry
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

Truth Table 2M

Based on the truth table, the Boolean functions for Sum (S) and Carry – out (C_{out}) can be derived using K – Map.

For Sum S :

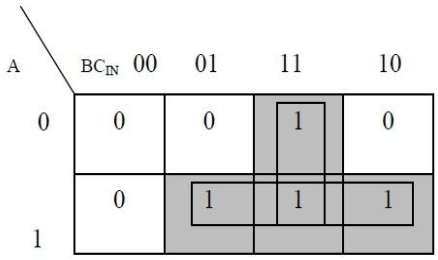
A \ BC_{IN}	00	01	11	10
0	0	1	0	1
1	1	0	1	0

The simplified equation for sum is

S = $\bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C} + ABC_{in}$

= $A \oplus B \oplus Cin$

For Carry – out (C_{out}) :

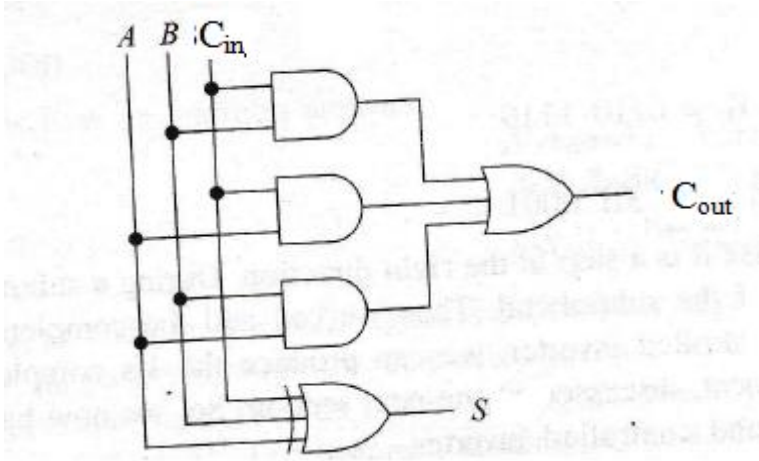


K map design 2M

The simplified equation for $C_{out}$is

$C_{out} = AB + AC_{in} + BC_{in}$

Logic Circuit Diagram



Logic diagram 2M

	c)
	
	Ans
		Write an assembly language program to find the largest number from an array of a 10 numbers. Assume suitable data.
		Note: Either 8bit or 16bit data shall be considered.
		DATA SEGMENT
		ARR DB 1,4,2,3,9,8,6,7,5,10
		LN DW 10
		L DB ?
		DATA ENDS
		CODE SEGMENT
		ASSUME DS:DATA, CS:CODE
		START:
		MOV AX,DATA
		MOV DS,AX
		LEA SI,ARR
		MOV AL,ARR[SI]
		MOV L,AL
		MOV CX,LN
		REPEAT: MOV AL,ARR[SI]
		CMP L,AL
		JG NOCHANGE (or JNC NOCHANGE)
		MOV L,AL
		NOCHANGE: INC SI
		LOOP REPEAT
		MOV AH,4CH
		INT 21H
		CODE ENDS
		END START
			6M
			
			Correct logic 3M
			
			Correct Instructions 3M