18BECS303 **Digital Electronics**

L	T	P	C
3	0	0	3

Course Outcomes:

At the end of this course, students will demonstrate the ability to

- Understand working of logic families and logic gates.
- Design and implement Combinational and Sequential logic circuits.
- Understand the process of Analog to Digital conversion and Digital to Analog conversion.
- Be able to use PLDs to implement the given logical problem.

UNIT 1:Fundamentals of Digital Systems and logic families (9Hours)

Digital signals, digital circuits, AND, OR, NOT, NAND, NOR and Exclusive-OR operations, Boolean algebra, examples of IC gates, number systems-binary, signed binary, octal hexadecimal number, binary arithmetic, one's and two's complements arithmetic, codes, error detecting and correcting codes, characteristics of digital ICs, digital logic families, TTL, Schottky TTL and CMOS logic, interfacing CMOS and TTL, Tri-state logic.

UNIT 2: Combinational DigitalCircuits (9Hours)

Standard representation for logic functions, K-map representation, simplification Of logic functions using K-map, minimization of logical functions. Don't care conditions, Multiplexer,De-Multiplexer/Decoders, Adders, Subtractors, BCD arithmetic, carry look ahead adder,serialadder, ALU, elementary ALU design, popular MSI chips, digital comparator,paritychecker/generator, code converters, priority encoders, decoders/drivers for display devices-M method of function realization.

UNIT 3: Sequential circuits and systems (9Hours)

A 1-bit memory, the circuit properties of Bistable latch, the clocked SR flip flop, J-K-T and D types flipflops, applications of flipflops, shift registers, applications of shift registers, serial to parallel converter, parallel to serial converter, ring counter, sequence generator, ripple(Asynchronous) counters, synchronous counters, counters design using flip flops, special counter IC's, asynchronous sequential counters, applications of counters.

UNIT 4: A/D and D/A Converters (9Hours)

Digital to analog converters: weighted resistor/converter, R-2R Ladder D/Aconverter,specifications for D/A converters, examples of D/A converter ICs, sample and hold circuit, analog to digital converters: quantization and encoding, parallel comparator A/Dconverter,successive approximation A/D converter, counting A/D converter, dual slope A/D converter/Converter using Voltage to frequency and voltage to time conversion, specifications of A/D converters, example of A/D converter ICs.

UNIT 5: Semiconductor memories and Programmable logic devices. (9Hours)

Memory organization and operation, expanding memory size, classification and characteristics of memories, sequential memory, read only memory (ROM), read and write memory(RAM), content addressable memory (CAM), charge de coupled device memory (CCD), commonly used memory chips, ROM as a PLD, Programmable logic array, Programmable array logic, complex Programmable logic devices (CPLDS), Field Programmable Gate Array (FPGA).

Text/References:

1. R. P. Jain, "Modern Digital Electronics", McGraw Hill Education, 2009.
2. M. M. Mano, "Digital logic and Computer design", Pearson Education India, 2016.
3. A. Kumar, "Fundamentals of Digital Circuits", Prentice Hall India, 2016.

KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed to be University Established Under Section 3 of UGC Act 1956)

Pollachi Main Road, Eachanari Post,

Coimbatore – 641 021

FACULTY OF ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

LECTURE PLAN

NAME OF THE STAFF : Mrs.A.MOHANARATHINAM

DESIGNATION : ASSISTANT PROFESSOR

CLASS : B.E-II YEAR CSE

SUBJECT : DIGITAL ELECTRONICS

SUBJECT CODE : 18BECS343

S.No	TOPICS TO BE COVERED	TIME DURATION	SUPPORTING MATERIALS	TEACHING AIDS
	UNIT-I BASIC MEASUREMENT CONCEPTS			
1	Digital signals, digital circuits,	01	T1- Page.no : 1-3	BB
2	AND, OR, NOT, NAND, NOR and Exclusive-OR operations		T1- Page.no :4-9	BB
3	Boolean algebra, examples of IC gates.	01	T1- Page.no :9-12	BB
4	number systems-binary, signed binary, octal hexadecimal number	01	T1 Page.no : 21-28, : 34-42	BB
5	binary arithmetic, one's and two's complements arithmetic	01	T1 Page.no. 30-33	BB
6	codes, error detecting and correcting codes	01	T1 page.no. 42-53	BB
7	characteristics of digital ICs, digital logic families,	01	T1 page.no.90-93	BB,PPT
8	TTL, Schottky TTL and CMOS logic	01	T1 Page.no. 105-108, 119-120	BB,PPT
9	interfacing CMOS and TTL, Tri-state logic	01	T1 page.no.123-125, 126-128	BB
Total Lecture Hours		**09**		
Total Hours		**09**		

	UNIT-II BASIC ELECTRONIC MEASUREMENTS			
10	Standard representation for logic functions, K-map representation	01	T1 Page.no 138-150	BB
11	simplification Of logic functions using K-map,	01	T1 Page.no 150-155	BB
12	minimization of logical functions. Don't care conditions,	01	T1 Page.no 155-160, 161-170	BB
13	Multiplexer, De-Multiplexer / Decoders	01	T1 Page.no 191-200,	BB

14	Adders, Subtractors	01	T1 Page.no 201-205	BB
15	BCD arithmetic, carry look ahead adder, serial adder,	01	T1 Page.no 205-206,202-203	PPT , BB
16	ALU, elementary ALU design, popular MSI chips	01	T1 Page.no 209-211	PPT , BB
17	digital comparator, parity checker/ generator, code converters	01	T1 Page.no 212-225	BB
18	Priority encoders, decoders/drivers for display devices-M method of function realization	01	T1 Page.no 226-232	BB
Total Lecture Hours		**09**		
Total Hours		**09**		

UNIT-III SIGNAL GENERATORS AND ANALYZERS				
19	A 1-bit memory, the circuit properties of Bistable latch, the clocked SR flip flop	01	T1 Page.no.237-242	BB, PPT
20	J-K, T and D types flipflops, applications of flipflops	01	T1 Page.no.243-258	BB, PPT
21	shift registers, applications of shift registers	01	T1 Page.no.267-272	BB, PPT
22	serial to parallel converter, parallel to serial converter	01	T1 Page.no.272-273	BB
23	ring counter, sequence generator	01	T1 Page.no.273-276	BB
24	ripple(Asynchronous) counters, synchronous counters,	01	T1 Page.no.276-287	BB
25	counters design using flip flops	01	T1 Page.no.288-292	BB, PPT
26	special counter IC's	01	T1 Page.no.282-300	BB
27	asynchronous sequential counters, applications of counters	01	T1 Page.no.306-307	BB
Total Lecture Hours		**09**		
Total Hours		**09**		

UNIT-IV DIGITAL INSTRUMENTS				
28	Digital to analog converters: weighted resistor/converter, R-2R Ladder D/A converter.	01	T1 Page.no.366-376	BB
29	specifications for D/A converters, examples of D/A converter ICs,	01	T1 Page.no 376-380	BB
30	sample and hold circuit, analog to digital converters: quantization and encoding	01	T1 Page.no 382-385	BB,PPT
31	parallel comparator A/D converter	01	T1 Page.no 385	BB,PPT
32	successive approximation A/D converter	01	T1 Page.no 385	BB,PPT
33	counting A/D converter, dual slope A/D converter	01	T1 Page.no 389-392	BB
34	Converter using Voltage to frequency	01	T1 Page.no 392	BB

35	Converter using voltage to time conversion	01	T1 Page.no 393	BB
36	specifications of A/D converters, example of A/D converter ICs	01	T1 Page.no 395-396	BB,PPT
Total Lecture Hours		**09**		
Total Hours		**09**		

UNIT-V DATA ACQUISITION SYSTEMS AND FIBER OPTIC MEASUREMENTS				
37	Memory organization and operation, expanding memory size, , ,	01	T1 Page.no 401-408	BB
38	classification and characteristics of memories, sequential memory	01	T1 Page.no 410	BB
39	read only memory (ROM), read and write memory(RAM),	01	T1 Page.no 419-433	BB
40	content addressable memory (CAM),	01	T1 Page.no 433-440	BB
41	charge de coupled device memory (CCD),	01	T1 Page.no 440-442	BB,PPT
42	commonly used memory chips, ROM as a PLD	01	T1 Page.no 426-434,451	BB
43	Programmable logic array, Programmable array logic,	01	T1 Page.no 451-465, 465-472	BB
44	complex Programmable logic devices (CPLDS),	01	T1 Page.no,478-483	BB
45	Field Programmable Gate Array (FPGA).	01	T1 Page.no,483-490	BB,PPT
Total Lecture Hours		**09**		
Total Hours		**09**		

Total No of Lecture Hours Planned: 45 Hrs

Total No of Hours Planned : 45 Hours

Text/References:

1. R. P. Jain, "Modern Digital Electronics", McGraw Hill Education, 2009.
2. M. M. Mano, "Digital logic and Computer design", Pearson Education India, 2016.
3. A. Kumar, "Fundamentals of Digital Circuits", Prentice Hall India, 2016.

STAFF IN-CHARGE **HOD/ECE**

S.No	Questions	opt1	opt2	opt3	opt4	Answer
1	The radix of a decimal number is____	2	8	10	16	**10**
2	The radix of an octal number is____	2	8	10	16	**8**
3	The base of a binary number is_____	2	8	10	16	**2**
4	The base of a hexa decimal number is_____	2	8	10	16	**16**
5	Thr single digit value in digital system is known as_____	Bit	Byte	Nibble	Word	**Bit**
6	A 4-bit value is known as ____ in a digital systems	Bit	Byte	Nibble	Word	**Nibble**
7	A 8-bit value is known as ____ in a digital systems	Bit	Byte	Nibble	Word	**Byte**
8	A 16-bit value is known as ____ in a digital systems	Bit	Byte	Nibble	Word	**Word**
9	The binary value of hexadecimal number "A"is___	1010	1111	1110	1001	**1010**
10	The decimal value of a binary number 0101 is___	8	7	9	5	**5**
11	The octal value of a binary number 0111 is____	4	5	6	7	**7**
12	The hexadecimal value of a decimal number 15 is____	1010	1110	1011	1111	**1111**
13	The decimal value of a hexadecimal number "B" is_____	10	11	12	13	**11**
14	The binary value of a decimal number "3"is___	11	101	1101	111	**11**
15	The octal value of a decimal number 444 is____	534	356	636	674	**674**
16	The hexadecimal value of a binary number 1110 is___	A	D	E	F	**E**
17	The decimal value of a hexadecimal number "D" is___	11	12	13	14	**13**
18	The decimal value of a binary numbe 1001 is____	9	10	8	7	**9**
19	The decimal value of an octal number 237 is____	160	159	158	157	**159**
20	The hexadecimal value of a decimal nymber115 is___	AF	4B	84	73	**73**
21	The binary value of a hexadecimal number 2D5 is____	1100 1101 0101	1011010101	1001 1010 0111	1010 1001 0001	**1011010101**
22	The octal value of a hexadecimal number 47 is_____	66	215	107	532	**107**
23	The hexadecimal value of an octal number 32 is_____	5A	3A	4A	1A	**1A**
24	The decimal value of a binary number 101111 is _____	47	74	64	2F	**47**
25	The octal value of a binary number 1001010 is _____	111	112	113	114	**112**
26	The octal value of a decimal number 0.23 is___	0.1432	0.732	0.1656	0.1414	**0.1656**
27	The decimal value of a binary number 10110 is_____	22	24	23	25	**22**
28	The binary value of an octal number 27 is____	100110	110101	10111	111101	**10111**

29	The decimal value of a hexadecimal number E9 is_____	233	234	255	258	**233**
30	The hexadecimal value of binary number 1101110 is____	4E	3D	6E	5D	**6E**
31	The 1's complement value of 11011 is____	1101	11011	oo100	11100	**100**
32	The 1's complement value of 101101 is____	o10010	oo1110	111010	101101	**10010**
33	The 2's complement value of 11011 is_____	11010	11101	10101	101	**101**
34	The 2's complement value of ooo1o is____	11101	11110	10101	11011	**11110**
35	The 1's complement value of 1010 is____	1010	1100	101	1011	**101**
36	The signed 1's complement value os -12 is____	11001101	10101100	11110011	1101010	**11110011**
37	The signed 2's complement value of -93is____	10100011	10010101	10101010	11011011	**10100011**
38	Expansion of BCD is _____	Binary Coder Decoder	Binary Complement Decimal	Binary Coded Decimal	Binary Comparator Decoder	**Binary Coded Decimal**
39	8421 code is a type of___ code	Excess 3	BCD	Gray	ASCII	**BCD**
40	What is the weight of binary digit 1001 using 8421 code____	6	9	13	15	**9**
41	Convert the decimal number 170 to BCD___	1101 0111 0000	1010 0101 0000	0001 0111 0000	1110 1000 111	**0001 0111 0000**
42	The decimal number for the BCD number oo1101010001 is____	255	351	127	194	**351**
43	The expansion of ASCII is _____	American Standard Cod for information Interchange	Alphabet standard code for information Interchange	Arithimetic symbol code for Information Interchange	Arthimetic symbol code for instruments information	**American Standard Cod for information Interchange**
44	The sum of two binary numbers (100)2 and (10)2is____	111	101	110	10	**110**
45	The sum of two binary numbers (11)2 and (10)2is____	101	111	110	10	**101**
46	The binary multiplication of 2 binary numbers 11 and 11 is____	1001	1000	1110	1111	**1001**
47	The quatent of a binary number 110,when divided by 10 is____	10	1	10	11	**11**
48	The gray code for binary number oo10 is___	1001	1101	11	110	**11**
49	The binary code for the gray code 1110 is____	1001	1101	1011	1111	**1011**
50	The gray code for binary number o100 is___	110	1100	1010	1101	**110**
51	The ASCII code is ___ bit binary code	4	7	8	16	**7**
52	The ASCII code consists if ___ characters and symbols	130	150	128	144	**128**
53	The excess 3 code for a BCD code ooo1 is_____	100	1010	1	1001	**100**
54	The BCD value of a decimal number 9 is____	1110	1010	1001	1011	**1001**
55	The excess 3 code for a decimal number 3 is_____	110	101	1100	1101	**110**
56	The parity bit is used to detct ____bit errors	Double	Single	Four	Eight	**Single**
57	The ____method is used to detect double bit errors	Parity bit	Hamming code	Hollerith code	Check Sum	**Check Sum**
58	A minimum distance of atleast _____ bit requird for single error detection	1	2	3	4	**2**
59	The _____ provides methodical way by adding one or more parity bits to a data charcter to detect and correct errors	Hollerith code	Hamming code	Check Sum	parity generator	**Hamming code**
60	If total number of 1's is even is a binary number ,then it is called_____ error detection	Odd parity	Even parity	Hollerith code	Hamming code	**Even parity**

S.No	Questions	opt1
1	Which of these sets of logic gates are designated as universal gates?	NOR, NAND.
2	The output of an AND gate is LOW _____.	when any input is LOW
3	The output of an OR gate is LOW when _____.	all inputs are LOW
4	When used with an IC, what does the term "QUAD" indicate?	4 circuits
5	The Boolean expression for a 3-input AND gate is _____.	X = AB
6	What is the Boolean expression for a three-input OR gate?	X = AB
7	Which of the following expressions is in the sum-of-products (SOP) form?	AB + CD
8	The systematic reduction of logic circuits is accomplished by:	symbolic reduction
9	The output of an AND gate with three inputs, A, B, and C, is HIGH when _____.	A = 1, B = 1, C = 0
10	The minimal form of the Boolean Expression F=(x+y)+x'y' is	1
11	The Dual of a Boolean Expression is A+B. The Expression is	A.B
12	The Complement of complement of A'.B + A.B' is	AB+A'B'
13	(A+B.C)(A+B'+C') would simplify to	(A+B'+C')
14	The Equality (A+B+C)'=A'.B'.C' is known as	Involution law
15	A NOR Gate is	Inversion followed by an OR Gate
16	A NAND Gate is	Inversion followed by an AND Gate
17	The MIN Term designator of the term A	4
18	Two Input Exclusive NOR Gate gives high output	When one Input is High & the other is Low
19	. The Output of a two Input OR Gate is high	Only if both the Inputs are High
20	The Output of a two Input AND Gate is high	Only if both the Inputs are High
21	The Output of a two Input NAND Gate is high	Only if both the Inputs are High
22	The Output of a two Input NOR Gate is high	Only if both the Inputs are High
23	If an Input A is given one inverter, the output will be	1/A
24	If a 3-input NOR gate has eight input possibilities, how many of those possibilities will result in a HIGH output?	1

25	If a signal passing through a gate is inhibited by sending a LOW into one of the inputs, and the output is HIGH, the gate is a(n):	AND
26	Which of the following logical operations is represented by the + sign in Boolean algebra?	inversion
27	Output will be a LOW for any case when one or more inputs are zero for a(n):	OR gate
28	The output of a NOR gate is HIGH if _____.	all inputs are HIGH
29	A NAND circuit with positive logic will operate as :	NOR with negative logic
30	Which of the following ICs has only one NAND gate:	7400
31	OR operation is:	X + XY
32	What are the pin numbers of the outputs of the gates in a 7432 IC?	3, 6, 10, and 13
33	The output of a NOT gate is HIGH when _____.	the input is LOW
34	If the input to a NOT gate is A and the output is X, then _____.	X = A
35	How many inputs of a four-input AND gate must be HIGH in order for the output of the logic gate to go HIGH?	any one of the inputs
36	What is the no. of OR IC. :	7402
37	What is the no. of AND IC.	7408
38	What is the no. of NOR IC. :	7402
39	What is the no. of NAND IC. :	7402
40	What is the no. of NOT IC. :	7402
41	What is the no. of EX-OR IC. :	7402
42	Which of the following is Boolean eq. of EX-OR gate:	A+B
43	Which of the following gates has the exact inverse output of the OR gate for all possible input combinations?	AND
44	The output of an exclusive-OR gate is HIGH if _____.	all inputs are LOW
45	Determine the values of A, B, C, and D that make the sum term A'+B+C'+D equal to zero.	A = 1, B = 0, C = 0, D = 0
46	Which statement below best describes a Karnaugh map?	Variable complements can be eliminated by using Karnaugh maps.
47	Which of the examples below expresses the commutative law of multiplication?	A • (B • C) = (A • B) • C
48	The observation that a bubbled input OR gate is interchangeable with a bubbled output AND gate is referred to as:	a Karnaugh map

49	The commutative law of addition and multiplication indicates that:	the way we OR or AND two variables is unimportant because the result is the same
50	Which of the examples below expresses the distributive law of Boolean algebra?	A • (B + C) = (A • B) + (A • C)
51	A single transistor can be used to build which of the following digital logic gates?	AND
52	Exclusive-OR (XOR) logic gates can be constructed from what other logic gates?	AND gates, OR gates, and NOT gates
53	The logic gate that will have HIGH or “1” at its output when any one of its inputs is HIGH is a(n):	AND
54	How many NAND circuits are contained in a 7400 NAND IC?	3
55	How many truth table entries are necessary for a four-input circuit?	12
56	A NAND gate has:	LOW inputs and a HIGH output
57	The basic logic gate whose output is the complement of the input is the:	INVERTER gate
58	A Karnaugh map with 4 variables has :	2 cells
59	An AND gate with schematic "bubbles" on its inputs performs the same function as a(n)_____ gate.	AND
60	NOT gate can also Called as	INVERTER gate

opt2	opt3
XOR, NOR, NAND. when all inputs are HIGH	OR, NOT, AND. when any input is HIGH
any input is LOW	any input is HIGH
2 circuits	8 circuits
X = ABC	X = A + B + C
X = ABC	X = A + B + C
AB(CD)	(A + B)(C + D)
using Boolean algebra	TTL logic
A = 0, B = 0, C = 0	A = 1, B = 1, C = 1
0	x+y
A'.B'	A'+B'
(A'+B)(A+B') (A+B.C) Absorption Law	A'B+AB' A.B. Complementation Law
OR Gate followed by an Inverter	NOT Gate followed by an OR Gate
AND Gate followed by an Inverter 15	AND Gate followed by an OR Gate 11
Only When both the Inputs are Low	When Both the Input are same
Only if both the Inputs are Low	If atleast one of the Inputs is high
Only if both the Inputs are Low	If atleast one of the Inputs is high
. Only if both the Inputs are Low	. If at least one of the Inputs is Low
Only if both the Inputs are Low	If at least one of the Inputs is High
1	A'
2	7

NAND	NOR
AND	OR
NOT gate any input is HIGH	AND gate any input is LOW
AND with negative logic	OR with negative logic input
7420 XY	7430 X+Y
1, 4, 10, and 13	3, 6, 8, and 11
the input is HIGH	power is applied to the gate's IC
X=A'	X = 0
any two of the inputs 7486 7486 7486 7404 7486 7486	any three of the inputs 7432 7432 7447 7400 7404 7447
A+B '	AB
NAND	NOR
all inputs are HIGH	the inputs are unequal
A = 1, B = 0, C = 1, D = 0	A = 0, B = 1, C = 0, D = 0
It is simply a rearranged truth table.	The Karnaugh map eliminates the need for using NAND and NOR gates.
A + B = B + A	A • B = B • A
the commutative law of addition	DeMorgan's second theorem

we can group variables in an AND or in an OR any way we want	the factoring of Boolean expressions requires the multiplication of product terms that contain like variables
$A \bullet (B \bullet C) = (A \bullet B) + C$	$A + (B + C) = (A \bullet B) + (A \bullet C)$
NAND	NOR
OR gates only	AND gates and NOT gates
NAND	OR
2	1
4	8
HIGH inputs and a HIGH output	Any One LOW inputs and a LOW output
comparator	OR gate
4 cells	8 cells
NAND	OR
comparator	OR gate

opt4

NOR, NAND, XNOR.

all the time

all inputs are HIGH

6 circuits

X = AB + C

X = AB + C

(A)B(CD)

using a truth table

A = 1, B = 0, C = 1

x’y’

A+B

None of the above

none of the above

DeMorgan’s Law

NAND Gate followed by an OR Gate

None of These

None of These

Only when both the Inputs are High

If atleast one of the Inputs is Low

. If atleast one of the Inputs is Low

. None of these

None of these

A

8

OR

complementation

NOR gate
all inputs are LOW

AND with positive logic output

7447
(X+Y) (X+Y')

1, 4, 8, and 11

power is removed from the gate's IC

X=1

all four inputs
7404
7404
7492
7492
7492
7492

A' B + A B'

NOT

any input is LOW

A = 1, B = 0, C = 1, D = 1

A Karnaugh map can be used to replace Boolean rules.

A • B = B + A

the associative law of multiplication

an expression can be expanded by multiplying term by term just the same as in ordinary algebra

(A + B) + C = A + (B + C)

NOT

OR gates and NOT gates

NOT

4

16
all the time

AND gate
16 cells

NOT
AND gate

Answer

NOR, NAND.
when any input is LOW

all inputs are LOW

4 circuits

X = ABC

X = A + B + C

AB + CD

using Boolean algebra

A = 1, B = 1, C = 1

1

A.B

A’B+AB’
A.B.
DeMorgan’s Law

OR Gate followed by an Inverter

AND Gate followed by an Inverter
11

When Both the Input are same

If atleast one of the Inputs is high

Only if both the Inputs are High

. If at least one of the Inputs is Low

If at least one of the Inputs is High

A’

1

NAND

OR

AND gate
all inputs are LOW

AND with negative logic

7400
X+Y

3, 6, 8, and 11

the input is LOW

X=A'

all four inputs
7432
7408
7402
7400
7404
7486

A' B + A B'

NOR

the inputs are unequal

A = 1, B = 0, C = 1, D = 0

A Karnaugh map can be used to replace Boolean rules.

A • B = B • A

DeMorgan's second theorem

the way we OR or AND two variables is unimportant because the result is the same

$A \bullet (B + C) = (A \bullet B) + (A \bullet C)$

NOT

AND gates, OR gates, and NOT gates

OR

4

16

Any One LOW inputs and a LOW output

INVERTER gate

16 cells

NAND

INVERTER gate

S.No	Questions	opt1
1	T Flip flop is used as	Transfer data Circuit
2	Following Flip Flop is used to eliminate race around problem	R-S Flip Flop
3	In T Flip Flop the output frequency is	Same as the input frequency
4	Shifting a binary data to the left by one bit position using Shift Left register, amount to	Addition of 2
5	If a Counter is connected using Six Flip Flops, then the maximum number of states that the counter can count are	6
6	The minimum number of Flip flops required for a Mod-10 ripple Counter are	4
7	A Mod-5 Synchronous Counter is designed using J-k Flip flop, the number of counts skipped by it will be	2
8	A BCD Counter has	3 Distinct states
9	The Output Frequency of a Mod-16 Counter, Checked from a 10 MHZ Clock input signal is	10 KHZ
10	The Output Frequency of a Decade Counter, which is clocked from a 50 KHZ signal is	5 KHZ
11	The Table which shows the necessary levels at J and K inputs to produce every possible flip-flop state transition is called	Truth Table of J-K Flip-flop
12	The number of Flip-flops required for a Mod-16 Ring Counter is	4 Flip-flops
13	The Counter which requires maximum number of Flip-flop for a given Mod Number is	ripple counter
14	The type of register, in which we have access only to left most and right most Flip-flops is	Shift Left and Shift Right Register
15	The Maximum Frequency to which a mod-16 ripple counters using four J-K Flip Flops with Propagation Delay time of 50 ns is	4 MHZ
16	The maximum number of Flip-Flops required to construct a Mod-64 (divide by 64) ripple counter are	4 Flip-flops
17	The Maximum modulo number that can be obtained by a ripple counter using five flip-flops is	16
18	The type of register, in which data is entered into it only one bit at a time, but has all data bits available as output, is	Parallel in / Parallel out Register
19	Race Condition occurs in	Synchronous circuit
20	The Minimum number of Flip-flops needed to construct a BCD Decade counter is	4

21	A shift Counter comprising five Flip-flops with an inverse feedback from the output of the MSB Flip-flop to the input of the LSB Flip-flop is a	Divide-by- 32 Counters
22	A four bit binary UP / Down Counter is initially reset to 0000. The UP / DOWN mode select terminal designated as U'/ D on the pin connection diagram of the IC is tied to logic HIGH level. What would be the counters output state at the end of the first clock pulse ?	1
23	In any Asynchronous Counter,	all flip-flops change state at the same time
24	A Five bit Counter	has a modulus of 5
25	A Counter that has a modulus of 64 should use a minimum of	6 Flip-flops
26	A MOD-32 Binary synchronous counter would require	6 Flip-flops and 3 AND gates
27	In what type of shift register do we have access to only the left most and right most flip-flops?	Parallel in / Parallel out Shift Register
28	The Minimum number of Flip-flops required to construct a MOD-10 Johnson Counter and a MOD-5 ring Counter, respectively, is	10, 5
29	The decoding glitches as far more likely to occur in the case of	Ripple counters
30	A Five-bit Johnson Counter in Cascade with a five-bit ring counter produces a frequency divider of	25
31	⇔ A four-bit Pre-Settable down Counter initially loaded with 0101 will divide the input clock frequency by	16
32	⇔Shifting a binary data to the right by one bit position using Shift Right register, amount to	Addition of 2
33	If a Counter is connected using Five Flip Flops, then the maximum number of states that the counter can count are	6
34	⇔If a Counter is connected using Seven Flip Flops, then the maximum number of states that the counter can count are	6
35	⇔If a Counter is connected using Ten Flip Flops, then the maximum number of states that the counter can count are	6
36	The minimum number of Flip flops required for a Mod-9 ripple Counter is	2

37	The minimum number of Flip flops required for a Mod-12 ripple Counter is	2
38	The minimum number of Flip flops required for a Mod-13 ripple Counter is	2
39	A Mod-6 Synchronous Counter is designed using J-K Flip flop, the number of counts skipped by it will be	2
40	The Output Frequency of a Mod-8 Counter, Checked from a 50 MHZ Clock input signal is	10 KHZ
41	The Output Frequency of a Decade Counter, which is clocked from a 10 KHZ signal, is	5 KHZ
42	The Table which shows the necessary levels at R and S inputs to produce every possible flip-flop state transition is called	Truth Table of R-S Flip-flop
43	The Table which shows the necessary levels at T inputs to produce every possible flip-flop state transition is called	Truth Table of T Flip-flop
44	The Table which shows the necessary levels at D inputs to produce every possible flip-flop state transition is called	Truth Table of D Flip-flop
45	The number of Flip-flops required for a Mod-8 Ring Counter is	4 Flip-flops
46	The number of Flip-flops required for a Mod-10 Ring Counter is	4 Flip-flops
47	The number of Flip-flops required for a Mod-16 Johnson Counter is	4 Flip-flops
48	The number of Flip-flops required for a Mod-10 Johnson Counter is	4 Flip-flops
49	The number of Flip-flops required for a Mod-8 Johnson Counter is	4 Flip-flops
50	The maximum number of Flip-Flops required to construct a Mod-256 (divide by 256) ripple counter are	4 Flip-flops
51	A Counter that has a modulus of 32 should use a minimum of	5 D Flip-flops
52	A Counter that has a modulus of 128 should use a minimum of	6 D Flip-flops
53	The Minimum number of Flip-flops required to construct a MOD-8 Johnson Counter and a MOD-4 ring Counter, respectively, is	10, 5
54	The Minimum number of Flip-flops required to construct a MOD-20 Johnson Counter and a MOD-10 ring Counter, respectively, is	10, 5

55	A Four- bit Johnson Counter in Cascade with a four-bit ring counter produces a frequency divider of	25
56	A Four bit Counter	has a modulus of 4
57	The Maximum modulo number that can be obtained by a ripple counter using six flip-flops is	16
58	The Maximum modulo number that can be obtained by a ripple counter using ten flip-flops is	16
59	Type of register, in which data is all data bits are entered simultaneously but only one data bit can be read at a time	Parallel in / Parallel out Register
60	A four bit binary UP / Down Counter is initially reset to 0000. The UP / DOWN mode select terminal designated as U'/ D on the pin connection diagram of the IC is tied to logic LOW level. What would be the counters output state at the end of the first clock pulse?	1

opt2	opt3
Toggle Switch	Time Delay Switch
Master Slave J-K Flip Flop	J-K Flip Flop
One-half its input frequency	Double of its input frequency
Multiplication by 2	Subtraction of 2
8	256
2	10
3	5
8 Distinct states	10 Distinct states
26 KHZ	160 KHZ
50 KHZ	500 KHZ
Excitation Table of J-K Flip-flop	Excitation Table of Mod-N Counter using J-K Flip-flop
8 Flip-flops	10 Flip-flops
BCD Counter	ring Counter
Serial in / Serial out Register	Parallel in / Serial out Register
5 MHZ	3.2 MHZ
6 Flip-flops	16 Flip-flops
32	5
Serial in / Serial out Register	Parallel in / Serial out Register
Asynchronous circuit	Combinational circuit
3	10

Divide-by- 10 Counters	Divide-by- 5 Counters
1000	1111
Only D Flip-flops are used	the Counter responds to negative going clock edges
cannot have a modulus greater than 32	has a modulus that is less than or equal to 32
6 J-K type Flip-flops	6D Flip-flops
5 Flip-flops	5 Flip-flops and 3 AND gates
Serial in / Serial out Shift Register	Parallel in / Serial out Shift Register
5, 10	5, 5
Parallel Counters	Johnson Counters
10	50
5	11
Multiplication by 2	Subtraction of 2
32	256
32	128
1024	256
4	10

4	11
10	4
3	5
26 KHZ	160 KHZ
1 KHZ	50 KHZ
Excitation Table of R-S Flip-flop	Excitation Table of Mod-N Counter using R-S Flip-flop
Excitation Table of T Flip-flop	Excitation Table of Mod-N Counter using T Flip-flop
Excitation Table of D Flip-flop	Excitation Table of Mod-N counters using D Flip-flop
8 Flip-flops	10 Flip-flops
Flip-flops	10 Flip-flops
8 Flip-flops	10 Flip-flops
8 Flip-flops	5 Flip-flops
8 Flip-flops	10 Flip-flops
6 Flip-flops	16 Flip-flops
6 J-K type Flip-flops	6D Flip-flops
6 J-K type Flip-flops	7 D Flip-flops
5, 10	4, 4
5, 10	5, 5

10	32
cannot have a modulus greater than 16	has a modulus that is less than or equal to 16
32	64
32	5
Serial in / Serial out Register	Parallel in / Serial out Register
1000	1111

opt4	Answer
None of the above	**Toggle Switch**
D Flip Flop	**Master Slave J-K Flip Flop**
None of the above	**One-half its input frequency**
Division by 2	**Multiplication by 2**
64	**64**
5	**4**
0	**3**
16 Distinct states	**10 Distinct states**
625 KHZ	**625 KHZ**
530 KHZ	**5 KHZ**
None of these	**Excitation Table of J-K Flip-flop**
16 Flip-flops	**16 Flip-flops**
Programmable Counter	**ring Counter**
Serial in / Parallel out Register	**Shift Left and Shift Right Register**
320 MHZ	**5 MHZ**
64 Flip-flops	**6 Flip-flops**
31	**32**
Serial in / Parallel out Register	**Serial in / Parallel out Register**
all the digital Circuits	**Asynchronous circuit**
11	**4**

Six bit Shift Register	**Divide-by- 10 Counters**
0	**1111**
Each flip-flop output serves as clock input to the next flip-flop	**Each flip-flop output serves as clock input to the next flip-flop**
Both B and C are true	**Both B and C are true**
64 Flip-flops	**6 Flip-flops**
8 Flip-flops	**5 Flip-flops and 3 AND gates**
Serial in / Parallel out Shift Register	**Serial in / Serial out Shift Register**
10, 10	**5, 5**
Ring Counters	**Ripple counters**
15	**50**
10	**5**
Division by 2	**Division by 2**
64	**32**
64	**128**
64	**1024**
5	**4**

4	**4**
5	**4**
0	**2**
625 KHZ	**625 KHZ**
530 KHZ	**1 KHZ**
None of these	**Excitation Table of R-S Flip-flop**
None of these	**Excitation Table of T Flip-flop**
None of these	**Excitation Table of D Flip-flop**
16 Flip-flops	**8 Flip-flops**
16 Flip-flops	**10 Flip-flops**
16 Flip-flops	**8 Flip-flops**
16 Flip-flops	**5 Flip-flops**
16 Flip-flops	**4 Flip-flops**
8 Flip-flops	**8 Flip-flops**
64 Flip-flops	**5 D Flip-flops**
64 Flip-flops	**7 D Flip-flops**
10, 10	**4, 4**
10, 10	**10, 10**

15	**32**
Both B and C are true	**Both B and C are true**
31	**64**
1024	**1024**
Serial in / Parallel out Register	**Parallel in / Serial out Register**
0	**1**

S.No	Questions
1	The stable condition in transition table is given by the expression
2	The SR latch consists of
3	Table that is not a part of asynchronous analysis procedure
4	The making of transition table consists of
5	The outputs of SR latch are
6	In asynchronous circuits the changes occur with the change
7	In synchronous circuits present state is determined by
8	The present states of asynchronous circuits are also called
9	The race in which stable state depends on order is called
10	The wire introduces delay of
11	Primitive flow table for gated latch, for each state has
12	In all the cases final stable state is
13	The complexity of asynchronous circuits is involved in timing problems of
14	Instability condition can be determined from
15	Transition table consists of
16	Transition table that terminates in total stable state gives
17	Time delay device is the memory element of
18	The second step of making transition table is
19	Each logic gate gives delay of

20	The NAND latch works when both inputs are
21	The effect of change of input to more than one state is called
22	The change is state occurs during
23	Asynchronous sequential logic circuits do not use
24	Naming the states is done in
25	The delay elements provide
26	The analysis procedure is to obtain
27	Race condition is present in
28	The output variables&39; value in flow table in written
29	In the design procedure of asynchronous circuit flow table is
30	The race in which stable state does not depends on order is called
31	⇔ The x variable in analysis procedure is used for
32	⇔ Asynchronous sequential logic circuits are used when primary need is
33	Internal state and input values altogether are called
34	⇔ The change in state from 00 to 11 will cause change in
35	⇔ The third step of making transition table is
36	The y variable in analysis procedure is used for
37	The first step of making transition table is
38	The first step of analysis procedure of SR latch is to
39	The transition table gives values of

40	Table having one state in each row is called
41	The next states of asynchronous circuits are also called
42	The inputs of SR latch are
43	Memory elements in asynchronous circuits are
44	One of the properties of asynchronous circuits is
45	Memory elements in synchronous circuits are
46	_____ is a combinational logic circuit that essentially performs the reverse decoder function.
47	In a 2-to-1 multiplexer the no. of control inputs will be
48	In a 4-to-1 multiplexer the no. of control inputs will be
49	The main advantage of dynamic RAMs over static RAMs that they are
50	Dynamic RAMs are best suited to
51	EAROM memory is
52	Choose the correct statement(s) from the following
53	If a decoder has n inputs then the outputs will be
54	If an encoder has 2^n inputs then the outputs will be
55	The demultiplexer is basically a combinational logic circuit to perform the operation
56	Asynchronous sequential logic circuits usually perform operations in
57	In fundamental mode the circuit is assumed to be in
58	The SR latch consists of two cross coupled
59	The circuit removing series of pulses is called

60	The fourth step of making transition table is

opt1	opt2
Y=x	X=x
1 input	2 inputs
transition table	state table
2 steps	4 steps
x and y	a and b
inputs	outputs
unclocked flip-flops	clocked flip-flops
secondary variables	primary variables
critical race	identical race
1 ns	2 ns
1 row	2 rows
changed	same
inputs	outputs
table	map
squares	rectangles
sequence	series
unclocked flip-flops	clocked flip-flops
determining feedback loop	designating output of loops
1 to 5 ns	2 to 10 ns

1	0
undefined condition	race condition
pulse transition	outputs
inputs	outputs
transition table	stable state
large memory	outputs
table	map
synchronous logic circuit	asynchronous logic circuit
inside the square	outside the square
increased to max states	reduced to min states
critical race	identical race
rows	columns
time	pressure
full state	total state
first variable	second variable
determining feedback loop	designating output of loops
rows	columns
determining feedback loop	designating output of loops
label inputs	label outputs
1's	0's

transition table	state table
secondary variables	primary variables
x and y	a and b
unclocked flip-flops	clocked flip-flops
identical mode	map
unclocked flip-flops	clocked flip-flops
Demultiplexer	Encoder
2	3
1	2
non-volatile	easier to use
slow systems	small systems
magnetically alterable	electrically alterable
PROM contains a programmable AND array and a fixed OR array	PLA contains a fixed AND array and a programmable OR array
n	2^n
n	2^n
AND – AND	OR-OR
identical mode	fundamental mode
unstable state	stable state
NOR	NAND
defined circuit	bounce circuit

determining feedback loop	designating output of loops

opt3	opt4
Y=y	X=y
3 inputs	4 inputs
flow table	excitation table
5 steps	6 steps
s and r	q and q'
clock pulses	time
flip-flops	latches
excitation variables	short term memory
non critical race	defined race
3 ns	4 ns
3 rows	4 rows
inverted	undefined
clock pulses	feedback path
graph	logic diagram
circles	oval
unique sequence	unique series
synchronous circuits	asynchronous circuits
deriving functions of Y	plotting Y
3 to 10 ns	3 to 5 ns

inverted	dont cares
reset condition	ideal condition
clock pulses	inputs
clock pulses	time
flow table	excitation table
clock pulses	short term memory
graph	chart
ideal logic circuit	both a and b
beside the square	next to square
changed	remain same
non critical race	defined race
matrix	both a and b
speed	accuracy
initial state	output state
third variable	all variables
deriving functions of Y	plotting Y
matrix	both a and b
deriving functions of Y	plotting Y
label states	label tables
x	y

flow table	primitive flow table
excitation variables	short term memory
s and r	j and k
clock pulses	latches
feedback loop	chart
clock pulses	latches
Multiplexer	None of the above
1	4
3	4
less expensive	all of the above
large systems	one-bit systems
either of (A) or (B) above	none of the above
PROM contains a fixed AND array and programmable OR array	none of the above
1	n^2
1	0
AND-OR	OR-AND
reserved mode	reset mode
reset state	clear state
AND	both a and b
debounce circuit	undefined circuit

deriving functions of Y	plotting Y

Answer
Y=y
2 inputs
excitation table
6 steps
q and q'
inputs
flip-flops
secondary variables
critical race
1 ns
1 row
same
feedback path
logic diagram
squares
unique sequence
asynchronous circuits
designating output of loops
2 to 10 ns

1
race condition
pulse transition
clock pulses
flow table
short term memory
table
asynchronous logic circuit
inside the square
reduced to min states
non critical race
columns
speed
total state
first variable
deriving functions of Y
rows
determining feedback loop
label outputs
y

primitive flow table
excitation variables
s and r
unclocked flip-flops
feedback loop
clocked flip-flops
Encoder
1
2
less expensive
large systems
electrically alterable
PROM contains a fixed AND array and programmable OR array
2^n
n
OR-AND
fundamental mode
stable state
both a and b
debounce circuit

plotting Y

S.no	Questions
1	How many address bits are needed to select all memory locations in the 2118 16K × 1 RAM?
2	The check sum method of testing a ROM:
3	What is the meaning of RAM, and what is its primary role?
4	The storage element for a static RAM is the _____.
5	In a DRAM, what is the state of R/W during a read operation?
6	The condition occurring when two or more devices try to write data to a bus simultaneously is called _____.
7	Which is/are the basic refresh mode(s) for dynamic RAM?
8	One of the most important specifications on magnetic media is the _____.
9	A 64-bit word consists of _____.
10	Which of the following RAM timing parameters determine its operating speed?
11	The reason the data outputs of most ROM ICs are tristate outputs is to:
12	Select the statement that best describes Read-Only Memory (ROM).
13	How many 2K × 8 ROM chips would be required to build a 16K × 8 memory system?
14	What is the maximum time required before a dynamic RAM must be refreshed?
15	Which of the following best describes random-access memory (RAM)?
16	A CD-R disk is created by applying heat to special chemicals on the disk and these chemicals reflect less light than the areas that are not burned, thus creating the same effect as a pit does on a regular CD
17	Which of the following best describes static memory devices?
18	Which is not a removable drive?
19	Which of the following best describes EPROMs?

20	How many storage locations are available when a memory device has 12 address lines?
21	FIFO is formed by an arrangement of _____.
22	Why do most dynamic RAMs use a multiplexed address bus?
23	CCD stands for _____.
24	What is the major difference between SRAM and DRAM?
25	Which of the following best describes volatile memory?
26	What is a major disadvantage of RAM?
27	What two functions does a DRAM controller perform?
28	Dynamic memory cells store a data bit in a _____.
29	Which is not part of a hard disk drive?
30	⇔ ROMs retain data when the _____.
31	⇔ Typically, how often is DRAM refreshed?
32	Which type of ROM can be erased by an electrical signal?
33	⇔ Suppose that a certain semiconductor memory chip has a capacity of 8K × 8. How many bytes could be stored in this device?
34	⇔ Data is written to and read from the disk via a magnetic _____ head mechanism in the floppy drive.
35	What does the term "random access" mean in terms of memory?
36	A 64-Mbyte SIMM is installed into a system, but when a memory test is executed, the SIMM is detected as a 32-Mbyte device. What is a possible cause?
37	How many address lines would be required for a 2K × 4 memory chip?
38	When a RAM module passes the checkerboard test it is:
39	Which type of ROM has to be custom built by the factory?

40	What is the computer main memory?
41	A major disadvantage of the mask ROM is that it:
42	The periodic recharging of DRAM memory cells is called _____.
43	Which of the following is normally used to initialize a computer system's hardware?
44	What is the difference between static RAM and dynamic RAM?
45	Microprocessors and memory ICs are generally designed to drive only a single TTL load. Therefore, if several inputs are being driven from the same bus, any memory IC must be
46	What are the typical values of t_{OE}?
47	Which type of ROM can be erased by UV light?
48	Which of the following is NOT a type of memory?
49	How many address bits are required for a 4096-bit memory organized as a 512 × 8 memory?
50	In general, the _____ have the smallest bit size and the _____ have the largest
51	Advantage(s) of an EEPROM over an EPROM is/are:
52	The mask ROM is _____.
53	How many 1K × 4 RAM chips would be required to build a 1K × 8 memory system?
54	Which of the following memories uses a MOS capacitor as its memory cell?
55	Which of the following faults will the checkerboard pattern test for in RAM?
56	On a CD-ROM, _____ are raised areas representing a 1.
57	The location of a unit of data in a memory array is called its _____.
58	Why is a refresh cycle necessary for a dynamic RAM?
59	Which is not a magnetic storage device?

opt1
8
indicates if the data in more than one memory location is incorrect.
Readily Available Memory; it is the first level of memory used by the computer in all of its operations.
diode
Low
address decoding
Burst refresh
rotation speed
4 bytes
t_{ACC}
allow for three separate data input lines.
nonvolatile, used to store information that changes during system operation
2
2 ms
a type of memory in which access time depends on memory location
TRUE
memory devices that are magnetic in nature and do not require constant refreshing
Zip
EPROMs can be programmed only once.

144
diodes
It is the only way to do it.
capacitor charging device
DRAMs must be periodically refreshed.
memory that retains stored information when electrical power is removed
Its access speed is too slow.
address multiplexing and data selection
diode
Spindle
power is off
2 to 8 ms
ROM
8000
cylinder
Addresses must be accessed in a specific order.
The memory module was not installed properly.
8
able to read and write only 1s
ROM

Hard drive and RAM
is time consuming to change the stored data when system requirements change
multiplexing
Bootstrap memory
Static RAM must be refreshed, dynamic RAM does not.
buffered
10 to 20 ns for bipolar
ROM
RAM
2
EEPROMs, Flash
the EPROM can be erased with ultraviolet light in much less time than an EEPROM
permanently programmed during the manufacturing process
2
SRAM
Short between adjacent cells
mounds
storage
to clear the flip-flops
Magnetic disk

opt2
10
provides a means for locating and correcting data errors in specific memory locations.
Random Access Memory; it is memory that can be reached by any sub- system within a computer, and at any time
resistor
High
bus contention
Distributed refresh
tracks per inch
8 bytes
t_{AA} and t_{ACS}
allow the bidirectional flow of data between the bus lines and the ROM registers.
nonvolatile, used to store information that does not change during system operation
4
4 ms
a type of memory that can be written to only once but can be read from an infinite number of times
FALSE
memory devices that are magnetic in nature and require constant refreshing
Jaz
EPROMs can be erased by UV.

512
transistors
to make it faster
capacitor-capacitor drain
SRAMs can hold data via a static charge, even with power off.
memory that loses stored information when electrical power is removed
Its matrix size is too big.
address multiplexing and the refresh operation
resistor
Platter
power is on
4 to 16 ms
mask ROM
64000
read/write
Any address can be accessed in any order.
The voltage on the memory module is incorrect.
10
faulty.
mask ROM

CD-ROM and hard drive
is very expensive to change the stored data when system requirements change
bootstrapping
Volatile memory
There is no difference.
decoded
25 to 100 ns for NMOS
mask ROM
ROM
4
SRAM, mask ROM
the EEPROM can be erased and reprogrammed without removal from the circuit
volatile
4
DRAM
Ability to store both 0s and 1s
lands
RAM
to set the flip-flops
Magnetic tape

opt3
14
allows data errors to be pinpointed to a specific memory location.
Random Access Memory; it is the memory used for short-term temporary data storage within the computer.
capacitor
Hi-Z
bus collisions
Open refresh
data transfer rate
10 bytes
t_{CO} and t_{OD}
permit the connection of many ROM chips to a common data bus.
volatile, used to store information that changes during system operation
8
8 ms
a type of memory in which access time is the same for each memory location
-
semiconductor memory devices in which stored data will not be retained with the power applied unless constantly refreshed
Hard
EPROMs can be erased by shorting all inputs to the ground.

2048
MOS cells
to keep the number of pins on the chip to a minimum
charged-capacitor device
The only difference is the terminal from which the data is removed—from the FET Drain or Source.
magnetic memory
It is volatile.
data selection and the refresh operation
capacitor
Read/write head
system is down
8 to 16 μs
EPROM
65536
recordable
-
The most significant address line is stuck high or low.
11
probably good.
EPROM

RAM and ROM
cannot be reprogrammed if stored data needs to be changed
refreshing
External mass memory
Dynamic RAM must be refreshed, static RAM does not.
addressed
12 to 50 ns for CMOS
EPROM
FPROM
8
mask ROM, SRAM
the EEPROM has the ability to erase and reprogram individual words
easy to reprogram
8
ROM
Dynamically introduced errors between cells
holes
address
The refresh cycle discharges the capacitor cells
Magneto-optical disk

opt4
16
simply indicates that the contents of the ROM are incorrect.
Resettable Automatic Memory; it is memory that can be used and then automatically reset, or cleared, after being read from or written to
flip-flop
None of the above
address multiplexing
Burst refresh and distributed refresh
polarity reversal rate
12 bytes
t_{RC} and t_{WC}
isolate the registers from the data bus during read operations.
volatile, used to store information that does not change during system operation
13
10 ms
mass memory
-
semiconductor memory devices in which stored data is retained as long as power is applied
SuperDisk
All of the above.

4096
shift registers
-
charge-coupled device
Dynamic RAMs are always active; static RAMs must reset between data read/write cycles.
nonmagnetic
High power consumption
data selection and CPU accessing
flip-flop
Valve
all of the above
1 to 2 s
EEPROM
8192
cluster
-
The address decoder on the SIMM is faulty.
12
able to read and write only 0s.
EEPROM

CMOS and hard drive
has an extremely short life expectancy and requires frequent replacement
flashing
Static memory
-
stored
All of the above
EEPROM
EEPROM
9
DRAM, PROM
the EEPROM can be erased and reprogrammed without removal from the circuit, and can erase and reprogram individual words
extremely expensive
16
FIFO
All of the above
pits
data
The refresh cycle keeps the charge on the capacitor cells
Optical disk

Answer
14
simply indicates that the contents of the ROM are incorrect.
Random Access Memory; it is the memory used for short-term temporary data storage within the computer
flip-flop
High
bus contention
Burst refresh and distributed refresh
data transfer rate
8 bytes
t_{RC} and t_{WC}
permit the connection of many ROM chips to a common data bus.
nonvolatile, used to store information that does not change during system operation
8
2 ms
a type of memory in which access time is the same for each memory location
TRUE
semiconductor memory devices in which stored data is retained as long as power is applied
Hard
EPROMs can be erased by UV.

4096
shift registers
to keep the number of pins on the chip to a minimum
charge-coupled device
DRAMs must be periodically refreshed.
memory that loses stored information when electrical power is removed
It is volatile.
address multiplexing and the refresh operation
capacitor
Valve
all of the above
4 to 16 ms
EEPROM
8192
read/write
Any address can be accessed in any order.
The address decoder on the SIMM is faulty.
11
probably good.
mask ROM

RAM and ROM
cannot be reprogrammed if stored data needs to be changed
refreshing
Bootstrap memory
Dynamic RAM must be refreshed, static RAM does not.
buffered
All of the above
EPROM
FPROM
9
EEPROMs, Flash
the EEPROM can be erased and reprogrammed without removal from the circuit, and can erase and reprogram individual words
permanently programmed during the manufacturing process
2
DRAM
All of the above
lands
address
The refresh cycle keeps the charge on the capacitor cells
Optical disk

Digital Electronics

UNIT-I NUMBER SYSTEMS

Introduction to Digital System

The term digital refers to any process that is accomplished in discrete manner. A best example of a digital system is Digital computer.

The Electronic circuits are basically classified into two types namely

i) Analog
ii) Digital

Almost all digital circuits are logical circuits because

1. It is easier to manipulate logical values (i.e., 0/1 or low/high values) than to manipulate and process multiple (Discrete) voltage levels.

2. Everyone can use the formal laws of Boolean logic to design logic circuits easily and systematically. Hence these laws which are used to formulate digital logic are called as Boolean algebra or Switching algebra

Types of Number Systems are

i) Decimal Number system
ii) Binary Number system
iii) Octal Number system
iv) Hexadecimal Number system

Complements

Complements are used in digital computers for simplifying the subtraction operation and for logical manipulation. For any numbering system, there are two types of complements available, which are

i) r's complement
ii) (r-1)'s complement.

For Binary numbering system we have 2's complement and 1's complement.

Binary Codes

Dec	Hex	Oct	Bin
0	0	000	00000000
1	1	001	00000001
2	2	002	00000010
3	3	003	00000011
4	4	004	00000100
5	5	005	00000101
6	6	006	00000110
7	7	007	00000111
8	8	010	00001000
9	9	011	00001001
10	A	012	00001010
11	B	013	00001011
12	C	014	00001100
13	D	015	00001101
14	E	016	00001110
15	F	017	00001111

Dec	Hex	Oct	Bin
16	10	020	00010000
17	11	021	00010001
18	12	022	00010010
19	13	023	00010011
20	14	024	00010100
21	15	025	00010101
22	16	026	00010110
23	17	027	00010111
24	18	030	00011000
25	19	031	00011001
26	1A	032	00011010
27	1B	033	00011011
28	1C	034	00011100
29	1D	035	00011101
30	1E	036	00011110
31	1F	037	00011111

Conversion of codes from one form to another:

1. Binary to Decimal

Binary	Decimal
11011_2	
$2^4+2^3+0^1+2^1+2^0$	=16+8+0+2+1
Result	27_{10}

2. Decimal to binary

Division	Remainder	Binary
25/2	= 12+ remainder of 1	1 (Least Significant Bit)
12/2	= 6 + remainder of 0	0
6/2	= 3 + remainder of 0	0
3/2	= 1 + remainder of 1	1
1/2	= 0 + remainder of 1	1 (Most Significant Bit)
Result	25_{10}	= 11001_2

3.Binary to Octal

100 111 0102 = (100) (111) (010)2 = 4 7 28

Decimal to octal

Division	Result	Binary
177/8	= 22+ remainder of 1	1 (Least Significant Bit)
22/ 8	= 2 + remainder of 6	6
2 / 8	= 0 + remainder of 2	2 (Most Significant Bit)
Result	177_{10}	= 261_8
Binary		= 010110001_2

4. Decimal to Hexadecimal

Division	Result	Hexadecimal
378/16	= 23+ remainder of 10	A (Least Significant Bit)23
23/16	= 1 + remainder of 7	7
1/16	= 0 + remainder of 1	1 (Most Significant Bit)
Result	378_{10}	= $17A_{16}$
Binary		= 0001 0111 1010_2

5. Hexadecimal to binary

$1011\ 0010\ 1111_2$ = (1011) (0010) $(1111)_2$ = B 2 F_{16}

6. Hexadecimal to octal

Hexadecimal	Binary/Octal
5A816	= **0101** 1010 **1000** (Binary)
	= **010** 110 **101** 000 (Binary)
Result	= 2 6 5 0 (Octal)

Binary Arithmetic Operations:

Rules of Binary Addition

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0, and carry 1 to the next more significant bit

For example,

00011010 + 00001100 = 00100110

	1	*1*						*carries*
	0	0	0	1	1	0	1	0 = 26(base 10)
+	0	0	0	0	1	1	0	0 = 12(base 10)
	0	0	1	0	0	1	1	0 = 38(base 10)

00010011 + 00111110 = 01010001

1	*1*	*1*	*1*	*1*				*carries*
0	0	0	1	0	0	1	1	= 19(base 10)
+ 0	0	1	1	1	1	1	0	= 62(base 10)
0	1	0	1	0	0	0	1	= 81(base 10)

Rules of Binary Subtraction

0 - 0 = 0
0 - 1 = 1, and borrow 1 from the next more significant bit
1 - 0 = 1
1 - 1 = 0

For example,

00100101 - 00010001 = 00010100

	0							*borrows*
	0	0	~~1~~	*1*0	0	1	0	1 = 37(base 10)
-	0	0	0	1	0	0	0	1 = 17(base 10)
	0	0	0	1	0	1	0	0 = 20(base 10)

00110011 - 00010110 = 00011101

	0	*1*	*0*	*1*				*borrows*
	0	0	~~1~~	~~1~~	~~0~~	*1*0	1	1 = 51(base 10)
-	0	0	0	1	0	1	1	0 = 22(base 10)
	0	0	0	1	1	1	0	1 = 29(base 10)

Rules of Binary Multiplication

0 x 0 = 0
0 x 1 = 0
1 x 0 = 0
1 x 1 = 1, and no carry or borrow bits

For example,

00101001 × 00000110 = 11110110

		0	0	1	0	1	0	0	1	=	$41_{(base\ 10)}$
		× 0	0	0	0	0	1	1	0	=	$6_{(base\ 10)}$
		0	0	0	0	0	0	0	0		
	0	0	1	0	1	0	0	1			
0	0	1	0	1	0	0	1				
0	0	1	1	1	1	0	1	1	0	=	$246_{(base\ 10)}$

00010111 × 00000011 = 01000101

		0	0	0	1	0	1	1	1	=	$23_{(base\ 10)}$
		× 0	0	0	0	0	0	1	1	=	$3_{(base\ 10)}$
					1	*1*	*1*	*1*			*carries*
		0	0	0	1	0	1	1	1		
	0	0	0	1	0	1	1	1			
	0	0	1	0	0	0	1	0	1	=	$69_{(base\ 10)}$

Another Method: Binary multiplication is the same as repeated binary addition; add the multicand to itself the multiplier number of times.

For example,

00001000 × 00000011 = 00011000

	1									*carries*
	0	0	0	0	1	0	0	0	=	$8_{(base\ 10)}$
	0	0	0	0	1	0	0	0	=	$8_{(base\ 10)}$
+	0	0	0	0	1	0	0	0	=	$8_{(base\ 10)}$
	0	0	0	1	1	0	0	0	=	$24_{(base\ 10)}$

Binary Division

Binary division is the repeated process of subtraction, just as in decimal division.

For example,

00101010 ÷ 00000110 1 1 1 = $7_{(base\ 10)}$

= 00000111

1 1 0) 0 0 0 ~~1~~ 10 1 0 1 0 = 42(base 10)
- 1 1 0 = 6(base 10)

1 *borrows*

~~1~~ ~~0~~ 10 1
- 1 1 0

1 1 0
- 1 1 0

0

10000111 ÷ 00000101
= 00011011

1 1 0 1 1 = 27(base 10)

1 0 1) ~~1~~ ~~0~~ ~~0~~ 10 0 1 1 1 = 135(base 10)
- 1 0 1 = 5(base 10)

1 ~~1~~ 1 0
- 1 0 1

1 1
- 0

1 1 1
- 1 0 1

1 0 1
- 1 0 1

0

Types of binary codes

Binary codes are codes which are represented in binary system with modification from the original ones. Below we will be seeing the following: Weighted codes and Non-Weighted codes

I. Weighted binary codes

Weighted binary codes are those which obey the positional weighting principles, each position of the number represents a specific weight. The binary counting sequence is an example.

0	0000	0000	0000	0011
1	0001	0001	0001	0100
2	0010	0010	0011	0101

3	0011	0011	0101	0110
4	0100	0100	0111	0111
5	0101	1011	1000	1000
6	0110	1100	1010	1001
7	0111	1101	1100	1010
8	1000	1110	1110	1011
9	1001	1111	1111	1100

1. **8421 code/BCD code**

The BCD (Binary Coded Decimal) is a straight assignment of the binary equivalent. It is possible to assign weights to the binary bits according to their positions. The weights in the BCD code are 8,4,2,1.

Example: The bit assignment 1001, can be seen by its weights to represent the decimal 9 because 1x8+0x4+0x2+1x1 = 9

2. **2421 code**

This is a weighted code; its weights are 2, 4, 2 and 1. A decimal number is represented in 4-bit form and the total four bits weight is 2 + 4 + 2 + 1 = 9. Hence the 2421 code represents the decimal numbers from 0 to 9.

3. **5211 code**

This is a weighted code; its weights are 5, 2, 1 and 1. A decimal number is represented in 4-bit form and the total four bits weight is 5 + 2 + 1 + 1 = 9. Hence the 5211 code represents the decimal numbers from 0 to 9.

Reflective code

A code is said to be reflective when code for 9 is complement for the code for 0, and so is for 8 and 1 codes, 7 and 2, 6 and 3, 5 and 4. Codes 2421, 5211, and excess-3 are reflective, whereas the 8421 code is not.

Sequential code

A code is said to be sequential when two subsequent codes, seen as numbers in binary representation, differ by one. This greatly aids mathematical manipulation of data. The 8421 and Excess-3 codes are sequential, whereas the 2421 and 5211 codes are not.

II. Non-Weighted code

Non weighted codes are codes that are not positionally weighted. That is, each position within the binary number is not assigned a fixed value.

1. **Excess-3 code**

Excess-3 is a non weighted code used to express decimal numbers. The code derives its name from the fact that each binary code is the corresponding 8421 code plus 0011(3).

Example: 1000 of 8421 = 1011 in Excess-3

2. Gray code

The gray code belongs to a class of codes called minimum change codes, in which only one bit in the code changes when moving from one code to the next. The Gray code is non-weighted code, as the position of bit does not contain any weight. The gray code is a reflective digital code which has the special property that any two subsequent numbers codes differ by only one bit. This is also called a unit-distance code. In digital Gray code has got a special place.

Decimal Number	Binary Code	Gray Code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

3. Error detecting and correcting codes

For reliable transmission and storage of digital data, error detection and correction is required. Below are a few examples of codes which permit error detection and error correction after detection.

Error detecting codes:
When data is transmitted from one point to another, like in wireless transmission, or it is just stored, like in hard disks and memories, there are chances that data may get corrupted. To detect these data errors, we use special codes, which are error detection codes.

Parity bit:
In parity codes, every data byte, or nibble (according to how user wants to use it) is checked

if they have even number of ones or even number of zeros. Based on this information an additional bit is appended to the original data. Thus if we consider 8-bit data, adding the parity bit will make it 9 bit long. At the receiver side, once again parity is calculated and matched with the received parity (bit 9), and if they match, data is ok, otherwise data is corrupt.

Two types of parity

Even parity: Checks if there is an even number of ones; if so, parity bit is zero. When the number of ones is odd then parity bit is set to 1.

Odd Parity: Checks if there is an odd number of ones; if so, parity bit is zero. When number of ones is even then parity bit is set to 1.

Error correcting codes:

Error-correcting codes not only detect errors, but also correct them. This is used normally in Satellite communication, where turn-around delay is very high as is the probability of data getting corrupt.

1. **Hamming codes**

Hamming code adds a minimum number of bits to the data transmitted in a noisy channel, to be able to correct every possible one-bit error. It can detect (not correct) two-bit errors and cannot distinguish between 1-bit and 2-bits inconsistencies. It can't - in general - detect 3(or more)-bits errors.

III. Alphanumeric codes

The binary codes that can be used to represent all the letters of the alphabet, numbers and mathematical symbols, punctuation marks, are known as alphanumeric codes or character codes. These codes enable us to interface the input-output devices like the keyboard, printers, video displays with the computer.

1. **ASCII codes**

ASCII stands for American Standard Code for Information Interchange. It has become a world standard alphanumeric code for microcomputers and computers. It is a 7-bit code representing 2^7 = 128 different characters. These characters represent 26 upper case letters (A to Z), 26 lowercase letters (a to z), 10 numbers (0 to 9), 33 special characters and symbols and 33 control characters.

2. **EBCDIC codes**

EBCDIC stands for Extended Binary Coded Decimal Interchange. It is mainly used with large computer systems like mainframes. EBCDIC is an 8-bit code and thus accommodates up to 256 characters. An EBCDIC code is divided into two portions: 4 zone bits (on the left) and 4 numeric bits (on the right).

Simplification of Boolean functions can be done by using

i) Boolean theorems and postulates.

ii) Karnaugh Map(K-Map)
iii) Prime Implicant Method or Tabulation methods or Quine Mc-Cluskey Method

Boolean Algebra theorems and postulates.

Principle of duality

Principle of duality states that a dual expression can be obtainbed by Interchanging the OR and AND operatio expression, Interchanging the 0 and 1 elements of the expression and Not changing the form of the variables.

T1: Commutative Law
(a) $A + B = B + A$
(b) $A\,B = B\,A$

T2: Associative Law
(a) $(A + B) + C = A + (B + C)$
(b) $(A\,B)\,C = A\,(B\,C)$

T3: Distributive Law
(a) $A\,(B + C) = A\,B + A\,C$
(b) $A + (B\,C) = (A + B)\,(A + C)$

T4: Idempotent Law for Addition
(a) $A + A = A$
(b) $A\,A = A$

T5: Negation Law
(a) $(\overline{A}) = \overline{A}$
(b) (A)''= A

T6: Redundance Law
(a) $A + A\,B = A$
(b) $A\,(A + B) = A$

T7: Law of Identity
(a) $0 + A = A$
(b) $1\,A = A$
(c) $1 + A = 1$
(d) $0\,A = 0$

T8 : Idempotent law for Multiplication
(a) $\overline{A} + A = 1$
(b) $\overline{A}\,A = 0$

T9 : Absorption Law
(a) $A + \overline{A}\,B = A + B$
(b) $A\,(\overline{A} + B) = A\,B$

T10 : De Morgan's Theorem
(a) (A+B)'= A' . B'
(b) (A.B)'= A'+B'

Canonical Form of Boolean Expressions

An expanded form of Boolean expression, where each term contains all Boolean variables in their true or complemented form, is also known as the canonical form of the expression. As an illustration, $f(A.B, C) = \overline{A}.\overline{B}.\overline{C} + \overline{A}.\overline{B}.C + A.B.C$ is a Boolean function of three variables expressed in canonical form. This function after simplification reduces to $\overline{A}.\overline{B} + A.B.C$ and loses its canonical form

MIN TERMS AND MAX TERMS

Any boolean expression may be expressed in terms of either minterms or maxterms. To do this we must first define the concept of a literal. A literal is a single variable within a term which may or may not be complemented. For an expression with N variables, minterms and maxterms are defined as follows :

- A minterm is the product of N distinct literals where each literal occurs exactly once.
- A maxterm is the sum of N distinct literals where each literal occurs exactly once.

Product-of-Sums Expressions

Standard Forms

A product-of-sums expression contains the product of different terms, with each term being either a single literal or a sum of more than one literal. It can be obtained from the truth table by considering those input combinations that produce a logic _0‘ at the output. Each such input combination gives a term, and the product of all such terms gives the expression.

Different terms are obtained by taking the sum of the corresponding literals. Here, _0‘ and _1‘ respectively mean the un complemented and complemented variables, unlike sum-of-products expressions where _0‘ and _1‘ respectively mean complemented and un complemented variables.

Since each term in the case of the product-of-sums expression is going to be the sum of literals, this implies that it is going to be implemented using an OR operation. Now, an OR gate produces a logic _0‘ only when all its inputs are in the logic _0‘ state, which means that the first term corresponding to the second row of the truth table will be A+B+C. The product-of-sums Boolean expression for this truth table is given by Transforming the given product-of-sums expression into an equivalent sum-of-products expression is a straightforward process. Multiplying out the given expression and carrying out the obvious simplification provides the equivalent sum-of-products expression:A given sum-of-products expression can be transformed into an equivalent product-of-sums expression by (a) taking the dual of the given expression, (b) multiplying out different terms to get the sum-of products form, (c) removing redundancy and (d) taking a dual to get the equivalent product-of-sums expression. As an illustration, let us find the equivalent product-of-sums expression of the sum-of products expression

$A.B + \overline{A}.\overline{B}$

The dual of the given expression = $(A+B).(\overline{A}+\overline{B})$:

$(A+B).(\overline{A}+\overline{B}) = A.\overline{A} + A.\overline{B} + B.\overline{A} + B.\overline{B} = 0 + A.\overline{B} + B.\overline{A} + 0 = A.\overline{B} + \overline{A}.B$

The dual of $(A.\overline{B} + \overline{A}.B) = (A+\overline{B}).(\overline{A}+B)$. Therefore

$$A.B + \overline{A}.\overline{B} = (A+\overline{B}).(\overline{A}+B)$$

Minimization Technique

The primary objective of all simplification procedures is to obtain an expression that has the minimum number of terms. Obtaining an expression with the minimum number of literals is usually the secondary objective. If there is more than one possible solution with the same number of terms, the one having the minimum number of literals is the choice.

There are several methods for simplification of Boolean logic expressions. The process is usually called logic minimization‖ and the goal is to form a result which is efficient. Two methods we will discuss are algebraic minimization and Karnaugh maps. For very complicated problems the former method can be done using special software analysis programs. Karnaugh maps are also limited to problems with up to 4 binary inputs. The Quine–McCluskey tabular method is used for more than 4 binary inputs.

Karnaugh Map Method

Maurice Karnaugh, a telecommunications engineer, developed the Karnaugh map at Bell Labs in 1953 while designing digital logic based telephone switching circuits.Karnaugh maps reduce logic functions more quickly and easily compared to Boolean algebra. By reduce we mean simplify, reducing the number of gates and inputs. We like to simplify logic to a lowest cost form to save costs by elimination of components. We define lowest cost as being the lowest number of gates with the lowest number of inputs per gate.

A Karnaugh map is a graphical representation of the logic system. It can be drawn directly from either minterm (sum-of-products) or maxterm (product-of-sums) Boolean expressions. Drawing a Karnaugh map from the truth table involves an additional step of writing the minterm or maxterm expression depending upon whether it is desired to have a minimized sum-of-products or a minimized product of-sums expression

Construction of a Karnaugh Map

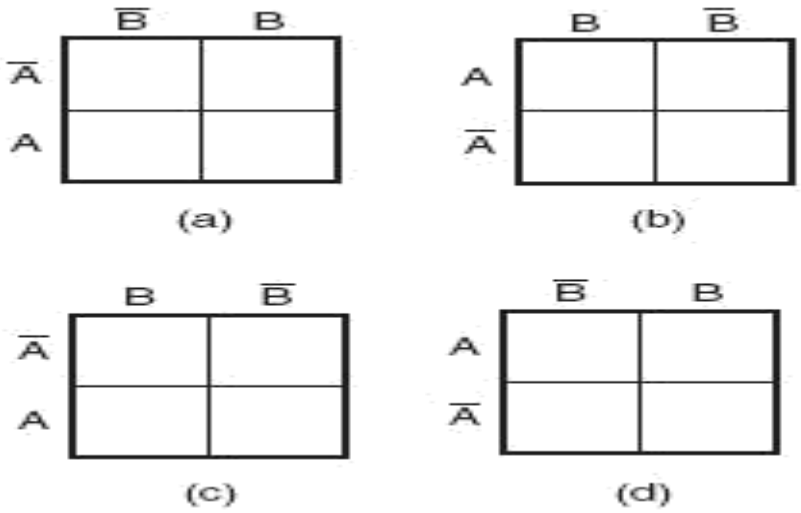
An n-variable Karnaugh map has 2n squares, and each possible input is allotted a square. In the case of a minterm Karnaugh map, ‗1' is placed in all those squares for which the output is ‗1', and ‗0' is placed in all those squares for which the output is ‗0'. 0s are omitted for simplicity. An ‗X' is placed in squares corresponding to ‗don't care' conditions. In the case of a maxterm Karnaugh map, a ‗1' is placed in all those squares for which the output is ‗0', and a ‗0' is placed for input entries corresponding to a ‗1' output. Again, 0s are omitted for simplicity, and an ‗X' is placed in squares corresponding to ‗don't care' conditions. The choice of terms identifying different rows and columns of a Karnaugh map is not unique for a given number of variables. The only condition to be satisfied is that the designation of adjacent rows and adjacent columns should be the same except for one of the

literals being complemented. Also, the extreme rows and extreme columns are considered adjacent.

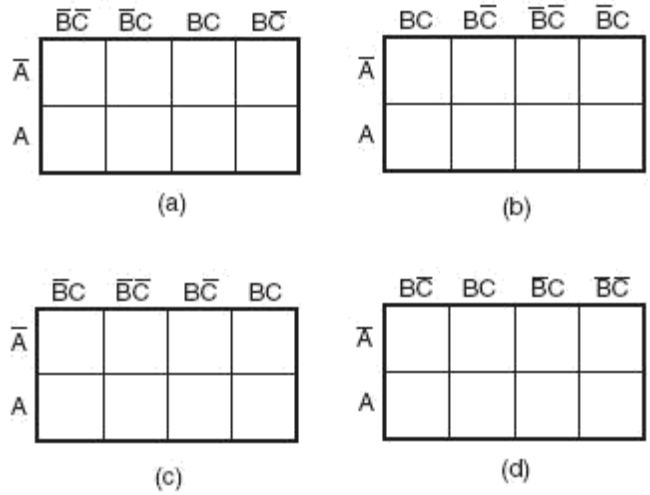
Some of the possible designation styles for two-, three- and four-variable minterm Karnaugh maps are shown in the figure below.

The style of row identification need not be the same as that of column identification as long as it meets the basic requirement with respect to adjacent terms. It is, however, accepted practice to adopt a uniform style of row and column identification. Also, the style shown in the figure below is more commonly used. A similar discussion applies for maxterm Karnaugh maps. Having drawn the Karnaugh map, the next step is to form groups of 1s as per the following guidelines:

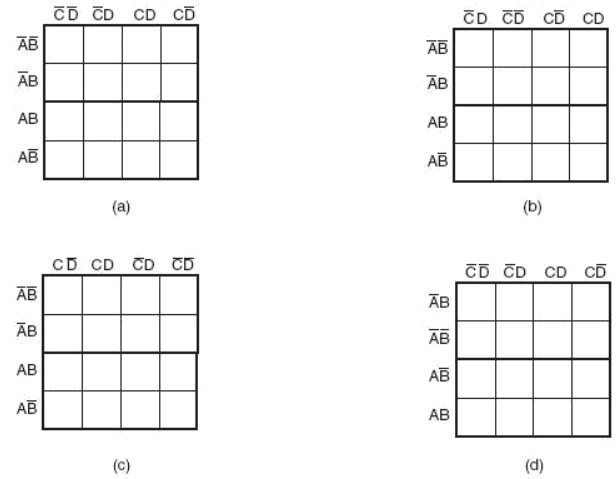
- Each square containing a _1‘ must be considered at least once, although it can be considered as often as desired.
- The objective should be to account for all the marked squares in the minimum number of groups.
- The number of squares in a group must always be a power of 2, i.e. groups can have 1,2, 4_ 8, 16, squares.
- Each group should be as large as possible, which means that a square should not be accounted for by itself if it can be accounted for by a group of two squares; a group of two squares should not be made if the involved squares can be included in a group of four squares and so on.
- _Don‘t care‘ entries can be used in accounting for all of 1-squares to make optimum groups. They are marked _X‘ in the corresponding squares. It is, however, not necessary to account for all _don‘t care‘ entries. Only such entries that can be used to advantage should be used.



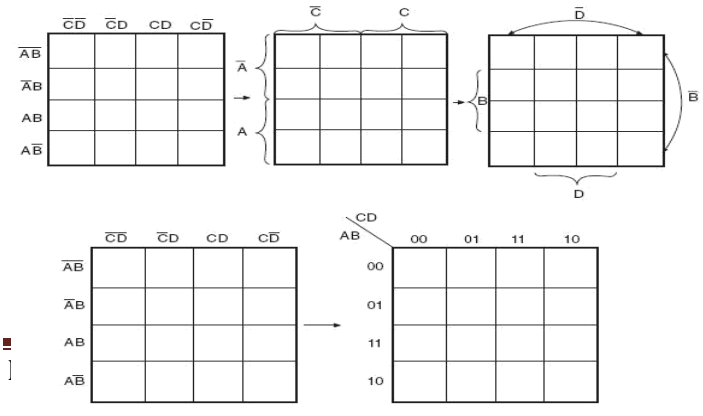
Two variable K Map



Three variable K Map



Four variable K Map



Different Styles of row and column identification

Having accounted for groups with all 1s, the minimum _sum-of-products‘ or _product-of-sums‘ expressions can be written directly from the Karnaugh map. Minterm Karnaugh map and Maxterm Karnaugh map of the Boolean function of a two-input OR gate. The Minterm and Maxterm Boolean expressions for the two-input OR gate are as follows:

$$Y = A + B \text{ (maxterm or product-of-sums)}$$

$$Y = \overline{A}.B + A.\overline{B} + A.B \text{ (minterm or sum-of-products)}$$

Truth table

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

	$\overline{B}$	B
$\overline{A}$		1
A	1	1

Sum-of-products K-map

	$\overline{B}$	B
$\overline{A}$		
A		1

Product-of-sums K-map

Minterm Karnaugh map and Maxterm Karnaugh map of the three variable Boolean function

$$Y = \overline{A}.\overline{B}.\overline{C} + \overline{A}.B.\overline{C} + A.\overline{B}.\overline{C} + A.B.\overline{C}$$

$$Y = (\overline{A} + \overline{B} + \overline{C}).(\overline{A} + B + \overline{C}).(A + \overline{B} + \overline{C}).(A + B + \overline{C})$$

A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

	$\overline{B}\overline{C}$	$\overline{B}C$	BC	$B\overline{C}$
$\overline{A}$	1			1
A	1			1

Sum-of-products K-map

	$\overline{B}+\overline{C}$	$\overline{B}+C$	B+C	$B+\overline{C}$
$\overline{A}$	1			1
A	1			1

Product-of-sums K-map

The truth table, Minterm Karnaugh map and Maxterm Karnaugh map of the four variable

Boolean function

$$Y = \overline{A}.\overline{B}.\overline{C}.\overline{D}+\overline{A}.\overline{B}.\overline{C}.D+\overline{A}.B.\overline{C}.\overline{D}+\overline{A}.B.\overline{C}.D+A.\overline{B}.\overline{C}.\overline{D}+A.\overline{B}.\overline{C}.D+A.B.\overline{C}.\overline{D}+A.B.\overline{C}.D$$

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

To illustrate the process of forming groups and then writing the corresponding minimized Boolean expression, The below figures respectively show minterm and maxterm Karnaugh maps for the Boolean functions expressed by the below equations. The minimized expressions as deduced from Karnaugh maps in the two cases are given by Equation in the case of the minterm Karnaugh map and Equation in the case of the maxterm Karnaugh map:

Quine–McCluskey Tabular Method

The Quine–McCluskey tabular method of simplification is based on the complementation theorem, which says that XY +XY' =X. where X represents either a variable or a term or an expression and Y is a variable. This theorem implies that, if a Boolean expression contains two terms that differ only in one variable, then they can be combined together and replaced with a term that is smaller by one literal

Let us consider an example. Consider the following sum-of-products expression:

$$\overline{A}.B.C+\overline{A}.\overline{B}.D+A.\overline{C}.D+B.\overline{C}.\overline{D}+\overline{A}.B.\overline{C}.D$$

In the first step, we write the expanded version of the given expression. It can be written as follows:

$$\overline{A}.B.C.D+\overline{A}.B.C.\overline{D}+\overline{A}.\overline{B}.C.D+\overline{A}.\overline{B}.\overline{C}.D+A.B.\overline{C}.D+A.\overline{B}.\overline{C}.D+A.B.\overline{C}.\overline{D}$$
$$+\overline{A}.B.\overline{C}.\overline{D}+\overline{A}.B.\overline{C}.D$$

The formation of groups, the placement of terms in different groups and the first-round matching are shown as follows:

A	B	C	D
0	0	0	1
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	1
1	1	0	0
1	1	0	1

A	B	C	D	
0	0	0	1	✓
0	1	0	0	✓
0	0	1	1	✓
0	1	1	0	✓
0	1	1	1	✓
1	0	0	1	✓
1	1	0	0	✓
0	1	1	1	✓
1	1	0	1	✓

A	B	C	D	
0	0	–	1	✓
0	–	0	1	✓
–	0	0	1	✓
0	1	0	–	✓
0	1	–	0	✓
–	1	0	0	✓
0	–	1	1	✓
0	1	–	1	✓
–	1	0	1	✓

The second round of matching begins with the table shown on the previous page. Each term in the first group is compared with every term in the second group. For instance, the first term in the first group 00−1 matches with the second term in the second group 01−1 to yield 0−−1, which is recorded in the table shown below. The process continues until all terms have been compared for a possible match. Since this new table has only one group, the terms contained therein are all prime implicants. In the present example, the terms in the first and second tables have all found a match. But that is not always the case.

A	B	C	D	
0	–	–	1	*
–	–	0	1	*
0	1	–	–	*
–	1	0	–	*

The next table is what is known as the prime implicant table. The prime implicant table contains all the original terms in different columns and all the prime implicants recorded in different rows as shown below:

0001	0011	0100	0101	0110	0111	1001	1100	1101		
✓	✓		✓		✓				0−−1	$P \rightarrow \overline{A}.D$
✓			✓			✓		✓	−−01	$Q \rightarrow \overline{C}.D$
		✓	✓	✓	✓				01−−	$R \rightarrow \overline{A}.B$
		✓	✓				✓	✓	−10−	$S \rightarrow B.\overline{C}$

Each prime implicant is identified by a letter. Each prime implicant is then examined one by one and the terms it can account for are ticked as shown. The next step is to write a product-of-sums expression using the prime implicants to account for all the terms. In the present illustration, it is given as follows.

$$(P+Q).(P).(R+S).(P+Q+R+S).(R).(P+R).(Q).(S).(Q+S)$$

Obvious simplification reduces this expression to PQRS which can be interpreted to mean that all prime implicants, that is, P, Q, R and S, are needed to account for all the original terms. Therefore, the minimized expression = $\overline{A}.D+\overline{C}.D+\overline{A}.B+B.\overline{C}$.

What has been described above is the formal method of determining the optimum set of prime implicants. In most of the cases where the prime implicant table is not too complex, the exercise can be done even intuitively. The exercise begins with identification of those terms that can be accounted for by only a single prime implicant. In the present example, 0011, 0110, 1001 and 1100 are such terms. As a result, P, Q, R and S become the essential prime implicants. The next step is to find out if any terms have not been covered by the essential prime implicants. In the present case, all terms have been covered by essential prime

implicants. In fact, all prime implicants are essential prime implicants in the present example. As another illustration, let us consider a product-of-sums expression given by

(A'+B'+C'+D') (A'+B'+C'+D) (A'+B'+C+D') (A+B'+C+D')

The procedure is similar to that described for the case of simplification of sum-of-products expressions.

The resulting tables leading to identification of prime implicants are as follows:

A	B	C	D
0	1	0	1
0	1	1	1
1	1	0	1
1	1	1	0
1	1	1	1

A	B	C	D	
0	1	0	1	✓
0	1	1	1	✓
1	1	0	1	✓
1	1	1	0	✓
1	1	1	1	✓

A	B	C	D	
0	1	–	1	✓
–	1	0	1	✓
–	1	1	1	✓
1	1	–	1	✓
1	1	1	–	*

A	B	C	D	
–	1	–	1	*

The prime implicant table is constructed after all prime implicants have been identified to look for the optimum set of prime implicants needed to account for all the original terms. The prime implicant table shows that both the prime implicants are the essential ones:

0101	0111	1101	1110	1111	Prime implicants
			✓	✓	111–
✓	✓	✓		✓	–1–1

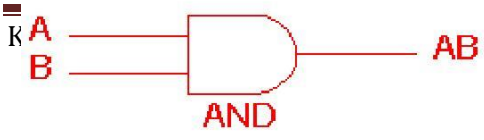
The minimized expression = $(\overline{A}+\overline{B}+\overline{C}).(\overline{B}+\overline{D})$.

UNIT-II LOGIC GATES AND COMBINATIONAL CIRCUITS LOGIC GATES

Logic gates

Digital systems are constructed using logic gates. Logic gates are digital circuits which has one or more inputs and a single output and which performs arithmetic and logical operations like addition, multiplication and inversion. There are three basic gates, which are called as AND, OR and NOT gates. The derived gates are gates which are derived using basic gates. Examples of derived gates are NAND, NOR, EXOR and EXNOR gates. The basic operations of all the gates are described below along with truth tables.

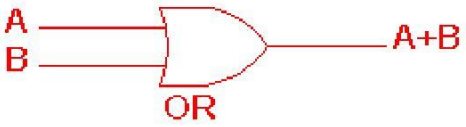
AND gate



2 Input AND gate		
A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

The AND gate is an electronic circuit that gives a **high** output (1) only if **all** its inputs are high. A dot (.) is used to show the AND operation i.e. A.B. Bear in mind that this dot is sometimes omitted i.e. AB

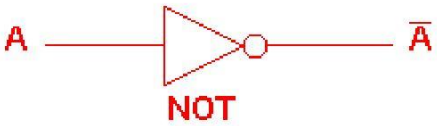
OR gate



2 Input OR gate		
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

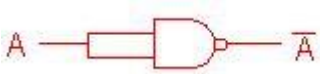
The OR gate is an electronic circuit that gives a high output (1) if **one or more** of its inputs are high. A plus (+) is used to show the OR operation.

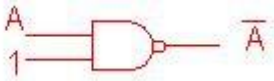
NOT gate



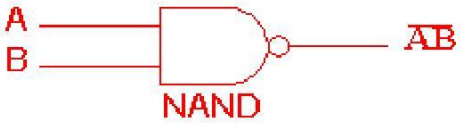
NOT gate	
A	$\overline{A}$
0	1
1	0

The NOT gate is an electronic circuit that produces an inverted version of the input at its output. It is also known as an *inverter*. If the input variable is A, then the inverted output is complement or inversion of A. This is also shown as A', or A with a bar over the top, as shown at the outputs. The diagrams below show two ways that the NAND logic gate can be configured to produce a NOT gate. It can also be done using NOR logic gates in the same way.





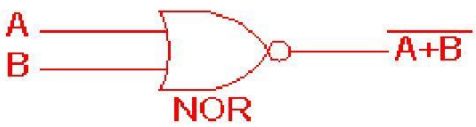
NAND gate



2 Input NAND gate		
A	B	$\overline{A.B}$
0	0	1
0	1	1
1	0	1
1	1	0

This is a NOT-AND gate which is equal to an AND gate followed by a NOT gate. The outputs of all NAND gates are high if **any** of the inputs are low. The symbol is an AND gate with a small circle on the output. The small circle represents inversion.

NOR gate

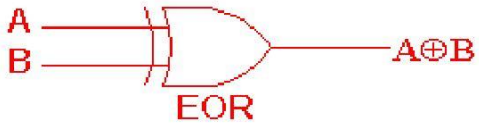


2 Input NOR gate		
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

This is a NOT-OR gate which is equal to an OR gate followed by a NOT gate. The outputs of all NOR gates are low if **any** of the inputs are high.

The symbol is an OR gate with a small circle on the output. The small circle represents inversion.

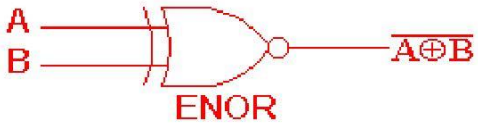
EXOR gate



2 Input EXOR gate		
A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

The Excusive OR gate is a circuit which will give high output if both the inputs are high or low. An encircled plus sign(⊕), is used to show the EX-OR operation.

EXNOR gate



2 Input EXNOR gate		
A	B	$\overline{A⊕B}$
0	0	1
0	1	0
1	0	0
1	1	1

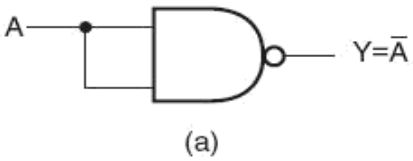
The '**Exclusive-NOR**' gate circuit does the opposite to the EOR gate. It will give a low output if **either, but not both**, of its two inputs are high. The symbol is an EXOR gate with a small circle on the output. The small circle represents inversion. The NAND and NOR gates are called *universal functions* since with either one the AND and OR functions and NOT can be generated.

Note:

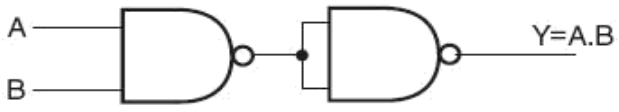
A function in *sum of products* form can be implemented using NAND gates by replacing all AND and OR gates by NAND gates.A function in *product of sums* form can be implemented using NOR gates by replacing all AND and OR gates by NOR gates.

Universal Gates

NAND, & NOR are called as Universal gates as all the other gates can be constructed using NAND and NOR

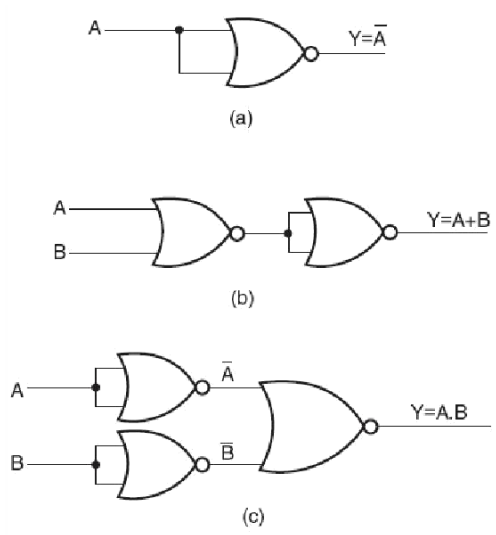


(a)



(b)

Implementation of basic logic gates using only NAND gates. a connection for open collector NAND gates. The output in this case would be



$$Y = \overline{AB}.\overline{CD}.\overline{EF}$$

Combinational Circuits

A combinational circuit neither contains a periodic clock signal nor has any provisions for storage. There are no feedbacks involved and the output at all time is dependent on the inputs provided. The name combinational is derived from the combinations of logic gates used for such circuits.A sequential circuit involves feedback and has memory (so it is employed for designing RAM). It also has a periodic clock signal and hence the output is also a function of time in addition to being a function of inputs and previous outputs. The name sequential is derived as the output is produced in sequences as the clock circuit enables and disables the functioning. (A latch is also a sequential circuit but has no clock signal and hence is a special case. It is also the basic building block of any sequential circuit.)

Designing Combinational Circuits

In general we have to do following steps:

1. **Problem description**
2. **Input/output of the circuit**

3. **Define truth table**
4. **Simplification for each output**
5. **Draw the circuit**

ADDERS

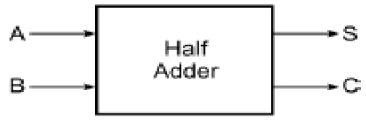
Half-Adder

A half-adder is an arithmetic circuit block that can be used to add two bits. Such a circuit thus has two inputs that represent the two bits to be added and two outputs, with one producing the SUM output and the other producing the CARRY. Figure 3.2 shows the truth table of a half-adder, showing all possible input combinations and the corresponding outputs.

The Boolean expressions for the SUM and CARRY outputs are given by the equations below

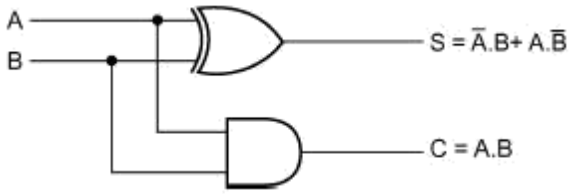
$$\text{SUM } S = A.\overline{B} + \overline{A}.B$$
$$\text{CARRY } C = A.B$$

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



Truth Table of Half Adder

An examination of the two expressions tells that there is no scope for further simplification. While the first one representing the SUM output is that of an EX-OR gate, the second one representing the CARRY output is that of an AND gate. However, these two expressions can certainly be represented in different forms using various laws and theorems of Boolean algebra to illustrate the flexibility that the designer has in hardware-implementing as simple a combinational function as that of a half-adder.

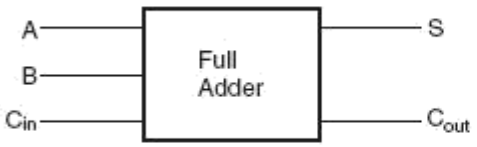


Logic Implementation of Half Adder

Although the simplest way to hardware-implement a half-adder would be to use a two-input EX-OR gate for the SUM output and a two-input AND gate for the CARRY output, as shown in Fig. 3.3, it could also be implemented by using an appropriate arrangement of either NAND or NOR gates.

Full Adder

A full adder circuit is an arithmetic circuit block that can be used to add three bits to produce a SUM and a CARRY output. Such a building block becomes a necessity when it comes to adding binary numbers with a large number of bits. The full adder circuit overcomes the limitation of the half-adder, which can be used to add two bits only. Let us recall the procedure for adding larger binary numbers. We begin with the addition of LSBs of the two numbers. We record the sum under the LSB column and take the carry, if any, forward to the next higher column bits. As a result, when we add the next adjacent higher column bits, we would be required to add three bits if there were a carry from the previous addition. We have a similar situation for the other higher column bits. Also until we reach the MSB. A full adder is therefore essential for the hardware implementation of an adder circuit capable of adding larger binary numbers. A half-adder can be used for addition of LSBs only.



A	B	C_{in}	SUM (S)	C_{out}
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 of Full Adder

Figure shows the truth table of a full adder circuit showing all possible input combinations and corresponding outputs. In order to arrive at the logic circuit for hardware implementation of a full adder, we will firstly write the Boolean expressions for the two output variables, that is, the SUM and CARRY outputs, in terms of input variables. The Boolean expressions for the two output variables are given in Equation below for the SUM output (S) and in above Equation for the CARRY output (Cout):

$$S = \overline{A}.\overline{B}.C_{in} + \overline{A}.B.\overline{C}_{in} + A.\overline{B}.\overline{C}_{in} + A.B.C_{in}$$

$$C_{out} = \overline{A}.B.C_{in} + A.\overline{B}.C_{in} + A.B.\overline{C}_{in} + A.B.C_{in}$$

The next step is to simplify the two expressions. We will do so with the help of the Karnaugh mapping technique. Karnaugh maps for the two expressions are given in Fig. 3.5(a) for the SUM output and Fig. 3.5(b) for the CARRY output.

$$C_{out} = B.C_{in} + A.B + A.C_{in}$$

Figure shows the logic circuit diagram of the full adder. A full adder can also be seen to comprise two half-adders and an OR gate. The expressions for SUM and CARRY outputs can be rewritten as follows:

$$S = \overline{C}_{in}.(\overline{A}.B + A.\overline{B}) + C_{in}.(A.B + \overline{A}.\overline{B})$$

$$S = \overline{C}_{in}.(\overline{A}.B + A.\overline{B}) + C_{in}.\overline{(\overline{A}.B + A.\overline{B})}$$

Similarly, the expression for CARRY output can be rewritten as follows:

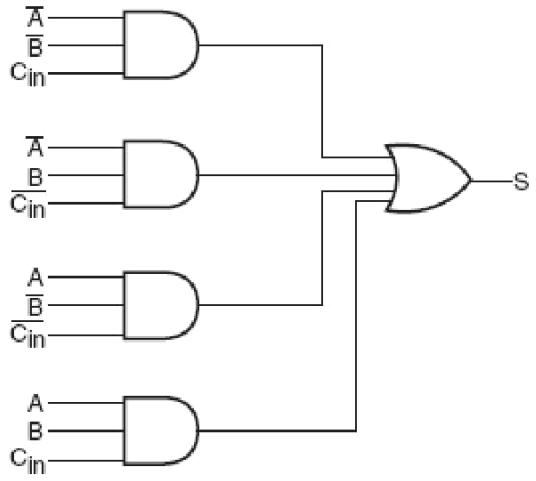
$$C_{out} = B.C_{in}.(A + \overline{A}) + A.B + A.C_{in}.(B + \overline{B})$$

$$= A.B + A.B.C_{in} + \overline{A}.B.C_{in} + A.B.C_{in} + A.\overline{B}.C_{in} = A.B + A.B.C_{in} + \overline{A}.B.C_{in} + A.\overline{B}.C_{in}$$

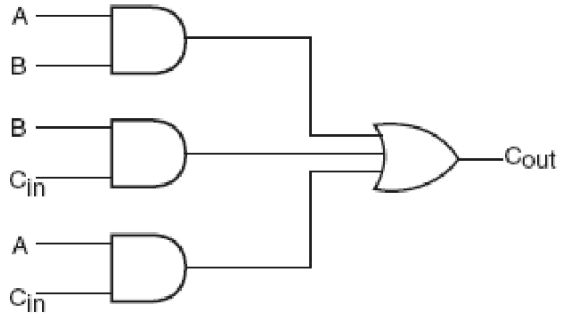
$$= A.B.(1 + C_{in}) + C_{in}.(\overline{A}.B + A.\overline{B})$$

Karnaugh Map for the sum and carry out of a full adder

$$C_{out} = A.B + C_{in}.(\overline{A}.B + A.\overline{B})$$

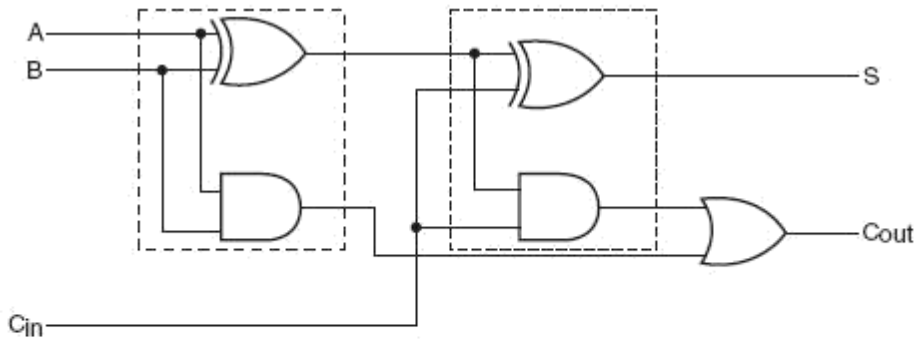


(a)

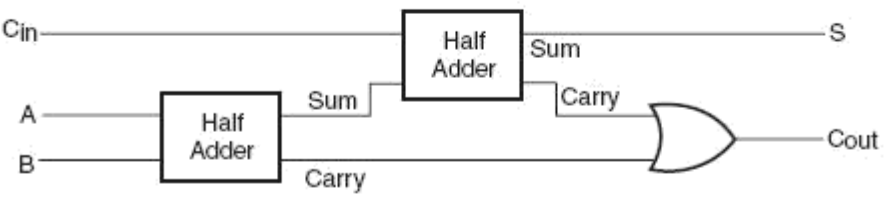


(b)

Logic circuit diagram of full adder

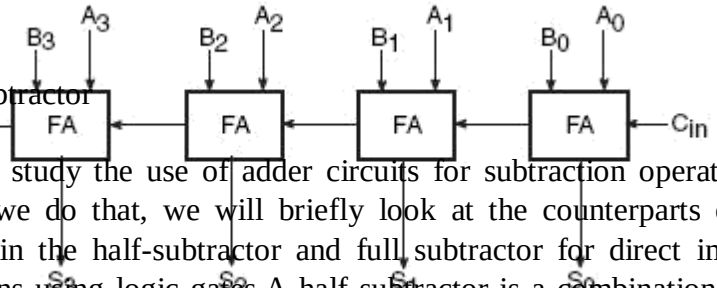


(a)



(b)

Logic Implementation of a full adder with Half Adders



Half-Subtractor

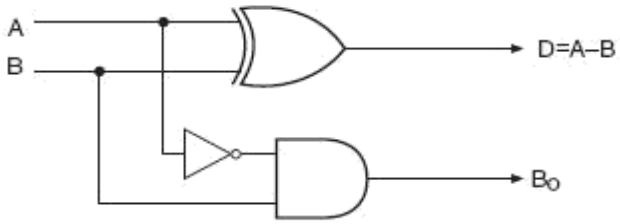
We will study the use of adder circuits for subtraction operations in the following pages. Before we do that, we will briefly look at the counterparts of half-adder and full adder circuits in the half-subtractor and full subtractor for direct implementation of subtraction operations using logic gates.A half-subtractor is a combinational circuit that can be used to subtract one binary digit from another to produce a DIFFERENCE output and a BORROW output. The BORROW output here specifies whether a _1' has been borrowed to perform the subtraction. The truth table of a half-subtractor, as shown in Fig. 3.9, explains this further. The Boolean expressions for the two outputs are given by the equations

$$D = \overline{A}.B + A.\overline{B}$$

$$B_o = \overline{A}.B$$

A	B	D	B_o
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Half Subtractor



Logic Diagram of a Half Subtractor

Full Subtractor

A full subtractor performs subtraction operation on two bits, a minuend and a subtrahend, and

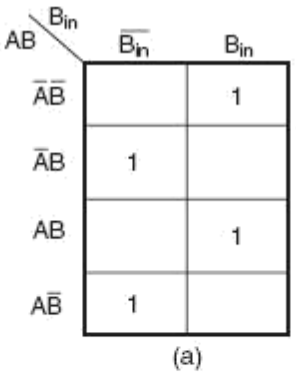
also takes into consideration whether a _1' has already been borrowed by the previous adjacent lower minuend bit or not. As a result, there are three bits to be handled at the input of a full subtractor, namely the two bits to be subtracted and a borrow bit designated as Bin . There are two outputs, namely the DIFFERENCE output D and the BORROW output Bo
The Boolean expressions for the two output variables are given by the equations

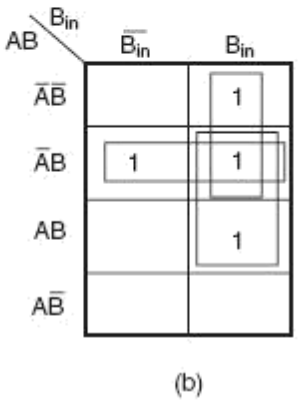
$$D = \overline{A}.\overline{B}.B_{in} + \overline{A}.B.\overline{B}_{in} + A.\overline{B}.\overline{B}_{in} + A.B.B_{in}$$

$$B_o = \overline{A}.\overline{B}.B_{in} + \overline{A}.B.\overline{B}_{in} + \overline{A}.B.B_{in} + A.B.B_{in}$$

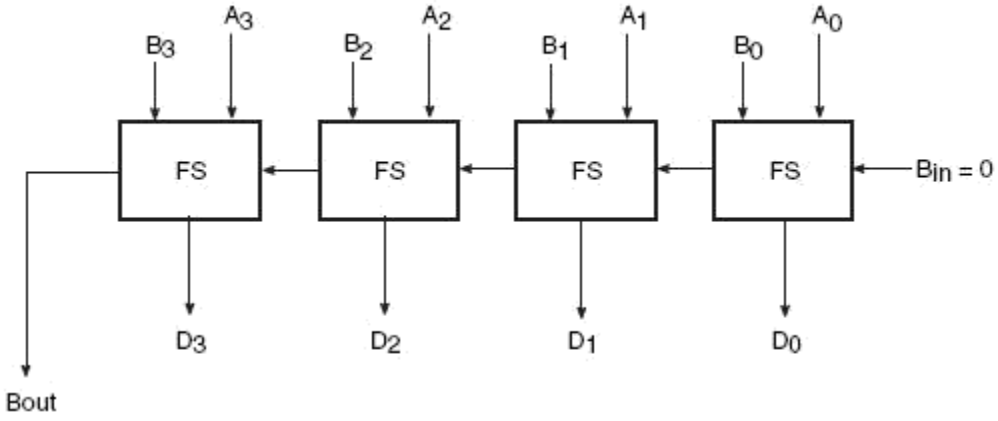
Minuend (A)	Subtrahend (B)	Borrow In (B_{in})	Difference (D)	Borrow Out (B_O)
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Truth Table of Full Subtractor





K Maps for Difference and Borrow outputs. Figure


A3
B3
A2
B2
A1
B1
A0
B0
FS
FS
FS
FS
Bin = 0
D3
D2
D1
D0
Bout

Adder Subtractor

- The addition and subtraction can be combined into one circuit with one common binary adder (see next slide).
- The mode M controls the operation. When M=0 the circuit is an adder when M=1 the circuit is subtractor. It can be don by using exclusive-OR for each Bi and M. Note that $1 \oplus x = x'$ and $0 \oplus x = x$

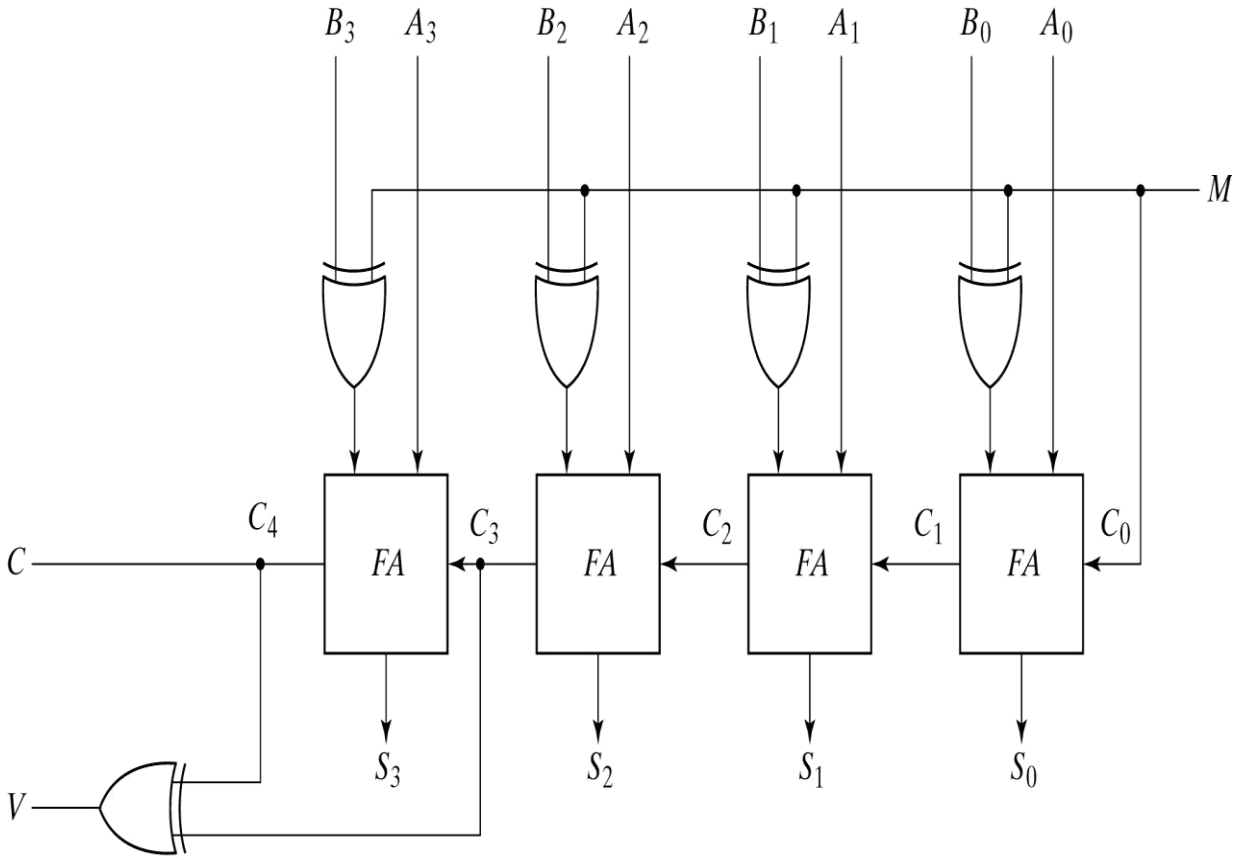


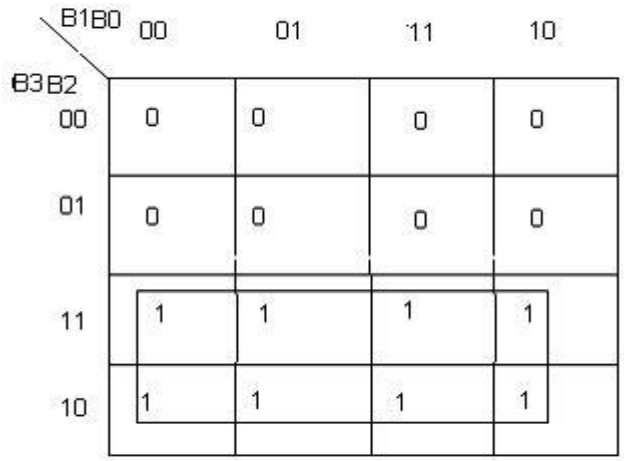
Fig. 4-13 4-Bit Adder Subtractor

Binary to Gray converter

In this circuit we'll convert BINARY numbers to GRAY numbers. Following is the truth table for it:

	B3	B2	B1	B0	G3	G2	G1	G0
0.	0	0	0	0	0	0	0	0
1.	0	0	0	1	0	0	0	1
2.	0	0	1	0	0	0	1	1
3.	0	0	1	1	0	0	1	0
4.	0	1	0	0	0	1	1	0
5.	0	1	0	1	0	1	1	1
6.	0	1	1	1	0	1	0	0
8.	1	0	0	0	1	1	0	0
9.	1	0	0	1	1	1	0	1
10.	1	0	1	0	1	1	1	1
11.	1	0	1	1	1	1	1	0
12.	1	1	0	0	1	0	1	0
13.	1	1	0	1	1	0	1	1
14.	1	1	1	0	1	0	0	1
15.	1	1	1	1	1	0	0	0

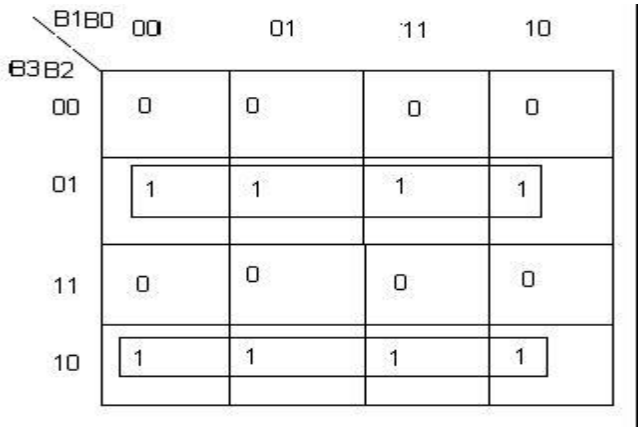
K-MAP FOR G3:



B3B2 \ B1B0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

Equation for G3= B3

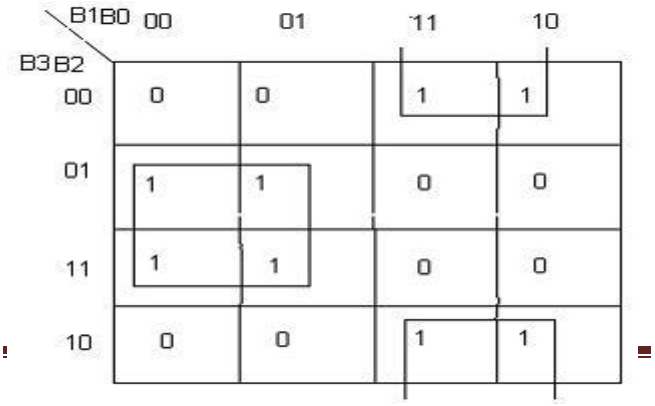
K-MAP FOR G2:



B3B2 \ B1B0	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

Equation for G2= B3' B2 + B3 B2'= B3 XOR B2

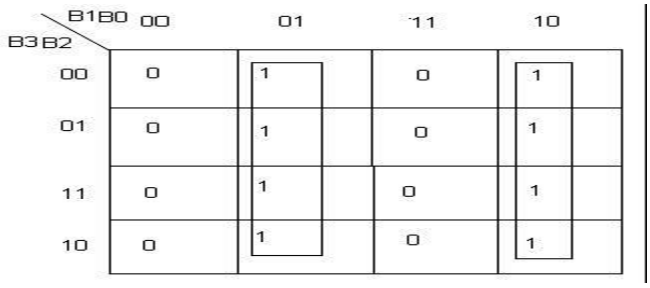
K-MAP FOR G1:



B3B2 \ B1B0	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	1	1	0	0
10	0	0	1	1

Equation for G1= B1' B2 + B1 B2'= B1 XOR B2

K-MAP FOR G0:



Equation for G0= B1' B0 + B1 B0'= B1 XOR B0

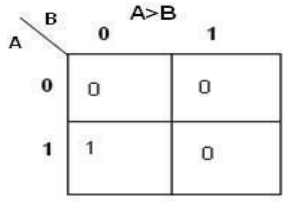
COMPARATORS

- It is a combinational circuit that compares to numbers and determines their relative magnitude .
- For example to design a comparator for 2 bit binary numbers A (A1A0) and B (B1B0) we do the following steps:

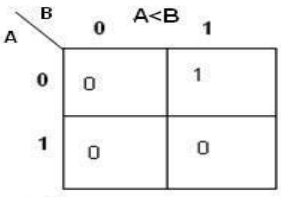
1-bit comparator: Let's begin with 1 bit comparator and from the name we can easily make out that this circuit would be used to compare 1 bit binary numbers.f we list all the input combinations at the input then we get the following table describing the corresponding outputs.

A	**B**	**f (A>B)**	**f (A=B)**	**f (A<B)**
0	**0**	**0**	**1**	**0**
1	**0**	**1**	**0**	**0**
0	**1**	**0**	**0**	**1**
1	1	0	1	0

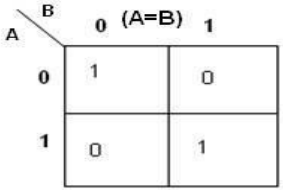
And now we find the equations using K-maps each for f (A>B), f (A=B) and f (A<B) as follow:



Equation is $A>B = A.\overline{B}$



Equation is $A<B = \overline{A}.B$



The equation is $f(A=B) = \overline{A}.\overline{B} + A.B$
$= A \text{ XNOR } B$

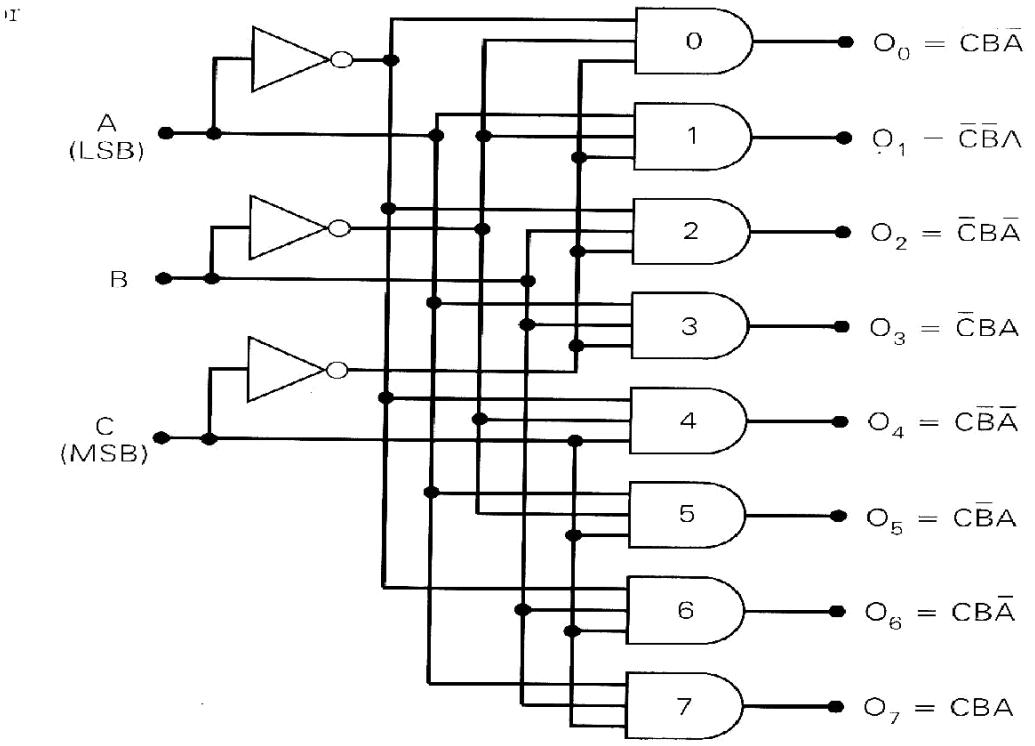
or we can write the equation for $f(A=B)$ as $\overline{A.\overline{B} + \overline{A}.B} = \overline{f(A>B)+f(A<B)}$

BCD Adder

Number	C	S8	S4	S2	S1
0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	0	1	0
3	0	0	0	1	1
4	0	0	1	0	0
5	0	0	1	0	1
6	0	0	1	1	0
7	0	0	1	1	1
8	0	1	0	0	0
9	0	1	0	0	1
Number	C	S8	S4	S2	S1
10	1	0	0	0	0
11	1	0	0	0	1
12	1	0	0	1	0
13	1	0	0	1	1
14	1	0	1	0	0
15	1	0	1	0	1
16	1	0	1	1	0
17	1	0	1	1	1
18	1	1	0	0	0
19	1	1	0	0	1

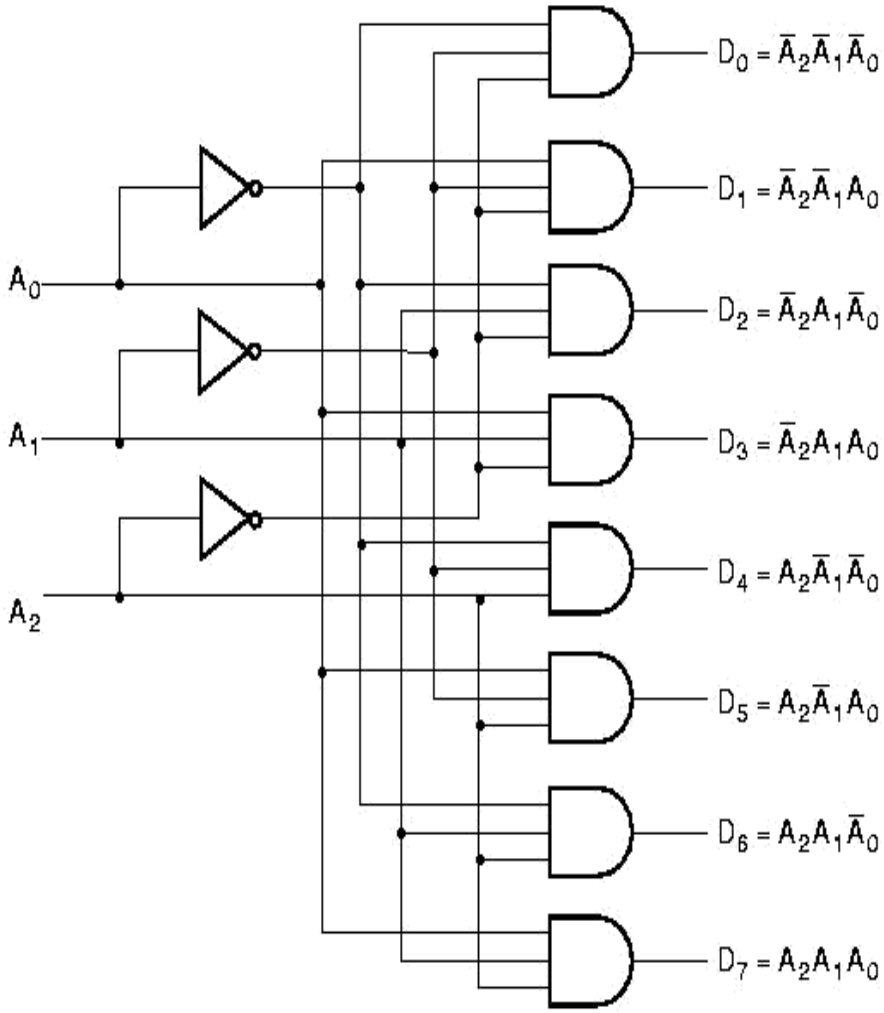
Decoders

- **Accepts a value and decodes it**
  - **Output corresponds to value of *n* inputs**
- **Consists of:**
  - Inputs (n)
  - Outputs (2n , numbered from 0 □ 2n - 1)
  - Selectors / Enable (active high or active low)



C	B	A	O_7	O_6	O_5	O_4	O_3	O_2	O_1	O_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

Octal Numbar	D0	D1	D2	D3	D4	D5	D6	D7	a	b	c
0	1	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	1
2	0	0	1	0	0	0	0	0	0	1	0
3	0	0	0	1	0	0	0	0	0	1	1
4	0	0	0	0	1	0	0	0	1	0	0
5	0	0	0	0	0	1	0	0	1	0	1
6	0	0	0	0	0	0	1	0	1	1	0
7	0	0	0	0	0	0	0	1	1	1	1



Encoder

- Perform the inverse operation of a decoder
- 2n (or less) input lines and n output lines

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_2	A_1	A_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

Can be implemented with 3 OR gates A0

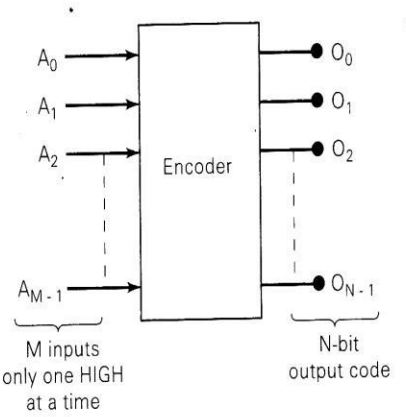
= D1 + D3 + D5 + D7;

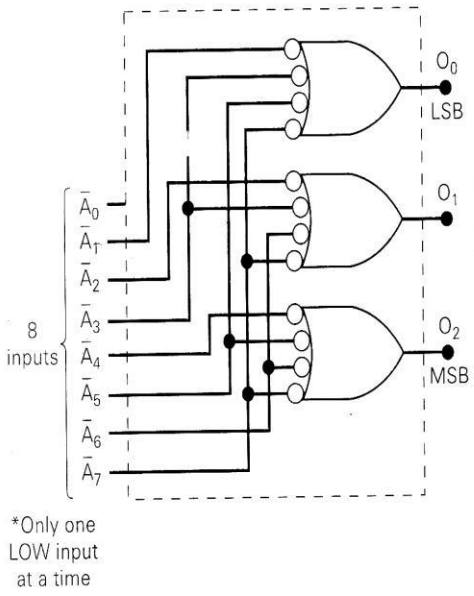
A1 = D2 + D3 + D6 + D7;

A2 = D4 + D5 + D6 + D7;

If more than 2 inputs are active we need to use priority encoder (priority for inputs)

FIGURE 9-12 General encoder diagram.



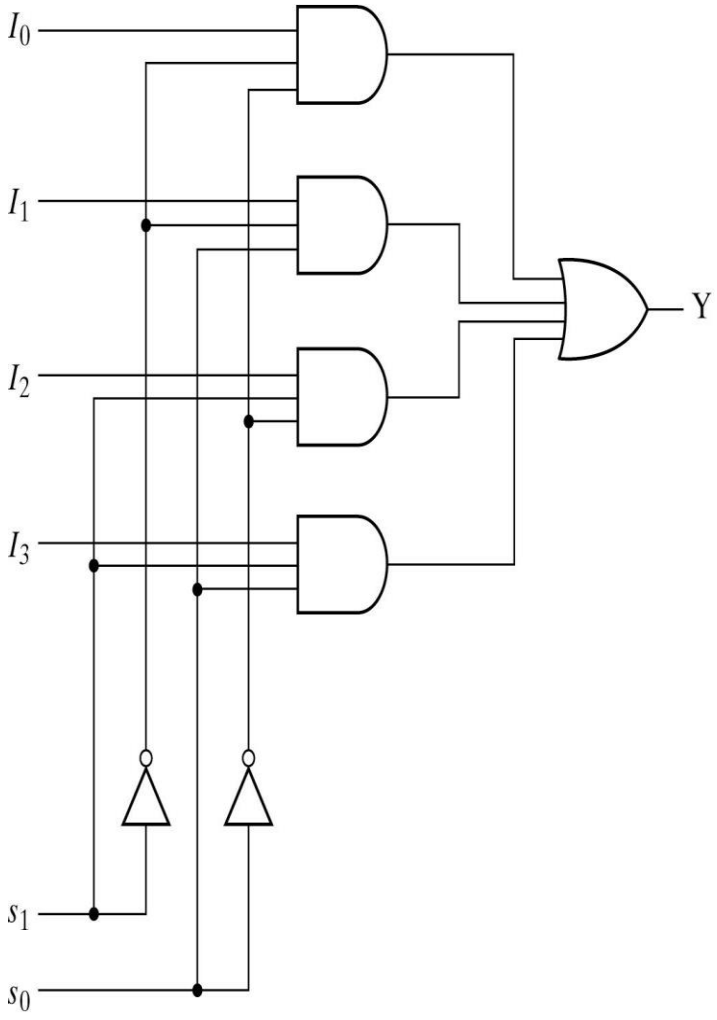


Inputs								Outputs		
$\bar{A}_0$	$\bar{A}_1$	$\bar{A}_2$	$\bar{A}_3$	$\bar{A}_4$	$\bar{A}_5$	$\bar{A}_6$	$\bar{A}_7$	O_2	O_1	O_0
X	1	1	1	1	1	1	1	0	0	0
X	0	1	1	1	1	1	1	0	0	1
X	1	0	1	1	1	1	1	0	1	0
X	1	1	0	1	1	1	1	0	1	1
X	1	1	1	0	1	1	1	1	0	0
X	1	1	1	1	0	1	1	1	0	1
X	1	1	1	1	1	0	1	1	1	0
X	1	1	1	1	1	1	0	1	1	1

not corrected.

Multiplexer

- It is a combinational circuit that selects binary information from one of the input lines and directs it to a single output line
- Usually there are 2n input lines and n selection lines whose bit combinations determine which input line is selected
- For example for 2-to-1 multiplexer if selection S is zero then I0 has the path to output and if S is one I1 has the path to output



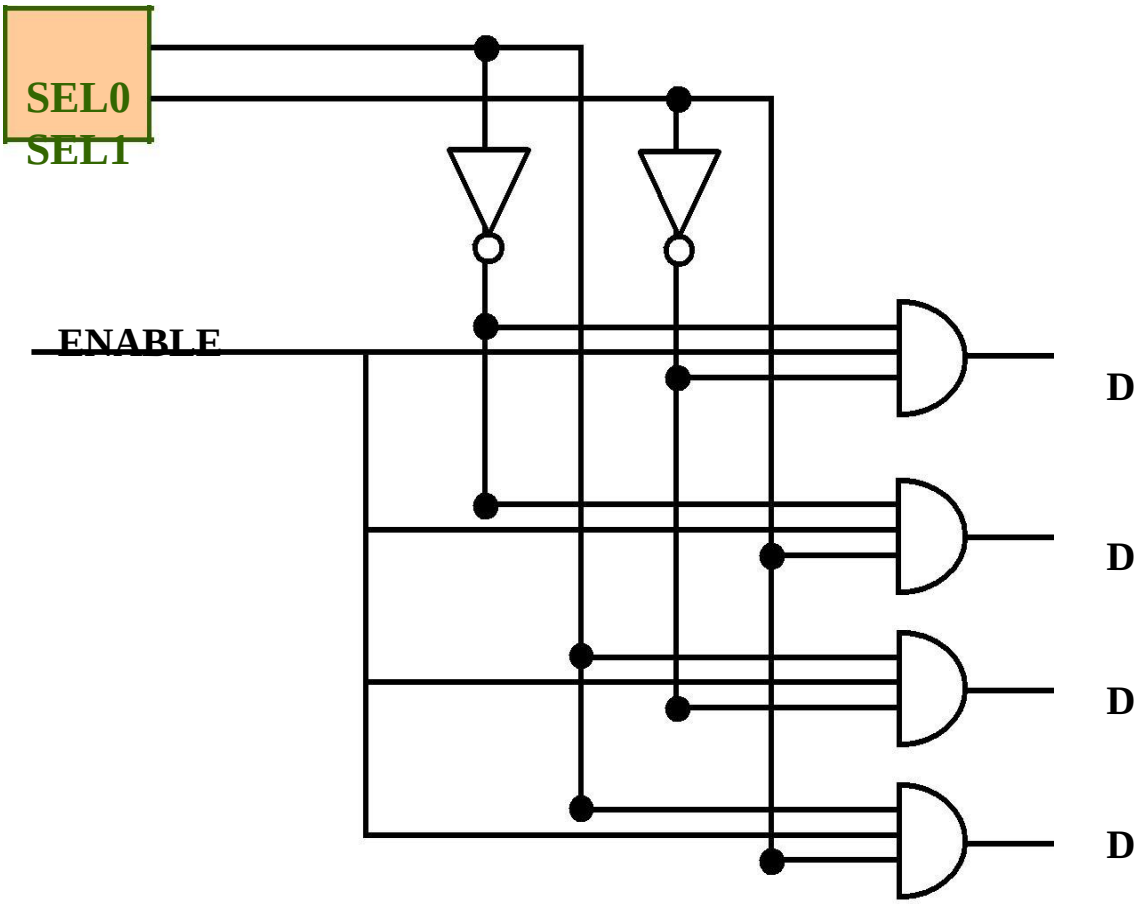
(a) Logic diagram

s_1	s_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

(b) Function table

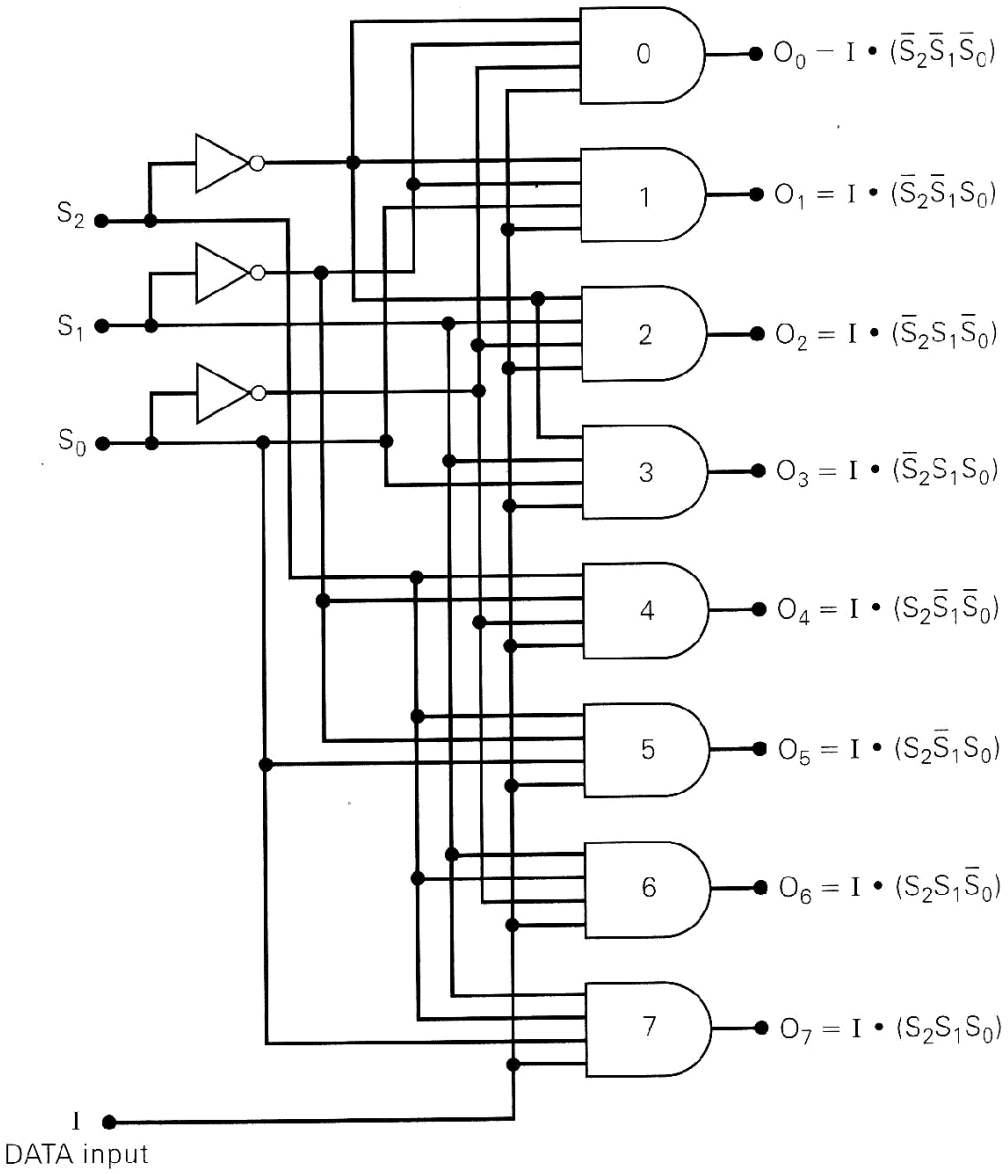
Fig. 4-25 4-to-1-Line Multiplexer

Demultiplexer



ENBL	SEL1	SEL2	D00	D01	D10	D11
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

1 to 8 line Demultiplexer

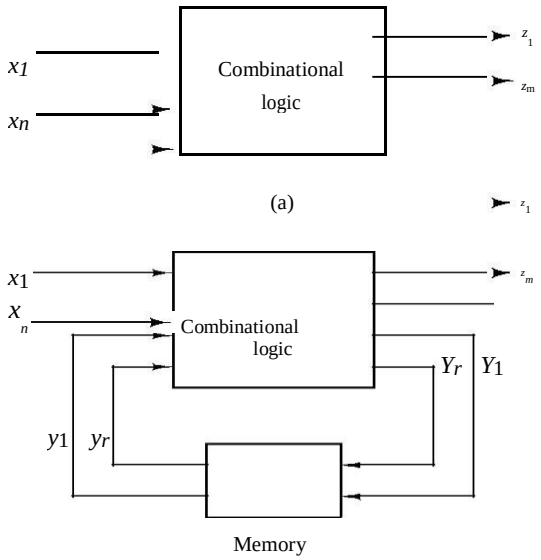


SELECT code			OUTPUTS							
S_2	S_1	S_0	O_7	O_6	O_5	O_4	O_3	O_2	O_1	O_0
0	0	0	0	0	0	0	0	0	0	**I**
0	0	1	0	0	0	0	0	0	**I**	0
0	1	0	0	0	0	0	0	**I**	0	0
0	1	1	0	0	0	0	**I**	0	0	0
1	0	0	0	0	0	**I**	0	0	0	0
1	0	1	0	0	**I**	0	0	0	0	0
1	1	0	0	**I**	0	0	0	0	0	0
1	1	1	**I**	0	0	0	0	0	0	0

Note: I is the data input

UNIT-III SEQUENTIAL CIRCUIT

Sequential Circuits

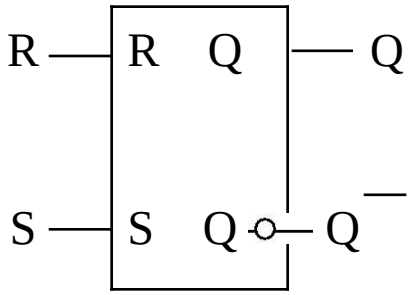


Flip Flop

A circuit that changes from 1 to 0 or from 0 to 1 when current is applied. It is one bit storage location.

SR Flip-Flop

Graphical Symbol



Characteristic table

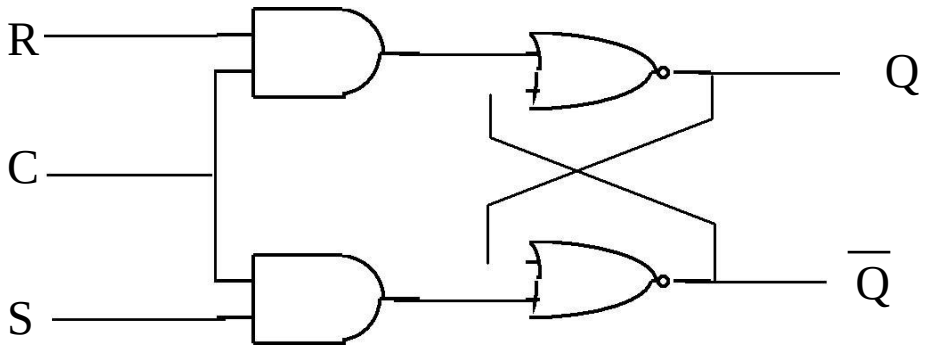
C	D	Q

0	x	No change
1	0	0 (reset)
1	1	1 (set)

Characteristic Equation

$Q(t+1)=S+R^1Q$

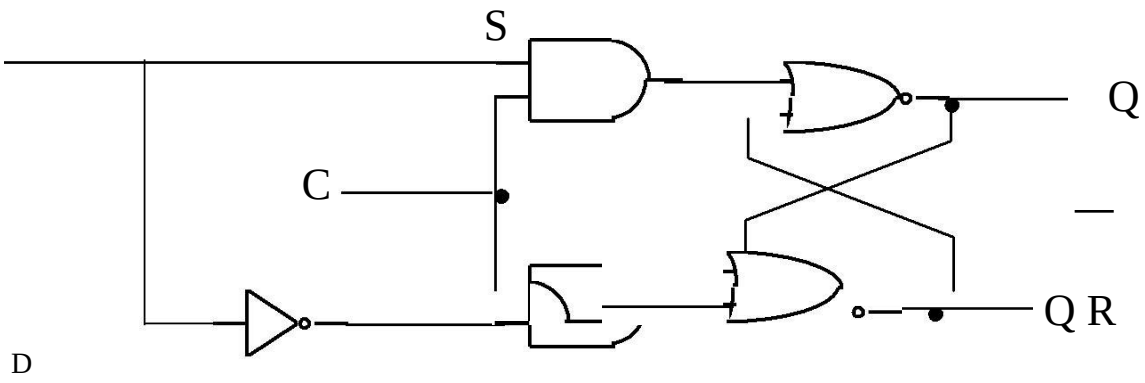
Logic diagram



EXCITATION TABLE

Q	**$Q_{(next)}$**	**S**	**R**
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

D Flip-Flop



Characteristic table

D	Q(t+1)	Operation
0	0	Reset
1	1	Set

Characteristic Equation

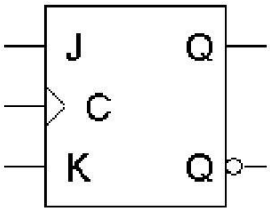
Q(t+1) = D

EXCITATION TABLE

Q	**$Q_{(next)}$**	**D**
0	0	0
0	1	1
1	0	0
1	1	1

JK Flip-Flop

Graphical Symbol

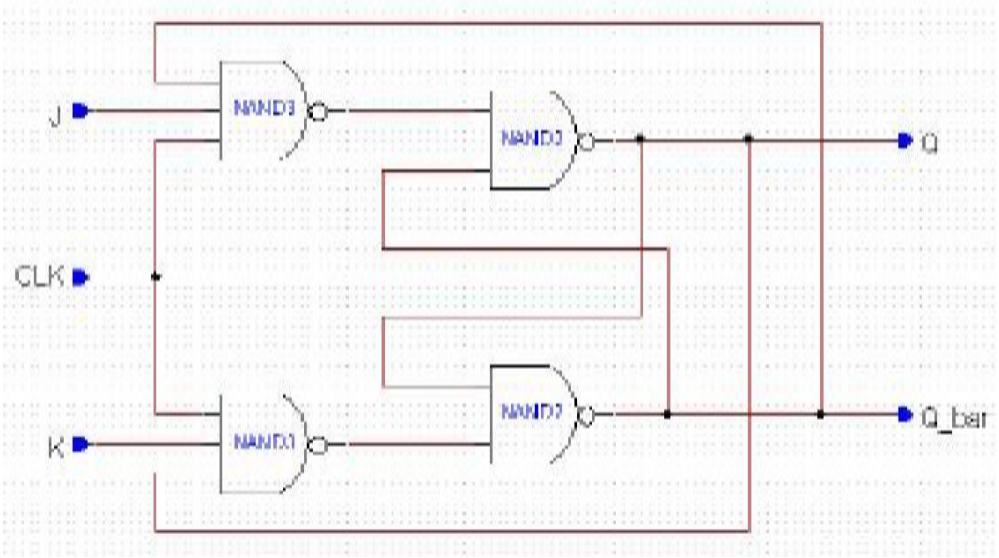


J	K	Q(t+1)	Operation
0	0	Q(t)	No change
0	1	0	Reset
1	0	1	Set
1	1	Q'(t)	Complement

Characteristic Equation

$Q(t+1) = K'Q(t) + JQ'(t)$

Logic Diagram

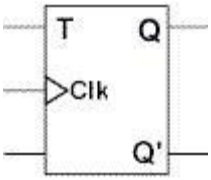


EXCITATION TABLE

Q	$Q_{(next)}$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

T Flip-Flop

Graphical Symbol



Characteristic Table

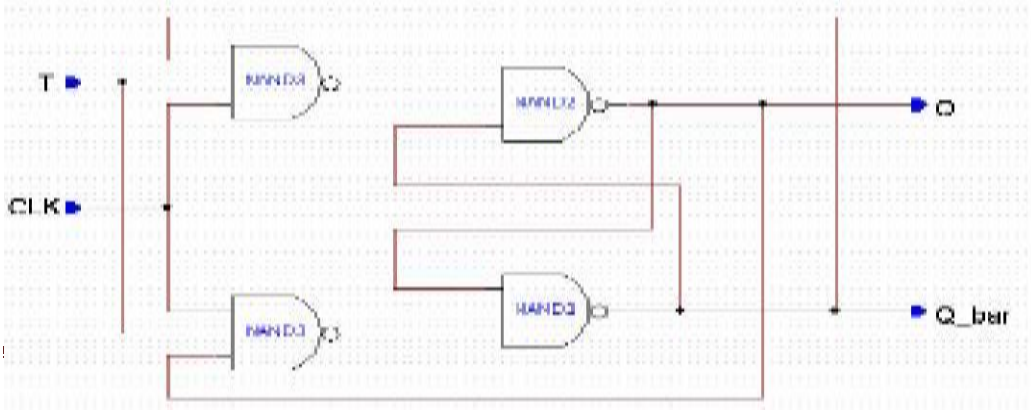
T	Q(t+1)	Operation
0	Q(t)	No change
1	Q'(t)	Complement

Characteristic Equation

Q(t+1) = T'Q(t) + TQ'(t)

= T ⊕ Q(t)

Logic Diagram

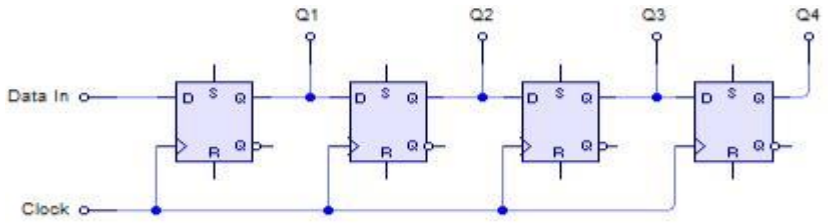


Q	$Q_{(next)}$	T
0	0	0
0	1	1
1	0	1
1	1	0

EXCITATION TABLE

Shift Register

A **shift register** is a cascade of Flip flops, sharing the same clock, which has the output of any one but the last flip-flop connected to the "data" input of the next one in the chain, resulting in a circuit that shifts by one position the one-dimensional "bit array" stored in it, *shifting in* the data present at its input and *shifting out* the last bit in the array, when enabled to do so by a transition of the clock input. More generally, a **shift register** may be multidimensional, such that its "data in" input and stage outputs are themselves bit arrays: this is implemented simply by running several shift registers of the same bit-length in parallel.



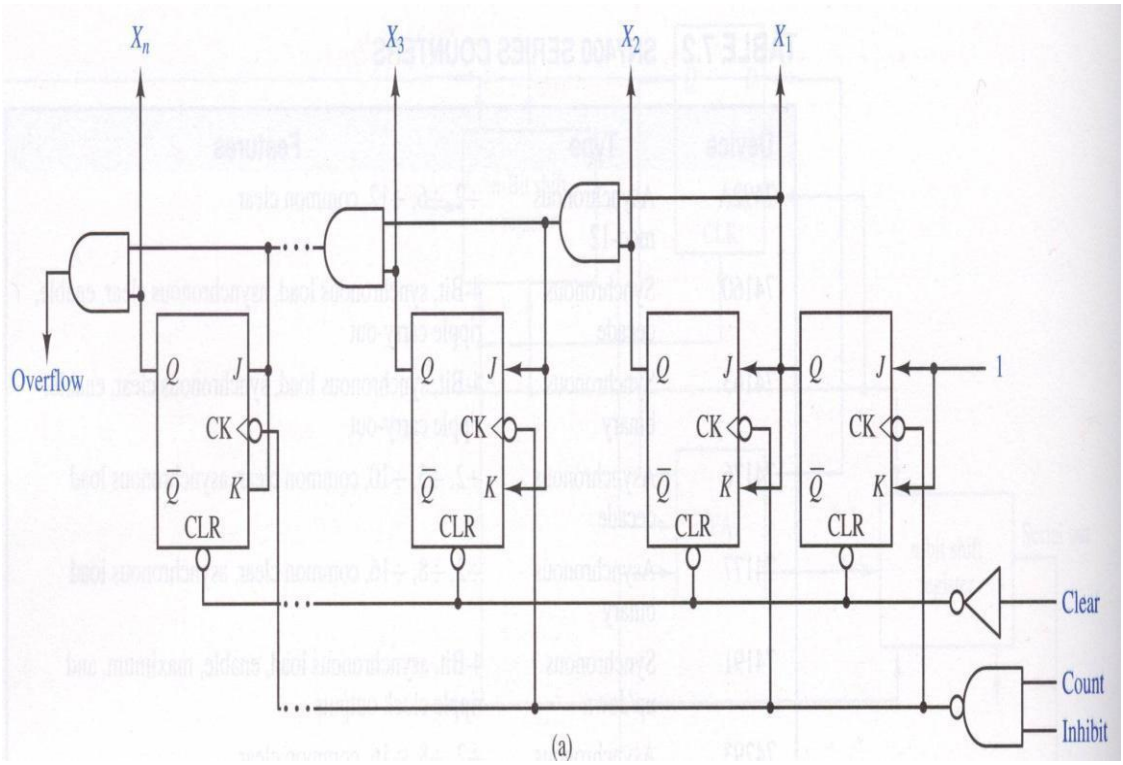
Counters

In digital logic and computing, a **counter** is a device which stores (and sometimes displays) the number of times a particular event or process has occurred, often in relationship to a clock signal.

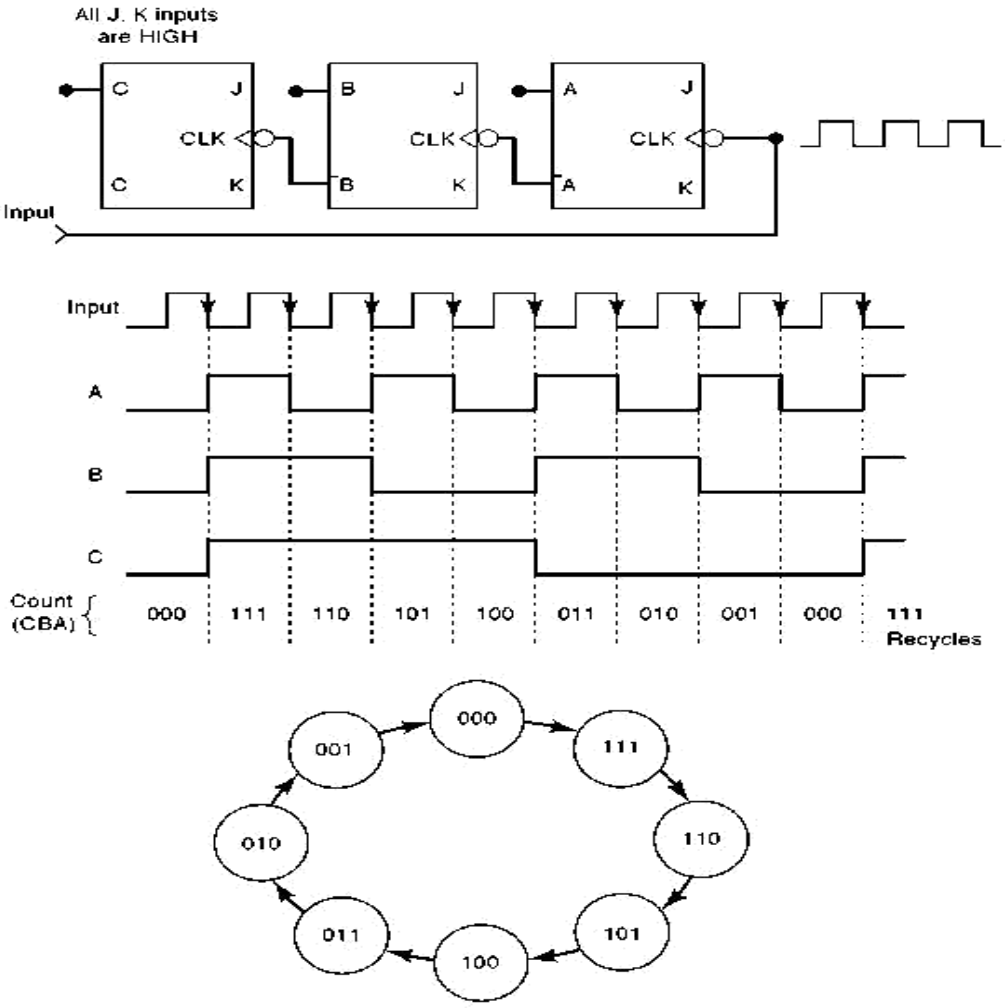
- Asynchronous (ripple) counter – changing state bits are used as clocks to subsequent state flip-flops
- Synchronous counter – all state bits change under control of a single clock
- Decade counter – counts through ten states per stage
- Up–down counter – counts both up and down, under command of a control input
- Ring counter – formed by a shift register with feedback connection in a ring

- Johnson counter – a *twisted* ring counter
- Cascaded counter

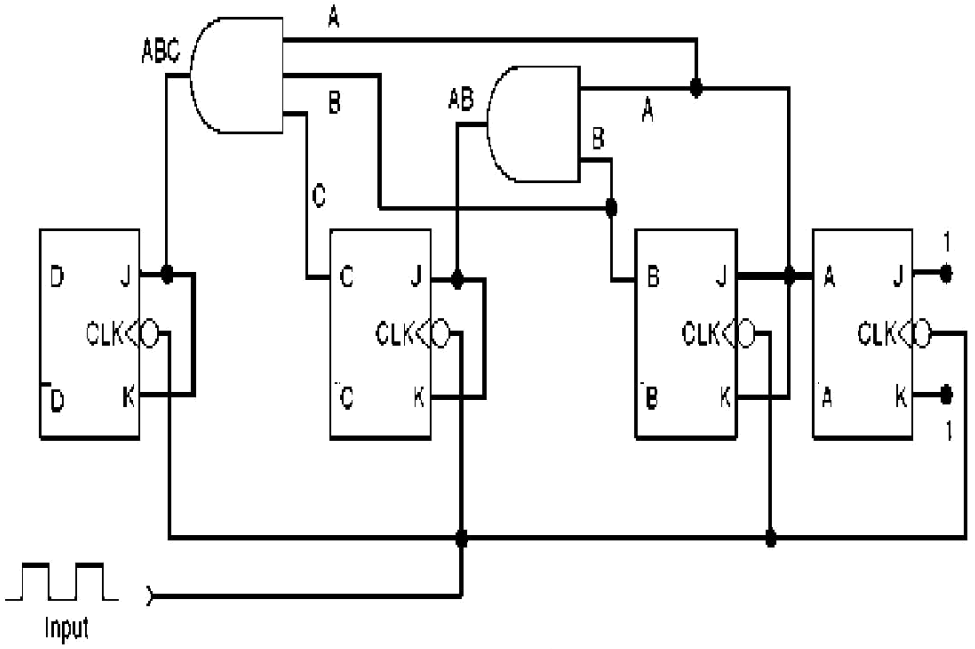
Asynchronous Up-counter (Ripple Counter)



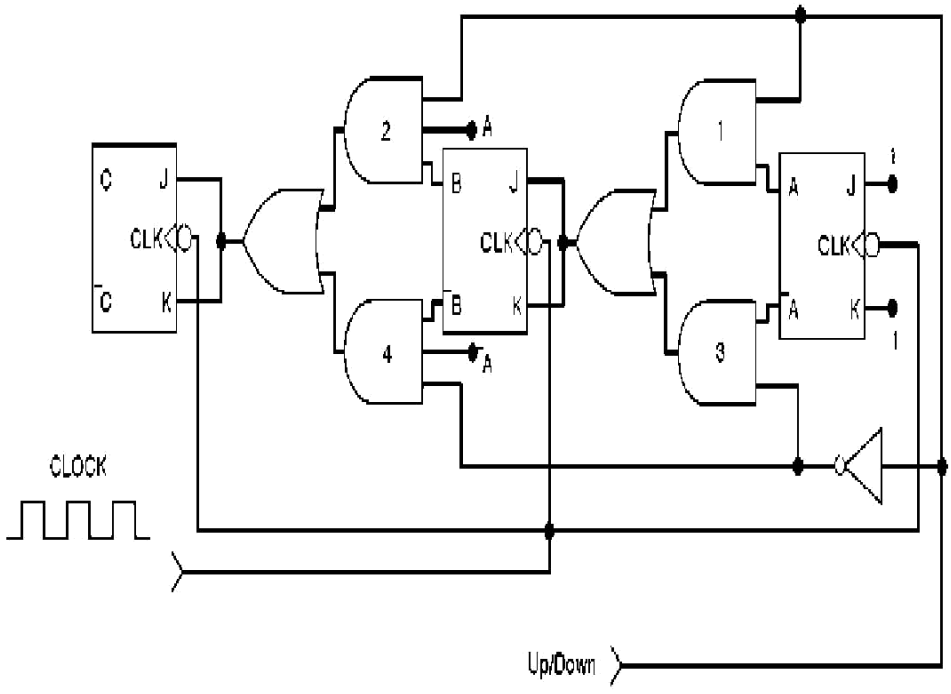
Asynchronous Down Counter



Synchronous (Parallel) Counter


ABC
A
B
C
AB
A
B
D
J
CLK
D
K
C
J
CLK
C
K
B
J
CLK
B
K
A
J
CLK
A
K
1
1
Input

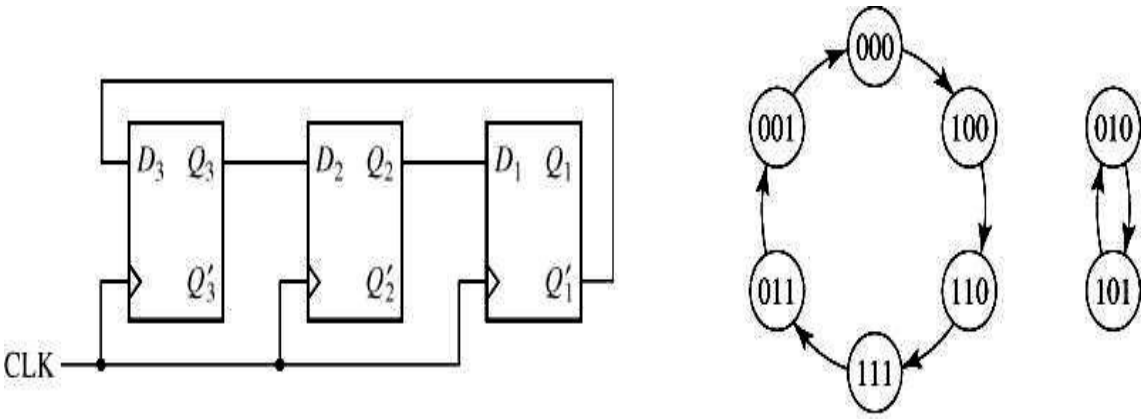
(a)


C
J
CLK
C
K
2
A
B
J
CLK
B
K
4
A
1
A
J
CLK
A
K
1
3
CLOCK
Up/Down

(a)

J0 = K0 = 1

Johnson Counter



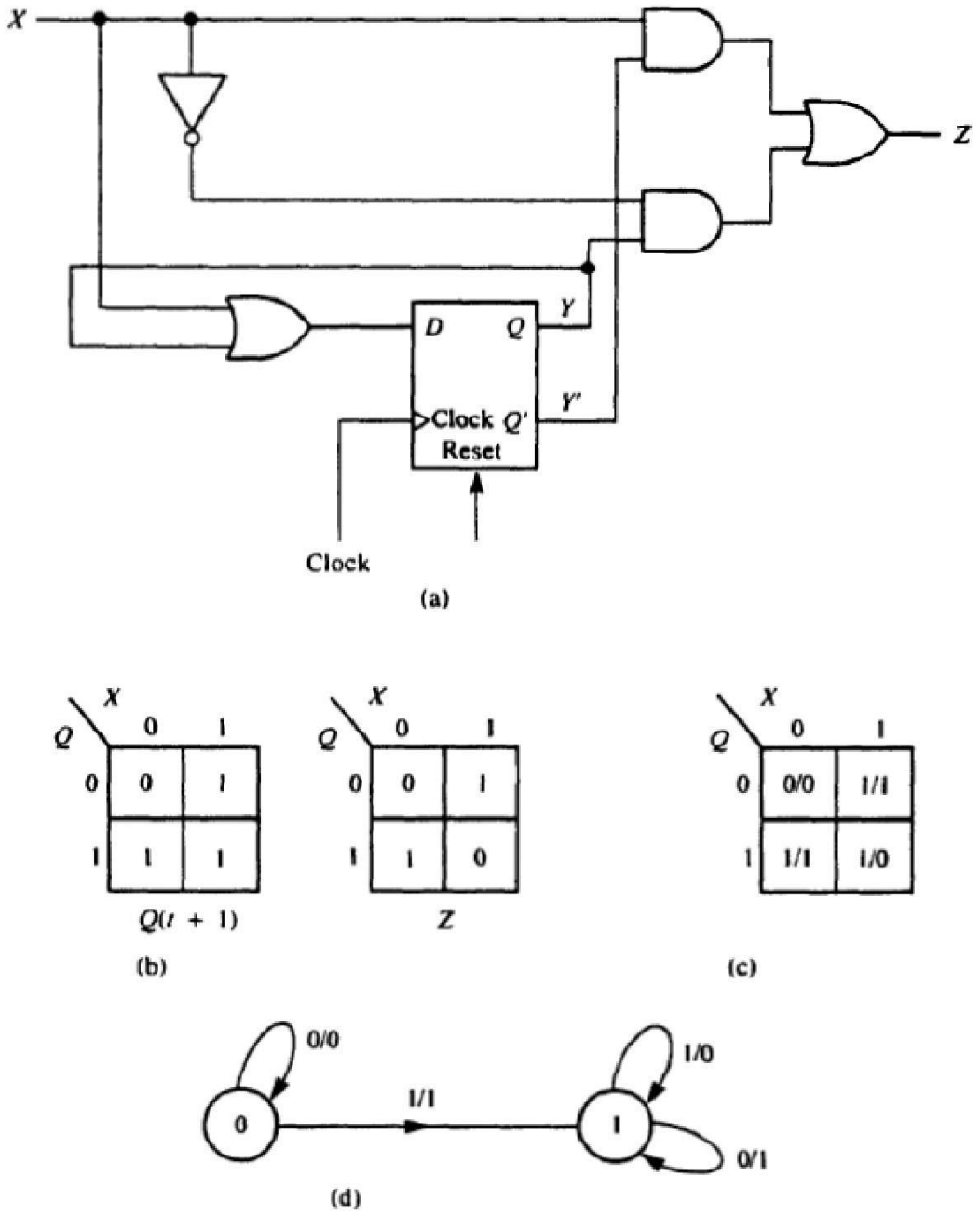
(a) Flip-flop connections (b) State graph

Analysis and Synthesis of Synchronous Sequential Circuit

The analysis of a synchronous sequential circuit is the process of determining the functional relation that exists between its outputs, its inputs, and its internal states. The contents of all the flip-flops in the circuit combined determine the internal state of the circuit. Thus, if the circuit contains n flip-flops, it can be in one of the 2^n states. Knowing the present state of the circuit and the input values at any time t, we should be able to derive its next state (i.e., the state at time $t + 1$) and the output produced by the circuit at t.

A sequential circuit can be described completely by a state table that is very similar to the ones shown for flip-flops

For a circuit with n flip-flops, there will be 2^n rows in the state table. If there are m inputs to the circuit, there will be 2^m columns in the state table. At the intersection of each row and column, the next-state and the output information are recorded. A *state diagram* is a graphical representation of the state table, in which each state is represented by a circle and the state transitions are represented by arrows between the circles. The input combination that brings about the transition and the corresponding output information are shown on the arrow. Analyzing a sequential circuit thus corresponds to generating the state table and the state diagram for the circuit. The state table or state diagram can be used to determine the output sequence generated by the circuit for a given input sequence if the *initial state* is known. It is important to note that for proper operation, a sequential circuit must be in its initial state before the inputs to it can be applied. Usually the power-up circuits are used to initialize the circuit to the appropriate state when the power is turned on.



Sequential circuit analysis. (a) Circuit; (b) next-state and output tables; (c) transition table; (d) state diagram; (e) timing diagram for level input; (f) timing diagram for synchronous pulse input.

Design of synchronous sequential circuit

The design of a sequential circuit is the process of deriving a logic diagram from the specification of the circuit's required behavior. The circuit's behavior is often expressed in words. The first step in the design is then to derive an exact specification of the required behavior in terms of either a state diagram or a state table. This is probably the most difficult step in the design, since no definite rules can be established to derive the state diagram or a state table. The designer's intuition and experience are the only guides. Once the description is converted into the state diagram or a state table, the remaining steps become mechanical. We will examine the classical design procedure through the examples in this section. It is not always necessary to follow this classical procedure, as some designs lend themselves to more direct and intuitive design methods. (The design of shift registers, described in Chapter 7, is one such example.) The classical design procedure consists of the following steps:

1. Deriving the state diagram (and state table) for the circuit from the problem statement.
2. Deriving the number of flip-flops (p) needed for the design from the number of states in the state diagram, by the formula

$$2^{p-1} < n \leq 2^p$$

   where n = number of states.
3. Deciding on the types of flip-flops to be used. (This often simply depends on the type of flip-flops available for the particular design.)
4. Assigning a unique p-bit pattern (state vector) to each state.
5. Deriving the state transition table and the output table.
6. Separating the state transition table into p tables, one for each flip-flop.
7. Deriving an input table for each flip-flop input using the excitation

State Table

The state table representation of a sequential circuit consists of three sections labelled *present state*, *next state* and *output*. The present state designates the state of flip-flops before the occurrence of a clock pulse. The next state shows the states of flip-flops after the clock pulse, and the output section lists the value of the output variables during the present state.

State Diagram

In addition to graphical symbols, tables or equations, flip-flops can also be represented graphically by a state diagram. In this diagram, a state is represented by a circle, and the transition between states is indicated by directed lines (or arcs) connecting the circles. An example of a state diagram is shown in Figure below.

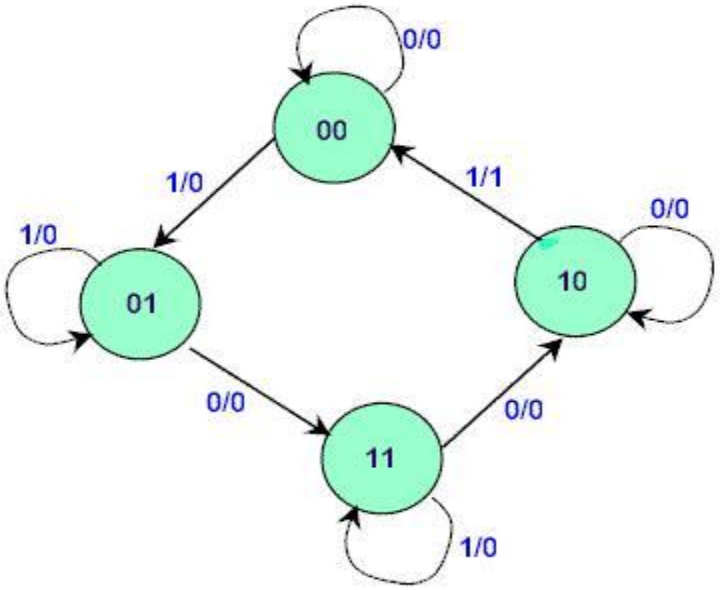


Figure . State Diagram

State Table

Present State	Next State		Output	
	X=0	X=1	X=0	X=1
00	11	01	0	0
01	11	00	0	0
10	10	11	0	1
11	10	10	0	1

State reduction

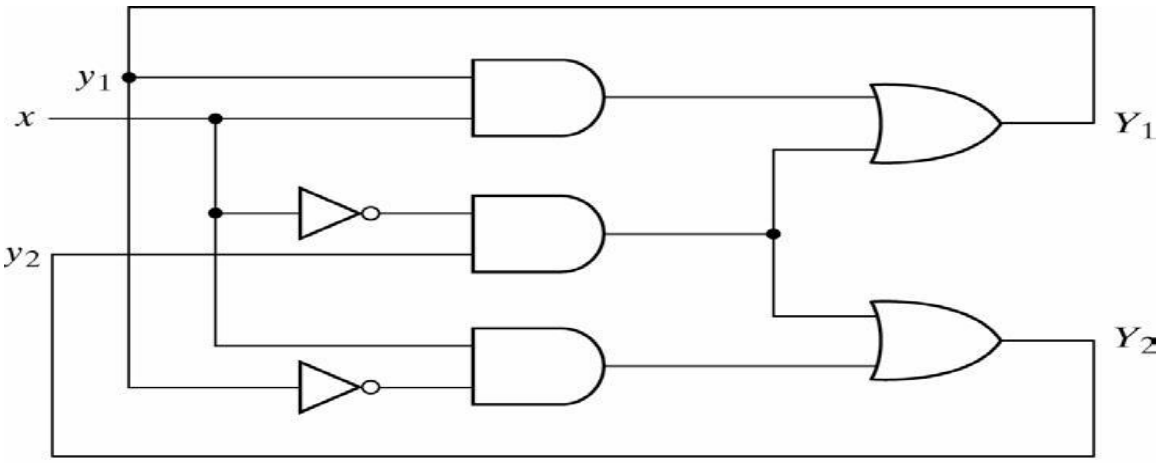
In the design of sequential circuits, we need to reduce the number of flip flops and the number of logic gates used in the combinational circuit part. Reduction of the number of flip-flops may result from the reduction of the number of states in the circuit. This is possible if we are interested in the input output relationship of the circuit and not in the outputs of the flip-flops.

State Assignment

State Assignment procedures are concerned with methods for assigning binary values to states in such a way as to reduce the cost of combinational circuit that drives the flipflop.

State	Assignment 1	Assignment 2
A	001	000
B	000	010
C	010	101
D	110	111

Transition Table



The analysis of the circuit starts by considering the excitation variables (*Y*1 and *Y*2) as outputs and the secondary variables (*y*1 and *y*2) as inputs.
The next step is to plot the *Y*1 and *Y*2 functions in a map:

$y_1 y_2$ \ x	0	1
00	0	0
01	1	0
11	1	1
10	0	1

(a) Map for $Y_1 = xy_1 + x'y_2$

$y_1 y_2$ \ x	0	1
00	0	1
01	1	1
11	1	0
10	0	0

(b) Map for $Y_2 = xy'_1 + x'y_2$

$y_1 y_2$ \ x	0	1
00	(00)	01
01	11	(01)
11	(11)	10
10	00	(10)

Combining the binary values in corresponding squares the following *transition table* is obtained:

Primitive flow table: one that has only one stable state in each row. To obtain the circuit described by a flow table, it is necessary to assign to each state a distinct binary value, which converts the flow table into a transition table.

	x = 0	x = 1
a	(a)	b
b	c	(b)
c	(c)	d
d	a	(d)

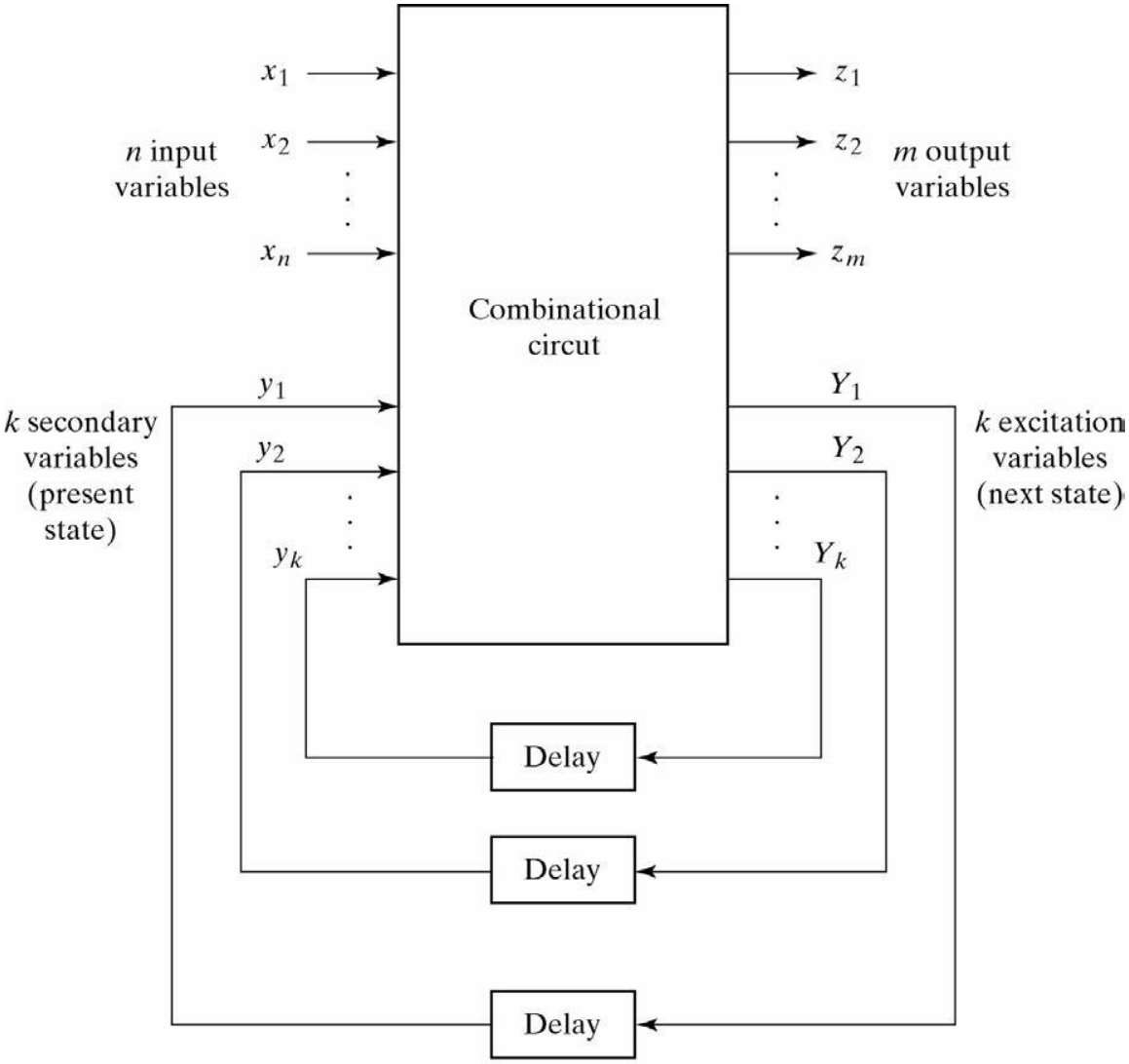
(a) Four states with one input

	x_1x_2 = 00	01	11	10
a	(a), 0	(a), 0	(a), 0	b , 0
b	a , 0	a , 0	(b), 1	(b), 0

(b) Two states with two inputs and one output

Synchronous sequential logic

Nearly all sequential logic today is 'clocked' or 'synchronous' logic: there is a 'clock' signal, and all internal memory (the 'internal state') changes only on a clock edge. The basic storage element in sequential logic is the flip-flop.



The main advantage of synchronous logic is its simplicity. Every operation in the circuit must be completed inside a fixed interval of time between two clock pulses, called a 'clock cycle'. As long as this condition is met (ignoring certain other details), the circuit is guaranteed to be reliable. Synchronous logic also has two main disadvantages, as follows.

1. The clock signal must be distributed to every flip-flop in the circuit. As the clock is usually a high-frequency signal, this distribution consumes a relatively large amount of

power and dissipates much heat. Even the flip-flops that are doing nothing consume a small amount of power, thereby generating waste heat in the chip.

2. The maximum possible clock rate is determined by the slowest logic path in the circuit, otherwise known as the critical path. This means that every logical calculation, from the simplest to the most complex, must complete in one clock cycle. One way around this limitation is to split complex operations into several simple operations, a technique known as 'pipelining'. This technique is prominent within microprocessor design, and helps to improve the performance of modern processors.

In digital circuit theory, **sequential logic** is a type of logic circuit whose output depends not only on the present input but also on the history of the input. This is in contrast to *combinational logic*, whose output is a function of, and only of, the present input. In other words, sequential logic has *storage* (*memory*) while combinational logic does not.

Sequential logic is therefore used to construct some types of computer memory, other types of delay and storage elements, and finite state machines. Most practical computer circuits are a mixture of combinational and sequential logic.

There are two types of finite state machine that can be built from sequential logic circuits:

- Moore machine: the output depends only on the internal state. (Since the internal state only changes on a clock edge, the output only changes on a clock edge too).
- Mealy machine: the output depends not only on the internal state, but also on the inputs.

Depending on regulations of functioning, digital circuits are divided into synchronous and asynchronous. In accordance with this, behavior of devices obeys synchronous or asynchronous logic.

UNIT-IV ASYNCHRONOUS SEQUENTIAL CIRCUITS

Asynchronous sequential logic

Asynchronous sequential logic expresses memorizing effect by fixing moments of time, when digital device changes its state. These moments are represented not in explicit form, but taking into account principle "before/after" in temporal relations of logical values. For asynchronous logic it is sufficient to determine a sequence of switchings irrespective of any connections of the corresponding moments with real or virtual time.Theoretical apparatus of sequential logic consists of mathematical instruments of sequention and venjunction as well as of logic-algebraic equations on their basis.An asynchronous sequential circuit is a sequential circuit whose behavior depends only on the order in which its input signals change and can be affected at any instant of time.Memory (delay) elements are either latches (unclocked) or time-delay elements (instead of clocked FFs as in a synchronous sequential circuit).

+ An asynchronous sequential circuit quite often resembles a combinational circuit with feedback.

+ Faster and often cheaper than synchronous ones, but more difficult to design, verify, or test (due to possible timing problems involved in the feedback path).

Analysis Procedure

The analysis consists of obtaining a table or a diagram that describes the sequence

of internal states and outputs as a function of changes in the input variables.

1. Determine all feedback loops.

2. Designate each feedback-loop output with Yi and its corresponding input with yi for i = 1; : : : ; k, where k is the number of feedback loops.

3. Derive the boolean functions for all Y "s.

4. Plot the transition table from the equations.

Design Procedure

1. Obtain a primitive flow table.

2. Reduce the flow table.

3. Assign binary state variables to obtain the transition table.

4. Assign output values to the dashes to obtain the output maps.

5. Simplify the excitation and output functions.

6. Draw the logic diagram.

Example problem

Design a gated latch circuit with two inputs, *G* (gate) and *D* (data), and one output *Q*. The gated latch is a memory element that accepts the value of *D* when *G* = 1 and retains this value after *G* goes to 0. Once *G* = 0, a change in *D* does not change the value of the output *Q*.

Solution

State table

State	Inputs		Output
	D	G	Q
a	0	1	0
b	1	1	1
c	0	0	0
d	1	0	0
e	1	0	1
f	0	0	1

Primitive Flow table

	DG = 00	DG = 01	DG = 11	DG = 10
a	c , –	(a), 0	b , –	– , –
b	– , –	a , –	(b), 1	e , –
c	(c), 0	a , –	– , –	d , –
d	c , –	– , –	b , –	(d), 0
e	f , –	– , –	b , –	(e), 1
f	(f), 1	a , –	– , –	e , –

Informal Merging

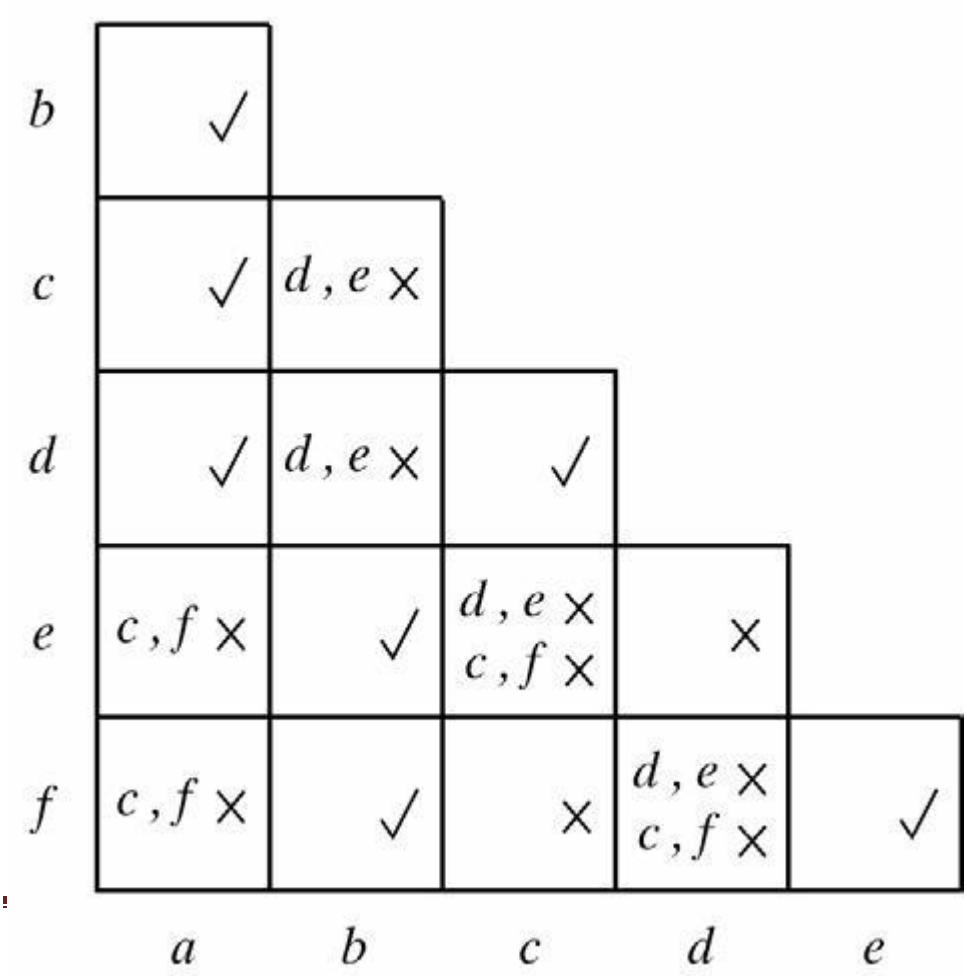
	DG 00	01	11	10
a, c, d	(*c*), 0	(*a*), 0	*b* , –	(*d*), 0
b, e, f	(*f*), 1	*a* , –	(*b*), 1	(*e*), 1

	DG 00	01	11	10
a	(*a*), 0	(*a*), 0	*b* , –	(*a*), 0
b	(*b*), 1	*a* , –	(*b*), 1	(*b*), 1

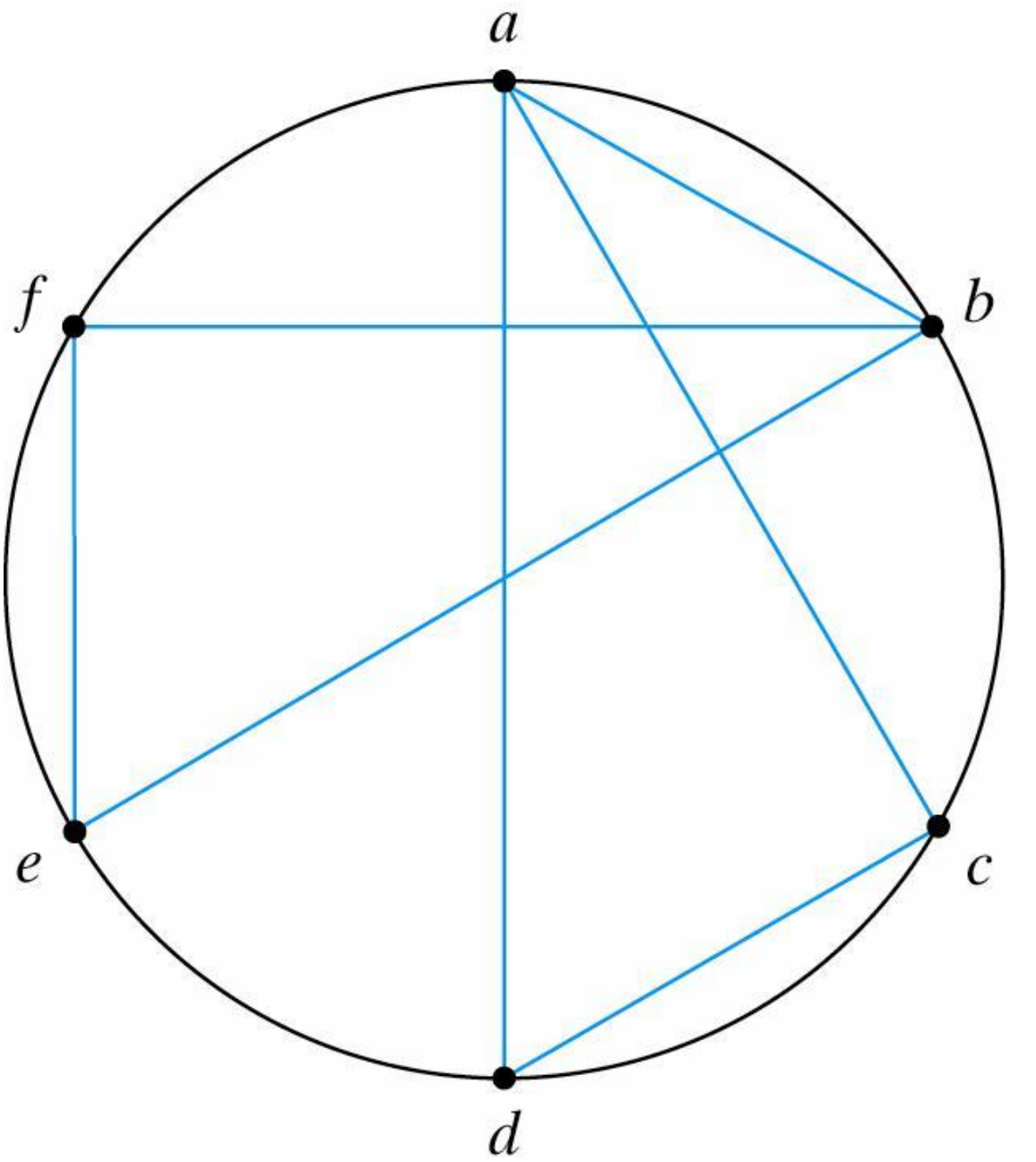
(b) Reduced table (two alternatives)

Formal Merging

Compatible Pairs



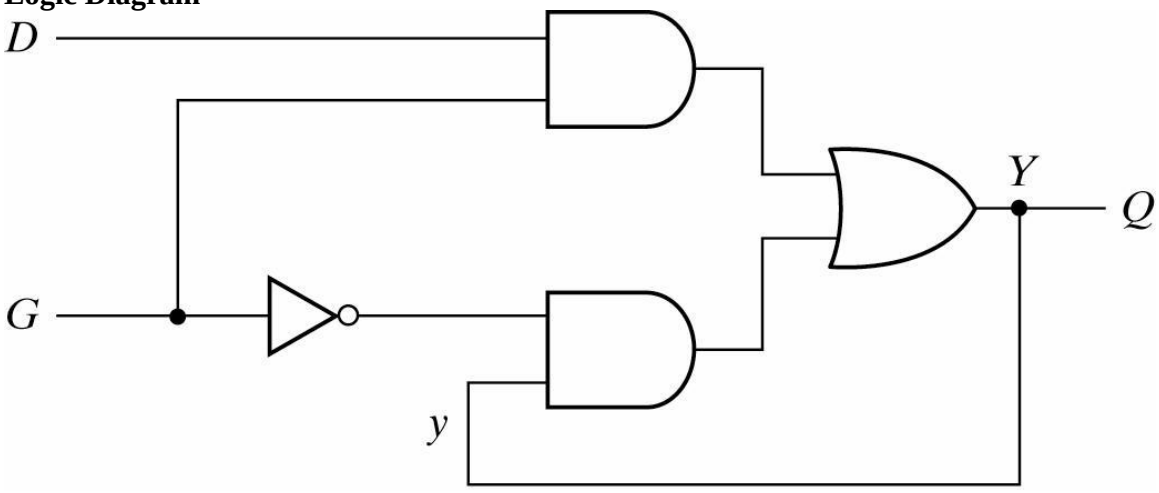
Maximal Compatibles



Reduced Table

	DG = 00	DG = 01	DG = 11	DG = 10
a	a, 0	a, 0	b , –	a, 0
b	b, 1	a , –	b, 1	b, 1

Logic Diagram



The same problem using SR Latch

Lists the required inputs *S* and *R* for each of the possible transitions from the secondary variable *y* to the excitation variable *Y*.

y	*Y*	*S*	*R*
0	0	0	*X*
0	1	1	0
1	0	0	1
1	1	*X*	1

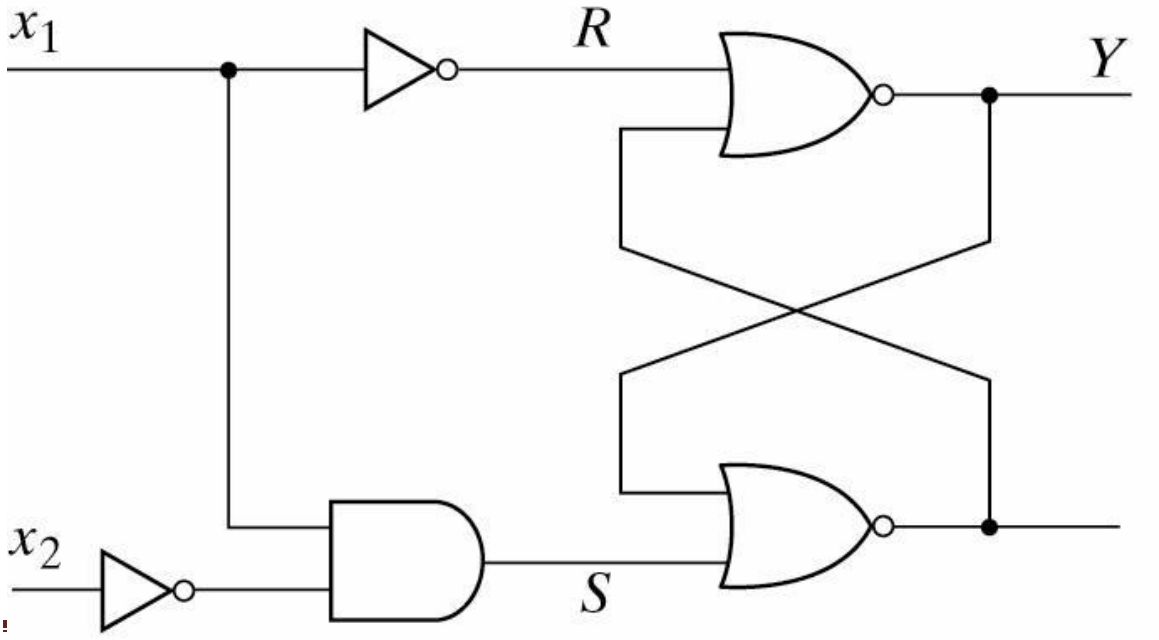
y \ x_1x_2	00	01	11	10
0	0	0	0	1
1	0	0	*X*	*X*

(c) Map for $S = x_1x'_2$

y \ x_1x_2	00	01	11	10
0	*X*	*X*	*X*	0
1	1	1	0	0

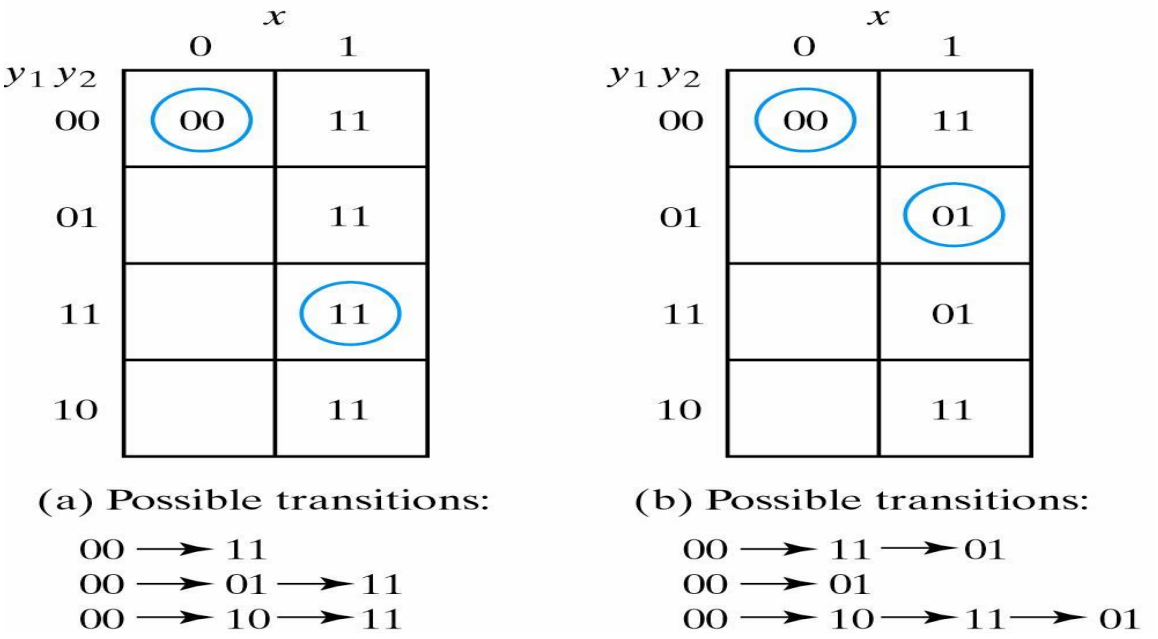
(d) Map for $R = x'_1$

SR Latch Logic Diagram

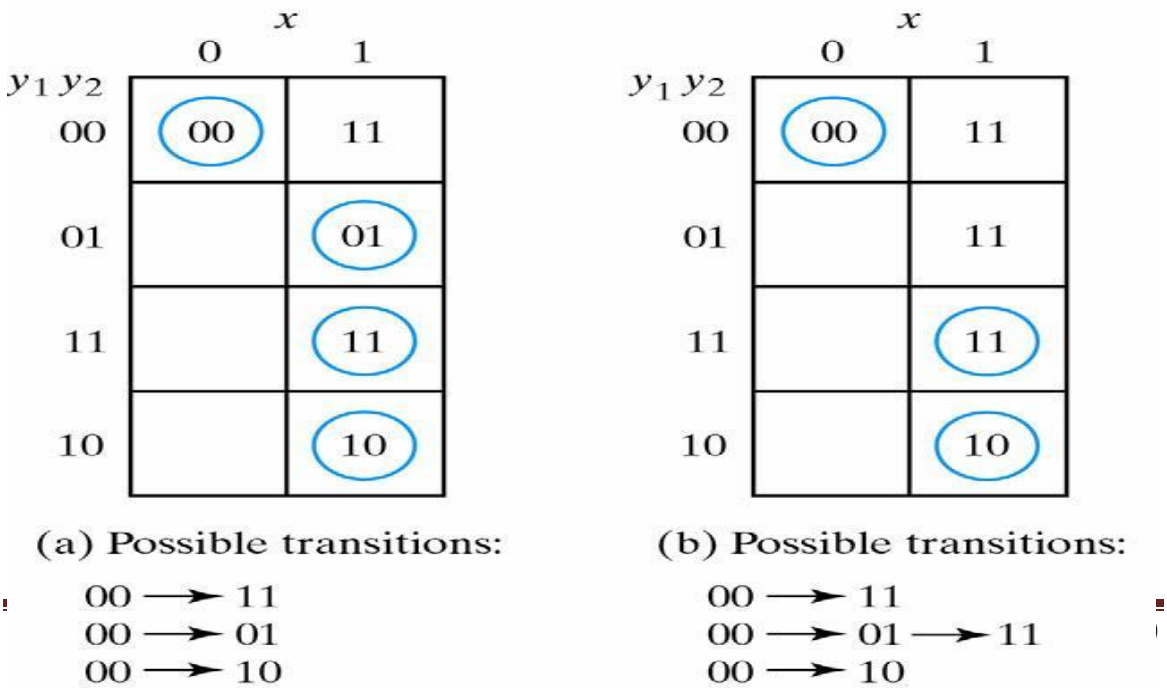


Races

A *race* condition exists in an asynchronous circuit when two or more binary state variables change value in response to a change in an input variable. When unequal delays are encountered, a race condition may cause the state variable to change in an unpredictable manner. If the final stable state that the circuit reaches does not depend on the order in which the state variables change, the race is called a *noncritical race*. Examples of noncritical races are illustrated in the transition tables below:



The transition tables below illustrate critical races:



Hazard

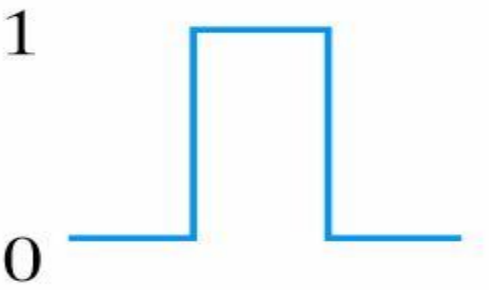
In digital logic, a **hazard** in a system is an undesirable effect caused by either a deficiency in the system or external influences. Logic hazards are manifestations of a problem in which changes in the input variables do not change the output correctly due to some form of delay caused by logic elements (NOT, AND, OR gates, etc.) This results in the logic not performing its function properly. The three different most common kinds of hazards are usually referred to as **static**, **dynamic** and **function hazards**.

Hazards are a temporary problem, as the logic circuit will eventually settle to the desired function. However, despite the logic arriving at the correct output, it is imperative that hazards be eliminated as they can have an effect on other connected systems.

A **static hazard** is the situation where, when one input variable changes, the output changes momentarily before stabilizing to the correct value. There are two types of static hazards:

- Static-1 Hazard: the output is currently 1 and after the inputs change, the output momentarily changes to 0 before settling on 1

- Static-0 Hazard: the output is currently 0 and after the inputs change, the output momentarily changes to 1 before settling on 0

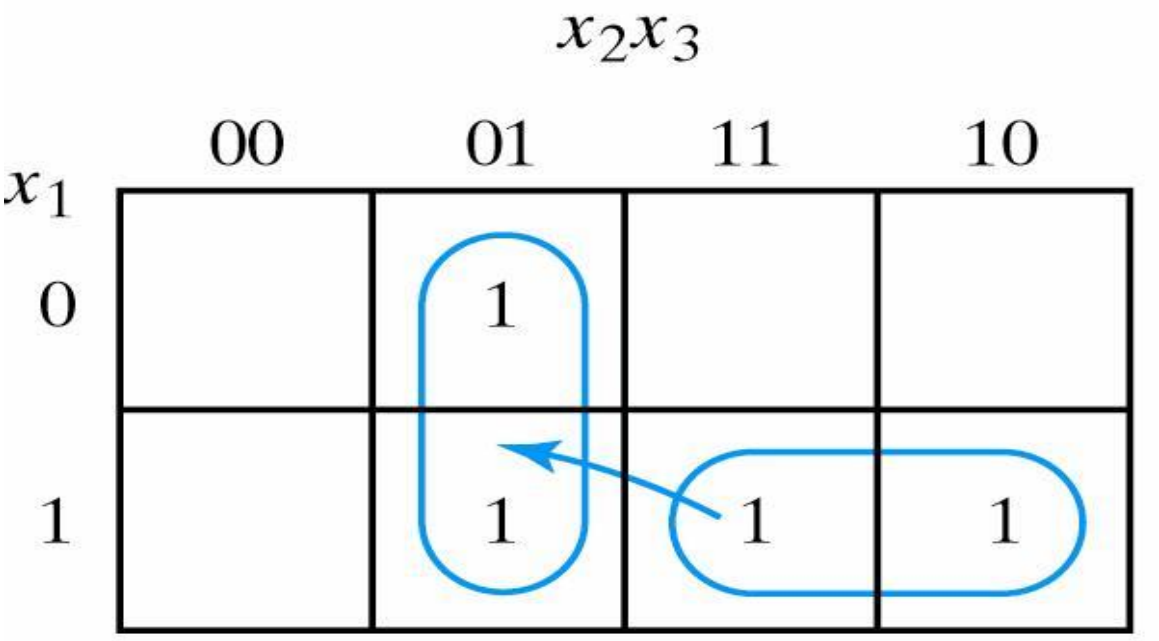


A **dynamic hazard** is the possibility of an output changing more than once as a result of a single input change. Dynamic hazards often occur in larger logic circuits where there are different routes to the output (from the input). If each route has a different delay, then it quickly becomes clear that there is the potential for changing output values that differ from

the required / expected output. e.g. A logic circuit is meant to change output state

from **1** to **0**, but instead changes from **1** to **0** then **1** and finally rests at the correct value **0**. This is a dynamic hazard.

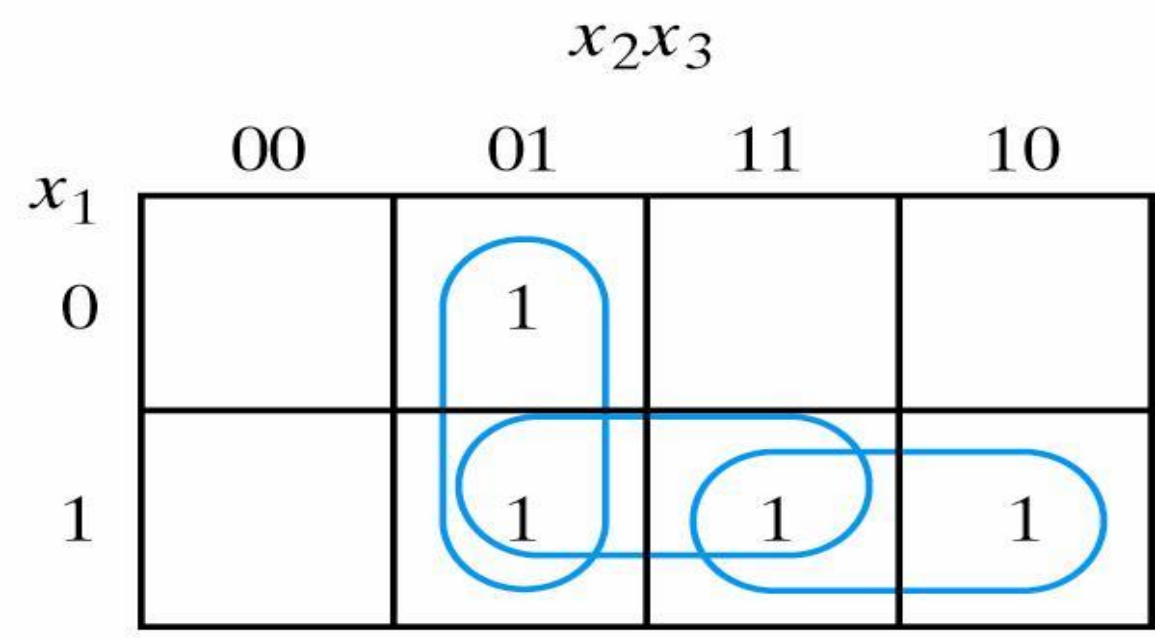
Example of Hazard free circuit is,



Normal K"Map answer is

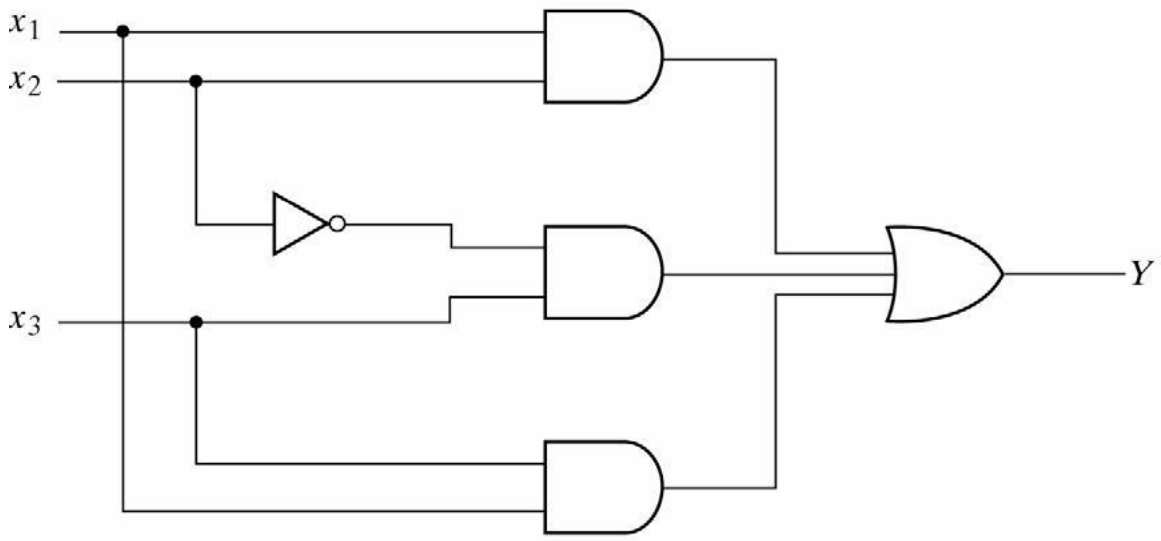
$$Y = x1x2 + x2'\,x3$$

Hazard free map is



$$Y = x1x2 + x2'\,x3 + x1x3$$

Hazard Free Logic diagram is



UNIT-V MEMORY DEVICES

Memory is a storage A memory for which bits can both be easily stored or retrieved (\written to" or \read from"). Here is a rundown on some terms:

RAM. In general, refers to random access memory. All of the devices we are considering to be \memories" (RAM, ROM, etc.) are random access. The term RAM has also come to mean memory which can be both easily written to and read from. There are two main technologies used for RAM:

1.) Static RAM. These essentially are arrays of flip-flops. They can be fabricated in ICs as large arrays of tint flip-flops.) \SRAM" is intrisically somewhat faster than dynamic RAM.

2.) Dynamic RAM. Uses capacitor arrays. Charge put on a capacitor will produce a HIGH bit if its voltage $V = Q=C$ exceeds the threshold for the logic standard in use. Since the charge will \leak" o through the resistance of the connections in times of order 1 msec, the stored information must be continuously refreshed (hence the term \dynamic"). Dynamic RAM can be fabricated with more bits per unit area in an IC than static RAM. Hence, it is usually the technology of choice for most large-scale IC memories.

ROM. *Read-only memory*. Information cannot be easily stored. The idea is that bits are initially de ned and are never changed thereafter. As an example, it is generally prudent for the instructions used to initialize a computer upon initial power-up to be stored in ROM. The following terms refer to versions of ROM for which the stored bits *can* be over-written, but not easily.

PROM. *Programmable* ROM. Bits can be set on a programming bench by burning \fusible links," or equivalent. This technology is also used for programmable array logic (PALs), which we will briefly discuss in class.

EPROM. ROM which can be erased using ultraviolet light.

EEPROM. ROM which can be erased electronically.

A few other points of terminology:

As you know, a *bit* is a binary digit. It represents the smallest element of information.

A *byte* is 8 bits.

A \K" of memory is $2^{10} = 1024$ bits (sometimes written KB). And a megabit (MB) is $1K$ $1K$ bits.

RAM is organized into many data \words" of some prescribed length. For example, a RAM which has 8*K* = 8192 memory locations, with each location storing a data word of \width" 16 bits, would be referred to as a RAM of size 8*K* 16. The total storage capacity of this memory would therefore be 128KB, or simply a \128K" memory. (With modern *very large scale integration* (VLSI) technology, a typical RAM IC might be 16 MB. Besides the memory \size," the other important specification for memory is the *access time*. This is the time delay between when a valid request for stored data is sent to a memory and when the corresponding bit of data appears at the output. A typical access time, depending upon the technology of the memory, might be 10 ns.

7.2 Memory Configuration

As stated above, the term \memory" refers to a particular way of organizing information | by random access | which is distinct from the less specific term \data storage." Figure 36 shows how an 8-bit RAM (8 1) is organized. (This is a very small memory, but illustrates the concepts.) Our RAM consists of three main components: an 8-bit multiplexer, an 8-bit demultiplexer, and 8 bits of storage. The storage shown consists of edge-triggered D-type flip-flops. Hence, this is evidently a \static RAM." (There is no fundamental reason for using edge-triggered flip-flops. They could just as easily be level-triggered, like the simple \clocked" S-R flip-flop of Fig. 14.)

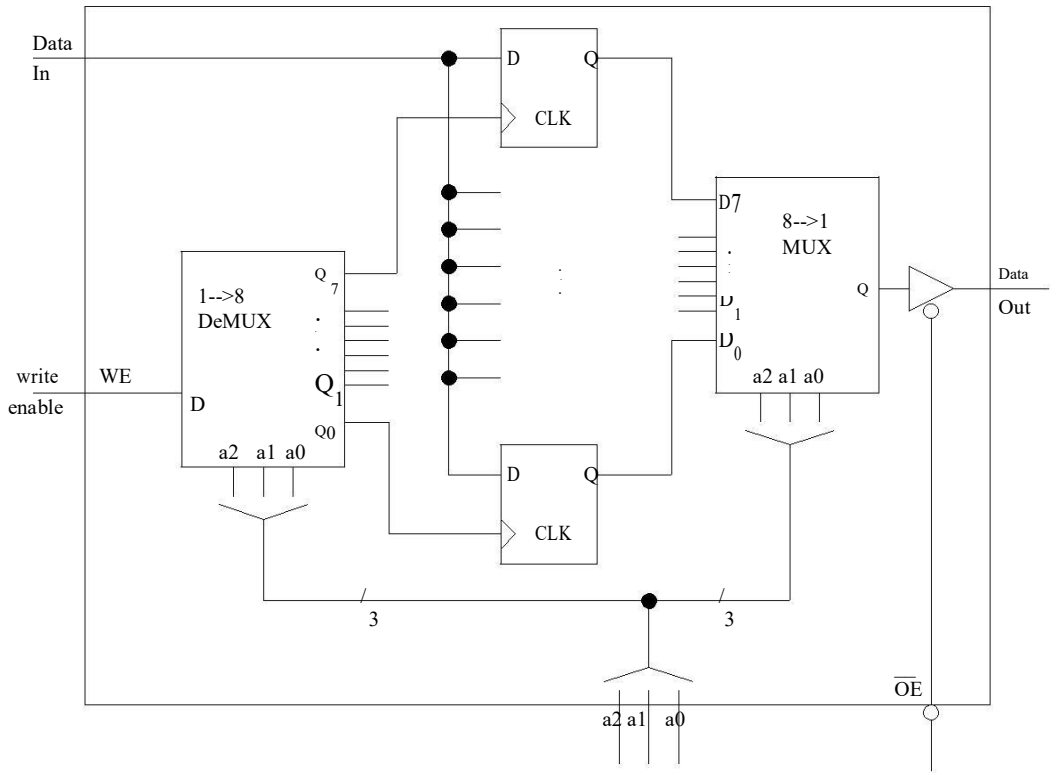


Figure 36: An 8 1 bit RAM.

Our example RAM has 6 external connections which are inputs (data in, write enable (WE), 3-state enable (OE), and 3 address bits ($A = a_2a_1a_0$), and has one output connection (data out), giving 7 external connections total, plus 2 for power/ground. To write information to the RAM, one would supply a valid address, for example A = 101. The data bit to be written to location 101 is to appear at the data input as either a logic HIGH or LOW signal. And to enable the writing into this bit, the WE signal must be asserted. This then appears at the Q_5 output of the demultiplexer, and is passed on to the appropriate flip-flop, which stores the input data bit and passes it on to the Q_5 multiplexer input.

To read data from our RAM, one asserts an address, so that the selected bit is sent to the MUX output and then the 3-state bu er. The purpose of the 3-state bu er is to ensure that no digital outputs are directly connected together, for example if our RAM output were connected to a data \bus," which in turn was connected to several other devices. Recall that the 3-state devices have outputs which are e ectively disconnected if there is no enable signal. So if the output data connection of our RAM is connected to a data bus, then the OE signal must be coordinated with any other outputs also connected to the data bus. When it is OK to read data from the RAM (all other output devices are disconnected from the bus), the OE signal is asserted and the MUX output will appear at the RAM output.

One could of course also store the 8 bits of data directly to an 8-bit data register, rather than using the RAM con guration outlined above. In this case, the number of external connections is 17 (8 data in, 8 data out, and 1 clock), compared with the 7 of our RAM. For a more realistic case where the number of bits of memory n is much larger than our example, we generalize the above to arrive at $4 + \log_2(n)$ external connections for the RAM, compared with $1 + 2n$ for the standalone register. Obviously for large n, the register is impractical, whereas the RAM remains reasonable. Actually, it is even somewhat better than this for the RAM case, since the number of external connections does not grow with the width of the stored data words. Hence, a RAM of size $1K$ 16 = 16 KB requires only 14 connections. This is to be compared with 32,001 connections for the register. Note that the RAM can only supply one bit at a time to the output. This may seem like a handicap, but is actually well matched to standard microprocessors.

7.3 A State Machine with Memory

For reference, our usual state machine con guration is shown again in Fig. 37. Now we consider the use of a memory with a state machine, as depicted in Fig. 38. A random access memory is used in place of the usual combinational logic. (A ROM has been speci ed, to emphasize that we are not changing the memory | once it is de ned initially, it is only read from. The memory is used to conveniently encode the connection between present and next states.

To start with, let's assume a state machine with no external inputs or outputs. Then the state machine's present state (PS) becomes an *address* which is input to the ROM. The *data word* stored in the ROM at that address then corresponds to the next state (NS). This correspondence had been initially programmed into the ROM, just as the speci c combina-tional logic in our old state machine had to be pre-determined. So if the PS as de ned by the Q bits at the data register are, for example, 1001, then the ROM data word at address 1001 will be the NS which is then passed back to the register. When there are also external inputs, as there will be for most anything of interest, these are combined with the PS bits to form a longer address for the ROM. Similarly, any external outputs are combined with the NS bits in the data word.

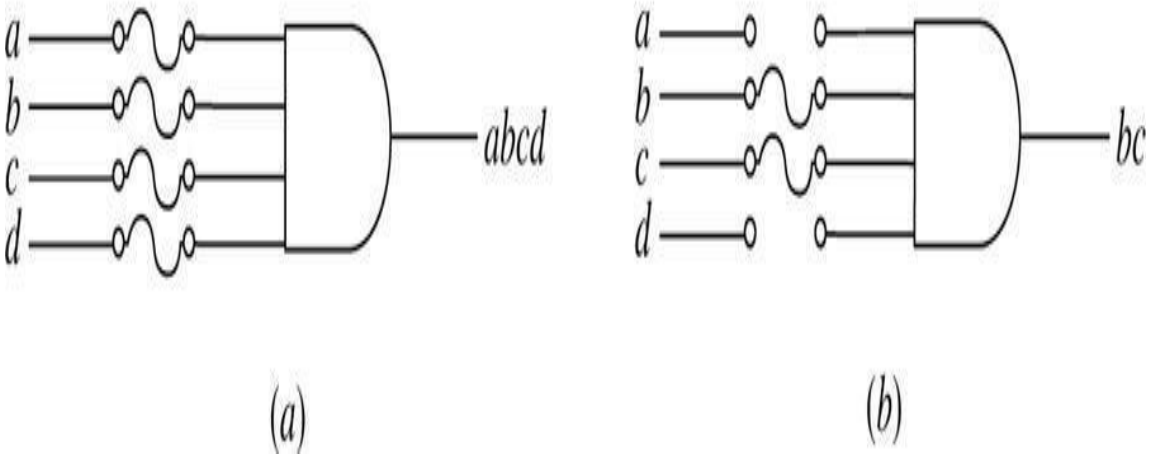
Programmable Logic Devices (PLD's)

- ***Programmable devices have their functionality programmed before they are first used.***
- ***Range in complexity from 100's to 10,000's of logic gates.***

- ***Types of Programmable Logic Devices***
- PLDs (Programmable Logic Devices)
  - ROM (Read-Only Memory)
  - PLA (Programmable Logic Array)
  - PAL (Programmable Array Logic)

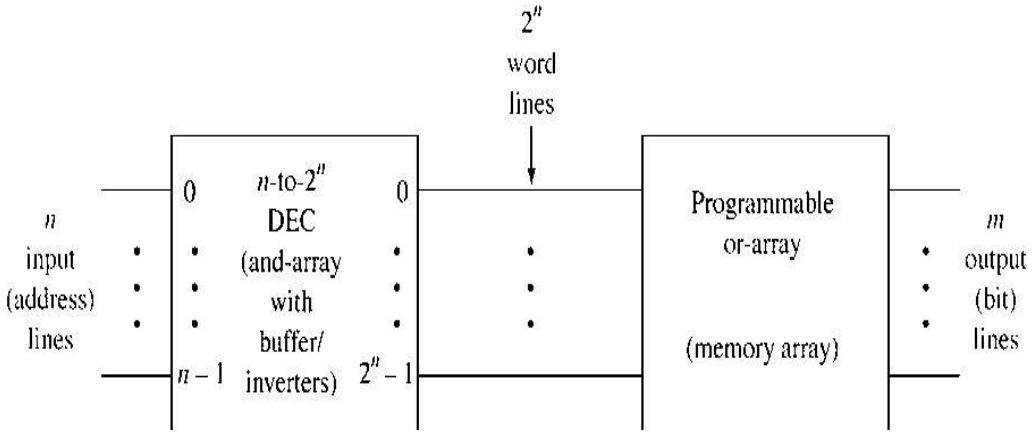
Device	AND-array	OR-array
PROM	Fixed	Programmable
PLA	Programmable	Programmable
PAL	Programmable	Fixed
GAL	Programmable	Fixed

Programming by blowing fuses.

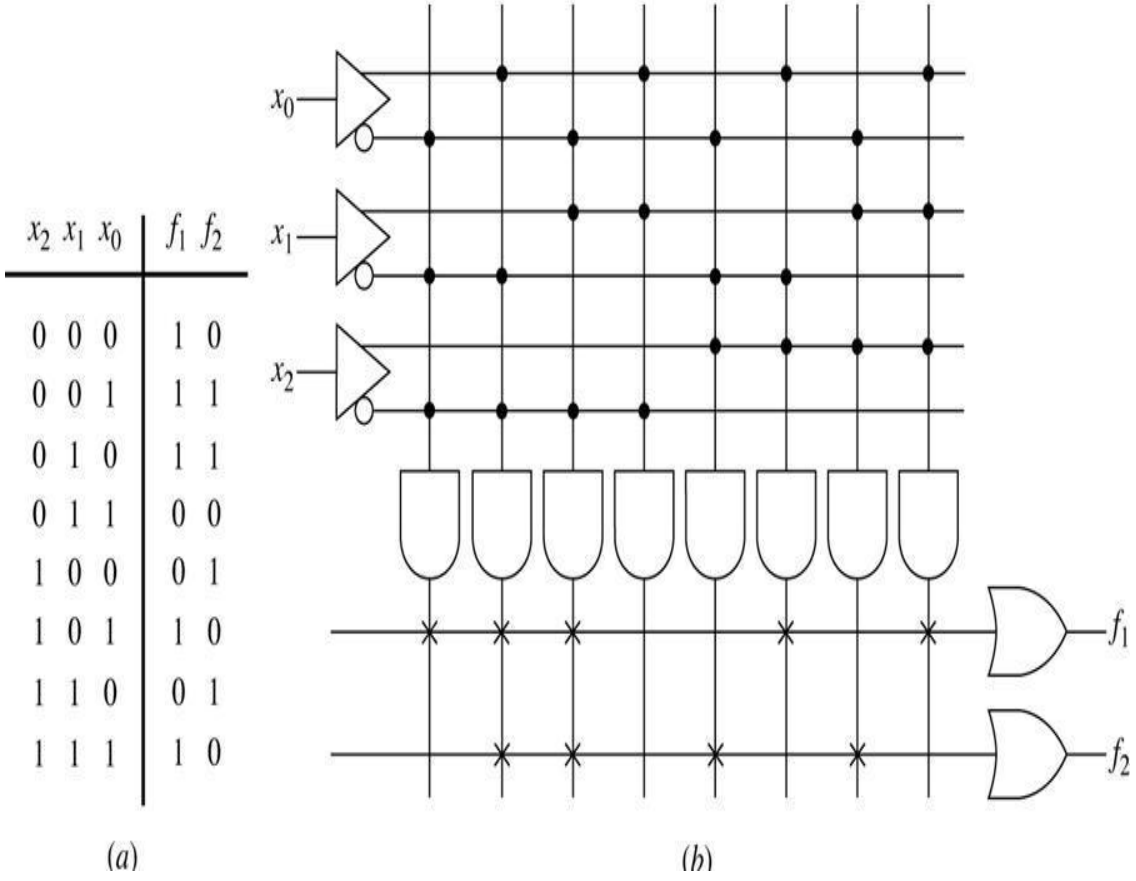


(*a*) Before programming. **(*b*) After Programming.**

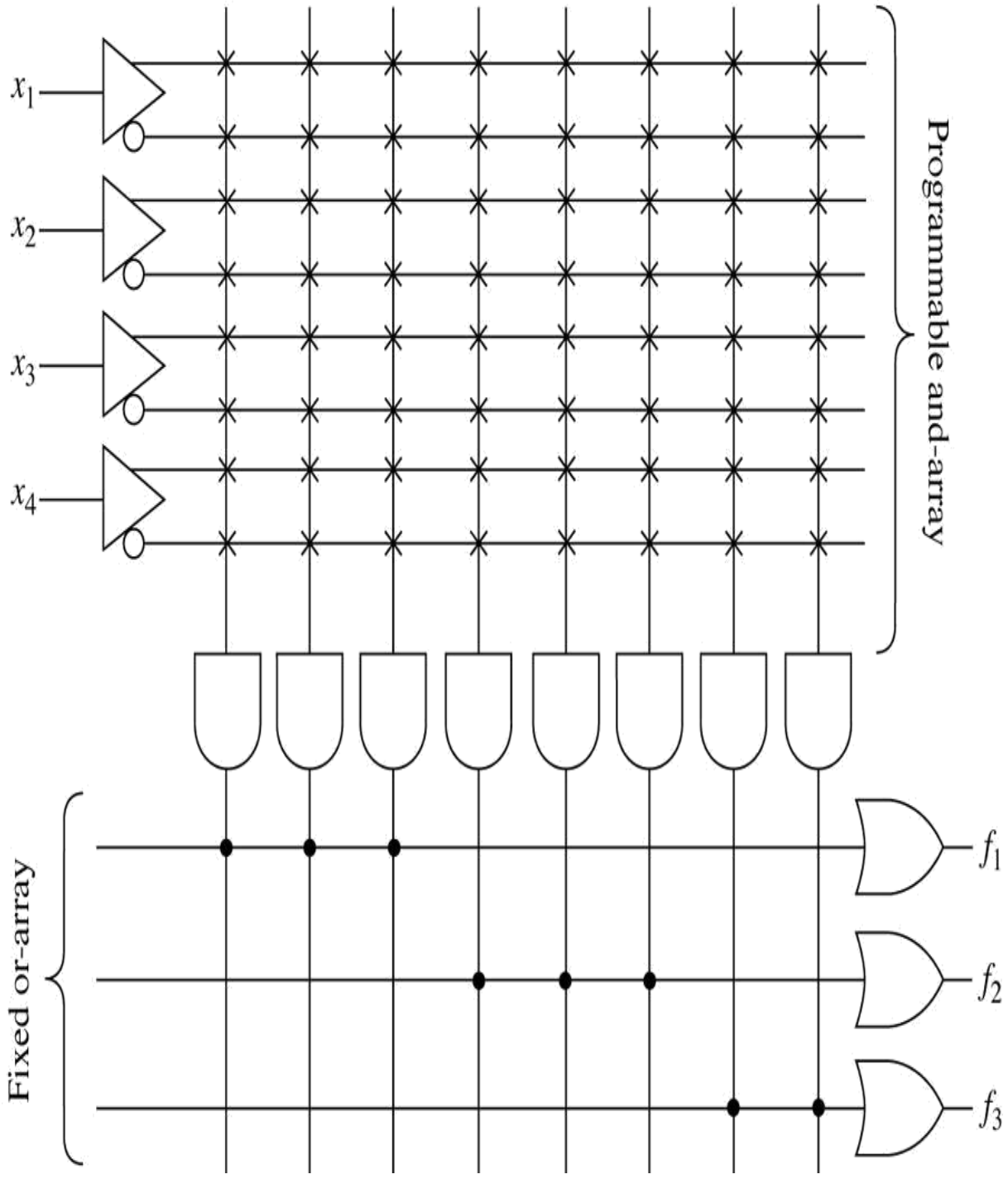
PROM Notation



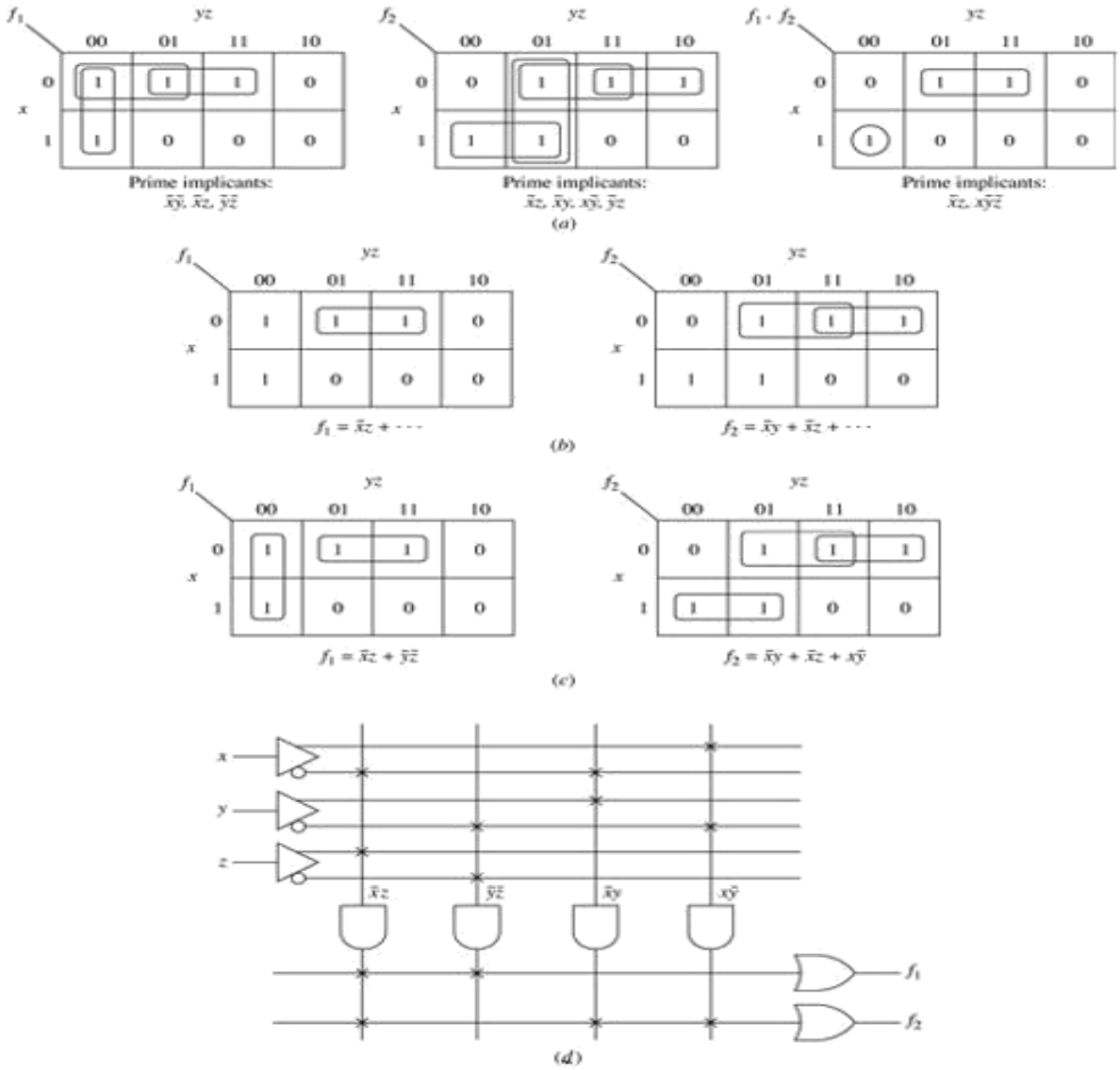
Using a PROM for logic design



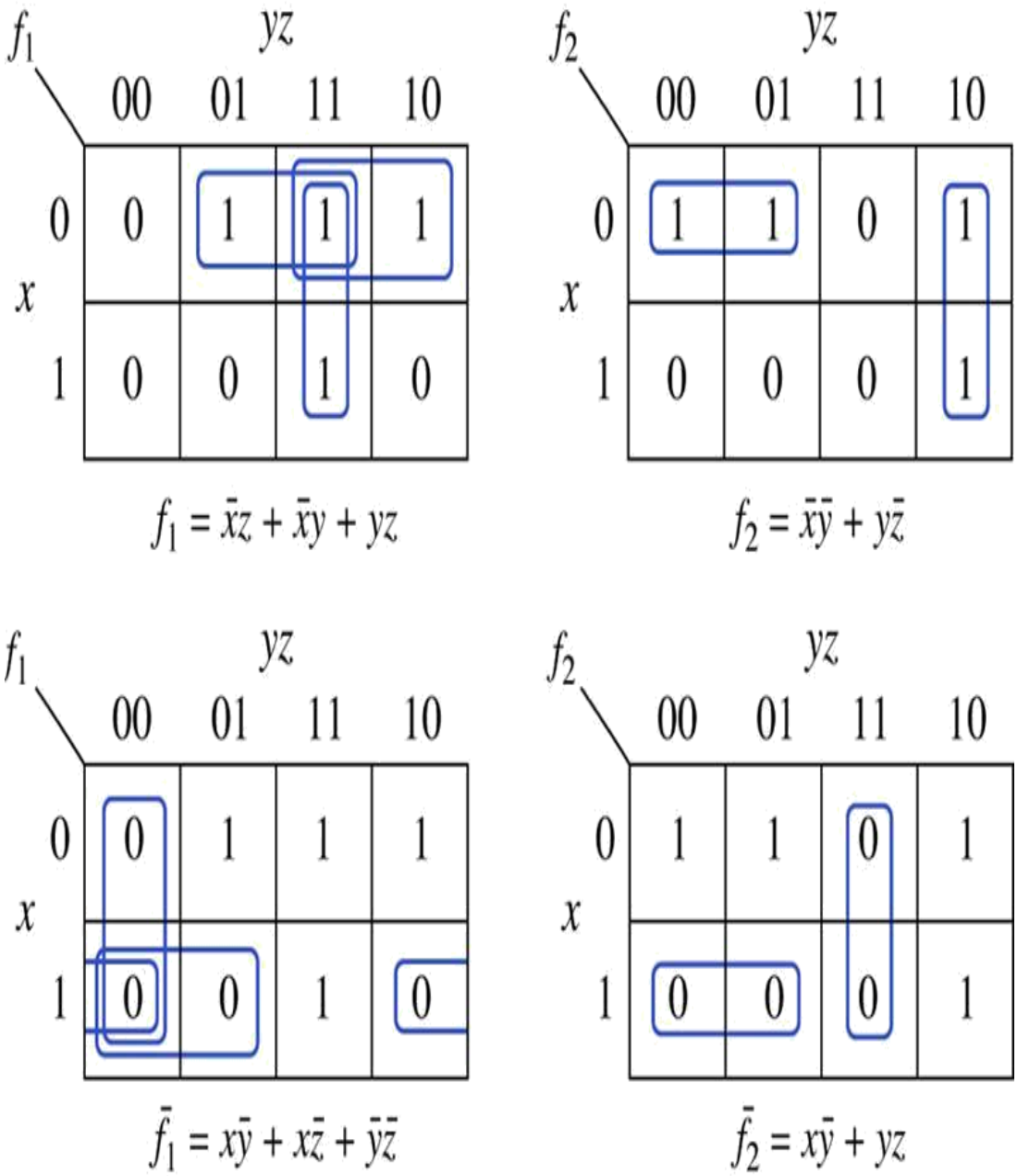
A simple four-input, three-output PAL device.



Example of combinational logic design using a PLA.

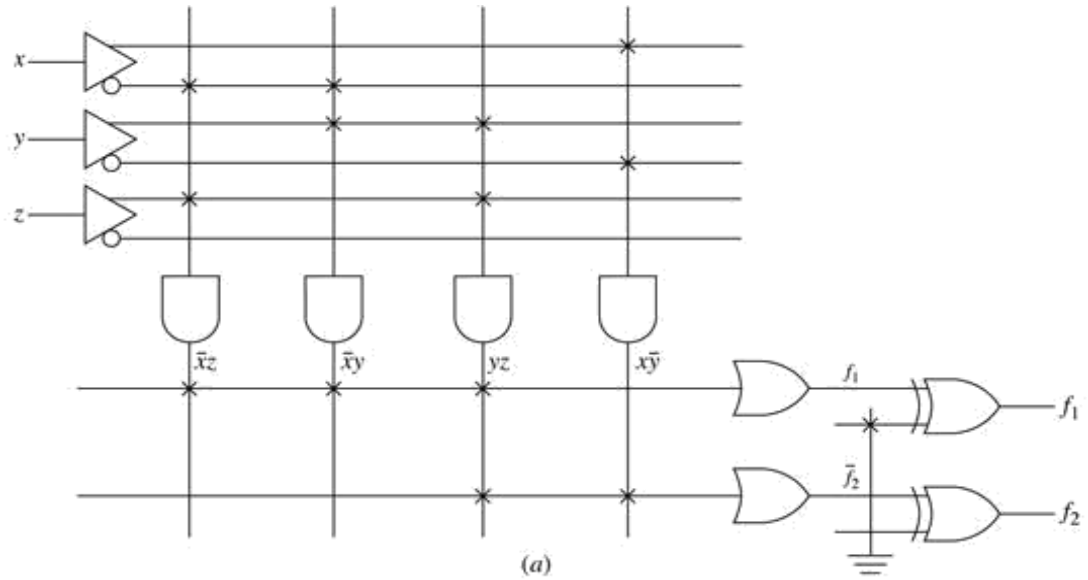


Karnaugh maps for the functions *f*1(*x*,*y*,*z*) = □*m*(1,2,3,7) and *f*2(*x*,*y*,*z*) = □*m*(0,1,2,6)

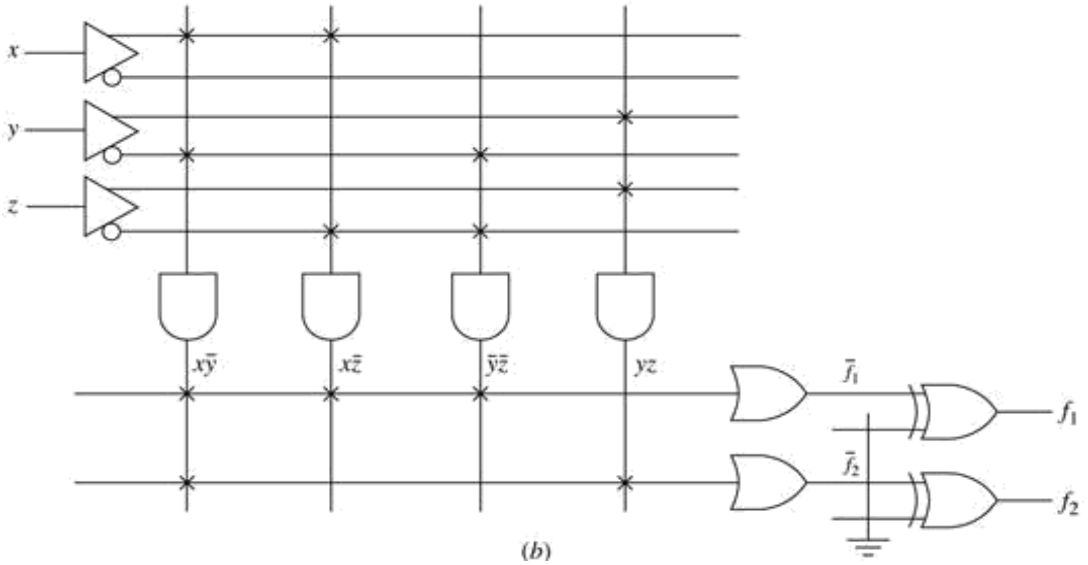


Two realizations of $f1(x,y,z) = \square m(1,2,3,7)$ and $f2(x,y,z) = \square m(0,1,2,6)$.

(*a*) Realization based on *f*1 and 2 (*b*) Realization based on 1 and 2


x
y
z
$\bar{x}z$
$\bar{x}y$
yz
$x\bar{y}$
f_1
$\bar{f}_2$
f_1
f_2

(*a*)


x
y
z
$x\bar{y}$
$x\bar{z}$
$\bar{y}\bar{z}$
yz
$\bar{f}_1$
$\bar{f}_2$
f_1
f_2

(*b*)

2 Marks Questions and Answers

1. Define the term digital.
   The term digital refers to any process that is accomplished using discrete units

2. What is meant by bit?
   A binary digit is called bit

3. What is the best example of digital system?
   Digital computer is the best example of a digital system.

4. Define byte?
   A group of 8 bits.

5. List the number systems?
   v) Decimal Number system
   vi) Binary Number system
   vii) Octal Number system
   viii) Hexadecimal Number system

6. State the sequence of operator precedence in Boolean expression?
   i) Parenthesis
   ii) AND
   iii) OR

7. What is the abbreviation of ASCII and EBCDIC code?
   ASCII-American Standard Code for Information Interchange.
   EBCDIC-Extended Binary Coded Decimal Information Code.

8. What are the universal gates? NAND and NOR

9. What are the different types of number complements?
   i) r" s Complement
   ii) (r-1)" s Complement.

10. Why complementing a number representation is needed?
    Complementing a number becomes as in digital computer for simplifying the subtraction operation and for logical manipulation complements are used.

11. How to represent a positive and negative sign in computers? Positive (+) sign by 0

Negative (-) sign by 1.

12. What is meant by Map method?

The map method provides a simple straightforward procedure for minimizing Boolean function.

13. What is meant by two variable map?

Two variable map have four minterms for two variables, hence the map consists of four squares, one for each minterm

14. What is meant by three variable map?

Three variable map have 8 minterms for three variables, hence the map consists of 8 squares, one for each minterm

15. Which gate is equal to AND-inverter Gate?

NAND gate.

16. Which gate is equal to OR-inverter Gate?

NOR gate.

17. Bubbled OR gate is equal to----------

NAND gate

18. Bubbled AND gate is equal to----------

NOR gate

19. What is the use of Don"t care conditions?

Any digital circuit using this code operates under the assumption that these unused combinations will never occur as long as the system

.

20. Express the function f(x, y, z)=1 in the sum of minterms and a product of maxterms? Minterms=□(0,1,2,3,4,5,6,7)

Maxterms=Nomaxterms.

21. What is the algebraic function of Exclusive-OR gate and Exclusive-NOR gate?

$F=xy^1 + x^1y$

$F=xy +x^1y^1$

22. What are the methods adopted to reduce Boolean function?

i) Karnaugh map
ii) Tabular method or Quine mccluskey method
iii) Variable entered map technique.

23. Why we go in for tabulation method?

This method can be applied to problems with many variables and has the advantage of being suitable for machine computation.

24. State the limitations of karnaugh map.

i) Generally it is limited to six variable map (i.e.) more then six variable involving expressions are not reduced.

ii) The map method is restricted in its capability since they are useful for simplifying only Boolean expression represented in standard form.

25. What is tabulation method?

A method involving an exhaustive tabular search method for the minimum expression to solve a Boolean equation is called as a tabulation method.

26. What are prime-implicants?

The terms remained unchecked are called prime-implecants. They cannot be reduced further.

27. Explain or list out the advantages and disadvantages of K-map method?

The advantages of the K-map method are

i. It is a fast method for simplifying expression up to four variables.
ii. It gives a visual method of logic simplification.
iii. Prime implicants and essential prime implicants are identified fast.
iv. Suitable for both SOP and POS forms of reduction.
v. It is more suitable for class room teachings on logic simplification.

The disadvantages of the K-map method are

i. It is not suitable for computer reduction.
ii. K-maps are not suitable when the number of variables involved exceed four.
iii. Care must be taken to fill in every cell with the relevant entry, such as a 0, 1 (or) don"t care terms.

28. List out the advantages and disadvantages of Quine-Mc Cluskey method?

The advantages are,

a. This is suitable when the number of variables exceed four.
b. Digital computers can be used to obtain the solution fast.
c. Essential prime implicants, which are not evident in K-map, can be clearly seen in the final results.

The disadvantages are,

a. Lengthy procedure than K-map.
b. Requires several grouping and steps as compared to K-map.
c. It is much slower.
d. No visual identification of reduction process.
e. The Quine Mc Cluskey method is essentially a computer reduction method.

29. Define Positive Logic.

When high voltage or more positive voltage level is associated with binary „1" and while the low or less positive level is associated with binary „0" then the system adhering to this is called positive logic.

30. Define Negative Logic.

When high voltage level is associated with binary „0" and while the low level is associated with binary „1" then the system adhering to this is called negative logic

31. List the characteristics of digital Ics

i) propagation delay
ii) power dissipation
iii) Fan-in
iv) Fan-out
v) Noise margin

32. What is propagation delay?

It is the average transition delay time for the signal to propagate from input to output when the signals change in value.

33. What is Noise margin?

It is the limit of a noise voltage, which may be present with out impairing the proper operation of the circuit.

34. What is power dissipation?

It is the power consumed by the gate, which must be available from the power supply.

35. Why parity checker is needed?

Parity checker is required at the receiver side to check whether the expected parity is equal to the calculated parity or not. If they are not equal then it is found that the received data has error.

36. What is meant by parity bit?

Parity bit is an extra bit included with a binary message to make the number of 1"s either odd or even. The message, including the parity bit is transmitted and then checked at the receiving and for errors.

37. Why parity generator necessary?

Parity generator is essential to generate parity bit in the transmitter.

38. What is IC?

An integrated circuit is a small silicon semiconductor crystal called a chip containing electrical components such as transistors, diodes, resistors and capacitors. The

various components are interconnected inside the chip to form an electronic circuit.

39. What are the needs for binary codes?

a. Code is used to represent letters, numbers and punctuation marks.
b. Coding is required for maximum efficiency in single transmission.
c. Binary codes are the major components in the synthesis (artificial generation) of speech and video signals.
d. By using error detecting codes, errors generated in signal transmission can be detected.
e. Codes are used for data compression by which large amounts of data are transmitted in very short duration of time.

40.Mention the different type of binary codes?

The various types of binary codes are,

f. BCD code (Binary Coded decimal).
g. Self-complementing code.
h. The excess-3 (X"s-3) code.
i. Gray code.
j. Binary weighted code.
k. Alphanumeric code.
l. The ASCII code.
m. Extended binary-coded decimal interchange code (EBCDIC).
n. Error-detecting and error-correcting code.
o. Hamming code.

41.List the advantages and disadvantages of BCD code? The advantages of BCD code are

a. Any large decimal number can be easily converted into corresponding binary number
b. A person needs to remember only the binary equivalents of decimal number from 0 to 9.
c. Conversion from BCD into decimal is also very easy.

The disadvantages of BCD code are

a. The code is least efficient. It requires several symbols to represent even small numbers.
b. Binary addition and subtraction can lead to wrong answer.
c. Special codes are required for arithmetic operations.
d. This is not a self-complementing code.
e. Conversion into other coding schemes requires special methods.

42. What is meant by self-complementing code?

A self-complementing code is the one in which the members of the number system complement on themselves. This requires the following two conditions to be satisfied.

a. The complement of the number should be obtained from that number by replacing 1s with 0s and 0s with 1s.

b. The sum of the number and its complement should be equal to decimal 9. Example of a self-complementing code is

i. 2-4-2-1 code.
ii. Excess-3 code.

43. Mention the advantages of ASCII code?

The following are the advantages of ASCII code

a. There are 2^7 =128 possible combinations. Hence, a large number of symbols, alphabets etc.., can be easily represented.
b. There is a definite order in which the alphabets, etc.., are assigned to each code word.
c. The parity bits can be added for error-detection and correction.

44. What are the disadvantages of ASCII code?

The disadvantages of ASCII code are

a. The length of the code is larger and hence more bandwidth is required for transmission.
b. With more characters and symbols to represent, this is not completely sufficient.

45. What is the truth table?

A truth table lists all possible combinations of inputs and the corresponding outputs.

46. Define figure of merit?

Figure of merits is defined as the product of speed and power. The speed is specified in terms of propagation delay time expressed in nano seconds.

47. What are the two types of logic circuits for digital systems? Combinational and sequential

48. Define Combinational circuit.

A combinational circuit consist of logic gates whose outputs at anytime are determined directly from the present combination of inputs without regard to previous inputs.

49. Define sequential circuits.

Their outputs are a function of the inputs and the state of memory elements. The state of memory elements, in turn, is a function of previous inputs.

50. What is a half-adder?

The combinational circuit that performs the addition of two bits are called a half-adder.

51. What is a full-adder?

The combinational circuit that performs the addition of three bits are called a half-adder.

52. What is half-subtractor?

The combinational circuit that performs the subtraction of two bits are called a half-sub tractor.

53. What is a full-subtractor?

The combinational circuit that performs the subtraction of three bits are called a half- sub tractor.

54. What is Binary parallel adder?

A binary parallel adder is a digital function that produces the arithemetic sum of two binary numbers in parallel.

55. What is BCD adder?

A BCD adder is a circuit that adds two BCD digits in parallel and produces a sum digit also in BCD.

56. What is Magnitude Comparator?

A Magnitude Comparator is a combinational circuit that compares two numbers, A and B and determines their relative magnitudes.

57. What is decoder?

A decoder is a combinational circuit that converts binary information from „n" input lines to a maximum of 2^n unique output lines.

58. What is encoder?

A decoder is a combinational circuit that converts binary information from 2^nInput lines to a maximum of „n" unique output lines.

59. Define Multiplexing?

Multiplexing means transmitting a large number of information units over a smaller number of channels or lines.

60. What is Demultiplexer?

A Demultiplexer is a circuit that receives information on a single line and transmits this information on one of 2^n possible output lines

.

61. Give the truth table for a half adder.

Input		Output	
X	Y	Sum (S)	Carry (C)
0	0	0	0

0	1	1	0
1	0	1	0

1	1	0	1

62. Give the truth table for a half Subtractor.

Input		Output	
X	Y	Borrow(B)	Diffe (D)
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

63.From the truth table of a half adder derive the logic equation S = X □ Y

C = X . Y

64. From the truth table of a half subractor derive the logic equation

D = X □ Y

$B = X^1 . Y$

65. From the truth table of a full adder derive the logic equation S = X □ Y □ Z

C = XY + YZ + XZ

66. What is code conversion?

If two systems working with different binary codes are to be synchronized in operation, then we need digital circuit which converts one system of codes to the other. The process of conversion is referred to as code conversion.

67. What is code converter?

It is a circuit that makes the two systems compatible even though each uses a different binary code. It is a device that converts binary signals from a source code to its output code. One example is a BCD to Xs3 converter.

68. What do you mean by analyzing a combinational circuit?

The reverse process for implementing a Boolean expression is called as analyzing a combinational circuit. (ie) the available logic diagram is analyzed step by step and finding the Boolean function

69. Give the applications of Demultiplexer.

i) It finds its application in Data transmission system with error detection.

ii) One simple application is binary to Decimal decoder.

70. Mention the uses of Demultiplexer.

Demultiplexer is used in computers when a same message has to be sent to different receivers. Not only in computers, but any time information from one source

can be fed to several places.

71. Give other name for Multiplexer and Demultiplexer.
Multiplexer is other wise called as Data selector.
Demultiplexer is otherwise called as Data distributor.

72. What is the function of the enable input in a Multiplexer?
The function of the enable input in a MUX is to control the operation of the unit.

73. Give the truth table for a full Subtractor.

Input			Output	
X	Y	Z	Borrow (B)	Diffe (D)
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

74. Give the truth table for a full adder.

Input			Output	
X	Y	Z	Sum (S)	Carry (C)
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

75. From the truth table of a full subtractor derive the logic equation S = X □ Y □ Z

$C = X^1Y + YZ + X^1Z$

76. What is priority encoder?

A priority encoder is an encoder that includes the priority function. The operation of the priority encoder is such that if two or more inputs are equal to 1 at the same time, the input having the highest priority will take precedence.

77. Can a decoder function as a Demultiplexer?

Yes. A decoder with enable can function as a Demultiplexer if the enable line E is taken as a data input line A and B are taken as selection lines.

78. List out the applications of multiplexer?

The various applications of multiplexer are

a. Data routing.

b. Logic function generator.

c. Control sequencer.

d. Parallel-to-serial converter.

79. List out the applications of decoder?

The applications of decoder are

a. Decoders are used in counter system.

b. They are used in analog to digital converter.

c. Decoder outputs can be used to drive a display system.

80. List out the applications of comparators?

The following are the applications of comparator

a. Comparators are used as a part of the address decoding circuitry in computers to select a specific input/output device for the storage of data.

b. They are used to actuate circuitry to drive the physical variable towards the reference value.

c. They are used in control applications.

81. What are the applications of seven segment displays?

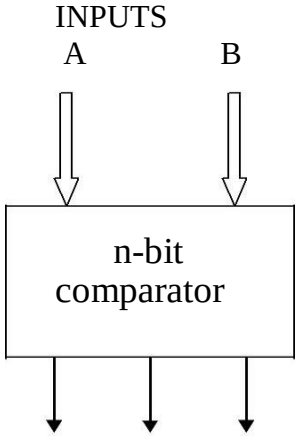
The seven segment displays are used in

a. LED displays

b. LCD displays

82. What is digital comparator?

A comparator is a special combinational circuit designed primarily to compare the relative magnitude of two binary numbers.



A>B A=B A<B
OUTPUTS
Block diagram of n-bit comparator

83. List the types of ROM.
i) Programmable ROM (PROM)
ii) Erasable ROM (EPROM)
iii) Electrically Erasable ROM (EEROM)

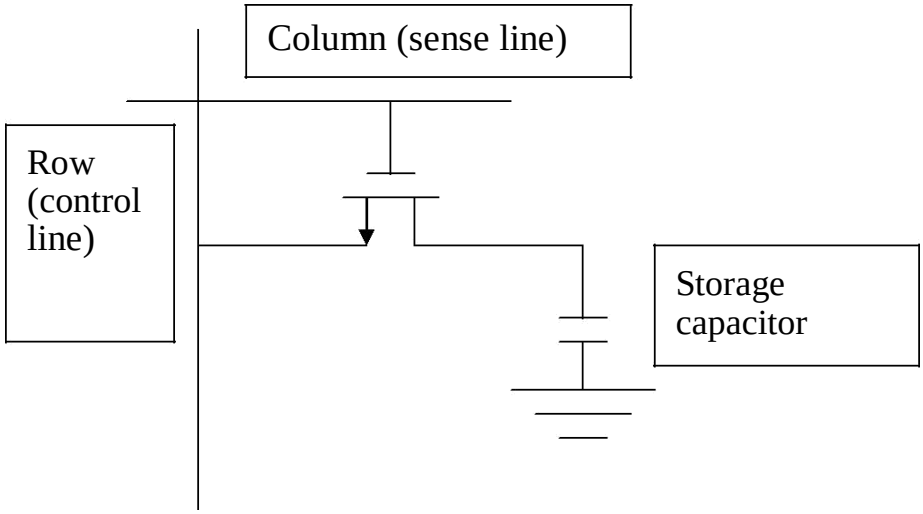
84. Differentiate ROM & PLD"s

ROM (Read Only Memory)	PLD"s (Programmable Logic Array)
1.It is a device that includes both the decoder and the OR gates with in a single IC package	1.It is a device that includes both AND and OR gates with in a single IC package
2.ROM does not full decoding of the variables and does generate all the minterms	2.PLD"s does not provide full decoding of the variable and does not generate all the minterms

85. What are the different types of RAM?
The different types of RAM are
a. NMOS RAM (Nitride Metal Oxide Semiconductor RAM)
b. CMOS RAM (Complementary Metal Oxide Semiconductor RAM)
c. Schottky TTL RAM
d. ELL RAM.

86. What are the types of arrays in RAM?
RAM has two type of array namely,
a. Linear array
b. Coincident array

87. Explain DRAM?
The dynamic RAM (DRAM) is an operating mod, which stores the binary information in the form of electric charges on capacitors.
The capacitors are provided inside the chip by MOS transistors.



DRAM cell

The stored charges on the capacitors tend to discharge with time and the capacitors must be tending to discharge with time and the capacitors must be periodically recharged by refreshing the dynamic memory.

DRAM offers reduced power consumption and larger storage capacity in a single memory chip.

88. Explain SRAM?

Static RAM (SRAM) consists of internal latches that store the binary information. The stored information remains valid as long as the power is applied to the unit.

SRAM is easier to use and has shorter read and write cycle.

The memory capacity of a static RAM varies from 64 bit to 1 mega bit.

89. Differentiate volatile and non-volatile memory?

Volatile memory	Non-volatile memory
They are memory units which lose stored information when power is turned off. E.g. SRAM and DRAM	It retains stored information when power is turned off. E.g. Magnetic disc and ROM

90. What are the terms that determine the size of a PAL?

The size of a PLA is specified by the

a. Number of inputs
b. Number of products terms
c. Number of outputs

91. What are the advantages of RAM?

The advantages of RAM are

a. Non-destructive read out

b. Fast operating speed
c. Low power dissipation
d. Compatibility
e. Economy

100. What is meant by memory decoding?

The memory IC used in a digital system is selected or enabled only for the range of addresses assigned to it .

101. What is access and cycle time?

The access time of the memory is the time to select word and read it. The cycle time of a memory is a time required to complete a write operation.

102. What is sequential circuit?

Sequential circuit is a broad category of digital circuit whose logic states depend on a specified time sequence. A sequential circuit consists of a combinational circuit to which memory elements are connected to form a feedback path.

103. List the classifications of sequential circuit.

i) Synchronous sequential circuit.
ii) Asynchronous sequential circuit.

104. What is Synchronous sequential circuit?

A Synchronous sequential circuit is a system whose behavior can be defined from the knowledge of its signal at discrete instants of time.

105. What is clocked sequential circuits?

Synchronous sequential circuit that use clock pulses in the inputs of memory elements are called clocked sequential circuit. One advantage as that they don"t cause instability problems.

106. What is called latch?

Latch is a simple memory element, which consists of a pair of logic gates with their inputs and outputs inter connected in a feedback arrangement, which permits a single bit to be stored.

107. List different types of flip-flops.

i) SR flip-flop
ii) Clocked RS flip-flop
iii) D flip-flop
iv) T flip-flop
v) JK flip-flop
vi) JK master slave flip-flop

108. What do you mean by triggering of flip-flop.

The state of a flip-flop is switched by a momentary change in the input signal. This momentary change is called a trigger and the transition it causes is said to trigger the flip-flop

109. What is an excitation table?

During the design process we usually know the transition from present state to next state and wish to find the flip-flop input conditions that will cause the required transition. A table which lists the required inputs for a given chance of state is called an excitation table.

110. Give the excitation table of a JK flip-flop

Q(t)	Q(t+1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

111. Give the excitation table of a SR flip-flop

Q(t)	Q(t+1)	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

112. Give the excitation table of a T flip-flop

Q(t)	Q(t+1)	T
0	0	0
0	1	1
1	0	1
1	1	0

113. Give the excitation table of a D flip-flop

Q(t)	Q(t+1)	T
0	0	0
0	1	1
1	0	0
1	1	1

114. What is a characteristic table?

A characteristic table defines the logical property of the flip-flop and completely characteristic its operation.

115. Give the characteristic equation of a SR flip-flop. $Q(t+1)=S+R^1Q$

116. Give the characteristic equation of a D flip-flop. $Q(t+1)=D$

117. Give the characteristic equation of a JK flip-flop. $Q(t+1)=JQ^1+K^1Q$

118. Give the characteristic equation of a T flip-flop. $Q(t+1)=TQ^1+T^1Q$

119. What is the difference between truth table and excitation table.

i) An excitation table is a table that lists the required inputs for a given change of state.

ii) A truth table is a table indicating the output of a logic circuit for various input states.

120. What is counter?

A counter is used to count pulse and give the output in binary form.

121. What is synchronous counter?

In a synchronous counter, the clock pulse is applied simultaneously to all flip-flops. The output of the flip-flops change state at the same instant. The speed of operation is high compared to an asynchronous counter

122. What is Asynchronous counter?

In a Asynchronous counter, the clock pulse is applied to the first flip-flops. The change of state in the output of this flip-flop serves as a clock pulse to the next flip-flop and so on. Here all the flip-flops do not change state at the same instant and hence speed is less.

123. What is the difference between synchronous and asynchronous counter?

Sl.No.	Synchronous counter	Asynchronous counter
1.	Clock pulse is applied simultaneously	Clock pulse is applied to the first flip-flop, the change of output is given as clock to next flip-flop
2.	Speed of operation is high	Speed of operation is low.

124. Name the different types of counter.

a) Synchronous counter
b) Asynchronous counter
   i) Up counter
   ii) Down counter
   iii) Modulo – N counter
   iv) Up/Down counter

125. What is up counter?

A counter that increments the output by one binary number each time a clock pulse is applied.

126. What is down counter?

A counter that decrements the output by one binary number each time a clock pulse is applied.

127. What is up/down counter?

A counter, which is capable of operating as an up counter or down counter, depending on a control lead.

128. What is a ripple counter?

A ripple counter is nothing but an asynchronous counter, in which the output of the flip-flop change state like a ripple in water.

129. What are the uses of a counter?

i) The digital clock
ii) Auto parking control
iii) Parallel to serial data conversion.

130. What is meant by modulus of a counter?

By the term modulus of a counter we say it is the number of states through which a counter can progress.

131. What is meant by natural count of a counter?

By the term natural count of a counter we say that the maximum number of states through which a counter can progress.

132.A ripple counter is a ---------- sequential counter.
Ans: Synchronous.

133. What is a modulo counter?
A counter that counts from 0 to T is called as modulo counter.

134. A counter that counts from to T is called a modulo counter. True or False. Ans: True

135. The number of flip-flops required for modulo-18 counter is -------
Ans: five.

136. Form the truth table for 3-bit binary down counter.

Clk	Q2	Q1	Q0
1	1	1	1
1	1	1	0
1	1	0	1
1	1	0	0
1	0	1	1
1	0	1	0
1	0	0	1
1	0	0	0
1	1	1	1

137. What is a ring counter?

A counter formed by circulating a „bit" in a shift register whose serial output has been connected to its serial input.

138. What is BCD counter?

A BCD counter counts in binary coded decimal from 0000 to 1001 and back to 0000. Because of the return to 0000 after a count of 1001, a BCD counter does not have a regular pattern as in a straight binary counter.

139. What are the uses of a ring counter?

i) Control section of a digital system.
ii) Controlling events, which occur in strict time sequence.

140. What is a register?

Memory elements capable of storing one binary word. It consists of a group of flip-flops, which store the binary information.

141. What is Johnson counter?

It is a ring counter in which the inverted output is fed into the input. It is also know as a twisted ring counter.

142. What is a shift register?

In digital circuits, datas are needed to be moved into a register (shift in) or moved out of a register (shift out). A group of flip-flops having either or both of these facilities is called a shift register.

143. What is serial shifting?

In a shift register, if the data is moved 1 bit at a time in a serial fashion, then the technique is called serial shifting.

144. What is parallel shifting?

In a shift register all the data are moved simultaneously and then the technique is called parallel shifting.

145. Write the uses of a shift register.

i) Temporary data storage
ii) Bit manipulations.

146. What is a cycle counter?

A cycle counter is a counter that outputs a stated number of counts and then stops.

147. Define state of sequential circuit?

The binary information stored in the memory elements at any given time defines the “state” of sequential circuits.

148. Define state diagram.

A graphical representation of a state table is called a state diagram.

149. What is the use of state diagram?

i) Behavior of a state machine can be analyzed rapidly.
ii) It can be used to design a machine from a set of specification.

150. What is state table?

A table, which consists time sequence of inputs, outputs and flip-flop states, is called state table. Generally it consists of three section present state, next state and output.

151. What is a state equation?

A state equation also called, as an application equation is an algebraic expression that specifies the condition for a flip-flop state transition. The left side of the equation denotes the next state of the flip-flop and the right side; a Boolean function specifies the present state.

152. What is meant by race around condition?

In JK flip-flop output is fed back to the input, and therefore changes in the output results change in the input. Due to this in the positive half of the clock pulse if J and K are both high then output toggles continuously. This condition is known as race around condition.

153. How many bits would be required for the product register if the multiplier has 6 bits and the multiplicand has 8 bits?

The product register is 14-bit width with extra bit at the left end indicating a temporary storage for any carry, which is generated when the multiplicand is added to the accumulator.

154. What is ASM chart?

Just as flow charts are useful in software design, flow charts are useful in the hardware design of digital systems. These flow charts are called as State Machine Flow Charts or SM charts. SM charts are also called as ASMC (Algorithmic State machine chart). ASM chart describes the sequential operation in a digital system.

155. What are the three principal components of SM charts?

The 3 principal components of SM charts are state box, decision box & Conditional output box.

156. What is decision box?

A diamond shaped symbol with true or false branches represents a decision box. The condition placed in the box is a Boolean expression that is evaluated to determine which branch to take in SM chart.

157. What is link path? How many entrance paths & exit paths are there in SM block? A path through an SM block from entrance to exit is referred to as link path. An SM block has one entrance and exit path.

158. Differentiate ASM chart and conventional flow chart?

A conventional Flow chart describes the sequence of procedural steps and decision paths for an algorithm without concern for their time relationship.

The ASM chart describes the sequence of events as well as the timing relationships between the states of a sequential controller and the events that occur while going from one state to the next.

159. What is flow table?

During the design of synchronous sequential circuits, it is more convenient to name the states by letter symbols without making specific reference to their binary values. Such table is called Flow table.

160. What is primitive flow table?

A flow table is called Primitive flow table because it has only one stable state

in each row.

161. Define race condition.

A race condition is said to exist in a synchronous sequential circuit when two or more binary state variables change, the race is called non-critical race.

162. Define critical & non-critical race with example.

The final stable state that the circuit reaches does not depend on the order in which the state variables change, the race is called non-critical race.

The final stable state that the circuit reaches depends on the order in which the state variables change, the race is called critical race.

163. How can a race be avoided?

Races can be avoided by directing the circuit through intermediate unstable states with a unique state – variable change.

164. Define cycle and merging?

When *a* circuit goes through a u*n*ique sequence of unstable states, *i*t is said to have a cyc*l*e.

The *g*rouping of stab*l*e states from s*e*parate rows into o*n*e common row is called merging.

165. Give state – reduction procedure.

The state – reduction procedure for co*m*pletely specified st*a*te tables is base*d* on t*h*e algorithm that two state*s* in a state table can be combined in to one if they can be shown to be equi*va*le*n*t.

166. Define hazards.

Hazards are unwanted switching transients that may appear at the output of a circuit because different paths exhibit different propagation delays.

167. Does Hazard occur in sequential circuit? If so what is the problem caused?

Yes, Hazards occur in sequential circuit that is Asynchronous sequential circuit. It may result in a transition to a wrong state.

168. Give the procedural steps for determining the compatibles used for the purpose of merging a flow table.

The purpose that must be applied in order to find a suitable group of compatibles for the purpose of merging a flow table can be divided into 3 procedural steps.

i. Determine all compatible pairs by using the implication table.
ii. Find the maximal compatibles using a Merger diagram
iii. Find a minimal collection of compatibles that covers all the states and is closed.

169. What are the types of hazards?

The 3 types of hazards are 1) Static – 0 hazards
2) Static – 1 hazard

3) Dynamic hazards

170. What is mealy and Moore circuit?

Mealy circuit is a network where the output is a function of both present state and input.

Moore circuit is a network where the output is function of only present state.

171. Differentiate Moore circuit and Mealy circuit?

Moore circuit	Mealy circuit
a. It is output is a function of present state only. b. Input changes do not affect the output. c. Moore circuit requires more number of states for implementing same function.	a. It is output is a function of present state as well as the present input. b. Input changes may affect the output of the circuit. c. It requires less numbers of states for implementing same function.

172. How can the hazards in combinational circuit be removed?

Hazards in the combinational circuits can be removed by covering any two min terms that may produce a hazard with a product term common to both. The removal of hazards requires the addition of redundant gates to the circuit.

173. How does an essential hazard occur?

An essential hazard occurs due to unequal delays along two or more paths that originate from the same input. An excessive delay through an inverter circuit in comparison to the delay associated with the feedback path causes essential hazard.

174. What is Timing diagram?

Timing diagrams are frequently used in the analysis of sequential network. These diagrams show various signals in the network as a function of time.

175. What is setup and hold time?

The definite time in which the input must be maintained at a constant value prior to the application of the pulse is setup time

The definite time is which the input must not chance after the application of the positive or negative going transition of the pulse based on the triggering of the pulse.

176. Define bit time and word time.

The time interval between clock pulses is called bit time.

The time required to shift the entire contents of a shift register is called word time

177. What is bi-directional shift register and unidirectional shift register?

A register capable of shifting both right and left is called bi-directional shift register.

A register capable of shifting only one direction is called unidirectional shift register.

178. Define equivalent state.

If a state machine is started from either of two states and identical output sequences are generated from every possible set of sequences, then the two states are said to be equivalent.

179. If a shift register can be operated in all possible ways then it is called as----------

Ans: Univerasal register: It can be operated in all possible modes with bi-directional shift facility.

180. What is gate delay?

If the change in output is delayed by a time □ with respect to the input. We say that the gate has a propagation delay of □. Normally propagation delay for 0 to 1 output (□1) may be different than the delay for 1 to 0 changes (□2).

181. Define state reduction algorithm.

State reduction algorithm is stated as "Two states are said to be equivalent if, for each member of the set of inputs they give the same output and send the circuit either to the same state or to an equivalent state. When two states are equivalent, one of them can be removed without altering the input-output relation.

182. What is meant by level triggering?

In level triggering the output of the flip-flop changes state or responds only when the clock pulse is present.

183. Write the uses of a shift register.

i) Temporary data storage.
ii) Bit manipulations.

184. What is meant by flow table?

During the design of asynchronous sequential circuits, it is more convenient to name the states by letter symbols without making specific reference to their binary values. Such a table is called a flow table.

185. What are the problems involved in asynchronous circuits?

The asynchronous sequential circuits have three problems namely,

a. Cycles
b. Races
c. Hazards

186. Define cycles?

If an input change includes a feedback transition through more than unstable state then such a situation is called a cycle.

187. Define primitive flow table?

A primitive flow table is a flow table with only one stable total state in each row. Remember that a total state consists of the internal state combined with the input.

188. Define merging?

The primitive flow table has only one stable state in each row. The table can be reduced to a smaller numbers of rows if two or more stable states are placed in the same row of the flow table. The grouping of stable states from separate rows into one common row is called merging.

189. What are the types of Registers?

i) Buffer Register
ii) Shift Register
iii) Uni directional Shift Register
iv) Bidirectional Shift Register