

CHAPTER 4

DIGITAL LOGIC FAMILIES

4.1 INTRODUCTION

The switching characteristics of semiconductor devices have been discussed in Chapter 3. Basically, there are two types of semiconductor devices: bipolar and unipolar. Based on these devices, digital integrated circuits have been made which are commercially available. Various digital functions are being fabricated in a variety of forms using bipolar and unipolar technologies. A group of compatible ICs with the same logic levels and supply voltages for performing various logic functions have been fabricated using a specific circuit configuration which is referred to as a *logic family*.

4.1.1 Bipolar Logic Families

The main elements of a bipolar IC are resistors, diodes (which are also capacitors) and transistors. Basically, there are two types of operations in bipolar ICs:

1. Saturated, and
2. Non-saturated.

In saturated logic, the transistors in the IC are driven to saturation, whereas in the case of non-saturated logic, the transistors are not driven into saturation.

The saturated bipolar logic families are:

1. Resistor-transistor logic (RTL),
2. Direct-coupled transistor logic (DCTL),
3. Integrated-injection logic (I^2L),
4. Diode-transistor logic (DTL),
5. High-threshold logic (HTL), and
6. Transistor-transistor logic (TTL).

The non-saturated bipolar logic families are:

1. Schottky TTL, and
2. Emitter-coupled logic (ECL).

4.1.2 Unipolar Logic Families

MOS devices are unipolar devices and only MOSFETs are employed in MOS logic circuits.

The MOS logic families are:

1. PMOS,
2. NMOS, and
3. CMOS (5-V and low-voltage CMOS)

While in PMOS only *p*-channel MOSFETs are used and in NMOS only *n*-channel MOSFETs are used, in complementary MOS (CMOS), both *p*- and *n*-channel MOSFETs are employed and are fabricated on the same silicon chip.

4.1.3 BiCMOS Logic Family

BiCMOS logic circuits use CMOS devices for input and logic operations and bipolar devices for output.

4.2 CHARACTERISTICS OF DIGITAL ICs

With the widespread use of ICs in digital systems and with the development of various technologies for the fabrication of ICs, it has become necessary to be familiar with the characteristics of IC logic families and their relative advantages and disadvantages. Digital ICs are classified either according to the complexity of the circuit, as the relative number of individual basic gates (2-input NAND gates) it would require to build the circuit to accomplish the same logic function or the number of components fabricated on the chip. The classification of digital ICs is given in Table 4.1.

Table 4.1 *Classification of Digital ICs*

IC Classification	Equivalent individual basic gates	Number of components
Small-scale integration (SSI)	Less than 12	Up to 99
Medium-scale integration (MSI)	12–99	100–999
Large-scale integration (LSI)	100–999	1,000–9,999
Very large-scale integration (VLSI)	Above 1,000	Above 10,000

The various characteristics of digital ICs used to compare their performances are:

1. Speed of operation,
2. Power dissipation,
3. Figure of merit,
4. Fan-out,
5. Current and voltage parameters,
6. Noise immunity,
7. Operating temperature range,
8. Power supply requirements, and
9. Flexibilities available.

4.2.1 Speed of Operation

The speed of a digital circuit is specified in terms of the propagation delay time. The input and output waveforms of a logic gate are shown in Fig. 4.1. The delay times are measured between the 50 per cent voltage levels of input and output waveforms.

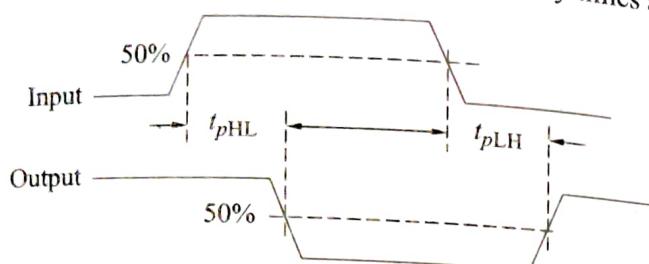


Fig. 4.1 *Input and Output Voltage Waveforms to Define Propagation Delay Times*

There are two delay times: t_{PHL} , when the output goes from the HIGH state to the LOW state and t_{PLH} , corresponding to the output making a transition from the LOW state to the HIGH state. The propagation delay time of the logic gate is taken as the average of these two delay times.

4.2.2 Power Dissipation

This is the amount of power dissipated in an IC. It is determined by the current, I_{CC} , that it draws from the V_{CC} supply, and is given by $V_{CC} \times I_{CC}$. I_{CC} is the average value of $I_{CC}(0)$ and $I_{CC}(1)$. This power is specified in milliwatts. It is known as static power dissipation, i.e., the power consumed by the circuit when input signals are not changing.

4.2.3 Figure of Merit

The figure of merit of a digital IC is defined as the product of speed and power. The speed is specified in terms of propagation delay time expressed in nanoseconds.

$$\text{Figure of merit} = \text{propagation delay time (ns)} \times \text{power (mW)}$$

It is specified in pico joules ($\text{ns} \times \text{mW} = \text{pJ}$)

A low value of speed-power product is desirable. In a digital circuit, if it is desired to have high speed, i.e. low propagation delay, then there is a corresponding increase in the power dissipation and vice-versa.

4.2.4 Fan-Out

This is the number of similar gates which can be driven by a gate. High fan-out is advantageous because it reduces the need for additional drivers to drive more gates.

4.2.5 Current and Voltage Parameters

The following currents and voltages are specified which are very useful in the design of digital systems.

High-level input voltage, V_{IH} : This is the minimum input voltage which is recognised by the gate as logic 1.

Low-level input voltage, V_{IL} : This is the maximum input voltage which is recognised by the gate as logic 0.

High-level output voltage, V_{OH} : This is the minimum voltage available at the output corresponding to logic 1.

Low-level output voltage, V_{OL} : This is the maximum voltage available at the output corresponding to logic 0.

High-level input current, I_{IH} : This is the minimum current which must be supplied by a driving source corresponding to 1 level voltage.

Low-level input current, I_{IH} : This is the minimum current which must be supplied by a driving source corresponding to 0 level voltage.

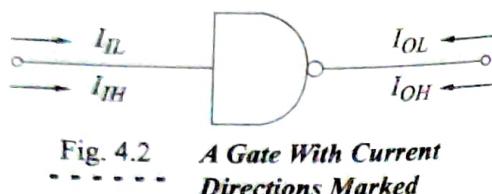
High-level output current, I_{OH} : This is the maximum current which the gate can sink in 1 level.

Low-level output current, I_{OL} : This is the maximum current which the gate can sink in 0 level.

High-level supply current, $I_{CC}(1)$: This is the supply current when the output of the gate is at logic 1.

Low-level supply current, $I_{CC}(0)$: This is the supply current when the output of the gate is at logic (0).

The current directions are illustrated in Fig. 4.2.



4.2.6 Noise Immunity

The input and output voltage levels defined above are shown in Fig. 4.3. Stray electric and magnetic fields may induce unwanted voltages, known as *noise*, on the connecting wires between logic circuits. This may

cause the voltage at the input to a logic circuit to drop below V_{IH} or rise above V_{IL} and may produce undesired operation. The circuit's ability to tolerate noise signals is referred to as the *noise immunity*, a quantitative measure of which is called *noise margin*. Noise margins are illustrated in Fig. 4.3.

The noise margins defined above are referred to as *dc noise margins*. Strictly speaking, the noise is generally thought of as an a.c. signal with amplitude and pulse width. For high speed ICs, a pulse width of a few microseconds is extremely long in comparison to the propagation delay time of the circuit and therefore, may be treated as d.c. as far as the response of the logic circuit is concerned. As the noise pulse width decreases and

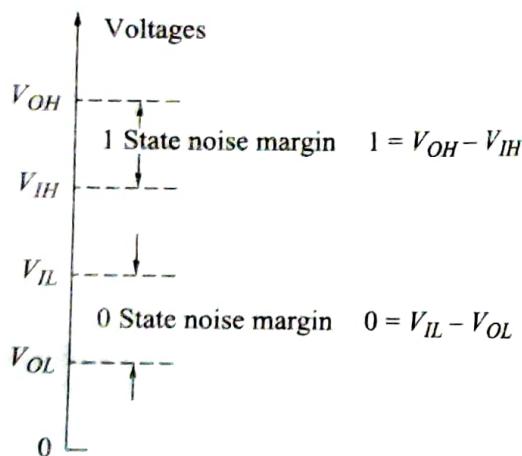


Fig. 4.3 Voltage Levels and Noise Margins of ICs

approaches the propagation delay time of the circuit, the pulse duration is too short for the circuit to respond. Under this condition, a large pulse amplitude would be required to produce a change in the circuit output. This means that a logic circuit can effectively tolerate a large noise amplitude if the noise is of a very short duration. This is referred to as *ac noise margin* and is substantially greater than the dc noise margin. It is generally supplied by the manufacturers in the form of a curve between noise margin and noise pulse width.

4.2.7 Operating Temperature

The temperature range in which an IC functions properly must be known. The accepted temperature ranges are: 0 to +70 °C for consumer and industrial applications and -55 °C to +125 °C for military purposes.

4.2.8 Power Supply Requirements

The supply voltage(s) and the amount of power required by an IC are important characteristics required to choose the proper power supply.

10/9/15

14
Pg. NO. ④, Q. 1(a).

(3)

Q1(a) Explain the current sourcing and sinking when two standard TTL gates are connected

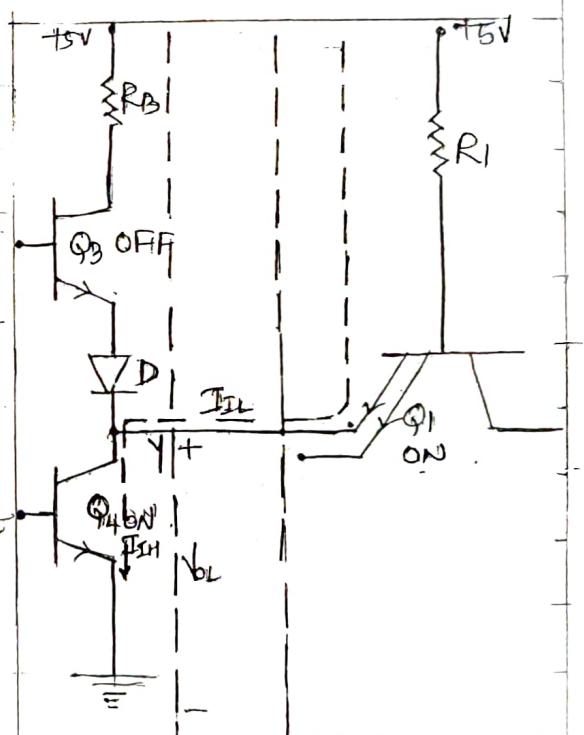
→ Sink current refers to the maximum amount of current that a pin can absorb through a load connected to an external supply. Sinking occurs through an NPN transistor, which provides a path to zero potential or ground.

Source current refers to the amount of current a pin can supply to drive an externally connected load such as an LED.

Sourcing occurs through a PNP transistor, which provides a path to the positive potential.

Current Sinking Action -

- The current sinking will take place when output $Y=0$.
- The current flows from the input stage of load gate into the output stage of driving gate.
- Transistor Q_4 of the driving gate acts as a closed switch because it is in saturation. This will forward bias the base emitter junction of Q_1 of the load gate.



Output

Circuit of
driving gate
with $Y=0$

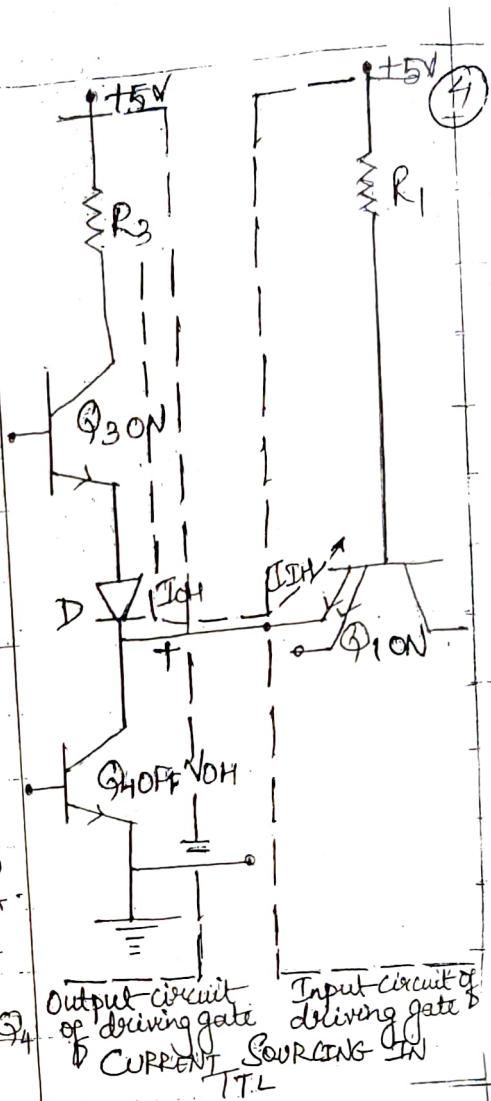
GATE SINKS CURRENT FROM
LOAD GATE

Input circuit of
driving gate

THE DRIVING

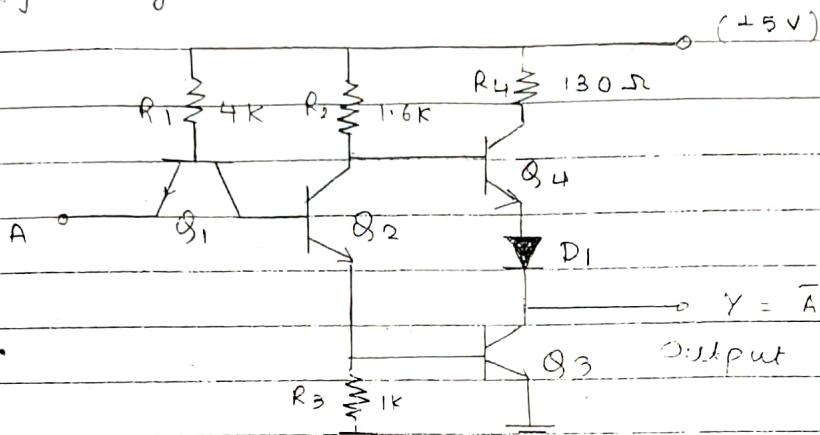
Current Sourcing Action -

- Current sourcing will take place when the output of the driving gate (Y) becomes HIGH i.e. $Y=1$.
- Q_3 will be ON and Q_4 will be OFF. So the base-emitter junction of Q_1 of the load gate will be reverse biased.
- Therefore, the reverse leakage current of Q_1 will be supplied (source) by the driving gate.
- Note that this current is extremely small typically of the order of 10 nA .
- Transistor Q_3 is called as current sourcing or pull up transistor and Q_4 is called as sinking or pull down transistor.



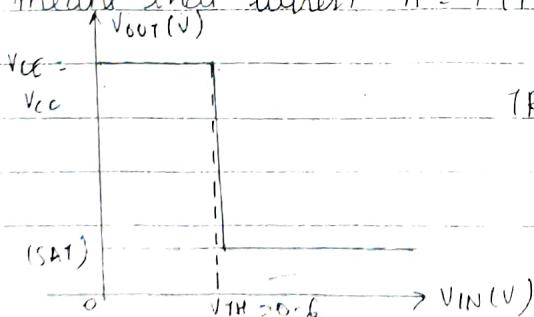
BASIC TTL INVERTER (NOT GATE)

The NAND gate explained above can be used as NOT gate by shorting A and B. The TTL-NOT gate is that circuit in which the Q₁ transistor of NAND has only one input. Such modified circuit is shown in the figure given below.



If A is low (i.e. 0V), the base of Q₁ is pulled down to approximately 0.7V. This reduces the base voltage of Q₂ to almost zero. Therefore, Q₂ is off and hence Q₃ goes to cutoff and Q₄ base is pulled high. Since Q₄ acts as an emitter follower, the Y output is high i.e. 1 when A=0, then Y=1.

On the other hand, if A is high (1 state) voltage, the emitter diode of Q₁ stop conducting and collector diode goes into forward conduction. This makes base of Q₂ high, as a result Q₃ goes into saturation producing low output at Y. This means that when A=1 (high) then Y=0 (low).



TRANSFER CHARACTERISTICS

Q Interfacing between CMOS and TTL (Pg. 3, Q. 1(a))

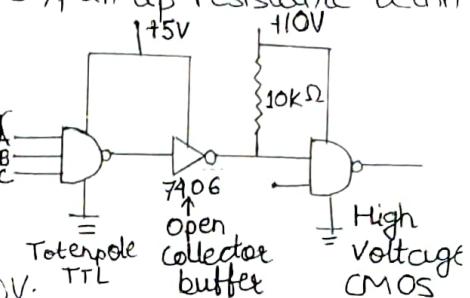
(5)

Ans. (i) TTL to CMOS INTERFACING:

- (a) It is necessary to check whether driving IC is capable of meeting I & V requirements of load IC or not.
- (b) Output current capability of TTL ICs is much higher than input current values of CMOS ICs.
- (c) There is problem when we compare voltage levels of TTL and CMOS. Because V_{OH} of TTL series is very low compared with $V_{IH(\min)}$ required for CMOS.
- (d) Due to presence of pull up resistor will result in rise in TTL output to $+5V$ in HIGH state.
- (e) This will provide sufficient voltage level at input of CMOS gate:
 - TTL driving High Voltage CMOS:-

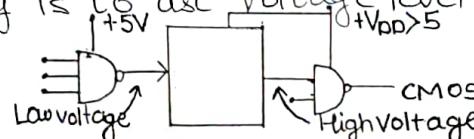
(a) For CMOS ICs operated with $V_{DD} > 5$, pull up resistance technique cannot be used.

(b) When it is not possible to pull up totempole output to $+V_{DD}$, a buffer is used as interface between TTL totempole and high voltage CMOS gate. 7406 is inverting CMOS buffer. Output voltage rating is 30V.



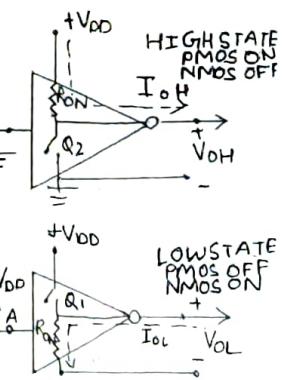
• Interfacing using level shifter:-

Another way is to use voltage level translator



(ii) CMOS TO TTL INTERFACING:-

- (a) CMOS has different characteristics in the two different states
 - HIGH state and LOW state output.
- (b) In HIGH state output, PMOS ' Q_1 ' will be ON and replaced by R_{on} while the NMOS will be OFF and replaced by open switch. Output is equivalent to V_{DD} with source resistance ranging from 100 to 1000Ω .
- (c) In LOW state output, PMOS ' Q_1 ' is OFF and represented by open switch while NMOS ' Q_2 ' is ON and replaced by R_{on} . CMOS inverter acts as a low resistance connected to ground and sinks current.



(d) CMOS can drive TTL in high state as CMOS output can apply sufficient voltage and current to TTL inputs. But in low state the output current I_{OL} is not enough to drive even one single TTL gate with $I_{IL} = 1.6\text{mA}$. This can be solved by inverting the output to increase current sourcing.

Compare performance of TTL CMOS ECL logic

TTL	CMOS	ECL
<ul style="list-style-type: none"> Bipolar saturated logic family Class of digital circuits built from BJT and resistors is called TTL It is called TTL as both the logic gating function and amplifying functions are performed by transistors TTL logic has high speed of operation but it's substantially slower than ECL logic 	<ul style="list-style-type: none"> Unipolar logic family CMOS is a technology for construction of ICs Metal Oxide semiconductor is field effect transistor having a metal gate electrode placed on top of an oxide insulator which turn is on top of a semiconductor More delayed as compared to TTL logic in operations. 	<ul style="list-style-type: none"> Bipolar non-saturated logic family High-speed integrated circuit bipolar logic family. Consist of an overdriving BJT diff amplifier with single-ended input and limited current to avoid the saturated region of operation and its slow turn off behaviour.
<ul style="list-style-type: none"> TTL devices have more power consumption than the CMOS at first but does not increase with clock speed as rapidly as for CMOS devices. 	<ul style="list-style-type: none"> Least power consumption CMOS does not produce as much waste heat as other logic families. Low static power consumption. 	<ul style="list-style-type: none"> In ECL transistors are never in saturation, input & output voltages have a small swing, input impedance is too high and output resistance is low. So transistor change state quickly. High speed of operation and low delay.
<ul style="list-style-type: none"> <u>Features</u> High fanout less power dissipation high speed of operation 	<ul style="list-style-type: none"> Lowest power dissipation Wide range of supply voltage Highest fanout among all logic families 	<ul style="list-style-type: none"> Relatively less power noise less susceptible to side channel attack Less propagation time Less than anandco High fanout
<ul style="list-style-type: none"> <u>Apps</u> Processors of mini computers and mainframe 	<ul style="list-style-type: none"> microprocessors, microcontrollers, static RAM 	<ul style="list-style-type: none"> oscillators, modulators/demodulators, frequency doublers etc.

1. Compare TTL and CMOS Logic families.

(7)

Parameters	TTL	CMOS
1. Switching Speed	More than CMOS	Less than TTL
2. Speed Power Product	100 pJ	10.5 pJ
3. Propagation Delay	10 ns for standard TTL	10.5 ns (for metal gate CMOS).
4. Power Supply Voltage	Fixed at 5V	Flexible from 3V to 15V.
5. Power dissipation per gate	10 mW	0.1 mW and hence it is used for battery backup applications
6. Device used	Bipolar Junction Transistor	N-channel MOSFET and P-channel MOSFET.
7. Component Density	Less than CMOS since BJT needs more space	More than TTL since MOSFET need smaller space while fabricating an IC.
8. Operating areas	Transistors are operated in saturation or cut-off region.	MOSFET's are operated as switches i.e. in cut-off or ohmic region.

(8)

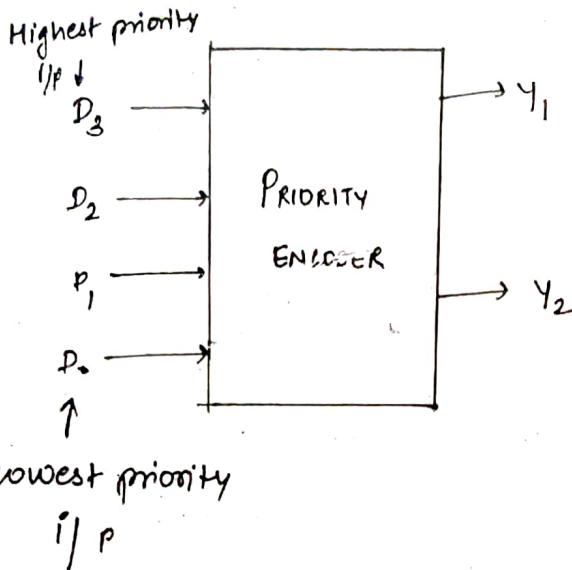
Parameters	TTL	CMOS
9. Unconnected inputs	Input can remain floating. Floating input are treated as logic 1.	Unused inputs should be returned to ground or V_{DD} . They should never be left floating.
10. High level noise margin	0.4 V.	1.45 V.
11. Noise immunity	Less than CMOS	Better than TTL
12. $V_{IH\min}$	2V	3.5V ($V_{DD} = 5V$)
13. $V_{IL\max}$	0.8V	1.5V
14. $V_{OH\min}$	2.7V	4.95V.
15. $V_{OL\max}$	0.4V	0.05V.

Q 7(b)

write a short note on priority encoders? (9)

Pg 22

~~Soln.~~



We know that encoders have ' n ' number of I/P lines and m number of O/P lines where an m bit binary code corresponding to a digital number is produced as an O/P.

In priority encoders, priorities are given to input lines. If two or more I/P lines are '1' at the same time, then the I/P line with the highest priority will be considered.

There are 4 inputs D_0 to D_3 and two outputs Y_0 & Y_1 . D_3 has the highest priority and D_0 has the lowest. This implies that if $D_3 = 1$ then O/P is 11 & if (Y_1, Y_0)

$D_3 = 0, D_2 = 1$ then the O/P is $Y_1 = 1, Y_0 = 0$.

Truth table :-

Highest	Inputs		lowest	outputs	
D ₃	D ₂	D ₁	D ₀	Y ₁	Y ₀
0	0	0	0	X	X
0	0	0	1	0	0
0	0	1	X	0	1
0	1	X	X	1	0
1	X	X	X	1	1

Kmaps

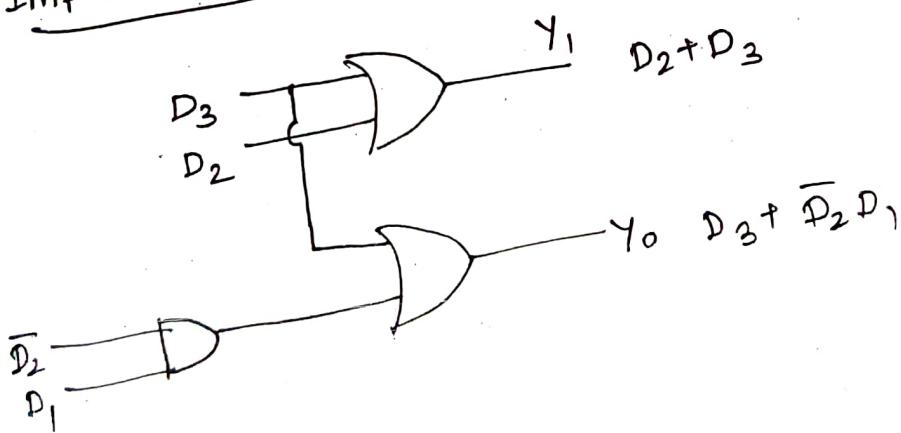
		D ₁	D ₀	00	01	11	10
		D ₂	D ₃	00	01	11	10
		00	00	X	0	0	0
		01	01	1	1	1	1
		11	11	1	1	1	1
		10	10	1	1	1	1

$$Y_1 = D_3 + D_2$$

		D ₁	D ₀	00	01	11	10
		D ₂	D ₃	00	01	11	10
		00	00	X	0	1	1
		01	01	0	0	0	0
		11	11	1	1	1	1
		10	10	1	1	1	1

$$Y_0 = D_3 + \overline{D}_2 D_1$$

Implementation

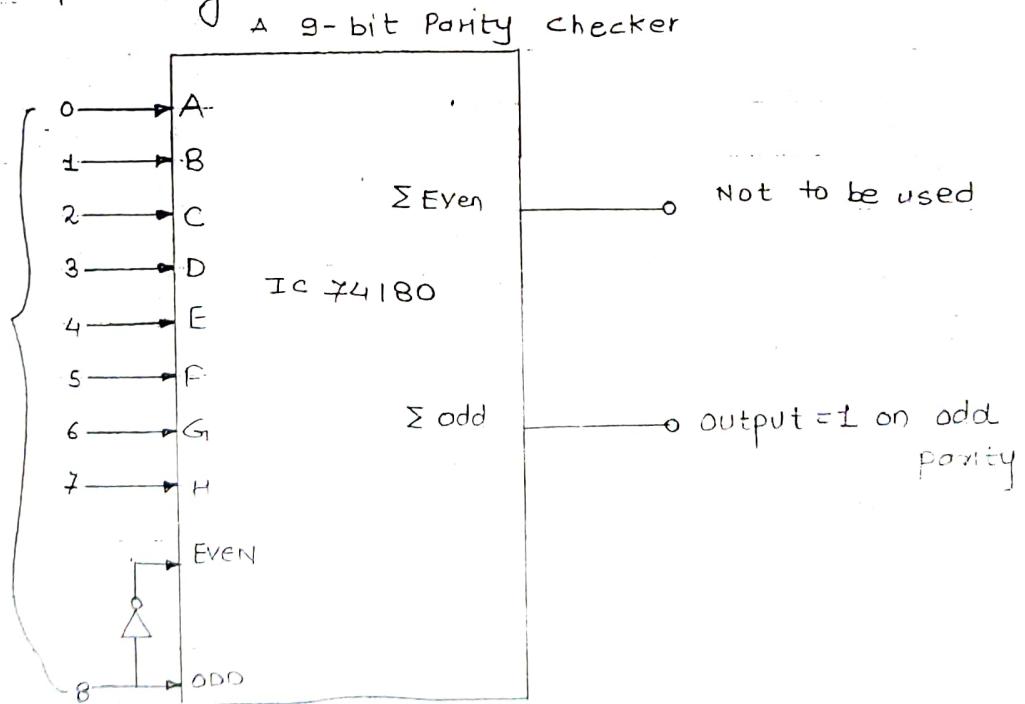


Q] Difference between CPLD and FPGA

CPLD	FPGA
① CPLD contains only a few blocks of logic (upto few thousands).	① FPGA contains upto 1,00,000 of tiny logic blocks.
② CPLDs are made of larger logic blocks.	② FPGAs are made of tiny logic blocks.
③ Delays are more predictable in CPLDs.	③ Delays are less predictable in FPGAs.
④ CPLD contains PAL like blocks , I/O blocks and interconnection wires.	④ FPGA does not contain the AND/OR matrices. Instead, logic blocks are provided for implementing the logic blocks circuits.
⑤ CPLDs are used for simpler applications.	⑤ FPGAs are used for complex applications.
⑥ CPLDs are cheap.	⑥ FPGAs are expensive.
⑦ Eg. XC 9500 family	⑦ Eg. Xilinx 4000 family.

Implement 9-bit odd parity checker circuit using IC = 74180.

- IC 74180 has 8-parity inputs A to H and two cascading inputs.
- There are two outputs Σ even and Σ odd.
- And there are two cascading inputs named EVEN and ODD.
- since the odd parity checker is to be drawn, we have to use the Σ odd output of 74180.
- since the 9-bit input is to be applied, we should apply 8 of the 9-bits to the input A-H, whereas the 9th bit is applied to the ODD input as shown in the fig below.
- Output Σ odd will be high if the parity of the 9-bit input is odd.
- The function table of IC 74180 explains the operation of the following circuit.

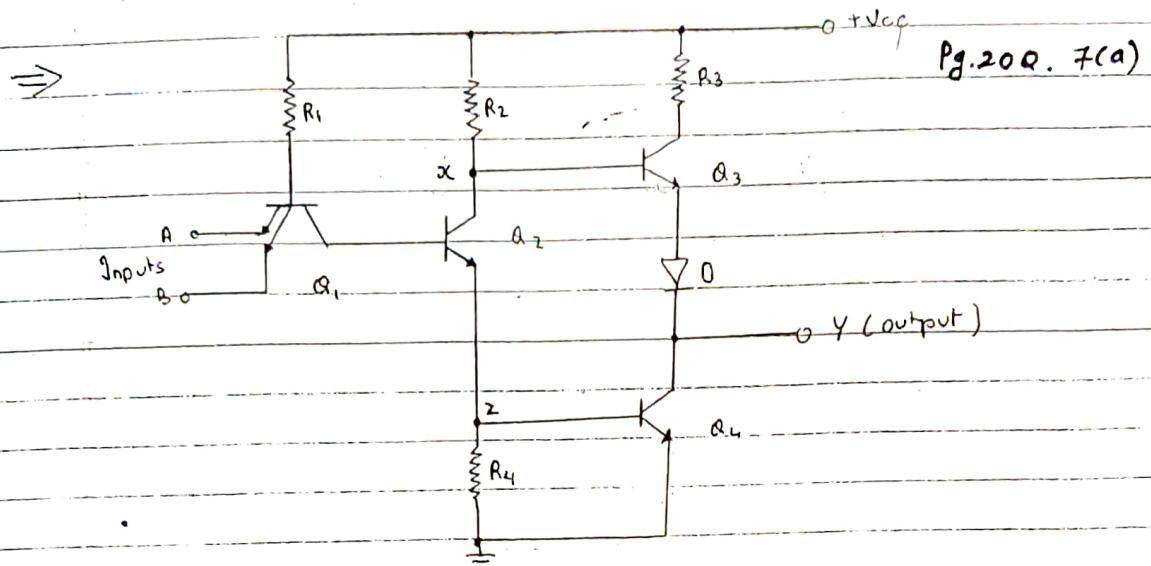


Function table of IC 74180 as parity checker

Parity inputs A to H	Cascading inputs		Parity of A to H and Σ even	Parity of A to H and Σ odd
	EVEN	ODD		
Odd	1	0	Odd	Even
Even	1	0	Odd	Even
Odd	0	1	Even	Odd
Even	0	1	Even	Odd

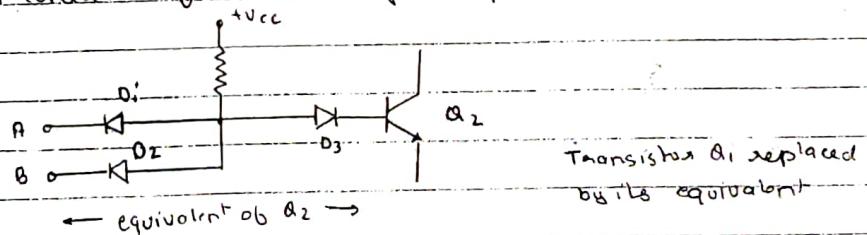
The cascading inputs should never be equal. The unused parity inputs must be connected to the ground.

7.a Draw a neat circuit diagram of 2 input TTL NAND gate and explain its operation.



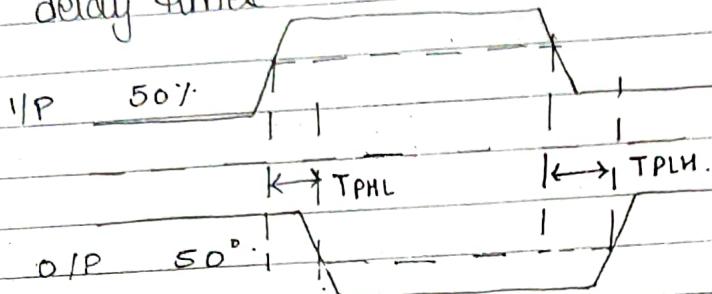
Two input TTL NAND gate

1. A and B are the input terminals. The input voltages A & B can be low or high.
2. A and B both low :- If A and B both are connected to ground, then both the B-E junctions of transistor Q1 are forward biased. Hence diodes D1 and D2 will conduct to force the voltage at point C to 0.7V.



This voltage is insufficient to forward bias base-emitter junction of Q2. Hence Q2 will remain OFF. Therefore its collector voltage V_x rises to V_{cc} . As transistor Q3 is operating in the emitter follower mode, output Y will be pulled up to high voltage. $\therefore Y = 1$ (High) $\text{If } A = B = 0 \text{ (Low)}$
The equivalent circuit for this input condition is as follows.

- Ques 4. SPEED: It is propagation delay time.
 In the fig. of I/P and O/P waveforms of logic gate the delay times are measured between the 50% voltage levels of I/P and O/P waveforms. 2 delay times:
 T_{PHL} - O/P goes from high to low state.
 T_{PLH} - O/P goes from low to high state.
 propagation delay time of a logic gate is average of the two delay times.



5. NOISE MARGIN: Stray electric & magnetic field can induce voltages on connecting wires between logic circuits. These unwanted spurious signals called as noise can sometimes cause the voltage at I/P to logic circuit to drop below V_{IL(min)} or rise above V_{IL(max)}. This produces unpredictable operations. Noise margin of a logic circuit refers to the ability to tolerate noise without causing spurious changes in O/P voltage.

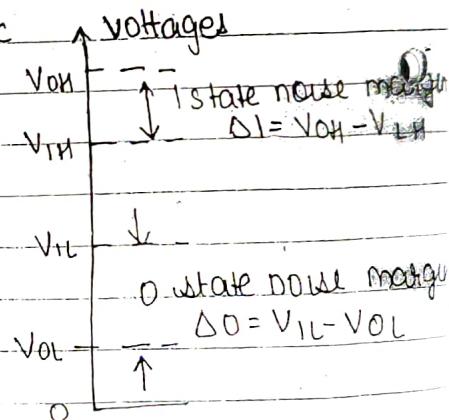
Range of voltages that occur at a logic circuit O/P.

Any voltage $>$ than V_{OH(min)} \rightarrow logic 1
 < than V_{OL(max)} \rightarrow logic 0.

under normal conditions

logic circuit responds to any I/P $>$ than V_{IL(min)} as logic 1.

Voltage $<$ than V_{IL(max)} as logic 0.



g. No. 18
 i. c) Convert following decimal number to Binary, Octal, Hexa-equivalent and gray code.

i) $(306.8)_{10}$

2	306	
2	153	0↑
2	76	1
2	38	0
2	19	0
2	9	1
2	4	1
2	2	0
1		0

$$0.8 \times 2 = 1.6 \quad 1 \downarrow$$

$$0.6 \times 2 = 1.2 \quad 1 \downarrow$$

8	306	
8	38	2↑
8	4	6
8	0	4

$$0.8 \times 8 = 6.4$$

$$0.4 \times 8 = 3.2$$

$$0.2 \times 8 = 1.6$$

$$0.6 \times 8 = 4.8 \downarrow$$

$$(306.8)_{10} = (462.6314)_8$$

$$(306.8)_{10} = (100110010.11)_2$$

$$(306.8)_{10} = (?) \text{gray code}$$

$$\text{Ans} - 0010\ 0000\ 0101.1100$$

$$(306.8)_{10} = (?)_{16}$$

16	306	
16	19	2↑
16	1	3
16		1
0.8 \times 16 = 12.8		\downarrow

$$\text{Ans} - (132.12)_{16}$$

ii) $(147.8)_{10}$

2	147	
2	73	1↑
2	36	1
2	18	0
2	9	0
2	4	1
2	2	0
2	1	0
1		1

$$0.8 \times 2 = 1.6 \quad 1 \downarrow$$

$$0.6 \times 2 = 1.2 \downarrow$$

$$8 \mid 147$$

8	18	3↑
8	2	2
8		2

$$0.8 \times 8 = 6.4 \quad 6$$

$$0.4 \times 8 = 3.2 \quad 3$$

$$0.2 \times 8 = 1.6 \quad 1$$

$$0.6 \times 8 = 4.8 \quad 4 \downarrow$$

$$(147.8)_{10} = (223.6314)_8$$

$$(147.8)_{10} = (10010011.11)_2$$

$(147.8)_{16} = (?)$ gray code
 $\rightarrow 000101100100 \cdot 1100$

$$(147.8)_{16} = (?)_{16}$$

	147	
	9	3↑
	9	

$$0.8 \times 16 = 12.8$$

Ans - $(93.12)_{16}$

d) Add $(57)_{10}$ and $(26)_{10}$ in BCD.

$$\begin{array}{r}
 (57)_{10} & 0101 & 0111 \\
 + (26)_{10} & + 0010 & 0110 \\
 & \hline
 & 0111 & 1101 \\
 & + 0000 & 0110 \\
 & \hline
 & 1000 & 0011
 \end{array}$$

Self Study

Pg No. (23)

Page No :- 14.

Q.4.b. Compare EEPROM and flash memory .



EEPROM (Electrically erasable programmable Read only memory).	Flash memory.
i. EEPROM is read only memory.	flash memory is type of EEPROM memory
ii. EEPROM uses NOR gates.	flash memory uses NAND Gate
iii. It gives faster response but it is expensive.	It gives slower response but less expensive.
iv. It is byte wise erasable.	It is block wise erasable.
v. EEPROM are seldom rewritten.	Flash is constantly rewritten.
vi. EEPROM is used when only small amounts are needed.	Flash is used when large amounts are needed.
vii. EEPROM works slowly,in EEPROM 1 byte write at a time.	flash memory works much faster than traditional EEPROM because it writes data in chunks . usually 512 bytes in size

Page. 17 Q. 4(cd)

I) Convert following into Binary, octal and hexadecimal:

(i) $(555)_{10}$

(ii) $(7905)_{10}$

(i) $(555)_{10} = (001000101011)_2$

2	555	
2	277	1
2	138	1
2	69	0
2	34	1
2	17	0
2	8	1
2	4	0
2	2	0
2	1	0
	0	1

Octal Conversion

0	0	1	0	0	0	1	0	1	0	1	0	1
		1			0		5		3			

$(555)_{10} = (1053)_8$

Hexadecimal
Conversion

0	0	1	0	0	1	0	1	0	1	1	B
		2		2		2		2			

$(555)_{10} = (22B)_{16}$

(ii) $(7905)_{10} = (0001\ 0000\ 1110\ 1111)_2$

2	7905	
2	3952	1
2	1976	0
2	988	0
2	494	0
2	247	0
2	123	1
2	61	1
2	30	1
2	15	0
2	7	1
2	3	1
2	1	1
	0	1

Hexadecimal Conversion

0	0	0	1	1	1	0	1	1	0	0	0	1
			1	7	3	4						

0	0	0	1	1	1	0	1	1	1	0	0	0
			1	E	E							

* SE-DLDA-3/6/2014.

Q. 1

q) Perform following without converting into other bases.

i) $(57)_8 + (24)_8$

Method 1:

$$\begin{array}{r}
 & 1 & 1 & \\
 \times & & & 8 \times 8 \\
 \hline
 & 2 & 3 & \\
 & 5 & 7 & \\
 \hline
 & 2 & 4 & \\
 & 1 & 2 & 7 & 4 \\
 \hline
 & 1 & 3 & 6 & 0 \\
 \hline
 & 1 & 6 & 5 & 4
 \end{array}$$

Steps

$7 \times 4 = 28$

$28 - 24 = 4$

$4 \times 5 = 20 + 3 = 23$

$23 - 16 = 7$

$2 \times 7 = 14 - 8 = 6$

$2 \times 5 = 10 + 1 = 11$

$11 - 8 = 3$

Method 2:

$$\begin{array}{r}
 57 \\
 \times 24 \\
 \hline
 20 \quad 28 \\
 \hline
 + 10 \quad 14 \quad x \\
 \hline
 10 \quad 34 \quad 28 \\
 - 8 \quad - 32 \quad - 24 \\
 \hline
 1 \quad 6 \quad 5 \quad 4
 \end{array}$$

i) $(3 \cdot 1 \cdot 2 \cdot 0)_4 + (2 \cdot 1 \cdot 3 \cdot 2)_4$

0	0
1	1
2	2
3	3
4	10
5	11
6	12
7	13

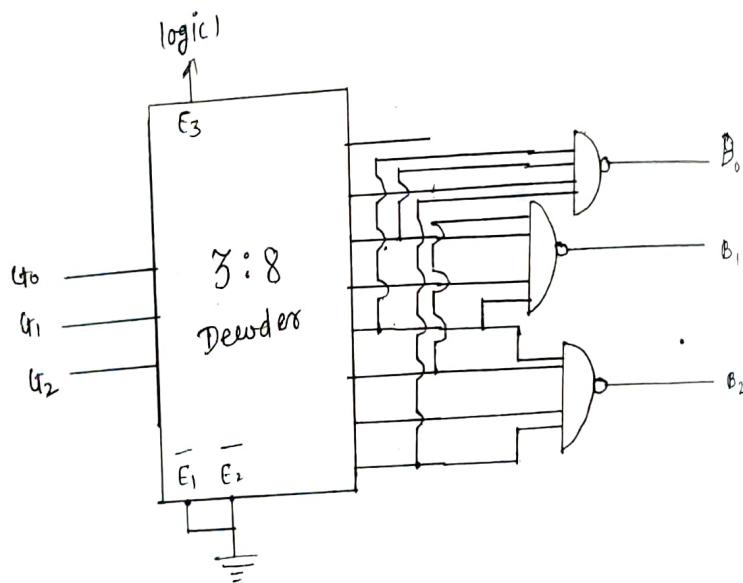
$(1 \cdot 1 \cdot 3 \cdot 1 \cdot 2)_4$

Pg No. 5
Q. 2B

State truth table of 3 bit Gray to Binary conversion then design it using 3:8 decoder and additional gates.

Pg. No. (26)

G_2	G_1	G_0	B_2	B_1	B_0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	1
1	0	1	1	1	0
1	1	0	1	0	0
1	1	1	1	0	1



ROLL NO.
24 / 29

Page No. 14

Q. 6 (c) Convert (118_{10}) into

- (i) BCD
- (ii) Hexadecimal
- (iii) Octal

Ans.

(i) BCD :

$$118_{10} = 0001 \ 0001 \ 1000$$

(ii) Hexadecimal :

$$118_{10} :$$

16	118	
16	007	6
	000	7

↑

$$\therefore 118_{10} = 76_{16}$$

(iii) Octal :

$$118_{10}$$

8	118	
8	014	6
8	001	6
	000	1

↑

$$\therefore 118_{10} = 166_8$$

Pg-15
Ques-2(a)

Pg. No. 28

i) $(63)_8 * (21)_8$

$$\Rightarrow (63)_8 * ((7)_8 + (7)_8 + (3)_8) \quad \text{since } (21)_8 = 7_8 + 7_8 + 3_8$$

$$\begin{array}{r} (63)_8 \\ \times (7)_8 \\ \hline (545)_8 \end{array} \quad \begin{array}{r} (63)_8 \\ \times (3)_8 \\ \hline (231)_8 \end{array}$$

$$\Rightarrow (63)_8 * (21)_8 = (545)_8 + (545)_8 + (231)_8$$

$$\begin{array}{r} (545)_8 \\ + (545)_8 \\ \hline (1312)_8 \end{array} \quad \begin{array}{r} (13127)_8 \\ + (231)_8 \\ \hline \underline{\underline{(1543)}_8} \end{array}$$

$$\therefore (63)_8 * (21)_8 = \underline{\underline{(1543)}_8}$$

ii) $(D9)_H - (80)_H$

$$\begin{array}{r} (D9)_H \\ - (80)_H \\ \hline \underline{\underline{(59)}_H} \end{array} \quad \text{since } D=13$$

Pg. 15.

Ques 2-a

$$\text{i) } (63)_8 * (21)_8$$

$$(21)_8 = (7_8 + 7_8 + 3_8)$$

$$\Rightarrow (63)_8 * (21)_8 = (63)_8 * (7_8 + 7_8 + 3_8)$$

$$\begin{array}{r} (63)_8 \\ * \quad 7_8 \\ \hline (545)_8 \end{array} \quad \text{also} \quad \begin{array}{r} (63)_8 \\ * \quad 3_8 \\ \hline (231)_8 \end{array}$$

$$\therefore (63)_8 * (21)_8 = (545)_8 + (545)_8 + (231)_8$$

$$\begin{aligned} &= \begin{array}{r} (545)_8 \\ + (545)_8 \\ \hline 1312 \end{array} \\ &\quad + \begin{array}{r} (231)_8 \\ \hline 1543 \end{array} \end{aligned}$$

Pg 9

Q[1](a)

Given number = $(41.62)_8$

(i) For Octal to Binary: Convert each digit into its equivalent binary number.

$$\therefore 4 = 100, 1 = 001, 6 = 110, 2 = 010$$

$$\therefore (41.62)_8 = (100001.110010)_2$$

(ii) For Octal to Hexadecimal: Using binary equivalent, making groups of 4 digits.

$$\therefore (\underline{0010} \underline{0001}. \underline{1100} \underline{1000})_2 = (21.C8)_{16}$$

(iii) For Octal to Decimal: Convert to decimal using binary.

$$\therefore (100001.110010)_2 = (1 \times 2^5 + 1 \times 2^0) + (1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-5}) \\ = (33.\underline{\underline{50}})_{10}$$

Q[2](c)

$$\begin{aligned} \text{Let } Y &= AB + B + \overline{AC} + A\overline{BC}(AB + C) \\ &= B(1+A) + \overline{AC} + AC(A \oplus B) + A\overline{BC}C \\ &= B(1) + \overline{AC} + 0 + (AC)\overline{B}C \\ &= B + \overline{AC} + \overline{B}C \\ &= B + C + (\overline{A} + \overline{C}) \\ &= B + \overline{A} + 1 \\ &= 1 \\ \therefore Y &= 1 \end{aligned}$$

pq-21

Q. 5) a) convert $(0.42)_{10}$ into binary. \rightarrow For integer, $0 \div 2 = 0$ with remainder of 0 \uparrow

$$\therefore (0)_{10} = (0)_2$$

 \rightarrow For fraction, $0.42 \times 2 = 0.84$ with a carry of 0 $0.84 \times 2 = 1.68$ with a carry of 1 $0.68 \times 2 = 1.36$ with a carry of 1 $0.36 \times 2 = 0.72$ with a carry of 0 $0.72 \times 2 = 1.44$ with a carry of 1 $0.44 \times 2 = 0.88$ with a carry of 1

$$\therefore (0.42)_{10} = (0.011011)_2$$

$$\rightarrow \boxed{(0.42)_{10} = (0.011011)_2}$$

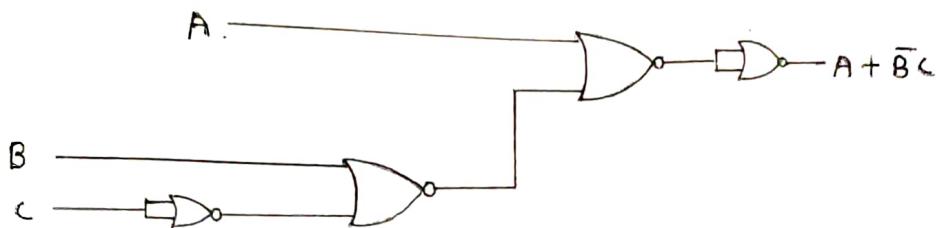
b) solve $111110.1 \div 0101$ $\rightarrow 111110.1 \div 101$

$$\begin{array}{r}
 \underline{1100.1} \\
 101) \underline{111110.1} \\
 - 101 \\
 \hline
 101 \\
 - 101 \\
 \hline
 01 \\
 - 0 \\
 \hline
 10 \\
 - 10 \\
 \hline
 0 \\
 - 0 \\
 \hline
 101 \\
 - 101 \\
 \hline
 0
 \end{array}$$

$$\boxed{111110.1 \div 101 = 1100.1}$$

Q. Implement $Y = A + \bar{B}C$ using only NOR gates.

Soln :



Q. Write $(32)_{10}$ into its BCD code and Ex-3 code

Soln :

$$(32)_{10} = (00110010)_{BCD}$$

For Ex-3 code

$$(32)_{10} + (3)_{10} = (35)_{10}$$

$$(35)_{10} = (100011)_{EX-3}$$

(Q.1)
(f) Convert $(126)_{10}$ to octal, Hex code or Hexadecimal
 $(126)_{10}$ to octal (divide by 8)

$$\begin{array}{l} 126/8 = 15 \text{ remainder } 6 \\ 15/8 = 1 \text{ remainder } 7 \\ 1/8 = 0 \text{ remainder } 1 \end{array}$$

$$\therefore (126)_{10} = 176 \text{ octal}$$

$(126)_{10}$ to Hexcode (divide by 16)

$$\begin{array}{l} 126/16 = 7 \text{ remainder } 14(E) \\ 7/16 = 0 \text{ remainder } 7 \end{array}$$

$$\therefore (126)_{10} = 7E \text{ hexcode or hexadecimal}$$

h) convert $(214.32)_{10}$ to binary

$(214.32)_{10}$ can be considered as $(214)_{10}$

so now $(214)_{10}$ to binary

$$(214)_{10} = \begin{array}{r} 01000001 \\ 2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0 \end{array}$$

$$\therefore (214)_{10} = 01000001$$

$$(214.32)_{10} = (01000001)_2$$

(i) Perform binary subtraction using's 2's complement
 for $(62)_{10}$ and $(99)_{10}$
 i.e. $(62)_{10} - (99)_{10} = -(37)_{10}$

$$(62)_{10} = (0011\ 1110)_2$$

$$(99)_{10} = (0110\ 1001)_2$$

Inverting $(99)_{10}$ i.e. $1001\ 1100$

Now adding 1

$$\begin{array}{r} 1001\ 1100 \\ + \quad 1 \\ \hline 1001\ 1101 \end{array} \quad \leftarrow \text{(Two's complement)}$$

Now

$$\begin{array}{r} 0011\ 1110 \\ + 1001\ 1101 \\ \hline 1101\ 1011 \end{array} \quad \begin{array}{l} 62 \\ + (-99) \end{array}$$

$$\text{inverting again} \rightarrow (0010\ 0100)_2$$

adding 1 in \swarrow , we get $(0010\ 0101)_2$
 (Two's complement)

$$(0010\ 0101)_2 = (-37)_{10}$$

i] Convert $(121.2)_2$ into base 10. Page. 6 Q. 1(a)(b), (1)d (35)

$$\text{Soln:} \quad \begin{array}{r} 2 \times 3^1 \\ \downarrow \\ 121 \cdot 2 \\ \downarrow \\ 1 \times 3^0 \end{array} = (16.66)_{10}$$

ii] Represent $(52)_{10}$ into Excess-3 and Gray code

To Excess-3 code

$$\begin{array}{r} 52 \\ + 33 \\ \hline 85 \\ \downarrow \quad \downarrow \\ 1000 \quad 0101 \end{array} \quad \therefore (52)_{10} = (1000 \oplus 101)_{\text{Ex-3}}$$

iii] To Gray code

$$(52)_{10} = (110100)_2$$

$$\begin{array}{r} 10100 \\ 110100 \\ \hline \end{array}$$

$$\boxed{(101110) \text{ Gray code}}$$

iv] Find the one's & two's complement of $(57)_{10}$

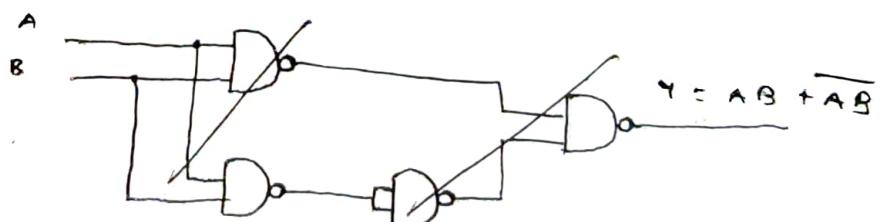
$$\text{Soln: } \therefore (57)_{10} = (111001)_2$$

$$1^{\text{'}} \text{ comp} = 000110$$

$$2^{\text{'}} \text{ comp} \quad \begin{array}{r} + 1 \\ \hline 000111 \end{array}$$

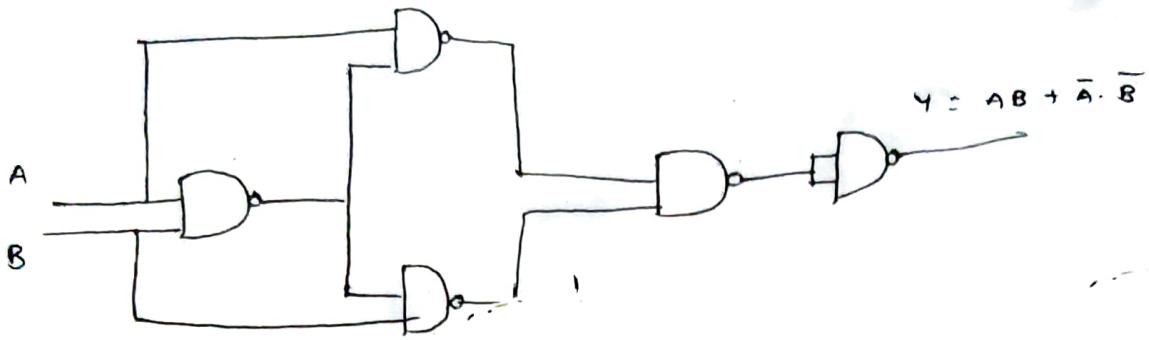
v] Realise $y = AB + \overline{AB}$ using NAND Gates only

$$\begin{aligned} \text{Soln: } y &= \overline{AB} + \overline{\overline{AB}} \\ &= \overline{AB} - \overline{AB} \end{aligned}$$



(36)

4



2a. Perform following operation using 2's complement

$$(i) (28)_{10} - (42)_{10} \quad (ii) (52)_{10} - (-18)_{10}$$

$$(42)_{10} = (101010)_2$$

$$(28)_{10} = (011100)_2$$

decimal \rightarrow binary \rightarrow 1's complement \rightarrow 2's complement

$$(42)_{10} \rightarrow (101010)_2 \rightarrow 010101$$

$$\begin{array}{r} 010101 \\ + 010101 \\ \hline 010110 \end{array}$$

$$(28)_{10} - (42)_{10}$$

$$\begin{array}{r} 010110 \\ + 010101 \\ \hline 01110010 \end{array}$$

(\because subtraction using 2's complement becomes addition)

$$\begin{array}{r} 110010 \\ - 1 \\ \hline 110001 \end{array}$$

$$- (001110)_2 = (-14)_{10}$$

$$\therefore (28)_{10} - (42)_{10} = (-14)_{10}$$

$$ii). (52)_{10} - (-18)_{10}$$

1. decimal \rightarrow binary \rightarrow 1's complement \rightarrow 2's complement

$$(52)_{10} \rightarrow 00110100 \rightarrow 11001011 \rightarrow 11001100$$

$$(18)_{10} \rightarrow 00010010 \rightarrow 11101101 \rightarrow 11101110$$

2. Addition of 2's complement of $(52)_{10}$ and $(18)_{10}$

$$\begin{array}{r} 00010010 \\ + 11101101 \\ \hline 11110110 \end{array} \quad \text{2's complement}$$

Converting into 1's complement

$$1's \text{ complement } 10111001$$

$$\begin{array}{r} 11110110 \\ + 10111001 \\ \hline 110111010 \end{array} \quad \text{Converting into binary}$$

$$(01000110)_2 = (70)_{10}$$

$$\text{Hence } (52)_{10} - (-18)_{10} = (70)_{10}$$

Q.1 (a) Represent $(29)_{10}$ in Excess-3 code & Gray code

$$(29)_{10} \rightarrow \text{given}$$

1) Excess-3 code $(29)_{10} + (3)_{10}$

$$= (32)_{10}$$

$$= (100000)_2 \times s-3.$$

2) Gray code:

$$(29)_{10}$$

First we convert decimal $(29)_{10}$ into binary.

$$\begin{array}{r} 2 | 29 \\ 2 | 14 \quad 1 \\ 2 | 7 \quad 0 \\ 2 | 3 \quad 1 \\ 2 | 1 \quad 1 \\ 0 \end{array}$$

binary number is $(11101)_2$.

For converting to Gray

$$\begin{array}{r} 1 \oplus 1 \quad 1 \oplus 0 \quad 0 \oplus 1 \\ \downarrow \quad \downarrow \quad \downarrow \\ 1 \quad 0 \quad 0 \quad 1 \quad 1 \end{array}$$

So the Gray number is 10011

Q.1(b) $(67.4A)_{16}$ into equivalent octal no.

0010 0111 0100 10100 ← equivalent 4 bit no
 1 4 7 2 2 4 ← grouping into 3 4 8 digits

147.224 ← writing binary of each

(89)

Q.1(c) $(215.32)_{10}$ into base '7'

$$\begin{array}{r} 215 \\ \hline 7 | 30 \\ \quad 7 | 4 \\ \quad \quad 0 \end{array} \qquad \begin{array}{r} 5 \\ 2 \\ 4 \end{array} \uparrow$$

425.0

For dec. part

$$0.32 \times 7 = 2\overset{1}{.}4 \qquad 0.2145$$

$$0.24 \times 7 = 1\overset{1}{.}6\overset{1}{.}8$$

$$0.68 \times 7 = 4\overset{1}{.}7\overset{1}{.}6$$

$$0.76 \times 7 = 5\overset{1}{.}32$$

∴ Eq. base 7 is 425.2145

(d) $(670.17)_8$ into binary & hexadecimal

$(110111000.00111)_2$ ← writing 3 digit bin. group

000110111000.0011100 ← group of 4
 1 B 8 3 C ← add zeroes
 For hexa: ← at start & end
 & add zeroes
 to make a valid 4
 bit number

1B8.3C

Q1] e] Add $(57)_{10}$ and $(26)_{10}$ in BCD

$$\begin{array}{r}
 \text{Sdn.} & \begin{array}{r} 0101 & 0111 \\ + 0010 & 0110 \\ \hline 0111 & 1101 \end{array} & \begin{array}{l} (57) \\ + (26) \\ \hline \end{array} \\
 & \begin{array}{r} + 0110 \\ \hline 1000 & 0011 \end{array} &
 \end{array}$$

\leftarrow We add correction factor of six
 (83)

f] Explain uses of Gray Code.

Ans. The uses of Gray Code are as follows :

1. When analog into is converted to digital or vice versa there are various ambiguities due to which the value changes. Gray Code helps in correcting such errors.
2. Digital logic designers use Gray Code extensively for passing multi bit count info between asynchronous logic that operates at different clock frequency. The logic is considered operating in different "clock domains".
3. A perfect real world application of Gray Codes is in stadiometer. Its a device which is used in medical settings to measure person's height.

g] Add $(DDCC)_{16}$ and $(BBAA)_{16}$

Sdn.

$$\begin{array}{r}
 1101 & 1101 & 1100 & 1100 \\
 + 1011 & 1011 & 1010 & 1010 \\
 \hline
 0001 & 1001 & 1001 & 0111 & 0110
 \end{array}$$

A	$10 \rightarrow 1010$
B	$11 \rightarrow 1011$
C	$12 \rightarrow 1100$
D	$13 \rightarrow 1101$
E	$14 \rightarrow$
F	$15 \rightarrow$

Ans. $(19976)_{16}$

Pg No 8) Q. 1 (d)
Convert following Octal numbers to binary, decimal and Hexadecimal Number.

1) $(6673)_8$

\Rightarrow (Writing 3 bit binary for all the digits)

Binary $\Rightarrow (110\ 110\ 111\ 011)_2$

Decimal $\Rightarrow 2^{11} + 2^{10} + 2^8 + 2^7 + 2^5 + 2^4 + 2^3 + 2^1 + 2^0 = \underline{\underline{3515}}$

Hexadecimal \Rightarrow (Grouping 4 bit binary number and writing Hexadecimal Equivalent).

110110111011

$\Rightarrow (\underline{\underline{D}\ B\ B})_{16}$

2) $(7466)_8$

\Rightarrow (Writing 3 Bit binary for all digits)

Binary: $(111\ 100110110)_2$

$2^{11} + 2^{10} + 2^9 + 2^8 + 2^5 + 2^4 + 2^2 + 2 = \underline{\underline{3894}}$

Decimal: $\underline{\underline{3894}}$

(Grouping 4 bits of binary to get Hexadecimal Number)

Hexadecimal: $(\underline{\underline{F36}})_{16}$

Short Note on Gray Code and Excess-3 Code.

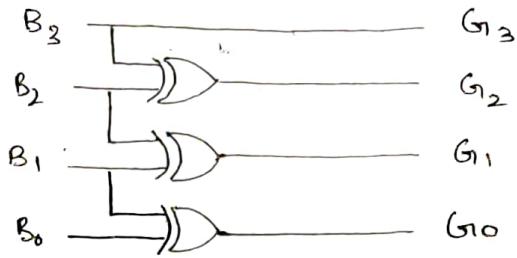
(43)

① Gray Code:

- It is an important code that is used to convert a decimal number into 8-bit binary sequence.
- This conversion is carried out in a manner that 2 consecutive binary sequences at consecutive binary sequence differ by 1 bit only.

→ eg:

Binary:	1	1	0	1
Gray:	1	0	1	1



- The MSB of Binary code is same as the MSB of Gray code.
- The bit after MSB is obtained by taking XOR of the first and second bit of binary code and so on.

② Excess-3 code

- The function of XS-3 code is to transform the decimal numbers into their 4-bit BCD code.
- In this code, the decimal number is transformed to the 4-bit BCD code by first adding '3' to all the digits of the number and then converting the excess digits into their corresponding 8421 BCD code.
- eg: converting the decimal no. 85 to XS-3 BCD code:
soln: Adding 3 to each digit: $8+3=11$
 $5+3=8$

$$\begin{aligned} \text{ReArrange } 11 &: 1011 \\ 8 &: 1000 \end{aligned}$$

Therefore, the XS-3 BCD representation of the decimal number 85 is 1001 1000.

Q.1.)

(b)

$$1) \quad (F8F)_{16} + (D49)_{16}$$

$$\begin{array}{rcl}
 (F8F)_{16} & = & 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 1 \\
 + (D49)_{16} & = & + 1\ 1\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 1 \\
 & & \hline
 & & 0\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 0
 \end{array}$$

↓ C D 8
 $= (\text{ECD8})_{16}$

$$2) \quad (762)_{BCD} + (238)_{BCD}$$

$$(762) = (0\ 1\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 1\ 0)_2$$

$$238 = (0\ 0\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0)_2$$

$$(762)_{BCD}$$

$$\begin{array}{rcl}
 + (238)_{BCD} & = & 0\ 1\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 1\ 0 \\
 & & + 0\ 0\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0 \\
 & & \hline
 & & 1\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0
 \end{array}$$

↓ Invalid BCD

[Add 6 as the correction factor]

$$\begin{array}{rcl}
 \text{(i)} \quad & 1\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0 \\
 & + \quad \quad \quad 0\ 1\ 1\ 0 \\
 & \hline
 & 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0
 \end{array}$$

↑ Invalid BCD

$$\begin{array}{rcl}
 \text{(ii)} \quad & 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 0\ 0 \\
 & + \quad \quad \quad 0\ 1\ 1\ 0 \\
 & \hline
 & 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0
 \end{array}$$

↑ Invalid BCD

$$\begin{array}{rcl}
 \text{(iii)} \quad & 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\
 & + \quad \quad \quad 0\ 1\ 1\ 0 \\
 & \hline
 & 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0
 \end{array}$$

↓ 0 0 0 0 0 0 0 0 0 0
 $= 1000$

(44)

$$(3) (246)_{10} - (435)_{10}$$

Binary equivalent of 246 = 11110110

Binary equivalent of 435 = 110110011

Let

$$A = 11110110, \quad B = 110110011$$

$$\bar{B} = 001001100$$

$$B' = \bar{B} + 1$$

$$= 001001101$$

$$A + B' = \begin{array}{r} 011110110 \\ + 001001101 \\ \hline 101000011 \end{array}$$

carry is
absent.

← (a)

The answer
is negative.

$$C = 101000011$$

$$\bar{C} = 010111100$$

$$C' = 010111101$$

thus decimal equivalent of C' is 189

\therefore no carry is present in step (a)

The answer has to be negative

$$\therefore (246)_{10} - (435)_{10} = -(189)_{10}$$

Pg. 22

Q1 a] Convert $(243)_5$ to equivalent base 8 and base 7 number.

Solⁿ: Converting $(243)_5$ to base 10 number.

$$\begin{aligned}(243)_5 &= 2 \times 5^2 + 4 \times 5^1 + 3 \times 5^0 \\ &= 2 \times 25 + 4 \times 5 + 3 = 73\end{aligned}$$

$$\therefore (243)_5 = (73)_{10}$$

i) Converting to base 8.

$$\begin{array}{rcl} 73 \div 8 &=& 9 \quad \text{remainder } 1 \\ 9 \div 8 &=& 1 \quad \text{remainder } 1 \\ 1 \div 8 &=& 0 \quad \text{remainder } 1 \end{array} \quad \uparrow$$

$$\Rightarrow \underline{\underline{(243)_5}} = \underline{\underline{(111)_8}} = \underline{\underline{(73)_{10}}}$$

ii) Converting to base 7

$$\begin{array}{rcl} 73 \div 7 &=& 10 \quad \text{remainder } 3 \\ 10 \div 7 &=& 1 \quad \text{remainder } 3 \\ 1 \div 7 &=& 0 \quad \text{remainder } 1 \end{array} \quad \uparrow$$

$$(243)_5 = (133)_7 = (73)_{10}$$

$$\Rightarrow \underline{\underline{(243)_5}} = \underline{\underline{(73)_{10}}} = \underline{\underline{(111)_8}} = \underline{\underline{(133)_7}}$$

(1)

Q. Differentiate between sequential and combinational circuits.

<u>Sequential circuits</u>	<u>Combinational circuits</u>
1) These are circuits whose output depends not only on present inputs but also on the past history of inputs.	1) These are circuits whose output is dependent only on the inputs at the same instant of time.
2) These contain memory elements.	2) These do not contain memory elements.
3) Their behavior is described by the set of next state (memory) functions and set of output functions.	3) Their behavior is described by set of output functions only.
4) These are basically combinational circuits with additional properties of storage (to remember the past inputs) and feedback.	4) Contain no feedback mechanism.
5) They make use of the clock.	5) They do not make use of clock.
6) Sequential circuits compute their output based on input state and the state gets updated based on every clock pulse.	6) They implement boolean functions so they are functions only of their inputs and are not dependent of clocks.
Examples: FlipFlops, counters.	Examples: Half adder, full adder Half subtractor, Full subtractor