

Digital Techniques (313303)

Model Question Paper — Set B · Model Answers

Q.1

a) $(93)_{10}$ to hexadecimal and octal.

2 M

Hexadecimal

Integer part (93): divide repeatedly by 16, write remainders from bottom to top.

Division	Quotient	Remainder
$93 \div 16$	5	D
$5 \div 16$	0	5

$$\therefore (93)_{10} = (5D)_{16}$$

Octal

Integer part (93): divide repeatedly by 8, write remainders from bottom to top.

Division	Quotient	Remainder
$93 \div 8$	11	5
$11 \div 8$	1	3
$1 \div 8$	0	1

$$\therefore (93)_{10} = (135)_8$$

(1 M each)

b) **Duality theorem.**

2 M

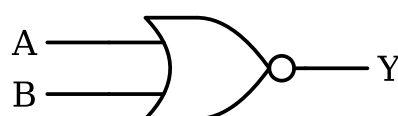
Duality theorem: The dual of any true Boolean identity is also true. The dual is obtained by interchanging AND ($\cdot$) and OR ($+$) operators and interchanging 0 and 1, keeping the variables (and their complements) unchanged.

Example: $A + 0 = A \leftrightarrow$ dual: $A \cdot 1 = A$; $A(B + C) = AB + AC \leftrightarrow$ dual: $A + BC = (A + B)(A + C)$.

(Statement 1 M, example 1 M)

c) **NOR gate: symbol and truth table.**

2 M



NOR gate symbol

A	B	$Y = \overline{A + B}$
0	0	1
0	1	0
1	0	0
1	1	0

Output is 1 only when all inputs are 0.

(Symbol 1 M, truth table 1 M)

d) Encoder and decoder.

2 M

Encoder: a combinational circuit that converts one active input out of 2^n input lines into an n-bit binary code. Application: keyboard encoder (decimal-to-BCD encoder), priority encoder in interrupt systems.

Decoder: a combinational circuit that converts an n-bit binary input code into one active output out of 2^n output lines. Application: memory address decoding, BCD to 7-segment display driver.

(Each function $\frac{1}{2}$ M, each application $\frac{1}{2}$ M)

e) Excitation tables of T and D flip-flops.

2 M

Q_n	Q_{n+1}	T	Q_n	Q_{n+1}	D
0	0	0	0	0	0
0	1	1	0	1	1
1	0	1	1	0	0
1	1	0	1	1	1

$$T = Q_n \oplus Q_{n+1}; \quad D = Q_{n+1}.$$

f) Modes of shift register.

2 M

1. Serial In Serial Out (SISO)
2. Serial In Parallel Out (SIPO)
3. Parallel In Serial Out (PISO)
4. Parallel In Parallel Out (PIPO)

(Also shift-left / shift-right and bidirectional modes in a universal shift register.)

($\frac{1}{2}$ M each)

g) De Morgan's theorems.

2 M

Theorem 1: The complement of a sum is equal to the product of the complements: $\overline{A + B} = \bar{A} \cdot \bar{B}$

Theorem 2: The complement of a product is equal to the sum of the complements: $\overline{A \cdot B} = \bar{A} + \bar{B}$

(1 M each)

Q.2

a) $(11010110)_2$ to decimal, hex, Gray and Excess-3.

4 M

i) Decimal

$$\begin{aligned}
 (11010110)_2 &= 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 \\
 &= 128 + 64 + 0 + 16 + 0 + 4 + 2 + 0 \\
 &= (214)_{10}
 \end{aligned}$$

ii) Hexadecimal

Make groups of 4 bits from the binary point:

4-bit group	1101	0110
Digit	D	6

$$\therefore (11010110)_2 = (D6)_{16}$$

iii) Gray code

Binary → Gray: MSB is copied as it is; every next Gray bit = XOR of the present and previous binary bits.

Binary	1	1	0	1	0	1	1	0
XOR step	(copy)	$1 \oplus 1$	$1 \oplus 0$	$0 \oplus 1$	$1 \oplus 0$	$0 \oplus 1$	$1 \oplus 1$	$1 \oplus 0$
Gray	1	0	1	1	1	1	0	1

$$\therefore (11010110)_2 = (10111101)_{\text{Gray}}$$

iv) Excess-3 code (of the decimal value 214)

Excess-3: add 3 (0011) to each decimal digit and write its 4-bit binary.

Decimal digit	2	1	4
Digit + 3	5	4	7
Excess-3	0101	0100	0111

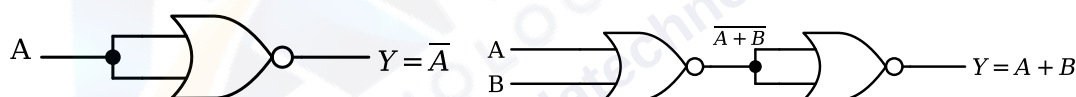
$$\therefore (214)_{10} = (0101 \ 0100 \ 0111)_{\text{XS-3}}$$

(1 M each)

b) Basic gates using only NOR gates.

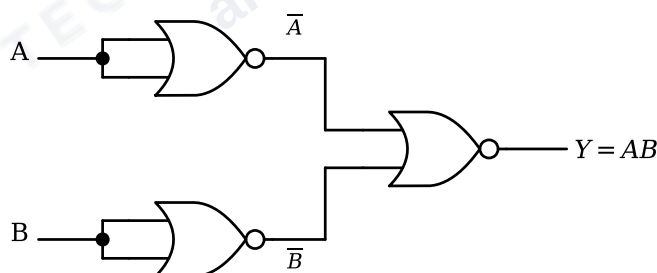
4 M

NOR is a **universal gate**: $Y = \overline{A + B}$.

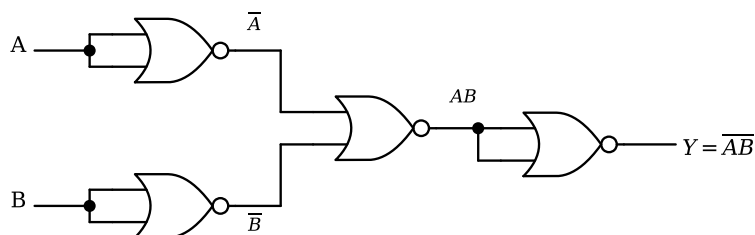


i) NOT: $Y = \text{complement of } (A + A)$

ii) OR: NOR followed by NOR-inverter



iii) AND: NOR of inverted inputs (De Morgan)



iv) NAND: AND followed by NOR-inverter

Expressions: NOT: $\overline{\overline{A}} = A$; OR: $\overline{\overline{A} + \overline{B}} = A + B$; AND: $\overline{\overline{A} \cdot \overline{B}} = A \cdot B$; NAND: $\overline{A \cdot B}$.

(Each gate 1 M)

c) Full adder using K-map.

4 M

A full adder adds three bits A, B and carry-in C_{in} (written C) and gives Sum S and Carry-out C_{out} .

A	B	C_{in}	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

BC \ A	00	01	11	10
0	0	1	0	1
1	1	0	1	0

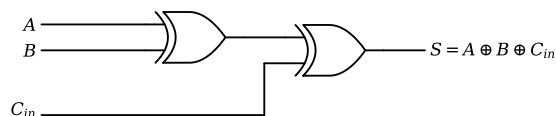
K-map for S (no pairs possible)

BC \ A	00	01	11	10
0	0	0	1	0
1	0	1	1	1

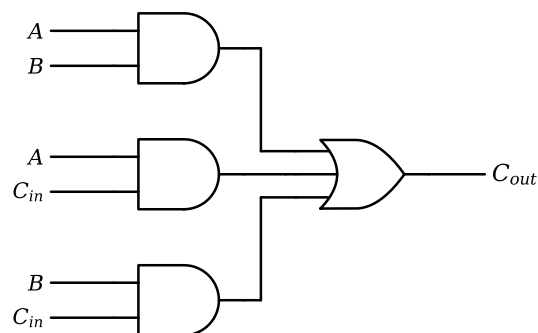
K-map for C_{out}

$$S = \overline{A}BC + A\overline{B}C + A\overline{B}\overline{C} + ABC = A \oplus B \oplus C_{in}$$

$$C_{out} = BC + AC + AB (= AB + AC_{in} + BC_{in})$$



Sum: two EX-OR gates

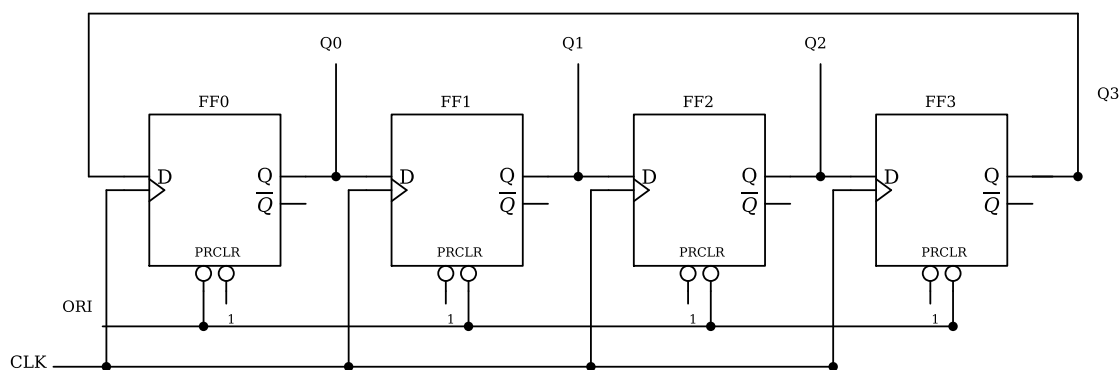


Carry: AND-OR

(Truth table 1 M, K-maps 1 M, logic diagram 2 M)

d) 4-bit ring counter using D flip-flops.

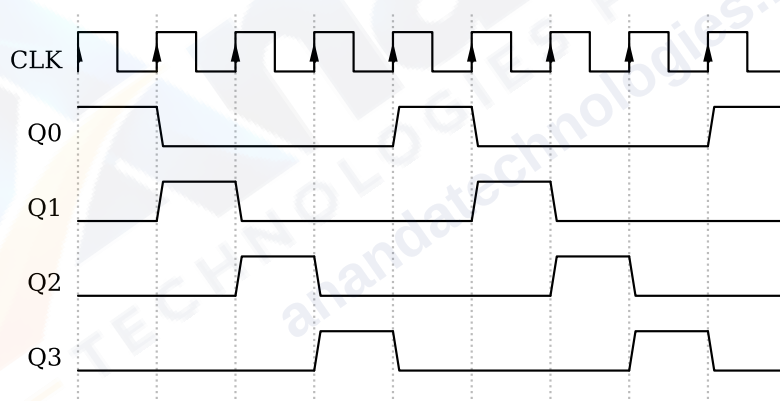
4 M



4-bit ring counter (Q3 fed back to D0); ORI presets FF0 and clears others

The output of the last flip-flop is fed back to the input of the first. Initially the ORI (overriding input) presets FF0 to 1 and clears the others, so the counter starts at 1000. With each clock pulse the single 1 circulates. A 4-bit ring counter has **4 states** (MOD-4).

Clock	Q0	Q1	Q2	Q3
Initial (ORI)	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
4	1	0	0	0



Waveforms of 4-bit ring counter

(Diagram 2 M, table 1 M, waveforms 1 M)

Q.3

a) K-map (POS): $F = \prod M(0, 1, 4, 5, 6, 10, 11, 14, 15)$.

4 M

Write 0 in the cells of the given maxterms and 1 in the remaining cells, then group the 0s.

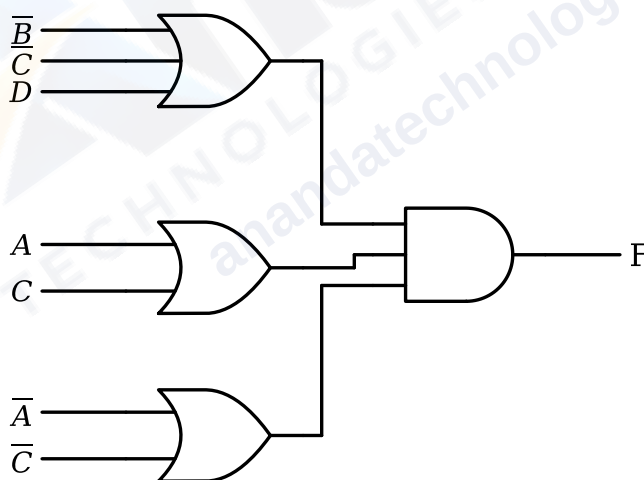
		CD			
AB		00	01	11	10
00		0	0	1	1
01		0	0	1	0
11		1	1	0	0
10		1	1	0	0

K-map for F (grouping 0s)

For POS, 0s are grouped; in each group a variable that is 0 is written as it is and a variable that is 1 is complemented (sum term).

Group	Cells (maxterms)	Sum term
Group 1	6, 14	$(\bar{B} + \bar{C} + D)$
Group 2	0, 1, 4, 5	$(A + C)$
Group 3	10, 11, 14, 15	$(\bar{A} + \bar{C})$

Minimal POS: $F = (\bar{B} + \bar{C} + D)(A + C)(\bar{A} + \bar{C})$



Logic diagram (OR-AND)

(K-map 1 M, grouping 1 M, expression 1 M, diagram 1 M)

b) BCD addition.

4 M

i) 2759 + 4386

Write each number in BCD and add digit-wise from the LSD. If a digit sum is greater than 9 (1001) or produces a carry, add 6 (0110) to correct it and pass the carry to the next digit.

$$(2759)_{10} = 0010\ 0111\ 0101\ 1001$$

$$(4386)_{10} = 0100\ 0011\ 1000\ 0110$$

Digit	BCD of 1st	BCD of 2nd	Carry in	Binary sum	Correction?	Result digit	Carry out
10^0	1001	0110	0	01111 (15)	Yes, +0110	0101	1
10^1	0101	1000	1	01110 (14)	Yes, +0110	0100	1
10^2	0111	0011	1	01011 (11)	Yes, +0110	0001	1
10^3	0010	0100	1	00111 (7)	No (≤ 9)	0111	0

$$\therefore 2759 + 4386 = 7145 = (0111\ 0001\ 0100\ 0101)_{\text{BCD}}$$

ii) **68 + 57**

Write each number in BCD and add digit-wise from the LSD. If a digit sum is greater than 9 (1001) or produces a carry, add 6 (0110) to correct it and pass the carry to the next digit.

$$(68)_{10} = 0110\ 1000$$

$$(57)_{10} = 0101\ 0111$$

Digit	BCD of 1st	BCD of 2nd	Carry in	Binary sum	Correction?	Result digit	Carry out
10^0	1000	0111	0	01111 (15)	Yes, +0110	0101	1
10^1	0110	0101	1	01100 (12)	Yes, +0110	0010	1

$$\therefore 68 + 57 = 125 = (0001\ 0010\ 0101)_{\text{BCD}}$$

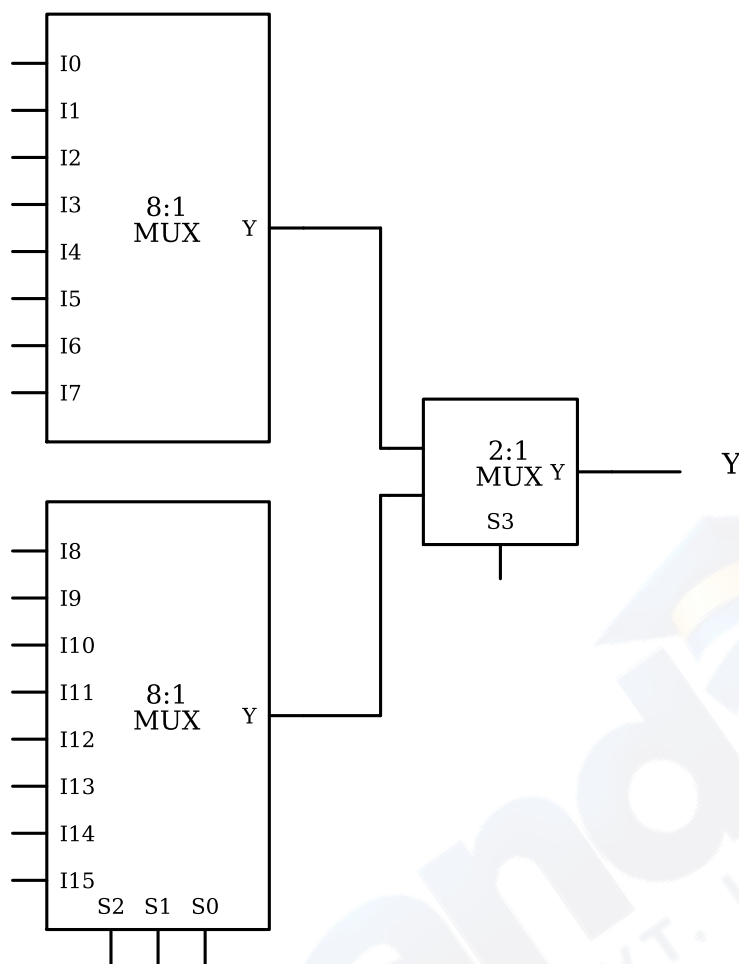
(2 M each)

c) **16 : 1 MUX using 8 : 1 MUX and 2 : 1 MUX.**

4 M

16 inputs need 4 select lines $S_3\ S_2\ S_1\ S_0$. Two 8:1 MUXes handle I_0 – I_7 and I_8 – I_{15} using the

common select lines S2 S1 S0. A 2:1 MUX selects one of their outputs using S3.



(S2, S1, S0 common to all)

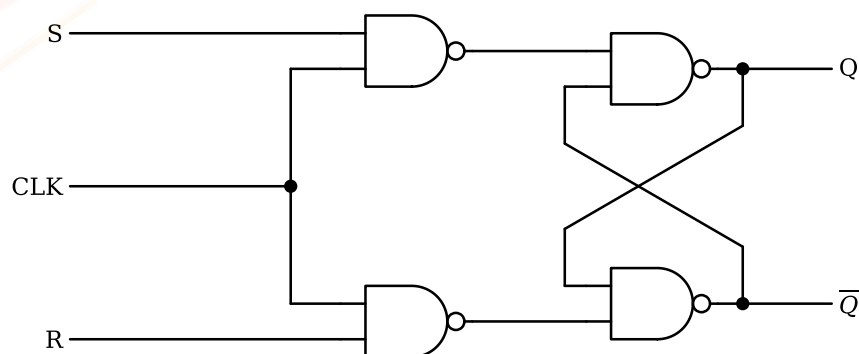
16:1 MUX using two 8:1 MUX and one 2:1 MUX

Selection of I_{12} : $12 = (1100)_2 \rightarrow S_3 S_2 S_1 S_0 = 1 1 0 0$. $S_2 S_1 S_0 = 100$ selects input 4 of each 8:1 MUX, i.e. I_4 (upper MUX) and I_{12} (lower MUX). $S_3 = 1$ makes the 2:1 MUX pass the lower MUX output $\rightarrow Y = I_{12}$.

(Diagram 3 M, explanation 1 M)

d) Clocked S-R flip-flop using NAND gates.

4 M



Clocked S-R flip-flop using NAND gates

Working: Gates 1 and 2 are steering gates and gates 3 and 4 form a NAND latch. When $CLK = 0$ the outputs of gates 1 and 2 are 1, so the latch keeps its previous state. When $CLK = 1$ the S and R

inputs reach the latch (in inverted form):

CLK	S	R	Q_{n+1}	State
0	X	X	Q_n	No change (disabled)
1	0	0	Q_n	No change
1	0	1	0	Reset
1	1	0	1	Set
1	1	1	?	Invalid (forbidden)

$S = R = 1$ makes both Q and $\bar{Q}$ equal to 1 while $CLK = 1$, which is not allowed.

(Diagram 2 M, working + truth table 2 M)

Q.4

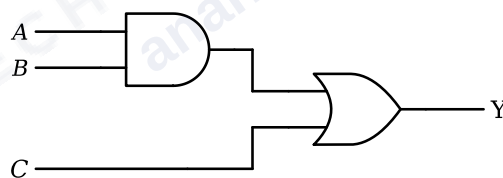
a) Simplification using Boolean laws.

4 M

Step	Law used
$Y = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC + \bar{A}B\bar{C}$	Given
$= \bar{A}C(\bar{B} + B) + AC(\bar{B} + B) + \bar{A}B\bar{C}$	Taking common (distributive law)
$= \bar{A}C \cdot 1 + AC \cdot 1 + \bar{A}B\bar{C}$	Complement law: $B' + B = 1$
$= \bar{A}C + AC + \bar{A}B\bar{C}$	Identity law: $X \cdot 1 = X$
$= C(\bar{A} + A) + \bar{A}B\bar{C}$	Distributive law
$= C + \bar{A}B\bar{C}$	$A' + A = 1, C \cdot 1 = C$
$= C + AB$	Absorption form: $X + X'Y = X + Y$ (with $X = C, Y = AB$)

$$Y = AB + C$$

Check by truth table: $Y = 1$ for minterms 1, 3, 5, 6, 7 in both the given and the simplified expression (verified).



Logic diagram of $Y = AB + C$

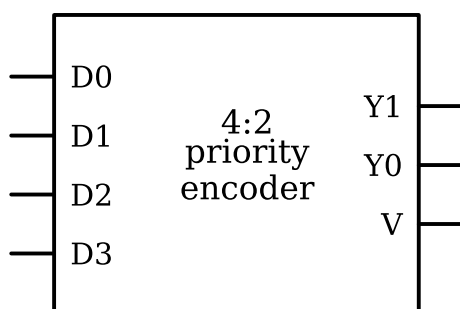
(Steps 3 M, logic diagram 1 M)

b) 4 : 2 priority encoder.

4 M

A priority encoder gives the binary code of the **highest-priority** active input when more than one input is 1. Here D3 has the highest and D0 the lowest priority. V (valid) = 1 when any input is

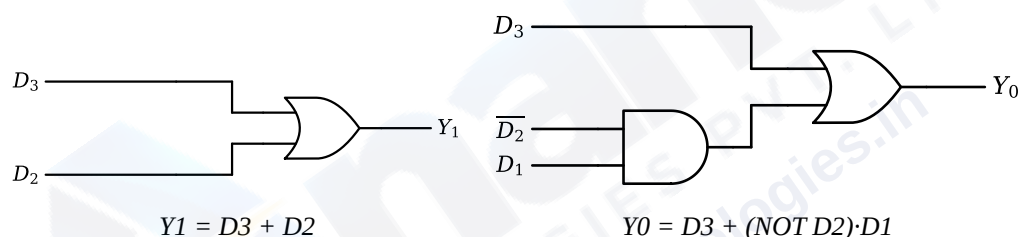
active.



Block diagram

D3	D2	D1	D0	Y1	Y0	V
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1

From the table: $Y1 = D3 + D2$, $Y0 = D3 + \overline{D2}D1$, $V = D0 + D1 + D2 + D3$ (checked for all 15 non-zero inputs).



(Block diagram 1 M, truth table 1 M, logic diagram 2 M)

c) R-2R ladder DAC numerical.

4 M

For a 4-bit R-2R ladder DAC (inverting type): $V_o = -(R_f/R) V_R (b_3/2 + b_2/4 + b_1/8 + b_0/16)$; with $R_f = R$, $V_o = -V_R \times D/16$ where D is the decimal value of the input.

i) 1101 (D = 13): $V_o = -10 \times (1/2 + 1/4 + 0 + 1/16) = -10 \times 0.8125 = -8.125 \text{ V}$

ii) 0110 (D = 6): $V_o = -10 \times (0 + 1/4 + 1/8 + 0) = -10 \times 0.375 = -3.75 \text{ V}$

Resolution = $V_R/2^n = 10/16 = 0.625 \text{ V per LSB}$ (or $1/2^4 = 1/16$ of full scale, i.e. 6.25 %).

(Formula 1 M, each output 1 M, resolution 1 M)

d) SRAM vs DRAM.

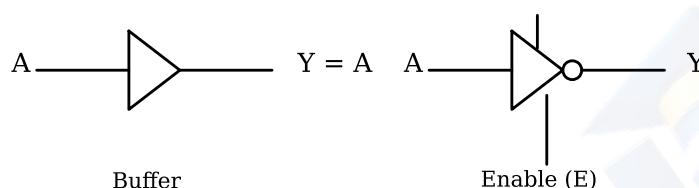
4 M

SRAM	DRAM
Bit stored in a flip-flop (latch) of about 6 transistors.	Bit stored as charge on a capacitor (1 transistor + 1 capacitor).
No refreshing needed.	Needs periodic refreshing (every few ms).
Faster, access time a few ns.	Slower than SRAM.
Low density, costly per bit.	High density, cheap per bit.
Used as cache memory.	Used as main memory.

(Any four points, 1 M each)

e) Buffer and tri-state buffer.

4 M



Buffer: a single-input gate whose output equals its input ($Y = A$). It does not change the logic level but provides more current (higher fan-out), isolation and delay. *Application:* driving LEDs/relays or many gate inputs; bus drivers (e.g. IC 74LS244).

A	Y
0	0
1	1

Tri-state buffer: a buffer with an extra enable input. When $E = 1$ the output follows the input; when $E = 0$ the output goes to the **high-impedance (Hi-Z)** state, i.e. it is electrically disconnected. So its output has three states: 0, 1 and Hi-Z. *Application:* connecting many devices to a common data bus in microprocessor systems (e.g. IC 74LS125/74LS245).

E	A	Y
0	X	Hi-Z
1	0	0
1	1	1

(Each with symbol, truth table and application 2 M)

Q.5

a) BCD to 7-segment decoder; segments c and f.

6 M

The BCD to 7-segment decoder accepts a 4-bit BCD digit (A B C D, A = MSB) and activates the proper segments a–g of a common-cathode display (segment ON = 1). Codes 1010–1111 never

occur, so they are don't cares.

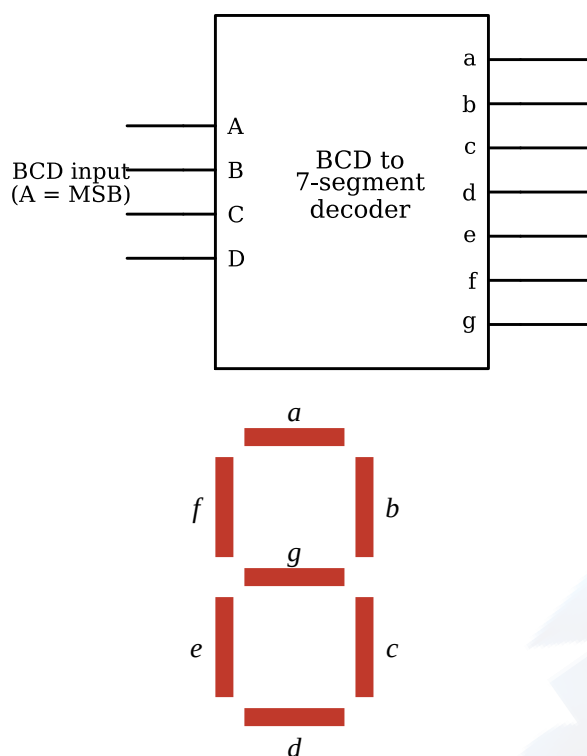


Fig.: BCD to 7-segment decoder driving a common-cathode display

Digit	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1
10–15	X	X	X	X	X	X	X	X	X	X	X

CD \ AB	00	01	11	10
00	1	1	1	0
01	1	1	1	1
11	X	X	X	X
10	1	1	X	X

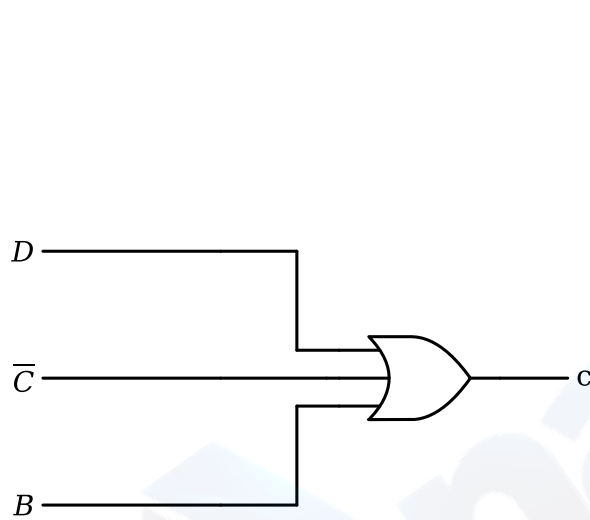
K-map for segment c (X = don't care)

CD \ AB	00	01	11	10
00	1	0	0	0
01	1	1	0	1
11	X	X	X	X
10	1	1	X	X

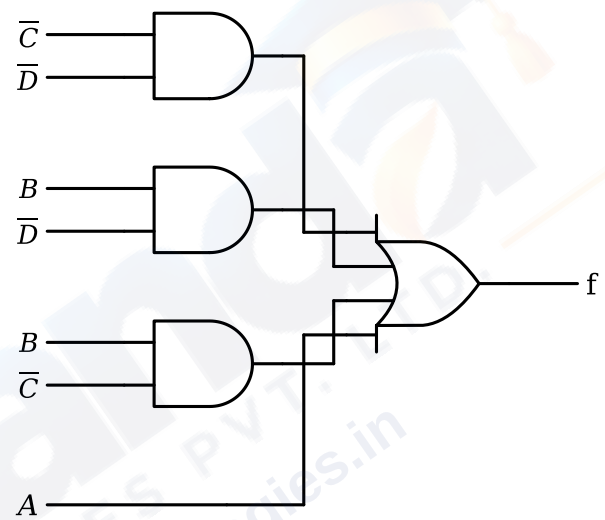
K-map for segment f (X = don't care)

$$c = D + \bar{C} + B \quad (\Sigma m(0, 1, 3, 4, 5, 6, 7, 8, 9) + d(10-15))$$

$$f = \bar{C}\bar{D} + B\bar{D} + B\bar{C} + A \quad (\Sigma m(0, 4, 5, 6, 8, 9) + d(10-15))$$



Logic diagram for segment c

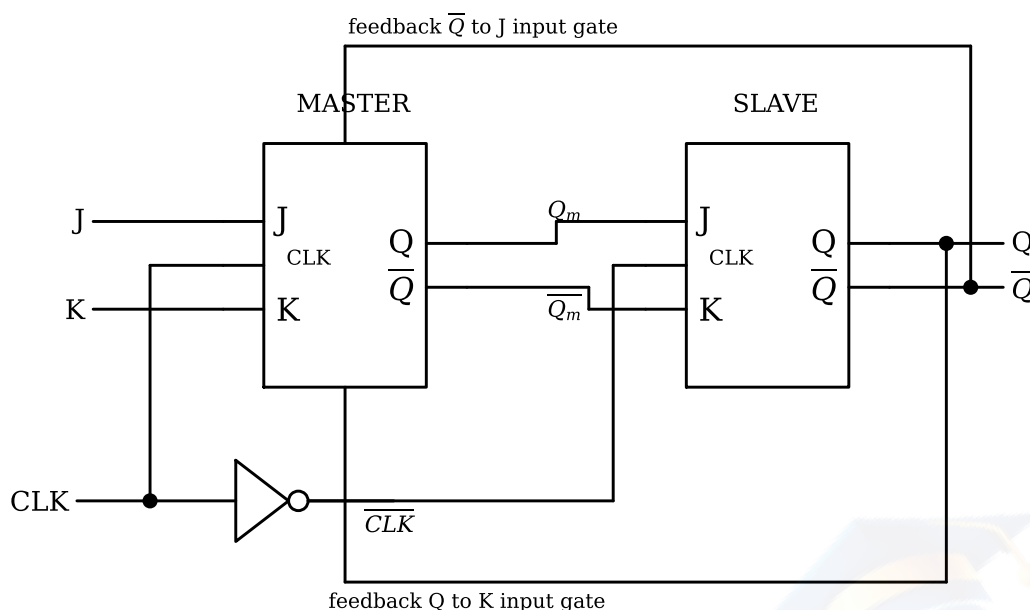


Logic diagram for segment f

(Truth table 2 M, block diagram 1 M, K-maps 2 M, logic diagrams 1 M)

b) Master-slave J-K flip-flop.

6 M



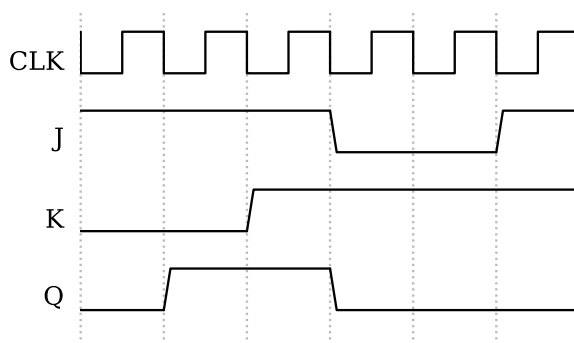
Master-slave J-K flip-flop (master enabled by CLK, slave by inverted CLK)

Construction: two J-K (clocked) flip-flops are connected in series. The master is enabled when $CLK = 1$; the slave gets the inverted clock and is enabled when $CLK = 0$. Outputs of the slave (Q , $\overline{Q}$) are fed back to the input gates of the master.

Working:

1. $CLK = 1$: master responds to J and K and its output Q_m is set/reset/toggled. Slave is disabled, so Q does not change.
2. CLK changes $1 \rightarrow 0$: master is disabled (holds Q_m); slave is enabled and copies Q_m to Q.
3. So the output changes only once per clock pulse, at the **negative edge**; even when $J = K = 1$ it toggles only once — **race-around is eliminated**.

CLK	J	K	Q_{n+1}	Action
↓	0	0	Q_n	No change
↓	0	1	0	Reset
↓	1	0	1	Set
↓	1	1	$\overline{Q_n}$	Toggle



Timing diagram: Q changes only at the falling edges of CLK (J, K taken while $CLK = 1$)

(Logic diagram 2 M, working 2 M, truth table 1 M, timing diagram 1 M)

c) MOD-6 synchronous counter using J-K flip-flops.

6 M

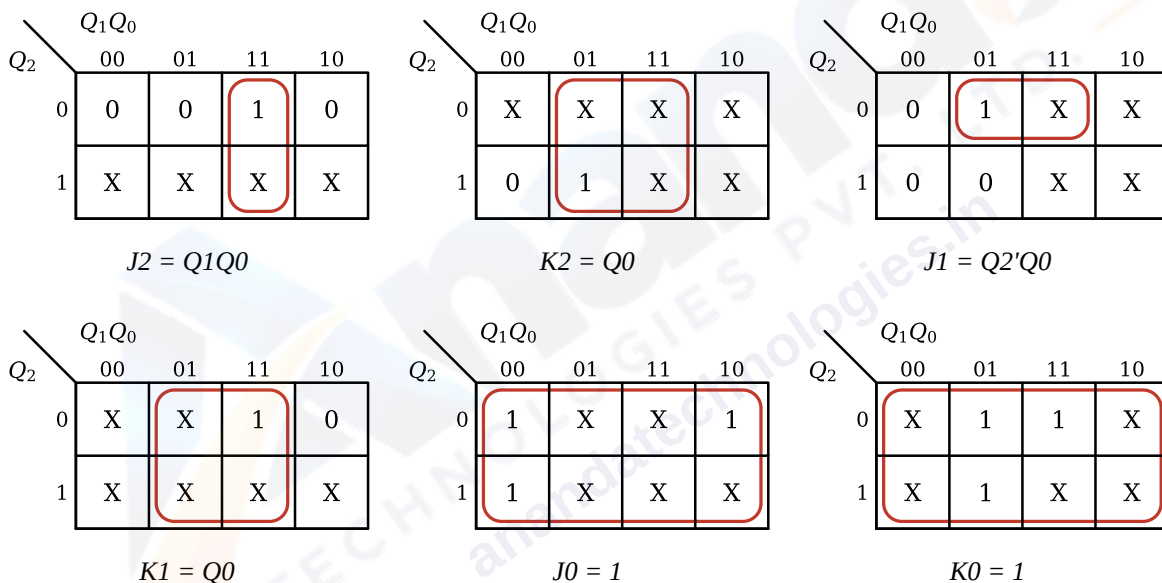
MOD-6 needs 3 flip-flops ($2^3 = 8 \geq 6$). Count sequence $000 \rightarrow 001 \rightarrow 010 \rightarrow 011 \rightarrow 100 \rightarrow 101 \rightarrow 000$. States 110 and 111 are unused (don't care).

State table with J-K excitation:

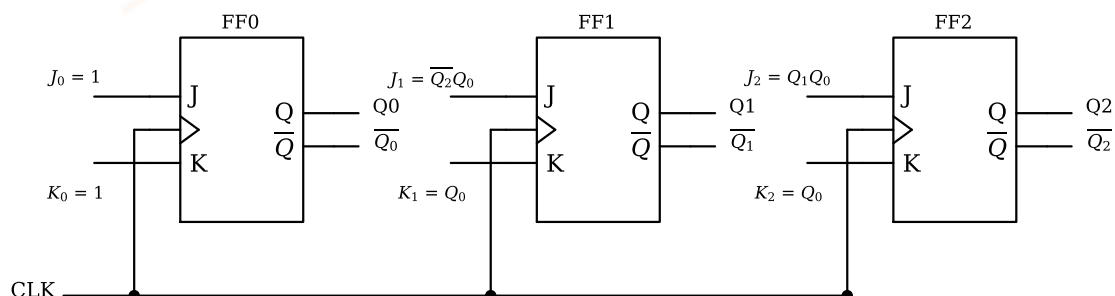
Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	0	0	0	X	1	0	X	X	1

Unused states 110, 111 (decimal 6, 7) are taken as don't care (X).

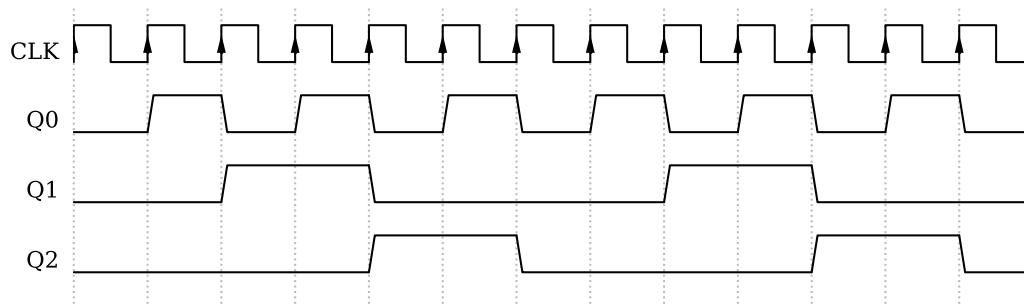
K-maps:



$$J_2 = Q_1Q_0, \quad K_2 = Q_0, \quad J_1 = \overline{Q_2}Q_0, \quad K_1 = Q_0, \quad J_0 = 1, \quad K_0 = 1$$



MOD-6 synchronous counter: all flip-flops clocked together; J, K inputs from the gates given by the equations



Timing diagram (repeats after 6 clock pulses)

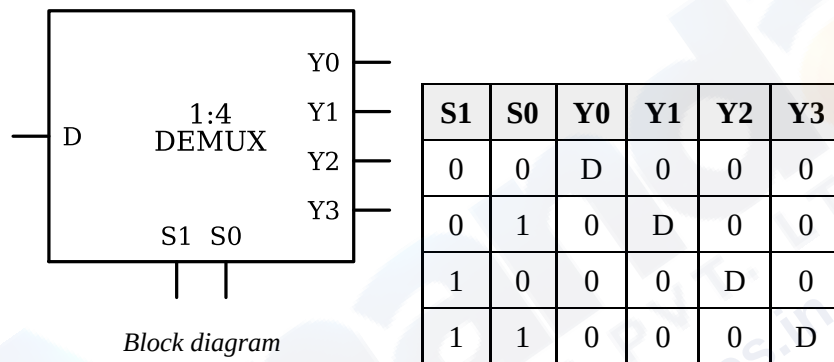
(State/excitation table 2 M, K-maps 2 M, logic diagram 1 M, timing 1 M)

Q.6

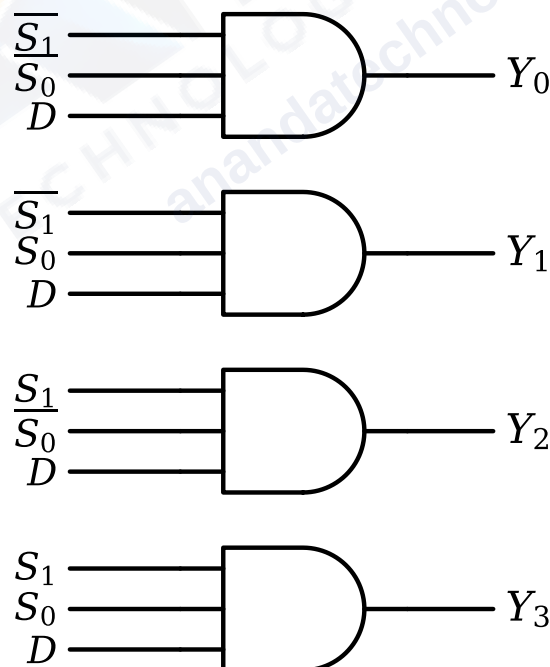
a) 1 : 4 DEMUX; 1 : 16 DEMUX tree.

6 M

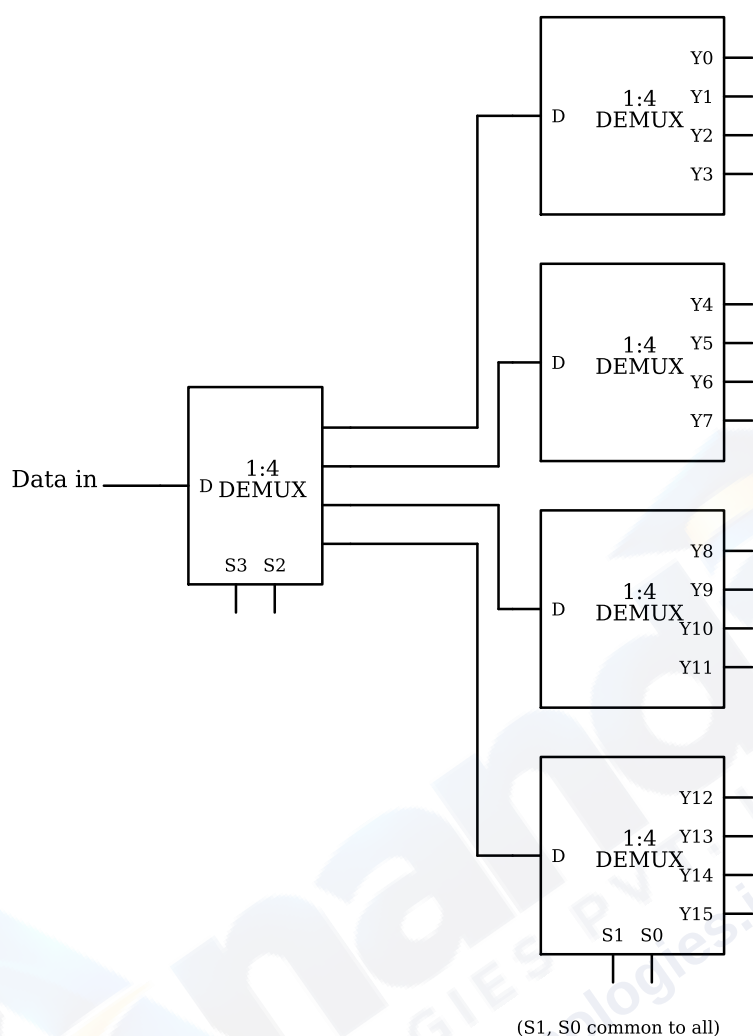
A **demultiplexer** has one data input, n select lines and 2^n outputs; the data input is routed to the output selected by the select lines. 1:4 DEMUX has 2 select lines $S_1 S_0$.



$$Y_0 = \overline{S_1}\overline{S_0}D, Y_1 = \overline{S_1}S_0D, Y_2 = S_1\overline{S_0}D, Y_3 = S_1S_0D$$



1:16 DEMUX using 1:4 DEMUX: 16 outputs need 4 select lines. One 1:4 DEMUX (select S3 S2) routes the data to one of four second-level 1:4 DEMUXes, which use S1 S0 to give Y0–Y15. Total 5 DEMUXes.

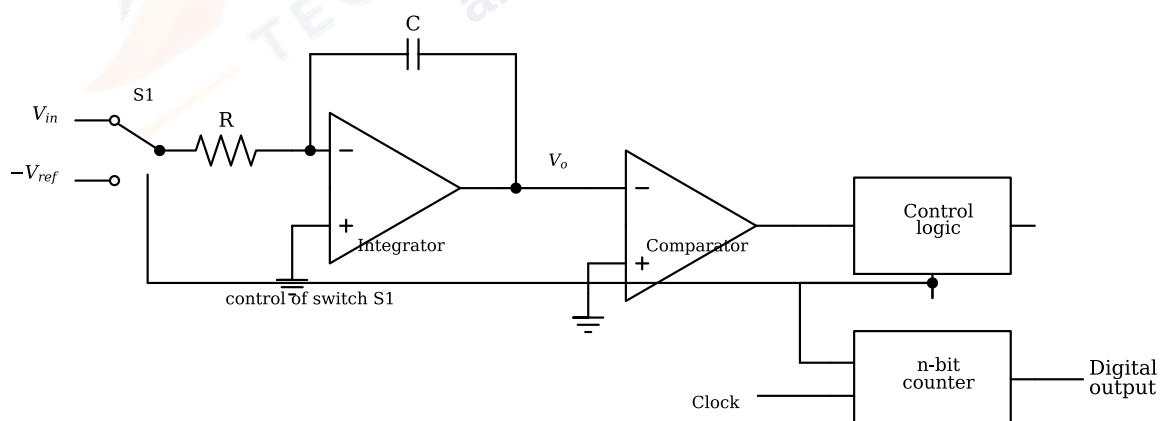


1:16 DEMUX using five 1:4 DEMUX

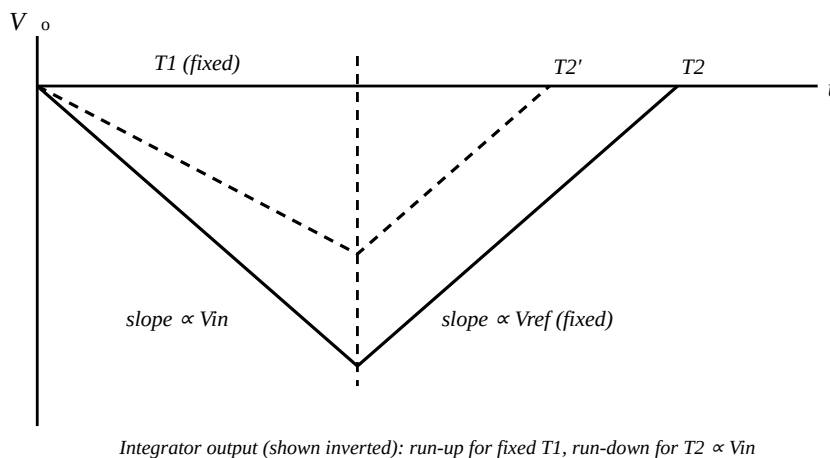
(Explanation + truth table 2 M, logic diagram 2 M, 1:16 tree 2 M)

b) Dual slope ADC.

6 M



Block diagram of dual slope ADC



Integrator output waveform of dual slope ADC (two input voltages)

Working:

1. Initially the counter is reset and the capacitor is discharged.
2. **Run-up (integration of input):** switch S_1 connects V_{in} to the integrator for a fixed time T_1 (the counter counts a fixed number N_1 of clock pulses, e.g. 2^n). The integrator output changes linearly with slope proportional to V_{in} .
3. At the end of T_1 the counter is reset and S_1 connects $-V_{ref}$. The integrator output now returns towards zero with a fixed slope (proportional to V_{ref}).
4. **Run-down:** the counter counts clock pulses until the comparator detects the zero crossing. This time T_2 is proportional to V_{in} : $V_{in} \cdot T_1 = V_{ref} \cdot T_2 \rightarrow T_2 = T_1 \cdot V_{in} / V_{ref}$, so the count $N_2 = N_1 \cdot V_{in} / V_{ref}$ is the digital output.

It is slow but very accurate and rejects noise (used in digital multimeters). The result does not depend on R , C or clock frequency.

Specifications of ADC: i) **Resolution** — smallest change in analog input that changes the output by 1 LSB = $V_{FS}/2^n$; ii) **Conversion time** — time taken to complete one conversion; iii) Accuracy / quantization error ($\pm \frac{1}{2}$ LSB); iv) Input voltage range.

(Block diagram 2 M, working with waveform 3 M, specifications 1 M)

c) Conversions.

6 M

i) $(B4E.6)_{16} \rightarrow$ decimal ($B = 11, E = 14$)

$$(B4E.6)_{16} = B(11) \times 16^2 + 4 \times 16^1 + E(14) \times 16^0 + 6 \times 16^{-1}$$

$$= 2816 + 64 + 14 + 0.375$$

$$= (2894.375)_{10}$$

ii) $(247.3)_{10} \rightarrow$ octal

Integer part (247): divide repeatedly by 8, write remainders from bottom to top.

Division	Quotient	Remainder
$247 \div 8$	30	7
$30 \div 8$	3	6
$3 \div 8$	0	3

Fraction part (0.3): multiply repeatedly by 8, take the integer parts from top to bottom.

Fraction $\times 8$	Product	Integer part
0.3×8	2.4	2
0.4×8	3.2	3
0.2×8	1.6	1
0.6×8	4.8	4

(Fraction does not terminate; taken up to 4 places.)

$$\therefore (247.3)_{10} = (367.2314)_8$$

iii) $(0110\ 1001\ 0111)_{BCD} \rightarrow$ **binary** (first convert BCD to decimal, then decimal to binary)

BCD group	0110	1001	0111
Decimal digit	6	9	7

$$(0110\ 1001\ 0111)_{BCD} = (697)_{10}$$

Integer part (697): divide repeatedly by 2, write remainders from bottom to top.

Division	Quotient	Remainder
$697 \div 2$	348	1
$348 \div 2$	174	0
$174 \div 2$	87	0
$87 \div 2$	43	1
$43 \div 2$	21	1
$21 \div 2$	10	1
$10 \div 2$	5	0
$5 \div 2$	2	1
$2 \div 2$	1	0
$1 \div 2$	0	1

$$\therefore (697)_{10} = (1010111001)_2$$

(2 M each)

Note: This is a **model question paper prepared by Ananda Technologies** only for practice and exam preparation, based on the pattern of previous MSBTE question papers and the K-scheme syllabus. It is **not an official MSBTE paper**, and it does **not** mean that these questions will come in the exam. Answers follow standard textbook points; marks shown are indicative. Free for students — anandatechnologies.in

