

डिजिटल के मूलभूत सिद्धांत एवं उपयोग (एस.टी.-52)

विषय सूची

क्र.सं.	नाम	पृष्ठ सं.
1.	डिजिटल प्रक्रिया	1
2.	नम्बर सिस्टम (Number System)	2
3.	बाइनरी अंक गणित	10
4.	ट्रांजिस्टर का स्विच की भांति प्रयोग	17
5.	लॉजिक गेट (Logic Gates)	19
6.	बूलीय बीज गणित	27
7.	डिजिटल कोड	29
8.	कोडर (Coder)	31
9.	डिकोडर (Decoder)	33
10.	पैरिटी (Parity) एवं पैरिटी परीक्षक	36
11.	मल्टीप्लेक्सर (MUX)	41
12.	अर्थमैटिक लॉजिक यूनिट (Arithmetic Logic Unit)	43
13.	फ्लिप-फ्लॉप (Flip-Flop)	49
14.	रजिस्टरस (Registers)	56
15.	काउन्टरस (Counters)	59
16.	स्मृतियाँ, अतिरिक्त स्मृतियाँ (Memories)	68
17.	विद्युत डिजिटल घड़ी	82
18.	वस्तुनिष्ठ	92

‘डिजिटल प्रक्रिया’ प्रोसेस (DIGITAL PROCESS)

डिजिटल प्रक्रिया/प्रोसेस का अर्थ ऐसी प्रक्रिया (Process) से है जो पूर्व निश्चित अवस्थाओं/कन्डीशनस से गुजरती है। इन अवस्थाओं/कन्डीशनस के निर्धारण/सलेक्शन के लिए स्विच, रिले कॉन्टेक्ट या इलेक्ट्रॉनिक लॉजिक सर्किट/सर्किट प्रयोग किये जाते हैं। इलेक्ट्रॉनिक्स लॉजिक सर्किट का प्रयोग डिजिटल सिस्टम पर यंत्र की कार्य गति तथा निर्भरता को बहुत बढ़ा देता है।

लॉजिक सर्किट (Logic Circuit)

इन्हें दो भागों में बांटा जा सकता है –

1. निर्णयात्मक एलीमेन्ट – जैसे AND, OR, NOT गेट आदि।
2. मेमोरी एलीमेन्ट—जैसे फ्लिप-फ्लॉप सर्किट।

इसमें वेरीयेबल (Variable) लॉजिक की निम्न विशेषता होती है।

- (a) यह केवल दो अवस्थाओं को प्राप्त कर सकता है। इन अवस्थाओं को कई नामों से जाना जाता है। जैसे—ट्रू—फाल्स (True-False) या हाई—लो (High-Low) या (1-0) वन—जीरो या ऑन—ऑफ (ON-OFF)।
- (b) वेरीयेबल (Variable) लॉजिक एक समय में दोनों अवस्थाओं में एक साथ नहीं हो सकता है।
- (c) वेरीयेबल (Variable) एक समय में दोनों अवस्थाओं में एक साथ नहीं हो सकता है।

पॉजिटिव लॉजिक (Positive Logic) – जब अधिक पॉजिटिव वोल्टता को लॉजिक (1) तथा कम पॉजिटिव वोल्टता को लॉजिक (0) की तरह प्रयोग करते हैं तो इसे पॉजिटिव लॉजिक कहा जाता है।

[2] ST-52

नेगेटिव लॉजिक (Negative Logic) – जब अधिक नेगेटिव वोल्टेज को (1) तथा कम नेगेटिव वोल्टेज को लॉजिक (0) की तरह प्रयोग किया जाता है तो उसे नेगेटिव लॉजिक कहा जाता है।

नम्बर सिस्टम (Number System)

युगों से नम्बर सिस्टम का प्रयोग होता रहा है चाहे उनका रूप बदलता रहा। उनके द्वारा गिनने की प्रक्रिया से फिर अंकगणित तथा अन्य गणितीय शाखाओं का विकास हुआ और वर्तमान काल में दस अंकों के नंबर सिस्टम का आम प्रचलन है। यह दस अंक 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, के रूप में मान्य है। इसे डेसीमल सिस्टम (Decimal number system) कहा जाता है परन्तु कुछ कार्यों में बाइनरी सिस्टम का प्रयोग अधिक होता है तथा यही सुविधा जनक होता है। विशेष रूप से कम्प्यूटर के क्षेत्र में, जहां स्विचों के द्वारा कार्य किया जाता है। नम्बर सिस्टम के अन्य महत्वपूर्ण सिस्टम भी हैं। जैसे—ऑक्टल सिस्टम (Octal number system), हैक्सा डेसीमल सिस्टम (Hexadecimal number system).

बाइनरी सिस्टम (Binary System): इसमें 0 तथा 1 का प्रयोग किया जाता है। इसमें गिनती निम्न प्रकार से की जाती है।

डेसीमल	बाइनरी (Binary)
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001

इस बाइनरी में 1001 का अर्थ क्या होगा। इसे इस प्रकार समझे कि सबसे दाएं स्थान का मान $2^0=1$, फिर बायें स्थान का मान $2^1=2$ आदि हुआ करता है। इसलिए संख्या का मान ज्ञात करने के लिए उसमें प्रयुक्त अंकों को उनके स्थानीय मान (Place Value) से गुणा करके आपस में जोड़ देते हैं। जैसे अभी 1001 का अर्थ पूछा गया था। तो नियमानुसार सबसे दायें '1' है फिर बायें 0 है तो अंकों का मान रखने पर देखें –

$$\begin{aligned} &1001 \\ = &1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ = &8 + 0 + 0 + 1 = 9 \end{aligned}$$

अर्थात् 1001 का मान 9 होता है।

इन्हें और स्पष्ट रूप में लिखने के लिए निम्न तरीका अपनाया जाता है।

$$(1001)_2 = (9)_{10}$$

अब प्रश्न उठता है कि जब संख्या बड़ी हो तो उसे दोनों नम्बर सिस्टम में किस प्रकार लिखा जाये अथवा परिवर्तित किया जाए। वास्तव में यह बहुत अधिक मुश्किल नहीं है। नीचे डेसीमल नम्बर (Decimal number) को बाइनरी में लिखने (या परिवर्तित करने) की विधि समझाई गई है।

उदाहरण—217 को बाइनरी में लिखो।

इसके लिए 217 को निम्न प्रकार से 2 से भाग देते चले जाएं और शेष को दायीं और अलग से लिखते जायें।

2	217	1 शेष
2	108	0 "
2	54	0 "
2	27	1 "
2	13	1 "
2	6	0 "
2	3	1 "
2	1	1 "
	0	

इसमें जो शेष बचे हैं उन्हें नीचे से प्रारंभ करते हुए बाएं से दाएं लें अब इसमें सबसे नीचे शेष 1 है फिर 1 है फिर 0 है तो इसी क्रम से बाएं से दाएं लिखना है।

11011001 अर्थात्

$$(217)_{10} = (11011001)_2$$

[4] ST-52

अब इस उत्तर को प्रमाणित करना चाहे तो पहले बताये गये तरीके से करके देखें –

$$11011001$$

$$= 1 \times 2^7 + 1 \times 2^8 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

$$= 128 + 64 + 0 + 16 + 8 + 0 + 0 + 1$$

$$= 217$$

अर्थात् हमारा उत्तर उपरोक्त 10011001 सही था। अब यदि कोई संख्या ऐसी हो जिसके मध्य में दशमलव चिन्ह (बिन्दु) हो तो तब उसे बाइनरी में कैसे लिखा जाए। इसके लिए दशमलव के बाद की संख्या को दो से गुणा करते जाएं और हासिल (दशमलव से बाएं ओर आने वाले अंक) पूर्णांक को लिखते जाएं। इन हासिल पूर्णांकों को ऊपर से नीचे के क्रम में बाएं से दाएं लिखते चले जाएं।

उदाहरण $(0.4375)_{10}$ को बाइनरी में लिखिए

$$\begin{array}{lcl} = 0.4375 \times 2 = & \downarrow & 0.8750 \\ = 0.8750 \times 2 = & & 1.7500 \\ = 0.7500 \times 2 = & & 1.5000 \\ = 0.5000 \times 2 = & \downarrow & 1.0000 \end{array}$$

पूर्णांकों को क्रमवार बायें से दायें लिखा –

$$= 0111$$

अर्थात् $(0.4375)_{10} = (0.0111)_2$ उत्तर

यहां दशमलव बिन्दु अपने पूर्व स्थान पर ज्यों का त्यों लगा दिया जाता है।

बाइनरी से डेसीमल—बाइनरी सिस्टम में विभिन्न स्थानों का स्थानीय मान निम्न तालिका में दिखाया गया है।

$$\begin{array}{ccccccc}
 1 & 1 & 1 & 1 & . & 1 & 1 & 1 & 1 & 1 \\
 2^3 & 2^2 & 2^1 & 2^0 & . & 2^{-1} & 2^{-2} & 2^{-3} & 2^{-4} & 2^{-5} \\
 & & & & \downarrow & & & & & \\
 & & & & \text{दशमलव बिन्दु} & & & & &
 \end{array}$$

अर्थात् बाइनरी संख्या को डेसीमल में बदलने के लिए प्रत्येक बाइनरी के स्थानीय मान के लिए तालिका के अनुसार 2 की घात लगानी पड़ती है। दशमलव बिन्दु से पहले की डिजिट के लिए +ve घात तथा बाद की डिजिट के लिए -ve घात लगती है।

उदाहरण-1 $(10101111)_2$ को डेसीमल सिस्टम में लिखो?

$$\begin{aligned}
 &= 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\
 &= 128 + 0 + 32 + 0 + 8 + 4 + 2 + 1 \\
 &= 175
 \end{aligned}$$

अर्थात् $(10101111)_2 = (175)_{10}$ उत्तर

उदाहरण-2 $(0.1101)_2$ को डेसीमल सिस्टम में लिखो?

$$\begin{aligned}
 0.1101 &= 1 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} \\
 &= .5 + .25 + 0 + 0 + 0.0625 \\
 &= 0.8125
 \end{aligned}$$

उत्तर = 0.8125 अर्थात् $(0.1101)_2 = (0.8125)_{10}$

इस प्रकार हम देखते हैं कि बाइनरी संख्याओं को डेसीमल में बदलने के लिए नियम यह होता है कि बाइनरी संख्या के विभिन्न अंकों को उनके स्थानीय मान से गुणा करके सबका योग कर लेना चाहिए।

डेसीमल सिस्टम (Decimal system) और बाइनरी सिस्टम (Binary system) के इस संबंध को जानने के पश्चात् अन्य प्रकार की अंक सिस्टम को समझना आसान है। यहां आगे मुख्य रूप से दो और नंबर सिस्टम पर चर्चा की जाएगी, यह हैं –

[6] ST-52

1. ऑक्टल नंबर सिस्टम (Octal Number System)
2. हैक्सा डेसीमल सिस्टम (Hexadecimal Number System)

‘ऑक्टल नम्बर सिस्टम’

(Octal Number System)

इस सिस्टम में केवल 8 (0, 1, 2, 3, 4, 5, 6, 7) अंकों का प्रयोग किया जाता है। यह नंबर सिस्टम भी कम्प्यूटर उद्योग में बहुत उपयोगी है किसी भी संख्या को ऑक्टल सिस्टम में निम्न प्रकार से लिखते हैं।

बाइनरी नं.	ऑक्टल नं.
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

उदाहरण

यदि 010111 को यदि ऑक्टल सिस्टम में लिखना हो तो पहले बाइनरी संख्या को दायें से बायें चलते हुए तीन तीन अंकों के समूह में बांट लेते हैं फिर प्रत्येक समूह को जिसमें तीन डिजिट होंगे ऑक्टल में बदल देते हैं जैसे –

010 111 दो, तीन-तीन, बाइनरी समूह बने।

010=2, 111=7, बाइनरी डिजिट का मान ऑक्टल में 27 हुआ।

$$\text{अर्थात् } (010111)_2 = (27)_8$$

नोट—जिस प्रकार बाइनरी को $(\dots)_2$ से दर्शाया जाता है उसी प्रकार ऑक्टल को $(\dots)_8$ से दर्शाया जाता है तथा घात 8 की लगती है।

'Binary to Octal' 'बाइनरी से ऑक्टल'

$(110\ 101\ 111\ 001)$ को ऑक्टल में लिखो।

= पहले बाइनरी संख्या को तीन-तीन के समूह में बांटो

= $110\ 101\ 111\ 001$ अब प्रत्येक समूह को ऑक्टल में लिखो

$110=6$, $101=5$, $111=7$, $001=1$ अर्थात् ऑक्टल संख्या बनी 6571

$$(110\ 101\ 111\ 001)_2 = (6571)_8 \text{ उत्तर}$$

'Octal to Binary/Decimal' 'ऑक्टल से बाइनरी/डेसीमल'

जिस प्रकार बाइनरी से डेसीमल संख्या बनाते समय 2 की घात लगाई जाती है उसी प्रकार ऑक्टल से कभी सीधे डेसीमल बनाना हो तो 8 की घात लगाई जाती है। परन्तु ऑक्टल से बाइनरी बनाते समय केवल प्रत्येक अंक को 3 बिट के बाइनरी रूप में लिख दिया जाता है।

उदाहरण — 1. $(270)_8 = (?)_2$

$$2 = 010, 7 = 111, 0 = 000$$

$$\text{अर्थात् } (270)_8 = (010\ 111\ 000)_2 \text{ उत्तर}$$

2. $(1357)_8 = (?)_2$

$$1 = 001, 3 = 011, 5 = 101, 7 = 111$$

$$(1357)_8 = (001\ 011\ 101\ 111)_2 \text{ उत्तर}$$

[8] ST-52

हैक्सा डेसीमल नम्बर सिस्टम (Hexa Decimal Number System)

हैक्सा डेसीमल नम्बर सिस्टम में 16 संकेतों का प्रयोग किया जाता है वे निम्न प्रकार से मान्य है।

बाइनरी	हैक्सा डेसीमल	बाइनरी	हैक्सा डेसीमल
0000	0	1001	9
0001	1	—	—
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F
1000	8		

‘बाइनरी से हैक्साडेसीमल (Binary to Hexadecimal)

सभी हैक्साडेसीमल संख्याओं को चार-चार समूहों के समूह की बाइनरी संख्या के रूप में लिखा जा सकता है। इसके लिए बाइनरी संख्या को हैक्सा डेसीमल में बदलने के लिए बाइनरी अंकों को चार-चार के समूह में बांट लेते हैं फिर प्रत्येक समूह को अलग-अलग हैक्सा डेसीमल संकेतों द्वारा प्रदर्शित कर देते हैं।

उदाहरण—बाइनरी संख्या 0110 0110 1011 का हैक्सा डेसीमल में बदलना है तो, पहले बाइनरी संख्या के अंकों को चार-चार अंकों के समूह में दायें से बायें की तरफ परिवर्तित किया, फिर प्रत्येक का मान रखा—

$$0010=6, 0110=6, 1011=B \rightarrow 66B.$$

$$\text{अर्थात् } (0110\ 0110\ 1011)_2 = (66B)_{16} \text{ उत्तर}$$

उदाहरण – बाइनरी संख्या में प्वाइंट आ जाने पर

$$\text{जैसे } (0001\ 1001\ 0101 . 0100\ 1000)_2 = (?)_{16}$$

इसमें प्वाइंट के बाद के अंको को बायें से दायें की तरफ चार-चार अंकों के समूह में परिवर्तित कर देते हैं और फिर प्रत्येक का मान रख देते हैं।

$$= \frac{0001}{1} \quad \frac{1001}{9} \quad \frac{0101}{5} \quad . \quad \frac{0100}{4} \quad \frac{1000}{8} = 195.48$$

$$\text{अर्थात् } (0001\ 1001\ 0101 . 0100\ 1000)_2 = (198.48)_{16}$$

हैक्सा डेसीमल से बाइनरी (Hexadecimal to Binary)

क्योंकि हैक्सा डेसीमल का मान 16 है और बाइनरी का 2 है। इसलिए हैक्साडेसीमल से बाइनरी में परिवर्तन बहुत सरल हो जाता है हमें केवल प्रत्येक हैक्सा डेसीमल डिजिट को एक-एक करके बाइनरी में बदलना होता है। इसमें प्रत्येक हैक्सा डेसीमल डिजिट की चार बाइनरी डिजिट होना आवश्यक है।

उदाहरण –

$(E7AB)_{16}$ को बाइनरी में लिखो?

$$E = 1110$$

$$7 = 0111$$

$$A = 1010$$

$$B = 1011$$

$$\text{इसलिए } (E7AB)_{16} = (1110\ 0111\ 1010\ 1011)_2 \text{ होगा।}$$

हैक्साडेसीमल संख्या के मध्य दशमलव आ जाने पर दशमलव (प्वाइंट) अपनी पूर्व स्थिति में ही रहता है।

[10] ST-52

जैसे – $(267.3D)_{16} = (?)_2$

$= 0010\ 0110\ 0111 . 0011\ 1101$

अर्थात् – $(267.3D)_{16} = (0010\ 0110\ 0111 . 0011\ 1101)_2$

बाइनरी अंकगणित (Binary Arithmetic)

डेसीमल सिस्टम (Decimal system) में योग, अन्तर, गुणा तथा भाग की प्रक्रिया से अधिकांश लोग परिचित हैं। इसलिए बाइनरी गणित की क्रिया को समझने में इस क्रिया के डेसीमल सिस्टम के रूप में समुचित मदद मिल सकती है।

योग Addition – डेसीमल सिस्टम में $7+8 = 15$

इसे निम्न प्रकार समझें –

	दहाई	इकाई
	0	7
	+ 0	8
स्थानीय योग	0	5
हासिल बार	1	
पूर्ण योग	1	5

$= (\text{हासिल} + \text{स्थानीय योग } 1 \times 10 + 5 = 15)$

[carry] [place value sum]

यदि दहाई के स्थान पर भी अंक होते तो उनसे भी स्थानीय योग मिलता और इकाई स्थान से जो हासिल आया था उसे उसमें जोड़ दिया जाता। अब इसी क्रिया को बाइनरी में करके देखें। परन्तु उससे पहले हमें यह पता होना चाहिए कि बाइनरी गणित के क्या-क्या नियम होते हैं।

नियम – $1+0=1$ $0+0=0$
 $0+1=1$ $1+1=10$ अर्थात् $(1+1=0$ तथा हासिल 1)

इसमें इस अन्तिम स्थिति में विचित्रता दिखाई देती है। परन्तु ध्यान यह होना चाहिए कि यहां 10 का अर्थ दस नहीं है। बल्कि बाइनरी सिस्टम में इसका अर्थ दो है। जो ठीक लगता है इसको इस प्रकार भी कहा जा सकता है।

$$\begin{aligned}
 1+1 &= \text{हासिल (Carry) 1 स्थानीय योग 0} \\
 &= 1 \times 2^1 + 0 \times 2^0 \\
 &= 2+0 \\
 &= 2
 \end{aligned}$$

और अधिक बड़ी बाइनरी संख्याओं को जोड़ते समय इसी नियम का पालन करना पड़ता है।

$$10+1 = 11 \text{ तथा } 11+1 = 100$$

$$\begin{array}{r}
 \text{उदाहरण :} \quad 110110 \\
 +101010 \\
 \hline
 1100000
 \end{array}$$

बाइनरी		डेसीमल
110110		54
101010		42
1100000	← योग →	96

$$\begin{aligned}
 \text{बाइनरी योग} &= 1100000 \\
 &= 1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 \\
 &= 64 + 32 \\
 &= 96
 \end{aligned}$$

[12] ST-52

अन्तर (Subtraction) – बाइनरी अन्तर के नियम

$$0 - 0 = 0$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

$$10 - 1 = 1 \quad 0 - 1 \Rightarrow D \text{ (Difference)} = 1 \text{ \& B (Borrow)} = 1$$

मूल रूप से घटाने की क्रिया करते समय उल्टी गिनती करते हैं। उदाहरण – जैसे 5 में से 3 को घटाने के लिए 5 से उल्टी गिनती करनी पड़ेगी। जब तक 3 नहीं आ जाता। इस प्रकार दो गिनते ही तीन आ जायेगा इसलिए परिणाम 2 होता है जब घटाने वाली संख्या बड़ी हो तो उच्च स्थान से उधार लेना पड़ता है।

उदाहरण -

दहाई	इकाई
6	2
-3	8
2	4

उत्तर = 24

उदाहरण - $(110)_2$ में से $(011)_2$ हटायें

6	1	1	0
-3	0	1	1
3	0	1	1

उत्तर = $(011)_2$

उदाहरण -

13	1101
-4	-0100
9	1001

उत्तर = $(1001)_2$

घटाने की विधियां :

- (i) परोक्ष विधि (Direct Method)
 - (ii) प्रतिलोम विधि (Inverse Method)
 - (iii) पूरक विधि (Complement Method)
- (i) परोक्ष विधि (Direct Method) – यह आम प्रचलित विधि है जैसा पिछले उदाहरणों से स्पष्ट किया गया है।
- (ii) प्रतिलोम विधि (Inverse Method) – इसमें ऐसी संख्या ज्ञात की जाती है जिसको यदि घटाने वाली संख्या (Subtract end) में जोड़ दें तो जिसमें से घटना था (minus end) प्राप्त हो जाती है। अर्थात्

यदि $A - B = C$ परोक्ष विधि में तो

$B + C = A$ प्रतिलोम विधि में

(iii) पूरक विधि (Complement Method) – इस विधि में घटाने का कार्य जोड़ कर किया जाता है। इसमें घटाने वाली संख्या का पूरक ज्ञात किया जाता है तथा पूरक में जिसमें से घटना था को जोड़ दिया जाता है फिर समुचित पूरक ज्ञात कर उत्तर प्राप्त कर लेते हैं इस प्रकार उपरोक्त विधि में घटाने का कार्य जोड़ विधि का प्रयोग करके किया जाता है क्योंकि कम्प्यूटर केवल जोड़ ही कर सकता है अतः कम्प्यूटर परिचालन के लिए इस विधि की जानकारी अति आवश्यक है। पूरक विधि दो प्रकार की होती है।

1. प्रथम पूरक विधि (1's Complement Method)
2. दूसरा पूरक विधि (2's Complement Method)

प्रथम पूरक विधि का प्रयोग कर घटना -

संख्या का प्रथम पूरक बनाना :- जिस बाइनरी संख्या का प्रथम पूरक (1's Complement) बनाना होता है उसमें सभी 1 को शून्य तथा सभी शून्यों को 1 लिख देते हैं। इस प्रकार प्राप्त बाइनरी को पहली बाइनरी का प्रथम पूरक (Complement) कहते

[14] ST-52

हैं। जैसे – 110011001 का प्रथम पूरक = 001100110 होगा। प्रथम पूरक उस संख्या का बनाते हैं जिसको घटना है अर्थात् Subtract end का ही प्रथम पूरक बनाते हैं।

घटाने के नियम -

- (i) सबसे पहले जिस संख्या को घटाना है (Subtract end) का प्रथम पूरक लिखो।
- (ii) प्राप्त बाइनरी में जिसमें से घटना था (Minus end) को जोड़ो।
- (iii) प्राप्त उत्तर दो प्रकार का हो सकता है –
 - (अ) उत्तर के अंत में (Carry) हासिल न आया हो,
 - (ब) उत्तर के अंत में (Carry) हासिल आया हो,
 - (अ) उत्तर में यदि अंत में कोई हासिल नहीं आया तब उत्तर (-ve) ऋणात्मक है एवं प्रथम पूरक के रूप में है। अब सही उत्तर ज्ञात करने के लिए प्राप्त बाइनरी को पुनः प्रथम पूरक में बदलकर सही उत्तर ज्ञात कर लेते हैं।
 - (ब) यदि उत्तर के अंत में हासिल आया है तो उत्तर धनात्मक +ve है तथा सही उत्तर ज्ञात करने के लिए L.S.B. (Least Significant Bit) न्यूनतम सार्थक बिट में एक जोड़कर सही उत्तर ज्ञात कर लेते हैं।

उदाहरण - (i) 7-9

(ii) 9-2

(i) (7-9) 0111

1001

1001 का प्रथम पूरक = 0110

0111 Minus End को जोड़ा = +0111

1101

उत्तर में हासिल नहीं आया अतः उत्तर -ve तथा प्रथम पूरक में।

अतः सही उत्तर 1101 का प्रथम पूरक = (0010)

(2) (9-7)	1001	0111 का प्रथम पूरक =	1000
	-0111	Minu end 1001 को जोड़ा	+1001
	1001		1000
			1 0001

जोड़ने पर उत्तर के अंत में हासिल प्राप्त हुआ जो में दिखाया गया है अतः सही उत्तर धनात्मक है जो L.S.B. में 1 जोड़ने पर प्राप्त होगा।

1 जोड़ा LSB में	1 0001	
	→ +1	
	0010	उत्तर = (0010) ₂

दूसरा पूरक (2's complement) विधि का प्रयोग करते हुए घटाना -

संख्या का दूसरा पूरक ज्ञात करना – जिस बाइनरी संख्या का 2's कोम्पलिमेंट ज्ञात करना है तो पहले उस बाइनरी का प्रथम पूरक ज्ञात करते हैं तथा फिर उसके L.S.B. में 1 जोड़कर 2's कोम्पलिमेंट ज्ञात कर लेते हैं।

- नोट करें – (1) किसी भी बाइनरी का 2's पूरक उस बाइनरी का -ve होता है जैसे डेसीमल में 2 का -ve -2 आदि।
- (2) किसी बाइनरी के 2's कोम्पलिमेंट का 2's पूरक (complement) पुनः वही बाइनरी प्राप्त हो जाती है।

घटाने के नियम -

- (i) घटाने वाली संख्या का 2's कोम्पलिमेंट ज्ञात करते हैं।
- (ii) Minus end जिसमें से घटाना है को इसमें जोड़ते हैं।
- (iii) अब यदि प्राप्त उत्तर में

[16] ST-52

(अ) हासिल (carry) नहीं है तो उत्तर -ve है तथा सही उत्तर ज्ञात करने के लिए पुनः दूसरा पूरक (2's complement) प्राप्त कर सही उत्तर प्राप्त कर लेते हैं।

(ब) हासिल है तो उत्तर धनात्मक है तथा उसके हासिल छोड़ कर (ignore carry) शेष बाइनरी ही उत्तर होता है।

उदाहरण-1 0111

—1001

1001 का 1's Complement = 0110

1001 का 2's Complement = + 1

0111

0111 को (Minus End) जोड़ा + 0111

1110

प्राप्त उत्तर में हासिल नहीं है
इसलिये उत्तर -ve है तथा 2's
complement में है अतः 1110 का
2's complement

1's Complement = 0001

2's Complement = + 1

0010

उत्तर = $-(0010)$

उदाहरण-2 $(9-7)$, $(1001-0111)$

(1) 0111 का 1^s Complement = 1000

(2) 0111 का 2^s Complement = + 1

1001

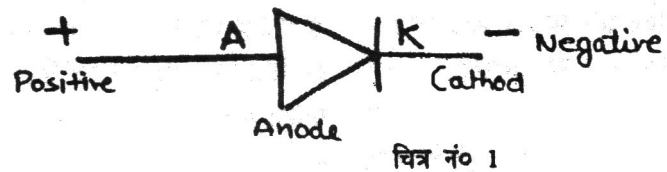
प्राप्त उत्तर में Minus end में जोड़ा 1001 + 1001

1 0010

उत्तर के अंत में हासिल प्राप्त हुआ अतः उत्तर धनात्मक है तथा 1 को छोड़कर शेष ही सही उत्तर है।

अतः $1001-0111 = 0010$ उत्तर

Symbol of Diode डायोड कर संकेत

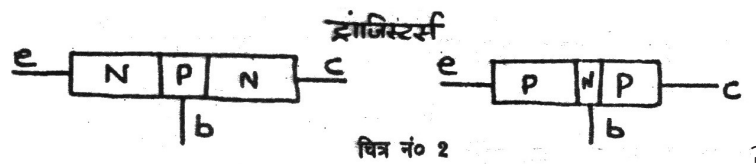


ट्रांजिस्टर (Transistor)

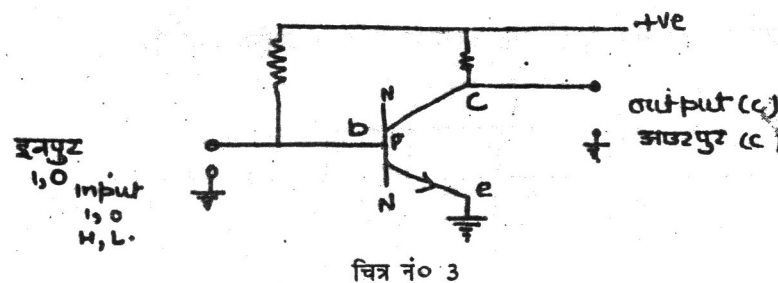
ट्रांजिस्टर के कार्य करने की दो अवस्थाएँ हैं –

- (i) एमीटर-बेस फॉरवर्ड बायस
- (ii) बेस-कलेक्टर रिवर्स बायस

NPN ट्रांजिस्टर में इलेक्ट्रॉन के बहने से इलेक्ट्रॉन (-ve) करंट एवं PNP ट्रांजिस्टर में होल के कारण होल (+ve) करंट बहती है क्योंकि इलेक्ट्रॉन करंट होल करंट से तेज होती है अतः डिजिटल सर्किट (Circuit) में हम केवल NPN ट्रांजिस्टरों का ही प्रयोग करते हैं –



ट्रांजिस्टर का स्विच की भांति प्रयोग - Transistor as Switch

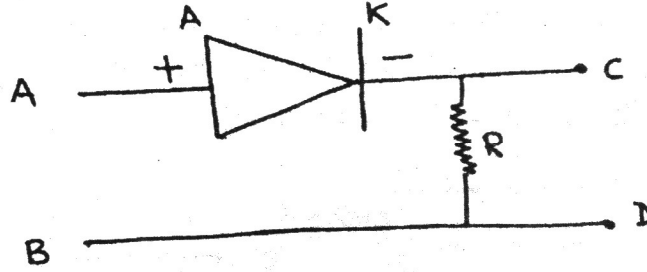


[18] ST-52

ट्रांजिस्टर बायसिंग इस प्रकार की जाती है कि जब इनपुट में (0) होता है तो टर्मिनल c और e के बीच आउटपुट वोल्टता मिलता रहता है। जब इनपुट हाई (1) तो ट्रांजिस्टर (NPN type) में करंट कलेक्टर से एमीटर की ओर तेजी से बहती है ऐसी स्थिति में ट्रांजिस्टर कलेक्टर एमीटर शॉर्ट माना जाता है इस प्रकार ट्रांजिस्टर के टर्मिनल c और e के मध्य शॉर्ट होने से टर्मिनल e का अर्थ c टर्मिनल पर पहुंच जात है। और आउटपुट लो (0) हो जाता है। इस प्रकार डिजिटल में ट्रांजिस्टर स्विच की भांति कार्य करता है। स्विचिंग के लिए प्रयुक्त ट्रांजिस्टर को बफर ड्राइवर भी कहा जाता है।

Diode As Switch डायोड का स्विच की भांति प्रयोग -

उपरोक्त चित्र के अनुसार जब A टर्मिनल पर इनपुट 'लो' (0) होता है तो C टर्मिनल पर कोई आउटपुट वोल्टेज नहीं मिलती। जब A टर्मिनल पर इनपुट हाई (1) होता है तो डायोड की फॉरवर्ड बायसिंग के कारण C टर्मिनल पर आउटपुट वोल्टेज प्राप्त होती है। इस प्रकार डायोड एक स्विच की भांति कार्य करता है।



चित्र नं० 4

Square Waves वर्गाकार तरंगें -

डिजिटल में चूंकि 1 और 0 (+5V और 0V) के सिगनल ही कार्य करते हैं इसलिए बीच में कोई परिवर्तन न होने के कारण वर्गाकार तरंग फंक्शन होता है चित्र में 0011000 का उदाहरण दिया गया है जो कि वर्गाकार तरंग में दर्शाया गया है।



चित्र नं० 5

लॉजिक गेट Logic Gates

लॉजिक गेट (तर्क द्वार) कुछ तर्क क्रियायें करने के लिए डिजिटल स्विच हैं या लॉजिक गेट डिजिटल सर्किट की आधार शिलाएं हैं लॉजिक गेट एक ऐसा इलेक्ट्रॉनिक्स सर्किट है जिसके इनपुट में एक, दो या दो से अधिक लॉजिक सिगनल दिए जा सकते हैं उससे निकलने वाला आउटपुट सिगनल किसी एक इनपुट सिगनल या सभी इनपुट सिगनल की पूर्व निश्चित अवस्थाओं पर निर्भर करता है।

फण्डामेन्टल गेट (Fundamental Gates)

फण्डामेन्टल लॉजिक गेट (आधारभूत) अथवा मूलभूत लॉजिक गेट (प्राइमरी) तीन प्रकार के होते हैं –

(अ) एण्ड गेट (AND Gate)

(ब) ऑर गेट (OR Gate)

(स) नॉट गेट (NOT Gate)

ये गेट जो कार्य करते हैं वह बहुत ही साधारण हैं तथा इसे समझना आवश्यक है। इन्हीं गेटों को आपस में जोड़कर बहुत अधिक जटिल कार्य कर सकने वाले सर्किटों का निर्माण किया जा सकता है। जैसे गिनती करने वाले, संख्याओं को जोड़ने वाले, गुणा या भाग करने में समर्थ आदि प्रकार के सर्किट। यहां तक कि सम्पूर्ण डिजिटल कम्प्यूटर भी इन्हीं मूलभूत गेट से मिलकर ही बना है।

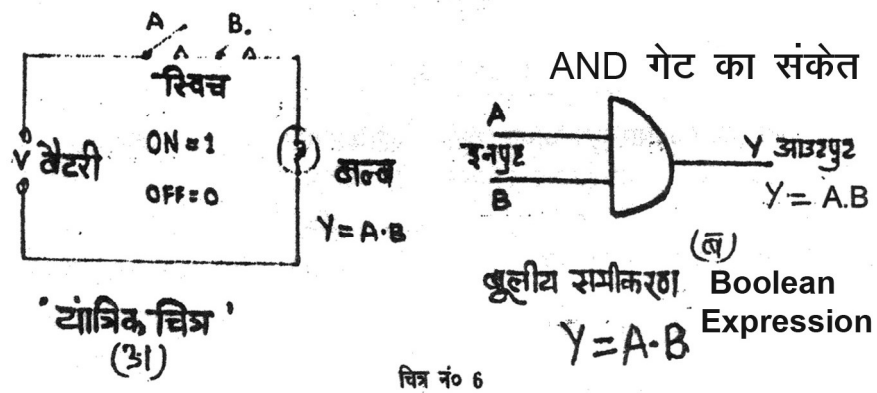
ट्रूथ टेबल (Truth Table) – यह क्रमिक रूप से लिखी गई एक सूची होती है जिसमें लॉजिक गेट के सभी संभव इनपुट लॉजिक संचय को तथा उनके द्वारा निर्धारित होने वाले आउटपुटों को वेरियेबल बाइनरी की दो अवस्थाओं के रूप में दिखाया जाता है। इन अवस्थाओं को 'लो' (0) या हाई (1) के रूप में लिखते हैं किसी भी लॉजिक गेट के कार्य को इस सूची से प्रतिपादित किया जा सकता है जब कभी भी किसी लॉजिक सर्किट का कार्य तुरन्त समझ में ना आये, उसे बुलीय समीकरण में सरलतापूर्वक न लिखा जा सके, तो उसकी ट्रूथ टेबल बनानी चाहिए। आगे आने वाले जटिल लॉजिक

[20] ST-52

सर्किटों को समझने के लिए भी ट्रुथ टेबिल बनाने और उससे निष्कर्ष निकालने की आदत डालनी चाहिए।

आइये, अब विभिन्न गेटों के विषय में जाना जाए।

एण्ड गेट (AND GATE)



नोट—बूलीय समीकरण में Y आउटपुट है तथा A व B इनपुट हैं। A.B के बीच बिन्दु (.) का चिन्ह 'AND' का प्रतीक है।

Output of an AND gate will be high (1) only when all the inputs are high (1).

अर्थात्

एण्ड गेट में Y आउटपुट तभी 'हाई' (1) होगी जब इसके सभी इनपुट AB (दो या दो से अधिक भी हो सकते हैं) आदि 'हाई' (1) होंगे।

नोट- डिजिटल में प्रायः हाई (1) = +5V DC तथा लो (0) = 0V माना जाता है।

इसकी ट्रुथ टेबिल निम्न प्रकार से बनती है।

TRUTH TABLE

A	B	Y	बुलीय समीकरण $Y=A.B$
0	0	0	
0	1	0	
1	0	0	
1	1	1	

POSSIBLE COMBINATION संभव संचय -

किसी भी गेट में दिए जा सकने वाले इनपुट सिगनल (1-0) के संभव संचयों को निम्न विधि द्वारा ज्ञात किए जा सकते हैं।

कुल इनपुट	संभव संचय	
1	$2^1 = 2$	1 or 0 only
2	$2^2 = 4$	
3	$2^3 = 8$	
4	$2^4 = 16$	
5	$2^5 = 32$	

इसी प्रकार किसी भी गेट में जितने भी इनपुट हो उतनी 2 की घात (पावर) मानकर संभव संचयों की संख्या ज्ञात हो जाती है। अर्थात् इनपुट की संख्या को **N** मान लें तो

संभव संचय (Possible Combination) = 2^N

नोट एक इनपुट के लिए संचय न होकर केवल 2 सिगनल ही प्रयोग होगा। जोकि 'हाई' (1) या लो (0) ही हो सकता है। यह केवल 'NOT गेट' पर ही लागू होता है। जिसके बारे में आगे बताया जायेगा।

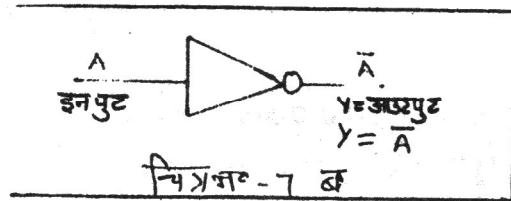
नॉट गेट (NOT GATE)

इस गेट में केवल एक ही इनपुट होता है और केवल एक ही आउटपुट होता है। इसकी विशेषता यह है कि इनपुट पर जो सिगनल दिया जाये, आउटपुट पर उसका ठीक विपरीत ही प्राप्त होता है। इसलिए इसे इनवर्टर भी कहा जाता है।

Out put of a NOT GATE will not be equal to input.

बुलीय समीकरण $Y = \bar{A}$

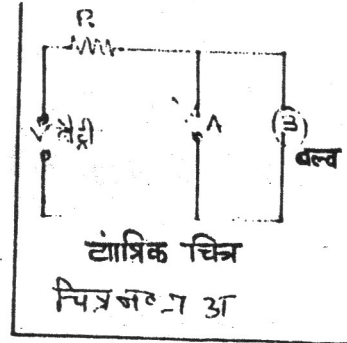
यहां A के ऊपर (—) चिन्ह लगाया गया है इसे 'बार' कहा जाता है। अर्थात् यदि A इनपुट है तो आउटपुट A का उल्टा होगा। इनपुट केवल एक ही होने के कारण इनपुट पर केवल 'हाई' (1) या 'लो' (0) सिगनल ही दिए जा सकते हैं अर्थात् वोल्ट या वोल्टता न होना।



TRUTH TABLE

A	Y
0	1
1	0

चित्र नं. 7 अ-ब



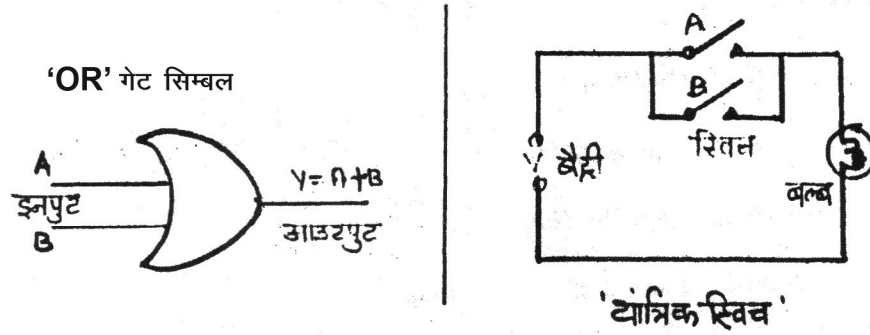
ऑर गेट (OR GATE)

Output of an OR gate will be high (1) When any one or more input are high (1) Or

In other words output of an OR gate will be low (0) only when all the inputs are low (0) अर्थात् OR गेट में यदि एक भी इनपुट हाई (1) हो जाए तो आउटपुट 'हाई' (1) हो जाता है।

बुलीय समीकरण $Y = A+B$

यहां (+) चिन्ह को OR कहा जाता है।



चित्र नं० 8 अ-ब

इस गेट में इनपुट दो से अधिक भी हो सकते हैं।

नोट-केवल NOT गेट को छोड़कर सभी गेट में दो या दो से अधिक इनपुट हो सकते हैं।

‘OR गेट’ की ट्रुथ टेबल

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

[24] ST-52

डिआईब्ड गेट (DRIVED GATES)

इससे पूर्व में बताये गये NOT गेट, AND गेट तथा OR गेट डिजिटल इलेक्ट्रानिक्स के आधारभूत गेट या फंडामेंटल गेट थे। इन्हीं आधार गेट के सहयोग से कुछ अन्य गेट बनाये गये हैं जिन्हें डिआईब्ड गेट कहा जाता है। यह 4 प्रकार के होते हैं –

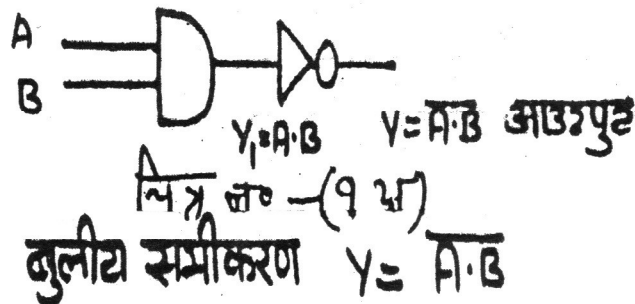
1. नैण्ड गेट (NAND gate)
2. नॉर गेट (NOR gate)
3. एक्स ऑर गेट (EX-OR gate)
4. एक्स नॉर गेट (EX-NOR gate)

'NAND' गेट

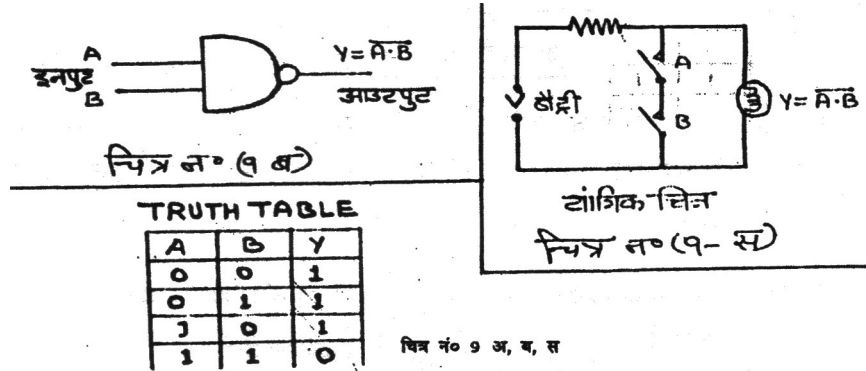
यह गेट 'AND' गेट और 'NOT' गेट से मिलकर बना है। कार्य प्रणाली में भी यह एण्ड गेट के आउटपुट का उल्टा परिणाम देता है।

Output of a NAND gate will be low (0) only when all the inputs are high (1).

AND गेट में हमने देखा कि जब सभी इनपुट हाई (1) होते हैं तभी आउटपुट हाई (1) होता था। परन्तु NAND गेट में उसका ठीक विपरीत होता है अर्थात् जब सभी इनपुट हाई (1) होंगे तो आउटपुट लो (0) होगा।



NAND गेट का संकेत

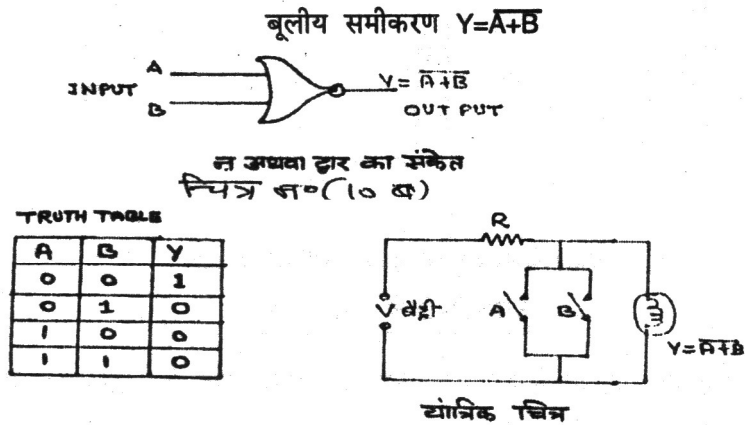


'NOR' गेट

यह गेट OR गेट तथा NOT गेट मिलने से बना है इस गेट का आउटपुट OR गेट के आउटपुट का ठीक उल्टा होता है।

Out put of a NOR gate will be high (1) only when all the inputs are low (0).
अर्थात्

OR गेट में हमने देखा कि जब सभी इनपुट 'लो' (0) होते हैं तभी आउटपुट 'लो' (0) होता था। जबकि सभी 'इनपुट' लो (0) होने पर NOR gate में उसका ठीक उल्टा अर्थात् हाई (1) होता है किसी भी एक या अधिक इनपुट के हाई (1) होने पर NOR गेट का आउटपुट 'लो' (0) हो जायेगा।



[26] ST-52

Ex-OR गेट (Exclusive OR gate)

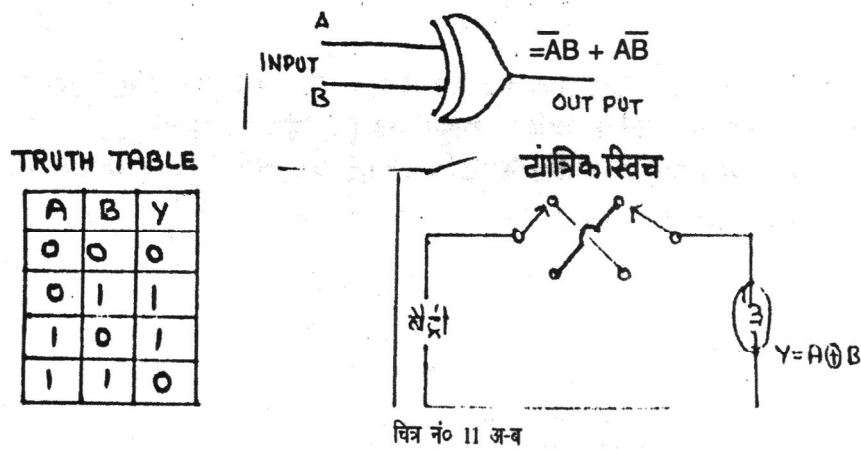
Ex-OR गेट अन्य गेटों की अपेक्षा अपनी एक अलग विशेषता रखता है इसमें चाहे जितनी इनपुट हो आउटपुट तभी हाई (1) होगा जब इनपुट पर विषम संख्या में हाई सिगनल होंगे।

Out put of a Ex OR gate will be high (1) only when the numbers of inputs are high (1) is in odd.

$$\text{बूलीय समीकरण } Y = A \oplus B = \bar{A}B + A\bar{B}$$

यहां $\oplus$ चिन्ह Ex-OR के लिए प्रयुक्त होता है।

2 INPUT EX-OR गेट का संकेत

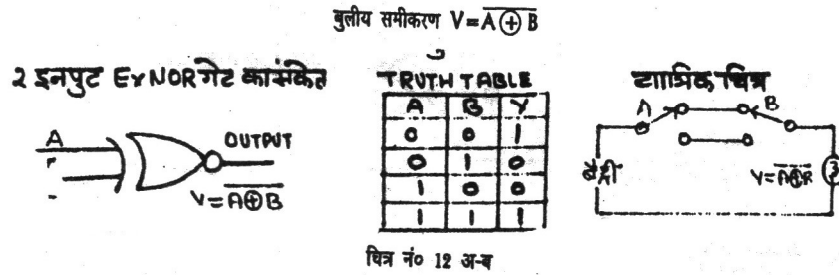


Ex-NOR गेट (Exclusive NOR gate)

यह गेट Ex-OR गेट में NOT गेट लगाकर बनाया जाता है। Ex-NOR गेट में आउटपुट हाई तब ही होगी जब

- (i) सभी इनपुट 'लो' (0) हो या,
- (ii) हाई इनपुट सम संख्या में हों।

उदाहरणार्थ—2 इनपुट Ex NOR गेट में आउटपुट तब ही 'हाई' होगी जब दोनों इनपुट एक सी हों। अतः इसका उपयोग कम्पेरेटर में किया जाता है।



बुलीय बीज गणित (BOOLEAN ALGEBRA)

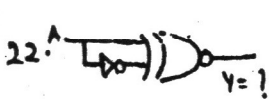
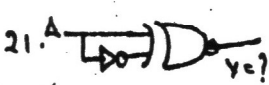
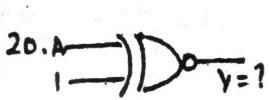
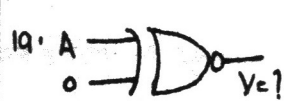
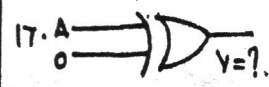
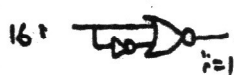
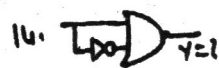
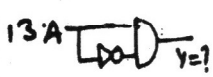
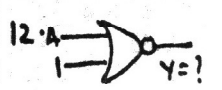
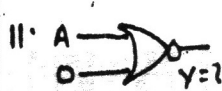
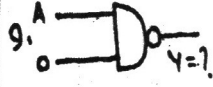
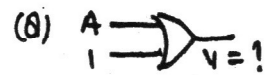
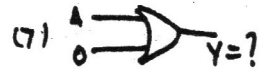
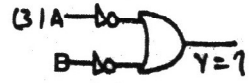
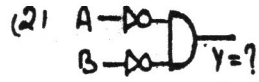
सूत्र—

$$\begin{aligned}
 A+B &= B+A \\
 A+(B+C) &= (A+B)+C \\
 A.(B+C) &= AB+AC \\
 A+0 &= A \\
 A+1 &= 1 \\
 A+A &= A \\
 A+\bar{A} &= 1 \\
 \bar{\bar{A}} &= A \\
 A+AB &= A \\
 A+\bar{A}B &= A+B
 \end{aligned}$$

$$\begin{aligned}
 A.B &= B.A. \\
 A.(B.C) &= (A.B).C \\
 A+B.C &= (A+B).(A+C) \\
 A.1 &= A \\
 A.0 &= 0 \\
 A.A &= A \\
 A.\bar{A} &= 0 \\
 A.(\bar{A}+B) &= AB \\
 \text{डिमोरगन की प्रमेय} \\
 \text{De'Morgan' Theorem} \\
 \overline{A+B} &= \bar{A}.\bar{B} \\
 \overline{AB} &= \bar{A}+\bar{B}
 \end{aligned}$$

[28] ST-52

निम्न की आउटपुट ज्ञात करें-



चित्र नं० 13

उत्तर- (1) $\overline{A.B}$, (2) $\overline{A.B}$, (3) $\overline{A+B}$, (4) $\overline{A+B}$, (5) 0, (6) A, (7) A, (8) 1, (9) 1, (10) $\overline{A}$, (11) $\overline{A}$, (12) 0, (13) 0, (14) 1, (15) 1, (16) 0, (17) A, (18) $\overline{A}$, (19) $\overline{A}$, (20) A, (21) 0, (22) 0

‘डिजिटल कोड’ (DIGITAL CODES)

जैसा कि हमने पिछले अध्यायों में देखा है कि संख्याओं को बाइनरी सिस्टम या अन्य किसी भी सुविधाजनक सिस्टम में लिखा जा सकता है। रोजमर्रा के काम-काज में डेसीमल सिस्टम का ही प्रयोग होता है। परन्तु कम्प्यूटर या अन्य डिजिटल उपस्कर में बाइनरी संख्याओं की आवश्यकता पड़ती है। यद्यपि शुद्ध बाइनरी संख्यायें डिजिटल उपस्कर के लिए सुविधाजनक हैं फिर भी मनुष्य और मशीन के बीच संचार व्यवस्था अथवा परस्पर सामंजस्य के लिए यह अपेक्षाकृत काफी मुश्किल है। इसलिए डेसीमल सिस्टम (Digital system) और बाइनरी सिस्टम (Binary system) के समायोजन से विशेष संकेत बनाये गये। जिन्हें डिजिटल कोड कहा जाता है।

बाइनरी कोडिड डेसीमल (Binary Coded Decimal) B.C.D. (8421) CODE:

इस कोड में 4 बिट का प्रयोग किया जाता है अर्थात् किसी भी डेसीमल नम्बर को बाइनरी सिस्टम में लिखने के लिए प्रत्येक अंक को अलग-अलग 4 बिट बाइनरी में लिखते हैं।

उदाहरण – 8421 को बाइनरी सिस्टम में लिखें।

4 बिट बाइनरी में	8	=	1000
	4	=	0100
	2	=	0010
	1	=	0001

अतः 8421 = 1000 0100 0010 0001 हुआ

[30] ST-52

ASCII-CODE

Digit No ↓ BINARY CODE	X ₇	X ₆	X ₅	X ₄	X ₃	X ₂	X ₁	X ₀				
	0	1	0	0	1	1	0	0	1	0	1	1
American Standard Code for Information Interchange	SP	0	@	P	→	†	0	0	0	0		
	!	1	A	a	a	q	0	0	0	1		
	,"	2	B	R	b	h	0	0	1	0		
	#	3	C	S	c	z	0	0	1	1		
	\$	4	D	T	d	t	0	1	0	0		
	%	5	E	U	e	u	0	1	0	1		
	&	6	F	V	f	v	0	1	1	0		
	,	7	G	W	g	w	0	1	1	1		
	(	8	H	X	h	x	1	0	0	0		
	)	9	I	Y	i	y	1	0	0	1		
	*	:	J	Z	j	z	1	0	1	0		
	+	;	K		k		1	0	1	1		
	,	<	L		l		1	1	0	0		
	-	=	M		m		1	1	0	1		
	.	>	N		n		1	1	1	0		
	/	?	O		o		1	1	1	1		

इसका उपयोग इनपुट-आउटपुट विधियों में किया जाता है। इसका प्रयोग अंकगणित क्रियाओं में नहीं किया जाता।

इसमें सात बिट का समूह होता है जैसा उपरोक्त तालिका में दर्शाया गया है अतः ASCII कोड को 7 बिट कोड भी कहते हैं। इस कोड में यदि STTC/GZB लिखना पड़े तो पहले अक्षर के ऊपर लिखी बाइनरी X_6, X_5, X_4 तथा फिर सामने लिखी बाइनरी X_2, X_1, X_0 , लिखेंगे जैसे—

$S = 1010011, T = 1010100, C = 1000011, / = 0101111$

$G = 1000111, Z = 1011010, B = 1000010.$

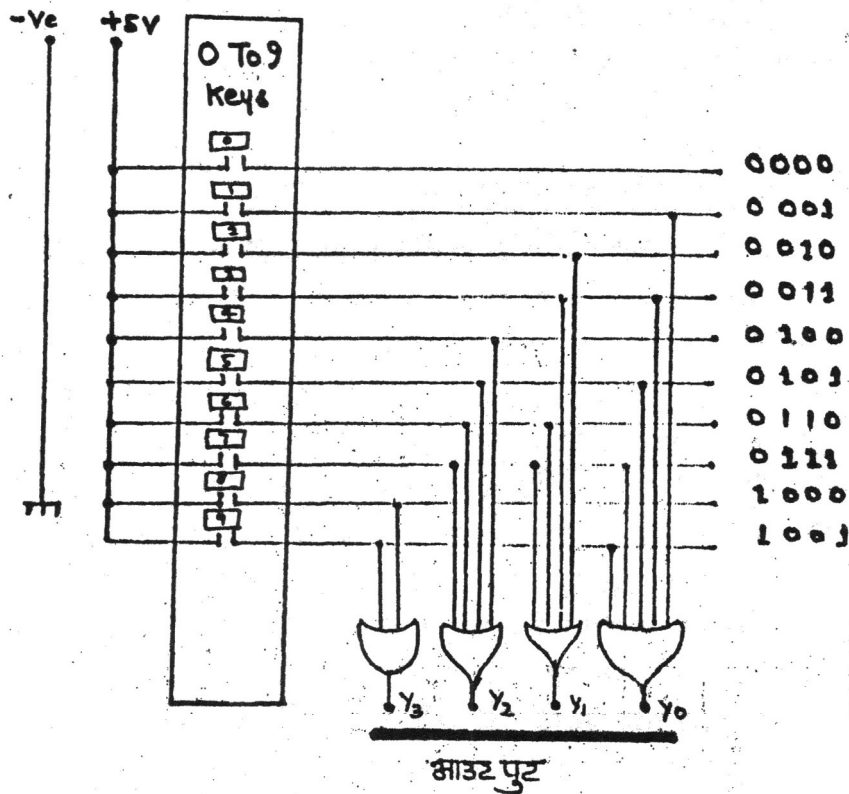
अतः STTC/GZB=(1010011-1010100-1010100-1000011-0101111-1000111-1011010-1000010)

ASCII Code होगा।

कोडर (CODER)

आम प्रचलित परस्पर संचार व्यवस्था में शब्दों व अंकों का ही प्रयोग होता है परन्तु डिजिटल में कार्य करने के लिए इनका सीधा प्रयोग संभव नहीं हो पाता। अतः डिजिटल में कार्य करने के लिए कुछ संकेत प्रत्येक शब्द व अंक आदि के लिए बनाने पड़ते हैं ताकि डिजिटल विधियां इन्हें अपनी सांकेतिक भाषा में समझ लें। इन शब्दों और अंकों को सांकेतिक रूप देने के लिए डिजाइन की गई विधियों को कोडर (Coder) कहते हैं। विभिन्न सांकेतिक सिस्टम के लिए भिन्न-भिन्न प्रकार के कोडर डिजाइन किये जाते हैं। जिनका वर्णन आगे किया जायेगा।

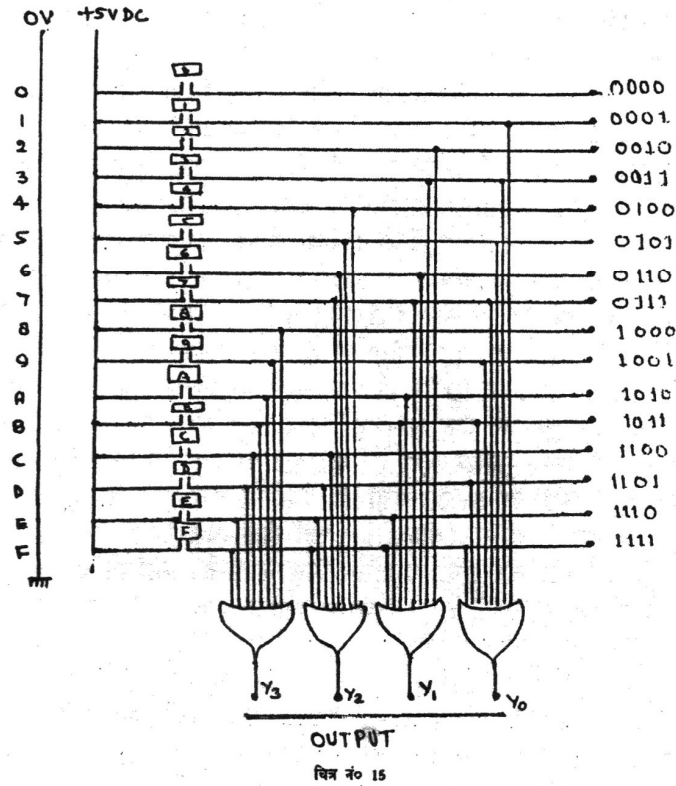
डेसीमल टू बाइनरी



इस कोडर में गणितीय कार्यों में प्रयुक्त होने वाले अंकों (0, 1, 2, 3, 4, 5, 6, 7, 8, 9,) को सांकेतिक भाषा में परिवर्तित किया जाता है। ये डेसीमल नम्बर इस कोडर की सहायता से 4 बिट बाइनरी में बदल दिये जाते हैं जैसा कि चित्र में दिखाया गया है। इसे इस प्रकार समझें। यदि 6 नं. डेसीमल को 4 बिट बाइनरी में बदलना हो तो (चित्र 14 में 6 नं. 'की' दबाई है) जैसे ही हम 6 नं. पुश बटन दबायेंगे +5VDC लाइन पर आ जायेगी। अर्थात् 6 नं. की स्थिति डिजिटल में हाई हो जायेगी। अब देखें इस 6 नं. की लाइन से दो तारें जुड़ी हैं। एक तार OR गेट Y_1 के इनपुट में पहुंची है। (OR गेट के सिद्धान्त के अनुसार एक भी हाई (1) इनपुट आ जाये तो आउटपुट हाई हो जाता है) दूसरी तार OR गेट Y_2 के इनपुट में पहुंची है। अब चूंकि 6 नं. लाइन से अन्य कोई तार OR गेट Y_0 या Y_3 पर नहीं गया है। 6 नं. के अतिरिक्त शेष पुश बटन भी ऑफ स्थिति में हैं। इसलिए सभी लाइन भी लो (0) हैं। अतः हाई (1) आउटपुट केवल Y_1 तथा Y_2 पर ही मिलेगा। इस प्रकार $Y_3=0$, $Y_2=1$, $Y_1=1$ तथा $Y_0=0$ हुआ अर्थात् 0110 आउटपुट हुआ जो कि डेसीमल 6 की ही बाइनरी प्रारूप है। इसी प्रकार अन्य कोडर भी बनते हैं।

हैक्साडेसीमल से बाइनरी (Hexadecimal to Binary Coder) :-

इस coder में गणितीय कार्यों में प्रयुक्त होने वाले अंकों (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) के अतिरिक्त A, B, C, D और E, F को सांकेतिक भाषा में परिवर्तित किया जाता है ये संख्यायें इस कोडर की सहायता से 4 बिट बाइनरी में बदल दिये जाते हैं जैसा कि आगे चित्र 15 में भी दिखाया गया है अब दिये चित्र के अनुसार समझें कि यदि हैक्साडेसीमल संख्या E को 4 बिट बाइनरी में बदलना हो तो हम पुशबटन D को जैसे ही दबाते हैं तो +5VDC अर्थात् High(1) स्थिति D लाइन पर आ जाती है। D लाइन से तीन (3) तारें जुड़ी हुई हैं। एक तार OR गेट Y_3 पर, दूसरी तार OR गेट Y_2 पर तथा तीसरी तार OR गेट Y_1 पर जुड़ी हुई हैं D लाइन से अन्य कोई तार नहीं जुड़ी है तथा D पुश बटन के अतिरिक्त शेष सभी पुशबटन सामान्य होने से उनकी लाइनों पर भी कोई वोल्टता नहीं पहुंच रही है अर्थात् उनकी स्थिति लो '0' है। इस प्रकार $Y_3=1$, $Y_2=1$, $Y_1=0$ तथा $Y_0=1$ आउटपुट बना है यानि के 1101 जो कि हैक्साडेसीमल के लिए मान्य है। इसी प्रकार अन्य हैक्सा डेसीमल संख्या भी बाइनरी में परिवर्तित होती है। देखें चित्र नं. 15।



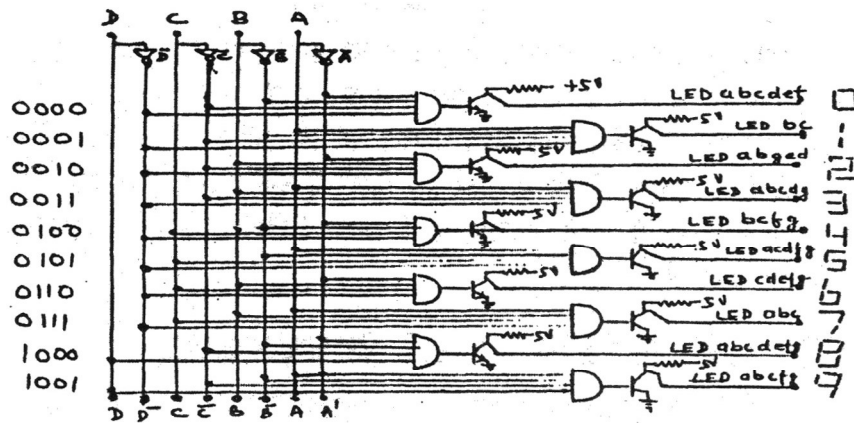
डिकोडर 'विकूटित्र' (DECODER)

डिजिटल सिस्टम (विधियों) में जब कोडर हमारी मांग के अनुसार फीड किये गये संदेशों की बाइनरी कोड में परिवर्तित कर देता है तो उसके पश्चात् यह देखने के लिए कि हमने जो संकेत फीड किये वे सही गये कि नहीं या हमारे द्वारा फीड किये गये संदेशों के प्रत्युत्तर में जो भी उत्तर हमें मिलने हैं उन्हें बाइनरी से हमारी समझ आ सकने वाली भाषा में बताने के लिए जो डिजिटल विधियां प्रयोग में आती हैं उसे डीकोडर कहते हैं।

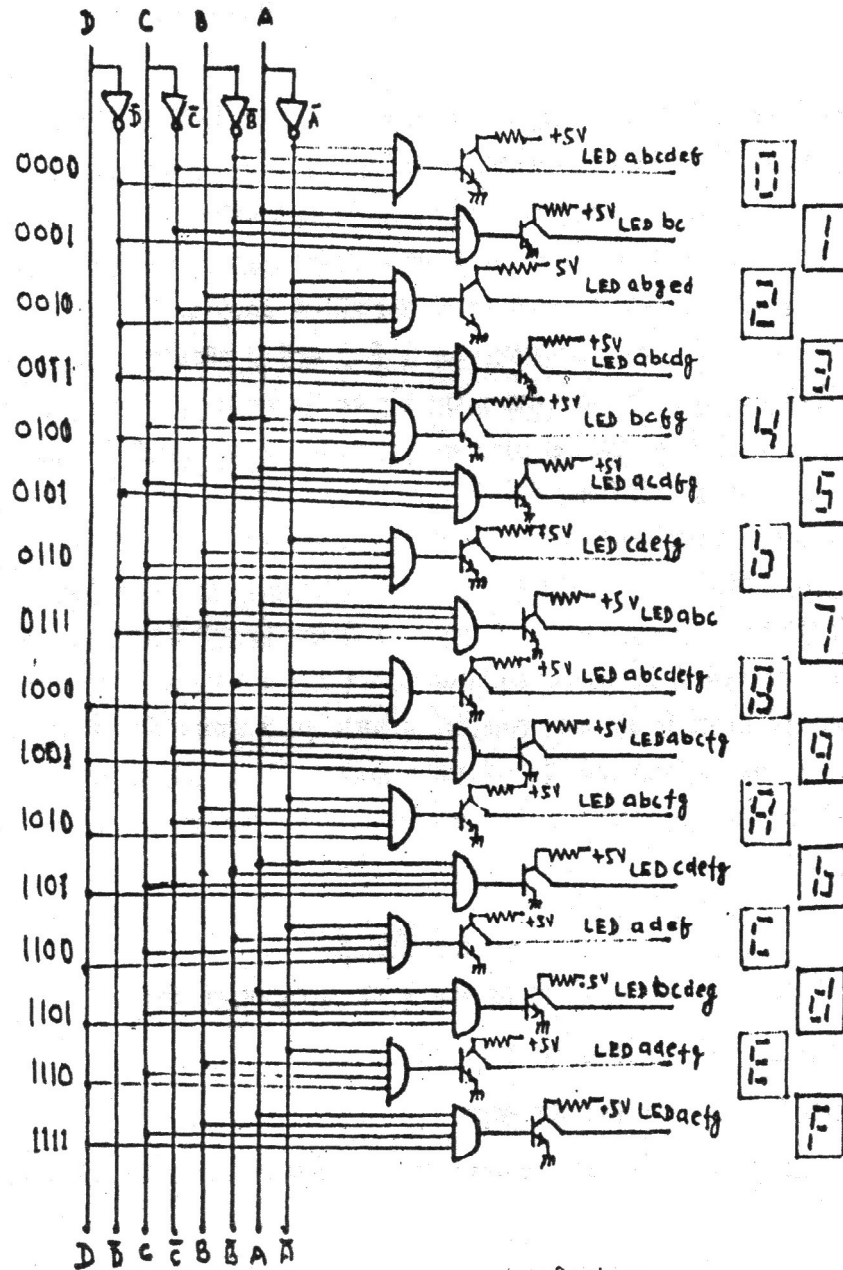
बाइनरी से डेसीमल डीकोडर (Binary To Decimal Decoder)

यह डीकोडर, कोडर द्वारा डेसीमल से बाइनरी में बदले गये संकेतों को वापिस डेसीमल में परिवर्तित कर 7 सेगमेंट डिस्प्ले अथवा प्रिन्टर द्वारा प्रस्तुत करता है। आगे दिए गए चित्र 16 के अनुसार इसकी कार्य सिस्टम समझें। टर्मिनल ABCD के

साथ अलग-अलग NOT गेट लगाये गये हैं जिससे इन टर्मिनलों पर आने वाले इनपुट हमें दो रूप में प्राप्त हैं। बार (–) वाली लाइनों पर साधारण लाइनों से ठीक विपरीत सिगनल होते हैं। आगे 4 इनपुट के AND गेट का प्रयोग किया गया है जिसका आउटपुट नियमानुसार तभी हाई (1) हो सकता है जब चारों इनपुट पर हाई (1) हो। अब उदाहरणार्थ हमें 7 नं. को चैक करना है तो देखें कि जब ABCD टर्मिनलों पर $D=0, C=1, B=1$ और $A=1$ (0111) इनपुट मिलती है तो $\bar{A}$ पर 0, $\bar{B}$ पर 0, $\bar{C}$ पर 0 तथा $\bar{D}$ पर 1 मिलती है। इन आठों लाइनों को आगे ले लिया गया है। अब 7 नं. AND पर चार इनपुट कनेक्शन क्रमशः A, B, C तथा $\bar{D}$ पर लगी हुई है। टर्मिनल A, B, C पर हाई (1) सिगनल पहले से है तथा $\bar{D}$ पर NOT गेट के कारण हाई (1) सिगनल हो गया है। अर्थात् चारों इनपुट हाई (1) हो गये हैं। इसलिए 7 No. AND गेट के आउटपुट पर सिगनल हाई (1) होगा। 7 No. AND गेट के आउटपुट पर एक NPN ट्रांजिस्टर लगा हुआ है। जिसका एमीटर अर्थ तथा बेस AND गेट के आउटपुट पर जुड़ा है। कलैक्टर पर एक प्रतिरोध (Resistance) जुड़ी है। जिसके दूसरे सिरे पर 5VDC जुड़ी है। बेस पर जब तक स्थिति लो '0' है तब तक बेस -ve होने से ट्रांजिस्टर क्रियाशील नहीं होता, बेस पर स्थिति हाई (1) अथवा +ve होते ही ट्रांजिस्टर को उचित बायसिंग मिल जाती है। इसलिए कलैक्टर से एमीटर की ओर करंट तेजी से बहती है। इस कारण एमीटर अर्थ की स्थिति कलैक्टर टर्मिनल तक पहुंच जाती है जो कि प्रतिरोध के साथ-साथ 7 सेगमेंट डिस्प्ले की LED abc से जुड़ा है सभी संबंधित LED abc के दूसरे सिरों पर +5VDC पहले से जुड़ी है जो ट्रांजिस्टर के कलैक्टर के माध्यम से अर्थ मिलते ही LED abc का सर्किट पूरा हो जाता है और वे जलने लगती है जिससे 7 सेगमेंट डिस्प्ले 7 अंक बना दिखाई देता है ऐसा ही अन्य डेसीमल संख्याओं के लिए भी किया जाता है।



चित्र नं० 16



चित्र नं० 17

[36] ST-52

बाइनरी से हैक्साडेसीमल डिकोडर (BINARY TO HEXADECIMAL DECODER) :

यह डिकोडर, कोडर द्वारा हैक्साडेसीमल से बाइनरी में बदले गये संकेतों को फिर से हैक्साडेसीमल में बदलकर 7 सैगमेन्ट डिसप्ले या प्रिन्टर द्वारा प्रस्तुत करता है उपरोक्त चित्र 17 के अनुसार इसकी कार्य सिस्टम समझें। इसमें प्रत्येक संकेत के लिए AND गेट का प्रयोग किया गया है जोकि 4 इनपुट का है। AND गेट सभी इनपुट हाई (1) होने पर ही 'आउटपुट' हाई (1) देता है। अब टर्मिनल ABCD पर मान लें कि इन पर D=0, C=1, B=1 और A=0 सिगनल आ रहे हैं तो चारों ही टर्मिनलों पर NOT गेट लगे होने से एक-एक लाइन पर पृथक-पृथक चार और नए सिगनल बन गए। जैसे D=1, C=0, B=0 और A=1। अब देखें कि 0110 यह हैक्साडेसीमल 6 का बाइनरी है। यहां एक 6 नं. AND गेट ही ऐसी स्थिति में है कि जिसके चारों इनपुट हाई (1) हो गये हैं। जिससे इस गेट के आउटपुट पर भी हाई (1) सिगनल मिला, जिससे आगे सर्किट में लगा बफर ड्राइवर 'ट्रांजिस्टर' क्रियाशील हो जाता है। जिससे एमीटर का अर्थ कलैक्टर टर्मिनल से होते हुए LED cdfg को मिल जाता है। 7 सैगमेन्ट डिसप्ले में सभी LED के दूसरे सिरों पर पहले से ही +5VDC कनेक्ट कर दिया जाता है। अब जिस-जिस LED को अर्थ मिलता है वह जलने लगती है इस प्रकार हमें 7 सैगमेन्ट डिसप्ले में डेसीमल का 6 अंक बना नजर आता है। इसी प्रकार कोडर जो भी आउटपुट इस विधि के इनपुट को देगा उसे हमारी समझ में आ सकने वाली भाषा में यह डीकोडर प्रस्तुत कर देते हैं।

PARITY

जैसा कि हम पहले भी पढ़ चुके हैं कि BIT = बाइनरी डिजिट

निब्ल (Nibble) = 4 बिट का समूह

बाइट (Byte) = 8 बिट का समूह

वर्ड (Word) = इन बिट का समूह जो एक अक्षर को प्रतिरूपित करें एक शब्द कहलाता है।

यहां यह जानना आवश्यक है कि जहां 8 बिट का कोई अपना विशेष महत्व न हो यह बाइट तो हो सकता है परन्तु शब्द नहीं जैसे एक संख्या लें 563 तो इसमें 6 का अपना स्वतंत्र अक्षर या महत्व नहीं है। यह बाइट ही कहलायेगा शब्द नहीं। परन्तु यह

यदि अकेला होता तो इसे शब्द भी कह सकते थे और बाइट भी।

पैरिटी (Parity) – किसी अक्षर में उपस्थित 'हाई' (1) की संख्या को दर्शाती है।

उदाहरणार्थ $(563)_{10} =$ BCD कोड में

$= 010101100011$ इसमें 6 बार हाई (1) आया है अर्थात्

6 सम संख्या है। अतः इसे सम पैरिटी शब्द (Even Parity Word) कहा जायेगा।

इसी प्रकार यदि $(269)_{10}$ को 4 बिट बाइनरी में देखें तो बनता है 001001101001, इसमें 5 बार हाई (1) आया अर्थात् 5 विषम संख्या है। अतः इसे विषम पैरिटी शब्द (odd Parity word) कहा जायेगा।

पैरिटी परीक्षक/पैरिटी टेस्टर (Parity Tester) – पैरिटी परीक्षक से हमें यह ज्ञात हो जाता है कि प्रयुक्त होने वाला शब्द विषम पैरिटी शब्द है या सम पैरिटी शब्द है। यह दो प्रकार से बनाया किया जाता है।

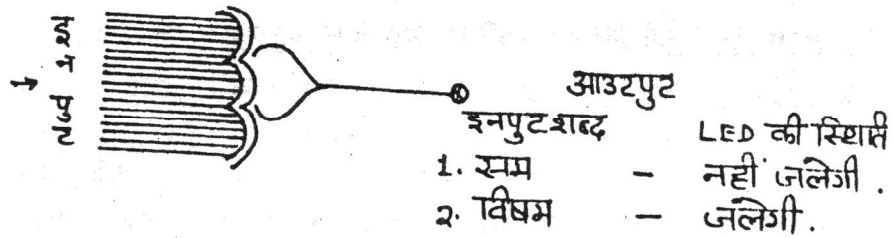
विषम पैरिटी परीक्षक/ODD पैरिटी टेस्टर (Odd Parity Tester) – जो कम्प्यूटर विषम पैरिटी शब्द के लिए बनाए जाते हैं उनमें सम पैरिटी शब्द फीड किये जायें तो एक (1) हाई और जुड़ जाता है।

उदाहरणार्थ – फीड किया गया शब्द सम पैरिटी में है 0101 0110 0011 तो विषम पैरिटी शब्द के लिए बने कम्प्यूटर उसे $1; 0\ 10\ 10\ 11\ 000\ 11;$ प्रस्तुत कर देता है विषम पैरिटी शब्द 0100, 0110 1001 में 1 को कोई जोड़ नहीं होता और वह $0; 0010\ 0110\ 1001;$ रहता है।

पहले हमने EX-OR गेट में पढ़ा है कि इनपुट 1, विषम संख्या में होने पर ही आउटपुट (1) हाई होता है। जैसा कि चित्र 18 में दिखाया गया है।

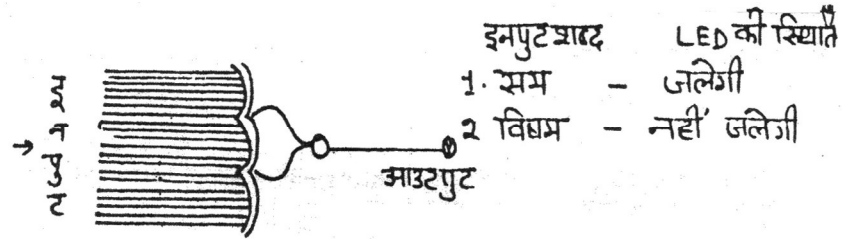
If output of an EX-OR gate is high (1), the input word is of odd Parity word, if out put is low the input word has been even parity.

Ex-OR गेट का उपयोग पैरिटी टेस्टर की भांति



चित्र नं० 18

Ex-NOR गेट का उपयोग पैरिटी टेस्टर के रूप में -



चित्र नं० 19

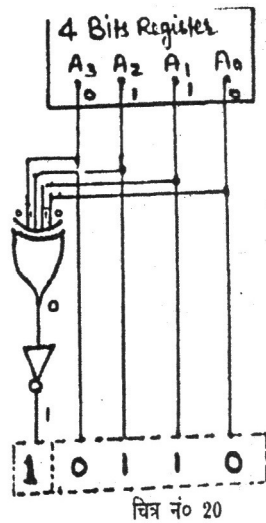
EX-OR गेट के प्रयोग में हमने देखा कि यह गेट विषम पैरिटी परीक्षक/पेरिटी टेस्टर के रूप में कार्य कर सकता है। उसी प्रकार Ex-NOR गेट भी सम पैरिटी परीक्षक के रूप में कार्य कर सकते हैं।

इसमें आउटपुट तभी हाई (1) होगा जब इनपुट हाई (1) सम संख्या में होंगे। यदि हमें आउटपुट 'हाई' मिले तो इनपुट शब्द सम पैरिटी होगा अन्यथा विषय पैरिटी शब्द होगा। देखें चित्र नं. 19

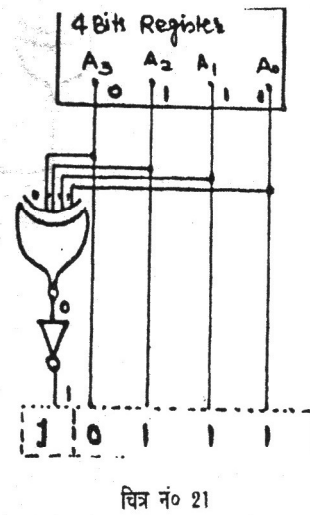
If the output of an Ex NOR gate is high (1), the input word is of even parity and if out put is low (0) the input word is of odd parity.

पेरिटी जेनरेटर : Parity Generator

विषम :- इसमें जब शब्द सम पैरिटी, का होता है तो निम्नचित्र 20 के अनुसार एक Ex-OR गेट तथा NOT गेट के द्वारा एक नई हाई (1) पैरिटी उत्पन्न हो जाती है। यह सम पैरिटी में जुड़ जाती है उसे विषम पैरिटी बना देती है। जब शब्द विषम पैरिटी का होता है तो नई हाई (1) पैरिटी उत्पन्न नहीं होती तथा शब्द विषम पैरिटी का ही रह जाता है।



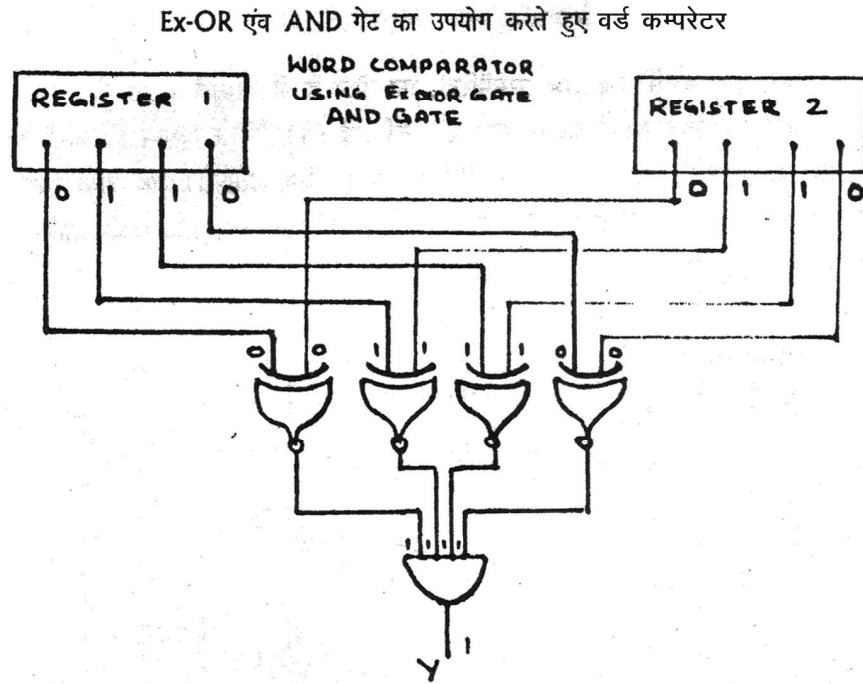
सम-EVEN :->



सम:- इसमें जब शब्द विषम पैरिटी का होता है तो चित्र 21 के अनुसार एक Ex-NOR तथा NOT गेट के द्वारा एक नई हाई (1) पैरिटी उत्पन्न होकर उसे सम पैरिटी शब्द बना देती है तथा सम पैरिटी शब्द होने पर नई हाई (1) उत्पन्न नहीं होती तथा शब्द सम पैरिटी का ही रह जाता है।

वर्ड कम्परेटर (WORD COMPARATOR)

यह एक ऐसी विधि होती है जो किन्हीं दो शब्दों की तुलना करके यह बताता है कि दोनों शब्द समान हैं अथवा नहीं। 4 बिट के बाइनरी शब्द में 4 Ex-NOR गेट और



चित्र नं० 22

एक AND गेट लगेगा। जितनी बिट का शब्द होगा उतने ही Ex-NOR गेट लगते हैं परन्तु AND गेट एक ही लगता है देखें चित्र नं. 22। शब्द कम्परेटर यदि Ex-OR गेट से बनाना हो तो Ex-NOR के स्थान पर Ex-OR गेटों तथा AND गेट की जगह OR गेट लगा दिया जाता है।

Ex-NOR गेट + AND गेट से बने शब्द कम्परेटर में –

If output is high (1) Both word all same अर्थात् Ex-NOR तथा AND गेट से बने शब्द कम्परेटर में यदि आउटपुट हाई (1) हो जाये तो दोनों शब्द समान होंगे।

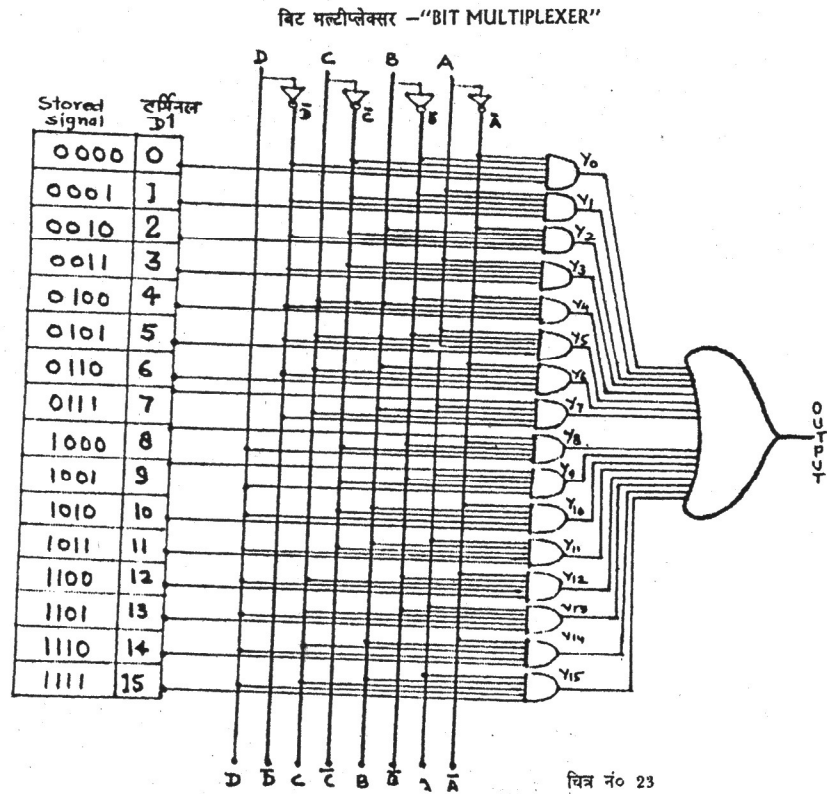
Ex-NOR तथा OR गेट से बने शब्द कम्परेटर में भी परिणाम ठीक वहीं होंगे जो कि Ex-NOR तथा AND गेट से बने शब्द कम्परेटर में होते हैं।

मल्टीप्लेक्सर (मक्स) MULTIPLEXER (MUX)

Multiplexer means many into one (एक में बहुत)

जो बहुत सारे सिगनल को एक साथ इकट्ठा कर लेता है और कोई भी सिगनल किसी भी दूसरे सिगनल को डिस्टर्ब भी नहीं करता तथा आवश्यकता पड़ने पर जिससे वांछित सिगनल लिया जा सके उसे मल्टीप्लेक्सर कहते हैं।

बिट मल्टीप्लेक्सर :- आगे दिये गये चित्र 23 में एक ऐसा मल्टीप्लेक्सर दिखाया गया है जिसमें 16 प्रकार के संकेत बिट के रूप में इकट्ठे कर दिये गए हैं बिट मल्टीप्लेक्सर एक ऐसा साधन है जिसमें बाइनरी बिट के रूप में संकेत इकट्ठे कर लिये जाते हैं आवश्यकता पड़ने पर इनमें से जो संकेत प्राप्त करना हो, वह प्राप्त हो जाता है। देखें हमने 0001 संकेत प्राप्त करना है तो संबंधित बटन दबाने पर DCBA पर क्रमशः 0001 प्राप्त होंगे जो अलग-अलग NOT गेट से $D = 1$, $C = 1$, $B = 1$ तथा $A = 0$ भी बनेंगे।

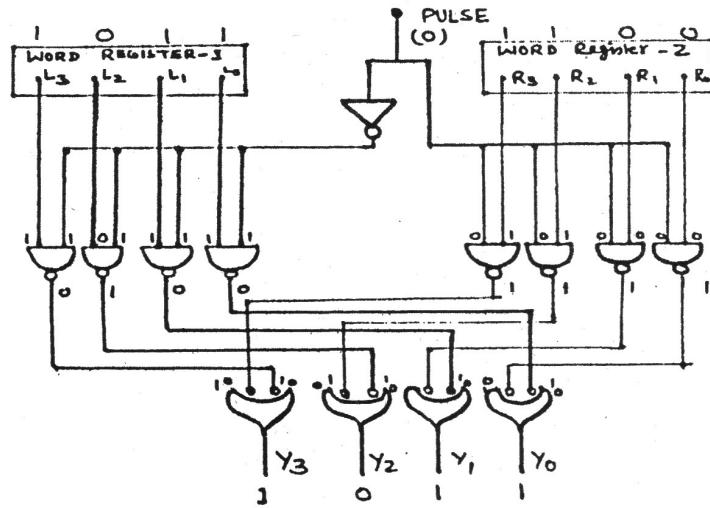


चित्र नं० 23

अब AND गेट Y_1 को देखें। जिसके कनेक्शन A,B,C,D पर हो रहे हैं अर्थात् हमें इस AND गेट पर चार हाई (1) इनपुट प्राप्त हो रहे हैं। शेष 5वीं इनपुट हमें D1 टर्मिनल से प्राप्त होती है। जहां 0001 सिगनल एकत्र है AND गेट के नियम से शेष चारों इनपुट हाई (1) होने से जो 5वीं इनपुट होगी वही आउटपुट हमें मिलेगा। अर्थात् हमें आउटपुट (AND गेट से) वही मिलेगी जो D1 टर्मिनल पर होगी। शेष सभी AND गेट की स्थिति इस प्रकार रहेगी कि उनका आउटपुट 'लो' (0) रहेगा। इन सभी AND गेट का आउटपुट एक OR गेट के इनपुट पर जायेंगी और OR गेट एक भी इनपुट (1) देने के कारण D_1 से होकर Y_1 आने वाले सिगनल OR गेट के आउटपुट पर प्राप्त होते रहेंगे। इसी प्रकार इस मल्टीप्लेक्सर से अन्य संकेत जो एकत्र हैं प्राप्त किये जा सकते हैं।

वर्ड मल्टीप्लेक्सर : इस शब्द मल्टीप्लेक्सर में शब्दों को एकत्र किया जाता है जैसा कि आगे चित्र 24 में दिखाया गया है। इसमें जब पल्स (0) लो होती है तो शब्द 1 हमें आउटपुट में प्राप्त होता है। जब पल्स 'हाई' (1) होती है तो हमें 2 शब्द आउटपुट में मिलेगा। पल्स 'लो' (0) देने पर दिखाया गया है कि किस प्रकार हमें Y_3, Y_2, Y_1 , तथा Y_0 पर आउटपुट क्रमशः 1011 मिलती है। जो कि शब्द रजिस्टर 1 में एकत्र है। इसी प्रकार यदि पल्स 'हाई' (1) दी जाती है स्थितियां बदल जाती हैं फिर हमें आउटपुट वह मिलता है जो कि शब्द रजिस्टर 2 में एकत्र होता है।

WORD MULTIPLEXER



चित्र नं० 24

‘अर्थमेटिक लॉजिक यूनिट’

ARITHMETIC LOGIC UNIT

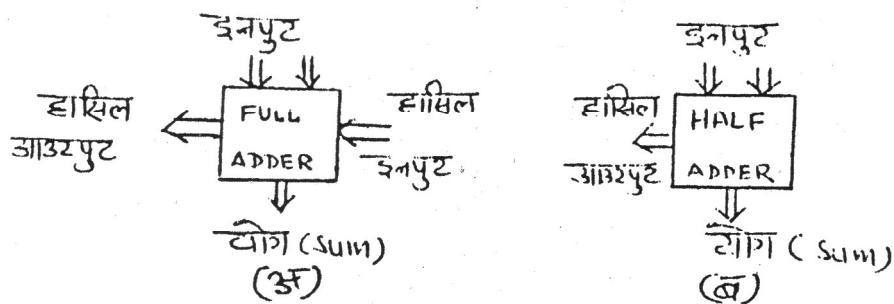
इस यूनिट में सभी गणितीय क्रियायें होती हैं। जैसे जोड़, घटा, गुणा, भाग आदि। सर्वप्रथम डिजिटल लॉजिक में जोड़ किस प्रकार होता है यह देखते हैं। जोड़ करने वाला योगक (Adder) दो प्रकार का होता है।

1. हाफ एडर (Half Adder)
2. फुल एडर (Full Adder)

जब हम दो बाइनरी नम्बरों को जोड़ते हैं तो हम सबसे दायें कॉलम (least significant column) से शुरू करते हैं अर्थात् हम दो बिटों को जोड़ते हैं जिसमें हासिल आने की सम्भावना होती है। इस प्रकार के जोड़ के लिए प्रयुक्त सर्किट को हाफ एडर कहते हैं।

हाफ एडर (Half Adder) में इनपुट में कोई हासिल नहीं होता इसलिए यह केवल प्रथम अंक को जोड़ने का कार्य करता है। चित्र 25(अ) फुल एडर (Full Adder) में इनपुट में हासिल भी होता है। इसलिए यह सभी अंकों को जोड़ने का कार्य करता है। चित्र 25(ब) हर एडर से 2 आउटपुट निकलती है। एक को योग (sum) तथा दूसरे को हासिल (carry) कहते हैं। केवल योग कार्य के लिए हाफ एडर प्रथम अंक को योग करता है तथा फुल एडर अन्य अंकों को हासिल सहित योग करता है।

योग करने के लिए ट्रुथ टेबिल बना लेते हैं जिसमें सभी अवस्थाएं होती हैं। पहला इनपुट, दूसरा इनपुट, योग और हासिल चारों मान रखकर पहले इनपुट 1 और 2 के योग से (योग को आउटपुट मानकर) तुलना की जाती है कि ये फलन (function)



चित्र नं० 25

[44] ST-52

किस-किस गेट में संभव है। यदि संभावना एक बार में एक से अधिक गेट की बनती है तो बुलीय बीजगणित (Boolean Algebra) या कारनाफ मानचित्र (k-map) की सहायता से सरलतम समीकरण बना ली जाती है फिर जिसका लॉजिक सर्किट बनाया जाता है।

ट्रुथ टेबिल

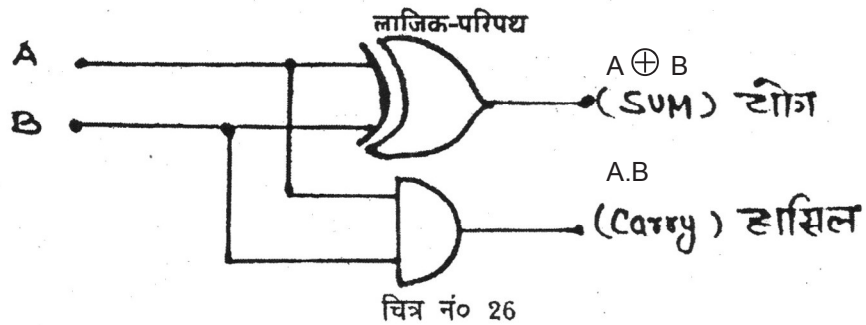
IP1	IP2	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

हाफ एडर (Half Adder)

IP_1 और IP_2 के योग (sum) से तुलना करने पर ज्ञात होता है कि यह स्थिति 2 इनपुट वाले Ex-OR गेट में होगी।

IP_1 और IP_2 की हासिल (carry) से तुलना करने पर ज्ञात होता है कि यह स्थिति 2 इनपुट वाले AND गेट में होगी।

इन दोनों स्थितियों का संयुक्त रूप चित्र 26 में देखें।



फुल एडर (Full Adder) -**ट्रुथ टेबिल**

IP1	IP2	IP3 Carry	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

IP₁, IP₂, IP₃ की sum से तुलना करने पर ज्ञात होता है कि यह स्थिति Ex-OR गेट पर लागू होती है।

IP₁, IP₂, IP₃ की carry से तुलना करने पर ज्ञात होता है कि यह स्थिति भिन्न होने पर किसी एक गेट पर लागू नहीं होती अतः यहां बूलीय बीजगणित या कारनाफ मानचित्र की सहायता से लॉजिक सर्किट बनाना चाहिए।

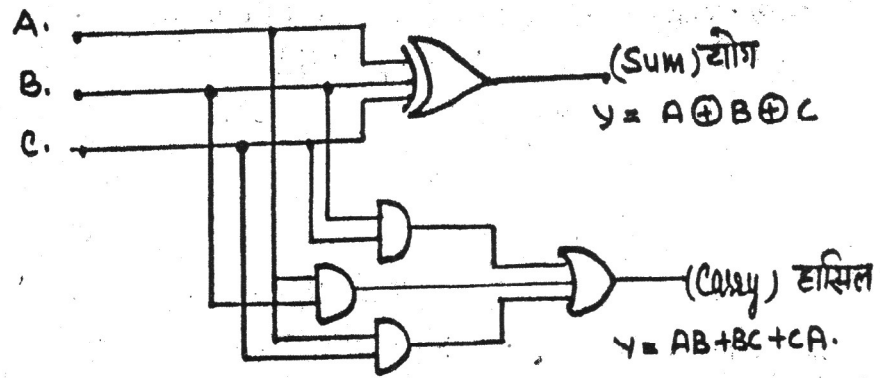
इनका संयुक्त रूप चित्र नं. 27 में देखें। तथा ब्लाक चित्र नं. 28 में दिया गया है।

कारनाफ मानचित्र (फुल एडर की ट्रुथ टेबिल)

	$\bar{C}$	C	
$\bar{A}\bar{B}$	0	0	
$\bar{A}B$	0	1	Pair I = BC
AB	1	1	Pair II = AB
$A\bar{B}$	0	1	Pair III = AC
			$\therefore$ समीकरण बनी $BC+AB+AC$

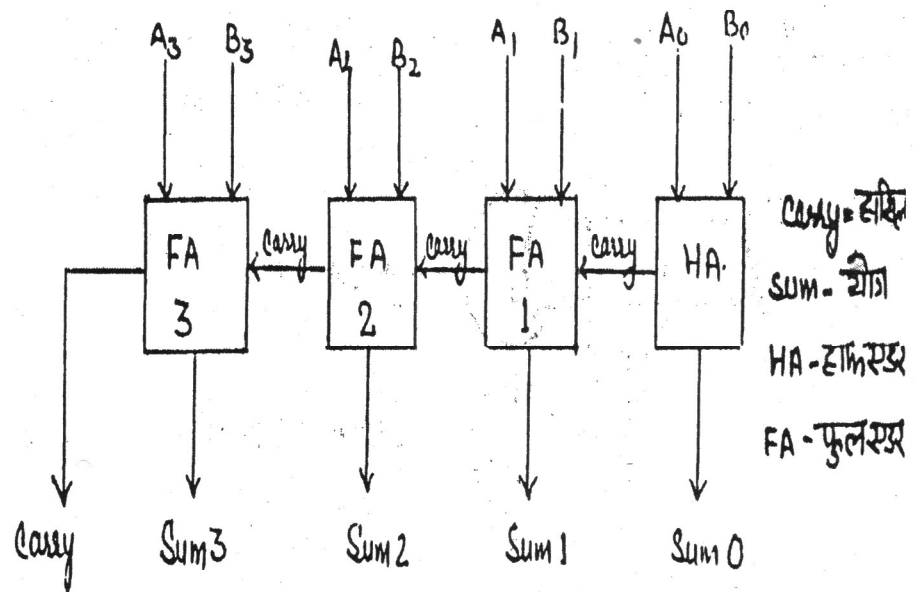
[46] ST-52

लॉजिक सर्किट (LOGIC CIRCUIT)



चित्र नं० 27

फुल एडर का ब्लाक चित्र:-



चित्र नं० 28

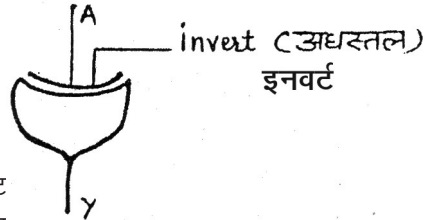
घटाना (SUBTRACTION)

घटाने के क्रिया को समझने के लिए पहले निम्न चित्र 29 समझें।

Ex-OR gate with controlled Invert-

Ex-OR गेट (नियंत्रित अधस्तल सहित)

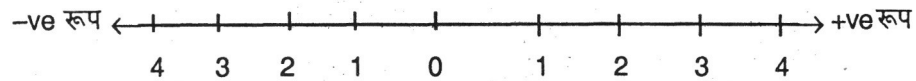
जब नियंत्रित अधस्तल 'लो' (0) है आउटपुट A होती है एवं अधस्तल हाई (1) होता है तब आउटपुट $\bar{A}$ होती है।



चित्र नं० 29

When controlled invert is low (0) out put is A and when controlled invert is high (1) and out is $\bar{A}$

किसी भी संख्या का 2's complement रूप उस संख्या का -ve होता है।



यदि दो बिट की संख्या लिखें तो 11 के बाद दुबारा 00 से गिनती प्रारंभ करते हैं। अर्थात् दुबारा जो 00 प्रारंभ हुआ उस स्थान को रिसेट कहते हैं उससे पूर्व की संख्या -ve मान लें तो रिसेट से आगे आने वाली संख्यायें +ve मान लें। इस प्रकार हम देखेंगे कि 00 के बाद 01 यानि एक +ve से पूर्व 11, -ve एक माना जायेगा।

00

01—3-ve इसी प्रकार 3 अंकों में ||| के बाद 000 रिसेट होगा

10—2-ve

11—1-ve 4 अंकों में |||| के बाद 0000 रिसेट माना जायेगा।

00—Reset

01—1+ve

10—2+ve

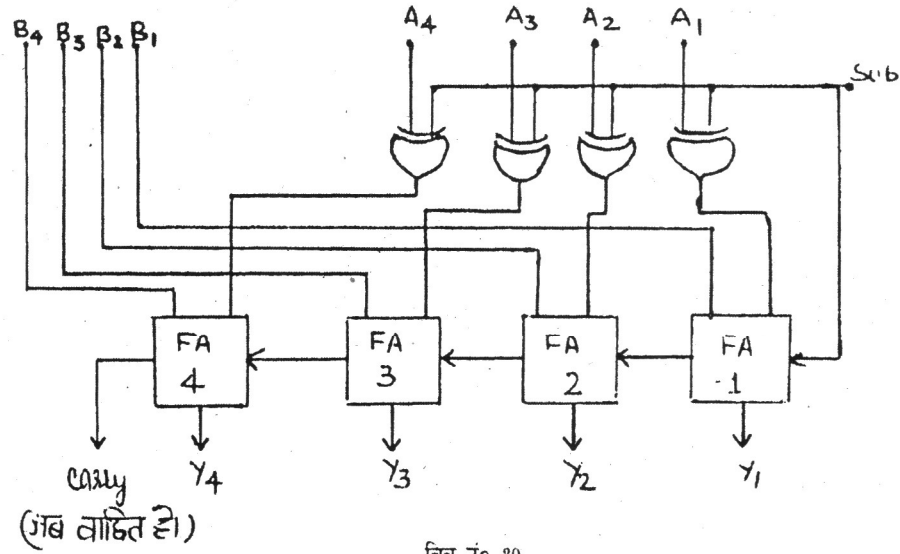
11—3+ve

[48] ST-52

जैसे:

18	8 बिट बाइनरी में	00010010
-6		00000110
00000110	का 1's complement	
11111001		
+1		
11111010	2's complement हुआ इसमें पहली	
संख्या	00010010 का जोड़ा	
100001100	यहां अंत में प्राप्त हासिल छोड़ दी जाती है तो	
उत्तर - 00001100	यानि $(12)_{10}$ हुआ	

बाइनरी एडर कम सबट्रैक्शन (BINARY ADDER CUM SUBTRACTION)



'की' बोर्ड पर -ve चिन्ह दबाते ही सबट्रेक्ट हाई (1) हो जाता है उसके पश्चात् शेष कार्य एडर की भांति ही होता है। सबट्रेक्ट हाई (1) होने से Ex-OR गेट के आउटपुट A_4, A_3, A_2, A_1 , में मिलेगा तथा 1st हाफ एडर को हासिल मिल जाता है जब सबट्रेक्ट

- (0) रहता है तो यह दी हुई संख्याओं को जोड़ने का काम करता है। जब सबट्रेक्ट हाई
(1) होता है तो B संख्या को A में से घटा देता है।

If subs is low the circuit works as Adder and when subs is high the circuit works as subtractor.

इस प्रकार घटाने के लिए जोड़ने की क्रिया में अन्तिम फुल एडर हासिल को अनदेखा कर दिया जाता है। इसे चित्र नं. 30 में दिखाया गया है।

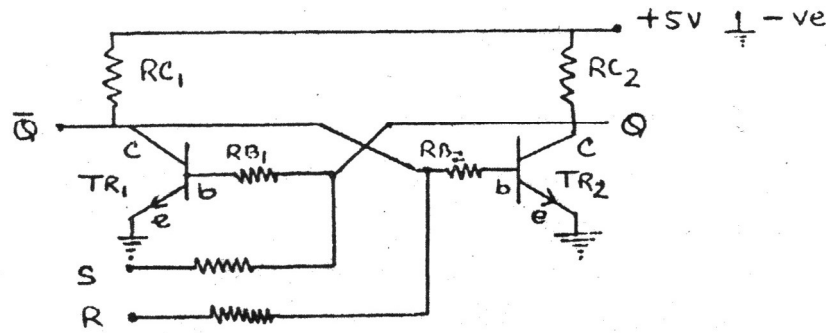
फ्लिप-फ्लॉप (FLIP-FLOP)

इसका प्रयोग रजिस्टर, काउन्टर (गुणक) एवं स्मृतियों में होता है। (It is used in Registers, Counters and Memories)

रजिस्टर (Register) :- इसमें किसी भी डाटा को एकत्र कर लिया जाता है। यह एक प्रकार से रफ कापी अथवा अस्थायी रिकार्ड होता है। क्योंकि इसमें कुछ समय के लिए ही डाटा एकत्र किया जाता है। जैसे कोई संख्या जमा की जा रही है तो यदि हासिल आ जाए तो इसमें हासिल को एकत्र कर लिया जाता है और थोड़ी देर बाद ही अगले अंकों में इस हासिल को जोड़ दिया जाता है।

मेमोरी (Memory) :- स्मृति या मैमोरी एक ऐसा साधन है जिसमें डाटा को स्थाई तौर पर रिकार्ड कर दिया जाता है जैसे जब कोई कम्प्यूटर निर्माण किया जाता है तो उससे क्या-क्या कार्य किया जा सकता है उनसे संबंधित कार्यक्रम मैमोरी (स्मृति) में एकत्र कर दिए जाते हैं।

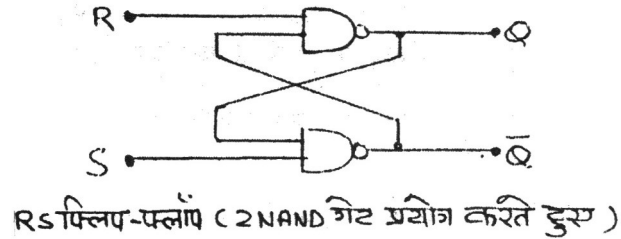
(1) फ्लिप-फ्लॉप (R.S. FLIP-FLOP) -



चित्र नं० 31

[50] ST-52

R	S	Q
0	0	* Race
0	1	1 Set
1	0	0 Reset
1	1	NC



चित्र नं० 32

यह R का तात्पर्य रिसेट तथा एस S का तात्पर्य सेट होता है। ट्रांजिस्टर से बने RS फ्लिप-फ्लॉप में +5V चालू होते ही दोनों ट्रांजिस्टर में से किसी एक को संतृप्त कारक करंट (Saturating current) जाती है जब वह संतृप्त हो जायेगा तो Q आउटपुट=0 हो जायेगा। यही 0 (लो) दूसरे ट्रांजिस्टर के बेस पर आ जाने से वह ट्रांजिस्टर संतृप्त नहीं हो पायेगा इसलिए वहां आउटपुट हाई (1) हो जायेगा। देखें चित्र नं. 31

When power supply will be made on, then racing will be start in both the transistors (T_1, T_2) and which out put will be (1) it is undetermined (No Body Can Say).

इसके पश्चात् ट्रुथ टेबिल के अनुसार R-S पर जिस अनुसार इनपुट प्रदान किए जायेंगे उनका आउटपुट Q पर प्राप्त हो जाता है। ट्रुथ टेबिल में Q सेट स्थिति पर होने पर (1) तथा Q रिसेट स्थिति पर होने पर 'लो' (0) हो जाता है। जो तब तक बना रहता है जब तक या तो RS पर क्रम (Order) में परिवर्तन न किये जाये अथवा पावर सप्लाई न काट दी जाये। अर्थात्

Q सेट का तात्पर्य Q पर +5V है।

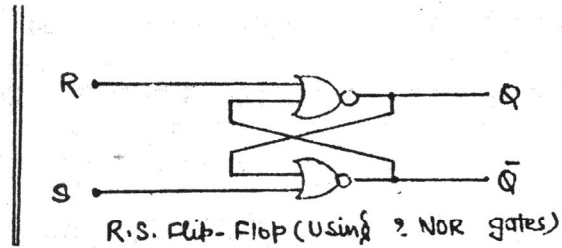
Q रिसेट का तात्पर्य Q पर 0V है।

RS दोनों क्रम (Order) (1) होने पर कोई परिवर्तन नहीं होता। दोनों क्रम (Order) (0) होने पर रेसिंग होती है। उस समय Q की क्या स्थिति होती है (1) या (0) कुछ नहीं कहा जा सकता। दो NOR गेटों से बने RS फ्लिप-फ्लॉप में भी उपरोक्त प्रकार से कार्य होता है जबकि दो NAND गेट से बने RS फ्लिप-फ्लॉप में RS क्रम (Order) दोनों 'लो' (0) होने पर Q रेसिंग में होता है। दोनों क्रम (Order) हाई (1) होने पर Q

स्थिति में कोई परिवर्तन नहीं होता। दोनों प्रकार के गेटों से बने फ्लिप-फ्लॉप क्रमशः चित्र नं. 32 तथा 33 में दिए गए हैं।

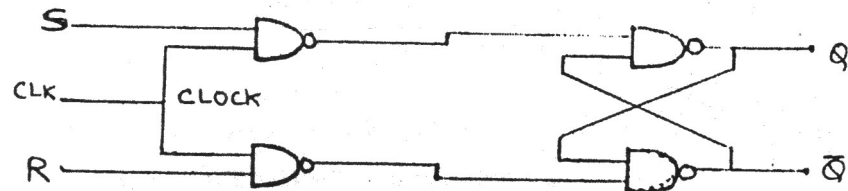
TRUTH TABLE

R	S	Q
0	0	0NC
0	1	1Set
1	0	0Reset
1	1	* Race



चित्र सं 33

(2) क्लॉक्ड R-S फ्लिप-फ्लॉप Clocked R.S. Flip Flop.



चित्र नं० 34

CLOCK (CLK) RS FLIP-FLOP

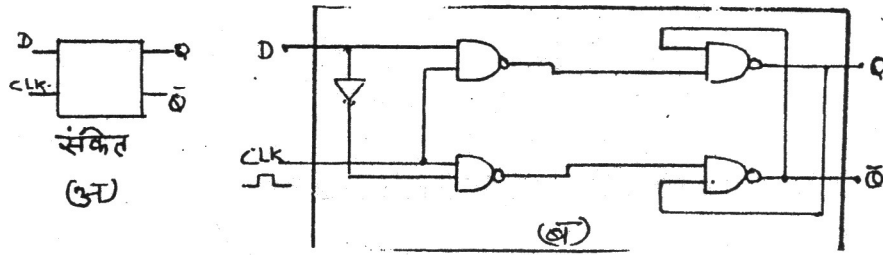
CLK	R	S	Q
0	0	0	NOCHANGE
0	0	1	NC.
0	1	0	NC.
0	1	1	NC.
1	0	0	NC.
1	0	1	1 Set.
1	1	0	0 Reset
1	1	1	* Race

Till clock signal is low (0) output will remain unchanged. No effect of input on output.

[52] ST-52

अर्थात् जब तक हम CLK बिन्दु पर लो (0) स्थिति रखेंगे तब तक आउटपुट Q की स्थिति में कोई परिवर्तन नहीं होगा। CLK पर (0) सिगनल रहने से इनपुट का कोई भी प्रभाव आउटपुट पर नहीं पड़ता चाहे इनपुट की स्थिति कितनी ही बार क्यों न बदली जाए। क्लॉकड फ्लिप चित्र नं. 34 में दिखाया गया है।

(3) क्लॉकड 'डी' फ्लिप-फ्लॉप - (Clocked D Flip-Flop)



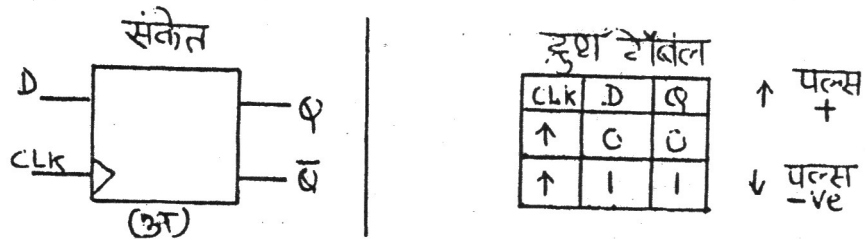
चित्र नं० 35

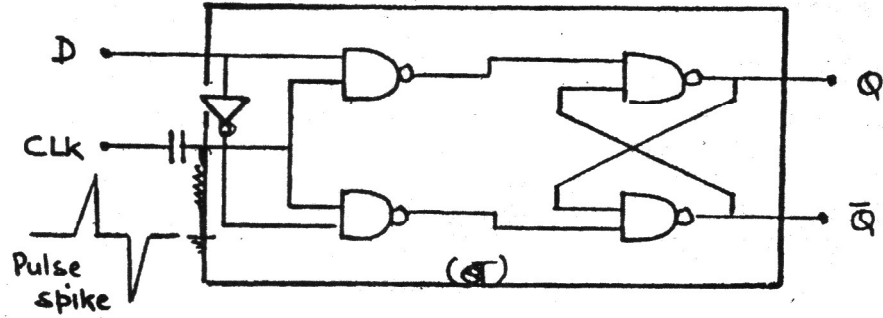
CLK	D	Q
0	0	NC
0	1	NC
1	0	0
1	1	1

अब तक जो फ्लिप-फ्लॉप पढ़े गये हैं उनमें सेट (S) तथा रीसेट R दो इनपुट होते थे। क्लॉकड D वाले फ्लिप-फ्लॉप में एक अतिरिक्त NOT का प्रयोग करके केवल एक ही इनपुट बनाया गया जिससे फ्लिप-फ्लॉप की कार्य सिस्टम और सरल हो जायेगी। इससे जो स्थिति D इनपुट पर होती है वही स्थिति Q पर प्राप्त हो जाती है बशर्ते क्लॉक सिगनल (CLK Signal) हाई (1) हो अन्यथा Q की स्थिति में कोई परिवर्तन नहीं होता।

देखें चित्र नं. 35(अ) तथा (ब)

(4) एज ट्रिगरड D फ्लिप-फ्लॉप - (Edge-Triggered D Flip-Flop)

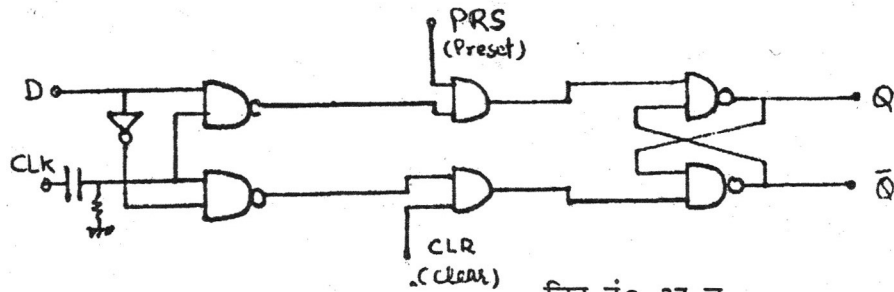




चित्र नं० 36 : Clocked D latch Flip=Flop क्लॉक डी लैचफ्लिप फ्लॉप

क्लॉकड D फ्लिप-फ्लॉप में जब क्लॉक बिन्दु (Clock Point) पर उच्च तल सिग्नल हाई (1) दिया जाता था तो +5VDC सप्लाय क्लॉक सर्किट (Clock Circuit) को मिलने पर वर्गाकार तरंग (Square Wave) के रूप में करंट प्रवाह होता था। अतः जब तक वर्गाकार तरंग में +ve रूप में होता था, तब तक उस समय के दौरान में जितने भी सिग्नल D पर पहुंचते थे वे सभी Q आउटपुट पर क्रियाशील होते थे। ऐसी स्थिति में अनावश्यक डाटा (जिन्हें एकत्र करना नहीं होता) भी एकत्र हो जाते थे क्योंकि क्लॉक +ve(1) का समय अन्तराल (Duration) एकत्र किए जाने वाले सिग्नल की तुलना में अधिक हो जाता था। इसलिए एक कंडेन्सर तथा प्रतिरोध की सहायता से एक ऐसा सर्किट तैयार किया गया जिसके क्लॉक बटन दबाते ही कंडेन्सर आवेशित होकर अनावेशित (Discharge) होने लगता है। इस समय में पल्स +ve स्पाइक का समय अन्तराल काफी कम हो गया और आवश्यकता के अनुरूप ही आउटपुट स्टोर को जाने लगी।

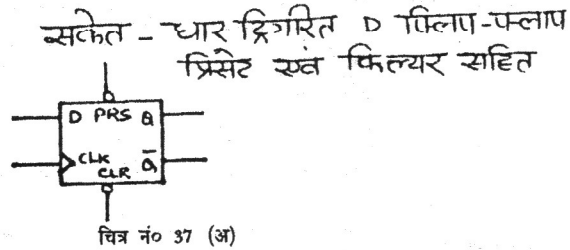
(5) एज ट्रिगरड डी फ्लिप-फ्लॉप प्रीसेट एवं क्लियर सहित (Edge Triggered D Flip-Flop with PRS and CLR)



चित्र नं० 37 ब

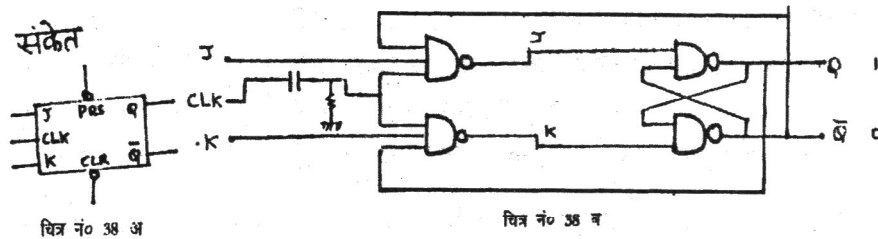
ट्रुथ टेबिल

PRs	CLR	CLK	D	Q
0	1	-	-	1
1	0	-	-	0
-	-	0↓	0	N/C
-	-	↑	0	0
-	-	↑	1	1



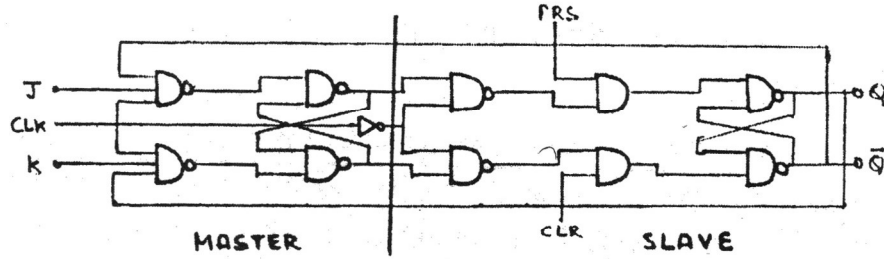
इस फ्लिप-फ्लॉप सर्किट में (PRS) प्रीसेट तथा (CLR) क्लीयर और लगाया जाता है। इसमें PRS=0 तथा CLR=1 करने पर क्लॉक (CLK) चाहे जो भी हो तथा D चाहे जो भी हो परन्तु Q=1 हो जाता है। पुराने एकत्र आंकड़ों को मिटाकर यदि नये आंकड़ों को जब एकत्र करना होता है तथा CLR 0 तथा PRS को हाई (1) करना पड़ता है। इस स्थिति में क्लॉक व D चाहे जो भी हो परन्तु आउटपुट Q=0 हो जाता है। अन्य स्थितियां ट्रुथ टेबिल के अनुसार रहती है। देखें चित्र नं. 37

(6) जे के फ्लिप-फ्लॉप : (JK Flip Flop)



यह सर्किट RS फ्लिप-फ्लॉप सर्किट में दो और NAND गेट चित्र नं. 38 के अनुसार लगाकर बनाया जाता है। इन दोनों NAND गेटों को R.S. फ्लिप-फ्लॉप की आउट पुट से जोड़ते हैं। इसका प्रयोग काउन्टर (गणक) में होता है काउन्टर या गणक गणना कार्य में ही प्रयोग किया जाता है। ट्रुथ टेबिल के अनुसार जब क्लॉक +ve पल्स स्पाइक होती है तथा JK बिन्दुओं पर दोनों इनपुट सिगनल हाई (1) होते हैं तो आउटपुट पर टोगलिंग (Toggling) होती है। अर्थात् जितनी बार क्लॉक पल्स देते हैं हर बाद Q परिवर्तित होता रहता है।

(7) मास्टर-स्लेव जे-के फ्लिप-फ्लॉप : (Master Slave JK Flip Flop) -



चित्र नं० 39 ब

हुए टेबल

CLK	J	K	Q
0	0	0	NC
0	0	1	NC
0	1	0	NC
0	1	1	NC
1	0	0	NC
1	0	1	0
1	1	0	1
1	1	1	Toggle

चित्र नं० 39 अ

एज ट्रिगरित डी फ्लिप-फ्लॉप में (Edge Triggered D Flip-Flop) ये दिक्कतें आने लगी कि कभी-कभी इनपुट पर दिए गए आंकड़े आउटपुट Q पर एकत्र नहीं हो पाते थे क्योंकि जो समय आंकड़ों को इनपुट से आउटपुट Q पर पहुंचने में लगता था उससे भी कम समय में +ve पल्स स्पाइक +ve होने पर किसी भी इनपुट आंकड़े का आउटपुट पर प्रभाव नहीं पड़ता था। इसलिए एक ऐसा सर्किट बनाया गया जिसमें यह दोष भी समाप्त हो गया। क्योंकि ज्योंही पल्स स्पाइक धनात्मक (+ve) वाले भाग में (-ve) होती है त्योंही स्लेव वाले भाग में +ve हो जाती है और इनपुट आंकड़े ट्रुथ टेबल के अनुसार Q आउटपुट पर एकत्र हो जाते हैं। देखें चित्र नं. 39 (ब)

नोट - इनपुट देने एवं आउटपुट मिलने के समय अन्तराल को संचरण देरी (Propogation Delay) कहते हैं।

रजिस्टर (REGISTERS)

इनका प्रयोग बाइनरी डिजिटो (BITS) / आंकड़ों को अस्थायी तौर पर एकत्र करने में होता है।

(1) समानान्तर इनपुट समानान्तर आउटपुट या बफर रजिस्टर – (Parallel I/P Parallel O/P or Buffer Register) इसमें एकत्र की जाने वाली बिट (BIT) इनपुट में भी समानान्तर होती है तथा एकत्र होने के लिए आउटपुट में भी समानान्तर ही निकलती है। देखें चित्र नं. 40।

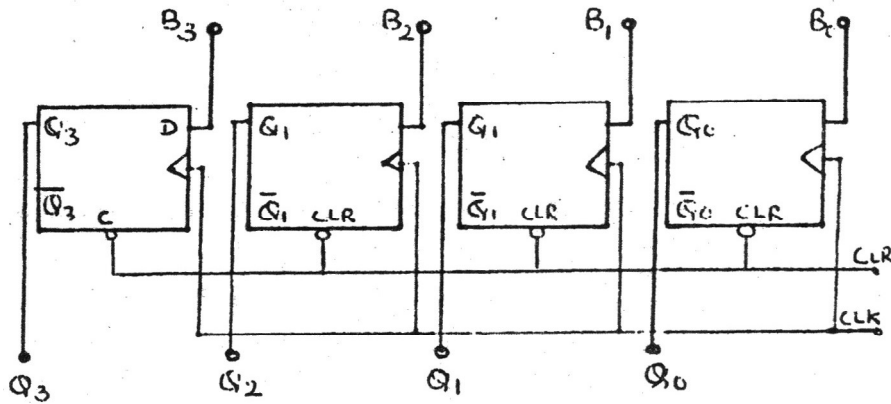
(a) जब क्लीयर CLR=0

(b) तब $Q_3, Q_2, Q_1, Q_0 = 0000$

माना इनपुट में $\rightarrow B_3B_2B_1B_0 = 1010$

तथा यदि क्लॉक (CLK) = $\uparrow$ (+ve पल्स स्पाइक)

तब प्रथम पल्स पर निकाला $Q_3Q_2Q_1Q_0$ होगा = 1010



चित्र नं० 40

PI PO (Parallel input Parallel output) Regular

(2) समानान्तर इनपुट श्रेणीबद्ध आउटपुट रजिस्टर या शिफ्ट रजिस्टर – (PISO or Shift Register) जैसा कि हम पहले भी देख चुके हैं कि दो इनपुट वाले NAND गेट के

एक इनपुट को हाई (1) रखा जाए तो वह NOT गेट का कार्य करता है इस सर्किट में 4 AND गेटों का प्रयोग किया गया है सभी 2 इनपुट वाले हैं इनका एक-एक टर्मिनल कॉमन (Common) कर दिया गया है इस सर्किट में.....

(a) पहले $CLR=0$ 'D' नं. के फ्लिप-फ्लॉप में = 0

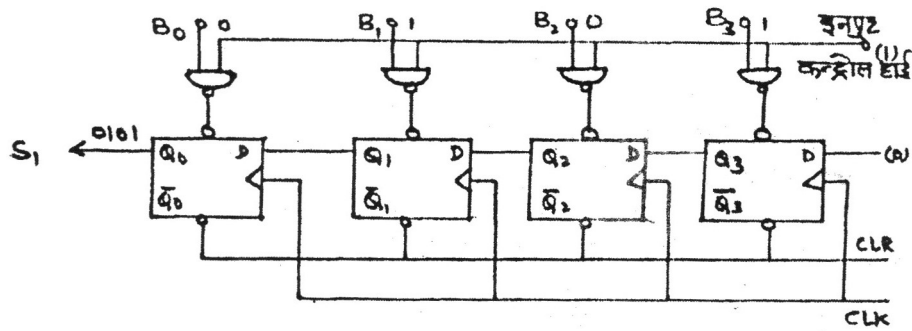
ताकि $Q_3Q_2Q_1Q_0 = 0000$

(b) $B_3B_2B_1B_0 = 0101$ इनपुट समान्तर क्रम में

— अब इनपुट नियंत्रण को हाई (1) पर रखे

— क्लॉक चार बार हाई दे।

तब $Q_3Q_2Q_1Q_0 = 0101$ श्रेणीबद्ध आउटपुट। इसका प्रयोग समान्तर आंकड़ों को श्रेणीबद्ध आंकड़ों में परिवर्तित करने के लिए किया जाता है। चित्र नं. 41 से संबंधित



चित्र नं० 41

P1 S0 (Parallel Input serial output) Register

(3) श्रेणीबद्ध इनपुट समान्तर आउटपुट (SIPO) रजिस्टर — SIPO (Serial input Parallel output (Register) इसका प्रयोग श्रेणीबद्ध इनपुट आंकड़ों को समान्तर आंकड़ों के रूप में बदलने के लिए किया जाता है इसके लिए

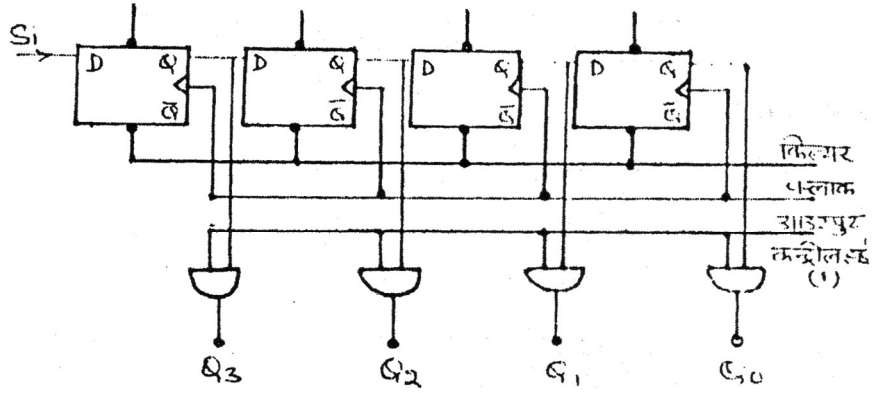
(अ) पहले क्लीयर (CLR) = 0 दें

(ब) श्रेणीबद्ध इनपुट दे माना = 1010

तत्पश्चात् चार बार क्लॉक पल्स हाई $\uparrow$ दे एवं आउटपुट कंट्रोल को हाई रखें ताकि

[58] ST-52

$Q_3 Q_2 Q_1 Q_0$ 1010 आउटपुट में समान्तर क्रम में निकलें। देखें चित्र नं. 42।



चित्र नं० 42

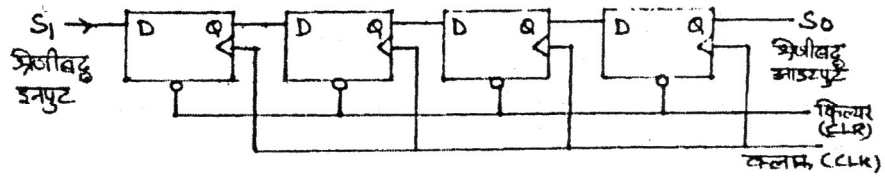
SIPO श्रेणीबद्ध इनपुट-समान्तर आउटपुट रजिस्टर

(4) **SISO (serial input serial out put) Register** श्रेणीबद्ध इनपुट श्रेणीबद्ध आउटपुट रजिस्टर -

कार्य विधि - (अ) पहले क्लीयर (CLR) = 0 दें ताकि $Q_3 Q_2 Q_1 Q_0 = 0000$

(ब) अब श्रेणीबद्ध इनपुट $S_1 = 1010$ दें तथा (हाई) क्लॉक पल्स चार बार दें इस प्रकार रजिस्टर 'लोड' हो जाता है।

(स) अब पुनः 4 क्लॉक पल्स हाई दें ताकि रजिस्टर अनलोड होकर श्रेणीबद्ध आउटपुट (Serial output) दें चित्र नं. 43।



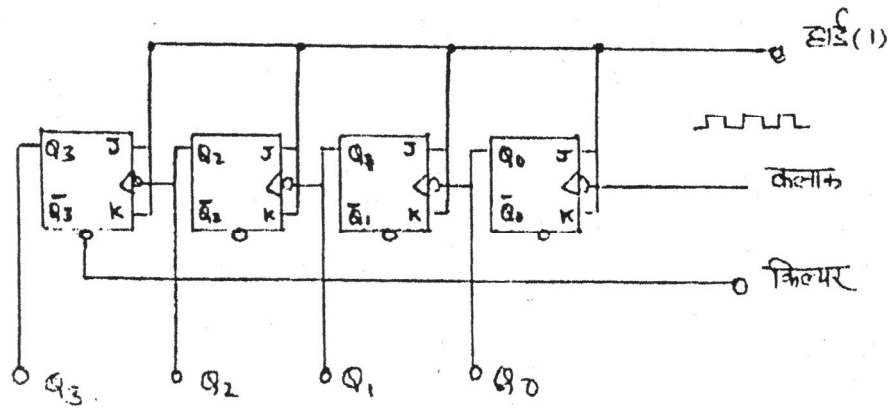
चित्र नं० 43

$S_1 S_0$ श्रेणीबद्ध इनपुट-श्रेणीबद्ध आउटपुट रजिस्टर

काउन्टरस (COUNTERS)

एसिन्क्रोनस काउन्टर या रिपिल काउन्टर (Asynchronous Counter or Ripple Counter) – असिन्क्रोनस या रिपिल काउन्टर में जितने भी फ्लिप-फ्लॉप लगे होते हैं वे एक के बाद एक करके आउटपुट देते हैं। विशेष रूप से ये काउन्टर 0 से प्रारंभ कर एक-एक पग घटते हुए गणना करते हैं गणना जब बढ़ते हुए क्रम में होती है तो उस काउन्टर को अप काउन्टर तथा जिस काउन्टर में घटते क्रम में गणना होती है उसे डाउन काउन्टर कहते हैं अप या डाउन काउन्टर के साथ लिखे जाने वाले शब्द संकेत जैसे MOD-8, MOD-16 आदि यह दर्शाते हैं कि यह काउन्टर कहां तक गणना कर सकता है। जैसे MOD-8 में $8=2^3$ होता है तो यह काउन्टर बाइनरी की तीन डिजिट तक गणना कर सकता है। अर्थात् $(000)_2$ से $(111)_2$ अर्थात् डेसीमल 0 से 7 तक गणना कर सकता है।

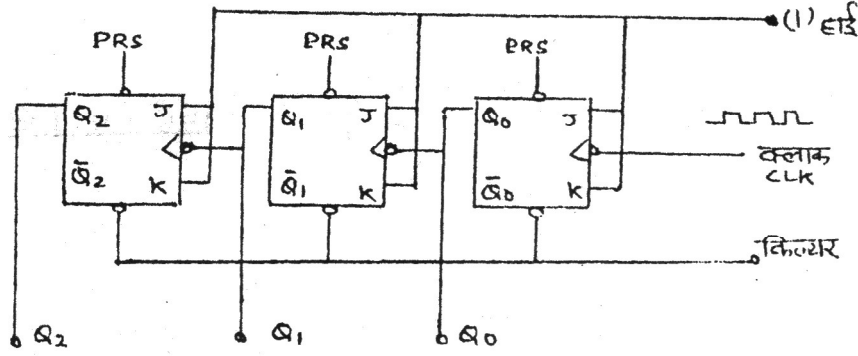
गणना में जितने बिट (बाइनरी डिजिट) का प्रयोग होता है उतने ही फ्लिप-फ्लॉप लगाये जाते हैं। चित्र 44 “4 बिट” बाइनरी गणना के लिए बनाया गया है। इसमें 4 मास्टर स्लेव J-K फ्लिप-फ्लॉप लगाये गये हैं जो -ve एज ट्रिगरित (Edge-Triggered) होते हैं। अर्थात् जब क्लॉक पल्स 1 से 0 होगी तभी टोगलिंग होती है अन्यथा नहीं इस प्रकार जहां क्लॉक 1 से 0 होगी Q बदल जाता है।



चित्र नं 0 44

[60] ST-52

(1) MOD-8 अप काउन्टर - देखें चित्र नं. 45



चित्र नं० 45

कार्य विधि - (1) क्लीयर 0 दें ताकि $Q_2, Q_1, Q_0 = 000$

अब प्रथम क्लॉक पल्स दें $Q_2, Q_1, Q_0 = 001$

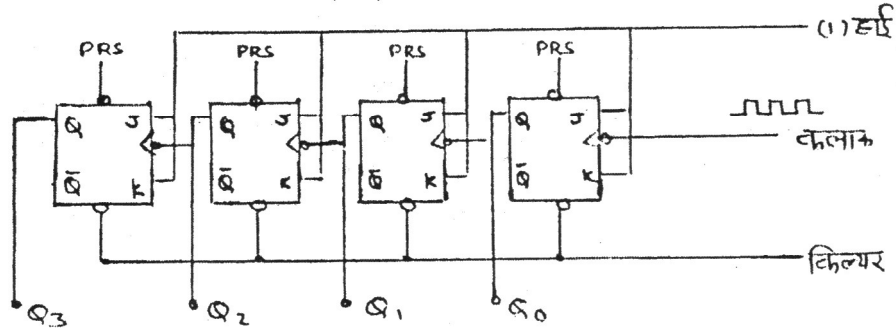
अब दूसरी क्लॉक पल्स दें $Q_2, Q_1, Q_0 = 010$

अब तीसरी क्लॉक पल्स दें $Q_2, Q_1, Q_0 = 011$

इस प्रकार प्रत्येक क्लॉक पल्स देने पर बाइनरी गणना बढ़ जाती है तथा अंत तमें 111 प्राप्त हो जाता है उसके बाद पुनः 000 आ जाता है।

क्योंकि यह काउन्टर -ve एज ट्रिगरित है अतः पल्स +ve के बाद -ve होती है, तभी आउटपुट में परिवर्तन होता है।

(2) MOD-16 अप काउन्टर-देखें चित्र नं. 46

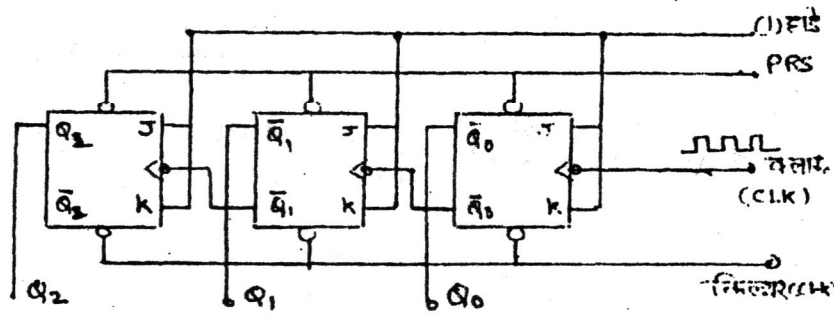


चित्र नं० 46

कार्य विधि :-

MOD-8 की भांति ही है। अंतर केवल इतना ही है कि MOD-8 गणना 000 से 111 तक करता है तथा यह 0000 से 1111 तक गणना करेगा तथा 1111 के बाद पुनः 0000 आ जायेगा।

(3) **MOD-8 डाउन काउन्टर** — डाउन काउन्टर इसलिए कहलाता है कि यह उच्चतम बाइनरी से एक-एक करके नीचे की ओर गणना करता है। MOD-8 होने के कारण यह 3 बिट बाइनरी की गणना के लिए बना होता है। अप काउन्टर में क्लियर का प्रयोग किया जाता है जबकि डाउन काउन्टर में प्रिसेट का प्रयोग करते हैं। दूसरा अंतर यह है कि अप काउन्टर में क्लॉक को Q से जोड़ा गया था लेकिन इसमें क्लॉक को $\bar{Q}$ से जोड़ा गया है चित्र नं. 47।



चित्र नं० 47

कार्य विधि — PRS प्रीसेट दे तब $Q_2, Q_1, Q_0 = 111$

प्रथम क्लॉक पल्स दें $Q_2, Q_1, Q_0 = 110$

दूसरी क्लॉक पल्स दें $Q_2, Q_1, Q_0 = 101$

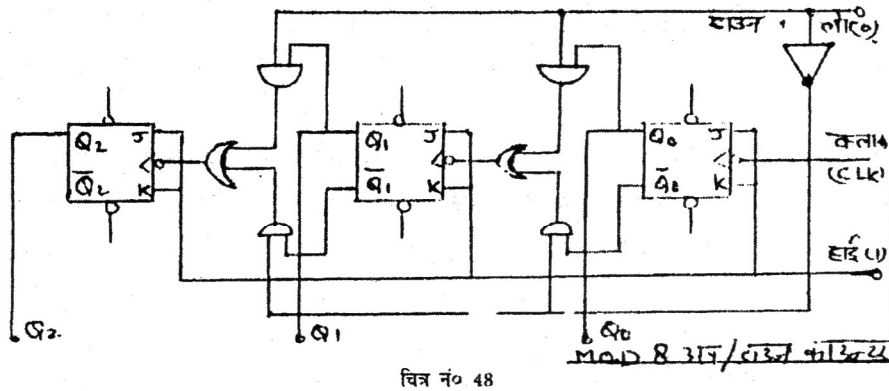
तीसरी क्लॉक पल्स दें $Q_2, Q_1, Q_0 = 100$

इस प्रकार प्रत्येक पल्स के साथ एक-एक करके डाउन गणना होती है 000 तक गणना होने के बाद पुनः 111 बन जाता है।

[62] ST-52

इसी प्रकार MOD-16 डाउन काउन्टर बना होता है इसमें 3 फ्लिप-फ्लॉप के स्थान पर चार फ्लिप-फ्लॉप होते हैं। कार्य विधि समान है यह 1111 से 0000 तक गणना कर सकता है।

(4) MOD-8 अप/डाउन काउन्टर -



इसमें PRS एवं CLR (प्रीसेट एवं क्लीयर) की आवश्यकता नहीं होती फिर भी यदि चाहे तो इनका प्रयोग किया जा सकता है।

- (1) अप काउन्टर के लिए अप (UP) Side-high (हाई) (1)
(डाउन) (DN) Side-Low (लो) (0)
- (2) डाउन काउन्टर के रूप में प्रयोग करते समय
डाउन (DN) Side-high (हाई) (1)
(अप) (UP) Side-Low (लो) (0)

अर्थात् यदि गणना टर्मिनल (काउंटिंग टर्मिनल) पर (1) high रखे तो अप गणना होगी तथा यदि (0) (Low) 'लो' रखें तो डाउन गणना होगी। इस काउन्टर में AND गेट तथा OR गेट का भी प्रयोग होता है जैसा उपरोक्त चित्र नं. 48 दिखाया गया है।

इसी प्रकार से MOD-16 अप/डाउन काउन्टर भी उपलब्ध है। इसमें एक फ्लिप-फ्लॉप और जुड़ गया है यह 0000 से 1111 तक तथा 1111 से 0000 तक गणना कर सकता है। चित्र नं. 51

(5) **MOD-10 काउन्टर (BCD Counter)** – यह BCD अर्थात् बाइनरी में कोडित डेसीमल संख्याओं (0 से 9) की बाइनरी को ही गिनता है अर्थात् 0000 से 1001 तक उसके पश्चात इसके कनेक्शन ऐसे होते हैं कि अगली क्लॉक पल्स आने पर 0000 हो जाता है। इसमें चार फ्लिप-फ्लॉप एक NAND गेट एवं एक AND गेट का प्रयोग होता है चित्र नं. 49 दिया है।

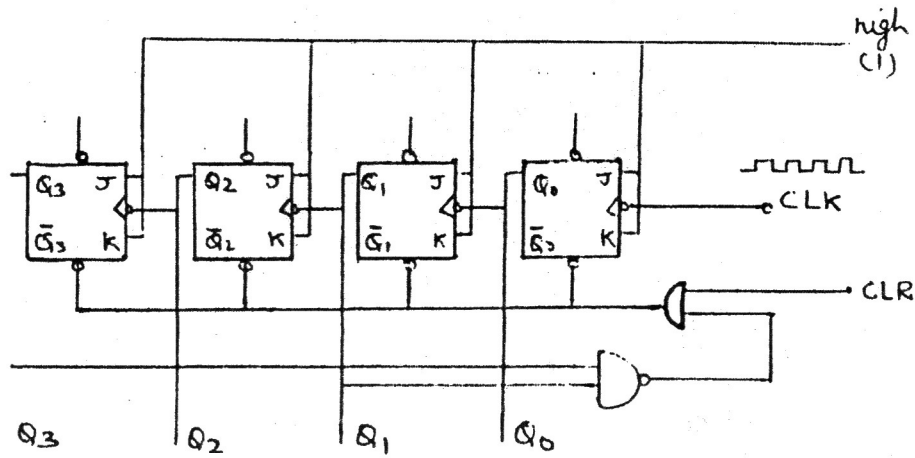
(6) **पूर्व निर्देशित काउन्टर (Pre Settable Counter)** – इस प्रकार के काउन्टर में हम जहां से चाहे गणना करनी प्रारंभ करवा सकते हैं माना हमें 0101 से प्रारंभ करानी है तो $P_3P_2P_1P_0$ पर 0101 सिगनल दिया (चित्र 50) लोड पर हाई (1) दे ताकि $Q_3Q_2Q_1Q_0$ पर 0101 पहुँचे अब क्लॉक पल्स दे।

प्रथम पल्स पर $Q_3Q_2Q_1Q_0 = 0110$

द्वितीय पल्स पर $Q_3Q_2Q_1Q_0 = 0111$

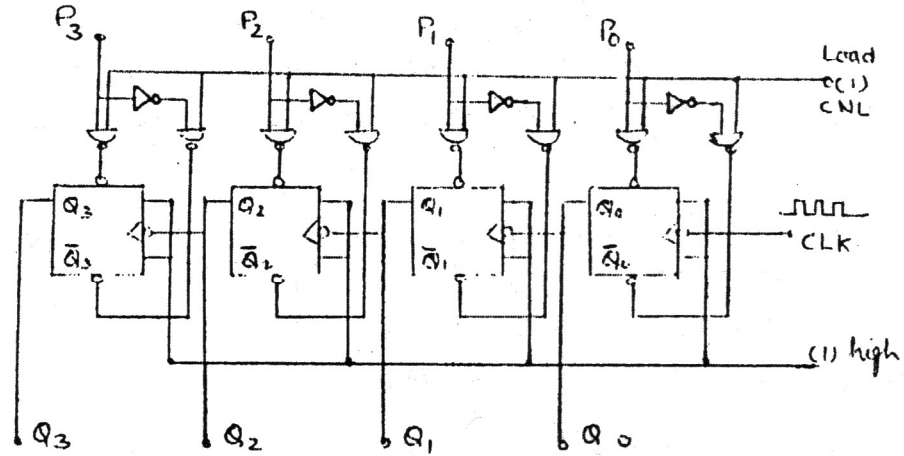
इस प्रकार 1111 तक गणना करता रहेगा तथा फिर अगली पल्स देने पर आउटपुट $Q_3Q_2Q_1Q_0 = 0000$ हो जायेगा क्योंकि लोड नार्मल हो चुका होता है। चित्र नं. 50।

MOD 10 COUNTER (BCD COUNTER)



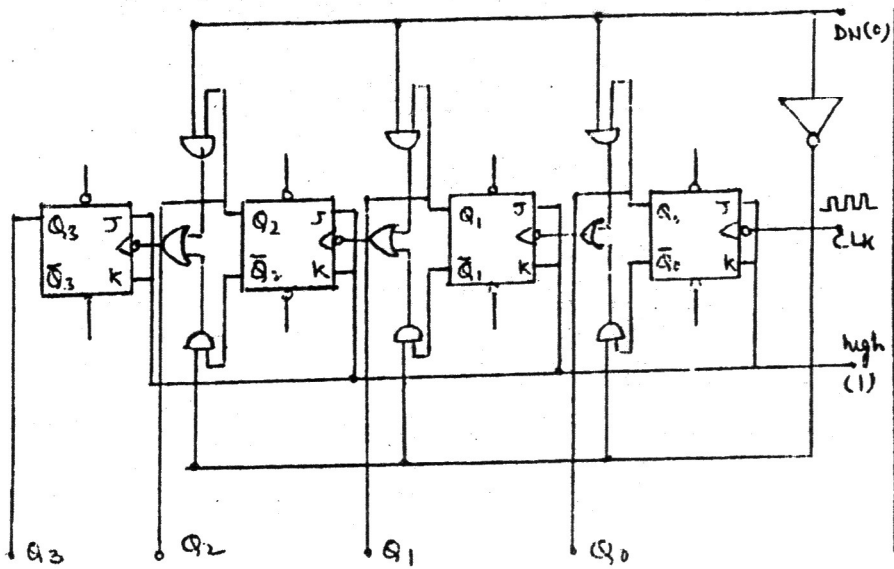
चित्र नं. 49

PRESETTABLE COUNTER (पूर्व निर्देशित काउन्टर)



चित्र नं. 50

MOD-16 UP/DOWN COUNTER (MOD 16 अप/डाउन काउन्टर)

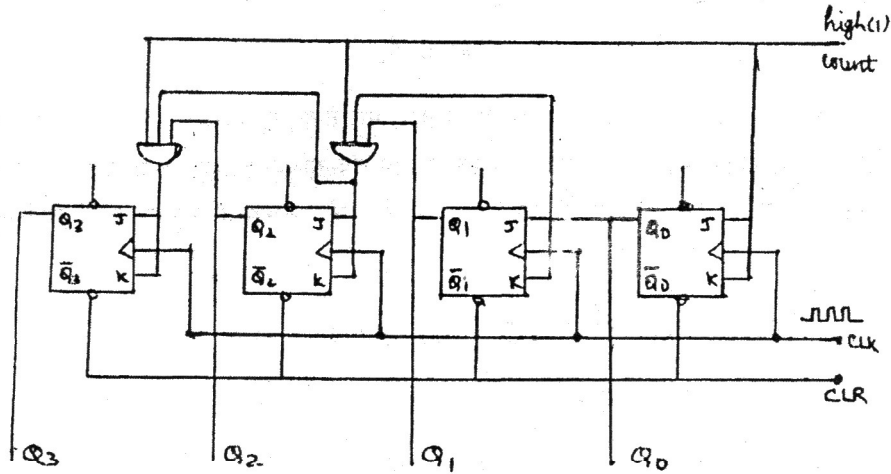


चित्र नं. 51

सिन्क्रोनस काउन्टर (Synchronous Counters) :

सिन्क्रोनस काउन्टर्स ऐसे काउन्टर होते हैं जिसमें सभी फ्लिप-फ्लॉप एक साथ काम करते हैं एवं आउटपुट देते हैं इसमें क्योंकि सभी फ्लिप-फ्लॉप एक साथ कार्य करते हैं अतः इसमें +ve एज ट्रिगरित (edge Triggered) J.K. फ्लिप-फ्लॉप का उपयोग किया जाता है इसमें भी MOD-16 काउन्टर, रिंग काउन्टर या शिफ्ट काउन्टर, ट्रिवर्स्टिड काउन्टर आदि होते हैं। इन सभी के चित्र नीचे दिए गए हैं। इसमें क्योंकि सभी फ्लिप-फ्लॉप एक साथ टोगल करते हैं अतः गणना तीव्र गति से करते हैं।

MOD-16 Counter (मोड 16 काउन्टर)



चित्र नं. 52 माड 16 काउन्टर

क्लीयर (CLR) = 0 दें ताकि $Q_3Q_2Q_1Q_0 = 0000$

काउन्टर = 1 high और इसे तब तक हाई रखना है जब तक गणना करानी है।

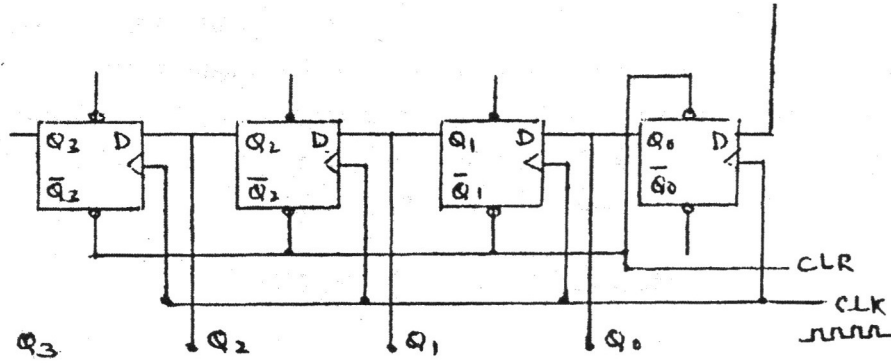
यह काउन्टर गणना करता है। = 0000 से 1111 तक इसमें गणना तीव्र गति से होने के कारण –

इसमें क्योंकि सभी फ्लिप-फ्लॉप एक साथ टोगल करते हैं इसीलिए गणना तीव्र गति से होती है। देखे चित्र नं. 52।

[66] ST-52

रिंग काउन्टर/शिफ्ट रजिस्टर काउन्टर

RING COUNTER/SHIFT REGISTER COUNTER



चित्र नं. 53

इस काउन्टर में हाई (1) फिलप-फ्लॉप पर तेजी से अगले फिलप-फ्लॉप पर शिफ्ट कर जाता है। शिफ्ट करते समय अंतिम फिलप-फ्लॉप के बाद वापस पहले फिलप-फ्लॉप पर पहुंच जाता है। इस प्रकार ये रिंग के आकार में घूमता है। देखें चित्र नं. 53।

क्लीयर (CLR) = 0 ताकि $Q_3Q_2Q_1Q_0 = 0001$

क्लॉक पल्स दे -

पहली (1st) पल्स पर $Q_3Q_2Q_1Q_0 = 0010$

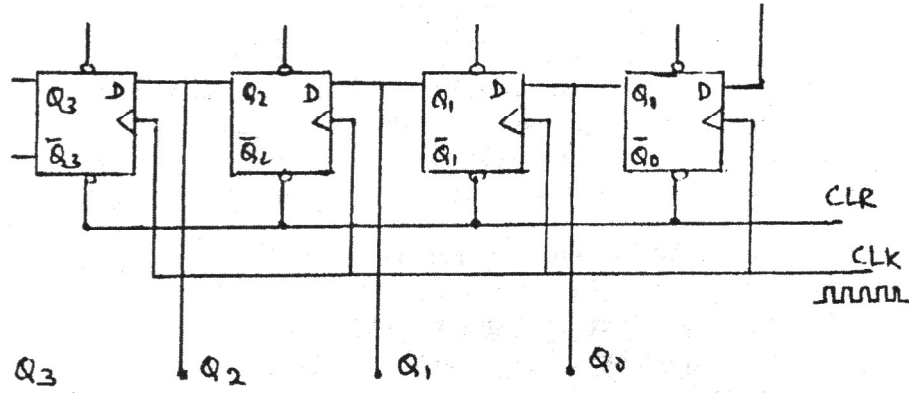
दूसरी (2nd) $Q_3Q_2Q_1Q_0 = 0100$

तीसरी (3rd) $Q_3Q_2Q_1Q_0 = 1000$

चौथी (4th) $Q_3Q_2Q_1Q_0 = 0001$

इसे तेजी से या धीमे से लगातार घुमाने के लिए मल्टी वाइबरेटर या बाइबरेटर जनरेटर का प्रयोग करना चाहिए। इस काउन्टर में D फिलप-फ्लॉप उपयोग में लाते हैं।

ट्विस्टिड काउन्टर
TWISTED COUNTER



चित्र नं. 54 ट्विस्टिड काउन्टर

इस काउन्टर में जितने फ्लिप-फ्लॉप लगाये जाते हैं उनका हाई (1) अगले फ्लिप-फ्लॉप पर बढ़ता है जब सभी फ्लिप-फ्लॉप के हाई (1) हो जाते हैं तो फ्लिप-फ्लॉप पर फिर पहले वाला क्रम 'लो' '0' से चलता है इस प्रकार हाई (1) और लो (0) एक प्रकार ट्विस्टिड हो जाते हैं। चित्र 54।

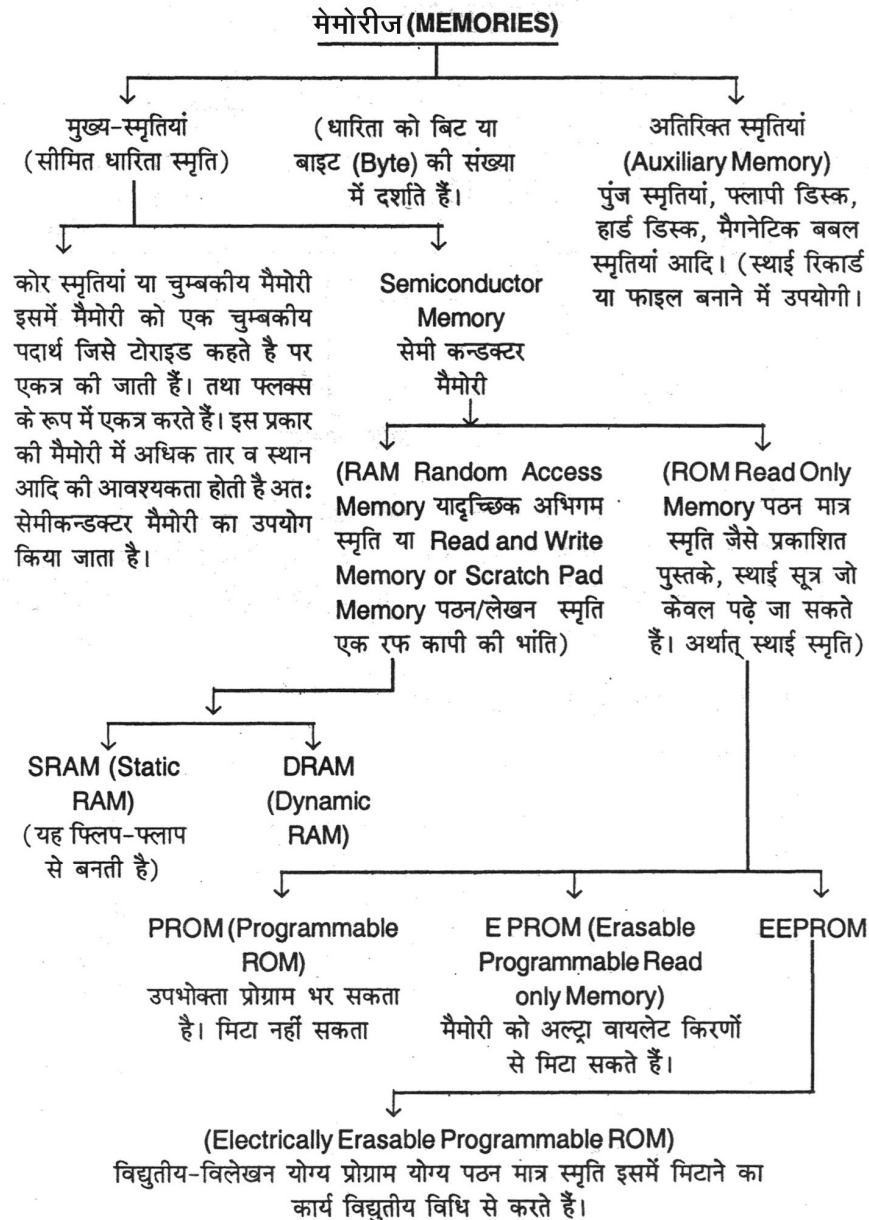
$$\text{क्लियर (CLR)} = Q_3 Q_2 Q_1 Q_0 = 0000$$

क्लॉक पल्स दे -

पहली पल्स (1st)	Q_3	Q_2	Q_1	Q_0	=	0001
दूसरी पल्स (2nd)	Q_3	Q_2	Q_1	Q_0	=	0011
तीसरी पल्स (3rd)	Q_3	Q_2	Q_1	Q_0	=	0111
चौथी पल्स (4th)	Q_3	Q_2	Q_1	Q_0	=	1111
पांचवी पल्स (5th)	Q_3	Q_2	Q_1	Q_0	=	1110
छठी पल्स (6th)	Q_3	Q_2	Q_1	Q_0	=	1100
सातवीं पल्स (7th)	Q_3	Q_2	Q_1	Q_0	=	1000
आठवीं पल्स (8th)	Q_3	Q_2	Q_1	Q_0	=	0000
नौवीं पल्स (9th)	Q_3	Q_2	Q_1	Q_0	=	0001

इसी प्रकार हाई (high (1) लो (Low) (0) का क्रम चलता रहता है।

मेमोरीज (MEMORIES)



वोलेटाइल मेमोरी (**Volatile Memories**) :- वे मेमोरी जो पावर बंद होने पर तुरंत समाप्त हो जाये वोलेटाइल मेमोरी कहलाती है। RAM मेमोरी इसी प्रकार की होती है। अन्य सभी मेमोरी नॉन वोलेटाइल (Non Volatile) हैं।

Random Access Memory or Read/Write Memory (RAM)

रेन्डम एक्सेस मेमोरी / रीड-राइट मेमोरी

इस प्रकार की स्मृतियां (Memory) आई.सी. (चिप्स) से बनायी जाती है जिसमें कुछ विशेष क्षमता (Parameter) निहित होते हैं।

चिप की क्षमता (Capacity of a Chip) – जितने बाइनरी अंक एक चिप अपने अन्दर संग्रह कर सकता है उस संख्या को चिप की धारिता कहते हैं। जैसे 64 K Bit धारिता का अर्थ है कि वह चिप अपने अन्दर 64x1024 बिट (बाइनरी डिजिट) संग्रह कर सकता है।

8 बिट या 1 बाइट पैदा करने के स्थान को मेमोरी लोकेशन कहते हैं। मेमोरी लोकेशन बिट तथा निबिल के लिए भी हो सकती है। एक बिट स्टोर करने के लिए एक मेमोरी सैल की आवश्यकता होती है। अर्थात् जितने बिट स्टोर करने हैं। उतने ही मेमोरी सैल चाहिए।

मेमोरी स्पीड :- मेमोरी की पहुंच तक लगा समय मेमोरी गति कहलाता है। यह नैनो सेकण्ड में होना चाहिए।

पावर खपत :- (i) निष्क्रिय पावर (Idle Power) या स्टैंड बाई पावर खपत

(ii) कार्य करते समय पावर खपत

उपरोक्त दो प्रकार से आई.सी. मेमोरी में पावर खपत होती है। जहां पर गति वॉन्छनीय है वहां पर DTL (डायोड ट्रांजिस्टर लाजिक) का प्रयोग करते हैं उसमें पावर खपत मिली वॉट में होती है अर्थात् पावर खपत अधिक होती है।

जहां पर पावर खपत कम करनी है वहां पर MOS (मेटल आक्साइड सेमी कन्डक्टर) आई.सी. का प्रयोग करते हैं इसमें पावर खपत माइक्रो वॉट में होती है।

मेमोरी पता, कोड एवं आंकड़े :- चित्र में 8 बाइनरी वाले शब्दों का संग्रह करने के लिए 6 रजिस्टर दिखाए गए हैं। जिन्हें लोकेशन नं. 0 से 5 तक रखा गया है। किसी रजिस्टर विशेष में पहुंचने के लिए कम से कम 3 बिट कोड की आवश्यकता है उस कोड को जिसकी सहायता से रजिस्टर विशेष के लोकेशन तक पहुंचा जा सकता है को मेमोरी का पता कहते हैं “मेमोरी पता” का कोड बाइनरी में होता है तथा उस पते

[70] ST-52

के लोकेशन वाले रजिस्टर में जो आंकड़े लिखे होते हैं उन्हें मैमोरी आंकड़े कहते हैं ये भी बाइनरी में होते हैं। जैसे नीचे दिए गए चित्र में मैमोरी पते '000' वाली लोकेशन पर रजिस्टर R₀ है तथा उसमें 8 बिट वाला आंकड़ा 01010110 लिखा है तो मैमोरी पता $(0000)_2$ है तथा मैमोरी आंकड़ा $(01010110)_2$ है।

कोड	लोकेशन									
		↓								
0000	0	0	1	0	1	0	1	1	0	R ₀
0001	1									R ₁
0010	2									R ₂
0011	3	1	0	1	0	1	1	0	1	R ₃
0100	4									R ₄
0101	5									R ₅

मैमोरी पता, पते का कोड एवं पते के आंकड़े

पता बिट एवं पतों की संख्या में संबंध

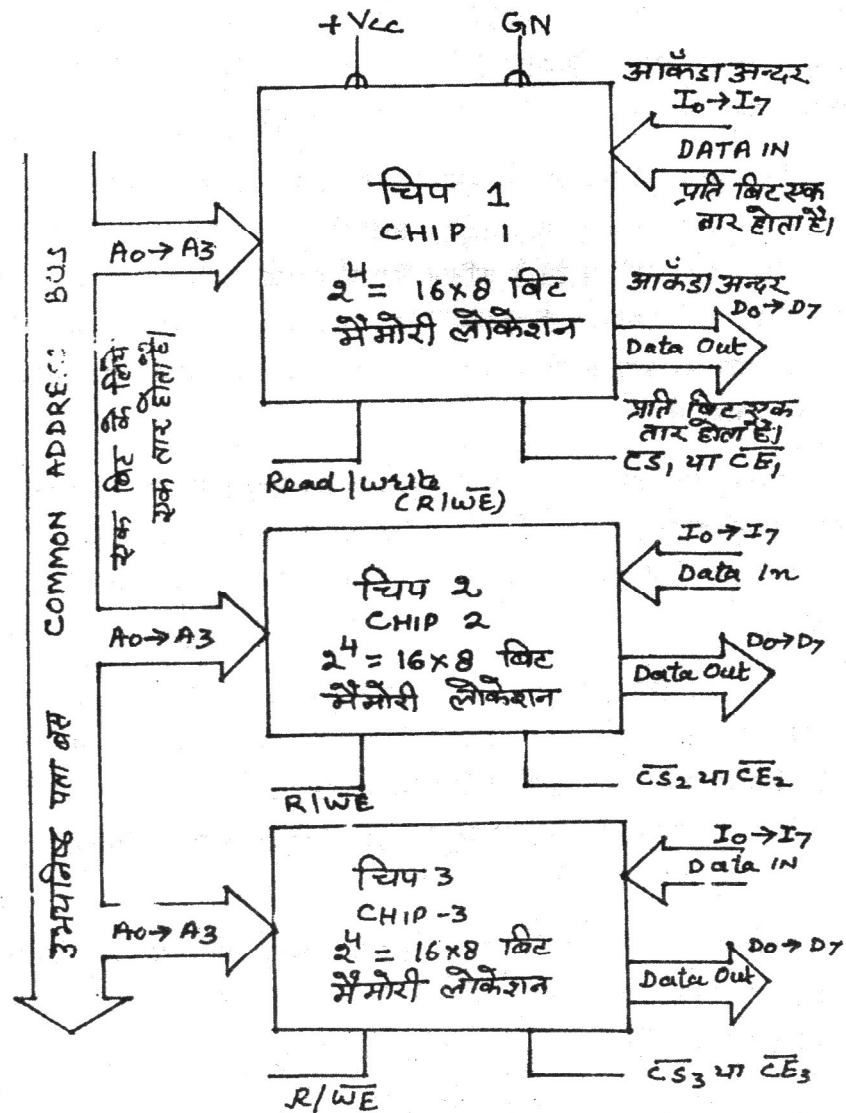
2^n = पतों की संख्या = लोकेशन की संख्या

जहां n = पता बिट (Address bit)

माना यदि एड्रेस बिट 4 है तो लोकेशन संख्या = $2^4 = 16$

साथ वाले चित्र 55 में 8 बिट वाले 16 शब्दों के लिए तीन मैमोरी चिप 1, 2, एवं 3 दिखाए गए हैं। प्रत्येक चिप में अलग-अलग R₀ से R₁₅ तक रजिस्टर होते हैं एवं प्रत्येक रजिस्टर में 8 बिट भण्डारण करने के लिए 8 फिलप-फ्लॉप लगे होते हैं इन रजिस्ट्रों का अभिनिर्धारण (Identification) टर्मिनल A₀ से A₃ पर आये लॉजिक तल से होता है। ये बिट ही "पता बिट" कहलाती है तथा A₃A₂A₁A₀ की "अरे" बिट ही पता (Address) कहलाती है। चार बिट से $2^4=16$ रजिस्ट्रों को अभिनिर्धारित किया जा सकता है प्रत्येक

रजिस्टर का अपना एक 'पता' होता है तथा रजिस्टर विशेष पर पहुंचने के लिए उस रजिस्टर के "पते" की बिट को पते के टर्मिनलों पर दिया जाता है।



चित्र 55 :- स्मृति चिप (Memory Chip)

8 बिट, जिन्हें स्टोर करना है अर्थात् पते वाले रजिस्टर में लिखना है, को मैमोरी चिप के इनपुट टर्मिनलों I_0 से I_7 पर दिया जाता है (अर्थात् Data IN बस पर)। किसी शब्द को रजिस्टर में लिखने के लिए, रजिस्टर विशेष को, संबोधित किया जाना चाहिए, डाटा इनपुट पर उपस्थिति रहना चाहिए तथा राइट इनेबिल (write enable) WE एवं चिप सलैक्टर विशेष, जिस पर लिखना है (जैसे चित्र में 3 चिप दिखाए गए हैं) CS इनेबिल अर्थात् क्रमशः लाजिक 0, 1 अवस्था या एक्टिव होने चाहिए।

मैमोरी पढ़ने की प्रक्रिया :- इसमें “संबोधित पते” वाले रजिस्टर में स्टोर हुए डाटा को आउटपुट टर्मिनल D_7 to D_0 पर लाना होता है। मैमोरी पढ़ने के लिए हम “पते” को “पता बस” में देते हैं वॉन्छित चिप के CS को इनेबिल करते हैं तथा WE को लॉजिक “1” पर रखते हैं, क्योंकि लिखने की अवस्था में WE को शून्य लॉजिक पर रखा जाता है। इसी कारण से निर्माता E टर्मिनल को पठन/लेखन” (Write/Read) में हमने देखा कि चिप सलैक्टर हमेशा इनेबिल अवस्था में रहा यदि चिप एक्टिव अवस्था में नहीं होगा तो बाह्य जगत से पृथक् रहेगा इसमें इस अवस्था में न कुछ लिखा जा सकेगा तथा न ही पढ़ा जा सकेगा।

क्योंकि हम उपरोक्त प्रकार की मैमोरी को पढ़ एवं लिख सकते हैं अतः इस कारण इस प्रकार की मैमोरी “रीड एण्ड राइट” मैमोरी कहलाती है। तथा इस मैमोरी में हम किसी भी रजिस्टर पर पहुंच सकते हैं अचानक (at Random), उसे पढ़ने या उसमें कोई शब्द लिखने के लिए/इसी कारण से इसे Random Access Memory (RAM) भी कहते हैं।

RAM मैमोरी के निर्माण में फिलप-फ्लॉप या संघनित्र कन्डैन्सर का प्रयोग किया जाता है उसी आधार पर इन की क्रिया विधि में अंतर आ जाता है अतः RAM मैमोरी दो प्रकार की होती है।

(1) स्टैटिक RAM (Static RAM)

(2) डायनेमिक RAM (Dynamic RAM)

इन दोनों की तुलना इस प्रकार कर सकते हैं।

क्र.सं.	स्टैटिक RAM (S. RAM)	डायनेमिक RAM (D. RAM)
1.	फिलप-फ्लॉप से बनाये जाते हैं। तथा एक फिलप-फ्लॉप में एक बिट स्टोर होती है।	ये MOS ट्रांजिस्टर एवं छोटे-छोटे कन्डैन्सरों से बनाये जाते हैं। तथा एक कन्डैन्सर में एक बिट स्टोर होती है। लॉजिक 1 = कन्डैन्सर चार्ज, लॉजिक 0 = कन्डैन्सर डिसचार्ज
2.	फिलप-फ्लॉप चिप पर अधिक स्थान (कन्डैन्सर से चार गुना) घेरता है।	कन्डैन्सर चिप पर कम स्थान घेरते हैं।
3.	स्टोर करने की क्षमता कम। अतः उस स्थान पर प्रयोग किया जाता है, जहां पर 64K bits या कम मैमोरी धारिता की आवश्यकता हो। आजकल उपलब्ध इसकी अधिकतम धारिता $16 \text{ Kbit} = 163584 \text{ bit} = 2^{14}$ वाले SRAM हैं।	स्टोर करने की क्षमता अधिक। अतः उस स्थान पर प्रयोग करते हैं, जहां पर लगभग 128K bits या अधिक धारिता की आवश्यकता हो यह आर्थिक रूप से तब ही लाभदायक है। इसकी अधिकतम धारिता $64 \text{ Kbit} = 65536 = 2^{16}$ होती है।
4.	इसमें बिट तब तक स्टोर रहेगी जब तक मैमोरी चिप को पावर मिलती रहेगी।	यदि चार्ज हुए कन्डैन्सर को छोड़ दिया जाये तो धीरे-धीरे अपना आवेश (चार्ज) खो देता है।
5.	इसमें रिफ्रेशिंग चार्ज की आवश्यकता नहीं होती।	इसमें कन्डैन्सरों को चार्ज रखने के लिए बार-बार, अधिकतम 2 मिली सै. के अन्तराल पर, रिफ्रेशिंग चार्ज एक अलग सर्किट से देते हैं।

6.	ये क्योंकि फ्लिप-फ्लॉप से बने होते हैं, अतः इन्हें स्टैटिक मैमोरी कहते हैं तथा इन्हें SRAM से लिखते हैं।	इसमें क्योंकि बार-बार रिफ्रेशिंग चार्ज दिया जाता है कन्डैन्सरों को चार्ज करने के लिए, अतः संग्राहक मैमोरी को डायनेमिक मैमोरी कहते हैं। तथा DRAM से लिखते हैं।
7.	पावर खपत अधिक	पावर खपत कम

आई.सी चिप में टर्मिनल होते हैं जो चिप को विद्युतीय पावर देते हैं। इस पावर का उपयोग चिप में ट्रांजिस्टर्स को आपरेट करने में किया जाता है। यदि पावर काट दी जाये तथा पुनः चालू की जाये तो RAM मैमोरी में वे आंकड़े उपलब्ध नहीं रहते जो पावर करने से पहले उपलब्ध थे। इस प्रकार अस्थायी रूप से पावर कटने पर भी RAM मैमोरी समाप्त हो जाती है इसलिए सेमीकन्डैक्टर RAM मैमोरी को वोलेटाइल मैमोरी (वाष्पशील स्मृति) कहते हैं।

अतः RAM मैमोरी में डाटा को संग्रहित रखने के लिए पावर व्याधान को नहीं होने देना है। इसलिए इसमें बैटरी बैक-अप सिस्टम, जो सामान्यतः बैट्री सप्लाय होती है, का उपयोग किया जाता है।

इसके अतिरिक्त अन्य सभी मैमोरी (Non Volatile) अवाष्पशील स्मृति होती है।

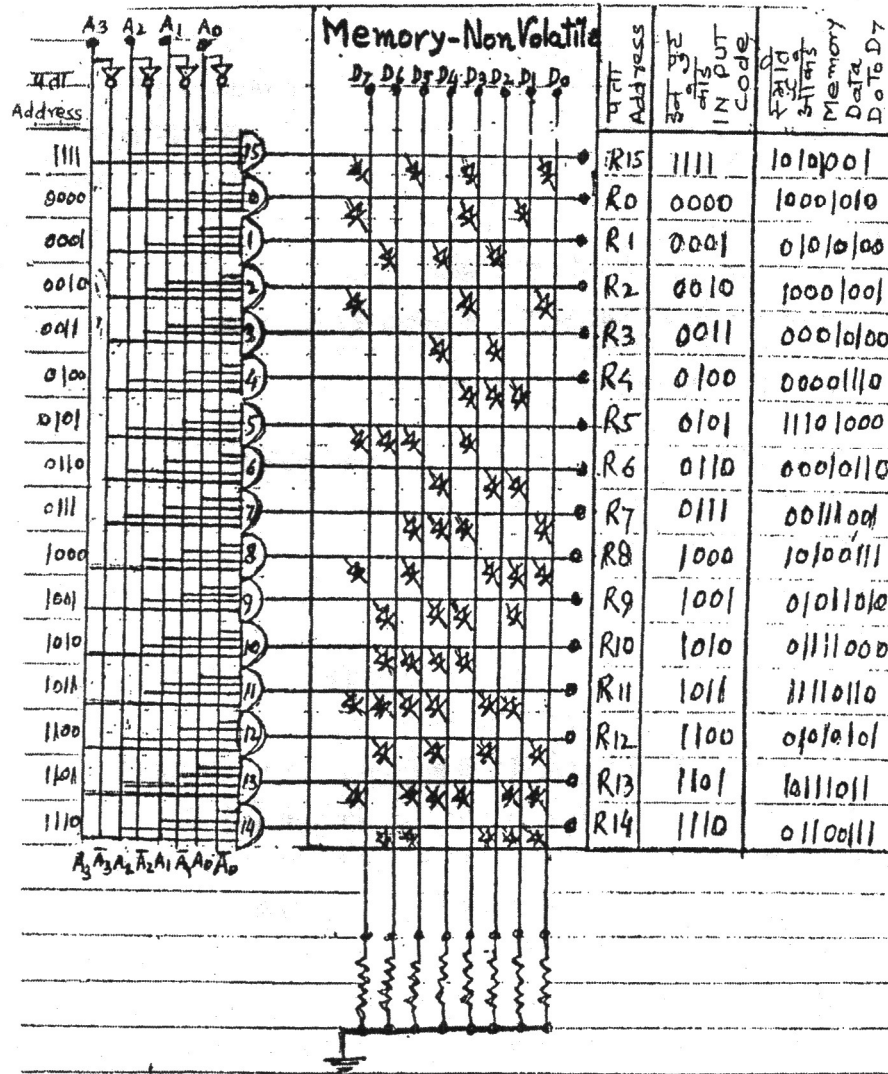
रीड ओनली मेमोरी (Read Only Memory)

ROM

ये मैमोरी स्थाई मैमोरी होती है। इनमें डाटा निर्माता द्वारा; उपभोक्ता के निर्देशानुसार, निर्माण के समय भरा जाता है। इसे उपभोक्ता द्वारा समाप्त नहीं किया जा सकता केवल उपभोक्ता द्वारा पढ़ा जा सकता है। इसमें उपभोक्ता कोई परिवर्तन नहीं कर सकता अर्थात् पुनः प्रोग्रामेबिल नहीं है।

ROM-में संग्रहित एक बिट जो किसी भी लोकेशन पर की गयी है को पुनः परिवर्तित करने की आवश्यकता नहीं है। अतः बिट को फ्लिप-फ्लॉप में एकत्र करने की आवश्यकता नहीं है। इसलिए ROM केवल संयुक्त सर्किट (Combinational Circuit)

से बनाया जाता है। ROM एक कोड कन्वर्टर, डीकोडर एवं एनकोडर (Decoder and encoder) से बना होता है। चित्र में एक 16 शब्द/8 बिट वाला मैमोरी सर्किट



चित्र नं० 56 ROM-16 शब्द/8 बिट मैमोरी परिपथ

[76] ST-52

दिखाया गया है। चित्र 56 में A_0, A_1, A_2 एवं A_3 - "पता बिट" (Address bit) है। चित्र से हम देखते हैं कि डाटा $D_7, \text{High}(1)$ तब ही होगा जब $R_{15}, R_0, R_2, R_5, R_8, R_{11}, R_3$, में से किसी भी डायोड की इनपुट high (उच्चतल) (1) पर होगी। यह तब ही संभव होगा जब A_0, A_1, A_2, A_3 से उचित बिट अर्थात् पता (Address) दिया जायेगा। इसी प्रकार D_6 से D_0 के बारे में स्पष्टीकरण दिया जा सकता है। अर्थात् इच्छानुसार किसी रजिस्टर विशेष में किसी भी आंकड़े को डायोड लगा कर या न लगा कर उच्च तल या निम्न तल लॉजिक (high or low logic) में कर कुछ भी आंकड़े लिखे जा सकते हैं। लेकिन यह केवल निर्माण के समय ही हो सकता है। बाद में इसे बदला नहीं जा सकता। अर्थात् स्थाई रूप से एक बार ही भरी जा सकती है।

रीड ओनली मेमोरी ROM तीन प्रकार की होती है।

(क) **प्रोग्रामेबल रीड ओनली मेमोरी (Programmable Read only Memory PROM)**

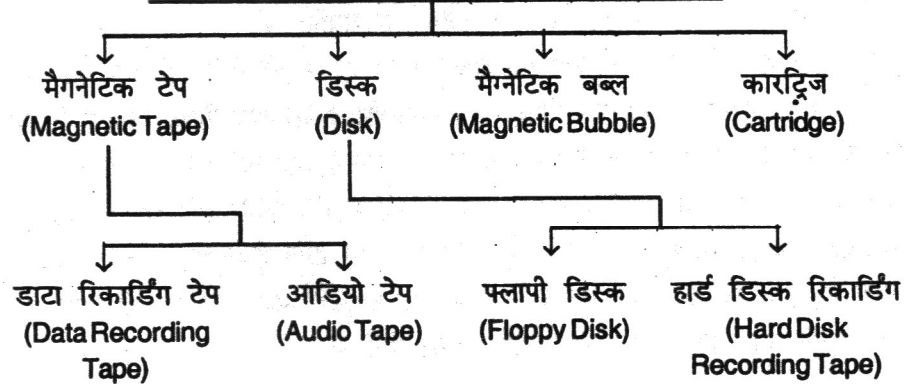
:- कुछ ROM चिप्स इस प्रकार के भी उपलब्ध हैं जिसमें उपभोक्ता स्वयं प्रोग्राम भर सकता है लेकिन एक बार प्रोग्राम भरने के बाद मिटा नहीं सकता इस ROM को PROM कहते हैं।

(ख) **विलेखन योग्य-प्रोग्राम योग्य पठन मात्र स्मृति (Erasable Programmable Read Only Memory) :-** कुछ मैमोरी मिटा कर पुनः लिखी भी जा सकती है। अतः उन्हें EPROM कहते हैं। मिटाने का कार्य अल्ट्रा वायलेट तरंगों द्वारा किया जाता है। ये तरंग उच्च ऊर्जा वाली होती है। लेकिन इस प्रकार के चिप मैमोरी में एक दोष यह है कि इसमें चिप की एक लोकेशन या ए भाग को नहीं मिटाया जा सकता वरन् पूरे चिप की मैमोरी को ही मिटाना पड़ता है। चिप पर अल्ट्रा वायलेट तरंगे 20 से 30 मिनट तक डालनी होती है। यद्यपि इस प्रकार के मैमोरी चिप पर लिख एवं मिटा तथा फिर से लिख सकने पर भी यह पठन मात्र स्मृति ही कहलाती है, क्योंकि स्मृति मिटाने तथा पुनः लिखने का कार्य उस समय नहीं किया जा सकता जबकि यह चिप डिजिटल सर्किट में कार्यरत हो। मैमोरी को सिस्टम से निकालना पड़ता है तथा मैमोरी में कोई भी परिवर्तन करने में घंटों का समय लगता है।

(ग) इलेक्ट्रिकली इरेजेबल प्रोग्रामेबल रीड ओनली मेमोरी (**Electrically Erasable Programmable Read Only Memory (EEPROM)**): इसमें विलेखन का कार्य विद्युत द्वारा किया जाता है। इसमें पूरी चिप मेमोरी को मिटाना आवश्यक नहीं है। चयन कर किसी भी लोकेशन विशेष को मिटाया जा सकता है। लेकिन यह चिप काम करने में समस्या पैदा करता है। एक महंगा भी है। इसलिए सामान्यतः इसका प्रयोग नहीं किया जाता है।

आक्जलरी मेमोरी (Auxiliary Memories)

(अधिक क्षमता वाली मेमोरी **Mass Capacity Memories**)



(1) **मैग्नेटिक टेप** :- घरेलू कम्प्यूटरों में अच्छी क्वालिटी की आडियो टेप का प्रयोग रिकार्डिंग के लिए किया जाता है। लेकिन बड़े कम्प्यूटरों के लिए बड़ी रील उपलब्ध है। एक विशेष प्रकार की रील का आकार (10x5 इंच) 2400 फिट लंबी होती है एवं चौड़ाई ½ इंच होती है इसमें साथ-साथ (चौड़ाई में) 9 बिट लिखे जा सकते हैं। इस प्रकार की बहु ट्रैक में लिखने एवं पढ़ने के लिए उतने ही पठन/लेखन हैड (Read and Write Heads) लगाने होंगे। लंबाई में टेप पर एक ट्रैक में बिट घनत्व (Bit density) 1600 बिट (Bit) प्रति इंच है।

[78] ST-52

कम्प्यूटर टेप की गति = 45 इंच प्रति सैकेंड

(लेखन/पठन) = $45 \times 1600 = 72$ हजार बिट/सै.

अतः यदि टेप के एक सिरे की सूचना से दूसरे सिरे तक की सूचना चाहिए तो कुल लगा समय = $2400 \times 12 / 45 \times 60 = 10$ मिनट

टेप में सही प्रकार से पढ़ने एवं लिखने के लिए आवश्यक है कि टेप सही गति से घूम रही हो। जब टेप को अवॉन्डित सामग्री को शीघ्र आगे बढ़ाने के लिए टेप को असामान्य गति से घुमाया जा रहा हो तो टेप न पढ़ी/लिखी जाये/इस कार्य के लिए टेप में मध्यवर्ती ब्लॉक बनाये जाते हैं।

(2) **चुम्बकीय डिस्क Magnetic Disk** :- मैग्नेटिक डिस्क स्मृति में फोनेग्राफ जैसी डिस्क पर समकेन्द्रक (Concentric) ट्रैकों पर संग्रहित की जाती है। अंतर केवल यह है फोनेग्राफ में ट्रैक एक बंद वृत्त होता है। जो एक दूसरों से अलग होता है। जैसा टेप मैमोरी में सूचनाओं का पठन/लेखा कार्य पठन/लेखन हैड द्वारा किया जाता है डिस्क की वृत्ताकार ज्योमिति होने के कारण इसे लगातार घुमाएं रखा जाता है चाहे लेखन/पठन कार्य हो रहा हो या न हो रहा हो। इससे लाभ यह होता है वेगवृद्धि या वेग ह्रास (Acceleration and Deceleration) के कारण जो समय एवं स्थान में हानि होती थी उसकी बचत हो जाती है। ट्रैक के साथ बिट घनत्व (Bit density) 500 से 600 बिट प्रति सै. होती है। तथा घूमने की गति 3600 चक्कर प्रति मिनट होती है।

बाजार में बहुत से प्रकार की डिस्क स्मृतियां उपलब्ध हैं। कुछ डिस्क में एक ओर रिकार्डिंग होता है किन्हीं में दोनों ओर, कुल रिकार्डिंग सतह 24 तक हो सकती है। जब पठन/लेखन के लिए एक ही पठन/लेखन हैड का प्रयोग किया जा रहा हो तब यूनिट में हैड को एक ट्रैक से दूसरे ट्रैक पर ले जाने का प्रावधान होना चाहिए। सतह पर ट्रैक एक दूसरे से 10 मिल (mil) दूर होने चाहिए तथा उनकी चौड़ाई 3.5 से 5 मिल तक होनी चाहिए। (1mil = $1/1000$ Inch)।

डिस्क के किसी विशेष भाग तक पहुंचने के लिए पहले हैड को उचित ट्रैक पर ले जाना चाहिए तथा उसके बाद उस विशेष स्थान के आने की प्रतीक्षा करनी चाहिए। यह कुल समय लगभग 30 मि. सै. होता है।

डिस्क पर ट्रैक घनत्व = 200 ट्रैक/इंच

बिट घनत्व/ट्रैक = 400 बिट/इंच

इस प्रकार डिस्क की बिट घनत्व सैकड़ों करोड़ बिट होती है।

(3) **मैग्नेटिक बबल Magnetic Bubble Memory** :- इस प्रकार की मेमोरी में प्लैट थिन फिल्म (सपाट पतली झिल्ली) के रूप में सूचनाएं एकत्र की जाती हैं। इस झिल्ली को इस प्रकार चुम्बकित किया जाता है कि लघु चुम्बकीय ध्रुवित क्षेत्र (Tiny Magnetically Polarised Region) बन जाता है। इस क्षेत्र को बबल कहते हैं डाटा (0 या 1) इन्हीं बबल के होने या न होने से निर्मित किए जाते हैं।

गुण :- (1) यह उच्च ताप, अधिक कंपन, धूल एवं आर्द्रता पर कार्य कर सकती है।

(2) संग्रह घनत्व (Storage Density) अधिक है।

(3) ये स्मृतियां अवाष्पशील (non-volatile) है।

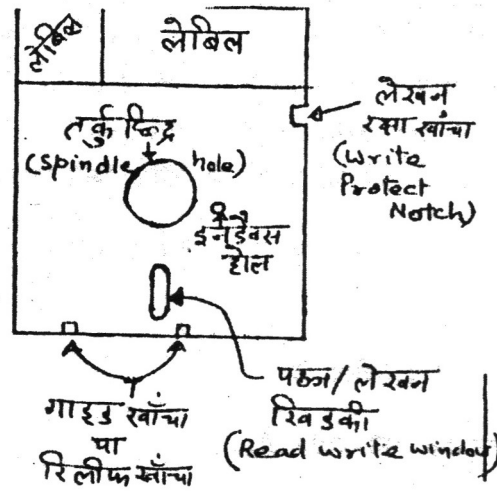
(4) **फ्लॉपी डिस्क Floppy Disk** :- टेप मेमोरी में यदि एक सिरे से दूसरे सिरे तक पहुंचना है तो टेप को रिवाइन्ड करना होगा जिसमें मिनटों समय लग जाएगा परन्तु यदि डिस्क की मेमोरी है तो हैड को बिना समय गवाएं तुरन्त उसी ट्रैक पर ले आया जाता है तथा कुछ ही मिली सै. में वांछित स्थान पर पहुंच जाते हैं। लेकिन डिस्क घुमाने वाली यान्त्रिक बहुत नाजुक (Sophisticated) तथा महंगी है।

इससे कम महंगी डिस्क मेमोरी भी बाजार में उपलब्ध है जिस फ्लक्सबिल डिस्क-स्टोरेज सिस्टम या केवल फ्लॉपी डिस्क सिस्टम कहते हैं। फ्लॉपी डिस्क सिस्टम में डिस्क तथा हैड का यान्त्रिक सम्पर्क ठीक उसी प्रकार बनता है जिस प्रकार टेप एवं हैड का टेप सिस्टम में। इसलिए डिस्क विशेष रूप से कठोर (Regid) नहीं होती। फ्लॉपी डिस्क की घूमने की गति

= 300 चक्कर प्रति मिनट

[80] ST-52

ट्रैक की संख्या = 77
ट्रैक घनत्व = 48 ट्रैक प्रति इंच
धारिता = 30 करोड़ प्रति सतह
डिस्क पहुंच समय = कुछ सौ मिली सै.



चित्र नं० 57 : फ्लोपी डिस्क

डिस्क का प्रकार

- (i) एक ओर एकल घनत्व (Single Side Single Density)
- (ii) एक ओर दोहरी घनत्व (Single Side Double Density)
- (iii) दोहरी ओर दोहरी घनत्व (Double Side Double Density)
- (iv) दोहरी ओर चार गुणी घनत्व (Double Side Quad Density)

डिस्क का आकार

- (i) 18" व्यास वाली डिस्क

(ii) 5¼" व्यास वाली डिस्क (मिनी फ्लॉपी डिस्क)

(iii) 3¼" व्यास वाली डिस्क (माइक्रो फ्लॉपी डिस्क)

डिस्क को आसानी से प्रयोग करने के लिए एवं धूल व निशान आदि पड़ने से बचाने के लिए स्थाई रूप से एक प्लास्टिक खोल, जिसे कार्ट्रेज कहते हैं, में सील कर दिया जाता है तथा पूरी डिस्क ऐसी लगती है जैसे फोनोग्राफ रिकार्ड को एक खोल में सील कर दिया गया हो। डिस्क के केन्द्र पर एक छिद्र है जिसे इस छिद्र की सहायता से यह स्पिन्दल पर फिट की जाती है। डिस्क के सभी भाग चित्र 57 में दिखाये गये हैं। जो अपना वर्णन स्वयं करते हैं।

फ्लॉपी डिस्क के लाभ

(1) सस्ती (2) धारिता अधिक (3) आकार छोटा (4) भार में हल्की (5) लाने-ले जाने में आसान।

फ्लॉपी डिस्क सिस्टम के हानि :-

- (i) धूल से वाष्पशीलता
- (ii) इसे घुमाने के लिए विशेष प्रकार के डिस्क ड्राइव की आवश्यकता होती है जो कुछ महंगा है।

नोट: आजकल फ्लॉपी डिस्क का प्रयोग नहीं होता है।

(5) हार्ड डिस्क (Hard Disk)

यह फ्लॉपी डिस्क की भांति लचीली नहीं है तथा कठोर सतह की बनी होती है। यह एक मशीन का ही भाग है। इसे फ्लॉपी डिस्क की भांति एक स्थान से दूसरे स्थान ले जाया नहीं जा सकता इसकी कुछ सूचनाएं इस प्रकार हैं।

- | | |
|---------------------------|---------------------------|
| (i) घूमने की गति | = 3600 चक्कर प्रति मिनट |
| (ii) ट्रैकों की संख्या | = 500 या अधिक |
| (iii) टैक घनत्व | = 200 प्रति इंच |
| (iv) धारिता | = 300 करोड़ बिट प्रति सतह |
| (v) हार्ड डिस्क पहुंच समय | = कुछ दहाई मिली सैकिन्ड |

[82] ST-52

हार्ड डिस्क के लाभ :

- (i) सूचनाएं सीधे जाती हैं।
- (ii) धारिता बहुत अधिक

हार्ड डिस्क के दोष :

- (i) अधिक खर्चीली
- (ii) खराब होने से बचाने के लिए प्रहस्तन (Handling) सावधानी पूर्वक करनी पड़ती है।

वैनचिस्टर डिस्क : इन्हें वाष्प से बचाने के लिए सन्मुद्रित (Hermetically Sealed) कर दिया जाता है।

वैद्युत डिजिटल घड़ी

Electric Digital Clock

वृहत् समाकलित सर्किटों (Large Scale Integrated Circuits) (LSIs) का उपयोग कर डिजिटल सर्किटों में जटिल वायरिंग आदि को कम किया जा सकता है आजकल ऐसी LSI बाजार में उपलब्ध है जिनकी सहायता से एक उपकरण की पूर्ण सिस्टम का निर्माण किया जा सकता है। इसी प्रकार का घरेलू उपयोग का एक साधन विद्युत डिजिटल घड़ी LSI की सहायता से बनायी जा सकती है।

एक डिजिटल घड़ी जो घंटे, मिनट तथा आवश्यक होने पर सैकिन्ड भी दर्शाए एक 24 पिन LSI प्रकार की MM5311 या MM5314 से बनायी जा सकती है। इन चिप में यह सुविधा उपलब्ध है कि हम (i) 4 डिजिट डिस्प्ले, दो घंटों के लिए एव दो मिनट के लिए या 6 डिजिट डिस्प्ले (दो अतिरिक्त डिजिट सैकिन्ड के लिए) (ii) 12 घंटे वाला फोरमैट (आरूप) जैसा सामान्य घड़ियों में होता है या 24 घंटे वाला आरूप (जैसा विमानन (Aviation) आदि में प्रयोग होता है) ले। 12 घंटे वाले आरूप (फोरमैट में) पहले 10 घंटों (1 बजे से 9 बजे तक) में प्रारम्भिक शून्य बुझी रहती है जबकि उसकी आवश्यकता नहीं होती परन्तु 24 घंटे वाले फोरमैट में यह शून्य आवश्यकतानुसार

जलती रहती है।

यह चिप किसी भी 11 वोल्ट से 19 वोल्ट तक की पावर सप्लाय पर कार्य कर सकता है। अर्थात् इसे रेग्युलेटिड पावर सप्लाय की आवश्यकता नहीं है टाइम सैट करने के लिए दो नियन्त्रण एक तेज गति (50 मिनट/सै.) एक दूसरा मंद गति (1 मिनट/सै.) वाला लगा होता है।

डिसप्ले आउटपुट को मल्टीप्लैक्स किया जाता है चिप के अंदर एक मल्टीप्लेक्स आसीलेटर लगा होता है। जिसमें आवृत्ति सैट करने के लिए एक संघनित्र—एक प्रतिरोध की आवश्यकता होती है। इसमें मल्टीप्लैक्सिंग टाइम डिविजन प्रकार की होती है, जिसमें सभी डिसप्ले को चलाने के लिए एक काल सहभाजन विधि (Time Shared Mode) में उपयोग किया जाने वाला सिंगल 7 सैगमैन्ट डिकोडर ड्राइवर प्रयोग किया जाता है। प्रत्येक डिसप्ले होने वाली डिजिट को कतार में (एक समय में एक) सिंगल 7 सैगमैन्ट डिकोडर ड्राइवर पर पहुंचना होता है। ताकि एक ही डिकोडर सभी डिजिटों को एक—एक करके समयानुसार सही तरीके से चला सके। जिस समय डिकोडर जिस डिजिट के लिए कार्यरत है वही डिजिट जलेंगी अन्य बुझी रहेगी। स्विचन कार्य इतनी तीव्र गति से होता है कि हमारी आंखें उसका अनुभव नहीं कर पाती।

इस प्रकार हम देखते हैं कि मल्टीप्लैक्सिंग विधि का प्रयोग कर कनेक्शनों की संख्या बहुत कम हो गई। इस अवस्था में 42 कनेक्शनों (7 प्रति डिजिट) के स्थान LSI में कुल 13 कनेक्शन लगा कर घंटे, मिनट एवं सैकिन्ड डिसप्ले किए जाते हैं।

मल्टीप्लैक्स हुए 7 सैगमैन्ट को इनेबिल/डिसेबिल (Enable/Disable) करने के लिए एक नियंत्रण प्रदान किया होता है। डिसेबिल वाली अवस्था में घड़ी कार्य करती रहेगी लेकिन 7—सैगमैन्ट बुझा रहेगा ताकि पावर की बचत की जा सके क्योंकि LSI चिप की अपेक्षा डिसप्ले में पावर खपत अधिक होती है। जब समय देखने की आवश्यकता हो एक पुश बटन दबा कर देखा जा सकता है। इसके अतिरिक्त एक “होल्ड” गणक नियंत्रण (Count Control) भी लगाया जाता है। जिसकी सहायता से समय (डिसप्ले) को होल्ड करा सकते हैं। इस होल्ड से घड़ी को किसी मानक (Standard) घड़ी से

[84] ST-52

तुल्यकाली (Synchronising) कर सकते हैं।

घड़ी ए.सी. मेन्स पर कार्य करती है इसी सप्लाई को आवृत्ति को मानव आवृत्ति स्रोत (Standard Frequency Source) माना गया है। अर्थात् यह 50 Hz या 60 Hz दोनों प्रकार की आवृत्ति वाली सप्लाई पर कार्य कर सकती है। यह घड़ी आवृत्ति विभाजन (Frequency Division) के सिद्धान्त पर कार्य करती है। इसमें प्रारम्भिक (Initial) विभाजन लाइन आवृत्ति के अनुसार 50 या 60 से विभाजित करने से प्रारंभ होता है इससे सैकिन्ड की पल्स प्राप्त होती है। इस पल्स को पुनः 60 से विभाजित कर मिनट की एवं फिर 60 से विभाजित कर घंटे को पल्स प्राप्त कर लेते हैं। इसके बाद अन्तिम विभाजन 12 या 24 से फारमेट के अनुसार किया जाता है। सम्बन्धित पल्स मल्टीप्लेक्सर के सहयोग से एवं सिंगल 7 सैगमेन्ट डिकोडर ड्राइवर को सहायता से घंटे, मिनट एवं सैकिन्ड का डिसप्ले देते हैं। चित्र नं. (59) में 1.25 से.मी. उभय एनोड 7-सैगमेन्ट LED डिसप्ले (जिसका प्रयोग किया जाता है) के पिन विन्यास (Pin Configuration) दिए गये हैं।

चित्र नं. (60) में घड़ी का सम्पूर्ण सर्किट दिया गया है इसका पावर सप्लाई खण्ड एक 500 मिली. ऐम्पी, 230/12V या 230/6-0-6V एक डायोड D1, एक इलैक्ट्रोलाइट फिल्टर संघनित्र C_1 , प्रतिरोध R_1 तथा C_2 से मिल कर बना है। R_1 व C_2 एक साधारण RC फिल्टर बनाता है जो लाइन वोल्टता के क्षणिक (Transients) उतार-चढ़ाव को दूर कर सके जो घड़ी की गति में अंतर ला सकता है या चिप को खराब कर सकता है। R_2 , C_3 से मल्टीप्लैक्स आवृत्ति ज्ञात होती है। घड़ी में जो मान प्रयुक्त किए गए हैं उनसे मल्टीप्लैक्स आवृत्ति लगभग 1.1 MHz तक होती है जिससे पिलकर रहित डिसप्ले प्राप्त हो सके।

T_7 से होते हुए T_1 ट्रांजिस्टर अपने संबंधित प्रतिरोधों सहित सैगमेन्टों के लिए स्विचन सर्किट बनाते हैं। सभी समान सैगमेन्टों को साथ-साथ डिसप्ले के लिए बांधा जाता है तथा सामान्य लीड को संबंधित सैगमेन्ट स्विचन सर्किट से जोड़ जाता है ट्रांजिस्टर T_8 , T_{11} के साथ मिल कर घंटे एवं मिनट की डिजिट को बनने में सक्षम करता है।

सैकिण्ड का डिसप्ले प्राप्त करने के लिए दो और समान प्रकार के ट्रांजिस्टर की आवश्यकता पड़ेगी जिन्हें आई.सी. के पिन नं. 17 एवं 18 पर जोड़ा जाएगा। इसके अतिरिक्त पिन नं. 24 (416 डिजिट सलैक्ट), जो इस समय तक खाली पड़ा है (4 डिजिट आपरेशन में) को 6 डिजिट आपरेशन के लिए VC से जोड़ा जाएगा।

आई.सी. का पिन नं. 10 जब VOD से जुड़ा होता है (S1 स्विच से होकर) 12 घंटे वाले फोरमैट को बनाता है। यदि इसे खाली छोड़ दिया जाये तो 24 घंटे वाला फोरमैट बनाता है स्विच S_2 , S_3 , S_4 , से होकर क्रमशः पिन नं. 15, 14, 13 को ताकि एक समय में एक बार घड़ी में टाइम सैट करने की सुविधा प्राप्त की जा सके।

उपरोक्त वर्णन में जो चित्र प्रयोग किया गया है वह IC No 5314 के लिए हैं यदि आई सी नं. 5311 का प्रयोग करना है तो पिन कनेक्शनों में परिवर्तन कर बनाई जा सकती है। इन दोनों चिप्स में अंतर केवल यह है कि 5311 चिप में 4 अतिरिक्त BCD (बाइनरी कोडिड डेसीमल) मल्टीप्लैक्स वाली आउटपुट भी प्रदान की होती है जिनका उपयोग किसी अन्य कार्य में किया जा सकता है।

पावर फेल होने को जांचने वाला लैच (कब्जा) चित्र नं. (61) में दिया गया है। सामान्यतः जब पावर चालू की जाती है तो ट्रांजिस्टर T_{12} कार्य नहीं करता (कट आफ रहता है) इस अवस्था में T_{13} की कलेक्टर वोल्टता V_{58} के बराबर होगी तथा T_{14} के बेस पर जायेगी। क्योंकि सभी डेसीमल प्वाइन्ट एक दूसरे से बंधे हैं। तथा T_{14} से जुड़े हैं। वे जल जायेंगे। स्विच S_2 एवं S_3 को दबा कर घड़ी का सकय सैट करने पर, T_{12} संतृप्त हो जाता है। क्योंकि इनका बेस एमीटर जंक्शन D_2/D_3 तथा R_{24} से होकर फारवर्ड बायस हो जाता है। T_{12} का कलेक्टर विभव (Potential) VSS के बराबर होकर T_{13} को सन्तृप्त (Saturate) करता है। इससे T_{13} का कलेक्टर विभव कम लगभग 1 वोल्ट हो जाता है जो T_{14} को चालू करने में सक्षम नहीं है। अतः यह कट-आफ रहता है। यह स्थिति तब तक बनी रहेगी जब तक पावर में रूकावट नहीं आती। D_4 कोई भी सिलिकॉन डायोड (यहां पर BC 148B, BC 104 का बेस-एमीटर जंक्शन प्रयोग किया गया है) जो पावर के वापस आने पर T_{12} तथा T_{13} के कट-आफ रहने की स्थिति को सुनिश्चित करेगा।

[86] ST-52

घड़ी का निर्माण -

ट्रांसफार्मर सहित सभी सर्किट, 17.5 x 10.0 सेमी वाली डोट मैट्रिक बहुउद्देशीय (Multipurpose) प्रिन्टिड सर्किट बोर्ड जो आई.सी. के लिए उपयोगी हो, पर बनाया जा सकता है। इसे अपनी दक्षता के अनुसार उचित आकार में बनाया जा सकता है।

इसे कार रेडियो या टेप रिकार्ड के लिए कार में प्रयुक्त स्पीकर के कैबिनेट में बनाया जा सकता है। इसमें सभी अवयव लगाकर टेबिल मॉडल घड़ी बनायी जा सकती है।

इस कैबिनेट में घड़ी बनाने के लिए, PC प्रिन्टिड सर्किट बोर्ड को जहां स्पीकर फिट किया जाता है फिट करें। सामने वाले बैफिल पर निशान लगाकर इस प्रकार काट ले कि 4 डिसप्ले एवं LED D₅ को फिट किया जा सके जैसा घड़ी के सर्किट में दिखाया गया है। अब पी.सी. बोर्ड को बैफिल प्लेट के अंदर पेंच से कसा जाता है। अब पी. सी.वी. पर डिसप्ले तथा D₅ लगाने के लिए बैफिल स्लाट के बाहर से निशान लगायें। चित्र नं. (62) में पी.सी.बी. पर अन्य अवयव लगाने का स्थान दिया गया है उस पट्टी पर जहां कट-आफ लिखा है, डिसप्ले एवं D₅ लगाने का स्थान है।

सैगमेन्ट के सभी समान पिनों को एक साथ जोड़ा जाता है एवं एक उभयनिष्ठ लीड निकाली जाती है। आई.सी. पिन नं. 6, 7 तथा 5, 4 से निकाली लीडों को स्ट्रिप के बायीं ले जाते हैं। तथा पिन नं. 9, 10 तथा 2, 1 से आने वाली लीडों को बिल्कुल दायीं ओर ले जाते हैं।

यद्यपि ले-आउट इतनी जटिल नहीं है। फिर भी प्रयुक्त लीड छोटी से छोटी होनी चाहिए।

घड़ी चिप 5311 या 5314 पर कार्य करते समय बरती जाने वाली सावधानियां :-

जैसा कि हम जानते हैं कि IC No. 5311 या 5314 MOS (मैटल आक्साइड सेमीकन्डक्टर) आई.सी. है, जो स्थिर आवेश (Static charge) से खराब हो सकती है। अतः चिप को खराब होने से बचाने के लिए निम्न सावधानी बरती जाती है।

(i) कार्य करते समय सूती कपड़े पहने क्योंकि पोलिस्टर, टेरीलीन आदि शीघ्र आवेशित हो जाते हैं।

(ii) पूरी तरह से अर्थ किया सोल्डरिंग आयरन प्रयोग करें तथा आई.सी. को सीधे पी.सी.बी पर सोल्डर न करें, सॉकिट का प्रयोग करें।

(iii) घड़ी चिप को प्रोटेक्टिव फोम पैकेज से तब तक न निकाले जब तक आवश्यकता न हो। आई.सी. के पिनों को अंगुली से स्पर्श न करें, आई.सी.को सर्किट से निकालने या लगाने के लिए आई.सी. निवेशक/निवासक (Insertor/Extractor) का प्रयोग करना चाहिए।

(iv) आई.सी को गिराना नहीं चाहिए।

(v) जब तब यह सुनिश्चित न हो जाये की सभी कनेक्शन ठीक है आई.सी. को सर्किट में न लगाया जाए।

पार्ट लिस्ट

ICI - घड़ी चिप MM 5314 या MM 5311 चिप या	= 1 No
T ₁ से T ₇ -- एवं T ₁₄ ट्रांजिस्टर BC 148B	= 8 No
T8 से T11 -- ट्रांजिस्टर BK 100 या SK 100	= 4No
D ₂ , D ₃ -- डायोड IN 66 या कोई सामान्य कार्य के लिए जर्मेनियम डायोड	= 2No
D4 -- कोई सामान्य कार्य वाला सिलिकान डायोड	= 1No
डिसप्ले 1 से डिसप्ले 4--FND 507कामन एनोड LED डिसप्ले	= 4 No
L ₅ -- कोई भी लाल LED	= 1 No

[88] ST-52

X_1 - 230 डिसप्ले प्राइमरी / 12 वोल्ट सैकिन्डरी, 500 मिली एम्प.

या 230 वोल्ट प्राइमरी / 6V-0-6V सैकिन्डरी 500 मि. एम्पीयर

करंट वाला ट्रांसफार्मर

= 1 No

C_1 , — 500 माइक्रो फैरेड इलैक्ट्रोलाइटिक कैपेसिटर

= 1 No

C_1, C_3 — 0.01 MF कैपेसिटर

= 2 No

R_1, R_2 — 100 K, ½ वाट प्रतिरोध

= 2 No

R_3 से R_9, R_{25}, R_{27} , — 2.7K, ¼ वाट प्रतिरोध

= 2 No

R_{10} से R_{16}, R_{24}, R_{26} , 5.6 ¼ वाट प्रतिरोध

= 9 No

R_{17} से — 470, ¼ वाट प्रतिरोध

= 8 No

R_{28} से — 470 ¼ वाट प्रतिरोध

= 1 No

R_{28} — 62 K ½ वाट प्रतिरोध

= 1 No

R_{30} — 1.5 K ½ वाट प्रतिरोध

= 1 No

S_1 — 1 पोल, 1 वे / 2 वे स्लाइड स्विच

= 1 No

S_2, S_3, S_4 — पुश टू आन स्विच

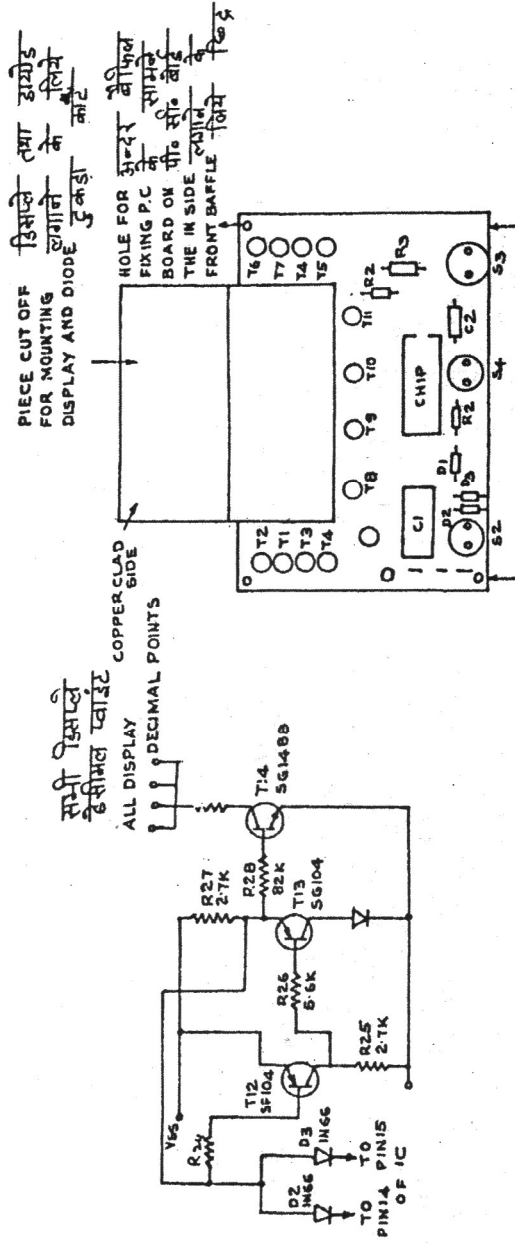
= 3 No

मिलेजुले पार्ट— वैक्टर बोर्ड, सामान्य कार्य वाला प्रिन्टिड सर्किट बोर्ड, कैबिनेट, आई, सी, सर्किट, पेंच, नेट, वोल्ट कनैक्टिंग वायर, फिनोलिक बोर्ड ट्रांसफार्मर आदि लगाने के लिए।

डिजिटल घड़ी का सिद्धांत

डिजिटल घड़ी के कार्य करने का सिद्धांत है—एक पल्स प्रति सैकिन्ड के परिशुद्ध स्रोत (Accurate repetition frequency source) के द्वारा जनित पल्स की गणना करना।

प्रति पल्स एक सैकिन्ड के अनुसार गणना की जाती है। तथा डिसप्ले किया जाता है।

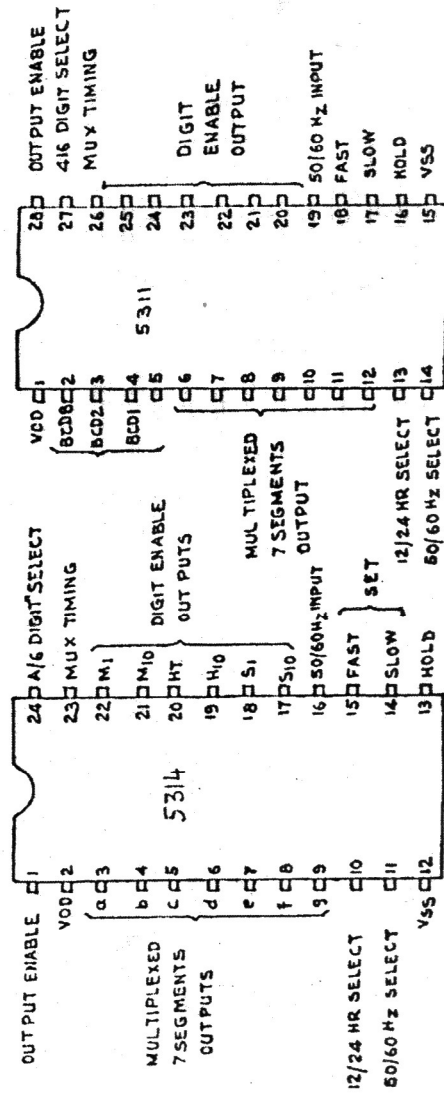


बैफिल से पी.सी.
बोर्ड लगाने
के लिये चिह्न

मुख्य पी.सी.
बोर्ड (प्रिना
कलेड वाली ओर से)

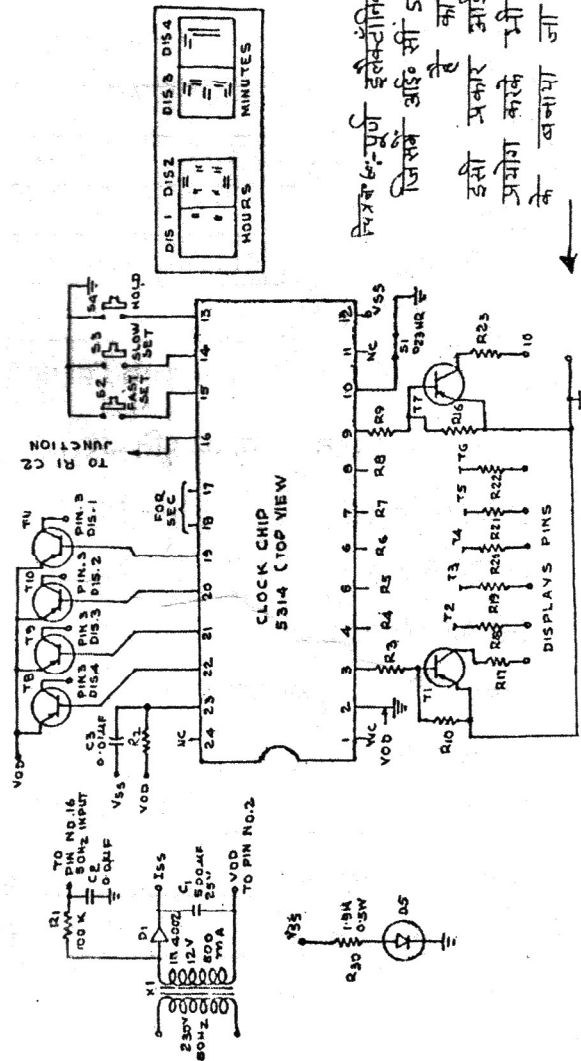
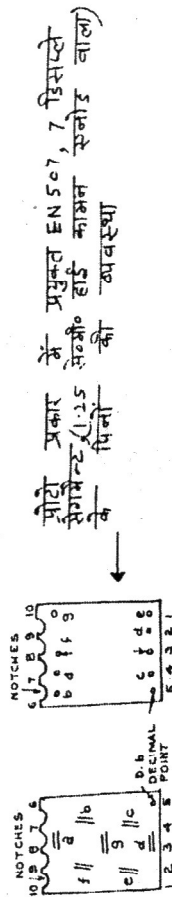
चित्र 62:- दाड़ी में प्रयुक्त लगभग सभी
अवयवों की ले आउट

चित्र 61:- पावर विफलता (फैल्यर) सूचक
परिपथ "जो समग्र सभी इसप्ले नहीं
भरी हो सकता" की चेतावनी देने के लिये प्रयोग
होता है तथा दाड़ी के चालू होने या
बैबारा पावर के आने पर कार्य
करता है



चित्र: 58

यही बनाने के लिए आई० सी नं० 5314 या 5311 का प्रयोग किया जा सकता है।
 दोनों प्रकार की आई० सी० में पिनों की व्यवस्था चित्र में दिखायी गयी है।



चित्र 6-पूर्ण इलेक्ट्रॉनिक डिजिटल घड़ी
जिसमें अदि सी 5314 लगायी गयी
है का सर्किट परिपथ।
इसी प्रकार आई. सी. 5311 का
प्रयोग करके भी बिना कठनाई
के बनाया जा सकता है।

[92] ST-52

वस्तुनिष्ठ

क. निम्नलिखित का पूरा नाम लिखे

1. BIT _____
2. BCD _____
3. RAM _____
4. ROM _____
5. ASCII _____
6. DIP _____
7. MOS _____
8. SIPO _____
9. EEPROM _____
10. ALU _____

ख. निम्नलिखित रिक्त स्थानों की पूर्ति करे।

1. $(132)_{10} = (\quad)_2$
2. $(1520)_8 = (\quad)_2$
3. $(EFA5)_{16} = (\quad)_2$
4. $(11101110)_2 = (\quad)_{10}$
5. NAND गेट का बूलीय समीकरण होता है।
6. $A+A.B = \dots\dots\dots$
7. फुल ऐडर बिटों को जोड़ सकता है।

8. एक किलो बिट में बिट होती है।
9. हैक्साडेसीमल में कुल डिजिट होते हैं।

ग. निम्नलिखित में सही (✓) पर तथा गलत (X) पर का निशान लगाएं।

1. RAM वॉलेटाइल मैमोरी है। ()
2. EEPROM एक नॉन वॉलेटाइल मैमोरी है। ()
3. ट्रांजिस्टर को एम्प्लीफायर तथा स्विच की तरह प्रयोग किया जा सकता है। ()
4. EX-OR गेट में आउटपुट तभी हाई होगी जब सभी इनपुट हाई होंगी। ()
5. एक निबल (Nibble) में चार बिट होती है। ()
6. यूनिकोड कोडिंग सिस्टम में 32 बिट होते हैं जिसके द्वारा किसी भी भाषा तक के संकेत की कोडिंग की जा सकती है। ()
7. रैम मैमोरी में रीड तथा राइट किया जा सकता है। ()
8. फ्लिप-फ्लॉप का प्रयोग रजिस्टर, काउन्टर तथा मैमोरी में होता है। ()