

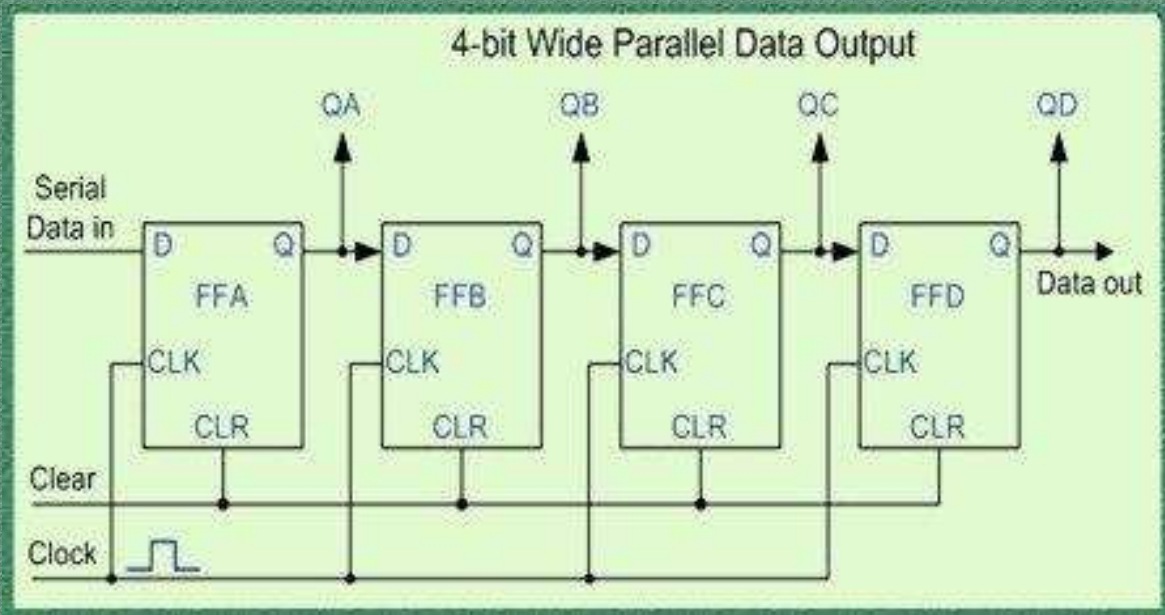
# इरिसेट



# IRISET

## टी.बी.3

## डिजिटल इलेक्ट्रॉनिक्स



भारतीय रेल सिग्नल इंजीनियरी और दूरसंचार संस्थान  
सिकंदराबाद-500017

# टी.बी.3

## डिजिटल इलेक्ट्रॉनिक्स

**दर्शन** : इरिसेट को अंतर्राष्ट्रीय प्रसिद्धि का संस्थान बनाना, जो कि अपने मानक व निर्देशचिह्न स्वयं तय करे.

**लक्ष्य** : प्रशिक्षण के माध्यम से सिगनल एवं दूरसंचार कर्मियों की गुणवत्ता में सुधार तथा उनकी उत्पादक क्षमता में वृद्धि लाना.

इस इरिसेट नोट्स में उपलब्ध की गई सामग्री केवल मार्गदर्शन के लिए प्रस्तुत की गयी है. इस नियमावली या रेलवे बोर्ड के अनुदेशों में निहित प्रावधानों को निकालना या परिवर्तित करना मना है.



भारतीय रेल सिगनल इंजीनियरी और दूरसंचार संस्थान

सिकंदराबाद - 500 017

## टी.बी.3 डिजिटल इलेक्ट्रॉनिक्स

### विषय सूची

क्र.सं.	अध्याय	पृष्ठ सं.
1	डिजिटल इलेक्ट्रॉनिक्स	1
2	लॉजिक गेट	9
3	संयोजन लॉजिक सर्किट्स	19
4	फ्लिप फ्लॉप	25
5	अनुक्रमिक लॉजिक सर्किट	33
6	मेमोरी	40
7	डिजिटल कोड	44
8	डिजिटल लॉजिक परिवार	51
9	माइक्रोकंट्रोलर - 8051	58

1. पृष्ठों की संख्या - 41
2. जारी करने की तारीख - जून - 2015
3. हिंदी और अंग्रेजी संस्करण में कोई विसंगति या विरोधाभास होने पर इस विषय का अंग्रेजी संस्करण ही मान्य होगा.

© IRISSET

“यह केवल भारतीय रेलों के प्रयोगार्थ बौद्धिक संपत्ति है. इस प्रकाशन के किसी भी भाग को इरिसेट, सिकंदराबाद, भारत के पूर्व करार और लिखित अनुमति के बिना न केवल फोटो कॉपी, फोटो ग्रॉफ, मेग्नेटिक, ऑप्टिकल या अन्य रिकार्ड तक सीमित नहीं, बल्कि पुनः प्राप्त की जाने वाली प्रणाली में संग्रहित, प्रसारित या प्रतिकृति तैयार नहीं किया जाए.”

<http://www.iriset.indianrailways.gov.in>

# अध्याय 1

## डिजिटल इलेक्ट्रॉनिक्स

### 1.1 परिचय:

वर्तमान दुनिया में, डिजिटल इलेक्ट्रॉनिक्स, वह आधुनिक तकनीक है, जिसका उपयोग मानव जीवन के अधिकतर कार्यों में होता है। हम लोग सामान्य कैलकुलेटर से लेकर तेज अंतरिक्ष उपग्रहों के अध्ययन तक के प्रत्येक क्षेत्र में इसका उपयोग करते हैं। इलेक्ट्रॉनिक्स, इस दुनिया में लगभग सर्वव्यापी हो चुकी है। इलेक्ट्रॉनिक्स में, मुख्यतया, दो शाखाएं हैं।

- क) एनलॉग इलेक्ट्रॉनिक्स तथा
- ख) डिजिटल इलेक्ट्रॉनिक्स

पहले हम, एनलॉग तथा डिजिटल पदों की परिभाषा के बारे में समझेंगे।

### 1.2 एनलॉग क्या है?

एनलॉग पद का संबंध, वह परिमाण से है, जिसका स्वभाव लगातार है और उनमें परिवर्तन, मूल्यों की एक सीमा के अंदर एक क्रमिक ढंग से होता है। इसका दूसरा गुण यह है कि एनलॉग परिमाण वास्तविक परिमाण होता है।

इलेक्ट्रॉनिक्स में, जो सिस्टम एक सीमा के अंदर क्रमिक ढंग से बदलते विद्युत सिगनलों की संसाधन करने में सक्षम है, उसको “एनलॉग सिस्टम” कहलाता है। एक एनलॉग परिमाण, वोल्टेज या धारा से प्रतिनिधित्व हो सकती है, जो परिमाण के मूल्यों के समानुपात हो।

सभी साधारण परिमाण, जो वास्तविक दुनिया में उपयोग में हैं, एनलॉग स्वभाव के हैं। प्रकृति में मौजूद कुछ एनलॉग परिमाण के उदाहरण ये हैं:

- क) दिन और रात का तापमान
- ख) वायुमंडलीय दबाव
- ग) नहर में जल का प्रवाह
- घ) वर्ष के दौरान मौसम का परिवर्तन

हम लोग, आम दिनचर्या में उपयोग करने वाले विभिन्न उपकरण और यंत्र में भी एनलॉग परिमाण देख सकते हैं। जैसे:

- क) ऑटोमोबाइल के त्वरक
- ख) रेडियो या टीवी के ध्वनि नियंत्रण
- ग) ट्रेन यातायात

### 1.3 डिजिटल क्या है?

एनलॉग से भिन्न, परिमाण या सिस्टम, जो केवल खास निर्धारित मूल्यों या अवस्थाओं को ही मानते हैं, उसे ‘डिजिटल’ कहलाता है। इसका मतलब यह है कि डिजिटल परिमाण का मान लगातार नहीं बदलता है, बल्कि यह कुछ निश्चित या अनिरंतर मूल्यों के बीच आकस्मिक या चरणों में बदलता है।



जो सिस्टम अनिरंतर मूल्यों का संसाधन करता है, उसे डिजिटल सिस्टम कहलाता है। डिजिटल प्रतिनिधित्व में परिमाण को अनिरंतर मूल्यों से प्रतिनिधित्व नहीं करते हैं, बल्कि कुछ प्रतीक द्वारा करते हैं, जिसे डिजिट कहते हैं।

उदाहरण के लिए, डिजिटल घड़ी में घंटे और मिनट को डेसिमल अंकों के रूप में दर्शाते हैं। जैसे कि हम जानते हैं, दिन का समय लगातार बदलता है, लेकिन डिजिटल घड़ी में परिवर्तन लगातार नहीं है, बल्कि यह एक मिनट (या सेकण्ड) के चरण में बदलता है। दूसरे शब्दों में, एनलॉग घड़ी की तुलना में जहाँ सूई डायल पर लगातार समय परिवर्तन दर्शाता है, डिजिटल प्रतिनिधित्व में दिन का समय अनिरंतर चरणों में बदलता है।

कुछ उपकरण या परिमाण, जिसका व्यवहार या निर्माण डिजिटल रूप में है और हमारे दैनिक जिंदगी में उपयोग में आता है, उदाहरण के तौर पर नीचे दिया गया है।

क्र.सं.	परिमाण	निश्चित स्तर / कल्पित अवस्था
1	टोगल स्विच	ON तथा OFF
2	रिले	पिक-अप तथा ड्रॉप अवस्थाएं
3	दिया	जलता है या नहीं जलता है
4	फैन रेग्युलेटर	1 से 5 चरणों में गति
5	सीढ़ी/सीढ़ीनुमा	चरणों
6	ऑटोमोबाइल गियर सिस्टम	गति गियर

**तालिका 1.1 डिजिटल परिमाणों के उदाहरण**

#### 1.4 डिजिटल तकनीक के लाभ:

एनलॉग तकनीक की तुलना में डिजिटल तकनीक का कुछ फायदें निम्न प्रकार हैं।

- क) डिजिटल सिस्टम में प्रोग्रामिंग की सुविधा उपलब्ध है।
- ख) प्रोग्रामिंग द्वारा परिचालन को स्वचालित किया जा सकता है।
- ग) जानकारी का संग्रह की जा सकती है और ज़रूरत के अनुसार संसाधन कर सकते हैं।
- घ) जानकारी की आसान तथा प्रभावशाली संग्रह संभव।
- ङ) उच्च शुद्धता और सूक्ष्मता संभव।
- च) जानकारी की आसान संचरण तथा 100% शुद्ध पुनरुत्पादन संभव।
- छ) डिजिटल सर्किट, शोर से प्रतिरक्षित है, जब तक शोर के कारण उच्च अथवा कम अवस्था को पहचानने में बाधा न हो।
- ज) डिजिटल सर्किट की रचना बहुत आसान है, क्योंकि इसमें वोल्टेज या धारा की सही मूल्य का होना ज़रूरी नहीं है।

### 1.5 डिजिटल की सीमाएं:

वास्तविक दुनिया में, बहुत सारी भौतिक परिमाण एनलॉग प्रकृति की होती हैं और जब इसको डिजिटल सिस्टम द्वारा संसाधन करना होता है, तब पहले सिगनल को डिजिटल में परिवर्तित करना पड़ता है और कभी-कभी संसाधन के बाद इसको पुनः एनलॉग में परिवर्तित भी करना पड़ता है। इस कार्य के लिए एनलॉग से डिजिटल कन्वर्टर (ADC) तथा डिजिटल से एनलॉग कन्वर्टर (DAC) या CODEC का उपयोग करते हैं। इसलिए यह संभव है कि उपकरण द्वारा मूल सिगनल का परिवर्तन 100% शुद्ध न भी हो।

### 1.6 डिजिटल प्रणालियाँ:

अब यह स्पष्ट होना चाहिए कि इलेक्ट्रॉनिक्स में डिजिटल उपकरण या प्रणाली वह है जो केवल कुछ स्थिर वोल्टेज मान या अवस्था को प्रतिनिधित्व करता है।

अगर एक डिजिटल प्रणाली को ऐसे ज्यादा वोल्टेज स्तरों का संसाधन करना हो तब सर्किट का डिज़ाइन बहुत जटिल हो जाता है। क्योंकि ट्रांसिस्टर, MOSFETs, LEDs, रिले, स्विच आदि घटकों ज्यादा संख्या में वोल्टेज स्तरों में अंतर नहीं कर पाती हैं।

इसलिए, यह आसान और मुमकिन है कि डिजिटल सर्किट किसी भी दो स्थिर वोल्टेज मूल्यों पर ही काम करती है - एक निम्न वोल्टेज स्तर और एक उच्च वोल्टेज स्तर। यह दो वोल्टेज मूल्यों का प्रतिनिधित्व सामान्यतया 0V और Vcc (5V, 10V, 15V इत्यादि) से होती हैं।

### 1.7 0 और 1 का सिद्धांत:

बहुत से डिजिटल उपकरण गिनती और अंकगणित जैसे गणितीय प्रक्रिया में शामिल होते हैं। इस स्थिति में डिजिटल सिगनल, निम्न या उच्च वोल्टेज के संबंध में मुमकिन नहीं है। यह अस्पष्ट और अनुपयोगी प्रक्रिया की ओर बढ़ावा देता है।

इसलिए, इन डिजिटल सिगनलों का प्रतिनिधित्व करने के लिए एक वैकल्पिक पद्धति होना ज़रूरी बन गई। यह वैकल्पिक पद्धति अंकों के रूप में की गई, जिससे गणितीय कार्यों को सक्षम कर सकें।

इसके बाद, डिजिटल सिगनलों का प्रतिनिधित्व करने के लिए निम्न पद्धति को अपनाया गया।

निम्न वोल्टेज सिगनल ---- '0'

उच्च वोल्टेज सिगनल ---- '1'

डिजिटल सिगनलों का 0 और 1 के रूप में यह अंकन एक नयी संख्या पद्धति बनाने की ओर बढ़ गई, जिसमें केवल दो ही डिजिट (या चिह्न) हो, 0 और 1.

### 1.8 संख्या प्रणाली:

परंपरागत डेसिमल संख्या प्रणाली के अतिरिक्त, डिजिटल सर्किट निम्न दो नयी संख्या प्रणाली का भी उपयोग करती है।

1. बाइनरी संख्या प्रणाली
2. हेक्सा-डेसिमल संख्या प्रणाली

ये दोनों के विस्तार में जाने से पहले प्रत्येक संख्या पद्धति के लिए उपयोगी सामान्य सिद्धांत एवं नियमों के बारे में अध्ययन करते हैं। किसी भी संख्या पद्धति में, हिसाब में या गिनती में उपयोग के लिए कुछ प्रतीकों का एक समुच्चय होती है, जिसे हम डिजिट या अंक कहते हैं। डेसिमल प्रणाली में यह प्रतीक होते हैं 0, 1, 2, 3, 4, 5, 6, 7, 8 तथा 9. यह कुल 10 प्रतीक है। इसलिए इस 10 को डेसिमल प्रणाली के आधार कहलाते हैं।

इस 10 में से किसी भी अंकों का उपयोग करके किसी भी संख्या को दर्शाया जाता है, जैसे

25, 37, 105, 1145, आदि

वहीं संख्या, जब एक बहु अंकीय संख्या में अपना स्थान बदलता है, तब इसके स्थितीय गुणक मूल्य के अनुसार इसका मूल्य बदलता है। डेसिमल संख्या पद्धति में प्रत्येक अंक के लिए स्थितीय गुणक का क्रम नीचे दर्शाया गया है।

$$10^n, \dots 10^4, 10^3, 10^2, 10^1, 10^0 \cdot 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, \dots 10^{-n}$$

↑  
डेसिमल बिंदू

ऊपर की श्रृंखला में, स्थानीय गुणक का मान 10 की चरण में होता है, जो डेसिमल संख्या पद्धति का आधार होता है।

इस प्रकार, 347.45 का मान, स्थानीय मान के योगफल के बराबर होता है।

$$3 \times 10^2 + 4 \times 10^1 + 7 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2} \quad \text{जो देगा}$$

$$300 + 40 + 7 + 0.4 + 0.05 = \underline{\underline{347.45}}$$

यहीं नियम हर संख्या प्रणाली के लिए लागू हैं।

### 1.8.1 बाइनरी संख्या प्रणाली:

बाइनरी संख्या प्रणाली में केवल दो ही अंक उपलब्ध हैं, 0 तथा 1. इसका मतलब, इसके आधार 2 है। डेसिमल मूल्यों के समान बाइनरी संख्या निम्न तालिका में देखे जा सकते हैं।

डेसिमल	बाइनरी
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010

तालिका 1.2. डेसिमल संख्या तथा उसके समरूप बाइनरी

बाइनरी प्रणाली में, स्थितीय मल्टिप्लायरों का मूल्य निम्न प्रकार हैं।

$$2^n, 2^4, 2^3, 2^2, 2^1, 2^0, 2^{-1}, 2^{-2}, 2^{-3}, 2^{-4}, \dots 2^{-n}$$

किसी भी बाइनरी संख्या का मूल्य पता करने के लिए, हमको, संबंधित डिजिट के साथ उसकी स्थितीय गुणक की गुणा करना होता है।

उदाहरण के लिए: 1101 (बाइनरी में). इसका मूल्य जानने के लिए, नीचे दर्शाए अनुसार स्थितीय गुणक नियत करें।

1	1	0	1	- बाइनरी संख्या डिजिट में
$2^3$	$2^2$	$2^1$	$2^0$	- स्थितीय डिजिट गुणक

इसके बराबर है:  $1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$  यानी  $1 \times 8 + 1 \times 4 + 0 + 1 \times 1 = 13$

इसलिए बाइनरी 1101 के समान डेसिमल मूल्य होते हैं 13.

इस तरह किसी भी बाइनरी संख्या का डेसिमल मूल्य जाना जा सकता है। यह प्रक्रिया बाइनरी से डेसिमल में रूपांतरण देता है। इसी तरह, अगर एक डेसिमल संख्या दी जाती है तो उसकी बाइनरी समरूप "डबल डाबल" नामक एक विधि से निकाला जा सकता है। इस विधि में, दी हुई डेसिमल संख्या को 2 से विभाजित किया जाता है, जो बाइनरी का आधार है।

मान लीजिए, हम को 16 की बाइनरी चाहिए:

	भागफल	शेषफल		
16/2 =	8	0		अब, शेषफल को तीर (arrow) की दिशा में पढ़ें, यानी, 10000.
8/2 =	4	0		
4/2 =	2	0		
2/2 =	1	0		
1/2 =	0	1		

### 1.8.2 हेक्सा-डेसिमल प्रणाली:

हेक्सा-डेसिमल संख्या पद्धति का आधार 16 है। इसका मतलब, गिनती के लिए 16 डिजिट उपलब्ध हैं।

ये हैं - 0, 1, 2, 3, 4, ----- 9, A, B, C, D, E, F

स्थितीय गुणक हैं -  $16^n$ , -----,  $16^4$ ,  $16^3$ ,  $16^2$ ,  $16^1$ ,  $16^0$ ,  $16^{-1}$ ,  $16^{-2}$ , -----,  $16^{-n}$

इस संख्या प्रणाली का मुख्य उपयोग, बहुत लंबी बाइनरी मूल्यों को हेक्सा-डेसिमल अंकों द्वारा छोटे रूप में व्यक्त करने के लिए है।

**उदाहरण के लिए:** मान लें कि 110110110011, एक हेक्सा-डेसिमल संख्या है।

यह बाइनरी संख्या को, कम महत्वपूर्ण बिट (दायी तरफ से) से शुरू करके 4 बिट्स की समूह बनाकर, उसका समतुल्य हेक्सा-डेसिमल अंक में नीचे दिखाए अनुसार DB3 लिखा जा सकता है।

1101	1011	0011
D	B	3



### 1.8.3 हेक्सा-डेसिमल संख्या का बतलाना:

नीचे दिए गए दो रिवाजों में से किसी भी एक रिवाज के उपयोग करके, हेक्सा-डेसिमल मूल्य का प्रतिनिधित्व कर सकते हैं।

क) 2300h or 2300H - प्रत्यय h (या H) संकेत करते हैं कि 2300 एक हेक्स मूल्य है।

ख) 0x2300 - प्रत्यय के बदले, उपसर्ग 0x का भी उपयोग किया जाता है।

### 1.8.4 बाइनरी में धनात्मक तथा ऋणात्मक संख्याओं के प्रतिनिधित्व:

धनात्मक तथा ऋणात्मक डेसिमल मूल्यों का, बाइनरी रूप में प्रतिनिधित्व करने के लिए निम्न तीन पद्धतियों का उपयोग करते हैं।

क) साईन-बिट पद्धति,    ख) 1 की कॉम्प्लिमेंट पद्धति,    ग) 2 की कॉम्प्लिमेंट पद्धति

#### 1.8.4.1 साईन-बिट पद्धति:

इस पद्धति में, संख्या का निशान (+ or -) दर्शाने के लिए, अधिकतम महत्वपूर्ण बिट (MSB) का उपयोग करते हैं। धनात्मक संख्याओं के प्रतिनिधित्व के लिए MSB 0 होता है तथा ऋणात्मक संख्याओं के प्रतिनिधित्व के लिए MSB 1 होता है। MSB को छोड़कर बाकी बिट्स, धनात्मक या ऋणात्मक संख्याओं के मूल्य की प्रतिनिधित्व करते हैं।

**उदाहरण:** जब 7 को निशान के साथ, 8 बिट बाइनरी रूप में दर्शाना हो तो:

+7    00000111

-7    10000111

एक 8-बिट बाइनरी संख्या,  $-(2^{n-1} - 1)$  से  $+(2^{n-1} - 1)$  तक की सीमा में सभी डेसिमल संख्याओं को प्रतिनिधित्व कर सकते हैं। इसलिए 8-बिट चिह्नित प्रतिनिधित्व में, -127 से +127 तक की संख्याओं को दर्शाया जा सकता है।

#### 1.8.4.21 की कॉम्प्लिमेंट प्रणाली:

इस पद्धति में, धनात्मक संख्याओं के प्रतिनिधित्व, साईन-बिट पद्धति जैसे ही करते हैं लेकिन ऋणात्मक संख्याओं को उसकी धनात्मक मूल्य का, 1 की कॉम्प्लिमेंट में दर्शाते हैं। इस पद्धति में भी धनात्मक तथा ऋणात्मक संख्याओं में अंतर दिखाने के लिए MSB का ही उपयोग करते हैं।

**उदाहरण:** जब 7 को ही निशान के साथ, 8 बिट बाइनरी रूप में दर्शाना हो तो:

+7    00000111

-7    11111000

प्रतिनिधित्व की इस पद्धति में, एक n-बिट बाइनरी द्वारा प्रतिनिधित्व करने वाले संख्याओं की सीमा  $-(2^{n-1} - 1)$  से  $+(2^{n-1} - 1)$  है, जो साईन-बिट पद्धति के समान है। 8-बिट चिह्नित प्रतिनिधित्व में, -127 से +127 तक की डेसिमल संख्याओं को दर्शाया जा सकता है।

#### 1.8.4.32 की कॉम्प्लिमेंट प्रणाली:

इस पद्धति, ऋणात्मक संख्याओं की, 2 का पूरक प्रतिनिधित्व के अलावा, 1 की कॉम्प्लिमेंट पद्धति जैसे ही है। यहाँ भी MSB को ही साईन-बिट के लिए उपयोग करते हैं।

**उदाहरण:** यहाँ भी 7 को धनात्मक तथा ऋणात्मक बाइनरी रूप में इस तरह दर्शाते हैं:

+7      00000111

-7      11111001

अब हम, -7 के मूल्य का अभ्यास करते हैं, जो कि ऊपर 11111001 दिया गया है। 7 की बाइनरी मूल्य 0000111 (7 बिट्स) है। उसका 1 की कॉम्प्लिमेन्ट 1111000 है। इसके साथ 1 को जोड़ने से, 2 की कॉम्प्लिमेन्ट मिलते हैं, जो 1111001 है। इसी को, ऋणात्मक मूल्य के रूप में दर्शाने के लिए, MSB के पहले एक 8वा बिट 1 को जोड़ते हैं। अब, 2 की कॉम्प्लिमेन्ट पद्धति में 7 की अंतिम रूप 11111001 हो जाती है।

इस पद्धति में, n-बिट बाइनरी द्वारा प्रतिनिधित्व करने वाले संख्याओं की सीमा -  $(2^{n-1} - 1)$  से  $+(2^{n-1} - 1)$  है। 8-बिट प्रतिनिधित्व में, -128 से +127 तक के डेसिमल संख्याओं को दर्शाया जा सकते हैं। धनात्मक और ऋणात्मक मूल्यों की सीमा में यह असमानता, नीचे दिए गए तालिका 1.3 से समझ सकते हैं।

साइन-बिट	संख्या की मूल्य बिट							प्रतिनिधित्व डेसिमल मूल्य
0	1	1	1	1	1	1	1	= 127
0	1	1	1	1	1	1	0	= 126
0	0	0	0	0	0	1	0	= 2
0	0	0	0	0	0	0	1	= 1
0	0	0	0	0	0	0	0	= 0
1	1	1	1	1	1	1	1	= -1
1	1	1	1	1	1	1	0	= -2
1	0	0	0	0	0	0	1	= -127
1	0	0	0	0	0	0	0	= -128

**तालिका 1.3. 8-बिट रूप में, 2 की कॉम्प्लिमेन्ट पद्धति में, पूर्णांक की प्रतिनिधित्व**

ऊपर के सभी तीनों पद्धतियों में धनात्मक मूल्यों का प्रतिनिधित्व एक समान है, पर मुख्य अंतर ऋणात्मक मूल्यों को व्यक्त करने में है। ऋणात्मक मूल्यों को व्यक्त करने के लिए, कंप्यूटरों तथा माइक्रोप्रोसेसरों में, 2 की कॉम्प्लिमेन्ट पद्धति का ही उपयोग करते हैं।

### कुछ महत्वपूर्ण बाइनरी शब्द:

**बिट** - एक बाइनरी मूल्य के प्रत्येक बाइनरी डिजिट को बिट कहते हैं।

**बाइट** - 8 बिटों की एक समूह को बाइट कहते हैं।

**शब्द** - बिटों का एक समूह, जो CPU द्वारा एक ही समय में समानांतर रूप में एक्सेस किया जा सकता है।

### वस्तुनिष्ठ:

1. एक डिजिटल परिमाण वह है, जो कुछ \_\_\_\_\_ मूल्यों में व्यक्त करते हैं।
2. एक एनलॉग परिमाण वह है, जो \_\_\_\_\_ मूल्यों में व्यक्त करते हैं।
3. टोगल स्विच \_\_\_\_\_ के समतुल्य होते हैं।
4. सीढ़ीनुमा और रैंप, क्रमशः \_\_\_\_\_ और \_\_\_\_\_ परिमाण का उदाहरण हैं।
5. डेसिमल मूल्य को बाइनरी में परिवर्तित करने के लिए \_\_\_\_\_ पद्धति का उपयोग की जाती है।
6. 1 की कॉम्प्लिमेंट पद्धति से व्यक्त कर सकने वाला मूल्यों की सीमा \_\_\_\_\_ हैं।
7. 2 की कॉम्प्लिमेंट पद्धति से व्यक्त कर सकने वाला मूल्यों की सीमा \_\_\_\_\_ हैं।

### विषयनिष्ठ:

1. एनलॉग तथा डिजिटल परिमाणों को परिभाषित करें। दोनों के कुछ उदाहरण दें।
2. डिजिटल सर्किटों में बाइनरी संख्याओं का उपयोग क्यों करते हैं?
3. निम्न डेसिमल मूल्यों को बाइनरी में परिवर्तित करें।  
25, 37, 48, 59
4. निम्न बाइनरी मूल्यों को डेसिमल में परिवर्तित करें।  
11011, 101010, 111000
5. डबल डाबल पद्धति क्या है? इसका उपयोग कहाँ होता है?

## अध्याय 2

### लॉजिक गेट

#### 2.0 परिचय:

डिजिटल इलेक्ट्रॉनिक्स में, लॉजिक गेट, एक प्राथमिक सर्किट है। लॉजिक शब्द का मतलब है, तर्क या तर्क-वितर्क की एक विधि, जो अपने निष्कर्ष को सही दिखाता है। इसका मतलब यह है कि, प्रत्येक लॉजिक गेट को अपने आउटपुट मूल्य के लिए अपने कार्य प्रणाली होता है। इलेक्ट्रॉनिक्स क्षेत्र में, कुल मिलाकर सात लॉजिक गेट हैं, जो डिजिटल इलेक्ट्रॉनिक्स के रूप में एक अलग शाखा का विकास के लिए कारण बना है। ये गेटों की सूची निम्न प्रकार हैं।

1. OR गेट
2. AND गेट
3. NOT गेट
4. NOR गेट
5. NAND गेट
6. एक्सक्लूसीव- OR गेट
7. एक्सक्लूसीव- NOR गेट

#### 2.1 मूल गेट:

ये सभी लॉजिक गेट का विकास, तीन मौलिक या मूल लॉजिक ऑपरेशनों से की गई हैं।

ये तीन मूल लॉजिक हैं:

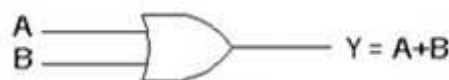
1. OR लॉजिक
2. AND लॉजिक
3. NOT या विपरीत लॉजिक

जो इलेक्ट्रॉनिक्स सर्किट, ये तीनों लॉजिक ऑपरेशनों को कार्यान्वित करते हैं, उसको मूल लॉजिक गेट कहा जाता है, और ये निम्नलिखित हैं।

- 1) OR गेट
- 2) AND गेट
- 3) NOT गेट

##### 2.1.1 OR गेट:

इस गेट (सर्किट) में कम से कम दो इनपुट और एक आउटपुट होते हैं। OR गेट का प्रतिनिधित्व, नीचे दिए गए एक प्रतीक से करना एक आम प्रथा है। (सभी लॉजिक गेटों का प्रतिनिधित्व स्थिर ग्राफिकल प्रतीकों से करते हैं, जिसे ANSI प्रतीक कहा जाता है)।



चित्र 2.1. OR गेट का प्रतीक

## लॉजिक गेट

A और B इनपुट तथा Y आउटपुट हैं। इनपुट और आउटपुट सिगनल दोनों सिर्फ बाइनरी 0(निम्न) तथा 1(उच्च) ही होती है। OR लॉजिक के नियमों के अधीन, दोनों इनपुट A और B मूल्यों के आधार पर आउटपुट 0 या 1 हो सकते हैं।

इस लॉजिक के अनुसार, कम से कम, A या B में, किसी एक इनपुट, 1 होता है तभी आउटपुट Y, 1 होगा। मतलब यह है कि, आउटपुट 1 मिलने के लिए, किसी भी एक इनपुट 1 होना चाहिए। गणितीय रूप में इस लॉजिक को  $Y = A + B$  दर्शाते हैं। यह समीकरण को, OR गेट के लिए बूलियन समीकरण कहा जाता है। '+' चिह्न को, 'OR' से उच्चारित की जाती है।

OR गेट के लिए लॉजिक समीकरण:  $Y = A + B$

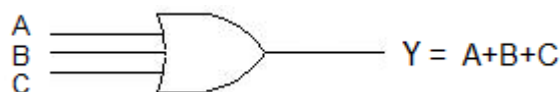
### ट्रूथ टेबल (Truth Table):

तालिका 2.1 का उपयोग, इनपुट और आउटपुट के संबंध को व्यवस्थित रूप से दर्शाने के लिए की जाती है, जिसे ट्रूथ टेबल कहा जाता है। सभी संभाव्य इनपुट अवस्थाओं को उनके संबंधित आउटपुट के साथ-साथ आरोही बाइनरी क्रम में लिखा जाता है।

इनपुट		आउटपुट
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

तालिका 2.1. OR गेट का ट्रूथ टेबल

यदि, एक OR गेट में तीन इनपुट हैं, तब उसकी बूलियन समीकरण,  $Y = A + B + C$  लिखी जाती है और इसका प्रतीक नीचे दर्शाया गया है।

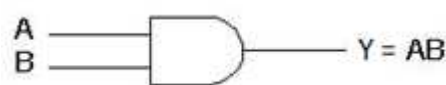


चित्र 2.2. 3 इनपुट के साथ OR गेट

इसी तरह, किसी भी संख्या में इनपुट के लिए OR गेट का उपयोग किया जा सकता है।

### 2.1.2 AND गेट:

इसमें भी कम से कम दो इनपुट होते हैं। इसकी ग्राफिक प्रतीक और बूलियन समीकरण निम्न प्रकार हैं।



चित्र 2.3 AND गेट का प्रतीक

लॉजिक समीकरण  $Y = A.B$  या संक्षिप्त में,  $Y = AB$

ट्रूथ टेबल:

इनपुट		आउटपुट
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

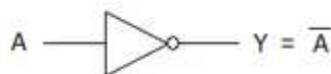
तालिका 2.2. AND गेट का ट्रूथ टेबल

इस गेट का लॉजिक कार्य इस तरह है कि, इसकी A और B दोनों इनपुट 1 (उच्च) होने पर ही आउटपुट 1 (उच्च) होता है। मतलब, चाहे इनपुट की संख्या कितनी भी हो, आउटपुट में 1 मिलने के लिए इसके सभी इनपुट 1 होने चाहिए।

इसका लॉजिक समीकरण  $Y = A.B$  को, A and B उच्चारित किया जाता है।

### 2.1.3 NOT गेट:

अन्य दो गेटों के विपरीत इस में एक ही इनपुट होता है। इसका लॉजिक भी सरल है। यह इनपुट सिगनल को उलट देता है और विपरीत आउटपुट देता है, यानि, 0 को 1 में और 1 को 0 में उलटता है। इसका प्रतीक और बूलियन समीकरण निम्न प्रकार है।



चित्र 2.4. NOT गेट का प्रतीक

NOT गेट की लॉजिक समीकरण:  $Y = \overline{A}$  (A को, A का उलटा कहा जाता है)

ट्रूथ टेबल:

A	Y
0	1
1	0

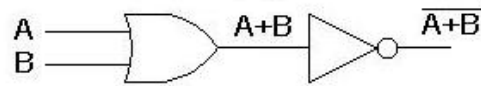
तालिका 2.3. NOT गेट का ट्रूथ टेबल

नीचे दर्शाए गए अनुसार, यह तीन मूल गेटों (लॉजिक) का उपयोग करके बहुत संख्या में अन्य गेट उत्पन्न कर सकते हैं। इसी कारण से इन तीनों को मूल लॉजिक गेट कहा जाता है।

## 2.2 संयुक्त गेट:

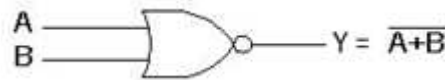
यह तीन मूल लॉजिक गेटों का संयोजन करके, चार और गेटों का विकास की गई है, जिसको संयुक्त गेट कहा जाता है, जिनमें से प्रत्येक का अपना महत्व है।

**2.2.1 NOR गेट:** नीचे दर्शाए गए अनुसार, NOR गेट, OR और NOT गेट का एक संयोजन है।



चित्र 2.5 NOR लॉजिक

इस संयोजन को NOR गेट कहा जाता है और इसका प्रतीक नीचे दिए गए हैं।



चित्र 2.6 NOR गेट का प्रतीक

NOR गेट का लॉजिक समीकरण:  $Y = \overline{A + B}$

मतलब, समान इनपुट मूल्यों के लिए, इसकी आउटपुट, OR गेट आउटपुट के विपरीत होता है।  $A+B$  के ऊपर के लाईन को 'बार' से उच्चारित करते हैं।

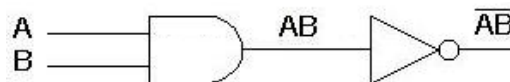
**NOR गेट के ट्रुथ टेबल:**

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

तालिका 2.4. NOR गेट का ट्रुथ टेबल

**2.2.2 NAND गेट:**

कार्यात्मक रूप से यह AND और NOT गेट का संयोजन है।



चित्र 2.7. NAND लॉजिक

इसका ANSI प्रतीक है:



चित्र 2.8. NAND गेट का प्रतीक

लॉजिक समीकरण:  $Y = \overline{AB}$



ट्रूथ टेबल:

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

तालिका 2.5. NAND गेट का ट्रूथ टेबल

ध्यान देना है कि, समान इनपुट मूल्यों के लिए, इसका आउटपुट, AND गेट आउटपुट के विपरीत होता है।

**2.2.3 एक्सक्लूसीव- OR गेट:** इसको संक्षिप्त में Ex-OR गेट भी कहा जाता है।

**Ex-OR गेट का लॉजिक:** इसका लॉजिक, गेट के उच्च (1) इनपुटों की संख्या में सादृश्य की जाँच करने की है। सादृश्य का मतलब है कि, गेट के इनपुट में 1 की संख्याओं को गिनने के बाद, जो मूल्य मिलता है, यह वहीं है। यदि यह सादृश्य विषम संख्या है तो, गेट का आउटपुट उच्च (1) होता है अन्यथा निम्न (0) होता है।



चित्र 2.9. EX-OR गेट का प्रतीक

**लॉजिक समीकरण:**

इसका लॉजिक या बूलियन समीकरण इस तरह लिखते हैं:  $Y = A \oplus B$

$\oplus$  प्रतीक Ex-OR ऑपरेशन को दर्शाता है। लेकिन गणितीय रूप से इस प्रतीक का कोई मतलब नहीं होता है। इसलिए को गणितीय मान्य समीकरण में इस तरह दर्शाते हैं।

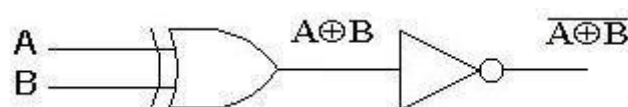
$$Y = A \oplus B = \bar{A}B + A\bar{B}$$

ट्रूथ टेबल:

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

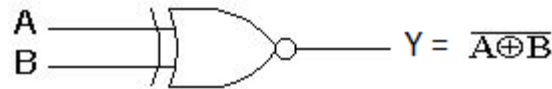
तालिका 2.6. EX-OR गेट का ट्रूथ टेबल

**2.2.4 एक्सक्लूसीव-NOR गेट:** इसे Ex-NOR गेट भी कहा जाता है। यह Ex-OR और NOT गेटों का संयोजन है।



चित्र 2.10. Ex-NOR का लॉजिक

इसका प्रतीक, लॉजिक समीकरण और ट्रूथ टेबल निम्न प्रकार है।



चित्र 2.11. EX - NOR गेट का प्रतीक

लॉजिक समीकरण:

$$Y = \overline{\overline{A}B + A\overline{B}} \quad \text{या} \quad Y = \overline{A}B + A\overline{B}$$

ट्रूथ टेबल:

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

तालिका 2.7. Ex-NOR गेट का ट्रूथ टेबल

इसका लॉजिक Ex-OR गेट से बिल्कुल विपरीत है।

## 2.3 यूनिवर्सल गेट:

सभी लॉजिक गेटों में, NOR गेट तथा NAND गेट, यूनिवर्सल गेट के नाम से जाना जाता है। इसका कारण यह है कि सिर्फ NOR या NAND गेट का उपयोग करके दूसरा सभी लॉजिक गेट या लॉजिक कार्य लागू किए जा सकते हैं। इसलिए इन्हें यूनिवर्सल गेट कहा जाता है।

DeMorgan का सिद्धांत के नाम से जानने वाले निम्न दो सिद्धांतों से यूनिवर्सल गेटों का कार्यप्रणाली अच्छी तरह समझ सकते हैं।

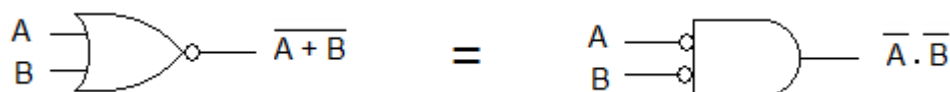
DeMorgan's सिद्धांत:

$$\begin{aligned} 1) \quad \overline{A + B} &= \overline{A} \cdot \overline{B} \\ 2) \quad \overline{A \cdot B} &= \overline{A} + \overline{B} \end{aligned}$$

### 1. DeMorgan का पहला सिद्धांत: (NOR के लिए)

$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

समीकरण में बाएं ओर का पद, NOR गेट के लिए बूलियन समीकरण है। यह सिद्धांत के अनुसार समान इनपुट वाले निम्न दो गेटों के लिए लॉजिक आउटपुट भी समान होता है। यह सिद्धांत के अनुसार, NOR का समीकरण नीचे चित्र 2.12 में दर्शाया गया है।

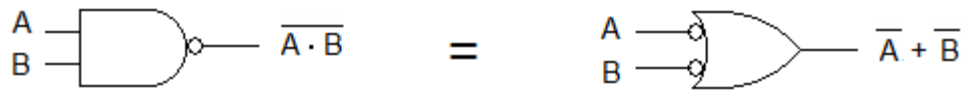


चित्र 2.12. DeMorgan का पहला सिद्धांत

## 2. DeMorgan का दूसरा सिद्धांत: (NAND के लिए)

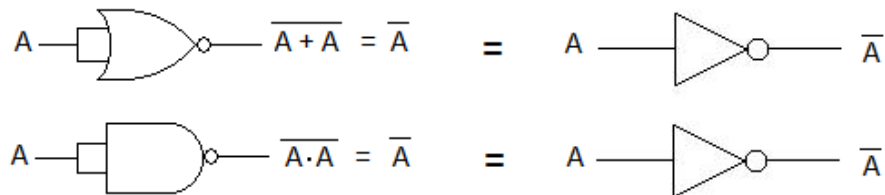
$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

समीकरण में बाएं ओर का पद, NAND गेट के लिए बूलियन समीकरण है। यह सिद्धांत के अनुसार, समान इनपुट वाले निम्न दो गेटों के लिए लॉजिक आउटपुट भी समान होता है। यह सिद्धांत के अनुसार NAND का लॉजिक नीचे चित्र 2.13 में दर्शाया गया है।

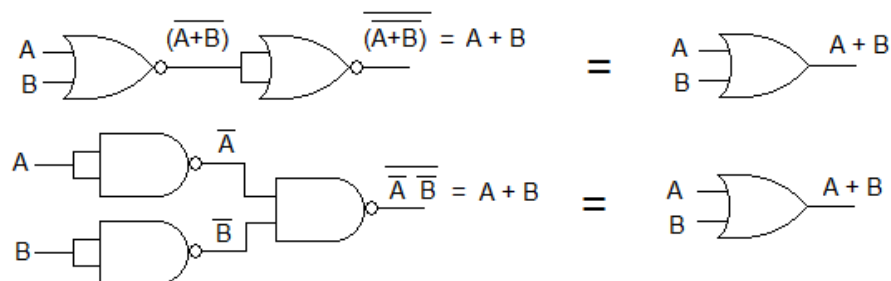


चित्र 2.13. DeMorgan का दूसरा सिद्धांत

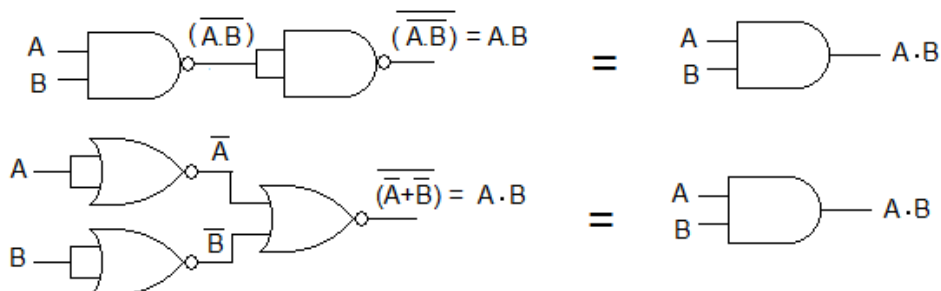
NOR तथा NAND गेट का उपयोग करके अन्य लॉजिक कार्य प्राप्त करने का कुछ उदाहरण नीचे दिए गए हैं।



चित्र 2.14. NOT लॉजिक के लिए यूनिवर्सल गेट का उपयोग



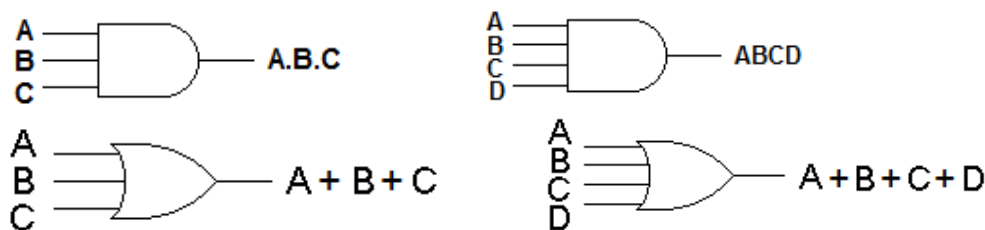
चित्र 2.15. OR लॉजिक के लिए यूनिवर्सल गेट का उपयोग



चित्र 2.16. AND लॉजिक के लिए यूनिवर्सल गेट का उपयोग

## 2.4 मल्टिपल इनपुट गेट:

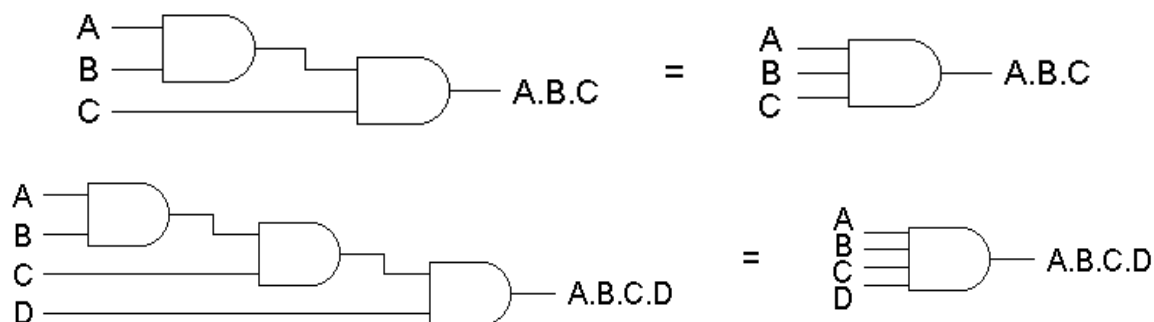
किसी भी संख्या में इनपुट के साथ, लॉजिक गेटों का विकास किया जा सकता है। कुछ उदाहरण नीचे दर्शाए गए हैं।



चित्र 2.17. मल्टीपल इनपुट गेट

## 2.5 मल्टी इनपुट गेट बनाना:

आवश्यकता से कम इनपुट वाले लेकिन समान प्रकार के गेट को जोड़कर मल्टी इनपुट गेट बनाए जा सकते हैं। दो इनपुट AND गेट का उपयोग करके तीन इनपुट AND गेट तथा चार इनपुट AND गेट को बनाने के तरीके नीचे चित्रों में दर्शाए गए हैं।



चित्र 2.18. दो इनपुट गेट से मल्टी-इनपुट गेट का निर्माण

## 2.6 बूलियन बीजगणित के नियम:

लॉजिक पदों को सुलझाने एवं सरल बनाने और लॉजिक सर्किटों के विश्लेषण के लिए DeMorgan का सिद्धांत के अलावा निम्नलिखित बुनियादी नियमों का भी उपयोग करते हैं। इन नियमों को बूलियन बीजगणित का नियम कहा जाता है।

1. कम्प्यूटेटिव नियम
2. एसोसिएटिव नियम
3. डिस्ट्रिब्यूटिव नियम
4. आइडेंटिटी नियम
5. इयुआलिटी नियम
6. प्रेसीडेन्स नियम
7. रिडंडन्सी नियम
8. DeMorgan's सिद्धांत

ध्यान दें कि, प्रत्येक नियमों में दो भाव (क) और (ख) हैं। यह प्रत्येक AND(.) को OR(+) में, OR(+) को AND(.) में और सभी 1 को 0 में और उल्टे बदलते हुए प्राप्त करते हैं। AND का प्रतीक (.) को निकाल देना पारंपरिक बन गया है, अर्थात्, A.B को AB लिखा जाता है।

## 2.7 बूलियन नियम:

T1: कम्यूटेटीव नियम

(क)  $A + B = B + A$

(ख)  $A B = B A$

T2: एसोसियेटीव नियम

(क)  $(A + B) + C = A + (B + C)$

(ख)  $(A B) C = A (B C)$

T3: डिस्ट्रिब्यूटीव नियम

(क)  $A (B + C) = A B + A C$

(ख)  $A + (B C) = (A + B) (A + C)$

T4: आइडेंटिटी नियम

(क)  $A + A = A$

(ख)  $A \cdot A = A$

T5: ड्युआलिटी नियम

(क)  $AB + \overline{AB} = A$

(ख)  $(A + B) (A + B) = A$

T6: प्रेसिडेन्स नियम

(क)  $A \cdot B + C = (A \cdot B) + C$

(ख)  $A + B \cdot C = A + (B \cdot C)$

T7: रिडंडन्स नियम

1) (क)  $A + A B = A$

(ख)  $A (A + B) = A$

2) (क)  $0 + A = A$

(ख)  $0 \cdot A = 0$

3) (क)  $1 + A = 1$

(ख)  $1 \cdot A = A$

4) (क)  $A + \overline{A} = 1$

(ख)  $A \cdot \overline{A} = 0$

5) (क)  $A \overline{A} B = A + B$

(ख)  $A (A + B) = A B$

De Morgan का सिद्धांत

(क)  $\overline{A + B} = \overline{A} \cdot \overline{B}$

(ख)  $\overline{AB} = \overline{A} + \overline{B}$

## 2.8 बूलियन पास्चुलेट्स:

ऊपर दर्शाए गए संबंधों से नीचे बताए अनुसार, बूलियन पास्चुलेट्स नामक कुछ और नियमों को प्राप्त कर सकते हैं।

P1: X मतलब 1 या 0

P2:  $0.0 = 0$

P3:  $1 + 1 = 1$

P4:  $0 + 0 = 0$

P5:  $1.1 = 1$

P6:  $1.0 = 0.1 = 0$

P7:  $1+0 = 0+1 = 1$

### वस्तुनिष्ठ:

1. OR गेट \_\_\_\_\_ गेटों में से एक है।
2. NOR गेट को \_\_\_\_\_ गेट कहा जाता है।
3. EX-OR गेट का लॉजिक \_\_\_\_\_ सादृश्य है।
4. \_\_\_\_\_ गेट इनपुट को उलट देता है।
5. NAND गेट \_\_\_\_\_ तथा \_\_\_\_\_ गेट के समान है।

### विषयनिष्ठ:

1. एक लॉजिक गेट क्या है? विविध लॉजिक गेटों के नाम बताएं।
2. मूल गेट पद से आप क्या समझते हैं? इसके बारे में वर्णन करें।
3. किसी भी दो संयुक्त गेटों का वर्णन करें।
4. यूनिवर्सल गेट क्या है? इसका वर्णन करें।
5. पूरे De Morgan का सिद्धांत लिखिए और कुछ उदाहरण के साथ उसकी उपयोगिता का वर्णन करें।

## अध्याय 3

### संयोजन लॉजिक सर्किट्स

#### 3.0 संयोजन लॉजिक उपकरण:

लॉजिक गेटों का उपयोग करके विकसित किया गया परिपथ को संयोजन लॉजिक सर्किट कहा जाता है। सबसे महत्वपूर्ण संयोजन लॉजिक सर्किट इस प्रकार है।

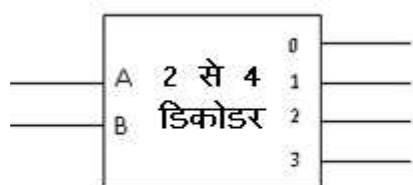
1. डिकोडर्स
2. ऐडर्स तथा सबट्रैक्टर्स
3. मल्टिप्लेक्सर्स तथा डीमल्टिप्लेक्सर्स

#### 3.1 डिकोडर:

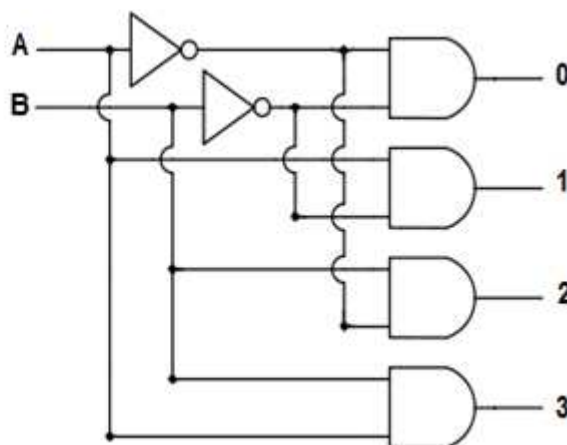
डिकोडर एक डिजिटल उपकरण है, जो इनपुट टर्मिनल में दिए गए बाइनरी मूल्य के आधार पर एक से ज्यादा आउटपुट से एक समय में केवल एक ही आउटपुट का चयन करते हैं।

मान लें कि, उपलब्ध इनपुट की संख्या 'n' है, तब आउटपुट की संख्या '2<sup>n</sup>' होती है। प्रत्येक आउटपुट, क्रमानुसार 0 से शुरू करके 2<sup>n-1</sup> में समाप्त हो, इस तरह नामांकित किया जाता है। यानि, 0, 1, 2, 3, --- 2<sup>n-1</sup>.

नीचे चित्र में एक 2 इनपुट तथा 4 आउटपुट वाला डिकोडर दिखाया गया है।



चित्र 3.1 2-4 लाइन डिकोडर का प्रतीक



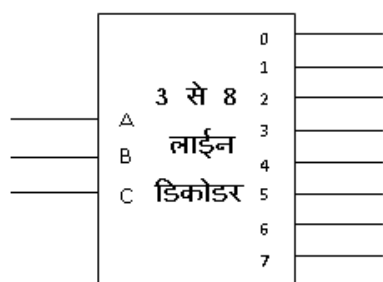
चित्र 3.1 2-4 लाइन डिकोडर का लॉजिक सर्किट

इनपुट		आउटपुट			
A	B	0	1	2	3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

तालिका 3.1. 2-4 डिकोडर का ट्रूथ टेबल



इसी तरह एक 3 इनपुट डिकोडर का चित्र नीचे दर्शाया गया है।



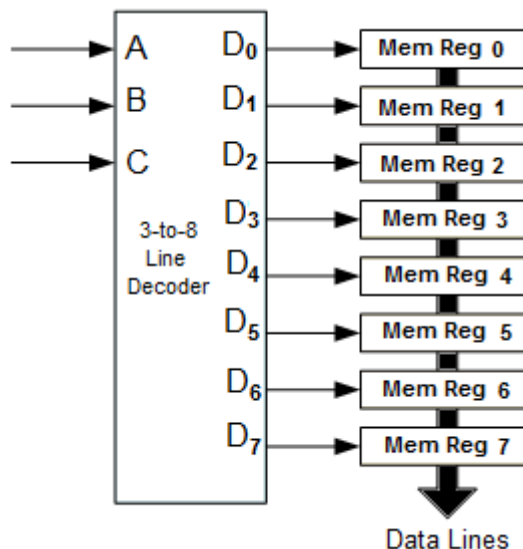
चित्र 3.3.3 - 8 डिकोडर

इनपुट			आउटपुट							
A	B	C	0	1	2	3	4	5	6	7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

तालिका 3.2. 3 - 8 डिकोडर की दृश्य टेबल

### 3.2 डिकोडर के प्रयोग:

व्यक्तिगत कार्यात्मक यूनिटों या उपकरणों के चयन के लिए, सभी माइक्रोप्रोसेसरों, माइक्रोप्रोसेसर आधारित प्रणालियों और सभी मेमोरी चिप में डिकोडर अनिवार्य है। एक मेमोरी चिप के अंदर व्यक्तिगत मेमोरी रजिस्टर की चयन की पद्धति नीचे चित्र 3.4 में दर्शाया गया है।



चित्र 3.4. एक मेमोरी चिप की मेमोरी रजिस्टर डिकोडर

इसी तरह, माइक्रोप्रोसेसर प्रणालियों में, चिप चयन सिगनल द्वारा व्यक्तिगत परिधीय चिप का चयन के लिए भी डिकोडर का उपयोग करते हैं।

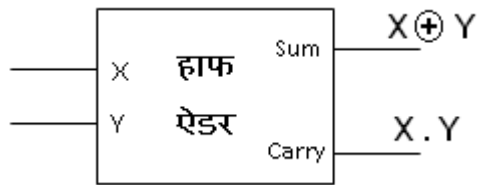
### 3.3 ऐडर:

यह बाइनरी बिटों (संख्याओं) को जोड़ने के लिए एक अंकगणितीय सर्किट है। दो तरह का ऐडर उपलब्ध है।

- 1) आधा ऐडर
- 2) पूर्ण ऐडर

#### 3.3.1 आधा ऐडर:

यह एक समय में किसी भी दो बाइनरी बिटों को जोड़ते हैं। X और Y दो विभिन्न बाइनरी संख्याओं के दो बिट हैं, जिसको जोड़ना है। इसका चित्र नीचे दर्शाया गया है।



चित्र 3.5. आधा ऐडर

X	Y	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

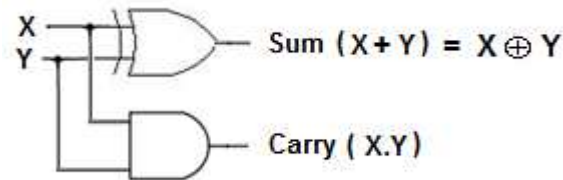
तालिका 3.3. आधा ऐडर का दृश्य टेबल

उदाहरण:

X	0	0	1	1
+Y	+0	+1	+0	+1
S	0	1	1	0
C	0	0	0	1

S = जोड़ आउटपुट

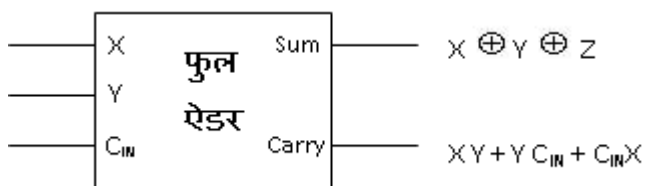
C = कैरी (carry) आउटपुट



चित्र 3.6. आधा ऐडर का लॉजिक चित्र

### 3.3.2 पूर्ण ऐडर:

आधा ऐडर के विपरीत एक पूर्ण ऐडर एक समय में 3 बिटों को जोड़ सकता है। X और Y दो विभिन्न बाइनरी संख्याओं के दो बिट हैं, जिसको जोड़ना है तथा  $C_{IN}$  कैरी (carry) इनपुट है, जो एक मल्टी-बिट संख्या के तत्काल निम्न स्थिति के दो बिटों के जोड़ से उत्पादित है।



चित्र 3.7. पूर्ण ऐडर

X	Y	$C_{IN}$	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

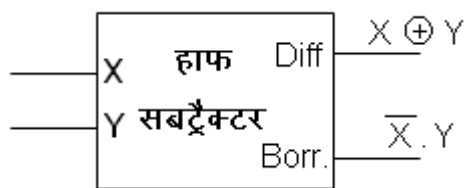
तालिका 3.4. पूर्ण ऐडर का दृश्य टेबल

### 3.4 सबट्रैक्टर:

ऐडर की तरह, सबट्रैक्टर भी दो प्रकार के हैं।

- 1) आधा सबट्रैक्टर
- 2) पूर्ण सबट्रैक्टर

### 3.4.1 आधा सबट्रैक्टर: यह दो बिटों का शेष निकालता है।



चित्र 3.8. आधा सबट्रैक्टर

X	Y	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

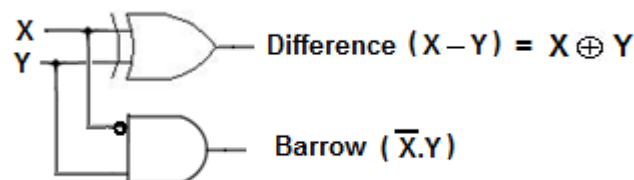
तालिका 3.5. आधा सबट्रैक्टर का दृश्य टेबल

उदाहरण:

X	0	0	1	1
-Y	-0	-1	-0	-1
D	0	1	1	0
B	0	1	0	0

D - शेष आउटपुट

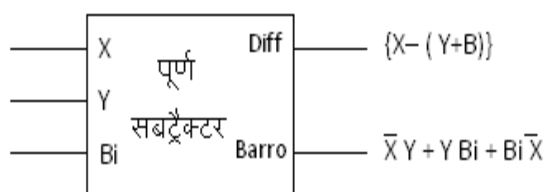
B - उधार आउटपुट



चित्र 3.9. आधा सबट्रैक्टर का लॉजिक आरेख

### 3.4.2 पूर्ण सबट्रैक्टर:

पूर्ण सबट्रैक्टर एक तीसरी इनपुट बिट का भी समायोजन करने के लिए डिजाइन किया गया है, जो पिछले बिट स्थिति में घटाव के लिए आवश्यक एक उधार बिट है। ऐसे उधार बिट की गणना के लिए, पूर्ण सबट्रैक्टर में एक तीसरे इनपुट (Bi) प्रदान की गई हैं। यह उधार बिट भी नेगटिव होने से Y तथा Bi दोनों को आपस में जोड़ते हैं और शेष मूल्य के लिए इस जोड़ को X से घटाते हैं। यह चित्र 3.10 में दिए गए समीकरण से स्पष्ट होता है।



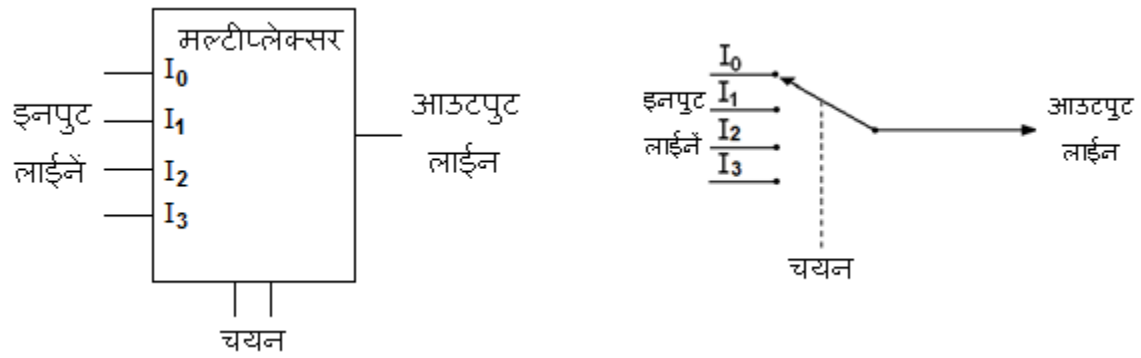
चित्र 3.10. पूर्ण सबट्रैक्टर

X	Y	Bi	D	Bo
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

तालिका 3.6. पूर्ण सबट्रैक्टर का दृश्य टेबल

### 3.5 मल्टीप्लेक्सर:

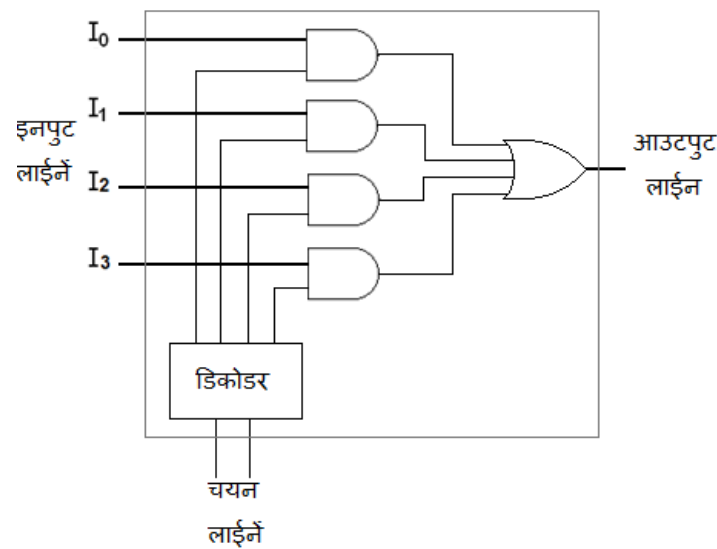
डिजिटल मल्टीप्लेक्सर एक साधन है, जो कई इनपुटों और एकल आउटपुट के बीच कनेक्टिविटी की सुविधा प्रदान करती है। मतलब इनपुट चयन लॉजिक द्वारा, एक समय में, किसी एक इनपुट लाईन को आउटपुट लाईन से जोड़ा जा सकता है। नीचे दिए गए चित्र 3.11 मल्टीप्लेक्सर लॉजिक का सरल वर्णन करता है।



चित्र 3.11. मल्टीप्लेक्सर का प्रतीक और उसका लॉजिक

### 3.6 4-बिट मल्टीप्लेक्सर:

चित्र 3.12 में गेट और डिकोडर से बना हुआ एक 4-बिट मल्टीप्लेक्सर लॉजिक सर्किट दर्शाया गया है। इनपुट लाइनों के चयन के लिए डिकोडर का उपयोग चित्र में स्पष्ट देखा जा सकता है। डिकोडर का उपयोग AND गेट को सक्षम करने के लिए की जाती है, जो आवश्यक इनपुट ( $I_0$ - $I_3$  में से) को आउटपुट में भेज देता है। यह चयन लाइनों पर एक बाइनरी मूल्य को लागू करके किया जाता है, जो जिस इनपुट लाइन को भेजना है, उसके क्रम संख्या के बराबर होता है। उदाहरण के लिए,  $I_2$  इनपुट के सिगनल को आउटपुट में भेजने के लिए, चयन लाइन पर बाइनरी 10 को लागू करना होता है। यह तीसरे AND गेट के दूसरे इनपुट में एक उच्च स्तर भेजते हैं और गेट को  $I_2$  इनपुट में उपलब्ध सिगनल को आउटपुट में भेजने के लिए सक्षम करते हैं।



चित्र 3.12. 4-इनपुट मल्टीप्लेक्सर का लॉजिक आरेख

**वस्तुनिष्ठ:**

1. डिकोडर का कार्य, जब एक बाइनरी इनपुट को लागू की जाती है, तब \_\_\_\_\_ का चयन करना है।
2. डिकोडर का एक प्रयोग \_\_\_\_\_ है।
3. एक पूर्ण ऐडर, एक समय में \_\_\_\_\_ बिटों को जोड़ता है।
4. एक मल्टीप्लेक्सर में \_\_\_\_\_ इनपुट और \_\_\_\_\_ आउटपुट होते हैं।
5. मल्टीप्लेक्सर में चयन लॉजिक \_\_\_\_\_ से प्रदान की जाती है।

**विषयनिष्ठ:**

1. लॉजिक आरेख के साथ डिकोडर का वर्णन करें।
2. डिकोडर के कम से कम दो प्रयोग बताएं।
3. किसी भी एक ऐडर का आरेख बनाएं और वर्णन करें।
4. आधा ऐडर और पूर्ण ऐडर के बीच क्या अंतर है?
5. एक आरेख के साथ मल्टीप्लेक्सर के कार्यशैली का वर्णन करें।

## अध्याय 4

### फ्लिप फ्लॉप

#### 4.0 परिचय:

फ्लिप फ्लॉप एक मूलभूत डिजिटल उपकरण है, जो डिजिटल इलेक्ट्रॉनिक्स के क्षेत्र में एक क्रांतिकारी परिवर्तन लाया है, जिससे डिजिटल लॉजिक सर्किट में अनुक्रमिक लॉजिक सर्किट नाम की एक नई शाखा का विकास हुआ।

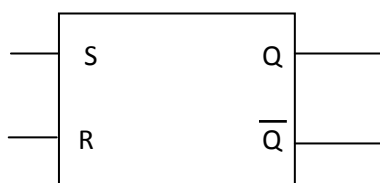
फ्लिप फ्लॉप, मूल रूप से एक मेमोरी या संग्रह तत्व है। फ्लिप फ्लॉप की विशेषताएं निम्न प्रकार हैं:

1. यह एक बाई-स्टेबल उपकरण है।
2. इसमें एक सामान्य आउटपुट 'Q' और उसका कॉम्प्लिमेन्टरी आउटपुट ' $\bar{Q}$ ' होते हैं।
3. यह एक एकल बिट डाटा का संग्रह कर सकता है।

फ्लिप फ्लॉप के बहुत प्रकार हैं। फ्लिप फ्लॉप के प्रकार की परवाह किए बिना सभी को ऊपर दिए गए विशेषताएं हैं। इसकी मुख्य प्रकार हैं:

#### 4.1 एसआर फ्लिप फ्लॉप:

यह फ्लिप फ्लॉप सरलतम रूप का है। इसका दो इनपुट 'एस' और 'आर' हैं। S 'सेट' के लिए और R 'रीसेट' के लिए माना जाता है। जब S इनपुट में एक उच्च सिगनल लागू किया जाता है, तब फ्लिप फ्लॉप का Q आउटपुट उच्च स्तर में (यानि  $Q=1$ ) बदल जाता है और इस स्थिति में फ्लिप फ्लॉप को "सेट स्थिति" कहा जाता है। इसी तरह जब R इनपुट में एक उच्च सिगनल लागू किया जाता है, तब  $Q=0$  जिसे "रीसेट स्थिति" कहा जाता है। एसआर फ्लिप फ्लॉप की प्रतीक नीचे दिया गया है।



चित्र 4.1. SR फ्लिप फ्लॉप

S	R	Q	$\bar{Q}$	स्थिति
0	0	$Q_n$	$\bar{Q}_n$	नो चेंज
0	1	0	1	रीसेट
0	0	0	1	पिछली
1	0	1	0	सेट
0	0	1	0	पिछली
1	1	?	?	अवैध

तालिका 4.1. SR फ्लिप फ्लॉप का ट्रुथ टेबल

$Q_n$  और  $\bar{Q}_n$  - पिछली अवस्थाएं हैं और '?' अवैध अवस्था है।

ऊपर के ट्रुथ टेबल S और R के विभिन्न इनपुट संयोजनों के लिए आउटपुट अवस्थाएं दर्शाती हैं। पहली अवस्था, जिसमें S और R दोनों 0 हैं, तब आउटपुट "नो चेंज" अवस्था में होता है। इसका मतलब आउटपुट पिछले अवस्था में ही रहती है।

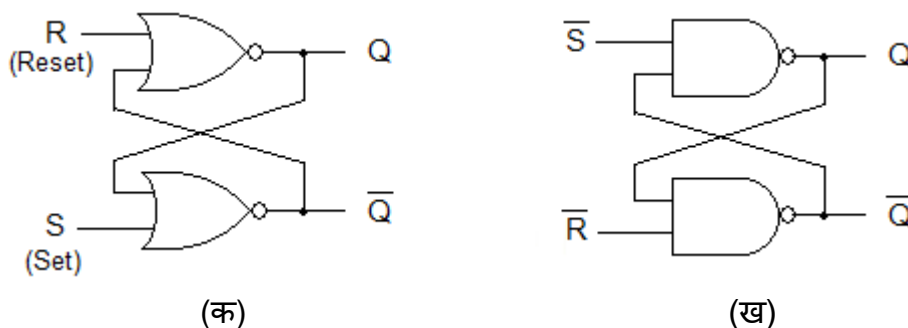
## फ्लिप फ्लॉप

S और R के दूसरे इनपुट संयोजन में, यानि  $S=0$  और  $R=1$  में, आउटपुट रीसेट अवस्था में चली जाती है। यानि,  $Q=0$  और  $\bar{Q}=1$  अवस्था को रीसेट अवस्था कही जाती है।

तीसरे संयोजन में यानि  $S=1$  और  $R=0$  में, आउटपुट सेट में बदल जाता है। तब  $Q=1$  और  $\bar{Q}=0$  होता है।

चौथा और अंतिम इनपुट संयोजन में  $S=1$  और  $R=1$  में फ्लिप फ्लॉप आउटपुट अवैध अवस्था में चली जाती है। इसका मतलब दोनों आउटपुट एक समय में समान हो जाते हैं यानि, 1 या 0. इसके अतिरिक्त, जब यह इनपुट ( $S=1$  और  $R=1$ ) को निकालकर दोनों इनपुट में 0 ( $S=0$  और  $R=0$ ) लागू किया जाए, अगली अवस्था जो मिलते हैं वह अप्रत्याशित होते हैं।

इसलिए यह चौथा संयोजन को “अवैध इनपुट” कहा जाता है और SR फ्लिप फ्लॉप में इसका उपयोग नहीं किया जाता है।



चित्र 4.2. SR फ्लिप फ्लॉप (क) NOR गेट से बना (ख) NAND गेट से बना

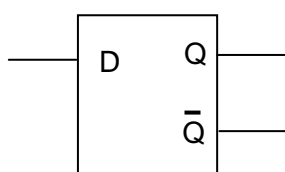
**SR फ्लिप फ्लॉप का प्रयोग:** SR फ्लिप फ्लॉप मुख्य रूप से सर्किट को सक्षम या निष्क्रिय करने के लिए उपयोग करते हैं। उदाहरण के लिए 555 टाइमर के नियंत्रण के लिए।

## 4.2 डी-फ्लिप फ्लॉप:

डी- फ्लिप फ्लॉप की मुख्य फायदें हैं:

- 1) SR फ्लिप फ्लॉप का अवैध इनपुट संयोजन, जो अप्रत्याशित अवस्था की ओर ले जाती है उसको हटाया गया।
- 2) SR फ्लिप फ्लॉप से भिन्न, आउटपुट में बदलाव लाने के लिए इसमें एक ही इनपुट सिगनल 'D' का ही आवश्यकता होती है। इसको डी (डाटा) फ्लिप फ्लॉप कहा जाता है क्योंकि D इनपुट का मूल्य Q में प्रतिबिंबित होते हैं।

इसका प्रतीक और ट्रूथ टेबल नीचे दिए गए हैं।

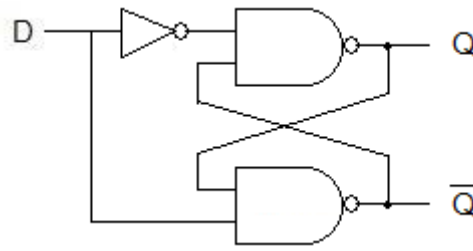


D	Q	$\bar{Q}$	अवस्था
0	0	1	रीसेट
1	1	0	सेट

चित्र 4.3. डी-फ्लिप फ्लॉप

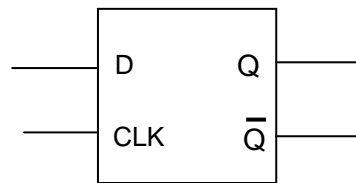
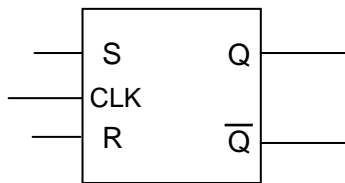
तालिका 4.2. डी-फ्लिप फ्लॉप का ट्रूथ टेबल





चित्र 4.4. डी-फ्लिप फ्लॉप का लॉजिक आरेख

अभी तक, जो SR और D फ्लिप फ्लॉप संस्करणों का वर्णन किया गया उसको क्रमशः SR लैच और D लैच कहा जाता है। यह दोनों का क्लॉकड संस्करण भी उपलब्ध है, जो 'क्लॉक' नामक एक अतिरिक्त इनपुट के साथ होता है। इसका लॉजिक सर्किट तथा ट्रूथ टेबल नीचे दिए गए हैं।



क) क्लॉक के साथ SR फ्लिप फ्लॉप

ख) क्लॉक के साथ D फ्लिप फ्लॉप

चित्र 4.5. क्लॉक के साथ SR और D फ्लिप फ्लॉप

CLK	S	R	Q	$\bar{Q}$	अवस्था
0	x	x	$Q_n$	$\bar{Q}_n$	नो चेंज
1	0	0	$Q_n$	$\bar{Q}_n$	नो चेंज
1	0	1	0	1	सेट
1	1	0	1	0	रीसेट
1	1	1	?	?	अवैध

CLK	D	Q	$\bar{Q}$	अवस्था
0	x	$Q_n$	$\bar{Q}_n$	नो चेंज
1	1	1	0	सेट
1	0	0	1	रीसेट

तालिका 4.3

क्लॉकड SR फ्लिप फ्लॉप की ट्रूथ टेबल

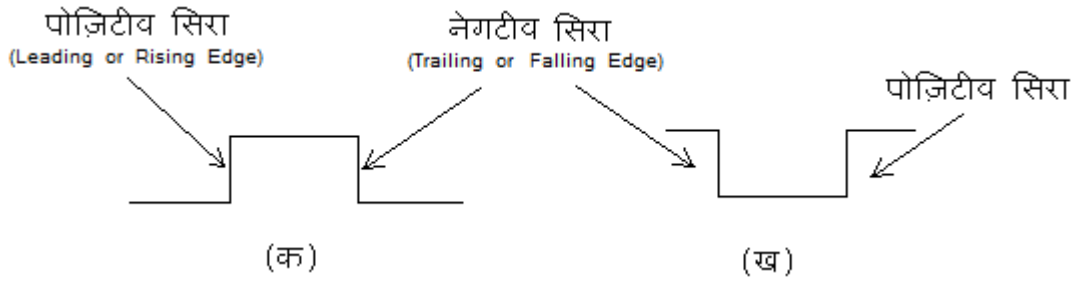
तालिका 4.4

क्लॉकड D फ्लिप फ्लॉप की ट्रूथ टेबल

**D फ्लिप फ्लॉप का प्रयोग:** इसका उपयोग मुख्य रूप से रजिस्ट्रों के निर्माण में किया जाता है।

### 4.3 टी-फ्लिप फ्लॉप:

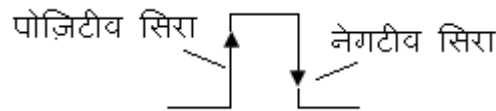
इस फ्लिप फ्लॉप में भी D फ्लिप फ्लॉप की तरह एक ही इनपुट 'T' होता है। इसमें फर्क इतना है कि, D फ्लिप फ्लॉप में D इनपुट में जो भी देते हैं वह आउटपुट में मिलते हैं, लेकिन टी फ्लिप फ्लॉप में बस एक 0 या 1 इनपुट से आउटपुट नहीं बदलती है। यह, प्रत्येक क्लॉक इनपुट पल्स का एक विशेष सिरा पर आउटपुट की अवस्था को बदल देती है। नीचे दिए गए चित्र क्लॉक पल्स की सिरा का वर्णन करता है।



चित्र 4.6. एक पल्स का पोज़िटिव तथा नेगटिव सिराएं

ऊपर दर्शाए गए चित्र दो अलग-अलग पल्सों को दर्शाते हैं। चित्र (क), पोज़िटिव की ओर जाने वाले पल्स को और (ख) उसकी विपरीत को, यानी, नेगटिव की ओर जाने वाले पल्स को दर्शाता है। पल्स के आकार कुछ भी हो, उसकी किनाराओं पर ध्यान देना आवश्यक है। हर पल्स को अनिवार्य रूप से दो सिराएं होते हैं, एक पोज़िटिव सिरा और एक नेगटिव सिरा।

वर्टिकल लाईन में पोज़िटिव सिरा तब प्राप्त होता है जब सिगनल का मूल्य 0 से 1 के तरफ बढ़ता है और 1 से 0 की तरफ बढ़ते हैं तब नेगटिव सिरा प्राप्त होता है। पोज़िटिव और नेगटिव सिराओं को क्रमशः ऊपर की तरफ और नीचे की तरफ का तीर से दर्शाते हैं।



चित्र 4.7. पोज़िटिव सिरा और नेगटिव सिरा दर्शाते तीर का चिह्न

टी फ्लिप फ्लॉप के आउटपुट इन दोनों में से किसी एक सिरा के दौरान अपनी अवस्था बदलती है, न कि दोनों सिराओं के दौरान। कुछ टी फ्लिप फ्लॉप, क्लॉक पल्स की पोज़िटिव सिरा के दौरान अवस्था बदलने के लिए डिज़ाइन की गई हैं, जिसको “**पोज़िटिव एडज ट्रिगर्ड**” टी फ्लिप फ्लॉप कहा जाता है और कुछ अन्य फ्लिप फ्लॉप, जो नेगटिव सिरा के दौरान अवस्था बदलने के लिए डिज़ाइन की गई हैं उसे “**नेगटिव एडज ट्रिगर्ड**” टी फ्लिप फ्लॉप कहा जाता है। ये दोनों को नीचे दर्शाए प्रतीकों के अनुसार दर्शाते हैं।



क) पोज़िटिव एडज ट्रिगर्ड

ख) नेगटिव एडज ट्रिगर्ड

चित्र 4.8. एडज ट्रिगर्ड टी फ्लिप फ्लॉप के प्रतीक

एक फ्लिप फ्लॉप, प्रत्येक क्लॉक पल्स के साथ एक बार अवस्था परिवर्तन करता है। ये परिवर्तन, पिछली अवस्था के विपरीत रूप में होती है। नीचे तालिका 4.5 और 4.6 में दिए गए ट्रूथ टेबल इसका स्पष्ट वर्णन करता है। Q और Q का पूर्व या पिछली अवस्था क्रमशः 1 और 0 माना गया है।

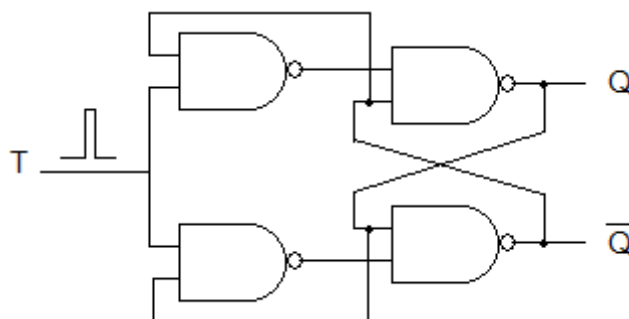
T	Q	$\bar{Q}$	अवस्था
--	1	0	पूर्व अवस्था
↑	0	1	रीसेट
↑	1	0	सेट
↑	0	1	रीसेट
↑	1	0	सेट

तालिका 4.5. पोज़िटिव एडज ट्रिगर्ड टी फ्लिप फ्लॉप का ट्रूथ टेबल

T	Q	$\bar{Q}$	अवस्था
--	1	0	पूर्व अवस्था
↓	0	1	रीसेट
↓	1	0	सेट
↓	0	1	रीसेट
↓	1	0	सेट

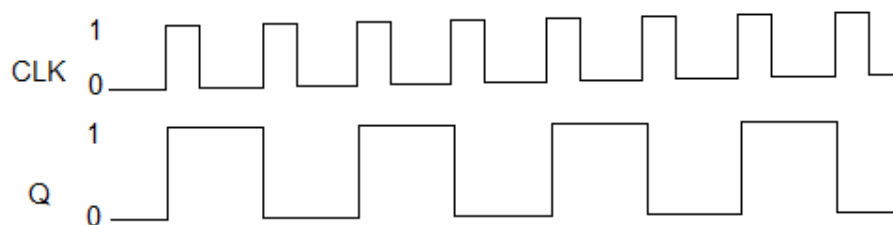
तालिका 4.6. नेगेटिव एडज ट्रिगर्ड टी फ्लिप फ्लॉप का ट्रूथ टेबल

प्रत्येक क्लॉक पल्स के साथ अवस्था को विपरीत होने के इस स्वभाव को “टोग्लिंग” कहा जाता है।



चित्र 4.9. टी फ्लिप फ्लॉप का लॉजिक आरेख

टी फ्लिप फ्लॉप के क्लॉक इनपुट और Q आउटपुट के बीच समय तथा आवृत्ति का संबंध चित्र 4.10 में दर्शाया गया है। इसके अनुसार आउटपुट वेवफॉर्म की आकृति, इनपुट क्लॉक के समान ही है लेकिन परिवर्तन केवल इतना है कि, आउटपुट सिगनल आवृत्ति इनपुट क्लॉक आवृत्ति के आधा हो जाता है। इसका मतलब है कि, टी फ्लिप फ्लॉप एक आवृत्ति विभाजक के तरह कार्य करता है, यानी यह क्लॉक आवृत्ति को 2 से विभाजन करता है।



चित्र 4.10. टी फ्लिप फ्लॉप द्वारा आवृत्ति विभाजन

#### 4.4 जे के फ्लिप फ्लॉप:

यह एक बहुमुखी और सभी फ्लिप फ्लॉप के बीच अधिक उपयोगी है। कार्यशैली में यह एसआर और टी दोनों फ्लिप फ्लॉप के संयोजन हैं। जेके फ्लिप फ्लॉप भी एक एडज ट्रिगर्ड फ्लिप फ्लॉप है। इसका चित्रात्मक प्रस्तुति नीचे चित्र 4.11 में दर्शाया गया है।



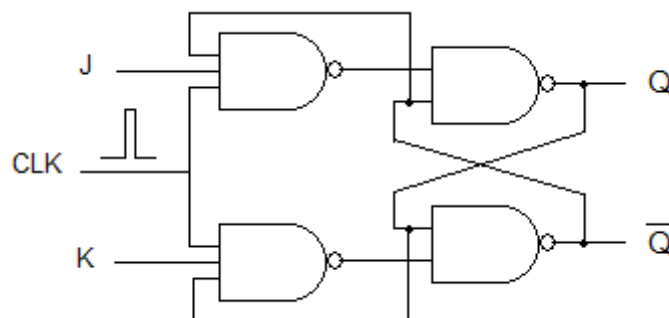
क) पोज़िटिव एडज ट्रिगर्ड

ख) नेगेटिव एडज ट्रिगर्ड

चित्र 4.11. एडज ट्रिगर्ड जे के फ्लिप फ्लॉप का प्रतीक

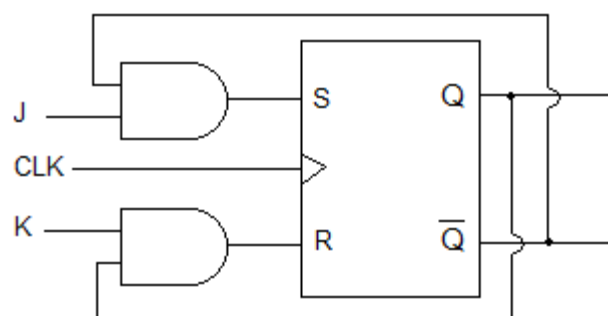
जेके फ्लिप फ्लॉप और एसआर फ्लिप फ्लॉप की कार्यशैली एक समान है, सिवाय कि विशेष मामले में जैसे जे = के = 1 (टोगल अवस्था) में होते हैं। इस मामले में, जब भी एक क्लॉक पल्स सिरा आती है, तब सर्किट के आउटपुट, उसके मौजूदा अवस्था के विपरीत अवस्था में बदल जाती है। जेके फ्लिप फ्लॉप की यह सुविधा, उसको अन्य सभी फ्लिप फ्लॉप से अधिक उपयोगी बनाते हैं।

नीचे दिए गए चित्र 4.12, जेके फ्लिप फ्लॉप का स्टैंडर्ड निर्माण पद्धति दर्शाता है। इसमें 4 NAND गेट का उपयोग की गई हैं। जे और के जो इनपुट गेट से जुड़े हैं, उसको Q और Q आउटपुट के साथ फीडबैक मिलता है, जिससे एक समय में एक ही इनपुट NAND गेट सक्रिय होते हैं या अनुकूल स्थिति में होते हैं।



चित्र 4.12. SR की जे के फ्लिप फ्लॉप में परिवर्तन

नीचे दी गई आरेख, दो अतिरिक्त NAND गेट के साथ, एक एसआर फ्लिप फ्लॉप को जेके फ्लिप फ्लॉप के तरह कैसे उपयोग कर सकते हैं इसको दिखाया है। एसआर फ्लिप फ्लॉप के इनपुट की AND गेट में फीडबैक, एक समय में एस और आर दोनों को 1 होने से रोकता है।



चित्र 4.13. जे के फ्लिप फ्लॉप का लॉजिक आरेख

### फ्लिप फ्लॉप

जेके फ्लिप फ्लॉप को एक बहुमुखी फ्लिप फ्लॉप माना जाता है क्योंकि सीधे या कुछ मामूली संशोधनों के साथ, इससे अन्य फ्लिप फ्लॉप के कार्यों को प्राप्त करना संभव है। इसे एक यूनिवर्सल फ्लिप फ्लॉप भी कहना उचित हो सकता है। एक पोज़िटिव एडज ट्रिगर्ड जेके फ्लिप फ्लॉप का ट्रूथ टेबल तालिका 4.7 में दर्शाया गया है।

**टी एवं जेके फ्लिप फ्लॉप के प्रयोग:** यह दोनों फ्लिप फ्लॉप का उपयोग बाइनरी काउंटर्स के निर्माण में किया जाता है।

C	J	K	Q	$\overline{Q}$	State	Remark
0	X	X	$Q_n$	$\overline{Q_n}$	previous	No change
1	X	X	$Q_n$	$\overline{Q_n}$	previous	
↑	1	0	1	0	Set	SR Mode
↑	0	1	0	1	Reset	
↑	0	0	$Q_n$	$\overline{Q_n}$	previous	
↑	1	1	1	0	Set	Toggle mode
↑	1	1	0	1	Reset	
↑	1	1	1	0	Set	

तालिका 4.7. जे के फ्लिप फ्लॉप का ट्रूथ टेबल

**वस्तुनिष्ठ:**

1. फ्लिप फ्लॉप एक \_\_\_\_\_ साधन है।
2. एसआर फ्लिप फ्लॉप में, एस और आर का मतलब \_\_\_\_\_ और \_\_\_\_\_ है।
3. डी फ्लिप फ्लॉप का उपयोग \_\_\_\_\_ में होता है।
4. टी फ्लिप फ्लॉप का उपयोग मुख्य रूप से \_\_\_\_\_ के निर्माण में होता है।
5. \_\_\_\_\_ फ्लिप फ्लॉप को यूनिवर्सल फ्लिप फ्लॉप कहा जा सकता है।

**विषयनिष्ठ:**

1. फ्लिप फ्लॉप क्या है? उसकी कार्यशैली का वर्णन करें।
2. फ्लिप फ्लॉप के विभिन्न प्रकार का उल्लेख करें और जेके फ्लिप फ्लॉप का वर्णन करें।
3. एसआर फ्लिप फ्लॉप का लॉजिक आरेख बनाएं और उसकी कार्यशैली का वर्णन करें।
4. एसआर और जेके फ्लिप फ्लॉप के बीच तीन अंतर का उल्लेख करें।
5. कुछ फ्लिप फ्लॉप को क्यों एडज ट्रिगरिंग की जरूरत होती है? एक संदर्भ के साथ समझाएं।

## अध्याय 5

## अनुक्रमिक (सीक्वेन्शियल) लॉजिक सर्किट

**5.0** फ्लिप फ्लॉप को मूल डिजाइन तत्व के रूप में उपयोग करके नीचे बताए गए लॉजिक उपकरणों को बनाया जा सकता है। इस उपकरणों को **अनुक्रमिक लॉजिक सर्किट** कहा जाता है।

1. काउंटेर्स
2. रजिस्टर्स
3. मेमोरी डिवाइजेस

**5.1 काउंटेर्स:** काउंटर एक साधन है, जो किसी भी घटना, (जैसे क्लॉक पल्स) के गिनती के लिए उपयोग किया जाता है। यह काउंटिंग सिर्फ बाइनरी रूप में होती है, इसलिए इसको बाइनरी काउंटर कहा जाता है। फ्लिप फ्लॉप को मूल निर्माण तत्व के रूप में उपयोग करके काउंटर बनाया जाता है इसलिए यह एक अनुक्रमिक लॉजिक सर्किट है। इस उद्देश्य के लिए, टी फ्लिप फ्लॉप या जेके फ्लिप फ्लॉप का उपयोग किए जा सकते हैं।

मूल रूप से काउंटर दो प्रकार के होते हैं, जैसे:

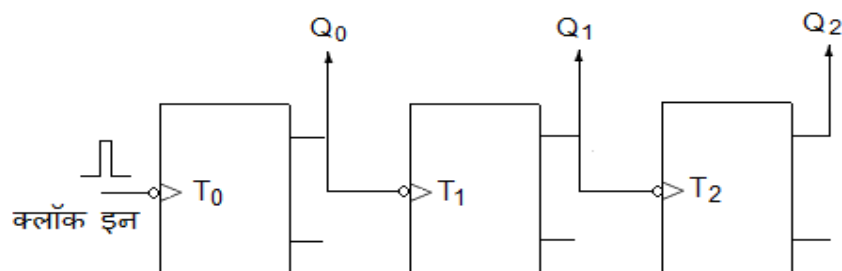
**5.1.1 एसिंक्रोनस या रिपल काउंटर:** इस प्रकार के काउंटर्स में एक बाह्य क्लॉक सिगनल को पहला फ्लिप फ्लॉप के इनपुट में जोड़ते हैं और उसके आउटपुट को दूसरे फ्लिप फ्लॉप के क्लॉक इनपुट में क्लॉक सिगनल की तरह जोड़ते हैं।

**5.1.2 सिंक्रोनस काउंटर:** सिंक्रोनस काउंटर में बाह्य क्लॉक सिगनल को सभी फ्लिप फ्लॉप से एक साथ जोड़ते हैं। रिंग काउंटर तथा जोहसन काउंटर सिंक्रोनस काउंटर के उदाहरण हैं।

**5.2 रिपल काउंटर:**

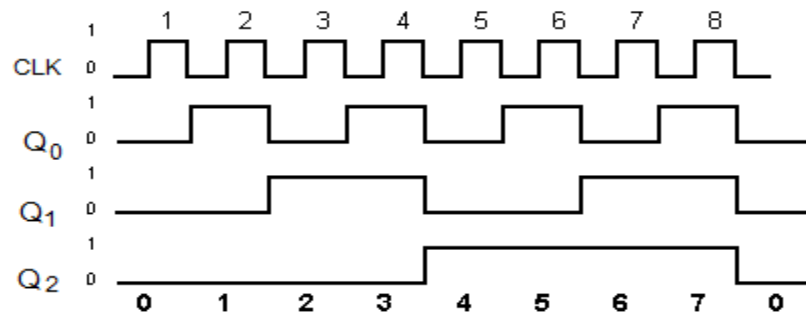
रिपल काउंटर्स में प्रयुक्त फ्लिप फ्लॉप एक ही समय में अपने आउटपुट अवस्था नहीं बदलती हैं बल्कि एक के बाद एक, क्रमबद्ध रूप में बदलते हैं। जबकि, सिंक्रोनस काउंटर में सभी फ्लिप फ्लॉप के आउटपुट एक साथ बदलते हैं।

टी फ्लिप फ्लॉप से बनाया हुआ एक 3-बिट रिपल काउंटर का चित्र नीचे दर्शाया गया है।



चित्र 5.1. 3-बिट रिपल काउंटर





चित्र 5.2. काउंटर में क्लॉक और आउटपुट तरंगों के बीच संबंध

पहला फ्लिप फ्लॉप के  $T_0$  में, क्लॉक पल्स सीधे लागू किए जाते हैं। इसके आउटपुट  $Q_0$  को दूसरे फ्लिप फ्लॉप के  $T_1$  में क्लॉक सिगनल के रूप में लगाए जाते हैं। दूसरे फ्लिप फ्लॉप के आउटपुट  $Q_1$  को तीसरे फ्लिप फ्लॉप के  $T_2$  से जोड़ा जाता है।

प्रत्येक क्लॉक पल्स के नेगटिव सिरा से  $Q_0$ , 0 और 1 के बीच परिवर्तित होते हैं। इस परिवर्तन, नेगटिव सिरा पर ट्रिगरिंग के लिए  $T_1$  में स्पंदन उत्पन्न करती है। यह  $Q_1$  पर परिवर्तन में परिणमित होते हैं और  $Q_1$ ,  $T_2$  के लिए क्लॉक के रूप में कार्य करते हैं। यह  $Q_0$ ,  $Q_1$  और  $Q_2$  आउटपुट पर एक व्यवस्थित परिवर्तन में परिणमित होते हैं।

यह परिवर्तन निम्न क्रम में है। मान लें कि  $Q$  आउटपुटों पर प्रारंभिक स्थिति 000 है। अब हम,  $T_0$  पर क्लॉक पल्स के एक स्ट्रीम को लगाते हैं जिससे यह होते हैं कि:

Pulse		$Q_2$	$Q_1$	$Q_0$
1	-----	0	0	1
2	-----	0	1	0
3	-----	0	1	1
4	-----	1	0	0
5	-----	1	0	1
6	-----	1	1	0
7	-----	1	1	1
8	-----	0	0	0

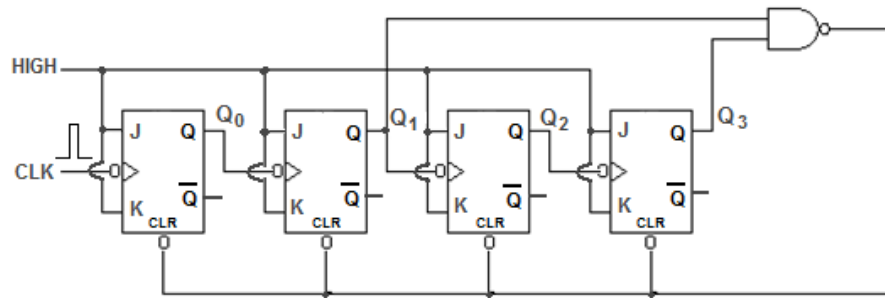
1 से 7 तक,  $Q_0$ ,  $Q_1$  और  $Q_2$  आउटपुट पर संघटित बाइनरी मूल्य लागू किए गए पल्सों की संख्या के बराबर हैं। इसका मतलब है, फ्लिप फ्लॉप की इस व्यवस्था बाइनरी क्रम में लागू पल्सों की गिनती कर सकती है। 8वा क्लॉक पल्स के साथ, यह प्रारंभिक रीडिंग 000 पर वापस आ जाते हैं। फिर से क्लॉक पल्स लागू किया जाए तो, वही गिनती क्रम दोहराई जाएगी।

**काउंटर का मापांक:** इस सर्किट व्यवस्था, अधिकतम 8 क्लॉक पल्सों की गिनती करने में सक्षम है, इसलिए इसको काउंटर का मापांक (मोड) कहते हैं। किसी भी बाइनरी काउंटर का मापांक, इस तरह, आसानी से जाना जा सकते हैं।

$$\text{मापांक} = 2^n, \quad n, \text{ काउंटर में बिटों (फ्लिप फ्लॉप) की संख्या है।}$$

### 5.3 दशक (डिकेड) काउंटर:

एक बाइनरी काउंटर, जिसकी गिनती अनुक्रम में दस अवस्थाएं (मोड-10) होती हैं, उसे दशक काउंटर कहा जाता है। 4-बिट काउंटर का उपयोग करके एक दशक काउंटर बनाया गया है, जिसको मापांक-10 काउंटर बनाने के लिए गिनती अनुक्रम दस पर संक्षिप्त की गई है। हमारे रोज़ के कामकाज में, विभिन्न उपकरणों तथा पैनल मीटरों पर डेसिमल रीडिंग प्राप्त करने के लिए दशक काउंटर एक अनिवार्य उपकरण है। नीचे दिए गए सर्किट एक दशक काउंटर का कार्यान्वयन है।



चित्र 5.3. दशक काउंटर

ऊपर की सर्किट में, काउंटर जब एक बार 10 (1010) तक गिनती करते हैं, सभी फ्लिप फ्लॉप क्लियर हो जाते हैं। ध्यान दें कि, दस की गणना को डिकोड करने के लिए सीर्फ Q1 और Q3 का ही उपयोग किया जाता है। किसी भी अन्य अवस्था (0 से 9 तक) में, Q1 और Q3 दोनों, एक ही समय में उच्च अवस्था में नहीं होते हैं इसलिए इसको आंशिक डिकोडिंग कहते हैं।

तालिका 5.1 में, दशक काउंटर की गिनती अनुक्रम दर्शाया गया है।

Clock Pulse	Q3	Q2	Q1	Q0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

तालिका 5.1. दशक काउंटर की गिनती अनुक्रम

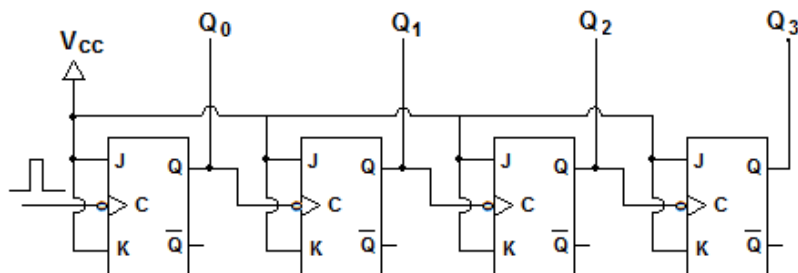
### 5.4 अप तथा डाउन काउंटर:

3-बिट बाइनरी काउंटर जिसकी पहले चर्चा की थी, वह पल्स की गिनती, आरोही क्रम में यानी ऊपर की ओर करती है। इसको अप काउंटर कहा जाता है। अप काउंटर में सभी फ्लिप फ्लॉप, नेगेटिव एडज ट्रिगर्ड प्रकार का उपयोग करते हैं।

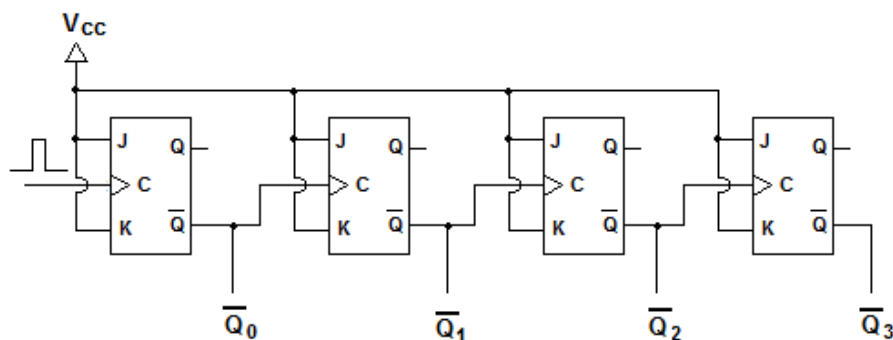
कुछ निश्चित प्रयोगों में अवरोही क्रम या उतरती क्रम की गिनती की भी जरूरत होती है। उदाहरण के लिए, एक फुटबॉल मैच में, घड़ी को कुछ निश्चित अवधी जैसे 50 मिनट के लिए सेट की जाती है। जब मैच शुरू होती है तो, घड़ी नीचे की ओर गिनती आरंभ करती है। हर मिनट में गिनती 1 बार घटती है यानी, 49, 48, 47, आदि। जब घड़ी की गिनती 00 पर पहुँचती है तो, मैच बंद करना है।

इसी तरह एक और अच्छी उदाहरण, रॉकेट छोड़ते समय की अवरोही गिनती है।

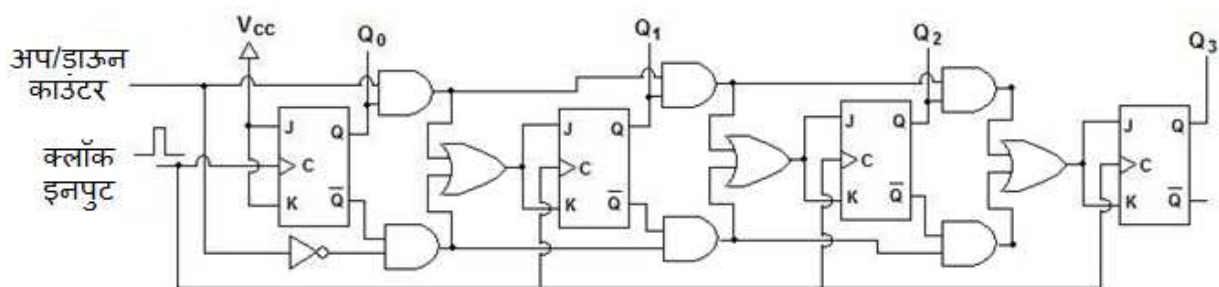
सामान्यतया, सभी अप काउंटर नेगटीव एडज ट्रिगर्ड फ्लिप फ्लॉप का उपयोग करते हैं जबकी सभी डाऊन काउंटर पोज़िटिव एडज ट्रिगर्ड फ्लिप फ्लॉप का उपयोग करते हैं। आरोही गिनती के लिए Q आउटपुट और अवरोही गिनती के लिए  $\bar{Q}$  आउटपुट को लेकर एक ही काउंटर को अप या डाऊन काउंटर के तरह उपयोग किए जा सकते हैं। चित्र 5.4 और 5.5 में क्रमशः एक 4-बिट अप काउंटर और 4-बिट डाऊन काउंटर के लॉजिक आरेख दर्शाए गए हैं, जबकी चित्र 5.6 अप/डाऊन काउंटर दर्शाते हैं।



चित्र 5.4. नेगटीव एडज ट्रिगर्ड फ्लिप फ्लॉप के साथ 4-बिट अप काउंटर



चित्र 5.5. पोज़िटिव एडज ट्रिगर्ड फ्लिप फ्लॉप के साथ 4-बिट अप काउंटर



चित्र 5.6. सिन्क्रोनास अप/डाऊन काउंटर

## 5.5 रजिस्टर्स:

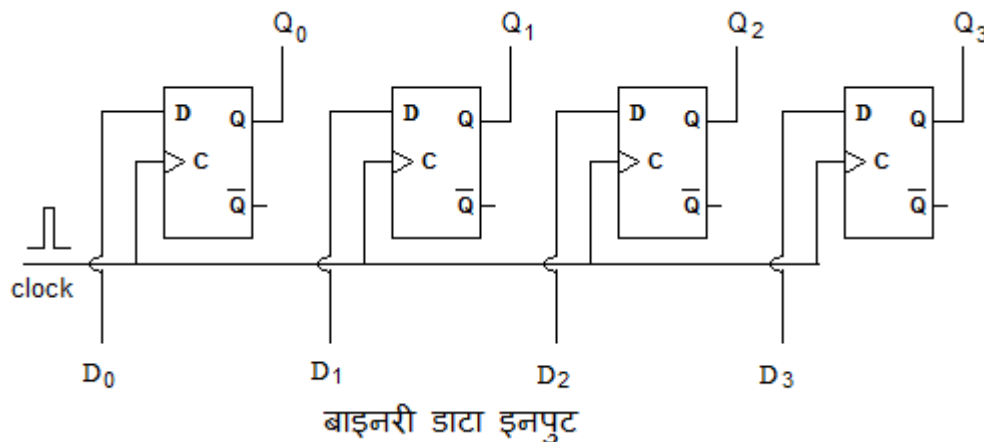
काउंटर के तरह, रजिस्टर भी एक अनुक्रमिक लॉजिक सर्किट है। मतलब, यह भी मूल साधन के तरह D फ्लिप फ्लॉप के उपयोग करके बनाया जाता है। जैसे इसके नाम बताते हैं, वैसे ही, रजिस्टर, बाइनरी जानकारी यानी 0 और 1 को रजिस्टर या संग्रह करने के लिए उपयोग किए जाते हैं।

मोटे तौर पर, रजिस्ट्रों के विभाजन इस तरह किए जा सकते हैं:

- क) डाटा/मेमोरी रजिस्टर
- ख) शिफ्ट रजिस्टर

### 5.5.1 मेमोरी रजिस्टर:

इसका उपयोग मेमोरी डिवाइजों में किया जाता है। मेमोरी डिवाइजों में प्रत्येक स्थान D फ्लिप फ्लॉप से बना हुआ रजिस्टर होता है।



चित्र 5.7. मेमोरी रजिस्टर

चित्र 5.7 एक 4-बिट रजिस्टर दर्शाते हैं, जो एक समय में 4 बाइनरी बिट मूल्य का संग्रह कर सकते हैं, जैसे 1101, 0101 और 1011 आदि। बाइनरी जानकारी, जिसका संग्रह करना है, उसको D इनपुट ( $D_0$  से  $D_3$ ) में लगाए जाते हैं और प्रत्येक क्लॉक पल्स के साथ यह Q ( $Q_0$  से  $Q_3$ ) आउटपुट में संग्रह किए जाते हैं।

### 5.5.2 शिफ्ट रजिस्टर:

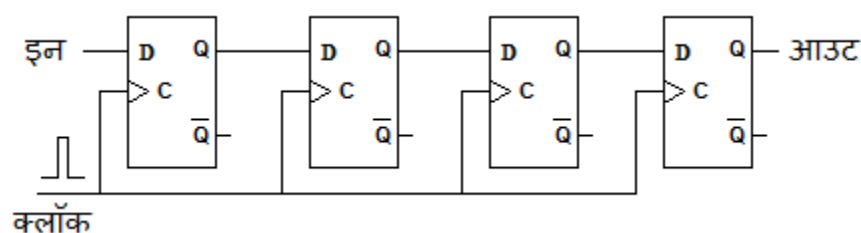
यह भी मेमोरी रजिस्टर की तरह ही है, लेकिन क्लॉक पल्स के प्रयोग के साथ इसमें संग्रहित जानकारियों को एक बिट स्थान से बाएं से दाएं या दाएं से बाएं स्थान बदला जा सकता है।

### 5.5.3 शिफ्ट रजिस्ट्रों के प्रकार:

डाटा इनपुट (लोडिंग) और आउटपुट मोड के आधार पर शिफ्ट रजिस्ट्रों को चार प्रकार में वर्गीकृत किया गया है। ये हैं:

#### क) सीरियल इनपुट सीरियल आउटपुट (SISO) शिफ्ट रजिस्टर:

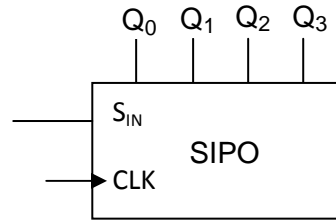
बिट लंबाई के बावजूद डाटा की सीरियल लोडिंग और पुनःप्राप्ति के लिए इस प्रकार की शिफ्ट रजिस्टर में सिर्फ एक एकल इनपुट तथा एकल आउटपुट टर्मिनल प्रदान की गई है। प्रत्येक नए बिट बाहर से लोड करते ही जो डाटा पहले से ही SISO में है वह एक बिट स्थान आगे क्रमानुसार शिफ्ट हो जाते हैं। चित्र 5.7 में SISO लॉजिक दर्शाया गया है।



चित्र 5.7. सीरियल इनपुट सीरियल आउटपुट (SISO) शिफ्ट रजिस्टर के निर्माण

**ख) सीरियल इनपुट पैरेलल आउटपुट (SIPO) शिफ्ट रजिस्टर:**

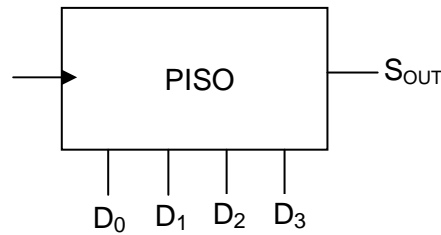
SIPO में डाटा को क्रमानुसार भेजने के लिए एक एकल इनपुट टर्मिनल और डाटा की पुनः प्राप्ति के लिए विभिन्न आउटपुट टर्मिनल होते हैं। चित्र 5.8 में SIPO के प्रतीक दिए गए हैं।



चित्र 5.8. SIPO का प्रतीक

**ग) पैरेलल इनपुट सीरियल आउटपुट (PISO) शिफ्ट रजिस्टर:**

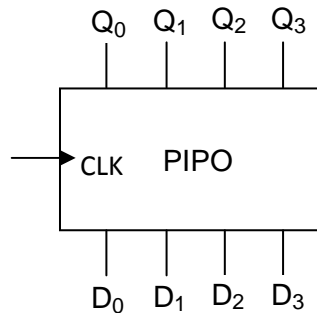
इस प्रकार के शिफ्ट रजिस्टर में डाटा एक एकल क्लॉक पल्स के साथ समांतर रूप में भेजते हैं और उसके बाद डाटा को एक समय में एक बिट के रूप में क्रमानुसार शिफ्ट किए जा सकते हैं। चित्र 5.9 में SIPO का प्रतीक दिया गया है।



चित्र 5.9. PISO का प्रतीक

**घ) पैरेलल इनपुट पैरेलल आउटपुट (PIPO) शिफ्ट रजिस्टर:**

PIPO में डाटा को भेजने तथा पुनःप्राप्ति समांतर रूप में करने की सुविधा उपलब्ध है। मतलब है कि, इसमें विभिन्न और समान संख्या में इनपुट/आउटपुट टर्मिनल होते हैं। ऊपर के तीनों रजिस्ट्रों के विपरीत इसके अंदर वास्तव में डाटा की शिफ्टिंग नहीं होती है। चित्र 5.10 में SIPO के प्रतीक दिए गए हैं। एक क्लॉक पल्स के साथ ही सभी डाटा बिट्स को भेजे जाते हैं।



चित्र 5.10. PIPO का प्रतीक

**वस्तुनिष्ठ:**

1. एक काउंटर \_\_\_\_\_ फ्लिप फ्लॉप से बना होता है।
2. सामान्यतया डाउन काउंटर के निर्माण के लिए \_\_\_\_\_ ट्रिगर्ड फ्लिप फ्लॉप का उपयोग करते हैं।
3. मुख्य रूप से रजिस्ट्रों का \_\_\_\_\_ प्रकार होते हैं।
4. रजिस्ट्रों के निर्माण में सिर्फ \_\_\_\_\_ फ्लिप फ्लॉप का ही उपयोग होता है।
5. रिपल और सिंक्रोनोस काउंटर में किस काउंटर तेज है? \_\_\_\_\_.

**विषयनिष्ठ:**

1. एक क्रमानुसार लॉजिक डिवाइज क्या है? किसी एक का वर्णन करें।
2. एक सरल 3-बिट काउंटर का चित्र बनाएं और वर्णन करें।
3. रिपल काउंटर और सिंक्रोनोस काउंटर के बीच के अंतर का उल्लेख करें।
4. अप और डाउन काउंटर का वर्णन करें।
5. एक रजिस्टर क्या है? एक 4-बिट रजिस्टर का चित्र बनाएं।

## अध्याय 6

### मेमोरी

#### 6.0 मेमोरी क्या है?

मेमोरी, स्थायी या अस्थायी रूप से जानकारी संग्रहीत करने के लिए बड़ी संख्या में मेमोरी रजिस्ट्रों को एक सरणी में क्रमबद्ध तरीके से रखने वाला एक साधन है। कंप्यूटर, इलेक्ट्रॉनिक एक्सचेंज, मोबाइल फोन, एसएसआई, एडब्ल्यूएस आदि प्रोग्राम योग्य सिस्टम में डाटा और प्रोग्राम संग्रह करने के लिए मेमोरी का प्रयोग किया जाता है। मेमोरी को दो मुख्य समूहों में वर्गीकृत किया गया है, ये हैं RAM और ROM.

**1. RAM - रैंडम एक्सेस मेमोरी अथवा रीड/राइट मेमोरी:** इस प्रकार के मेमोरी में दोनों डाटा संग्रहण (लेखन) और डाटा पुनः प्राप्ति (पढ़ने) के ऑपरेशन संभव हो सकते हैं। इसलिए इसे रीड/राइट मेमोरी भी कहा जाता है।

रैम एक वोलाटाईल मेमोरी है। इसका मतलब है कि बिजली सप्लाई निकाले जाने के बाद रैम में संग्रहीत जानकारी मिट जाएगी। कंप्यूटर की वर्किंग मेमोरी रैम का एक उदाहरण है। रैम के आम प्रकार नीचे प्रस्तुत हैं।

1. स्टैटिक रैम (SRAM)
2. डायनामिक रैम (DRAM) तथा
3. नॉन-वोलाटाईल रैम (NVRAM)

#### 6.1.1 स्टैटिक रैम

SRAM फ्लिप फ्लॉप की एक सरणी है, जहाँ फ्लिप फ्लॉप में संग्रहीत बिट तब तक रहेगा जब तक बिजली चली नहीं जाती या कोई अन्य बिट उसकी जगह नहीं ले लेता।

#### 6.1.2 डायनामिक रैम

डी-रैम, बिट को MOSFET गेट सबस्ट्रेट कपासिटन्स पर चार्ज की उपस्थिति या अनुपस्थिति के रूप में संग्रहित करता है। कपासिटन्स में रिसाव होता है, इसीलिए उसे हर कुछ मिल्ली-सेकंड में रिफ्रेश करना पड़ता है। एस-रैम को रिफ्रेश करने की आवश्यकता नहीं है। डी-रैम, एस-रैम से चार गुना ज्यादा घना है और इसलिए सस्ता है। हालांकि, एस-रैम का एक्सेस समय डी-रैम से तेज़ है।

#### 6.1.3 नॉन-वोलाटाईल रैम (NVRAM)

एक NVRAM चिप में, रैम और रॉम दोनों होते हैं। पावर रीसेट के दौरान, रॉम की डाटा रैम में नोन-वोलाटाईल भंडारण के लिए कॉपी कर दी जाती है। बिजली के बंद हो जाने से पहले, सिस्टम, रैम की सारी डाटा को ROM में कॉपी करते हैं। एक NVRAM में रैम को छाया रैम कहा जाता है। इस प्रकार NVRAM, रैम और रॉम दोनों का लाभ देता है।

## 6.2 रीड ओनली मेमोरी (रॉम)

रॉम नॉन-वोलाटाईल है। बिजली हटा देने से रॉम की डाटा लुप्त नहीं होगी। रॉम उसके नाम से केवल 'पढ़ने' का आपरेशन इंगित करता है और रॉम में संग्रहित डाटा और प्रोग्राम पुनः प्राप्त करना संभव है। सभी रॉम कम से कम एक बार प्रोग्राम किया जा सकता है। रॉम उपकरणों को मिटाने या उनकी प्रोग्रामिंग की विधि के आधार पर वर्गीकृत किया जाता है। नीचे दिए अनुसार रॉम, विभिन्न प्रकार के होते हैं।

### 6.1.4 मास्क रॉम

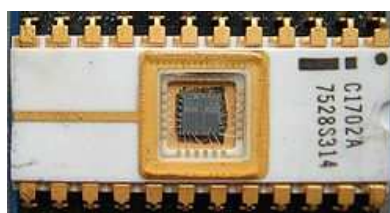
मास्क प्रोग्राम किया गया रॉम में निर्माण प्रक्रिया के दौरान डाटा संग्रहित करता है। यह डाटा स्थायी है और इसे बदला या नष्ट नहीं किया जा सकता है।

### 6.1.5 पी-रॉम (PROM) - प्रोग्राम योग्य रॉम

मेमोरी चिप, खाली निर्मित होता है और प्रोग्रामर इस पर प्रोग्राम/डाटा को संग्रहीत करता है। एक बार डाटा अंतरित होने के बाद इसे बदला या मिटाया नहीं जा सकता है। पी-रॉम को मेमोरी अरे में विशिष्ट ट्रांजिस्टर या डायोड को बिजली से जलाने द्वारा निर्माण करने के बाद केवल एक बार प्रोग्राम किया जा सकता है। इसलिए इसे एक बार प्रोग्राम योग्य मेमोरी (OTP) भी कहा जाता है।

### 6.1.6 EPROM - इरेसेबल पी-रॉम

इसे फ्लॉटिंग गेट MOSFETs के प्रयोग करके बनाया जाता है इसलिए यह उच्च घनता में संग्रहण की सुविधा प्रदान करती है। पी-रॉम से भिन्न, EPROM को पराबैंगनी (यूवी) प्रकाश के उपयोग से मिटाया जा सकता है और फिर पुनः प्रोग्राम किया जा सकता है। यह बाइनरी 1 और 0 को क्रमशः 'चार्ज' और 'नो-चार्ज' के रूप में स्टोर करते हैं। यह मुख्य रूप से कंप्यूटर, मोबाइल फोन आदि सिस्टम में स्थायी प्रोग्राम के संग्रहण के लिए उपयोग किया जाता है। EPROM, जो हाल के दिनों तक कंप्यूटर में BIOS (स्टार्ट-अप) प्रोग्राम के संग्रहण के लिए एक बहुत ही आम मेमोरी चिप था अब इसकी जगह फ्लैशरैम का इस्तेमाल करने लगे हैं।



चित्र 6.1 EPROM चिप का तस्वीर

### 6.1.7 EEPROM - विद्युत इरेसेबल PROM:

यह E<sup>2</sup>PROM के नाम से भी जाना जाता है। EPROM से भिन्न इसका एक लाभ यह है कि इसमें से डाटा को विद्युत से मिटाया जा सकता है और पराबैंगनी प्रकाश की जरूरत नहीं है। लेकिन इसका मुख्य नुकसान यह है कि इसमें प्रोग्राम को मिटाना या लिखना बाइट के स्तर पर होने से EPROM की तुलना में प्रोग्रामिंग में लंबी समय लेती है।



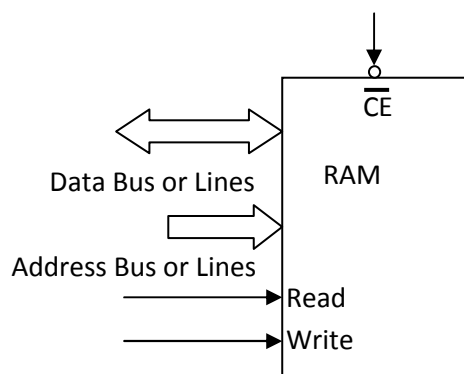
### 6.1.8 फ्लैशरैम

फ्लैश मेमोरी, अक्सर कंप्यूटर में कंट्रोल कोड जैसे मूल इनपुट/आउटपुट सिस्टम (BIOS) की संग्रहण के लिए उपयोग करते हैं तथा अपने को अद्यतन करना भी आसान बनाता है। इसकी एक अन्य लाभ यह है कि EPROM और EEPROM की तरह अलग से उच्च प्रोग्रामिंग वोल्टेज (10 V या 24V DC) की जरूरत नहीं है। इसके कार्य करने तथा प्रोग्रामिंग के लिए एक ही सप्लाइ वोल्टेज की ज़रूरत होती हैं।

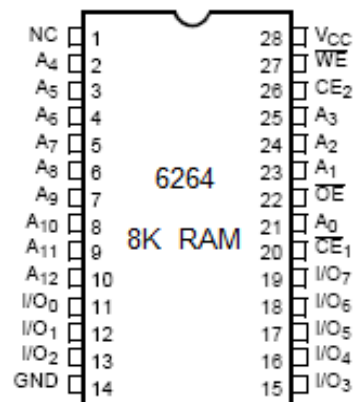
**फ्लैश मेमोरी के फायदे:**

- इसका मुख्य लाभ यह है की यह EPROM के कम लागत तथा उच्च घनत्व सुविधा के साथ EEPROM के विद्युत द्वारा मिटाने की सुविधाओं को भी जोड़ती हैं।
- एक और लाभ यह है कि इसको सर्किट से हटाये बिना ही मिटा और पुनः प्रोग्रामिंग किया जा सकता है।
- लेकिन इसकी एक गैर फायदा यह है कि फ्लैशरैम को रैम के स्थान पर प्रयोग नहीं किया जा सकता है क्योंकि रैम बाइट के स्तर पर पता योग्य है।

**मेमोरी उपकरणों के चिह्न:**

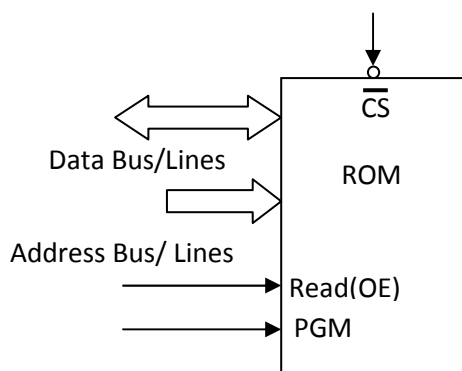


(क) RAM के चिह्न

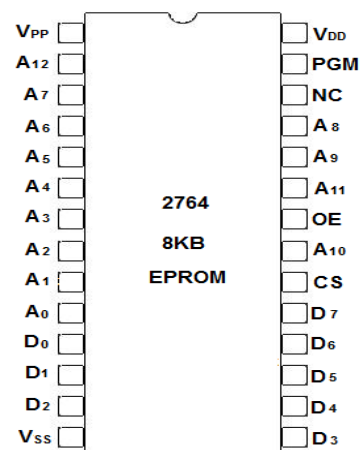


(ख) IC 6264 (8k RAM) के पिन

चित्र 6.2



(क) ROM के चिह्न



(ख) IC 2764 (8k EPROM) के पिन

चित्र 6.3

ऊपर दिए गए दो चित्र 6.2 तथा 6.3 क्रमशः रैम और रॉम के चिह्न और पिन ले-आउट दर्शाते हैं।

**वस्तुनिष्ठ:**

1. मेमोरी, मुख्य रूप से \_\_\_\_\_ प्रकार में विभाजित है।
2. रॉम को \_\_\_\_\_ मेमोरी कहा जाता है।
3. दरअसल, रैम को \_\_\_\_\_ नाम से बुलाना चाहिए।
4. EEPROM की मुख्य नुकसान \_\_\_\_\_ है।
5. सभी प्रकार के ROM में आप किसे सबसे अच्छा मानते हैं? \_\_\_\_\_

**विषयनिष्ठ:**

1. मेमोरी क्या है? उसके कितने प्रकार हैं?
2. रैम के विभिन्न प्रकारों का उल्लेख करें और उन्हें संक्षेप में समझाएं।
3. ROM के विभिन्न प्रकारों का उल्लेख करें और उन्हें संक्षेप में समझाएं।
4. फ्लैशरैम के लाभ क्या हैं?
5. आप एक समर्पित कार्यप्रणाली के लिए किस प्रकार की मेमोरी का सुझाव करेंगे और उसकी वजह बताएं।

## अध्याय 7

### डिजिटल कोड

#### 7.0 डिजिटल कोड

डिजिटल इलेक्ट्रॉनिक्स में विभिन्न प्रयोजनों के लिए विभिन्न प्रकार के कोड का उपयोग किए जाते हैं, जैसे कि:

1. अंकों और अक्षरों के प्रतिनिधित्व करने के लिए
2. विविध भाषाओं में अक्षरों के प्रतिनिधित्व करने के लिए
3. संख्या और पाठ अक्षरों को प्रदर्शित करने के लिए
4. संचरण डाटा का प्रतिनिधित्व करने के लिए

इन महत्वपूर्ण कोड में से कुछ पर एक नजर डालें।

#### न्यूमेरिकल कोड

ये कोड डेसिमल अंक को बाइनरी रूप में दर्शाने के लिए उपयोग करते हैं।

1. बीसीडी
2. अतिरिक्त -3
3. ग्रे

#### अक्षरांकीय कोड

अक्षरांकीय वर्णों का प्रतिनिधित्व करने के लिए इस्तेमाल बाइनरी कोड के नाम इस प्रकार हैं।

1. आस्की (ASCII)
2. EBCDIC
3. यूनिकोड

#### ट्रांसमिशन डाटा कोड (लाइन कोडिंग)

संचार लाइन पर प्रसारित करने के लिए एक बाइनरी सिगनल को विद्युत डिजिटल सिगनल में परिवर्तित करने के लिए उपयोग किया जाता है।

1. RZ - रिटर्न टु जीरो
2. NRZ - नोन रिटर्न टु जीरो
3. AMI - आल्टर्नेट मार्क इनवर्शन
4. HDB3 - हाई डेन्सिटी बाइपोलार-3

#### 7.1 न्यूमेरिकल कोड

निम्न न्यूमेरिकल कोड का संक्षेप में अध्ययन करते हैं।

##### 7.1.1 बाइनरी कोडेड डेसिमल - बीसीडी

यह डेसिमल प्रणाली के 10 अंकों को 4-बिट बाइनरी कोड में प्रतिनिधित्व करने के लिए प्रयोग किया जाता है।

0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

**उपयोग का एक उदाहरण:** निम्न तालिका 7.1, कुछ डेसिमल संख्याओं के लिए बीसीडी कोड दर्शाते हैं।

क्र.सं.	डेसिमल संख्या	BCD कोड
1	4	0100
2	25	0010 0101
3	347	0011 0100 0111
4	1060	0001 0000 0110 0000

**तालिका 7.1. कुछ डेसिमल मूल्यों का BCD कोड**

**बीसीडी का गैर-फायदे:** जब दो बीसीडी अंकों की जोड़ की राशी 9 से ज्यादा हो तब राशी की बाइनरी मूल्य अमान्य हो जाता है। इसको सही करने के लिए राशी के साथ 0110 (6) को जोड़ना पड़ते हैं।

### 7.1.2 अतिरिक्त-3 कोड

यह भी बीसीडी की तरह एक ही उद्देश्य के लिए है, लेकिन अलग-अलग 4-बिट बाइनरी कोड के साथ प्रयोग किया जाता है। प्रत्येक अतिरिक्त-3 बाइनरी कोड मूल्य, वास्तविक मूल्य से तीन अधिक का प्रतिनिधित्व किया जाता है। इसलिए, बीसीडी की तरह यह एक भारित कोड नहीं है। अतिरिक्त-3 कोड में डेसिमल अंकों के प्रतिनिधित्व नीचे दिए गए तालिका 7.2 में देखा जा सकता है।

**अतिरिक्त-3 कोड के फायदे:** बीसीडी कोड की तुलना में इसके निम्न फायदे हैं।

- एक डेसिमल संख्या के अतिरिक्त-3 मूल्य का बिट विलोमन (1 का कांप्लीमेंट) घटाव के लिए उसका 9 का कांप्लीमेंट देता है।
- अतिरिक्त-3 मूल्यों का घटाना तथा जोड़ना बहुत आसान है। अंतिम परिणाम के साथ/से 3 (0011) को घटाकर/जोड़कर वास्तविक मूल्य प्राप्त करते हैं।
- योगफल जब 1100 (9) से अधिक है, मतलब जब उतराई हो तब उससे (0011) 3 जोड़ा जाएगा।

**जोड़ने के उदाहरण:** हम निम्नलिखित जोड़ को देखते हैं।

1)  $52 + 36$

52	1000	0101	52 के लिए अतिरिक्त कोड
+ 36	0110	1001	36 के लिए अतिरिक्त कोड
88	1110	1110	यह 88 का सही अतिरिक्त 3 कोड नहीं है, 6 अधिक है
-0011	-0011		अतिरिक्त 3 मूल्य के लिए हर डिजिट मूल्य में 3 घटाए
<u>1011</u>	<u>1011</u>		यह 88 का सही अतिरिक्त 3 मूल्य है

2)  $23 + 39$

		1	
23	0101	0110	23 के लिए अतिरिक्त 3 कोड
+ 39	0110	1100	39 के लिए अतिरिक्त 3 कोड
62	1100	0010	यह 62 का सही अतिरिक्त 3 कोड नहीं है, 6 अधिक है
-0011	-0011		अतिरिक्त 3 मूल्य के लिए हर डिजिट मूल्य में 3 घटाए
<u>1001</u>	<u>0101</u>		यह 62 का सही अतिरिक्त 3 मूल्य है

**घटाने के उदाहरण:**

3)  $83 - 34$

		-1	
83	1011	0110	
- 34	- 0111	0110	
49	0100	1111	दूसरे डिजिट से उधार होने से पहला डिजिट के साथ 3 जोड़ो और दूसरे डिजिट से 3 घटावो
	+0011	-0011	
	<u>0111</u>	<u>1100</u>	62 के लिए अतिरिक्त 3 कोड

इसी को पूरक विधि द्वारा घटाना है तब

83		1011	0110
+ 65	(34 का 9वां कांप्लीमेंट होने से)	+1001	1000
(1) 48		(1) 0100	1110
<u>+1</u>		<u>+1</u>	
49		0100	1111
		+0011	-0011
			पहली डिजिट के साथ 3 जोड़ो और दूसरे डिजिट से 3 घटावो
		<u>0111</u>	<u>1100</u>
			अतिरिक्त 3 कोड में 49

### 7.1.3 ग्रे कोड

यह पहली बार कोई चालू मशीन का शाफ्ट या रॉड की स्थिति का पता करने के लिए और उसकी कोई मानदंडों को बदलने के लिए इस्तेमाल किया गया था। बाद में इस कोड को डाटा ट्रांसमिशन प्रयोजनों के लिए भी अपनाया गया। निम्न तालिका 7.2, डेसिमल संख्या के मूल्यों को सभी तीनों कोड में दिखाया है।

डेसिमल अंक	BCD कोड	अतिरिक्त-3 कोड	ग्रे कोड
0	0000	0011	0000
1	0001	0100	0001
2	0010	0101	0011
3	0011	0110	0010
4	0100	0111	0110
5	0101	1000	0111
6	0110	1001	0101
7	0111	1010	0100
8	1000	1011	1100
9	1001	1100	1101
10	0001 0000	0100 0011	1111
11	0001 0001	0100 0100	1110
12	0001 0010	0100 0101	1010
13	0001 0011	0100 0110	1011
14	0001 0100	0100 0111	1001
15	0001 0101	0100 1000	1000

तालिका 7.2. 1 से 15 तक के डेसिमल अंकों के BCD, अतिरिक्त-3 तथा ग्रे कोड

**बाइनरी से ग्रे कोड में परिवर्तन:**

बाइनरी से समकक्ष ग्रे कोड में परिवर्तन करने के लिए तीन बुनियादी नियम नीचे दिए गए हैं।

(क) बाइनरी में और ग्रे कोड में पहला डिजिट समान रहेगा।

(ख) अगले ग्रे डिजिट के लिए बाइनरी के दो आसन्न बिट्स के प्रत्येक जोड़ी को आपस में जोड़ें।

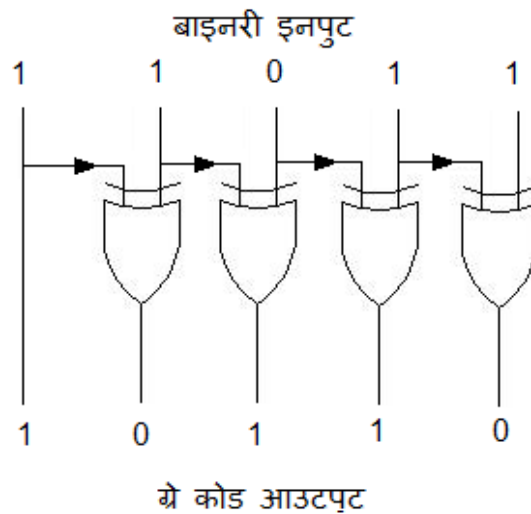
(ग) किसी भी प्रभार का उपेक्षा करें।

**यह निम्न रूप में विस्तार से समझाया जा सकता है:**

- बाइनरी के MSB (एकदम बाएँ तरफ का बिट) को ग्रे कोड में भी ऐसे ही रखें।
- MSB से उसके सटीक दाएँ तरफ की बिट को जोड़ें। कैरी (carry) को छोड़ के योगफल को ग्रे कोड के दूसरे डिजिट में लें।
- अब बाइनरी के MSB से अगले सटीक बिट को दाएँ ओर का सटीक बिट से जोड़ें। इसकी योगफल को ग्रे कोड के अगला बिट में लें।
- बाइनरी के एकदम दाएँ तरफ के दोनों बिट को जोड़ने तक यह दोहराओ, जिसकी योगफल ग्रे कोड का अंतिम बिट होगा।

चित्र 7.1 में स्वचालित रूप से बाइनरी कोड को ग्रे कोड में परिवर्तित करने के लिए प्रयुक्त गेट से बना हुआ एक लॉजिक सर्किट दिखाया है। उदाहरण के लिए, हम एक बाइनरी संख्या 11011 को ग्रे कोड में बदलने के लिए लिया है। इस सर्किट में उपर बताया गए सभी कदम अपने आप लागू हो रहे हैं और समकक्ष ग्रे कोड 10110 के रूप में प्राप्त किया जाता है।

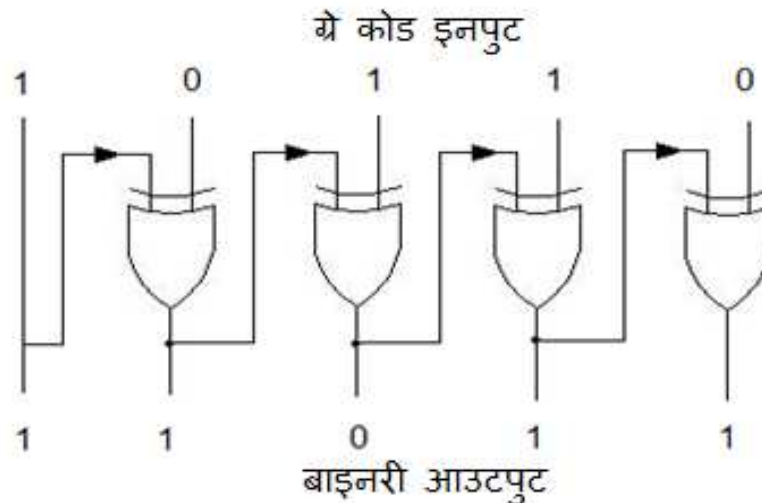
बाइनरी से ग्रे कोड में रूपांतरण के लिए लॉजिक सर्किट:



चित्र 7.1. बाइनरी से ग्रे कोड में रूपांतरण के लिए लॉजिक सर्किट

ग्रे से बाइनरी में रूपांतरण:

उपरोक्त के समान ही ग्रे कोड को बाइनरी में परिवर्तित करने के लिए भी एक प्रक्रिया है। यह प्रक्रिया चित्र 7.2 का परीक्षण करके अच्छी तरह समझ सकता है।



चित्र 7.2. ग्रे कोड से बाइनरी में रूपांतरण के लिए लॉजिक सर्किट

## 7.2 अक्षरांकीय कोड

डिजिटल प्रणालियाँ और कंप्यूटर, जानकारी को डिजिटल (बाइनरी) रूप में संस्करण करते हैं। इसलिए, अक्षरांकीय लिपि, उदाहरण के लिए की-बोर्ड से, बाइनरी रूप में प्रतिनिधित्व करना आवश्यक हैं। इस उद्देश्य से अलग अलग अक्षरों को इंगित करने के लिए 7 या 8 बिट बाइनरी कोड का उपयोग किया जाता है। इन 7 या 8 बिट के प्रत्येक कोड को आम तौर पर इसके समकक्ष हेक्सा-डेसिमल संख्या में प्रतिनिधित्व करते हैं।

### 7.2.1 अमेरिकन स्टैंडर्ड कोड फोर इन्फरमेशन इन्टरचेंज (ASCII):

यह 7 बिट बाइनरी कोड, आम तौर पर, एक की-बोर्ड का अक्षरों के प्रतिनिधित्व करने के लिए उपयोग किया जाता है। तालिका 7.2 में की-बोर्ड के कुछ अक्षरों का आस्की कोड में प्रतिनिधित्व देख सकते हैं।

### 7.2.2 EBCDIC - एक्सटेन्डेड बीसीडी इंटरचेंज कोड:

यह एक 8 बिट (1 बाइट) बाइनरी कोड है जो ASCII के तरह ही इस्तेमाल करते हैं। EBCDIC मुख्य रूप से आईबीएम मैनफ्रेम्स और उसके संगत उपकरणों में प्रयोग किया जाता है। प्रत्येक एकल बाइट EBCDIC को दो भागों में बांटा गया है। पहला चार बिट्स को झोन्स कहते हैं जो अक्षरों का वर्ग की प्रतिनिधित्व करता है और अंत का चार बिट्स डिजिट की प्रतिनिधित्व करता है।

### 7.2.3 Unicode - यूनिवर्सल कोड:

यूनिवर्सल कोड को यूनिकोड कहलाता है। यह दुनिया के सभी भाषाओं और प्रतीकों को समायोजित करने के लिए विकसित की गई है। यह इस उद्देश्य के लिए पर्याप्त से अधिक है, जो 65,536 वर्णों का प्रतिनिधित्व करने के लिए सक्षम एक 16-बिट कोड है। यह वर्तमान में एशियाई भाषाओं का समर्थन करने के लिए unix और Windows पर उपयोग की गई मल्टीबाइट अक्षर सेट की जटिलता को समाप्त करते हैं। यूनिकोड, apple, माइक्रोसॉफ्ट, HP, डिजिटल और आईबीएम सहित कंपनियों के एक संघ द्वारा बनाई गई और 1993 में यूनिकोड को एक मानक उत्पादन की तरह आईएसओ 10646 मानक के साथ विलय कर दिया गया। पहले से ही, विंडोज NT, विंडोज 2000, Windows Vista जैसे नवीनतम ऑपरेटिंग सिस्टम और एमएस ऑफीस 2007 जैसे प्रयोगों के लिए यूनिकोड आधार है। निम्न तालिका इन तीनों कोड में की-बोर्ड अक्षरों का प्रतिनिधित्व देता है।

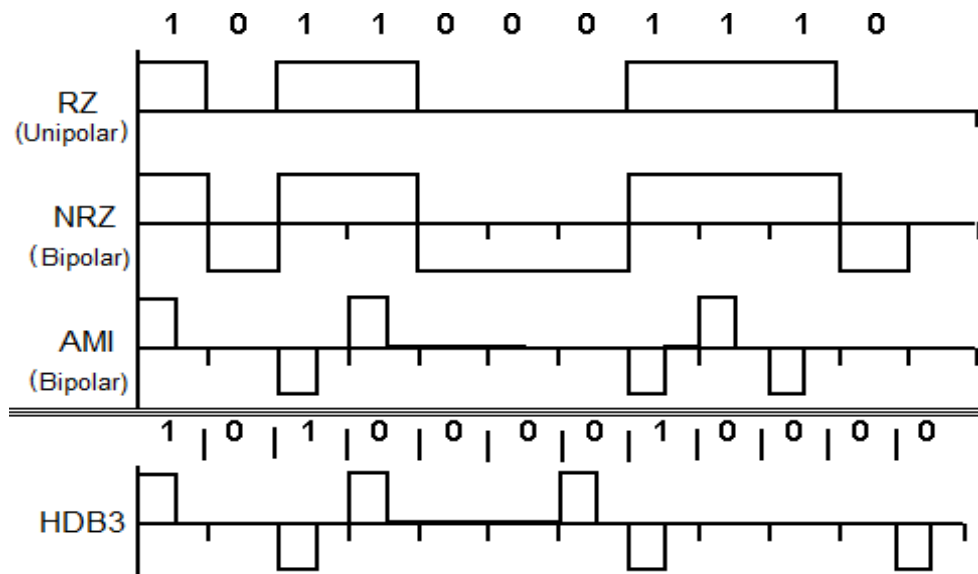
Character to be coded	ASCII	EBCDIC	Unicode
A	100 0001	1100 0001	0000 0000 0100 0001
B	100 0010	1100 0010	0000 0000 0100 0010
C	100 0011	1100 0011	0000 0000 0100 0011
a	110 0001	1000 0001	0000 0000 0110 0001
b	110 0010	1000 0010	0000 0000 0110 0010
c	110 0011	1000 0011	0000 0000 0110 0011
1	011 0000	1111 0001	0000 0000 0011 0001
2	011 0001	1111 0010	0000 0000 0011 0010
3	011 0010	1111 0011	0000 0000 0011 0011
+	010 1011	0100 1110	0000 0000 0010 1011
\$	010 0100	0101 1011	0000 0000 0010 0100
&	010 0110	0101 0000	0000 0000 0010 0110
?	011 1111	0110 1111	0000 0000 0011 1111
क	---	---	0000 1001 0001 0101
त	---	---	0000 1001 0010 0100
म	---	---	0000 1001 0010 1110
स	---	---	0000 1001 0011 1000

तालिका 7.2. ASCII, EBCDIC and यूनिकोड में अक्षरांकीय कोड के प्रतिनिधित्व



### 7.3 डाटा ट्रांसमिशन कोड्स (रेखा कोडिंग):

डाटा ट्रांसमिशन में प्रयुक्त लाइन कोडिंग, बिटों की एक श्रृंखला को डिजिटल वेवफार्म में परिवर्तित करने की प्रक्रिया है। उदाहरण के लिए, कंप्यूटर मेमोरी में संग्रहित डाटा, टेक्स्ट, संख्या, ग्राफिकल चित्र, ऑडियो तथा वीडियो सभी बिटों की श्रृंखला हैं। नीचे दिए गए चित्र में विभिन्न प्रकार के लाइन कोडिंग तकनीकों में प्रेषित सिगनल के वेवफार्म दिखाया है।



चित्र 7.3. विभिन्न लाइन कोड में डिजिटल सिगनल के वेवफार्म

#### वस्तुनिष्ठ:

1. डिजिटल सिस्टम में कोड का एक प्रयोग \_\_\_\_\_ है।
2. बीसीडी, मुख्य रूप से \_\_\_\_\_ के लिए प्रयोग किया जाता है।
3. ग्रे कोड, पहली बार \_\_\_\_\_ के साथ प्रयोग के लिए विकसित किया गया था।
4. आस्की का विस्तार \_\_\_\_\_ है।
5. यूनिकोड एक \_\_\_\_\_ बिट कोड है।

#### विषयनिष्ठ:

1. डिजिटल सिस्टम में, कोड का उपयोग करने का उद्देश्य क्या है?
2. संख्यात्मक कोड का उल्लेख करें और बीसीडी के बारे में समझाएं।
3. विभिन्न अक्षरांकीय कोड के नामों का उल्लेख करें।
4. आस्की कोड के बारे में समझाएं।
5. यूनिकोड और अन्य दो अक्षरांकीय कोड के बीच क्या-क्या अंतर है?

## अध्याय 8

### डिजिटल लॉजिक परिवार

#### 8.0 लॉजिक सर्किट तकनीकी/परिवार:

लॉजिक सर्किट, डायोड, ट्रांसिस्टर, MOSFETs, प्रतिरोध, आदि विभिन्न घटकों का उपयोग करके बनाया जा सकते हैं। प्रयुक्त घटकों के आधार पर लॉजिक सर्किट को अलग-अलग लॉजिक परिवारों में वर्गीकृत किया जाता है। लॉजिक सर्किट के निर्माण के प्रारंभिक दिनों से सर्किट तकनीकियों के नाम निम्न प्रकार हैं।

- डीएल - डायोड लॉजिक
- आरटीएल - रेसिस्टर ट्रांसिस्टर लॉजिक
- डीटीएल - डायोड ट्रांसिस्टर लॉजिक
- एचटीएल - उच्च थ्रेशोल्ड लॉजिक
- टीटीएल - ट्रांसिस्टर ट्रांसिस्टर लॉजिक
- IIL (I<sup>2</sup>L) - एकीकृत इंजेक्शन लॉजिक
- ईसीएल - एमिटर कपल्ड लॉजिक
- MOS - मेटल ऑक्साइड सेमिकंडक्टर लॉजिक (PMOS और NMOS)
- CMOS - कांमप्लिमेन्टरी मेटल ऑक्साइड सेमिकंडक्टर लॉजिक

डीएल, आरटीएल और डीटीएल प्रारंभिक लॉजिक सर्किट डिजाइन तकनीक था। एचटीएल, डीटीएल का ही एक प्रकार है, जिसे बहुत अधिक शोर वाले वातावरण में उपयोग करते हैं। अब ये उपयोग में नहीं हैं। वर्तमान में, केवल टीटीएल और CMOS सबसे व्यापक रूप से उपयोग करते हैं। निम्नलिखित पाँच बुनियादी प्रकार के लॉजिक परिवारों के उपकरण IC के रूप में उपलब्ध हैं।

1. **टीटीएल:** सर्किट डिजाइन के लिए इन उपकरणों में द्वि-ध्रुवीय ट्रांसिस्टर का उपयोग किया जाता है। द्वि-ध्रुवीय ट्रांसिस्टर कट-ऑफ तथा सैचुरेशन क्षेत्र पर ही संचालित करते हैं। टीटीएल उपकरण अन्य लॉजिक परिवारों की तुलना में अधिक धारा की खपत करती है। ये उपकरण 74, 74S, 74LS, 74ALS तथा 74F श्रृंखला में उपलब्ध हैं। इस लॉजिक परिवार में सभी डिजिटल कार्य उपलब्ध हैं।
2. **MOS लॉजिक:** इसमें MOSFET का उपयोग करते हैं। प्रयुक्त MOSFET के आधार पर इसको PMOS और NMOS में वर्गीकृत किया गया है। मेमोरी चिप और माइक्रोप्रोसेसर ज्यादातर MOS लॉजिक परिवार में से हैं।
3. **CMOS लॉजिक:** इस लॉजिक में MOSFET के कांमप्लिमेन्टरी जोड़ी का उपयोग करते हैं। CMOS उपकरण 4000/4500 सीरीज, 74 सीरीज और जो भी भाग उसकी भाग संख्या C से दर्शाते हैं सभी उपलब्ध हैं।
4. **ईसीएल:** यह भी द्वि-ध्रुवीय ट्रांसिस्टर का ही उपयोग करते हैं, लेकिन ये ट्रांसिस्टर लीनियर क्षेत्र में कार्य करता है। ईसीएल उपकरण 10K, 100K तथा 10E सीरीज में उपलब्ध हैं।
5. **Bi-CMOS:** CMOS तकनीकी, वास्तव में, एनलॉग, डिजिटल और मिश्रित सिगनल के इलेक्ट्रॉनिक सर्किट डिजाइन में उपयोग करते हैं। अब द्वि-ध्रुवीय ट्रांसिस्टर तथा MOSFET को भी समायोजित करके CMOS सर्किट को सुधार किया गया है और इसको Bi-CMOS तकनीकी कहते हैं।

## 8.1 डिजिटल IC के वर्गीकरण:

एकीकृत सर्किट में प्रयुक्त एकीकृत घटकों के आधार पर IC को 4 श्रेणियों में वर्गीकृत किए गए हैं, जो निम्न तालिका 8.1 में दर्शाया गया है।

क्र.सं.	IC का वर्गीकरण	समतुल्य गेटों की संख्या	घटकों की संख्या
1	छोटे पैमाने पर एकीकरण (SSI)	12 से कम	99 तक
2	मध्यम पैमाने पर एकीकरण (MSI)	12-99	100-999
3	बड़े पैमाने पर एकीकरण (LSI)	100-999	1,000-9,999
4	बहुत बड़े पैमाने पर एकीकरण (VLSI)	1,000 से ऊपर	10,000 से ऊपर

## 8.2 लॉजिक सर्किटों की महत्वपूर्ण विशेषताएं:

लॉजिक सर्किट, विभिन्न प्रकार के तकनीकों से परिकल्पित किए जा सकते हैं। निम्नलिखित परिमाणों अथवा विशेषताएं सभी लॉजिक परिवारों के लिए लागू होते हैं तथा प्रत्येक लॉजिक परिवार में सर्किट के निर्माण के लिए विभिन्न तकनीकों का उपयोग किया जाता है।

- फैन इन
- फैन आउट
- प्रसारण में देरी
- शोर सीमा
- पावर का अपव्यय
- सिंक तथा सोर्स धाराएं
- लॉजिक के लेवल

### 8.2.1 फैन इन:

फैन इन इनपुटों की संख्या है, जैसे गेटों के लिए होते हैं। एक, दो इनपुट गेट को दो फैन इन और 3 इनपुट गेट को तीन फैन इन होते हैं। एक NOT गेट को हमेशा एक ही फैन इन होता है। चित्र में, एक CMOS आधारित गेट के लिए देरी में फैन इन का असर दिखाया है। साधारणतया जैसे इनपुट की संख्या बढ़ती है, वैसे देरी भी बढ़ती है।

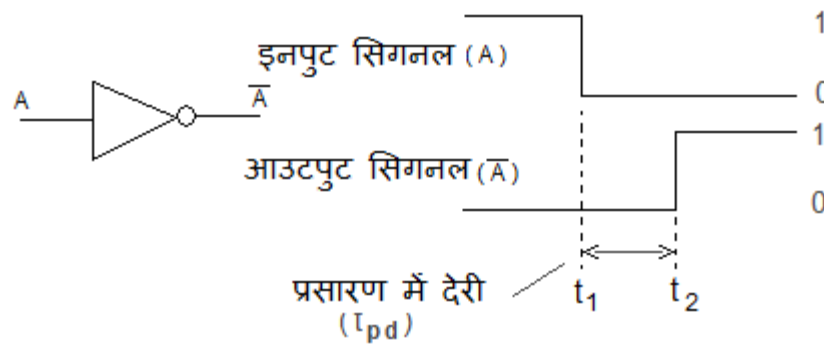
### 8.2.2 फैन आउट:

जब गारंटीकृत वोल्टेज स्तर प्रदान करते हुए गेटों की संख्या, जो एक गेट चला सकता है उसको मानक भार अथवा फैन आउट कहा जाता है। फैन आउट, एक गेट दूसरी गेटों को चलाते समय जो धारा की मात्रा होती है, इस पर निर्भर करता है। एक लॉजिक गेट को उसकी निर्धारित फैन आउट से ज्यादा भार देते हैं, तब निम्न असर पड़ते हैं।

- निम्न अवस्था में आउटपुट वोल्टेज  $V_{OL}$ ,  $V_{OLmax}$  से बढ़ सकते हैं।
- उच्च अवस्था में आउटपुट वोल्टेज  $V_{OH}$ ,  $V_{OHmin}$  से कम हो सकते हैं।
- प्रसारण में देरी में निर्दिष्ट मूल्य से वृद्धि हो सकती है।

### 8.2.3 प्रसारण में देरी (T<sub>pd</sub>):

प्रसारण में देरी, वह देरी है, जो एक लॉजिक गेट या सर्किट, इनपुट सिगनल को आउटपुट टर्मिनल में स्थानांतरित करने में प्रस्तुत करते हैं। इसको ऐसा कहा जा सकता है, “एक गेट उपकरण के इनपुट में लॉजिक संक्रमण तथा आउटपुट में परिणामस्वरूप लॉजिक संक्रमण के बीच का समय अंतराल है।” चित्र 8.1 में एक NOT गेट के प्रसारण में देरी दर्शाया गया है, जो आउटपुट सिगनल का परिवर्तन, इनपुट सिगनल में परिवर्तन के खिलाफ है।



चित्र 8.1 NOT गेट के प्रसारण में देरी

एक गेट के लिए, निम्न से उच्च अवस्था पर या उच्च से निम्न अवस्था पर परिवर्तन के दौरान प्रसारण में देरी अलग-अलग होती हैं। निम्न से उच्च का संक्रमण (t<sub>PLH</sub>) को “टर्न ओन देरी” कहा जाता है तथा उच्च से निम्न का संक्रमण (t<sub>PHL</sub>) को “टर्न ऑफ देरी” कहा जाता है।

### 8.2.4 शोर सीमा:

लॉजिक सर्किट, बाहरी शोर वोल्टेज के प्रभाव के तहत इनपुट और आउटपुट वोल्टेज के स्तर में बदलाव को संभालने के लिए, सक्षम निर्माण करते हैं। बाहरी शोर वोल्टेज को सामान्य डिजिटल सिगनल पर आरोपित करते हैं तथा यह उसकी लॉजिक स्थिति की बदलाव में परिणमित होते हैं। एक गेट या किसी अन्य लॉजिक डिवाइज़, जो अधिकतम शोर वोल्टेज स्तर का सहन कर सकता है, उस स्तर को उसकी “शोर सीमा” कहते हैं।

### 8.2.5 पावर का अपव्यय:

प्रत्येक गेट, एक पावर सप्लाई V<sub>CC</sub> (CMOS के मामले में V<sub>DD</sub>) से जुड़ा है। यह अपने ऑपरेशन के दौरान, एक निश्चित राशि की धारा लेती है। प्रत्येक गेट उच्च, संक्रमण या निम्न अवस्था में हो सकता है, इसलिए पावर सप्लाई से तीन अलग-अलग धाराएं लेती है।

- ICCH - उच्च अवस्था में लेने वाली धारा
- ICCT - उच्च से निम्न या निम्न से उच्च अवस्था में संक्रमण के समय लेने वाली धारा
- ICCL - निम्न अवस्था में लेने वाली धारा

TTL लॉजिक के लिए, ICCH तथा ICCL की तुलना में, संक्रमण के दौरान की धारा ICCT नगण्य है। अगर हम ICCH तथा ICCL को समान मान लेते हैं तब:

$$\text{औसत पावर अपव्यय} = V_{ic} \times (ICCH + ICCL)/2$$

CMOS लॉजिक के लिए, ICCT की तुलना में ICCH तथा ICCL नगण्य है। इसकी औसत पावर अपव्यय की गणना निम्न रूप से करते हैं।

$$\text{औसत पावर अपव्यय} = V_{CC} \times ICCT$$

ऊपर के दो समीकरणों से यह स्पष्ट है कि, TTL लॉजिक में पावर का अपव्यय ऑपरेशन की आवृत्ति पर निर्भर नहीं करती है। जबकि CMOS में यह ऑपरेशन, आवृत्ति पर निर्भर करता है। पावर का अपव्यय, चिप या प्रणाली द्वारा उत्पन्न गर्मी के अनुपात में होते हैं। अत्यधिक पावर का अपव्यय तापमान बढ़ाता है, जिससे लॉजिक सर्किट अपने सामान्य ऑपरेटिंग सीमा से बाहर जाता है और लॉजिक सर्किट खराब हो जाता है। इसलिए, किसी भी लॉजिक डिवाइज़ के पावर अपव्यय, जितना संभव हो उतना कम रखना चाहिए। पावर अपव्यय को स्थिर अपव्यय और गतिशील अपव्यय में वर्गीकृत किया जा सकता है।

- **स्थिर पावर अपव्यय (PS):** यह आउटपुट या इनपुट में बदलाव न होने के समय में जो उपभुक्त पावर है।
- **गतिशील पावर अपव्यय (PD):** यह, वह उपभुक्त पावर है, जो इनपुट या आउटपुट में संक्रमण के दौरान हो रहा हो।

ऐसा कहा जा सकता है कि, PD एक लॉजिक सर्किट के काम के दौरान खपत वास्तविक पावर है।

इस प्रकार:

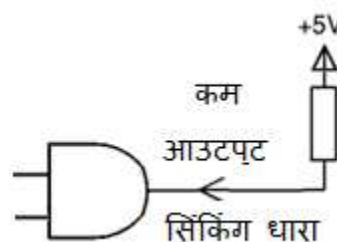
$$\text{कुल पावर अपव्यय} = \text{स्थिर पावर अपव्यय} + \text{गतिशील पावर अपव्यय}$$

$$PT = PS + PD$$

### 8.2.6 सिंकिंग और सोर्सिंग धाराएं:

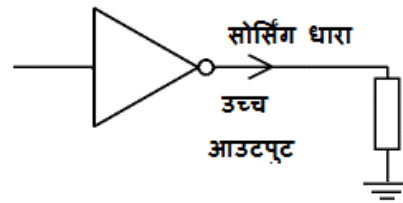
आउटपुट में लॉजिक स्तर के आधार पर डिजिटल IC के आउटपुट को, अक्सर, **सिंक** या **सोर्स** धारा कहा जाता है। ये दोनों धाराएं विपरीत दिशा में होती हैं। इन धाराओं के मूल्य, उपकरणों की ड्राइविंग क्षमता निर्णय करने में महत्वपूर्ण है।

- क) सिंकिंग धारा:** यह, आउटपुट के निम्न अवस्था के दौरान स्वीकार्य अधिकतम धारा है, जो आउटपुट टर्मिनल की तरफ बहती है। एक IC धारा स्वीकारते हैं (sinking), मतलब धारा उसकी आउटपुट टर्मिनल की तरफ बहती है। यह आउटपुट के निम्न होने की अवस्था में होता है।



### 8.2 गेट आउटपुट की तरफ सिंकिंग धारा

- ख) सोर्सिंग धारा:** यह, वह धारा है, जो आउटपुट की उच्च अवस्था में, एक लॉजिक डिवाइज़ के द्वारा आपूर्ति किया जाता है। एक IC, धारा की आपूर्ति कर रहा है मतलब उसके आउटपुट टर्मिनल से धारा बाहर की ओर बह रही है।



चित्र 8.3 गेट के आउटपुट से सोर्सिंग धारा

क) TTL IC के लिए अधिकतम सिंकिंग तथा सोर्सिंग धाराएं इस तरह हैं:

सिंक धारा (एक आउटपुट के लिए) : 16mA

सोर्स धारा (एक आउटपुट के लिए) : 2mA

ख) लेकिन CMOS डिवाइजों के लिए सिंकिंग तथा सोर्सिंग धाराएं समान होती हैं और यह लगभग 1μA जितना है।

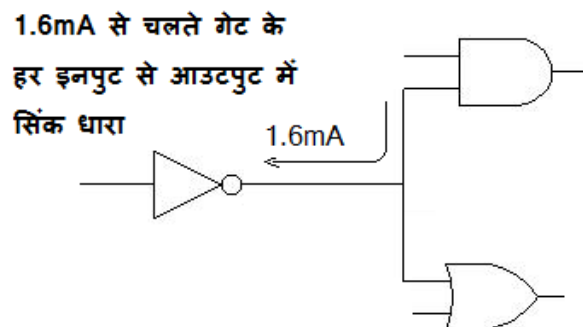
मान लें कि, एक TTL गेट का आउटपुट दूसरा TTL गेट के इनपुट से जुड़े है। TTL डिवाइजों में एक एकल इनपुट लीड से तथा लीड की ओर सिंक और सोर्स धाराएं निम्न प्रकार हैं।

सिंक धारा : 1.6mA (इनपुट/डिवाइज पर)

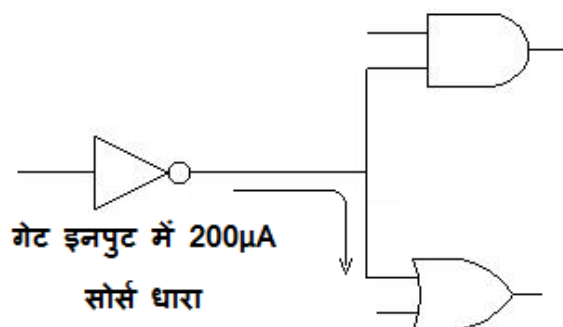
सोर्स धारा : 200μA (इनपुट/डिवाइज पर)

यह मूल्यों से, हर एक TTL डिवाइज के फैन आउट की गणना की जा सकती हैं। मतलब एक TTL आउटपुट दोनों अवस्था में अधिकतम कितने इनपुट को चला सकते हैं।

**फैन आउट** = आउटपुट का अधिकतम धारा / एक इनपुट की धारा



चित्र 8.4. TTL परिवार में, एक एकल इनपुट के असर से गेट आउटपुट टर्मिनल की ओर सिंक धारा



चित्र 8.5. एकल TTL इनपुट की ओर सोर्स धारा

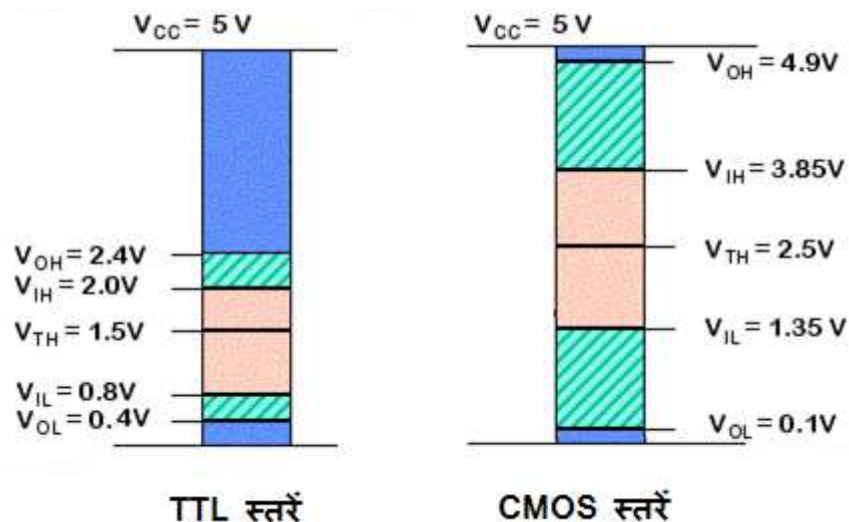
### 8.2.7 लॉजिक स्तरें:

लॉजिक सर्किट के, लॉजिक उच्च या लॉजिक निम्न अवस्थाओं के लिए आवश्यक वोल्टेज स्तर को, लॉजिक स्तर कहा जाता है। उदाहरण के लिए, TTL तथा CMOS (5V,  $V_{DD}$  पर) सर्किट में लॉजिक अवस्था के लिए वोल्टेज की सीमा नीचे तालिका 8.2 में प्रदान की गई हैं।

डिवाइज़ के प्रकार	लॉजिक अवस्था	इनपुट वोल्टेज	आउटपुट वोल्टेज
TTL	High	2.0 V to 5 V.	2.4 V to 5 V
	Low	0 V to 0.8 V	0 v to 0.4 V
CMOS	High	$\frac{2}{3} V_{DD}$ to $V_{DD}$	4.9 to $V_{DD}$
	Low	0 V to $\frac{1}{3} V_{DD}$	0 V to 0.1 V

तालिका 8.2 TTL तथा CMOS के लॉजिक वोल्टेज

इसी को चित्र 8.6 में रेखांकन में प्रतिनिधित्व किया गया है।



चित्र 8.6. TTL तथा CMOS डिवाइज़ों के लॉजिक वोल्टेज स्तरें दिखाने वाले चार्ट

### 8.2.8 लॉजिक स्तर से संबंधित पद:

- $V_{OHmin}$ : यह उच्च अवस्था (लॉजिक 1) में, न्यूनतम आउटपुट वोल्टेज है। TTL के लिए यह 2.4V तथा CMOS के लिए 4.9V होते हैं।
- $V_{OLmax}$ : यह निम्न अवस्था (लॉजिक 0) में अधिकतम आउटपुट वोल्टेज है। TTL के लिए यह 0.4V तथा CMOS के लिए 0.1V होते हैं।
- $V_{IHmin}$ : यह लॉजिक 1 के लिए मान्य, न्यूनतम इनपुट वोल्टेज है। TTL के लिए यह 2V तथा CMOS के लिए 3.5V होते हैं।
- $V_{ILmax}$ : यह लॉजिक 0 के लिए मान्य, न्यूनतम इनपुट वोल्टेज है। TTL के लिए यह 0.8V तथा CMOS के लिए 1.5V होते हैं।
- $V_T$  (थ्रेशोल्ड वोल्टेज): एक संक्रमण चलित डिवाइज़ को स्विचिंग कराने के लिए जो वोल्टेज की ज़रूरत पड़ती है, उसे थ्रेशोल्ड वोल्टेज कहा जाता है।

**वस्तुनिष्ठ:**

1. दो प्रचलित लॉजिक परिवारों के नाम बताएं। \_\_\_\_\_, \_\_\_\_\_
2. HTL \_\_\_\_\_ परिस्थितियों में उपयोग में लेते हैं।
3. TTL डिवाइजों की डिज़इन में \_\_\_\_\_ ट्रांसिस्टर्स का उपयोग करते हैं।
4. TTL डिवाइजों के अधिकतम सिंक धारा \_\_\_\_\_ है।
5. \_\_\_\_\_ लॉजिक डिवाइज़ कम धारा की खपत करती है।

**विषयनिष्ठ:**

1. विविध प्रकार के लॉजिक परिवारों के बारे में बताएं।
2. TTL लॉजिक परिवारों की विशेषताएं समझाएं।
3. लॉजिक डिवाइजों की विशेषताएं लिखें।
4. CMOS लॉजिक परिवारों के फायदाओं का उल्लेख करें।
5. TTL तथा CMOS में लॉजिक स्तरों का वोल्टेज स्तर लिखें।



## अध्याय 9

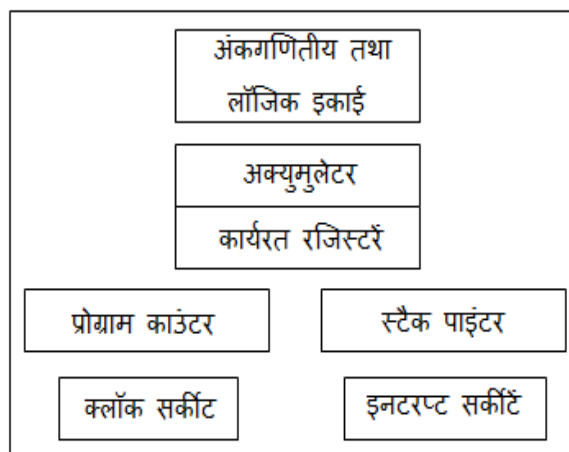
### माइक्रोकंट्रोलर - 8051

#### 9.0 माइक्रोकंट्रोलर क्या है?

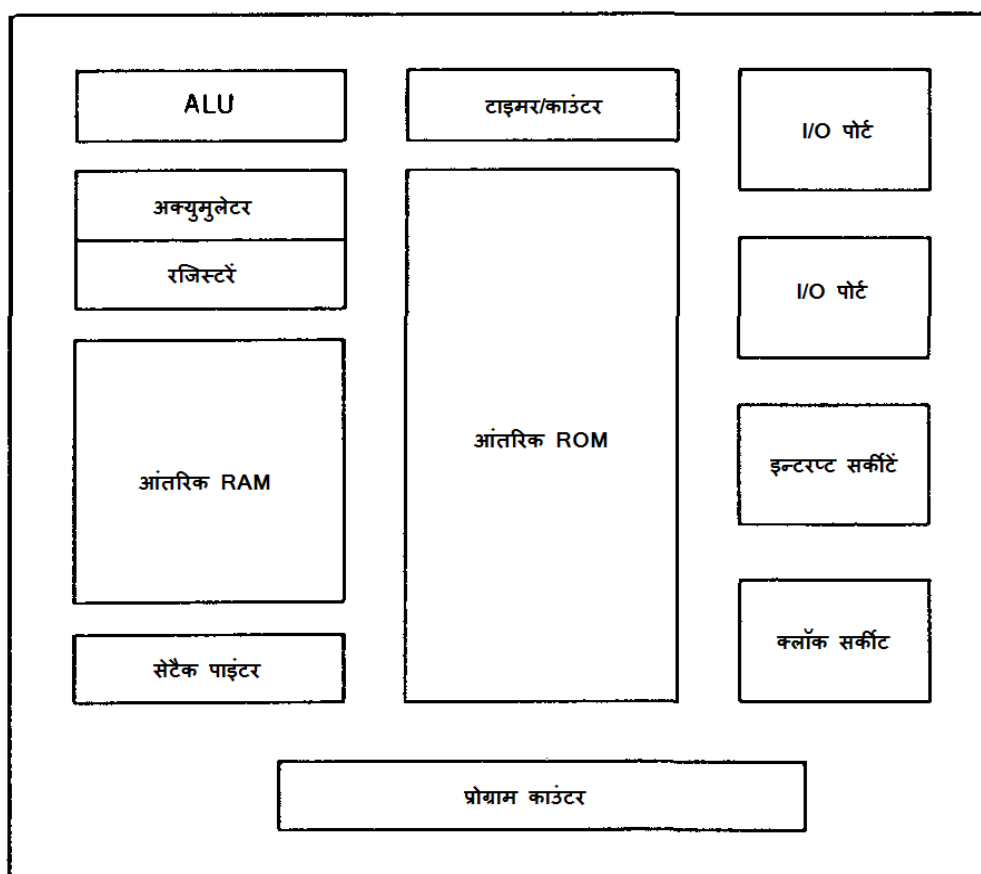
माइक्रोप्रोसेसर के बारे में अध्ययन करने के बाद माइक्रोकंट्रोलर के बारे में समझना बहुत आसान है। हमें याद करना होगा कि किसी भी प्रोग्रामयोग्य प्रणाली में माइक्रोप्रोसेसर CPU की तरह कार्य करता है। इसे पूरी तरह से काम करने वाली प्रणाली की तरह विकसित करने के लिए अन्य परिधीय उपकरण जैसे - मेमोरी (RAM तथा ROM), पैरेलल I/O इन्टरफेस, टाइमर, सीरियल संचार नियंत्रकों आदि बहुत ज़रूरी हैं। इसके अलावा, एक काम करने वाली प्रणाली बनाने के लिए इन सभी उपकरणों को CPU बस के साथ जोड़ते हैं। इसके लिए, अच्छा डिज़ाइन कौशल, बहुत प्रयास तथा सावधानी की ज़रूरत है। जब माइक्रोकंट्रोलर का प्रयोग किया जाता है, तब इनमें से अधिकांश समस्याएँ समाप्त हो जाती हैं।

एक माइक्रोकंट्रोलर, सिंगल IC में कंप्यूटर या एक पूर्ण प्रोग्रामयोग्य प्रणाली की तरह है। एक तैयार प्रोग्रामयोग्य प्रणाली हमें देने के लिए इसमें एक एकल अर्धचालक चिप में CPU के अलावा सभी आवश्यक परिधीय उपकरणों, सभी आंतर जोड़ों के साथ होते हैं। न्यूनतम बाह्य उपकरणों या घटकों के साथ हमें कोई भी उपयोग के लिए एक चालू प्रणाली इससे मिल जाती है। आम तौर पर एक माइक्रोकंट्रोलर, तय प्रोग्राम के साथ समर्पित अनुप्रयोगों के लिए उपयोग करते हैं। उदाहरण के लिए, चिप में लगे ROM में संग्रहित निश्चित प्रोग्राम का उपयोग एक मशीन के संचालन को नियंत्रण करने के लिए करते हैं। इसका कुछ अन्य अनुप्रयोग, सिक्का से चलने वाला टेलिफोन, वॉशिंग मशीन, माइक्रोवेव ओवन, आदि हैं।

नीचे चित्र 9.1 और 9.2 में क्रमशः एक लाक्षणिक माइक्रोप्रोसेसर और माइक्रोकंट्रोलर के ब्लॉक आरेख दिखाया गया है। इसकी रचना में एक माइक्रोप्रोसेसर CPU में शामिल सभी सुविधाएं, जैसे ALU, PC, SP तथा रजिस्ट्रों को सम्मिलित किया गया है। एक पूर्ण प्रोग्रामयोग्य प्रणाली बनाने के लिए अन्य आवश्यक सुविधाएं जैसे ROM, RAM, समांतर I/O, सीरियल I/O, काउंटर और क्लॉक सर्किट आदि भी इसमें जोड़े गए हैं।



चित्र 9.1. माइक्रोप्रोसेसर



चित्र 9.2. माइक्रोकंट्रोलर

**9.1 माइक्रोकंट्रोलर की विशेष सुविधाएं:** माइक्रोप्रोसेसर की तुलना में माइक्रोकंट्रोलर में निम्नलिखित विशेष सुविधाएं हैं।

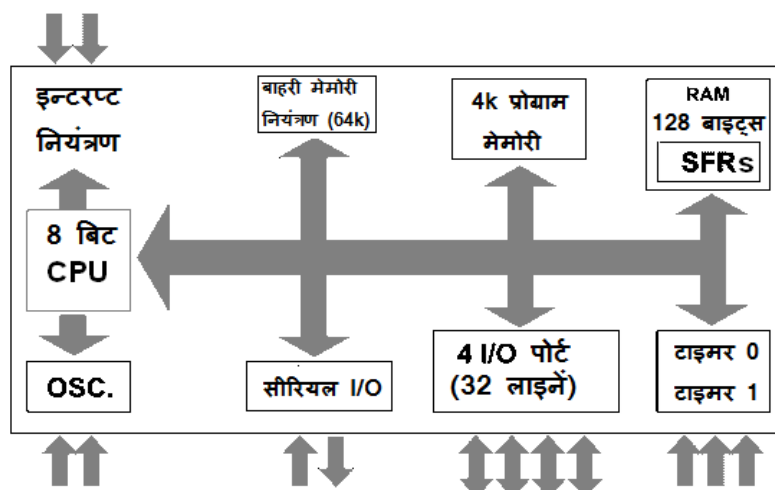
- क) माइक्रोकंट्रोलर के अंदर का कार्यात्मक इकाइयाँ, जैसे A, B, SP, T1, P2 आदि, नाम के अलावा एड्रेस से भी पहचाने जाते हैं।
- ख) परिधीय उपकरणों में बहुत संख्या में SFR होते हैं, जो नियंत्रण रजिस्ट्रों की तरह कार्य करते हैं।
- ग) रजिस्ट्रों का, व्यक्तिगत बिट संबोधन भी संभव है।
- घ) ऑन-चिप ROM या EPROM माइक्रोकंट्रोलर को एम्बेडेड सिस्टम अनुप्रयोगों के लिए एक उत्तम साधन बनाते हैं।

**9.2 माइक्रोप्रोसेसर और माइक्रोकंट्रोलर के कार्यों में महत्वपूर्ण अंतर निम्न प्रकार हैं।**

- माइक्रोप्रोसेसर में बाहरी एड्रेस से चिप की ओर कोड या डाटा की तीव्र चाल होती है, जबकि माइक्रोकंट्रोलर में चिप के भीतर ही बिटों की तीव्र चाल होती है।
- यह एक सामान्य प्रयोजन डिवाइज है।
- माइक्रोप्रोसेसर को चलाने के लिए कई अतिरिक्त भागों की ज़रूरत होती है लेकिन, माइक्रोकंट्रोलर किसी अतिरिक्त बाहरी डिजिटल भागों के बिना ही एक कंप्यूटर की तरह कार्य कर सकता है।
- यह एक एकल प्रयोजन डिवाइज है।

### 9.3 माइक्रोकंट्रोलर के intel 8051 परिवार:

Intel 8051, एक बहुत ही लोकप्रिय माइक्रोकंट्रोलर है, जो बहुत अनुप्रयोगों में बड़े पैमाने पर उपयोग की जाती हैं। इसकी बड़ी सफलता के कारण, बहुत अन्य निर्माताओं ने भी 8051 मूल डिज़ाइन पर आधारित माइक्रोकंट्रोलर के अपने संस्करण का उत्पादन किया है। निम्न चित्र 9.3 में, 8051 के मूल डिज़ाइन के मुख्य ब्लॉक आरेख दिखाया गया है। 8051 डिज़ाइन के आधार पर बाज़ार में उपलब्ध माइक्रोकंट्रोलर की सूची तालिका 9.1 में दिया गया है।



चित्र 9.3. 8051 के अंदर के साधन

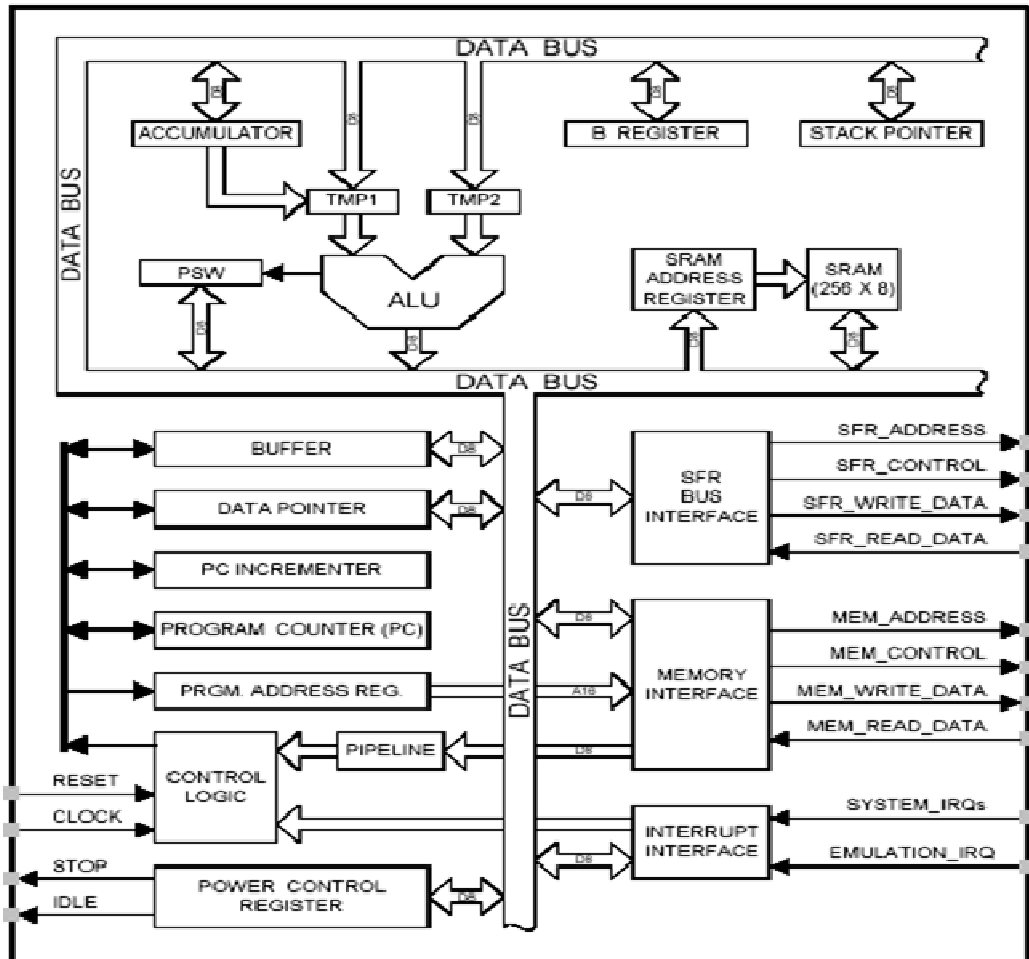
**9.4 अलग-अलग माइक्रोकंट्रोलर के मेक:** तालिका 9.1, विभिन्न निर्माताओं के 8051 परिवार का माइक्रोकंट्रोलर के सदस्यों की सूची दर्शाती है।

निर्माता / मोडल	पिन: I/O	काउंटर	RAM (बाइट्स)	ROM (बाइट्स)	अन्य सुविधाएं
Intel:8048	40:27	1	64	1K	बाह्य मेमोरी 8K
Intel:8051	40:32	2	128	4K	बाह्य मेमोरी 128K तथा सीरियल पोर्ट
Microchip: PIC16C56	18/12	0	25	1K	25/20 mA, सिंक/सोर्स, WDT, स्वतः रीसेट, RC ऑसिलेटर, कम कीमत
National:COP820	28:24	1	64	1K	सीरियल बिट I/O
Motorola:6805	28:20	1	64	1K	
Motorola:68HC11	52:40	2	256	8K	सीरियल पोर्ट; A/D; वॉचडॉग टाइमर(WDT)
Rockwell:6500/1	40:32	1	64	2K	
Philips:87C552	68:48	3	256	8K	सीरियल पोर्ट: A/D; WDT
TI:TMS 7500	40:32	1	128	2K	64K तक के बाहरी मेमोरी
TI: TMS370C050	68:55	2	256	4K	112K तक के बाहरी मेमोरी; A/D; सीरियल पोर्ट; WDT
Zilog:Z8	40:32	2	128	2K	124K तक के बाहरी मेमोरी, सीरियल पोर्ट
ZilogZ86C83	28:22	2	256	4K	8 चैनल A/D; बहुत कम कीमत

तालिका 9.1

### 9.5 8051 की आर्किटेक्चर:

Intel 8051, एक 8-बिट डिवाइज़ हैं। इसमें, सामान्य भाग संख्या में 8031 से 8751 तक के माइक्रोकंट्रोलर के पूरे परिवार शामिल हैं तथा यह N चैनल मेटल ऑक्साइड सिलिकॉन (NMOS) और कॉम्प्लिमेन्टरी मेटल ऑक्साइड सिलिकॉन (CMOS) प्रकार के निर्माण पैकेजों में उपलब्ध हैं। 8051 के एक अध्यतन संस्करण, 8052 भी अपनी विविधता के साथ उपलब्ध है और इसमें से एक सदस्य को BASIC में प्रोग्राम भी किया जा सकता है। यह अध्याय में, हम एक 40 पिन DIP में बनाये गए सामान्य 8051 के बारे में सीखेंगे।



चित्र 9.4. 8051 के आर्किटेक्चर

8051 की आर्किटेक्चर में निम्न विशिष्ट उपकरणों एवं सुविधाएं शामिल हैं।

- एक अक्युमुलेटर(A) तथा B रजिस्ट्रों के साथ 8 बिट CPU.
- 16 बिट प्रोग्राम काउंटर(PC) तथा डाटा पाइंटर (DPTR).
- फ्लैग के साथ 8 बिट प्रोग्राम स्टेटस वर्ड (PSW).
- 8 बिट स्टैक पाइंटर (SP).
- आंतरिक ROM या EPROM. (8031 में 0 बाइट से 8051 में 4K बाइट).
- 128 बाइट के आंतरिक RAM.
- चार 8 बिट पोर्ट (P0-P3) कुल 32 इनपुट/आउटपुट लाइनें।
- दो 16 बिट टाइमर/काउंटर (T0 तथा T1)

- पूरी तरह इयूप्लेक्स सीरियल रिसीवर/ट्रांसमीटर: SBUF.
- नियंत्रण रजिस्टर: TCON, TMOD, SCON, PCON, IP तथा IE.
- दो बाहरी तथा तीन आंतरिक इन्टरप्ट सोर्सें।
- ऑसिलेटर तथा क्लॉक सर्किट।
- 8051 में 34 सामान्य प्रयोजन या कार्यकारी रजिस्टर हैं।
- इसमें से दो A तथा B रजिस्टर हैं।
- A रजिस्टर अथवा अक्युमुलेटर का उपयोग गणितीय और लॉजिक कार्यों के लिए करते हैं।
- इसका उपयोग 8051 और कोई बाहरी मेमोरी के बीच डाटा के स्थानांतरण के लिए करते हैं।
- B रजिस्टर, गुणन और विभाजन के संचालन के लिए A रजिस्टर के साथ उपयोग किया जाता है।
- B रजिस्टर दूसरे संकार्य को रखता है और परिणाम की हिस्सा को भी रखता है।
  - गुणन परिणाम के 8 ऊपरी बिट्स
  - विभाजन के संदर्भ में शेष।
- अन्य 32 रजिस्ट्रों को चार बैंकों में आंतरिक RAM के हिस्से के रूप में व्यवस्थापित की गई है: B0 से B3, 8 रजिस्टर।
- PSW में RS1 और RS0 बिटों की सेटिंग द्वारा बैंक चुना जाता है।
- प्रोग्राम काउंटर (PC) एक 16 बिट रजिस्टर है, जो 8051 को यह बताते हैं कि अमल करने के लिए अगले निर्देश मेमोरी में जहाँ पर है।
- हर अनुदेश बाइट को लेने के बाद PC स्वतः ही बढ़ती है।
- PC ही एक ऐसी रजिस्टर है, जिसका आंतरिक एड्रेस नहीं होता है।
- DPTR, DPH तथा DPL नाम के दो 8 बिट रजिस्ट्रों से बना एक 16 बिट रजिस्टर है और केवल यहीं उपयोगकर्ता अभिगम्य रजिस्टर है।
- क्लॉक पल्स उत्पन्न करने वाला ऑसिलेटर (1MHz से 16MHz) को 8051 के हृदय माना जाता है।
- क्रिस्टल आवृत्ति, माइक्रोकंट्रोलर के मूल क्लॉक आवृत्ति होती है।
- फ्लैग, एक बिट रजिस्टर है, जो निश्चित प्रोग्राम अनुदेशों के परिणाम को संग्रह करता है।
- PSW में चार फ्लैगों को शामिल किया है।
- प्रोग्राम स्टेटस वर्ड (PSW) 8 बिट का एक विशेष कार्य रजिस्टर है।
- 8051 में एक 8 बिट स्टैक पाइंडर (SP) है।
- स्टैक, डाटा की संग्रह करने और जल्दी पुनःप्राप्त करने के लिए इस्तेमाल आंतरिक RAM का एक क्षेत्र को दर्शाता है।
- स्टैक पाइंडर, स्टैक के शीर्ष नामक एक आंतरिक RAM एड्रेस को धारण करने के लिए इस्तेमाल करते हैं।
- 8051 के विशेष कार्य रजिस्ट्रों, डाटा संग्रह के लिए आंतरिक RAM के जैसे ही एड्रेस की जा सकती है।
- आंतरिक ROM 0000h से FFFFh तक के कोड एड्रेस स्थान लेता है।
- I/O पोर्ट 8051 को बाहरी दुनिया से जोड़ते हैं।
- प्रोग्रामर के सामान्य उपयोग के लिए T0 तथा T1 नाम के दो 16 बिट टाइमर/काउंटर प्रदान किये गए हैं।
- काउंटर्स को टाइमर लो (TL0 और TL1) तथा टाइमर हाई (TH0 और TH1) नामक दो 8 बिट रजिस्ट्रों में विभाजित किया गया है।

- काउंटर की सभी काम TMOD और TCON की बिट अवस्था के द्वारा नियंत्रित किया जाता है।
- दोनों टाइमरों के संचालन का कार्य को संशोधित तथा कॉन्फिगर करने के लिए टाइमर कंट्रोल (TCON) SFR का इस्तेमाल करते हैं।
- SBUF, सीरियल बफर SFR आंतरिक सीरियल पोर्ट के माध्यम से डाटा भेजने तथा प्राप्त करने के लिए प्रयोग किया गया है।
- दूसरे शब्दों में, SBUF, लिखने में आउटपुट पोर्ट की तरह और पढ़ने में इनपुट पोर्ट की तरह कार्य करता है।
- सीरियल पोर्ट के बॉड रेट को नियंत्रित करने के लिए सीरियल कंट्रोल (SCON) SFR का इस्तेमाल करते हैं।
- इन्टरप्ट एनएबल (IE) SFR का उपयोग विशिष्ट इन्टरप्ट को सक्षम या अक्षम करने के लिए करते हैं।
- IE के लोवर बिट्स विशिष्ट इन्टरप्ट को तथा उच्च बिट्स सभी इन्टरप्टों को सक्षम या अक्षम करने के लिए इस्तेमाल करते हैं।
- इन्टरप्ट प्राथमिकता (IP) SFR का उपयोग प्रत्येक इन्टरप्ट के अपने प्राथमिकता को उल्लिखित करने के लिए प्रयोग करते हैं।
- सीरियल इन्टरप्ट रूटीन को सर्वोच्च प्राथमिकता है, इसलिए जब सीरियल इन्टरप्ट कार्यरत है तब दूसरा कोई इन्टरप्ट कार्य नहीं कर सकेंगे।

## 9.6 8051 के पिन कॉन्फिगरेशन:

Port 1 Bit 0	1	P1.0	Vcc	40	+ 5V
Port 1 Bit 1	2	P1.1	(AD0)P0.0	39	Port 0 Bit 0 (Address/Data 0)
Port 1 Bit 2	3	P1.2	(AD1)P0.1	38	Port 0 Bit 1 (Address/Data 1)
Port 1 Bit 3	4	P1.3	(AD2)P0.2	37	Port 0 Bit 2 (Address/Data 2)
Port 1 Bit 4	5	P1.4	(AD3)P0.3	36	Port 0 Bit 3 (Address/Data 3)
Port 1 Bit 5	6	P1.5	(AD4)P0.4	35	Port 0 Bit 4 (Address/Data 4)
Port 1 Bit 6	7	P1.6	(AD5)P0.5	34	Port 0 Bit 5 (Address/Data 5)
Port 1 Bit 7	8	P1.7	(AD6)P0.6	33	Port 0 Bit 6 (Address/Data 6)
Reset Input	9	RST	(AD7)P0.7	32	Port 0 Bit 7 (Address/Data 7)
Port 3 Bit 0 (Receive Data)	10	P3.0(RXD)	(Vpp)/EA	31	External Enable (EPROM Programming Voltage)
Port 3 Bit 1 (XMIT Data)	11	P3.1(TXD)	(PROG)ALE	30	Address Latch Enable (EPROM Program Pulse)
Port 3 Bit 2 (Interrupt 0)	12	P3.2( $\overline{\text{INT0}}$ )	$\overline{\text{PSEN}}$	29	Program Store Enable
Port 3 Bit 3 (Interrupt 1)	13	P3.3( $\overline{\text{INT1}}$ )	(A15)P2.7	28	Port 2 Bit 7 (Address 15)
Port 3 Bit 4 (Timer 0 Input)	14	P3.4(T0)	(A14)P2.6	27	Port 2 Bit 6 (Address 14)
Port 3 Bit 5 (Timer 1 Input)	15	P3.5(T1)	(A13)P2.5	26	Port 2 Bit 5 (Address 13)
Port 3 Bit 6 (Write Strobe)	16	P3.6( $\overline{\text{WR}}$ )	(A12)P2.4	25	Port 2 Bit 4 (Address 12)
Port 3 Bit 7 (Read Strobe)	17	P3.7( $\overline{\text{RD}}$ )	(A11)P2.3	24	Port 2 Bit 3 (Address 11)
Crystal Input 2	18	XTAL2	(A10)P2.2	23	Port 2 Bit 2 (Address 10)
Crystal Input 1	19	XTAL1	(A9)P2.1	22	Port 2 Bit 1 (Address 9)
Ground	20	Vss	(A8)P2.0	21	Port 2 Bit 0 (Address 8)

## 9.7 8051 के रजिस्ट्रों की सूची:

विशेष कार्य रजिस्टर	
A	अक्युमुलेटर
B	B रजिस्टर
PSW	प्रोग्राम स्टेटस वर्ड
P0	पोर्ट 0
P1	पोर्ट1
P2	पोर्ट2
P3	पोर्ट3
IP	इन्टरप्ट प्राथमिकता नियंत्रण
IE	इन्टरप्ट एनएबल नियंत्रण
TCON	टाइमर/काउंटरनियंत्रण
अन्य रजिस्टर	
SP	स्टैक पाइंटर
DPTR, DPL, DPH	डाटा पाइंटर
TMOD	टाइमर मोड
TH0	टाइमर/काउंटर0 MSB
TL0	टाइमर/काउंटर0 LSB
TH1	टाइमर/काउंटर1 MSB
TL1	टाइमर/काउंटर1 LSB

## 9.8 8051 में विविध SFR के स्थान:

80	P0	SP	DPL	DPH				PCON	87
88	ICON	TMOD	TL0	TL1	TH0	TH1			8F
90	P1								97
98	SCON	SBUF							9F
A0	P2								A7
A8	IE								AF
B0	P3								B7
B8	IP								B9
C0									C7
C8									CF
D0	PSW								D7
D8									DF
E0	ACC								E7
E8									EF
F0	B								F7
F8									FF

 Blue background are I/O port SFRs  
 Yellow background are control SFRs  
 Green background are other SFRs

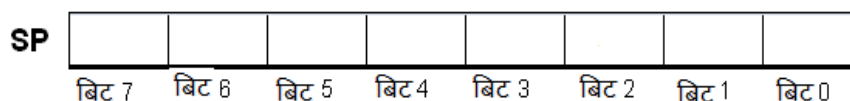
### 9.9 8051 के SFR के विवरण:

- क) **P0 (पोर्ट 0, एड्रेस 80h, बिट एड्रेस योग्य):** P0 पोर्ट दो कार्यों से विशेष हैं। जब बाहरी मेमोरी का उपयोग होता है, तब लोवर एड्रेस बाइट (A0-A7) इस पर लागू किया जाता है। अन्यथा, इस पोर्ट की सभी बिट इनपुट/आउटपुट के लिए कॉन्फिगर रहते हैं।
- ख) **P1 (पोर्ट 1, एड्रेस 90h, बिट एड्रेस योग्य):** यह इनपुट/आउटपुट के पोर्ट 1 है। इस SFR का प्रत्येक बिट माइक्रोकंट्रोलर के एक पिन से मेल खाता है। उदाहरण के तौर पर, पोर्ट 0 का बिट 0, पिन P.0.0, और बिट 7, P.0.7 होता है। इस SFR के एक बिट में मूल्य 1 लिखने से इसी I/O पिन पर एक उच्च स्तर भेजता है, जबकि मूल्य 0 लिखने से इसको निम्न स्तर में लाते हैं।
- ग) **P2 (पोर्ट 2, एड्रेस A0h, बिट एड्रेस योग्य):** जब बाहरी मेमोरी का प्रयोग किया जाता है, पोर्ट 2, P0 की तरह ही कार्य करता है। इस पोर्ट के पिन, बाहरी मेमोरी चिप के एड्रेसों के लिए अभिप्रेत है। अब की बार यह उच्च एड्रेस बाइट A8 से A15 को लागू किए जाते हैं। जब कोई मेमोरी नहीं जोड़ा हो, तब इस पोर्ट को सामान्य इनपुट/आउटपुट पोर्ट की तरह प्रयोग किए जा सकते हैं, जैसे P1 पोर्ट में होता है।
- घ) **P3 (पोर्ट 3, एड्रेस B0h, बिट एड्रेस योग्य):** इस पोर्ट के सभी पिन सामान्य इनपुट/आउटपुट की तरह प्रयोग किए जा सकते हैं, लेकिन इसके एक वैकल्पिक कार्य भी होता है। यह वैकल्पिक कार्यों के उपयोग करने के लिए P3 रजिस्टर के उपयुक्त पिन पर एक लॉजिक 1 लागू किया जाना चाहिए। हार्डवेयर के संदर्भ में यह पोर्ट P0 के समान है, अंतर इतना ही है कि इसके पिन में एक आंतरिक पुल-अप प्रतिरोध होता है।

#### • P3 के वैकल्पिक कार्य:

पिन संख्या	वैकल्पिक उपयोग	SFR
P3.0 - RXD	सीरियल डाटा इनपुट	SBUF
P3.1 - TXD	सीरियल डाटा आउटपुट	SBUF
P3.2 - INT0	बाहरी इन्टरप्ट 0	TCON.1
P3.3 - INT1	बाहरी इन्टरप्ट 1	TCON.3
P3.4 - T0	बाहरी टाइमर 0 इनपुट	TMOD
P3.5 - T1	बाहरी टाइमर 1 इनपुट	TMOD
P3.6 - WR	बाहरी मेमोरी write पल्स	--
P3.7 - RD	बाहरी मेमोरी read पल्स	--

- ड) **SP (स्टैक पॉइंटर, एड्रेस 81h):** यह माइक्रोकंट्रोलर का स्टैक पॉइंटर है। यह SFR आंतरिक RAM के स्टैक पर से कहाँ से अगला मूल्य लेना है, वह दर्शाता है। यह SFR, सभी निर्देशों से संशोधित होता है, जो स्टैक को संशोधित करता है, जैसे PUSH, POP, LCALL, RET, RETI और जब भी माइक्रोकंट्रोलर द्वारा इन्टरप्ट को उक्साया (enabled) जाता है।





स्टैक पाइंडर में संग्रहित किया गया मूल्य पहला मुक्त स्टैक एड्रेस की ओर इशारा करता है और स्टैक उपलब्धता की इजाज़त देता है। स्टैक पुश, स्टैक पाइंडर की मूल्य को 1 बार बढ़ाता है। इसी तरह, स्टैक पॉप, मूल्य को 1 बार घटाता है। रीसेट या पावर-ऑन के समय इसमें स्टैक शुरुआत स्थान पर (RAM की एड्रेस मूल्य) लोड हो जाता है। यह रजिस्टर में कोई अन्य मूल्य लिखने पर, पूरा स्टैक नयी मेमोरी स्थान पर चला जाता है।

- च) DPL/DPH (डाटा पाइंडर लो/उच्च, एड्रेस 82h/83h):** DPL और DPH SFR, एक 16 बिट मूल्य का डाटा पाइंडर को प्रतिनिधित्व करने के लिए काम करते हैं। डाटा पाइंडर बाहरी RAM और कोड मेमोरी से जुड़े कुछ निर्देशों के परिचालन के लिए कार्य करता है। इसके 16 बिट मुख्य रूप से बाहरी मेमोरी के एड्रेस बताने के लिए उपयोग करते हैं। इसके अलावा, DPTR रजिस्टर, आम तौर पर, डाटा और मध्यवर्ती परिणाम के संग्रह में इस्तेमाल होता है। यह एक अनसाइन्ड 2 बाइट पूर्णांक मूल्य होने से यह 0000h से FFFFh तक के मूल्यों को प्रतिनिधित्व कर सकते हैं।
- छ) PCON (पावर नियंत्रण, एड्रेस 87h):** इस SFR, 8051 के पावर नियंत्रण मोड को नियंत्रित करने के लिए उपयोग किया जाता है। 8051 के कुछ ऑपरेशन मोड, इसको “स्लीप/निष्क्रिय” मोड में जाने की अनुमति देती हैं, जिसमें बहुत कम पावर की आवश्यकता है। ऑपरेशन के यह मोड, PCON द्वारा नियंत्रित होता है। PCON के बिट्स से एक 8051 के सीरियल पोर्ट को प्रभावी बॉड दर को दुगुना करने के लिए उपयोग किया जाता है।

PCON	SMOD				GF1	GF0	PD	IDL
	बिट 7	बिट 6	बिट 5	बिट 4	बिट 3	बिट 2	बिट 1	बिट 0

- जब SMOD बिट उच्च स्तर में है, तब सीरियल पोर्ट का बॉड दर दुगुना हो जाता है।
- GF1 एक सामान्य कार्य बिट है। (उपयोग योग्य)
- GF0 भी एक सामान्य कार्य बिट है। (उपयोग योग्य)
- PD बिट को सेट करने से माइक्रोकंट्रोलर पावर डाउन में चला जाता है।
- IDL बिट को सेट करने से माइक्रोकंट्रोलर निष्क्रिय मोड में चला जाता है।

- ज) TCON (टाइमर नियंत्रण, एड्रेस 88h, बिट एड्रेस योग्य):** टाइमर नियंत्रण SFR का उपयोग 8051 के दोनों टाइमर चलाने का प्रकार को संशोधित और कॉन्फिगर करने के लिए किया जाता है। TCON SFR में कुछ नॉन-टाइमर संबंधित बिट्स भी होते हैं। ये बिट्स का उपयोग बाहरी इन्टरप्ट को किस तरह विन्यस्थ करना है, यह कॉन्फिगर करने के लिए इस्तेमाल करते हैं और बाहरी इन्टरप्ट फ्लैग भी रहते हैं, जो बाहरी इन्टरप्ट होने से सेट हो जाता है।

TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
	बिट 7	बिट 6	बिट 5	बिट 4	बिट 3	बिट 2	बिट 1	बिट 0

**टाइमर:** टाइमर हमेशा ऊपर की तरफ गिनते हैं। इससे कोई फर्क नहीं पड़ता कि टाइमर को टाइमर, काउंटर अथवा बॉड जनरेटर के रूप में उपयोग करते हैं। टाइमर हमेशा माइक्रोकंट्रोलर द्वारा बढ़ता है।

**टाइमर SFR:** 8051 में दो टाइमर होते हैं, दोनों एक समान ही कार्य करते हैं। एक टाइमर 0 है और दूसरा टाइमर 1. दोनों टाइमर दो SFR (TCON तथा TMOD) को साझा करते हैं, जो टाइमर का नियंत्रण करता है। दोनों टाइमरों में दो-दो SFR हैं, जो पूरी तरह से अपना-अपना है। (TL0 और TL1 तथा TH0 और TH1)।

**झ) TMOD (टाइमर मोड, एड्रेस 89h):** टाइमर मोड SFR का उपयोग दोनों टाइमरों के संचालन के मोड कॉन्फिगर करने के लिए होते हैं। इस SFR के उपयोग से आप दोनों टाइमरों को 16-बिट टाइमर, 8-बिट ऑटो रीलोड टाइमर, 13-बिट टाइमर या दो अलग-अलग टाइमर में कॉन्फिगर कर सकते हैं।

USB				LSB			
GATE	C/T	M1	M0	GATE	C/T	M1	M0
टाइमर 1				टाइमर 0			

- टाइमर 0 और टाइमर 1 दोनों एक ही मोड रजिस्टर TMOD का उपयोग करते हैं।
- यह एक बिट रजिस्टर है।
- लोवर 4 बिट्स टाइमर 0 के लिए तथा ऊपरी 4 बिट्स टाइमर 1 के लिए होते हैं।
- यह बिट एड्रेस योग्य नहीं हैं।

**व्यक्तिगत बिट्स के कार्य:**

बिट	नाम	कार्य	टाइमर
7	Gate-1	जब यह बिट सेट होता है, INT1 (P3.3) के उच्च अवस्था में ही टाइमर चालू होगी।	1
6	C/T-1	जब यह बिट सेट होता है, टाइमर T1 (P3.5) के घटनाओं की गिनती करेगी।	1
5	T1 M1	टाइमर मोड बिट	1
4	T1M0	टाइमर मोड बिट	1
3	Gate-0	जब यह बिट सेट होता है, INT0 (P3.2) के उच्च अवस्था में ही टाइमर चालू होगी।	0
2	C/T-0	जब यह बिट सेट होता है, टाइमर T0 (P3.4) के घटनाओं का गिनती करेगी।	0
1	T0 M1	टाइमर मोड बिट	0
0	T0M0	टाइमर मोड बिट	0

**टाइमर मोड्स:**

Tx M1	Tx M0	टाइमर मोड	मोड के विवरण
0	0	0	13 बिट टाइमर
0	1	1	16 बिट टाइमर
1	0	2	8 बिट ऑटो रीलोड
1	1	3	स्प्लिट टाइमर

- 13 बिट टाइमर मोड (मोड 0):** टाइमर मोड 0 एक 13 बिट टाइमर है। नये विकास में सामान्यतया, 13 बिट टाइमर मोड का उपयोग नहीं किया जाता है। जब टाइमर यह मोड में होता है, TLx 0 से 31 तक गिनती है तथा जब TLx 31 से बढ़ती है, TLx रीसेट हो जाता है और THx बढ़ते हैं। इस प्रकार प्रभावी ढंग से दोनों टाइमर बाइट्स के 13 बिट्स के उपयोग

होते हैं, जैसे TLx के 0 से 4 बिट्स और THx के 0 से 4 बिट्स। इसका मतलब, संक्षेप में इस टाइमर केवल 8192 मूल्यों का ही धारण कर सकते हैं। यदि आप, एक 13 बिट टाइमर को 0 में सेट करते हैं तो, यह 8192 मशीन आवृत्ति के बाद 0 हो जाता है।

- ii. **16 बिट टाइमर मोड (मोड 1):** टाइमर मोड 1 एक 16 बिट टाइमर है। यह आम तौर पर इस्तेमाल होने वाला मोड है। यह 13 बिट टाइमर जैसे ही कार्य करते हैं सिवाय कि इसमें 16 बिट्स का उपयोग किया जाता है। TLx, 0 से 255 तक बढ़ते हैं, तब यह 0 में रीसेट हो जाता है और THx एक बार बढ़ता है। यह एक पूरा 16 बिट टाइमर होने से यह 65536 भिन्न मूल्यों का धारण कर सकता है। 16 बिट टाइमर को आप 0 में सेट करते हैं, तो वह 65536 मशीन आवृत्तियों के बाद वापस 0 हो जाता है।
- iii. **8 बिट ऑटो रीलोड मोड (मोड 2):** टाइमर मोड 2 एक 8 बिट ऑटो रीलोड मोड है। जब टाइमर, मोड 2 में है, THx “रीलोड मूल्य” रखती है और TLx खुद ही टाइमर होता है। इस प्रकार TLx गिनना शुरू करते हैं और जब TLx में गिनति 255 के पार पहुँचती है, तब 0 में रीसेट होने के बजाय (जैसे मोड 0 और मोड 1 के मामले में था) यह THx में संग्रह किया गया मूल्य में रीसेट हो जाता है। ऑटो रीलोड मोड बहुत सामान्यतः बॉड दर की स्थापना के लिए प्रयोग किया जाता है।
- iv. **स्प्लिट टाइमर मोड (मोड 3):** टाइमर मोड 3 एक स्प्लिट टाइमर मोड है। टाइमर 0 को जब मोड 3 में रखते हैं, यह अनिवार्य रूप से दो अलग-अलग 8 बिट टाइमर बन जाते हैं। यानि TL0 टाइमर 0 और TH0 टाइमर 1 हो जाता है। दोनों टाइमर 0 से 255 तक गिनते हैं और 0 में रीसेट हो जाते हैं। टाइमर 1 से जुड़े हुए सभी बिट्स अब TH0 से जुड़ जाते हैं। जब टाइमर 0 स्प्लिट मोड में है, असली टाइमर 1 (यानि TH1 और TL1) सामान्य रूप से मोड 0, 1 या 2 में रखा जा सकता है, हालांकि आप असली टाइमर 1 को चालू या बंद नहीं कर सकेंगे, क्योंकि सभी बिट्स अभी TH0 से जुड़ा होता है। इस मामले में असली टाइमर 1 प्रत्येक मशीन चक्र के लिए बढ़ जायेगा।
- ज) **TL0/TH0 (टाइमर 0 कम/उच्च, 8Ah/8Bh):** इन दोनों SFR मिलकर टाइमर 0 का प्रतिनिधित्व करते हैं। इसके सटीक व्यवहार TMOD SFR में टाइमर को किस प्रकार कॉन्फिगर किया है इस पर निर्भर करता है, हालांकि यह टाइमर हमेशा ऊपर की ओर गिनती करते हैं। विन्यास योग्य यह है कि कब और कैसे यह मूल्य को बढ़ाना है।
- ट) **TL1/TH1 (टाइमर 1 कम/उच्च, 8Ch/8Dh):** इन दोनों SFR मिलकर टाइमर 1 का प्रतिनिधित्व करते हैं। इसकी सटीक व्यवहार TMOD SFR में टाइमर को किस प्रकार कॉन्फिगर किया है इस पर निर्भर करता है, हालांकि यह टाइमर हमेशा ऊपर की ओर गिनती करते हैं। विन्यास योग्य यह है कि कब और कैसे यह मूल्य को बढ़ाना है।
- ठ) **SCON (सीरियल कंट्रोल, एड्रेस 98h, बिट एड्रेस योग्य):** सीरियल नियंत्रण SFR 8051 के आंतरिक सीरियल पोर्ट के व्यवहार के विन्यास करने के लिए इस्तेमाल किया जाता है। इस SFR सीरियल पोर्ट के बॉड दर का नियंत्रण करता है, यदि सीरियल पोर्ट डाटा प्राप्त करने के लिए सक्रिय हो और इसमें फ्लैग्स भी होते हैं जो एक बाइट सफलतापूर्वक भेजने या प्राप्त करने पर सेट हो जाते हैं।

SCON	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
	बिट 7	बिट 6	बिट 5	बिट 4	बिट 3	बिट 2	बिट 1	बिट 0

SM0	SM1	मोड	विवरण	बॉड दर
0	0	0	8 bit shift register	1/12 क्वार्ट्स आवृत्ति
0	1	1	8 bit UART	टाइमर 1 द्वारा निर्धारित
1	0	2	9 bit UART	1/32 क्वार्ट्स आवृत्ति
1	1	3	9 bit UART	टाइमर 1 द्वारा निर्धारित

- SM0 - सीरियल पोर्ट मोड 0, सीरियल पोर्ट मोड के चयन के लिए उपयोग किया जाता है।
- SM1 - सीरियल पोर्ट मोड बिट 1.
- SM2 - सीरियल पोर्ट मोड बिट 2, मल्टिप्रोसेसर संचार सक्षम बिट के नाम से भी जाना जाता है।
- REN - रिसिप्शन सक्षम बिट जब सेट होते हैं, तब सीरियल रिसिप्शन को सक्षम करते हैं। जब यह क्लियर होते हैं, तब सीरियल रिसिप्शन को निष्क्रिय करते हैं।
- TB8 - ट्रांसमिटर बिट 8. सभी रजिस्टर 8 बिट के होने से यह बिट मोड 2 और 3 में 9वा बिट का संचरण की समस्या का हल करती है। 9वा बिट में लॉजिक 1 को संचारित करने के लिए यह सेट किया जाता है।
- RB8 - रिसीवर बिट 8 अथवा मोड 2 या 3 में प्राप्त 9वा बिट। प्राप्त 9वा बिट लॉजिक 0 होने से हार्डवेयर द्वारा क्लियर होता है और 9वा बिट लॉजिक 1 होने से हार्डवेयर द्वारा सेट होता है।
- TI - ट्रांसमिट इन्टरप्ट फ्लैग, जो एक बाइट का अंतिम बिट को भेजने पर स्वतः ही सेट होता है। यह प्रोसेसर के लिए एक संकेत है, जो बताते हैं कि नया बाइट के संचरण के लिए लाइन उपलब्ध है। यह सॉफ्टवेयर के भीतर से क्लियर होना चाहिए।
- RI - रिसीव इन्टरप्ट फ्लैग, जो एक बाइट प्राप्त होने पर स्वतः ही सेट होता है। यह संकेत देता है कि बाइट प्राप्त की गई है और नया डाटा से प्रतिस्थापित होने से पहले इसको तुरंत पढ़ें। यह बिट भी सॉफ्टवेयर के भीतर से ही क्लियर होना चाहिए।

**ड) SBUF (सीरियल बफ़र, ऐड्रेस 99h):** यह SFR, आंतरिक सीरियल पोर्ट द्वारा डाटा को भेजने या प्राप्त करने के लिए प्रयोग किया जाता है। SBUF में लिखे जाने वाले कोई भी मूल्य को सीरीयल पोर्ट के TXD पिन के माध्यम से बाहर भेजा जाता है। इसी प्रकार, सीरियल पोर्ट के RXD पिन के माध्यम से 8051 में प्राप्त होनेवाला कोई भी मूल्य SBUF द्वारा उपयोगकर्ता के लिए भेजा जाता है।

सीरियल पोर्ट उसका उपयोग करने से पहले कॉन्फिगर की जानी चाहिए। दूसरे शब्दों में, एक सीरियल “वर्ड” में कितने बिट होने चाहिए, बॉड दर तथा सिन्क्रनाइजेशन क्लॉक स्रोत आदि निर्धारित करना आवश्यक है। यह पूरी प्रक्रिया SCON (सीरियल नियंत्रण) रजिस्टर के बिट्स के नियंत्रण में होते हैं।

**ढ) IE (इन्टरप्ट एनएबल, ऐड्रेस A8h):** इन्टरप्ट सक्षम SFR, विशिष्ट इन्टरप्टों को सक्षम या निष्क्रिय करने के लिए इस्तेमाल किए जाते हैं। यह SFR के निचले 7 बिट्स का उपयोग विशिष्ट इन्टरप्टों को सक्षम/निष्क्रिय करने के लिए होते हैं, जबकि उच्चतर बिट का उपयोग सभी इन्टरप्टों को सक्षम/निष्क्रिय करने के लिए होता है। इस प्रकार, अगर IE के उच्च बिट शून्य (0) है तो, जो कोई विशिष्ट इन्टरप्ट को निचले बिट द्वारा सक्षम किया या नहीं, सभी इन्टरप्ट निष्क्रिय हो जाते हैं।

IE	EA	--	ET2	ES	ET1	EX1	ET0	EX0
	बिट 7	बिट 6	बिट 5	बिट 4	बिट 3	बिट 2	बिट 1	बिट 0

- EA - वैश्विक इन्टरप्ट सक्षम/निष्क्रिय
  - 0 - सभी इन्टरप्ट अनुरोधों को निष्क्रिय करते हैं।
  - 1 - सभी विशिष्ट इन्टरप्ट अनुरोधों को सक्षम करते हैं।
- ES - सीरियल इन्टरप्ट को सक्षम/निष्क्रिय करते हैं।
  - 0 - UART सिस्टम कोई इन्टरप्ट उत्पन्न नहीं कर सकते।
  - 1 - UART सिस्टम इन्टरप्ट को सक्षम करते हैं।
- ET1 - यह बिट टाइमर 1 इन्टरप्ट को सक्षम/निष्क्रिय करते हैं।
  - 0 - टाइमर 1 कोई भी इन्टरप्ट उत्पन्न नहीं कर सकते।
  - 1 - टाइमर 1 इन्टरप्ट को सक्षम करते हैं।
- EX1 - यह बिट बाहरी 1 इन्टरप्ट को सक्षम/निष्क्रिय करते हैं।
  - 0 - INT0 पिन की लॉजिक अवस्था में फेरबदल कोई भी इन्टरप्ट उत्पन्न नहीं कर सकती है।
  - 1 - INT0 पिन की लॉजिक अवस्था में फेरबदल बाहरी इन्टरप्ट को सक्षम करती है।
- ET0 - यह बिट टाइमर 0 इन्टरप्ट को सक्षम/निष्क्रिय करते हैं।
  - 0 - टाइमर 0 कोई भी इन्टरप्ट उत्पन्न नहीं कर सकते।
  - 1 - टाइमर 0 इन्टरप्ट को सक्षम करते हैं।
- EX0 - यह बिट बाहरी 0 इन्टरप्ट को सक्षम/निष्क्रिय करते हैं।
  - 0 - INT1 पिन की लॉजिक अवस्था में फेरबदल कोई भी इन्टरप्ट उत्पन्न नहीं कर सकती है।
  - 1 - INT1 पिन की लॉजिक अवस्था में फेरबदल बाहरी इन्टरप्ट को सक्षम करती है।

ण) IP (इन्टरप्ट प्राथमिकता, एड्रेस B8h, बिट एड्रेस योग्य): इन्टरप्ट प्राथमिकता SFR का उपयोग प्रत्येक इन्टरप्ट के सापेक्ष प्राथमिकता निर्दिष्ट करने के लिए किया जाता है। 8051 में कोई भी इन्टरप्ट निम्न (0) प्राथमिकता या उच्च (1) प्राथमिकता का हो सकता है।

उदाहरण के लिए, अगर हम 8051 को इस तरह कॉन्फिगर करें कि सीरियल इन्टरप्ट के अलावा सभी इन्टरप्ट निम्न प्राथमिकता का हो तो, दूसरा कोई भी इन्टरप्ट का पालन हो रहा हो तब भी सीरियल इन्टरप्ट हमेशा सिस्टम को इन्टरप्ट कर सकते हैं। हालांकि जब एक सीरियल इन्टरप्ट का निष्पादन हो रहा हो दूसरा कोई इन्टरप्ट सीरियल इन्टरप्ट रूटीन को प्रभावित नहीं कर सकते हैं, क्योंकि सीरियल इन्टरप्ट को उच्च प्राथमिकता होती है।

IP	--	--	PT2	PS	PT1	PX1	PT0	PX0
	बिट 7	बिट 6	बिट 5	बिट 4	बिट 3	बिट 2	बिट 1	बिट 0

- PS - सीरियल पोर्ट इन्टरप्ट प्राथमिकता बिट  
प्राथमिकता 0 तथा प्राथमिकता 1
- PT1 - टाइमर 1 इन्टरप्ट प्राथमिकता  
प्राथमिकता 0 तथा प्राथमिकता 1
- PX1 - बाहरी इन्टरप्ट INT1 प्राथमिकता  
प्राथमिकता 0 तथा प्राथमिकता 1
- PT0 - टाइमर 0 इन्टरप्ट प्राथमिकता  
प्राथमिकता 0 तथा प्राथमिकता 1
- PX0 - बाहरी इन्टरप्ट INT0 प्राथमिकता  
प्राथमिकता 0 तथा प्राथमिकता 1

त) PSW (प्रोग्राम स्टेटस वर्ड, एड्रेस D0h, बिट एड्रेस योग्य): PSW SFR में कैरी (carry) फ्लैग, सहायक कैरी (carry) फ्लैग, ओवरफ्लो फ्लैग और समानता फ्लैग शामिल होते हैं। इसके अतिरिक्त PSW रजिस्टर में रजिस्टर बैंक चयन फ्लैग भी होते हैं जो "R" रजिस्टर बैंक में से एक का चयन करने के लिए होते हैं।

PSW	CY	AC	F0	RS1	RS0	OV	--	P
	बिट 7	बिट 6	बिट 5	बिट 4	बिट 3	बिट 2	बिट 1	बिट 0

- P - पैरिटी (समानता) बिट - अगर अक्युमुलेटर में संग्रहित एक संख्या सम संख्या है, तब यह स्वतः ही सेट (1) हो जाता है अन्यथा यह क्लियर (0) हो जाता है। यह मुख्यतः सीरियल संचार के माध्यम से डाटा का प्रेषण या प्रप्ति के दौरान उपयोग करते हैं।
- बिट 1 - यह बिट माइक्रोकंट्रोलर के भविष्य के संस्करणों में इस्तेमाल करने के इरादे से रखे हैं।
- OV (ओवरफ्लो) - यह तब होता है, जब कोई अंकगणितीय कार्य के परिणाम 255 से बढ़ जाते हैं और यह एक रजिस्टर में संग्रहित करना नामुमकिन है। ओवरफ्लो अवस्था के कारण OV बिट सेट (1) हो जाता है अन्यथा यह क्लियर (0) हो जाता है।
- F0 (फ्लैग 0) - यह एक सामान्य कार्य बिट है, जो उपयोगकर्ता के लिए उपलब्ध है।
- AC (सहायक कैरी (carry) फ्लैग) - इसका उपयोग BCD ऑपरेशनों के लिए ही होते हैं।
- CY (कैरी (carry) फ्लैग) - यह (9वा) सहायक बिट है, जो सभी अंकगणितीय ऑपरेशनों एवं शिफ्ट निर्देशों के लिए प्रयोग की जाती है।
- RS0 तथा RS1 (रजिस्टर बैंक चयन बिट्स) - यह दो बिट्स RAM के चार रजिस्टर बैंकों में से एक के चयन करने के लिए उपयोग करते हैं। ये बिट्स की सेटिंग और क्लियरिंग करने से RAM के चार में से एक बैंक में R0 -R7 रजिस्टर संग्रहित होता है।

RS1	RS0	RAM में जगह
0	0	बैंक 0 (00h - 07h)
0	1	बैंक 1 (08h - 0Fh)
1	0	बैंक 2 (10h - 1fh)
1	1	बैंक 3 (18h - 1Fh)

थ) A (अक्युमुलेटर, एड्रेस E0h, बिट एड्रेस योग्य): A रजिस्टर एक सामान्य उद्देश्य रजिस्टर है, जो ऑपरेशन के दौरान प्राप्त मध्यवर्ती परिणामों के संग्रह करने के लिए इस्तेमाल करते हैं। कोई भी संख्या या संकार्य के ऊपर दिए गए निर्देश का पालन होने से पहले इसको अक्युमुलेटर में संग्रह करना अनिवार्य है। ALU द्वारा किए गए अंकगणितीय ऑपरेशन से प्राप्त सभी परिणामों को अक्युमुलेटर में संग्रह किया जाता है। एक रजिस्टर में से दूसरे रजिस्टर में स्थानांतरित करने वाले डाटा अक्युमुलेटर से होते हुए ही स्थानांतरित होता है। दूसरे शब्दों में, A रजिस्टर सबसे अधिक इस्तेमाल होने वाला रजिस्टर है और इस रजिस्टर के बिना माइक्रोकंट्रोलर की कल्पना करना असंभव है। 8051 माइक्रोकंट्रोलर द्वारा प्रयोग करने वाला आधे से अधिक निर्देश किसी भी तरह अक्युमुलेटर का इस्तेमाल करते हैं।

द) B (B रजिस्टर, एड्रेस F0h, बिट एड्रेस योग्य): B रजिस्टर गुणन और विभाजन निर्देश के समय ही उपयोग में आते हैं।

**9.11 अनुदेश समुच्चय:** 8051 के अनुदेश समुच्चय में निम्न निर्देशों के 5 श्रेणियाँ शामिल हैं।

- क) अंकगणित अनुदेश
- ख) लॉजिक अनुदेश
- ग) डाटा स्थानांतरण अनुदेश
- घ) बुलियन परवर्तनीय मैनिपुलेशन अनुदेश
- ङ) प्रोग्राम तथा मशीन नियंत्रण अनुदेश

Mnemonic	Description	Byte	Cycle
<b>Arithmetic Operations</b>			
ADD A,Rn	Add register to accumulator	1	1
ADD A,direct	Add direct byte to accumulator	2	1
ADD A,@Ri	Add indirect RAM to accumulator	1	1
ADD A,#data	Add immediate data to accumulator	2	1
ADDC A,Rn	Add register to accumulator with carry flag	1	1
ADDC A,direct	Add direct byte to A with carry flag	2	1
ADDC A,@Ri	Add indirect RAM to A with carry flag	1	1
ADDC A,#data	Add immediate data to A with carry flag	2	1
SUBB A,Rn	Subtract register from A with borrow	1	1
SUBB A,direct	Subtract direct byte from A with borrow	2	1
SUBB A,@Ri	Subtract indirect RAM from A with borrow	1	1
SUBB A,#data	Subtract immediate data from A with borrow	2	1
INC A	Increment accumulator	1	1
INC Rn	Increment register	1	1
INC direct	Increment direct byte	2	1
INC @Ri	Increment indirect RAM	1	1
DEC A	Decrement accumulator	1	1
DEC Rn	Decrement register	1	1
DEC direct	Decrement direct byte	2	1
DEC @Ri	Decrement indirect RAM	1	1
INC DPTR	Increment data pointer	1	2
MUL AB	Multiply A and B	1	4
DIV AB	Divide A by B	1	4
DA A	Decimal adjust accumulator	1	1



### Logic Operations

ANL	A,Rn	AND register to accumulator	1	1
ANL	A,direct	AND direct byte to accumulator	2	1
ANL	A,@Ri	AND indirect RAM to accumulator	1	1
ANL	A,#data	AND immediate data to accumulator	2	1
ANL	direct,A	AND accumulator to direct byte	2	1
ANL	direct,#data	AND immediate data to direct byte	3	2
ORL	A,Rn	OR register to accumulator	1	1
ORL	A,direct	OR direct byte to accumulator	2	1
ORL	A,@Ri	OR indirect RAM to accumulator	1	1
ORL	A,#data	OR immediate data to accumulator	2	1
ORL	direct,A	OR accumulator to direct byte	2	1
ORL	direct,#data	OR immediate data to direct byte	3	2
XRL	A,Rn	Exclusive OR register to accumulator	1	1
XRL	A direct	Exclusive OR direct byte to accumulator	2	1
XRL	A,@Ri	Exclusive OR indirect RAM to accumulator	1	1
XRL	A,#data	Exclusive OR immediate data to accumulator	2	1
XRL	direct,A	Exclusive OR accumulator to direct byte	2	1
XRL	direct,#data	Exclusive OR immediate data to direct byte	3	2
CLR	A	Clear accumulator	1	1
CPL	A	Complement accumulator	1	1
RL	A	Rotate accumulator left	1	1
RLC	A	Rotate accumulator left through carry	1	1
RR	A	Rotate accumulator right	1	1
RRC	A	Rotate accumulator right through carry	1	1
SWAP	A	Swap nibbles within the accumulator	1	1



**Data Transfer**

MOV A,Rn	Move register to accumulator	1	1
MOV A,direct <sup>*)</sup>	Move direct byte to accumulator	2	1
MOV A,@Ri	Move indirect RAM to accumulator	1	1
MOV A,#data	Move immediate data to accumulator	2	1
MOV Rn,A	Move accumulator to register	1	1
MOV Rn,direct	Move direct byte to register	2	2
MOV Rn,#data	Move immediate data to register	2	1
MOV direct,A	Move accumulator to direct byte	2	1
MOV direct,Rn	Move register to direct byte	2	2
MOV direct,direct	Move direct byte to direct byte	3	2
MOV direct,@Ri	Move indirect RAM to direct byte	2	2
MOV direct,#data	Move immediate data to direct byte	3	2
MOV @Ri,A	Move accumulator to indirect RAM	1	1
MOV @Ri,direct	Move direct byte to indirect RAM	2	2
MOV @Ri,#data	Move immediate data to indirect RAM	2	1
MOV DPTR,#data16	Load data pointer with a 16-bit constant	3	2
MOVC A,@A + DPTR	Move code byte relative to DPTR to accumulator	1	2
MOVC A,@A + PC	Move code byte relative to PC to accumulator	1	2
MOVX A,@Ri	Move external RAM (8-bit addr.) to A	1	2
MOVX A,@DPTR	Move external RAM (16-bit addr.) to A	1	2
MOVX @Ri,A	Move A to external RAM (8-bit addr.)	1	2
MOVX @DPTR,A	Move A to external RAM (16-bit addr.)	1	2
PUSH direct	Push direct byte onto stack	2	2
POP direct	Pop direct byte from stack	2	2
XCH A,Rn	Exchange register with accumulator	1	1
XCH A,direct	Exchange direct byte with accumulator	2	1
XCH A,@Ri	Exchange indirect RAM with accumulator	1	1
XCHD A,@Ri	Exchange low-order nibble indir. RAM with A	1	1

**Boolean Variable Manipulation**

CLR C	Clear carry flag	1	1
CLR bit	Clear direct bit	2	1
SETB C	Set carry flag	1	1
SETB bit	Set direct bit	2	1
CPL C	Complement carry flag	1	1
CPL bit	Complement direct bit	2	1
ANL C,bit	AND direct bit to carry flag	2	2
ANL C,/bit	AND complement of direct bit to carry	2	2
ORL C,bit	OR direct bit to carry flag	2	2
ORL C,/bit	OR complement of direct bit to carry	2	2
MOV C,bit	Move direct bit to carry flag	2	1
MOV bit,C	Move carry flag to direct bit	2	2

**Program and Machine Control**

ACALL addr11	Absolute subroutine call	2	2
LCALL addr16	Long subroutine call	3	2
RET	Return from subroutine	1	2
RETI	Return from interrupt	1	2
AJMP addr11	Absolute jump	2	2
LJMP addr16	Long jump	3	2
SJMP rel	Short jump (relative addr.)	2	2
JMP @A + DPTR	Jump indirect relative to the DPTR	1	2
JZ rel	Jump if accumulator is zero	2	2
JNZ rel	Jump if accumulator is not zero	2	2
JC rel	Jump if carry flag is set	2	2
JNC rel	Jump if carry flag is not set	2	2
JB bit,rel	Jump if direct bit is set	3	2
JNB bit,rel	Jump if direct bit is not set	3	2
JBC bit,rel	Jump if direct bit is set and clear bit	3	2
CJNE A,direct,rel	Compare direct byte to A and jump if not equal	3	2
CJNE A,#data,rel	Compare immediate to A and jump if not equal	3	2
CJNE Rn,#data rel	Compare immed. to reg. and jump if not equal	3	2
CJNE @Ri,#data,rel	Compare immed. to ind. and jump if not equal	3	2
DJNZ Rn,rel	Decrement register and jump if not zero	2	2
DJNZ direct,rel	Decrement direct byte and jump if not zero	3	2
NOP	No operation	1	1

**डाटा प्रोसेसिंग मोड:**

जब सिस्टम चलता है तब प्रोसेसर, प्रोग्राम अनुदेशों के अनुसार डाटा संसाधित करता है। प्रत्येक अनुदेश के दो भाग होते हैं। एक भाग क्या करना है इसका विवरण देता है और दूसरा भाग कैसे करना है बताता है। दूसरा भाग डाटा (बाइनरी संख्या) या डाटा कहाँ संग्रहित है उसका एड्रेस हो सकते हैं। मेमोरी के कौनसा हिस्सा का एक्सेस करना होता है, इसके आधार पर सभी 8051 माइक्रोकंट्रोलर्स दो तरह के एड्रेसिंग का प्रयोग करते हैं। यह हैं:

1. **तुरंत एड्रेसिंग:** तुरंत एड्रेसिंग को यह नाम इसलिए मिला कि मेमोरी में जो मूल्य का संग्रह करना है वह मूल्य ऑप-कोड के तुरंत बाद लिखी जाती है, मतलब, मेमोरी में संग्रह करने वाले मूल्य अनुदेश में ही देते हैं।  
उदा: MOVA,#20h
2. **प्रत्यक्ष एड्रेसिंग:** इस मोड में, मेमोरी में संग्रहित किए जाने वाले मूल्य सीधे दूसरे मेमोरी स्थान से निकालते हैं।  
उदा: MOVA,30h  
यह अनुदेश, आंतरिक RAM के एड्रेस 30 में से डाटा लेकर सीधे अक्युमुलेटर में लोड करते हैं। यह एड्रेसिंग भी तेज़ है, क्योंकि डाटा 8051 के आंतरिक RAM से ही निकालते हैं।

3. **अप्रत्यक्ष एड्रेसिंग:** अप्रत्यक्ष एड्रेसिंग एक बहुत शक्तिशाली एड्रेसिंग मोड है, जो कई मामलों में आसाधारण स्तर का लचीलेपन प्रदान करता है।

उदा: `MOVA,@R0`

यह अनुदेश पर, 8051 के अक्युमुलेटर, आंतरिक RAM का मूल्य से लोड करेगा, जो एड्रेस R0 से संकेत करता है।

उदाहरण के लिए, मान लें कि R0 में मूल्य 40h रखे हैं तथा आंतरिक RAM एड्रेस 40h में 67h मूल्य रखे हैं। जब ऊपर के अनुदेश का निष्पादन होगा, अक्युमुलेटर में 67h मूल्य लोड हो जाएगा। अप्रत्यक्ष एड्रेसिंग हमेशा आंतरिक RAM की ओर सूचित करता है, लेकिन एक SFR की ओर नहीं करता है।

4. **बाहरी प्रत्यक्ष एड्रेसिंग:** बाहरी मेमोरी का एक्सेस एक अनुदेशों के समूह से होता है, जो बाहरी प्रत्यक्ष एड्रेसिंग का उपयोग किया जाता है। यह प्रत्यक्ष एड्रेसिंग जैसे ही प्रतीत होता है, लेकिन यह आंतरिक मेमोरी के बजाय बाहरी मेमोरी का एक्सेस करने के लिए उपयोग किया जाता है। बाहरी प्रत्यक्ष एड्रेसिंग मोड का उपयोग करने वाले दो अनुदेश होते हैं।

`MOVXA,@DPTR`

`MOVX@DPTR,A`

एक बार जब DPTR सही बाहरी मेमोरी एड्रेस रखता है, पहला आदेश बाहरी मेमोरी एड्रेस का अंतर्निहित मूल्य को अक्युमुलेटर में स्थानांतरित करता है और दूसरा आदेश इसके विपरीत स्थानांतरण करता है।

5. **बाहरी अप्रत्यक्ष एड्रेसिंग:** एड्रेसिंग के इस रूप सामान्यतया अपेक्षाकृत छोटी परियोजनाओं में ही उपयोग किए जाते हैं, जिसमें बाहरी RAM में बहुत कम जगह होती है। यह एड्रेसिंग मोड का एक उदाहरण है: `MOVX@R0,A`

R0 का मूल्य पहले पढ़ लेते हैं और अक्युमुलेटर की मूल्य को बाहरी RAM का उस एड्रेस में अंतरित होता है।

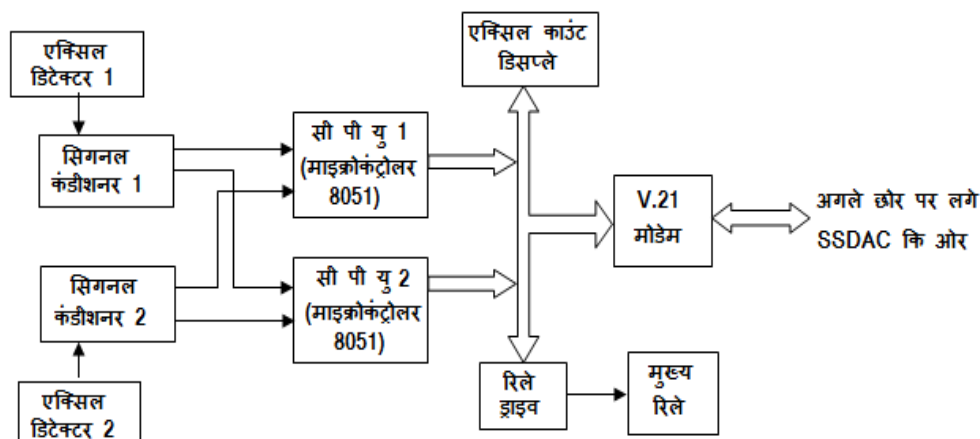
- 9.13 8051 माइक्रोकंट्रोलर के इन्टरफ़ेस:** 8051 के लिए पाँच इन्टरफ़ेस स्रोत होते हैं, मतलब माइक्रोकंट्रोलर 5 अलग-अलग घटनाओं को पहचान सकते हैं, जो नियमित प्रोग्राम का क्रियान्वयन को इन्टरफ़ेस कर सकता है।

- टाइमर 0 ओवरफ्लो - TF0
- टाइमर 1 ओवरफ्लो - TF1
- सीरियल अक्षर के रिसिप्शन/ट्रांसमिशन - RI/TI
- बाहरी घटना 0 - INT0
- बाहरी घटना 1 - INT1

IE रजिस्टर के बिट्स का सेटिंग करके प्रत्येक इन्टरफ़ेस को सक्षम या निष्क्रिय किए जा सकते हैं। इसी प्रकार, IE रजिस्टर का EA बिट को क्लियर करके पूरा इन्टरफ़ेस प्रणाली को निष्क्रिय भी किया जा सकता है।

## 9.14 8051 माइक्रोकंट्रोलर का प्रयोग:

### 9.14.1 एस एस डी ए सी (SSDAC):



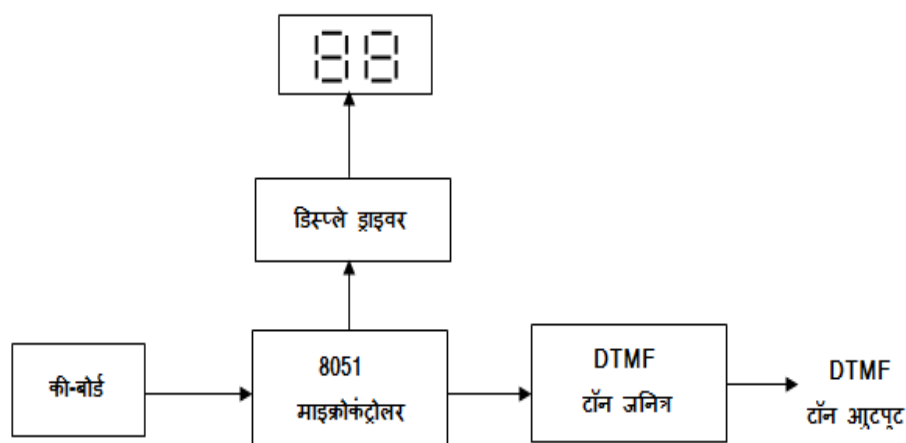
चित्र 9.6. SSDAC प्रणाली का ब्लॉक आरेख

माइक्रोकंट्रोलर ब्लॉक यह इकाई का केंद्र है। 2 में से 2 फैसलों को करने के लिए दो समान ब्लॉक

लागू की गई हैं। इन ब्लॉक मूल रूप से पल्स की गिनती, काउंट की पुष्टि, समीपवर्ती SSDAC इकाई के साथ संचार तथा आउटपुट देने के लिए निर्णय लेने से संबंधित सभी कार्य करते हैं। अगर परिणामों में मिलान नहीं होता है तो, प्रणाली सुरक्षित (fail-safe) ऑपरेशन में चला जाता है।

### 9.14. 2 DTMF कंट्रोल कार्यालय उपस्कर:

8051 माइक्रोकंट्रोलर का दूसरा मुख्य प्रयोग, ट्रेन ट्राफिक कंट्रोल संचार प्रणाली के कंट्रोल कार्यालय उपस्कर में की गई है। तुम्मला इलेक्ट्रॉनिक्स द्वारा शुरू में आपूर्ति की गई DTMF मुख्यालय उपस्कर में 8749 माइक्रोकंट्रोलर का उपयोग किया गया था, जो 8051 के पूर्व है। बाद में सभी नये आपूर्तियों में 8749 के जगह 8051 का उपयोग करने लगे। 8051 प्रयुक्त प्रणाली चित्र 9.7 में दिखाया गया है।



चित्र 9.7. DTMF मुख्यालय उपस्कर के ब्लॉक आरेख

**वस्तुनिष्ठ:**

1. दैनिक जीवन में 8051 के उपयोग करने वाले कुछ उपकरणों के नाम बताएं। \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_.
2. SSDAC प्रणाली में \_\_\_\_\_ 8051 का उपयोग किया गया है।
3. DTMF मुख्यालय उपस्कर के पहले अभिकल्प में \_\_\_\_\_ माइक्रोकंट्रोलर का उपयोग किया गया था।
4. DTMF मुख्यालय उपस्कर के दूसरे अभिकल्प में \_\_\_\_\_ माइक्रोकंट्रोलर का उपयोग किया गया था।
5. बाहरी दुनिया में दिखाई देने वाले 8051 के अन्य अनुप्रयोगों के नाम बताएं। \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_.

**विषयनिष्ठ:**

1. प्रत्येक के संक्षिप्त वर्णन करते हुए S&T विभाग में 8051 के कुछ और अनुप्रयोगों के बारे में उल्लेख करें।
2. सरल ब्लॉक आरेख के साथ डिजिटल एक्सिसल काउंटर में 8051 की भूमिका समझाएं।
3. ब्लॉक आरेख सहित 8051 का उपयोग करने वाले किसी भी DTMF नियंत्रण कार्यालय उपस्कर के बारे में समझाएं।