चैप्टर – 9
डिजिटल इलेक्ट्रानिक्स

डिजिटल सिग्नल की परिभाषाऐं :–

इलेक्ट्रॉनिक्स सर्किट और सिस्टम को सामान्यतः दो मुख्य कैटेगरी में विभाजित किया जा सकता है।

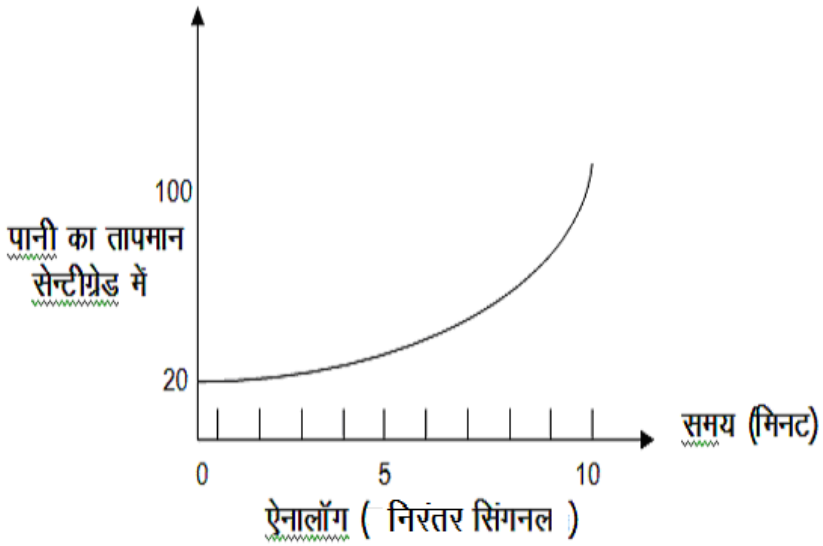
1. **ऐनालॉग** और 2. **डिजिटल**

ऐनालॉग सर्किटस का डिजाइन छोटे सिग्नल के उपयोग एंव रैखिक उपयोग के लिये किया जाता है, जबकि डिजिटल सर्किटस का उपयोग बड़े सिग्नलस और अरैखिक व्यवहार के लिये उपयोग किये जाते है।

कोई भी मात्रा जिसका परिवर्तन समय के सापेक्ष लगातार होता रहता है उसे ऐनालॉग सिग्नल कहते हैं।

उदाहरणार्थ – पानी से भरे कंटेनर को यदि स्टोव पर रख कर गर्म किया जाता है तो कंटेनर के पानी का तापमान परिवर्तित होता है। पानी के तापमान को एक समय अंतराल पर रिकार्ड किये जाने के दो तरीके होगें।

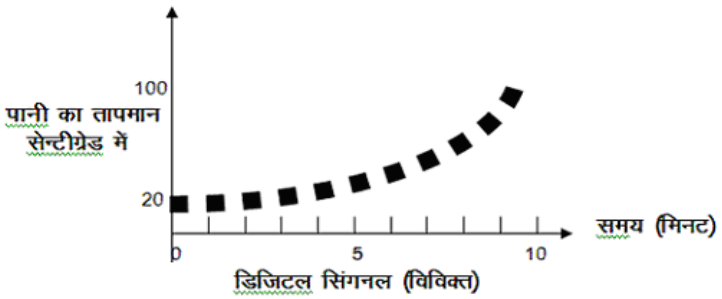
1. पानी के तापमान को निरंतर रिकार्ड किया जावे और यह पानी का तापमान 20 सेन्टीग्रेड से 100 सेन्टीग्रेड तक स्मूथली परिवर्तित होता है इस प्रक्रिया में पानी का तापमान 20 सेन्टीग्रेड से 100 सेन्टीग्रेड के मध्य प्रत्येक निष्चित मान से गुजरता है। यह **ऐनालॉग सिग्नल** का उदाहरण है। इस प्रकार ऐनालॉग सिग्नल सतत् होते है और प्रत्येक संभव मान को प्रदर्षित करते है। जिसे निम्न चित्र में दर्षाया गया है:–



ऐनालॉग सर्किटः–ऐसे सर्किट जिसका उपयोग ऐनालॉग सिग्नल को प्रोसेस किये जाने के लिये होता है उसे ऐनालॉग सर्किट कहते है।

ऐनालॉग सिस्टम – ऐसा सिस्टम जो इस प्रकार के ऑपरेषन के लिये बना होता है उसे ऐनालॉग सिस्टम कहते है।

2. यदि पानी के तापमान को प्रत्येक मिनट के अंतराल पर लिया जाकर रिकार्ड किया जावे तो इस प्रकरण में रिकार्ड किया गया तापमान सतत् नहीं होगा जबकी यह एक बिन्दु से दूसरे बिन्दु मे बीच जम्प करता है। और इस प्रकार 20 सेन्टीग्रेड तापमान से 100 सेन्टीग्रेड तापमान के बीच इसके कुछ निष्चित मान होगें। इस प्रकरण के लिये यह मान की संख्या 11 होगी। इसे निम्न ग्राफ के द्वारा दर्षाया गया है। इस प्रकार किसी मात्रा को विभिन्न बिन्दुओं को सीरिज में रिकार्ड किया जावे तो उसे सेम्पलस कहते है। यह **डिजिटल सिग्नल** का उदाहरण है। इस प्रकार डिजिटल सिग्नल किसी निष्चित संख्या के विभक्त मानों को प्रदर्षित करते है।



डिजिटल सर्किटः– ऐसे सर्किट जिसका उपयोग डिजिटल सिग्नल को प्रोसेस किये जाने के लिये होता है उसे डिजिटल सर्किट कहते है। वोल्टेज लेवल के आधार पर किसी इलेक्ट्रानिक्स सर्किट के ऑपरेषन को स्पष्ट किया जा सकता है। डिजिटल सर्किट के केस में सिर्फ दो वोल्टेज होते है एक अधिक पॉजिटिव वोल्टेज और एक कम मान का वोल्टेज। अधिक पॉजिटिव वोल्टेज का लेवल हाई होता है जिसे **H** से दर्षाते है और कम मान के वोल्टेज का लेवल कम होता हे जिसे **L** से दर्षाते है।

डिजिटल सिस्टम – ऐसा सिस्टम जो इस प्रकार के ऑपरेषन के लिये बना होता है या ऐसा इलेक्ट्रॉनिक्स सिस्टम जो कि डिजिटल सर्किट या डिवाइसेस से बना होता है, उसे डिजिटल सिस्टम कहते है।

डिजिटल सिग्नल – ऐसा सिग्नल जिसके केवल दो विभक्त मान होते है जिसे Low या High इसे **0** या **1** से भी दर्षाया जाता है, डिजिटल सिग्नल कहलाता है।

बाईनरी सिस्टम :–डिजिटल इलेक्ट्रॉनिक सर्किट की निष्चित दो संभव स्टेट होती है। ऐसा सिस्टम जिसकी सिर्फ दो स्टेट होती है बाइनरी (बाई–अर्थात दो) सिस्टम कहलाता है।

बाइनरी नम्बर सिस्टम – बाइनरी नम्बर सिस्टम की दो वेल्यू होती है या तो 0 या 1 बाईनरी नम्बर सिस्टम का उपयोग डिजिटल इलेक्ट्रॉनिक्स में किया जाता है। डिजिटल सिस्टम में अधिक पॉजीटिव वोल्टेज के लेवल को H से दर्षाते है और कम मान के वोल्टेज लेवल को L से दर्षाते है। यह बाईनरी नम्बर सिस्टम से रिलेटेड होता है जहॉ L=0 और H=1 असाईन किया जाता है। डिजिटल सर्किट के द्वारा कई फंक्षनस को लॉजिकल ऑपरेषनस से पूर्ण किया जाता है। इसी प्रकार सत्य (T) और असत्य के लिये (F) का उपयोग किया जाता है।

नम्बर सिस्टमसः– डिजिटल सर्किट के अर्थमेटिक में निम्न चार प्रकार के नम्बर सिस्टम का उपयोग किया जाता है–

1. डेसिमल नम्बर सिस्टम – इसका बेस या आधार या रेडिक्स 10 होता है अर्थात इस नम्बर सिस्टम को रिप्रेजेन्ट किये जाने के लिये 10 विभिन्न सिम्बोल्स 0,1,2,3,4,5,6,7,8,9 जिसे डिजिट्स के नाम से भी जाना जाता है, का उपयोग किया जाता है।

2. बाईनरी नम्बर सिस्टम – इस नम्बर सिस्टम का आधार 2 होता है अर्थात इस नम्बर सिस्टम को रिप्रेजेन्ट किये जाने के लिये 2 विभिन्न सिम्बोल्स 0 या 1 का उपयोग किया जाता है।

3. ऑक्टल नम्बर सिस्टम – इस नम्बर सिस्टम का आधार 8 होता है अर्थात इस नम्बर सिस्टम को रिप्रेजेन्ट किये जाने के लिये 8 विभिन्न सिम्बोल्स, 0,1,2,3,4,5,6,7 का उपयोग किया जाता है।

4. हेक्साडेसिमल नम्बर सिस्टम – इस नम्बर सिस्टम को आधार 16 होता है अर्थात इस नम्बर सिस्टम को रिप्रेजेन्ट किये जाने के लिये 16 विभिन्न सिम्बोल्स, 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F का उपयोग किया जाता है।

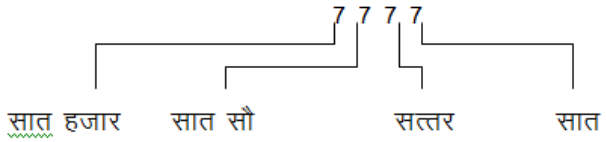
डेसिमल नम्बर सिस्टम –इस सिस्टम का बेस 10 होता है और यह स्थिती के मान पर निर्भर होने वाला सिस्टम है। अर्थात डिजिट का मान उसकी स्थिती पर निर्भर करता है।

(a) **बेस या रेडिक्स** – विभिन्न डिजिट्स की संख्या जिसके द्वारा नम्बर सिस्टम को रिप्रेजेन्ट किया जाता है बेस या रेडिक्स कहलाता है। इस नम्बर सिस्टम का बेस या रेडिक्स 10 होता है अर्थात इस नम्बर सिस्टम मे'10 विभिन्न सिम्बोल्स या डिजिट्स 0,1,2,3,4,5,6,7,8,9 का उपयोग किया जाता है। संख्या के प्रत्येक नम्बर की स्थिती को दर्षाये जाने के लिये उक्त डिजिट्स का प्रयोग किया जा सकता है।

(b) **स्थिती मान** – प्रत्येक डिजिट की ऐबसिलियुट वेल्यु स्थिर होती है किन्तु उसकी स्थिती वेल्यु (पोजिषन वेल्यू या वेट) का निर्धारण उस संख्या में उस डिजिट की स्थिती पर निर्भर करता है।

उदाहरणार्थ– 5000 में 5 की पोजिषन वेल्यु, 500 में स्थित 5 की पोजिशन वेल्यु के समान नहीं है।

संख्या 7777 में प्रत्येक 7 की पोजिषन वेल्यु निम्नानुसार होगी



उसी प्रकार एक संख्या 8530 को निम्नानुसार तोड़ा जा सकता है।

$8531= 8*10^3+5*10^2+3*10^1+1*10^0$

उक्त संख्या में अंक–1 लीस्ट सिंगनीफिकेन्ट डिजिट (LSD) है और अंक 8 मोस्ट सिंगनीफिकेन्ट डिजिट (MSD) है।

इसी प्रकार संख्या 8531.532 को निम्न प्रकार से लिखा जा सकता है–

$8531.532= 8*10^3+5*10^2+3*10^1+1*10^0+5*10^{-1}+3*10^{-2}+2*10^{-3}$

इस प्रकार स्पष्ट है कि,पोजिशन वेल्यु प्राप्त किये जाने के लिये उस नम्बर सिस्टम के आधार की घात को बढ़ाया जाकर स्थिती के अनुसार प्राप्त किया जा सकता है। दशमलव बिन्दु के बॉये घात की संख्या 0 से शुरू होती है और दशमलव के दाहिंने ओर घात की संख्या -1 से प्रारंभ होती है।

बाईनरी नम्बर सिस्टम – इस नम्बर सिस्टम का आधार 2 होता है अर्थात इस नम्बर सिस्टम को रिप्रेजेन्ट किये जाने के लिये 2 विभिन्न सिम्बोल्स 0 या 1 का उपयोग किया जाता है। इसे बिट्स भी कहते है। इस नम्बर सिस्टम में भी उक्तानुसार पोजिशन वेल्यु सिस्टम का उपयोग किया जाता है।

(a) बेस या रेडिक्स – इस नम्बर सिस्टम का बेस या रेडिक्स 2 होता है अर्थात इस नम्बर सिस्टम में दो विभिन्न डिजिट्स 0 और 1 का उपयोग किया जाता है। सभी बाईनरी नम्बर 0 और 1 की स्ट्रिन्गस से मिलकर बने होते है।

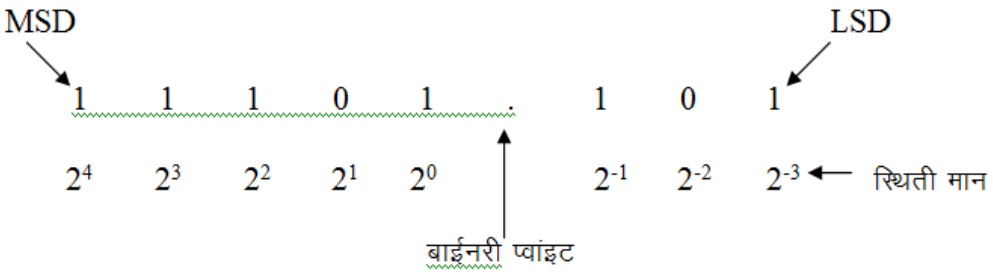
निबल– 2^2=4 बिट्स के समुह को **निबल** कहते है।
बाईट– 2^3= 8 बिट्स के ग्रुप को **बाईट** कहते है।
किलो बिट्स– 2^{10}= 1024 बिट्स = किलो बिट्स
मेगा बिट्स– 2^{20} = 1024*1024 बिट्स
गिगा बिट्स– 2^{30} = 1024*10^6
टेरा बिट्स – 2^{40} = 1024* 10

उदाहरणार्थ डेसिमल नम्बर से अलग किये जाने के लिये इस नम्बर सिस्टम में संख्या **11** को **वन–वन** और 11011 को **वन–वन–ज़ीरो–वन–वन** पड़ा जावेगा। तथा डेसिमल नम्बर सिस्टम से अलग किये जाने के लिये डेसीमल नम्बर सिस्टम के सबस्क्रिप्ट में 10 और बाईनरी नम्बर सिस्टम में सबस्क्रिप्ट 2 का उपयोग किया जाता है।

डेसिमल नम्बर – $(685)_{10}$, $(9985)_{10}$, $(15573)_{10}$
बाईनरी नम्बर – $(110)_2$, $(11011)_2$, $(1010)_2$

(b) स्थिति मान – डेसिमल नम्बर सिस्टम के समान बाईनरी सिस्टम में भी पोजिशन वेल्यु होती है। किन्तु बाईनरी के प्रकरण में प्रत्येक बिट का स्थिती मान आधार 2 के घातांक के रूप में होता है। बाईनरी प्वांइट के बाऐं साइड आधार 2 की घात 0 से प्रारंभ होकर बड़ती है जबकी बाईनरी प्वांइट के दांऐं साईड -1 से प्रारंभ होकर घटती है।

उदाहरणार्थ – एक 8 बिट के बाईनरी नम्बर 11101.101 के प्रत्येक बिट का स्थिती मान या वेट/पोजिशन वेल्यु निम्नानुसार दर्शाया जा सकता है–



बाईनरी प्वांइट के बाऐं पॉचवीं बिट का अधिकतम स्थिती मान होता है इसलिये इसे Most Significant Digit कहते है इसी प्रकार बाईनरी प्वाइंट के दाऐं साईड तीसरी बिट की स्थिती वेल्यु सबसे कम होती है इसलिये इसे Least Significant Digit कहते है।

बाईनरी से डेसिमल परिवर्तन – दिये गये बाईनरी इंटीजर (सम्पूर्ण) नम्बर को उसके तुल्य डेसीमल नम्बर में परिवर्तन करने के लिये निम्न विधि का उपयोग किया जाता है।

1. बाईनरी नम्बर की प्रत्येक बिट को पंक्ति में लिखते है।

उपरोक्तानुसार दर्शायी गई विधी अनुसार प्रत्येक बाईनरी बिट का वेट या स्थिती मान लिखा जाता है।

2. प्रत्येक वेट या स्थिती मान को जोड़ दिया जाता है।

यह बाईनरी के तुल्य डेसिमल नम्बर होगा ।

उदाहरण 1 :– $(110011)_2 = (?)_{10}$ को इसके तुल्य डेसीमल नम्बर में परिवर्तित किये जाने के लिये–

स्टेप–1	1	1	0	0	1	1
स्टेप–2 वेट निकालिये	$1*2^5$	$1*2^4$	$0*2^3$	$0*2^2$	$1*2^1$	$1*2^0$
	32	16	0	0	2	1
स्टेप–3 वेट को जोड़िये	32+16+2+1= 51					

उत्तर– $(110011)_2 = (51)_{10}$

उदाहरण 2:– $(1101.101)_2 = (?)_{10}$ को इसके तुल्य डेसीमल नम्बर में परिवर्तित किये जाने के लिये–

स्टेप–1	1	1	0	1	.	1	0	1
स्टेप–2 वेट निकालिये	$1*2^3$	$1*2^2$	$0*2^1$	$1*2^0$		$1*2^{-1}$	$0*2^{-2}$	$1*2^{-3}$
	8	4	0	1		1/2	0	1/8
स्टेप–3 वेट को जोड़िये	8+4+0+1+0.5+0.125= 13.625							

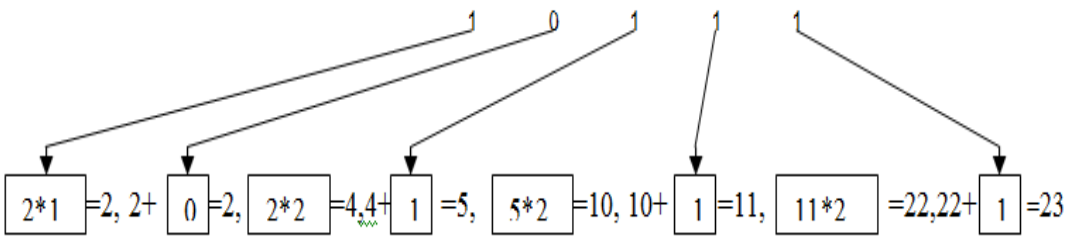
उत्तर– $(1101.101)_2 = (13.625)_{10}$

Double-Dadd विधी – इस विधी के द्वारा बाईनरी इंटीजर या पूर्णांक (ऐसी संख्या जिसमें कोई भी आंशिक भाग नहीं होता) को आसानी और कम समय में डेसीमल नम्बर में परिवर्तित किया जा सकता है। यह विधी निम्नानुसार है।

स्टेप–1:– सबसे बाईं बाईनरी बिट को डबल किया जाकर उसे दाईं बाईनरी बिट में जोड़ा जाता है।

स्टेप–2:– उक्तानुसार प्राप्त योग को डबल किया जाकर अगली दाई बाईनरी बिट में जोड़ा जाता है।

स्टेप–3:– स्टेप–2, को लगातार जारी रखते जब तक की अंतिम बिट अपनी पूर्व की डबल योग से जुड़ नहीं जाती है।



उदाहरण–2:– डबल डेड विधी का उपयोग कर $(110110)_2$ को उसके तुल्य डेसीमल नम्बर में बदलिये।

1. 2*1 = 2 इसे अगली बिट 1 में जोड़ने पर 2+1=3
2. 3*2= 6 इसे अगली बिट 0 में जोड़ने पर 6+0=6
3. 6*2= 12 इसे अगली बिट 1 में जोड़ने पर 12+1=13
4. 13*2= 26 इसे अगली बिट 1 में जोड़ने पर 26+1=27
5. 27*2= 54 इसे अगली बिट 0 में जोड़ने पर 54+0=54

डेसीमल से बाईनरी नम्बर में परिवर्तनः–

1.पूर्णांक– यदि दी गई संख्या पूर्णांक में है तो उसे बाईनरी नम्बर में परिवर्तन के लिये **डिवाईडेड बाई टू मेथड़** या **डबल ड़ेबल मेथड** का उपयोग किया जाता है। इस प्रक्रिया में दी गई संख्या को 2 से भाग देंगे तथा शेषफलों को सीधे हाथ की तरफ लिखेंगे ऐसा तब तक करेंगे तब तक कि भागफल शून्य नहीं हो जाता। अंत में शेषफलों को नीचे से ऊपर की ओर लिखकर बाइनरी संख्या प्राप्त करेंगे।

उधाहरण–1:–

2	12	0
2	6	0
2	3	1
2	1	1
	0	

शेषफल ↑ टॉप, बॉटम

उत्तर– $(12)_{10} = (1100)_2$

उधाहरण–2:– संख्या $(35)_{10}$ को बाईनरी में बदलिये ।

35÷2 = 17+ शेषफल 1 ↑ टॉप
17÷2 = 8 + शेषफल 1
8÷2 = 4 + शेषफल 0
4÷2 = 2 + शेषफल 0
2÷2 = 1 + शेषफल 0
1÷2 = 0 + शेषफल 1 ↑ बॉटम

शेषफलों को नीचे से उपर लिखा जाता है ।

$(35)_{10} = (100011)_2$

2. **आंशिक भाग–** इस प्रकरण में मल्टीप्लाई बाई 2 विधी का उपयोग किया जाता है। प्रत्येक बिट को 2 से गुणा किया जाता है एंव पूर्णांक वाले हिस्से को दाहिने हाथ की तरफ लिखना होगा तथा अंत में पूर्णांको को ऊपर से नीचे पढ़ना होगा।

उधाहरण–1:– संख्या $(0.625)_{10}$ को $(\ ?\ )_2$ में बदलिये ।

0.625*2 = 1.25 = 0.25 With a carry of 1 ↓ टॉप
0.25*2 = 0.50 = 0.50 With a carry of 0
0.50*2 = 1.00 = 0.00 With a carry of 0 ↓ बॉटम

पूर्णांकों को उपर से नीचे पढ़ा जावेगा

उत्तर– $(0.625)_{10} = (\ 100)_2$

उधाहरण–2:– संख्या $(0.80)_{10}$ को $(\ ?\ )_2$ में बदलिये ।

0.80*2 = 1.60 = 0.60 With a carry of 1 ↓ टॉप
0.60*2 = 1.20 = 0.20 With a carry of 1
0.20*2 = 0.40 = 0.40 With a carry of 0
0.40*2 = 0.80 = 0.80 With a carry of 0 ↓ बॉटम

उत्तर– $(0.80)_{10} \approx (\ 1100)_2$

इस उदाहरण से स्पष्ट है कि अपूर्णांको की स्थिति में प्रक्रिया बहुत देर तक चल सकती है कभी कभी तो इसका अंत प्राप्त किया ही नहीं जा सकता है। इस स्थिति में सन्नीकटन (Approximation) करना होगा। पूर्णांकों को उपर से नीचे पढ़ा जावेगा।

उधाहरण–3:– संख्या $(75.725)_{10}$ को $(?)_2$ में बदलिये ।

(अ) – पूर्णांक		(ब) – आंशिक भाग		
75÷2 = 37 + शेषफल 1	टॉप ↑	0.725*2 = 1.45	= 0.45 +1	टॉप ↓
37÷2 = 18 + शेषफल 1		0.45*2 = 0.90	= 0.90 + 0	
18÷2 = 9 + शेषफल 0		0.90*2 = 1.80	= 0.80 + 1	
9÷2 = 4 + शेषफल 1		0.80*2 = 1.60	= 0.60 + 1	
4÷2 = 2 + शेषफल 0		0.60*2 = 1.20	= 0.20 + 1	बॉटम
2÷2 = 1 + शेषफल 0				
1÷2 = 0 + शेषफल 1	बॉटम			

इसप्रकार सम्पूर्ण नम्बर – $(75.725)_{10} = (1001011.10111)_2$

साइनड बाईनरी नम्बरस –

साईन मेगनिट्यूड रिप्रेजेन्टेशन:– डेसीमल नम्बर सिस्टम में (+) चिन्ह का उपयोग धनात्मक संख्या के लिये किया जाता है और (-) चिन्ह का उपयोग ऋणात्मक संख्या को दर्शाने के लिये किया जाता है। धन चिन्ह को सामान्यतः छोड़ दिया जाता है। इस प्रकार चिन्ह की अनुपस्थिति, संख्या के धनात्मक होना दर्शाती है। नम्बरस का यह रिप्रेजेन्टेशन को साईनड नम्बर कहते है।

डिजिटल सर्किट सिर्फ दो सिम्बोलस 0 और 1 को समझते है। इस कारण बाईनरी नम्बर सिस्टम में संख्या के साईन को दर्शाने के लिये 0 या 1 का उपयोग किया जाता है। सामान्यतः एक अतिरिक्त बिट, साईन बिट के रूप में नम्बर के मोस्ट सिग्निफिकेंट बिट के स्थान पर रख दी जाती है। 0 बिट का उपयोग से धनात्मक संख्या को रिप्रेजेन्ट किया जाता है। जबकि बिट 1 ऋणात्मक संख्या को रिप्रेजेन्ट करती है।

उदाहरण – 8 बिट साईनड बाईनरी नम्बर 01000100 रिप्रेजेन्ट करती है धनात्मक संख्या को और इसकी वेल्यू/मेगनिट्यूड $(1000100)_2 = (68)_{10}$ लेफ्ट मोस्ट बिट 0 (MSB) जो यह दर्शाती है कि नम्बर धनात्मक है।

इसी प्रकार साइनड बाईनरी नम्बर 11000100 रिप्रेजेन्ट करती है कि यह ऋणात्मक संख्या है और इसकी वेल्यू/मेगनिट्यूड $(1000100)_2 = (68)_{10}$ है लेफ्ट मोस्ट बिट 1 (MSB) जो यह दर्शाती है कि नम्बर ऋणात्मक है और अन्य सात बिट इसका मेगनिट्यूड दर्शाती है।

इसप्रकार साईनड नम्बर का रिप्रेजेन्टेशन, साईनड मेगनिट्यूड रिप्रेजेन्टेशन कहलाता है।

उदाहरण:– नीचे बाईनरी नम्बर का साईनड मेगनिट्यूट रिप्रेजेन्टेशन दर्शाया गया है इस संख्या के तुल्य डेसिमल नम्बर ज्ञात कीजिये।

(अ) 1011001 (ब) 001000 (स) 01011 (द) 11111

हल–(अ) बाईनरी संख्या 1011001 में साईन बिट 1 यह दर्शाती है कि संख्या ऋणात्मक है। इस संख्या का
मेगनिट्यूड 011001 = $(25)_{10}$

इसलिये $(1011001)_2 = (-25)_{10}$

(ब) बाईनरी संख्या 001000 में साईन बिट 0 यह दर्शाती है कि संख्या धनात्मक है। इस संख्या का
मेगनिट्यूड 01000 = $(8)_{10}$

इसलिये $(001000)_2 = (+8)_{10}$

(स) बाईनरी संख्या 01011 में साईन बिट 0 यह दर्शाती है कि संख्या धनात्मक है। इस संख्या का
मेगनिट्यूड 1011 = $(11)_{10}$

इसलिये $(01011)_2 = (+11)_{10}$

(द) बाईनरी संख्या 11111 में साईन बिट 1 यह दर्शाती है कि संख्या ऋणात्मक है। इस संख्या का
मेगनिट्यूड 1111 = $(15)_{10}$

इसलिये $(11111)_2 = (-15)_{10}$

वनस् काम्प्लीमेन्ट रिप्रेजेन्टेशन :– बाईनरी नम्बर सिस्टम में यदि 1 को 0 से और 0 को 1 से रिप्लेस कर दिया जावे तो परिणामी नम्बर को पहली संख्या का वनस् काम्प्लीमेन्ट कहते है। इस प्रकार प्राप्त दोनों

संख्याऐं एक दूसरे की काम्प्लीमेन्ट होती है। यदि एक संख्या धनात्मक है तो इस संख्या का काम्प्लीमेन्ट, ऋणात्मक संख्या एंव समान मेगनिट्यूड और वाइस–वरसा होती है।

जैसे, यदि $(01111)_2$ रिप्रेजेन्ट करती है $(+15)_{10}$, इस रिप्रेजेन्टेशन में $(10000)_2$ रिप्रेजेन्ट करता $(-15)_{10}$

इस विधी का उपयोग साईनड नम्बर को रिप्रेजेन्ट किये जाने के लिये अधिकांशतः किया जाता है। इस रिप्रेजेन्टेशन में भी MSB 0 धनात्मक संख्या को एंव 1 ऋणात्मक संख्या को दर्शाती है।

उदाहरण 1:– निम्न बाईनरी नम्बर का वनस् काम्प्लीमेन्ट ज्ञात किजिये।

(अ) 111001010 (ब) 10101100

हलः– (अ) 000110101

(ब) 01010011

उदाहरण 2:– निम्न संख्याओं को वनस काम्प्लीमेन्ट फार्म में बदलिये।

(अ) +7 और –7 (ब) +8 और –8 (स) +15 और –15

हलः– (अ) $(+7)_{10} = (0111)_2$

और $(-7)_{10} = (1000)_2$

(ब) $(+8)_{10} = (01000)_2$

और $(-8)_{10} = (10111)_2$

(स) $(+15)_{10} = (01111)_2$

और $(-15)_{10} = (10000)_2$

नोटः–1. उक्त उदाहरण से स्पष्ट है कि, धनात्मक संख्या का साईन मेगनिट्यूड रिप्रेजेन्टेशन और वनस् काम्प्लीमेन्ट रिप्रेजेन्टेशन समान होता है। (सिर्फ धन साईन के लिये 0 का उपयोग किया जाता है।)

2. ऋणात्मक संख्या का वनस काम्प्लीमेन्ट रिप्रेजेन्टेशन ज्ञात करने के लिये, उस संख्या को धनात्मक संख्या मानते हुऐ उसका साईन मेगनिट्यूड रिप्रेजेन्टशन या वनस काम्प्लीमेन्ट रिप्रेजेन्टेशन ज्ञात करते है (क्योंकि धनात्मक संख्या का साईन मेगनिट्यूड रिप्रेजेन्टेषन और वनस काम्प्लीमेन्ट रिप्रेजेन्टेशन समान होता है) उसके बाद उस संख्या का काम्प्लीमेन्ट अर्थात 1 को 0 से एंव 0 को 1 से प्रतिस्थापित करते है।

इस स्थिति में भी MSB 0 धनात्मक संख्या को एंव 1 ऋणात्मक संख्या को दर्शाती है।

टूस काम्प्लीमेन्ट रिप्रेजेन्टेशन :–किसी बाईनरी नम्बर के वनस काम्प्लीमेन्ट में यदि 1 जोड़ दिया जावे तो परिणामी संख्या,दी गई बाईनरी संख्या की टूस काम्प्लीमेन्ट कहलाती है। इस रिप्रेजेन्टेशन में भी यदि MSB 0 है तो संख्या धनात्मक और यदि MSB 1 है तो संख्या ऋणात्मक होगी।

उदाहरण 1:– निम्न संख्याओं का टूस काम्प्लीमेन्ट ज्ञात किजिये।

(अ) 11001010 (ब) 101001101

हल– (अ) बाईनरी नम्बर– 11001010

वनस काम्प्लीमेन्ट– 00110101

टूस काम्प्लीमेन्ट के लिए वनस काम्प्लीमेन्ट में 1 को जोड़ा जावेगा।

```
  00110101
+        1
----------
  00110110
```

(ब) बाईनरी संख्या– 101001101

वन काम्प्लीमेन्ट – 010110010

टूस काम्प्लीमेन्ट के लिए वनस काम्प्लीमेन्ट में 1 को जोड़ा जावेगा।

```
  010110010
+         1
-----------
  010110011
```

नोटः– उक्त उदाहरण से स्पष्ट है कि

1. यदि दी गई संख्या की LSB 1 है तो इसका टूस काम्प्लीमेन्ट प्राप्त किये जाने के लिये LSB बिट 1 को छोड़कर प्रत्येक 0 को 1 और 1 को 0 से परिवर्तित किये जाकर संख्या का टूस काम्प्लीमेन्ट प्राप्त किया जा सकता है।
2. यदि दी गई संख्या की LSB 0 है तो इसका टूस काम्प्लीमेन्ट प्राप्त किये जाने के लिये LSB से MSB बिट की ओर स्केनिंग एंव उन बिट्स को यथावत रखा जाकर जब तक की LSB से MSB

बिट की ओर स्केनिंग के दौरान बिट 1 नहीं प्राप्त हो जाती, बिट 1 प्राप्त हो जाने पर उस बिट 1 को यथावत लिखा जाकर शेष बिट्स में 0 को 1 से और 1 को 0 से परिवर्तित किये जाने पर दी गई संख्या का टूस काम्प्लीमेन्ट प्राप्त किया जा सकता है।

उदाहरण 1:– निम्न संख्याओं का टूस काम्प्लीमेन्ट ज्ञात किजिये।

(अ) 01100101

हल– संख्या–	0	1	1	0	0	1	0	1
टूस काम्प्लीमेन्ट–	1	0	0	1	1	0	1	1

(ब) 11001100

हल– संख्या–	1	1	0	0	1	1	0	0
टूस काम्प्लीमेन्ट–	0	0	1	1	0	1	0	0

उदाहरण 2:– संख्या $(-20)_{10}$ को निम्नानुसार रिप्रेजेन्ट किजिये।

(1) साईन मेगनिट्यूड

(2) वनस काम्प्लीमेन्ट

(3) टूस काम्प्लीमेन्ट रिप्रेजेन्टेशन

हल– ऋणात्मक संख्या का वनस् काम्प्लीमेन्ट रिप्रेजेन्टेशन ज्ञात करने के लिये, उस संख्या को धनात्मक संख्या मानते हुऐ उसका साईन मेगनिट्यूड रिप्रेजेन्टेशन या वनस काम्प्लीमेन्ट रिप्रेजेन्टेशन ज्ञात करते है (क्योंकि धनात्मक संख्या का साईन मेगनिट्यूड रिप्रेजेन्टेशन और वनस काम्प्लीमेन्ट रिप्रेजेन्टेशन समान होता है) उसके बाद उस संख्या का काम्प्लीमेन्ट अर्थात 1 को 0 से एंव 0 को 1 से प्रतिस्थापित करते है।

$$(+20)_{10} = (010100)_2$$

अर्थात संख्या $(-20)_{10}$ को निम्न प्रकार रिप्रेजेन्ट किया जावेगा

(1) साईन मेगनिट्यूड – 110100

(2) वनस काम्प्लीमेन्ट रिप्रेजेन्टेशन – 101011

(3) टूस काम्प्लीमेन्ट रिप्रेजेन्टेशन – 101100

बाईनरी अरथमेटिक :– अरथमेटिक ऑपरेशन जैसे जोड़ना, घटाना, गुणा, एंव भाग किये जाना यह अरथमेटिक ऑपरेशन बाईनरी नम्बर के लिये भी किये जा सकते है जिसकी विधी डेसीमल नम्बर सिस्टम की अपेक्षा सरल होती है क्योंकि इस नम्बर सिस्टम में केवल दो ही बिट 0 और 1 होती है।

1. **बाईनरी का जोड़ किये जाना –** बाईनरी बिट्स को जोड़ने के लिये निम्न नियम होते है।

   अ. $0+0 = 0$

   ब. $0+1 = 1$

   स. $1+0 = 1$

   द. $1+1 = 0$ with a carry 1 या $(10)_2$

उदाहरण 1:– निम्न बाईनरी नम्बर को जोड़िये।

(अ) 10110 और 11001 (ब) 110011 और 101001

हल– (अ)

	1	0	1	1	0
+	1	1	0	0	1
1	0	1	1	1	1

Carry

हल–(ब)

			(1)	(1)	← Carry	
	1	1	0	0	1	1
+	1	0	1	0	0	1

1 0 1 1 1 0 0

Carry

उदाहरण 2:– निम्न बाईनरी नम्बर को जोड़िये।

111111+101011+110011

हल–

	(1)	(1)	(1)	$\begin{bmatrix}1\\1\end{bmatrix}$	(1)	
	1	1	1	1	1	1
	1	0	1	0	1	1
	1	1	0	0	1	1
1	0	0	1	1	1	0	1

नोटः–उक्त बाईनरी बिट्स के योग के उदाहरणों के अवलोकन से यह स्पष्ट होता है कि,

1. जिस कॉलम में जोड़ किया जाना है वहॉ 1 की संख्या सम है तो योग की बिट 0 होगी और यदि जिस कॉलम में जोड़ किया जाना है वहॉ यदि बिट 1 कि संख्या विषम है तो योग की बिट 1 होगी।
2. कॉलम में बिट 1 का प्रत्येक पेयर अथवा जोड़ा कैरी 1 उत्पन्न करता है जिसे अगले कॉलम के हायर बिट में जोड़ा जाता है।

2. बाईनरी का घटाना :– बाईनरी बिट्स को घटाने के लिये निम्न नियम होते है।

अ. 0-0 = 0

ब. 1-0 = 1

स. 1-1 = 0

द. 0-1 = 1 with a barrow of 1 from the next column of minuend या $(10)_2$-1 = 1

जब नियम ''द'' का उपयोग किया जाता है तो उधारी का या बॉरो, अगले कॉलम का शेष माइन्यूऐंड में से 1 कम कर देता है। यदि अगले कॉलम का माइन्यूऐंड 1 है तो वह 0 हो जावेगा एंव यदि अगले कॉलम का माइन्यूऐंड 0 है तो वह 1 में परिवर्तित हो जावेगा और यदि लगातार माइन्यूऐंड का अगला कॉलम 0 है तो वह 1 में परिवर्तित होता रहेगा जब तक कि 1 नहीं प्राप्त होकर वह 0 में परिवर्तित नहीं हो जाता।

उदाहरण 1:– निम्न बाईनरी संख्याओं को घटाइये।

$(1011)_2-(0110)_2$

हल–

	1	0	1	1	Minuend
–	0	1	1	0	Subtrahend
	0	1	0	1	Diffrence

उदाहरण 2:– बाईनरी संख्या 10110 में से बाईनरी संख्या 01011 घटाइये।

हल–

	1	0	1	1	0
–	0	1	0	1	1
	0	1	0	1	1

कॉलम 1:– 0–1 = 1 with a barrow of 1

कॉलम 2:– 0 (after barrow)–1 = 1 with a barrow of 1

कॉलम 3:– 0 (after barrow)–0 = 0

कॉलम 4:– 0–1 = 1 with a barrow of 1

कॉलम 5:– 0 (after barrow)–0 = 0

वनस काम्प्लीमेन्ट के द्वारा बाईनरी का घटाना :– इस विधी से बाईनरी बिट का घटाना निम्न नियमानुसार होगा–

1. सबट्राऐंड का वनस काम्प्लीमेन्ट ज्ञात किया जाता है।
2. प्राप्त इस वनस काम्प्लीमेन्ट को मान्यूऐंड़ में जोड़ा जाता है।
3. आखरी 1 या 0 पर ऐंड अरॉउन्ड कैरी को परफार्म किया जाता है।
4. यदि ऐंड अराउन्ड कैरी नहीं है अर्थात यह 0 है तब उत्तर के लिये उसका रिकाम्प्लीमेंट ज्ञात किये जाकर ऋण चिन्ह को इसके साथ संलग्न कर दिया जाता है।
5. यदि ऐंड अराउन्ड कैरी 1 है तो रिकाम्प्लीमेंट की आवष्यकता नहीं होती है। उस कैरी 1 को शेषफल में जोड़ दिया जाता है।

उदाहरण 1:– वनस काम्प्लीमेंट विधी का उपयोग कर बाईनरी नम्बर 1010 को बाईनरी संख्या 1110 में से घटाईये ।

हल– 1. सबट्राऐंड का वनस काम्प्लीमेन्ट ज्ञात किया जो – 0101 होगा।

2. प्राप्त इस वनस काम्प्लीमेन्ट को मान्यूऐंड़ में जोड़ा जाता है।

	1	1	1	0
+	0	1	0	1
1	0	0	1	1

Carry

3. उक्त दोनों संख्याओं को जोड़े जाने पर आखरी स्थिती पर कैरी 1 प्राप्त होती है, इस कैरी को शेषफल में जोड़ दिया जाता है जिसे ऐंड अरॉउन्ड कैरी कहते है। अर्थात–

	0	0	1	1
+				1
	0	1	0	0

उत्तर– $(0100)_2$

उदाहरण 2:– वनस काम्प्लीमेंट विधी का उपयोग कर बाईनरी नम्बर 11011 को बाईनरी संख्या 10001 में से घटाईये।

हल – 1. सबट्राऐंड का वनस काम्प्लीमेन्ट ज्ञात किया जो – 00100 होगा।

2. प्राप्त इस वनस काम्प्लीमेन्ट को मान्यूऐंड़ में जोड़ा जाता है।

	1	0	0	0	1
+	0	0	1	0	0
	1	0	1	0	1

No carry

3. इस प्रकरण में ऐंड अरॉउन्ड़ कैरी नहीं है तो नियम 4 के अनुसार इस प्राप्त संख्या का रिकाम्प्लीमेंट लिया जाकर ऋण चिन्ह को संलग्न कर दिया जाता है। जो उत्तर होगा।

अर्थात – उत्तरः– $(-01010)_2$

टूस काम्प्लीमेन्ट के द्वारा बाईनरी का घटाना :– इस विधी से बाईनरी बिट का घटाना निम्न नियमानुसार होगा।

1. सबट्राऐंड का टूस काम्प्लीमेन्ट ज्ञात किया जाता है।

2. प्राप्त इस टूस काम्प्लीमेन्ट को मान्यूऐंड़ में जोड़ा जाता है।
3. आखरी कैरी को छोड़ दिया जाता है।
4. यदि आखरी कैरी 1 है तब उत्तर धनात्मक होगा और उसके रिकाम्प्लीमेंट की आवश्यकता नहीं होगी।
5. यदि आखरी कैरी 0 है तो रिकाम्प्लीमेंट लिया जाकर ऋण चिन्ह को संलग्न कर दिया जाता है।

उदाहरण 1:– टूस काम्प्लीमेंट विधी का उपयोग कर बाईनरी नम्बर 10000 को बाईनरी संख्या 10110 में से घटाईये।

हल – 1. सबट्राऐंड का वनस काम्प्लीमेन्ट ज्ञात किया जो 01111 होगा। इससे टूस काम्प्लीमेंट प्राप्त किया जाता है। टूस काम्प्लीमेंट :– 01111+1 = 10000

2. प्राप्त इस टूस काम्प्लीमेन्ट को मान्यूऐंड़ में जोड़ा जाता है।

	1	0	1	1	0
	1	0	0	0	0
1	0	0	1	1	0

ड्राप किये जाने पर फायनल उत्तर 00110 होगा।

उदाहरण 2:– टूस काम्प्लीमेंट विधी का उपयोग कर बाईनरी नम्बर 1101 को बाईनरी संख्या 1010 में से घटाईये।

हल – 1. सबट्राऐंड का वनस काम्प्लीमेन्ट ज्ञात किया जो 0010 होगा। इससे टूस काम्प्लीमेंट प्राप्त किया जाता है । टूस काम्प्लीमेंट :– 0010+1 = 0011

2. प्राप्त इस टूस काम्प्लीमेन्ट को मान्यूऐंड़ में जोड़ा जाता है।

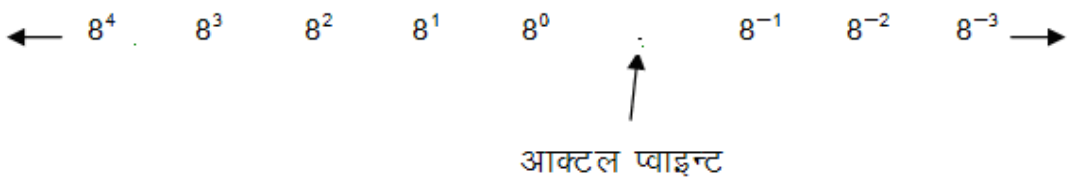
1	0	1	0	
0	0	1	1	
1	1	0	1	

कोई कैरी नहीं है। इस स्थिती में उत्तर का रिकाम्प्लीमेन्ट लिये जाने के लिये इसमें से 1 को घटाया जाता है तो 1100 प्राप्त होता है। पश्चात रिकाम्प्लीमेन्ट लिये जाने पर एवं माइनस साईन अटैच किये जाने पर यह संख्या –0011 प्राप्त होती है जो कि फायनल उत्तर है।

ऑक्टल नम्बर सिस्टमः– इस सिस्टम का बेस 8 होता है और यह स्थिती के मान पर निर्भर होने वाला सिस्टम है। अर्थात डिजिट का मान उसकी स्थिती पर निर्भर करता है।

(a) **बेस या रेडिक्स–** विभिन्न डिजिट्स की संख्या जिसके द्वारा नम्बर सिस्टम को रिप्रेजेन्ट किया जाता है बेस या रेडिक्स कहलाता है। इस नम्बर सिस्टम का बेस या रेडिक्स 8 होता है अर्थात इस नम्बर सिस्टम को 08 विभिन्न सिम्बोल्स या डिजिट्स 0,1,2,3,4,5,6, और 7 के उपयोग से प्रदर्शित किया जा सकता है।

(b) **स्थिती मान–** प्रत्येक डिजिट की ऐबसिलियुट वेल्यु स्थिर होती है किन्तु उसकी स्थिती वेल्यु (पोजिषन वेल्यू/वेट) का निर्धारण उस संख्या में उस डिजिट की स्थिती पर निर्भर करता है।



आक्टल से डेसिमल परिवर्तनः–

उदाहरणार्थ– $(440.123)_8 = 4*8^2+4*8^1+0*8^0+1*8^{-1}+2*8^{-2}+3*8^{-3}$
$= 4*64+32+0+1/8+2/64+3/512$
$= 256+32+0.125+0.03125+0.0058$

उत्तर– $(440.123)_8 = (288.16)_{10}$

डेसीमल से आक्टल में परिवर्तनः–

उधाहरण– $(180.15)_{10} = (?)_8$

इंटीजर पार्ट लेने पर

180/8 = 22 , शेषफल 4 ↑
22/8 = 2, शेषफल 6
2/8 = 0. शेषफल 2

शेषफल को विपरित कम में लेने पर $(180)_{10} = (264)_8$

फिक्शन पार्ट लेने पर

0.15*8 =1.20= 0.20 कैरी 1 ↓
0.20*8 = 1.60=0.60 कैरी 1
0.60*8 = 4.80=0.80 कैरी 4

इसलिये $(0.15)_{10} = (.114)_8$

इसप्रकार फायनल परिणाम निम्न होगा

उत्तर– $(180.15)_{10} = (264.114)_8$

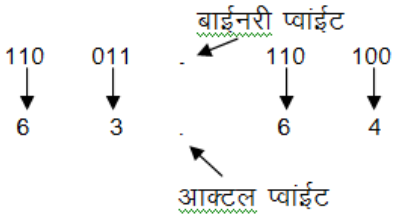
बाईनरी से आक्टल में परिवर्तनः– इस प्रकार के परिवर्तन में आक्टल प्वाइंट से प्रारंभ कर बाईनरी को तीन के ग्रुप में बॉट दिया जाता है और प्रत्येक ग्रुप को उसके तुल्य आक्टल नम्बर में परिवर्तित कर दिया जाता है। ऑक्टल प्वाइंट के बाऐं और तीन बिट का ग्रुप बनाते वक्त यदि तीन ग्रुप के लिये बिट कम होतो बाईनरी नम्बर के आगे 0 बिट को जोड़ दिया जाता है और आक्टल प्वांइट के दाहिनी और तीन बिट का ग्रुप बनाने के लिये बाईनरी संख्या के पीछे अर्थात दायें ओर 0 बिट को जोड़ दिया जाता है।

उधाहरण– $(110011.1101)_2 = (?)_8$

बाईनरी प्वांईट

110 011 . 110 100
↓ ↓ ↓ ↓
6 3 . 6 4

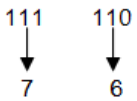
आक्टल प्वांईट



उत्तर–– $(110011.1101)_2 = (63.64)_8$

उधाहरण– $(111110)_2 = (?)_8$

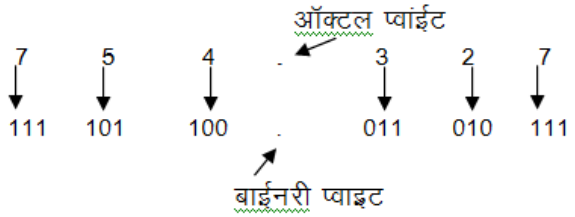
111 110
↓ ↓
7 6



उत्तर–– $(111110)_2 = (76)_8$

आक्टल से बाईनरी में परिवर्तनः– इस प्रकार के परिवर्तन में आक्टल प्वाइंट से प्रारंभ कर आक्टर बिट को तीन के बाईनरी ग्रुप में लिख दिया जाता है। तीन बिट का ग्रुप बनाते वक्त यदि तीन ग्रुप के लिये बिट कम हो तो बाईनरी नम्बर के आगे अर्थात बायें 0 बिट को जोड़ दिया जाता है।

उधाहरण– $(754.327)_8 = (?)_2$



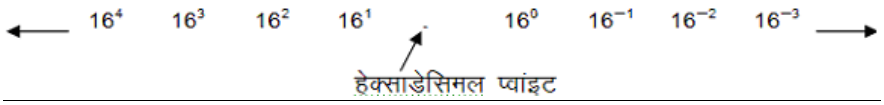
उत्तर–– $(754.327)_8 = (111101100.011010111)_2$

उपयोग – स्पष्ट है कि आक्टल नम्बर सिस्टम में नम्बर की संख्या बाईनरी नम्बर सिस्टम की अपेक्षा एक तिहाई होती है।

हैक्साडेसीमल नम्बर सिस्टमः– इस सिस्टम का बेस 16 होता है और यह स्थिती के मान पर निर्भर होने वाला सिस्टम है। अर्थात डिजिट का मान उसकी स्थिती पर निर्भर करता है।

(a) **बेस या रेडिक्स–** विभिन्न डिजिट्स की संख्या जिसके द्वारा नम्बर सिस्टम को रिप्रेजेन्ट किया जाता है बेस या रेडिक्स कहलाता है। इस नम्बर सिस्टम का बेस या रेडिक्स 16 होता है अर्थात इस नम्बर सिस्टम को 16 विभिन्न सिम्बोल्स या डिजिट्स 0,1,2,3,4,5,6,7,8,9, A, B, C, D, E, F के उपयोग से प्रदर्शित किया जा सकता है।

(b) **स्थिती मान–** प्रत्येक डिजिट की ऐबसिलियुट वेल्यु स्थिर होती है किन्तु उसकी स्थिती वेल्यु (पोजिषन वेल्यू/वेट) का निर्धारण उस संख्या में उस डिजिट की स्थिती पर निर्भर करता है।



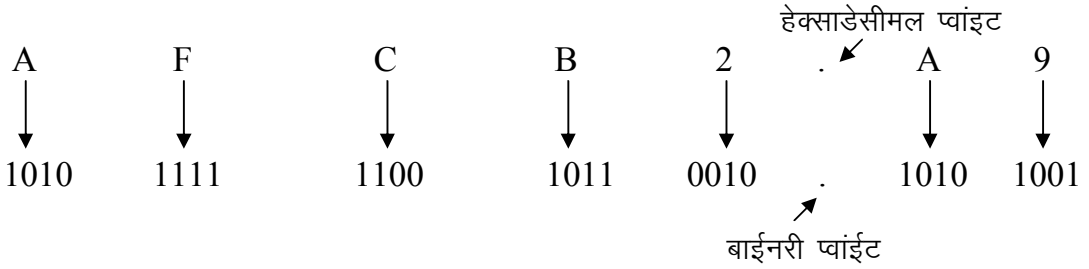
बाईनरी से हेक्साडेसीमल में परिवर्तनः– इस प्रकार के परिवर्तन में बाईनरी प्वांइट से प्रारंभ कर बाईनरी को चार ग्रुप में बॉट दिया जाता है और प्रत्येक ग्रुप को उसके तुल्य हेक्साडेसीमल नम्बर में परिवर्तित कर दिया जाता है। बाईनरी प्वाइंट के बाऐं और चार बिट का ग्रुप बनाते वक्त यदि चार ग्रुप के लिये बिट कम हो तो बाईनरी नम्बर के आगे 0 बिट को जोड़ दिया जाता है और बाईनरी प्वांइट के दाहिनी और चार बिट का ग्रुप बनाने के लिये बाईनरी संख्या के पीछे अर्थात दायें ओर 0 बिट को जोड़ दिया जाता है।

उदाहरणः– $(1110010101111)_2 = (?)_{16}$

उत्तर :– $(1110010101111)_2 = (1CAF)_{16}$

हेक्साडेसीमल से बाईनरी में परिवर्तनः– इस प्रकार के परिवर्तन में हेक्साडेसीमल प्वाइंट से प्रारंभ कर हेक्साडेसीमल बिट को चार के बाईनरी ग्रुप में बॉट दिया जाता है। चार बिट का ग्रुप बनाते वक्त यदि चार ग्रुप के लिये बिट कम हो तो बाईनरी नम्बर के आगे अर्थात बायें 0 बिट को जोड़ दिया जाता है ।

उदाहरणः– $(AFCB2.A9)_{16} = (?)_2$



हेक्साडेसीमल प्वांइट

A	F	C	B	2	.	A	9
↓	↓	↓	↓	↓		↓	↓
1010	1111	1100	1011	0010	.	1010	1001

बाईनरी प्वांईट

उत्तरः– $(AFCB2.A9)_{16} = (10101111110010110010.10101001)_2$

<u>**उपयोगः–**</u> उपरोक्त उदाहरण से स्पष्ट है हेक्साडेसिमल नम्बर सिस्टम में संख्या की लम्बाई बाईनरी नम्बर सिस्टम की अपेक्षा एक चौथाई होती है। इसलिये कम्प्यूटर प्रोग्राम या कम्प्यूटर ऑपरेशन में डाटा को हेण्डल किया जाना बहुत आसान है जबकि वह ऑक्टल या हेक्सानम्बर नम्बर सिस्टम में हो।

<u>लॉजिक गेट्स (Logic Gates)</u>

<u>**लॉजिक गेट्सः–**</u> लॉजिक गेट एक इलेक्ट्रॉनिक सर्किट होता है जो कि लॉजिक डिसिजन देता है। इसके एक आउटपुट और एक या एक से अधिक इनपुट्स हो सकते है। आउटपुट सिंनल, इनपुट सिग्नल के कुछ निश्चित कॉम्बिनेशन के पश्चात प्राप्त होता है। लॉजिक गेट्स बेसिक बिल्डिंग ब्लाक्स होते है जिससे अधिकांश डिजिटल सिस्टम बनाये जाते है। प्रत्येक गेट का एक संकेत होता है और प्रत्येक गेट के द्वारा की जाने वाली किया को एक समीकरण के द्वारा दर्शाया जा सकता है। इस समीकरण को बूलियन समीकरण कहते है। प्रत्येक गेट के सम्बन्ध में एक तालिका बनाई जाती है, जिसमें इनपुट के विभिन्न मानों के लिए आउटपुट का मान दर्शाया जाता है। इस टेबल को ट्रुथ टेबल कहते है।
यह गेट्स विभिन्न आई.सी. की विभिन्न फेमिलीज में पाये जाते है। कुछ महत्वपूर्ण आई.सी. फैमेली निम्न है। ट्रांजिस्टर–ट्रांजिस्टर लॉजिक (TTL), ईमिटर कपल्ड़ लॉजिक (ECL), मेटल ऑक्साईड सेमीकन्डक्टर (MOS), और कम्प्लीमेन्ट्री मेटल ऑक्साईड़ सेमीकंडक्टर (CMOS)

बेसिक लॉजिक गेट्स (फन्डामेन्टल गेट्स) :– NOT Gate (Inverter), OR Gate, और AND Gate इसे आधारभूत गेट्स भी कहते है क्योंकि इसका उपयोग किसी भी डिजिटल सिस्टम को बनाये जाने के लिये किया जा सकता है।

AND Gate :- इस प्रकार के गेट का आउटपुट 1 तभी होता है जब सारे इनपुट 1 हो। अर्थात यदि एक भी इनपुट 0 होगा तो आउटपुट 0 होता है। इस गेट में कम से कम दो इनपुट का होना आवश्यक है।

लॉजिक सिम्बोल–

जहॉ Q आउटपुट है और A तथा B इनपुट है = A.B

बुलियन समीकरण :– Q= A.B

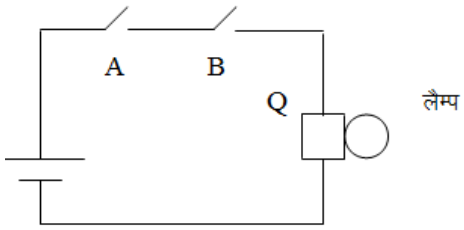
सत्य तालिका :–

(Truth table)

Inputs		Output
A	B	Q
0	0	0
0	1	0
1	0	0

1	1	1

स्वीचिंग सर्किट–



उक्त स्वीचिंग सर्किट से स्पष्ट है कि लैम्प तभी ग्लो (ऑन) होगा जबकि A और B दोंनों ऑन हो अन्यथा दोनों या कोई भी एक ऑफ होने पर लैम्प ग्लो नहीं होगा।

OR Gate :- इस प्रकार के गेट का आउटपुट 1 तभी होता है जब कोई भी एक इनपुट 1 हो। तथा आउटपुट 0 होता है जब सभी इनपुट 0 हो। इस गेट में कम से कम दो इनपुट का होना आवश्यक है ।

लॉजिक सिम्बोल–

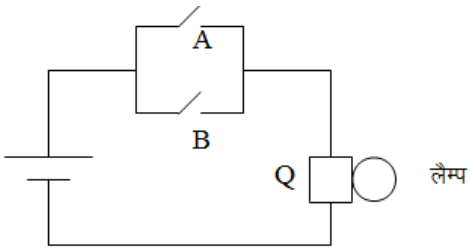
जहॉ Q आउटपुट है और A तथा B इनपुट है = A+B

बुलियन समीकरण :– Q= A+B

सत्य तालिका :–

Inputs		Output
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

स्वीचिंग सर्किट :–



उक्त स्वीचिंग सर्किट में यातो A या B या दोनों ऑन होने पर लैम्प ग्लो होता है जबकी यदि A और B दोनों ऑफ होने पर लैम्प ग्लो नहीं होगा। अर्थात आउटपुट जीरो होता है जबकि दोनो इनपुटस जीरो हो अन्यथा आउटपुट हाई होगा।

NOT Gate (Inverter) :- इस गेट में केवल एक ही इनपुट होता है। यदि इनपुट 0 होता है तो आउटपुट 1 प्राप्त होता है और यदि इनपुट 1 होता है तो आउटपुट 0 प्राप्त होता है। इस कारण इसे इनवर्टर भी कहते है।

लॉजिक सिम्बोल–

जहॉ Q आउटपुट है तथा A यदि इनपुट है तो

बुलियन समीकरण :– $Q=\overline{A}$

सत्य तालिका :–

Inputs	Output
A	Q
0	1
1	0

Universal Gates:- युनिवर्सल गेट वह होते है जिसके द्वारा किसी भी तार्किक परिपथ को बनाया जा सकता है।

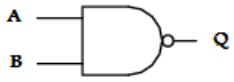
यह दो प्रकार के होते है।

1. NAND Gate 2. NOR Gate

1. NAND Gate:- इस गेट को एक NOT और एक NAND गेट की सहायता से बनाया जा सकता है।

$\overline{\text{NOT}}+\overline{\text{AND}} = \text{NAND}$

लॉजिक सिम्बोल–



जहॉ Q आउटपुट है तथा A एवं B इनपुट है तो

बुलियन समीकरण :– $Q=\overline{A.B}$

सत्य तालिका :–

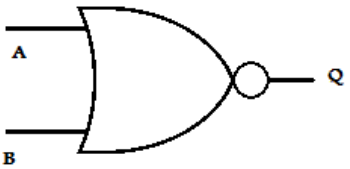
Inputs		**Output**
A	**B**	
0	**0**	**1**
0	**1**	**1**
1	**0**	**1**
1	**1**	**0**

उक्त सत्य तालिका से स्पष्ट है कि इस गेट का आउटपुट 0 तभी होता है जबकि इसके सभी इनपुट 1 हो अन्यथा कोई भी एक इनपुट 0 होने पर आउटपुट 1 प्राप्त होता है।

2. NOR Gate:- इस गेट को एक NOT और एक OR गेट की सहायता से बनाया जा सकता है।

$\overline{\text{NOT}}+\overline{\text{OR}} = \text{NOR}$

लॉजिक सिम्बोल–



जहॉ Q आउटपुट है तथा A एवं B इनपुट है तो

बुलियन समीकरण :– $Q=\overline{A+B}$

सत्य तालिका :–

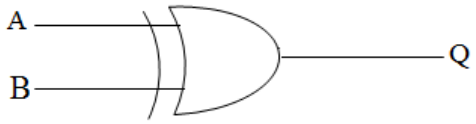
Inputs		**Output**
A	**B**	
0	**0**	**1**
0	**1**	**0**
1	**0**	**0**
1	**1**	**0**

उक्त सत्य तालिका से स्पष्ट है कि इस गेट का आउटपुट 1 तभी होता है जबकी इसके सभी इनपुट 0 हो अन्यथा कोई भी एक इनपुट 1 होने या सभी इनपुट 1 होने पर आउटपुट 0 प्राप्त होता है।

Special Logic Gates:-

XOR Gate (Exclusive OR Gate):- यह वह गेट होता है जिसके इनपुट में यदि 1 की संख्या विषम हो तो आउटपुट 1 होता है अन्यथा 0 होता है ।

लॉजिक सिम्बोल–



जहॉ Q आउटपुट है तथा A एवं B इनपुट है तो

बुलियन समीकरण :–

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

सत्य तालिका :–

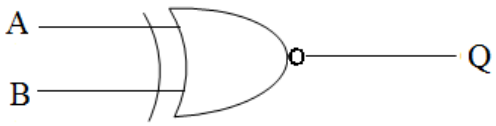
Inputs		Output
A	B	
0	0	0
0	1	1
1	0	1
1	1	0

XOR Gate की सत्य तालिका से निम्न निष्कर्ष प्राप्त होते है–

1. यदि A=B होतो आउटपुट 0 होगा और यदि A≠B नहीं है तो आउटपुट 1 होगा अर्थात यह गेट दो बाईनरी बिटों की तुलना करने का कार्य करता है । इस कारण इस गेट को कम्परेटर सर्किट में भी उपयोग किया जा सकता है।
2. सत्य तालिका से यह भी निष्कर्ष प्राप्त होता है कि यह दो बाईनरी बिटों को जोडने का कार्य करता है इसलिये इसका उपयोग ऐडर सर्किट में किया जाता है ।
3. इस गेट का आउटपुट का विश्लेषण करने पर ज्ञात होता है कि यह दो बाईनरी बिट्स को घटाने का भी कार्य करता है इस लिये इस गेट का उपयोग बाईनरी सबट्रेक्टर के लिये भी किया जा सकता है।
4. इस गेट में यदि इनपुट पर यदि 1 की संख्या विषम है तो आउटपुट 1 होगा अन्यथा आउटपुट 0 होगा इसलिये इस गेट का उपयोग पेरिटी चेकर या पेरिटी जनरेटर सर्किट में भी किया जा सकता है।

XNOR Gate (Exclusive NOR Gate):- यह वह गेट होता है जिसके इनपुट में यदि 1 की संख्या विषम हो तो आउटपुट 0 होता है अन्यथा 1 होता है। अथवा इस गेट का आउटपुट हाई या 1 होता है जबकि इसके सभी इनपुट्स समान हो तथा कोई भी एक इनपुट हाई या 1 किन्तु दोनों इनपुट 1 नहीं होने पर आउटपुट लो या 0 प्राप्त होता है।

लॉजिक सिम्बोल–



जहॉ Q आउटपुट है तथा A एवं B इनपुट है तो

बुलियन समीकरण :– Q=

सत्य तालिका :–

Inputs		Output
A	B	

0	**0**	**1**
0	**1**	**0**
1	**0**	**0**
1	**1**	**1**

XNOR Gate की सत्य तालिका से निम्न निष्कर्ष प्राप्त होते है।

1. यदि A=B होतो आउटपुट 1 होगा और यदि A≠B नहीं है तो आउटपुट 0 होगा अर्थात यह गेट दो बाईनरी बिटों की तुलना करने का कार्य करता है। इस कारण इस गेट को कम्परेटर सर्किट में भी उपयोग किया जा सकता है।
2. इस गेट में यदि इनपुट पर यदि 1 की संख्या विषम है तो आउटपुट 0 होगा तथा सम के लिये आउटपुट 1 होगा इसलिये इस गेट का उपयोग पेरिटी चेकर या पेरिटी जनरेटर सर्किट में भी किया जा सकता है।

लॉजिक सर्किटः– जिन सर्किट में लॉजिक गेट होते है उन्हे लॉजिक सर्किट कहते है। यह दो प्रकार के होते है।

1. काम्बिनेशनल लॉजिक सर्किट
2. सिक्वेन्शल लॉजिक सर्किट

Boolean Algebra:- बुलियन ऐलजेबरा, गणितीय लॉजिक का सिस्टम है। यह साधारण ऐलजेबरा और बाईनरी नम्बर सिस्टम से भिन्न होता है। बुलियन ऐलजेबरा में 1+1=1 जबकी बाईनरी अर्थमेटिक में इसका परिणाम 10 होता है। बुलियन सिस्टम में दो नियतांक होते है 0 और 1। इसमें कोई नेगेटिव या फ्रेक्शन संख्या नहीं होती है। इसका नाम बुलियन ऐलजेबरा, जार्ज बुल के नाम से दिया गया है जिन्होने सर्वप्रथम लॉजिक के ऐलजेबरा को कम्प्यूटर डिवाईस के ऑपरेशन के लिये उपयोग किया था।

इसप्रकार यदि $A=1$, then $A \neq 0$

यदि $A=0$, then $A \neq 1$

बुलियन ऐलज़ेब्रा के नियमः–

1. काम्प्लीमेन्ट का नियम– काम्प्लीमेन्ट टर्म का अर्थ है विपरीत अर्थात 1 का परिवर्तन 0 व 0 का परिवर्तन 1 होगा। इसके निम्न नियम है।

अ. नियम 1 $\overline{0}=1$

ब. नियम 2 $\overline{1}=0$

स. नियम 3 यदि $A=0$ तब $\overline{A}=1$

स. नियम 4 यदि $A=1$ तब $\overline{A}=0$

द. नियम 5 $\overline{\overline{A}}=A$

2. AND का नियम–

अ. नियम 1 $A.0=0$

ब. नियम 2 $A.1=A$

स. नियम 3 $A.A=A$

द. नियम 4 $A.\overline{A}=0$

3. OR का नियम–

अ. नियम 1 $A+0=A$

ब. नियम 2 $A+1=1$

स. नियम 3 $A+A=A$

द. नियम 4 $A+\overline{A}=1$

4. साहचर्य का नियम–

अ. नियम 1 $A+(B+C)=(A+B)+C$

ब. नियम 2 $A.(B.C) = (A.B).C$

5. **वितरण का नियम–**

अ. नियम 1 $A.(B+C) = (A.B)+(A.C)$

ब नियम 2 $A+(B.C) = (A+B).(A+C)$

स नियम 3 $A+(\bar{A}.B) = A+B$

6. **क्रम विनिमय का नियम–**

अ. नियम 1 $A+B = B+A$

7. **अवशोषण का नियम–**

अ. नियम 1 $A+A.B = A$

ब. नियम 2 $A.(A+B) = A$

स. नियम 3 $A.(\bar{A}+B) = A.B$

डी–मार्गन का थ्योरिम :– डी–मार्गन के दो नियम है जो कि जटिल तार्किक कथन को सरलीकृत किये जाने में अति महत्वपूर्ण है।

नियम–1:– जोड़ का काम्प्लीमेन्ट, काम्प्लीमेन्ट के गुणा के बराबर होता है।

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

नियम–2:– गुणा का काम्प्लीमेन्ट, काम्प्लीमेन्ट के योग के बराबर होता है।

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

1.काम्बिनेशनल लॉजिक सर्किट– ऐसे सर्किट में आउटपुट केवल इनपुट पर निर्भर करता है न की उसकी पुरानी अवस्था पर। इस सर्किट के निम्न उदाहरण है–

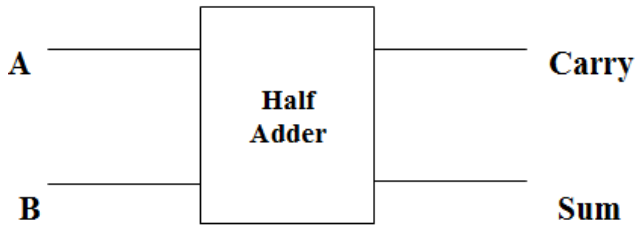
लॉजिक गेट्स, हाफ एडर, फुल एडर, मल्टिप्लेक्सर, डिमल्टिप्लेक्सर, डिकोडर और इन्कोडर आदि।

2. **सिक्वेन्शल लॉजिक सर्किट–** इस प्रकार के सर्किट में आउटपुट वर्तमान इनपुट और उसकी पुरानी अवस्था पर भी निर्भर करता है। इस सर्किट के उदाहरण है– फ्लिप–फ्लॉप, रजिस्टर, काउन्टर।

काम्बिनेशनल लॉजिक सर्किट–

हॉफ एडरः– यह एक ऐसा कांबिनेशनल लॉजिक सर्किट है जो दो बिटों का अंकगणितीय योग प्रदान करता है हॉफ एडर कहलाता है।

ब्लॉक डायग्राम ऑफ हॉफ एडरः–

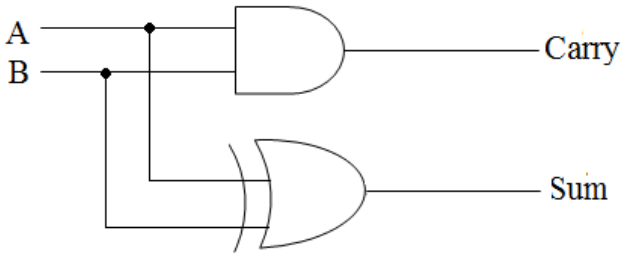


A और B दो इनपुट बाईनरी बिट्स है जिसका आउटपुट योग (सम) और हासिल (कैरी) है।

सत्य तालिकाः– निम्न प्रकार होगी।

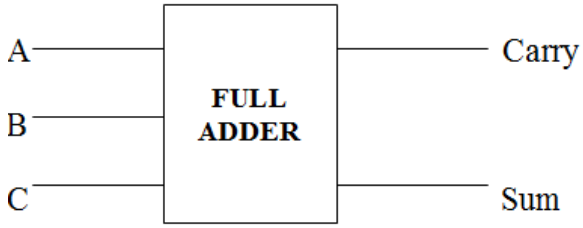
Inputs		CARRY	SUM
A	B		
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

लॉजिक सर्किटः– उक्त सत्य तालिका का परीक्षण करने पर स्पष्ट है कि सम प्राप्त किये जाने के लिये XOR Gate का उपयोग किया जाता है और कैरी AND Gate से प्राप्त होगी। इस प्रकार इसका लॉजिक सर्किट निम्नानुसार होगा–



Sum= AB+AB
Carry= A.B

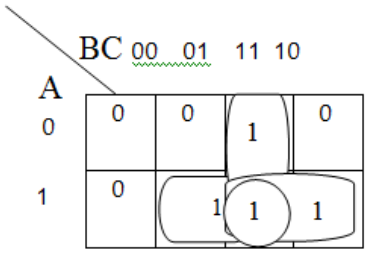
फुल एडरः– यह एक ऐसा कांबिनेशनल लॉजिक सर्किट है जो एक समय पर तीन बाईनरी बिट्स का अंकगणितीय योग प्रदान करता है। अर्थात इसमें तीन इनपुट और दो आउटपुट सम और कैरी होते है।
ब्लॉक डायग्राम ऑफ फुल एडरः–



सत्य तालिका truth table :–

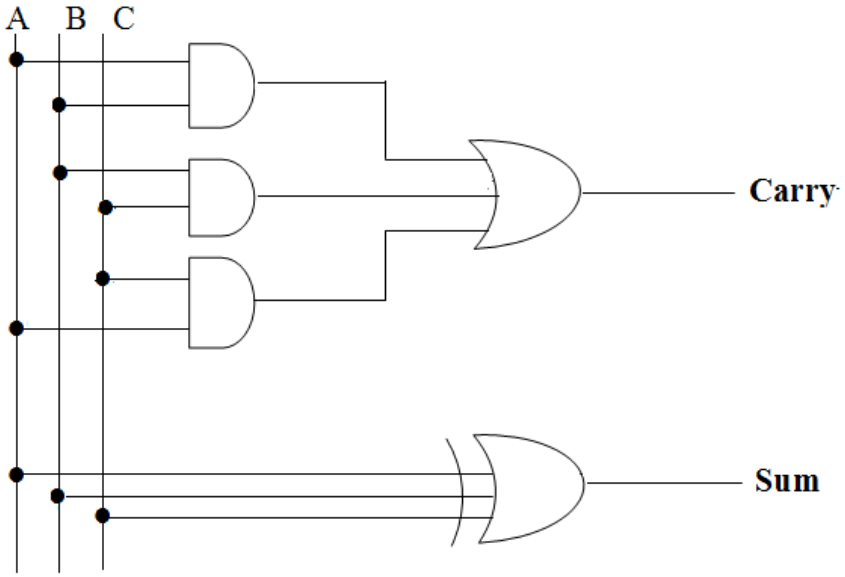
Inputs			Carry	Sum
A	B	C		
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

उक्त सत्य तालिका का परीक्षण किये जाने पर ज्ञात होता है कि सम को तीन इनपुट वाले XOR Gate से प्राप्त किया जा सकता है किन्तु कैरी को एक AND Gate से नहीं प्राप्त किया जा सकता है। इसलिये कैरी को प्राप्त किये जाने के लिये किरनॉप मैप का प्रयोग किया जाकर इस लॉजिक व्यंजक प्राप्त किया जा सकता है।

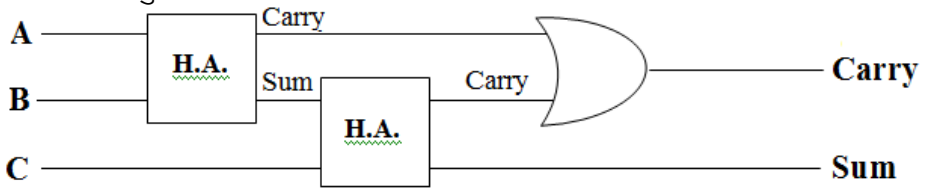


कैरी:– A.B+B.C+C.A

लॉजिक सर्किट ऑफ फुल एडरः–

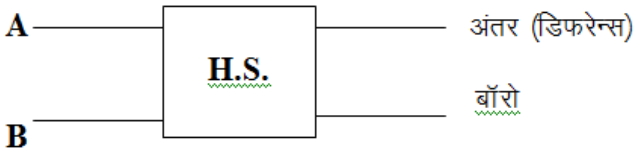


<u>फुल एडर का निर्माण हॉफ एडर द्वारा</u> – फुल एडर को दो हॉफ एडर और एक ऑर गेट के द्वारा भी बनाया जा सकता है। जो निम्नानुसार दर्शाया गया है–



<u>हॉफ सब्ट्रेक्टर :–</u> यह केवल दो बाईनरी बिट्स का एक समय में सब्ट्रेक्शन प्रदान करता है इसके दो आउटपुट होते है– दो बाईनरी बिट्स का अंतर और बॉरो।

हॉफ सब्ट्रेक्टर का ब्लॉक डायग्रामः–

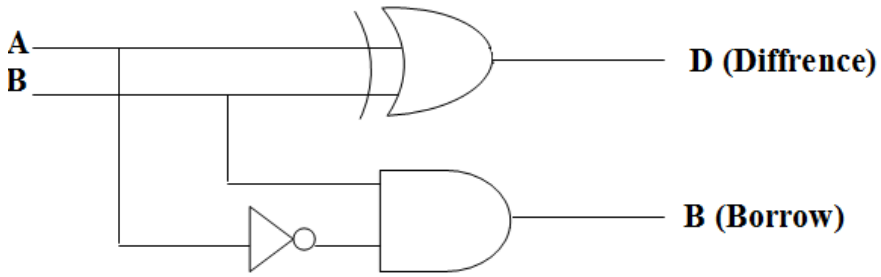


सत्य तालिका truth table :–

Inputs		B	D
A	B		
0	0	0	0
0	1	1	1

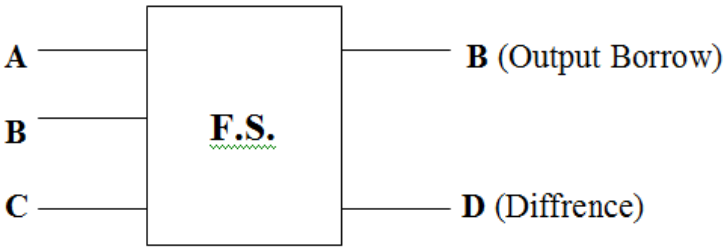
1	0	0	1
1	1	0	0

लॉजिक सर्किट :– उक्त सत्य तालिका का परीक्षण किये जाने पर ज्ञात होता है कि अंतर को XOR Gate के द्वारा प्राप्त किया जा सकता है और उधार (बॉरो) को A और B के $\overline{AND}$ ऑपरेशन से प्राप्त किया जा सकता है।

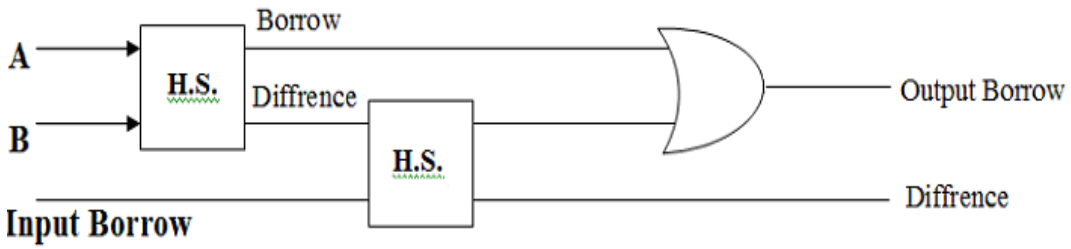


Difference = $\overline{A}B+A\overline{B}$
Borrow= $\overline{A}.B$

फुल सब्ट्रेक्टरः– इसके तीन इनपुट और दो आउटपुट अन्तर (डिफरेन्स) और उधार (बॉरो) होते है। यह एक समय पर तीन बाईनरी बिट्स पर सब्ट्रेक्शन ऑपरेशन संपादित कर सकता है जबकि हॉफ सब्ट्रेक्टर एक समय पर केवल दो बाईनरी बिट्स पर ही सब्ट्रेक्शन ऑपरेशन संपादित करता है जिसे लीस्ट सिग्निफिकेन्ट कॉलम के लिये उपयोग किया जाता है जबकी फुल सब्ट्रेक्टर को उच्च आर्डर कॉलम के लिये उपयोग कर सकते है।

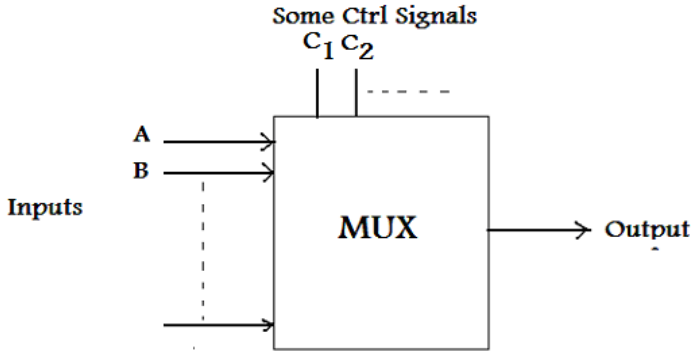


फुल सब्ट्रेक्टर को दो हॉफ सब्ट्रेक्टर और एक OR Gate की सहायता से बनाया जा सकता है।



मल्टीप्लेक्सरः– यह एक ऐसा काम्बीनेशनल लॉजिक सर्किट है जिसमें अनेक इनपुट और सिर्फ एक ही आउटपुट होता है। इसमें कुछ नियंत्रण सिग्नल होते है जिसके द्वारा ही किसी निश्चित इनपुट का चयन कर आउटपुट में भेजा जाता है। सिंगल ट्रांसमिशन लाईन का उपयोग बहुत से डिजीटल सिग्नल को ले जाने के लिये किया जाता है किन्तु एक समय में केवल एक ही सिग्नल भेजा जाता है।

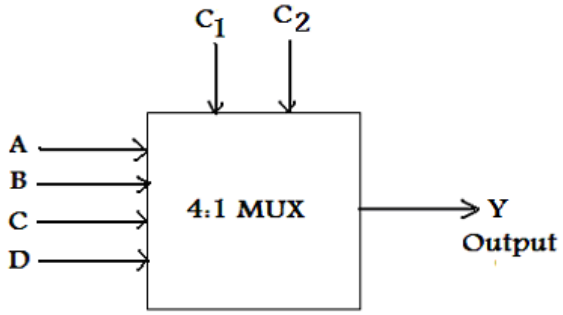
ब्लॉक डायग्रामः–



उपरोक्त ब्लॉक डायग्राम में A और B आदि में से कोई एक इनपुट से किसी एक का चयन कन्ट्रोल सिग्नल C_1, C_2, C_3 आदि के द्वारा किया जा सकता है।

4:1 मल्टीप्लेक्सर multiplexer:– इस मल्टीप्लेक्सर में 4 इनपुटस होते है और एक आउटपुट होगा तथा कन्ट्रोल सिग्नल की संख्या को इस प्रकार से निकालते है।
इनपुट की संख्या $= 4 = 2^2$ यहॉ 2 की घात कंट्रोल सिग्नल की संख्या बताती है। अर्थात 4:1 मल्टीप्लेक्सर में कन्ट्रोल सिग्नल की संख्या 2 होगी।

ब्लॉक डायग्रामः–

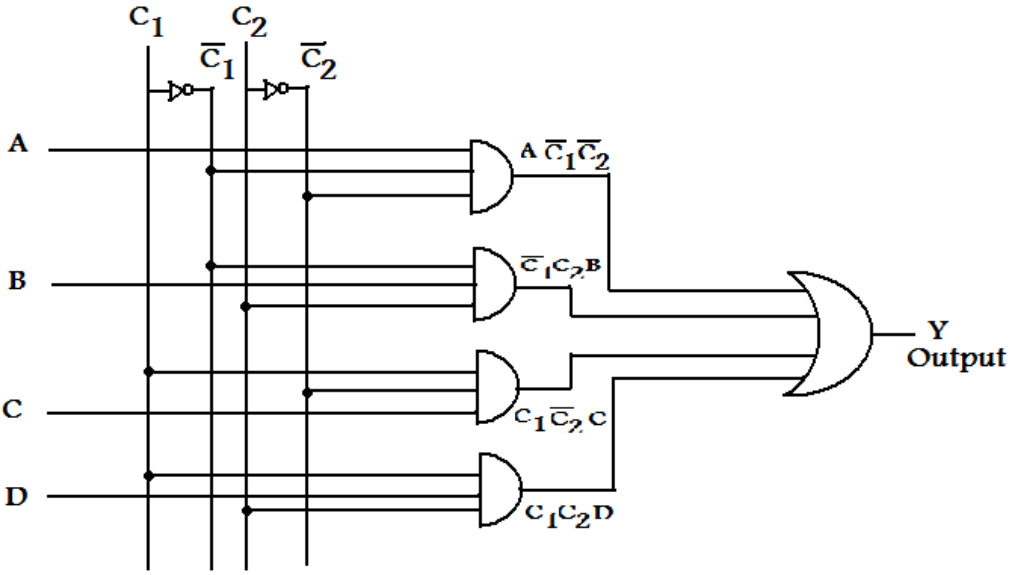


सत्य तालिकाः–

C_1	C_2	F
0	0	A
0	1	B
1	0	C
1	1	D

लॉजिक सर्किटः– उक्त सत्य तालिका से स्पष्ट है कि इसका आउटपुट निम्न होगा–

$$Y = A\,\overline{C}_1\overline{C}_2 + \overline{C}_1 C_2 B + C_1\overline{C}_2 C + C_1 C_2 D$$



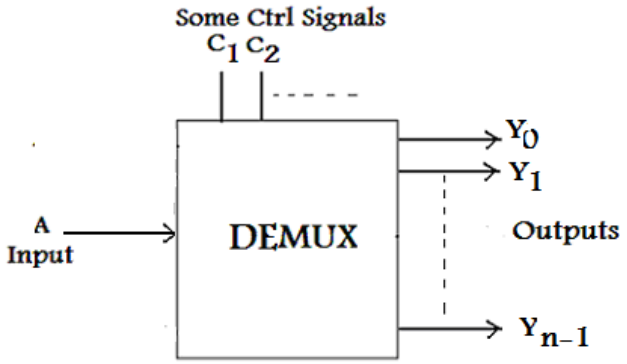
$$Y = A\overline{C}_1\overline{C}_2 + \overline{C}_1C_2B + C_1\overline{C}_2C + C_1C_2D$$

उक्त आउटपुट में सत्य तालिका के अनुसार C1 और C2 के मान रखने पर चाहा गया आउटपुट प्राप्त होता है।

यदि $C_1 = 0$ और $C_2 = 0$ होतो आउटपुट y=A होगा। और
यदि $C_1 = 0$ और $C_2 = 1$ होतो आउटपुट y=B होगा
यदि $C_1 = 1$ और $C_2 = 0$ होतो आउटपुट y=C होगा
यदि $C_1 = 1$ और $C_2 = 1$ होतो आउटपुट y=D होगा

डिमल्टीप्लेक्सर demultiplexer :– डिमल्टीप्लेक्सर, मल्टीप्लेक्सर के विपरीत कार्य करता है। जिसमें सिर्फ एक इनपुट और बहुत सारे आउटपुट तथा कुछ कन्ट्रोल सिग्नल होते है। अर्थात एक इनपुट को बहुत से आउटपुट पर कुछ नियंत्रण सिग्नल के द्वारा वितरित किया जा सकता है।

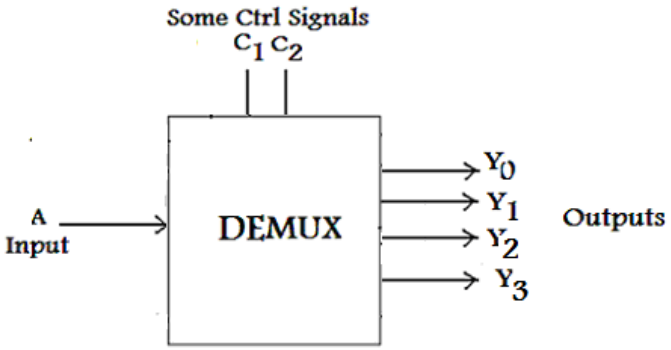
ब्लॉक डायग्रामः–



1:4 डिमल्टीप्लेक्सर :– इस डिमल्टीप्लेक्सर 1 इनपुट होता है और 4 आउटपुट होते है तथा कन्ट्रोल सिग्नल की संख्या को इस प्रकार से निकालते है।

आउटपुट की संख्या = $4 = 2^2$ यहॉ 2 की घात कंट्रोल सिग्नल की संख्या बताती है। अर्थात 1:4 डिमल्टीप्लेक्सर में कन्ट्रोल सिग्नल की संख्या 2 होगी।

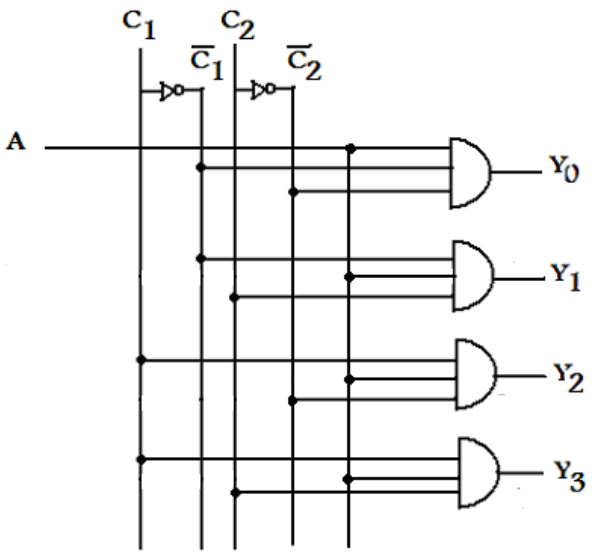
ब्लॉक डायग्रामः–



सत्य तालिकाः–

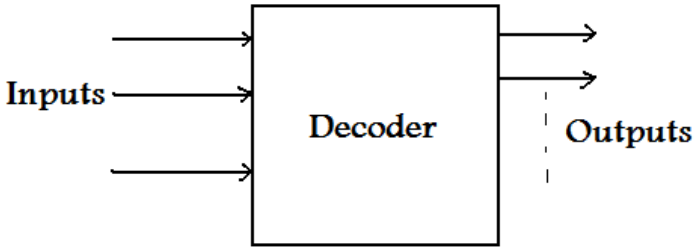
C_1	C_2	A
0	0	Y_0
0	1	Y_1
1	0	Y_2
1	1	Y_3

लॉजिक सर्किट :– चार आउटपुट को 4 एण्ड़ गेट के द्वारा तथा दो कन्ट्रोल सिग्नल और कन्ट्रोल सिग्नल के काम्प्लीमेन्ट नॉट गेट से प्राप्त कर निम्नानुसार सर्किट प्राप्त होता है।

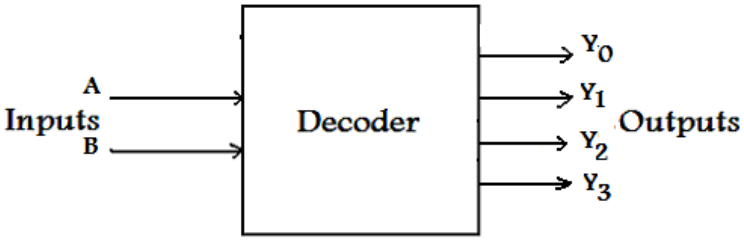


डिकोडरः– यह एक ऐसा कांबिनेशनल लॉजिक सर्किट है जिसमें कुछ इनपुट तथा बहुत सारे आउटपुट होते है। बहुत से आउटपुट में से किसी एक आउटपुट पर 1 या 0 होता है इस आउटपुट को इनपुट पर चाही गई संख्या देकर चुनते है।

ब्लॉक डायग्रामः–



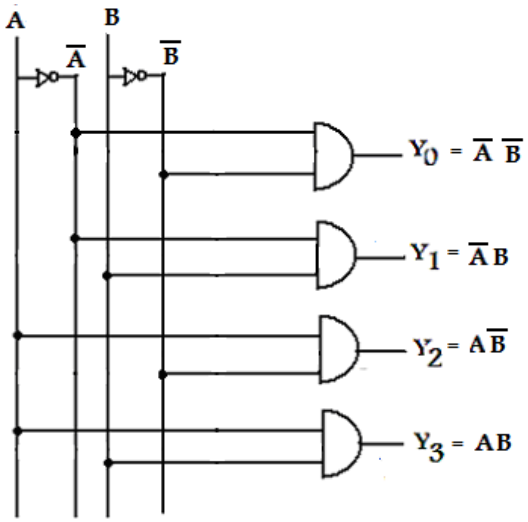
2 से 4 डिकोडरः– इसमें 2 इनपुटस होते है और 4 आउटपुट होते है।
ब्लॉक डायग्राम :–



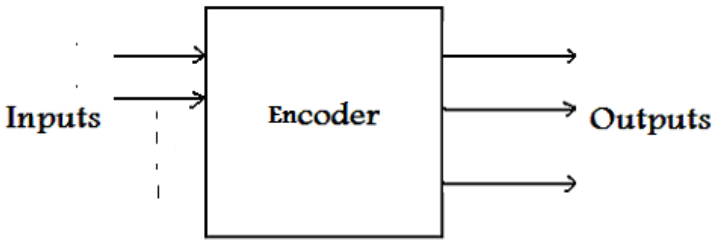
सत्य तालिकाः–

A	B	Y_0	Y_1	Y_2	Y_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

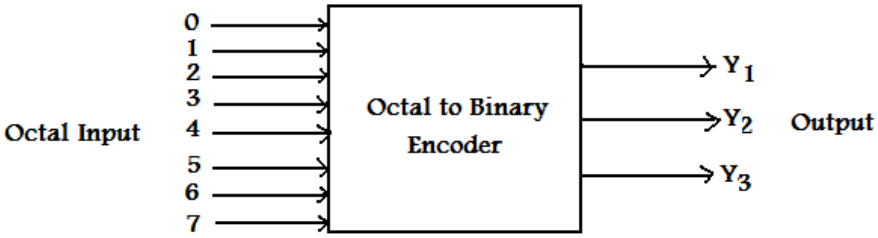
लॉजिक सर्किटः–



एन्कोडरः– यह एक ऐसा कांबिनेशनल लॉजिक सर्किट है जिसमें बहुत से इनपुट तथा कुछ ही आउटपुट होते है।



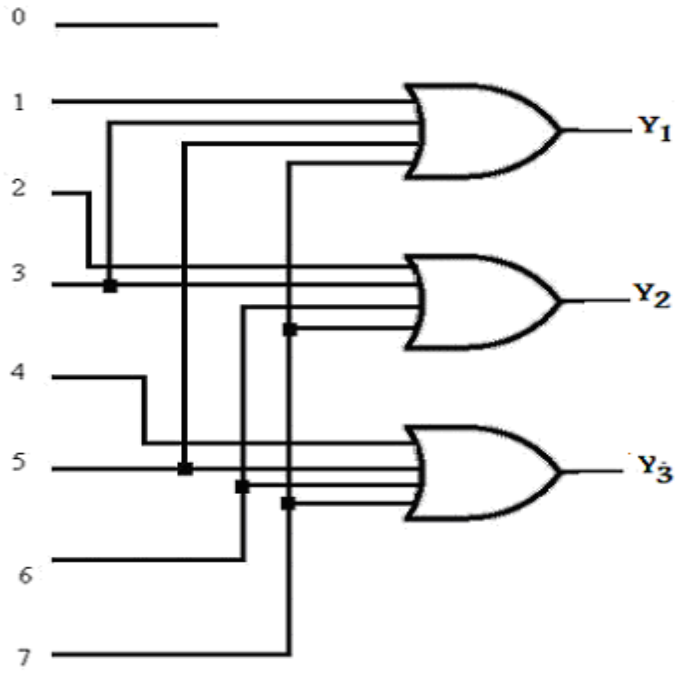
आक्टल से बाईनरी एनकोडरः– आक्टल नम्बर सिस्टम बेस 8 होता है अर्थात इस नम्बर सिस्टम को 8 विभिन्न नम्बरों 0,1,2,3,4,5,6,7 से दर्शाया जाता है। इन संख्याओं को अधिकतम तीन बाईनरी बिट से प्राप्त किया जा सकता है इसलिये इसके एनकोडर के तीन आउटपुट होते है।



सत्य तालिका :– इसकी सत्य तालिका निम्नप्रकार होगी–

Octal Digit	Y_3	Y_2	Y_1
0	0	0	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

लॉजिक सर्किटः–



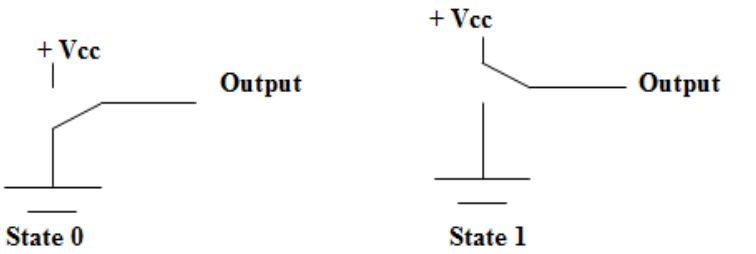
उक्त सत्य तालिका के अनुसार जब इनपुट 0 है तो आउटपुट $\mathbf{Y_3Y_2Y_1=000}$ है इसलिये उक्त तीनों OR गेट को 0 इनपुट लाईन से नहीं जोड़ा गया है। जब इनपुट 1 है तो आउटपुट $\mathbf{Y_3Y_2Y_1=001}$ है इसलिये सिर्फ $\mathbf{Y_1}$ के OR गेट को 1 से जोड़ा गया है। इसी प्रकार जब इनपुट 2 है तो आउटपुट $\mathbf{Y_3Y_2Y_1=010}$ है इसलिये सिर्फ $\mathbf{Y_2}$ के OR गेट को 1 से जोड़ा गया है। इस प्रकार जहॉ 1 चाहा गया है उसी OR गेट को जोड़ा गया है। इस प्रकार उपरोक्त लॉजिक सर्किट से सत्य तालिका सत्य सत्यापित होती है।

सिक्वेन्शल लॉजिक सर्किट

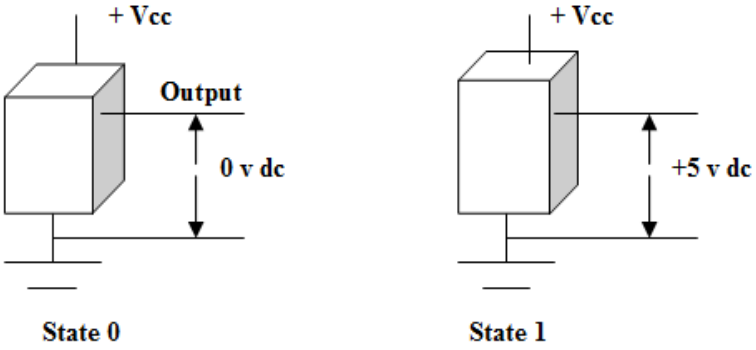
फ्लिप–फ्लॉप

फ्लिप–फ्लॉपः–किसी डिवाईस या सर्किट का जिसमें दो स्टेबल स्टेट होती है उसे बाईस्टेबल कहते है। टॉगल स्विच की दो स्टेबल स्टेट होती है। ये या तो अप होता है या डाउन, जो कि स्विच की स्थिती पर निर्भर करता है। स्विच को इस प्रकार भी कहा जा सकता है कि यह मेमोरी है। यह जब तक रहती है जब तक की स्विच की स्थिती नहीं बदलती है।

फ्लिप–फ्लॉप बाई स्टेबल इलेक्ट्रॉनिक सर्किट होता है जिसकी दो स्टेबल स्टेट होती है। इसका आउटपुट या तो लो (0) या हाई (+5 वोल्ट डी.सी.) होता है। इस प्रकार फ्लिप–फ्लॉप एक मेमोरी होती है अर्थात इसका आउटपुट तब तक सेट रहता है जब तक कि उसे परिवर्तित करने के लिये कुछ किया नहीं जावे। इस प्रकार फ्लिप–फ्लॉप और स्विच एक मेमोरी डिवाईस है। अर्थात किसी भी बाईस्टेबल डिवाईस को एक बाईनरी बिट के स्टोर के लिये उपयोग किया जा सकता है। यदि फ्लिप–फ्लॉप का आउटपुट 0 वोल्ट डीसी पर सेट है तो लॉजिकली यह 0 है जब इसका आउटपुट +5 वोल्ट डी.सी. होता है तो यह लॉजिकली 1 है। फ्लिप–फ्लॉप को लैच भी कहते है।



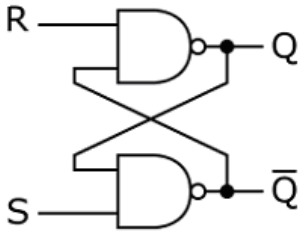
Toggle Switch



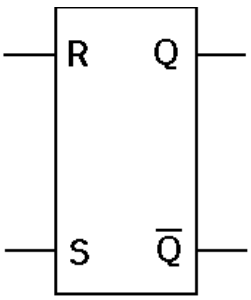
Flip-Flop

RS (रिसेट–सेट) फ्लिप– फ्लॉप:- सेट–रिसेट(RS)फ्लिप– फ्लॉप साधारण प्रकार की स्टोरेज डिवाईस है। इसे NAND या NOR Gate से बनाया जा सकता है।

लॉजिक सर्किट NAND latch के उपयोग के द्वारा :– इस लॉजिक सर्किट में दो NAND Gate का उपयोग किया गया है। एक गेट के आउटपुट को दूसरे गेट के इनपुट में एंव दूसरे गेट के आउटपुट को प्रथम गेट के इनपुट में जोड़ा गया है। इसके दो आउटपुट है Q और $\overline{Q}$ और दो इनपुट R और S है।



लॉजिक सिम्बोल ऑफ RS फ्लिप– फ्लॉपः–

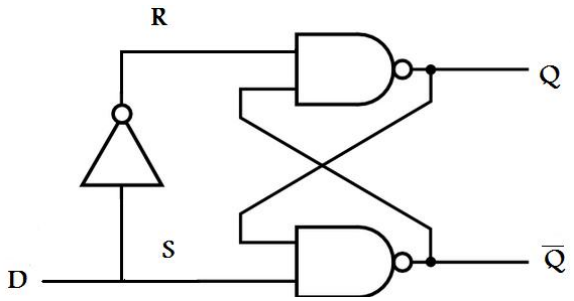


सत्य तालिकाः–

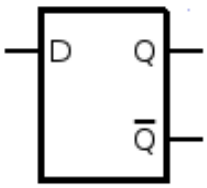
R	S	Q
0	0	Racing
0	1	1
1	0	0
1	1	Last State

उक्त सत्य तालिका से स्पष्ट है कि यदि $R = 0$ व $S = 0$ तो रेसिंग अवस्था प्राप्त होती है अर्थात अस्थिर अवस्था प्राप्त होती है। यदि $R = 1$ व $S = 1$ है तो फ्लिप– फ्लॉप लॉस्ट स्टेट में ही रहता है। यदि $R = 0$ व $S = 1$ है तो फ्लिप– फ्लॉप का आउटपुट $Q = 1$ होता है जिसे सेट अवस्था भी कहते है। और यदि $R = 1$ व $S = 0$ है तो फ्लिप– फ्लॉप का आउटपुट $Q = 0$ होता है जिसे रीसेट अवस्था भी कहते है।

D (डाटा) फ्लिप– फ्लॉपः– आर.एस. फ्लिप– फ्लॉप के दो डाटा इनपुट होते है आर व एस यदि फ्लिप– फ्लॉप में हाई बिट स्टोर करना हो तो हाई एस की आवश्यकता होती है और लो बिट को स्टोर किये जाने के लिये हाई आर की आवश्यकता होती है। इस फ्लिप– फ्लॉप में यदि आर और एस दोनों हाई होतो रेसिंग कंडिशन प्राप्त होती है। इस स्थिती को दूर किये जाने के लिये डी फ्लिप – फ्लॉप का उपयोग किया जाता है। इसमें सिंगल डाटा इनपुट का उपयोग किया जाता है। अर्थात यह एक ही इनपुट वाला फ्लिप– फ्लॉप होता है। इसे आर.एस. फ्लिप– फ्लॉप से निम्न प्रकार से बनाया जा सकता है–



लॉजिक सिम्बोल ऑफ D फ्लिप– फ्लॉपः–



सत्य तालिकाः–

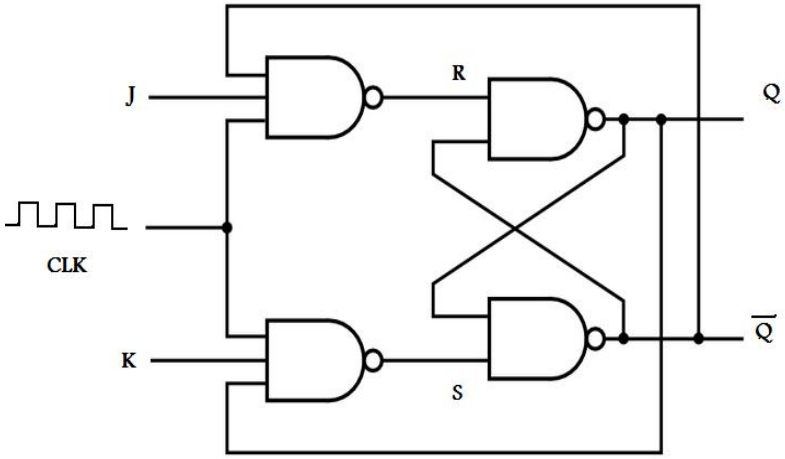
D	Q
0	0
1	1

उक्त सत्य तालिका से स्पष्ट है कि यदि D=0 तो Q=0 एंव यदि D=1 तो Q=1 प्राप्त होता है। इस प्रकार इस फ्लिप – फ्लॉप में कोई रेसिंग कंडीशन नहीं होती है।

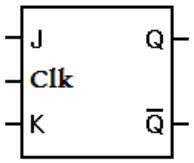
JK फ्लिप– फ्लॉपः– इस फ्लिप – फ्लॉप में दो इनपुट J और K होते है। निम्न प्रकार से JK फ्लिप– फ्लॉप का उपयोग किया जा सकता है।

1. लेवल क्लॉक्ड़ JK फ्लिप– फ्लॉप
2. ऐजट्रिगर्ड JK फ्लिप– फ्लॉप
3. मास्टर स्लेव JK फ्लिप– फ्लॉप

1- लेवल क्लॉक्ड़ JK फ्लिप– फ्लॉप :–



उक्त लेवल क्लॉक्ड़ JK फ्लिप – फ्लॉप का लॉजिक सर्किट दर्शाया गया है। इस सर्किट में फीड़बैक का उपयोग किया गया है। अर्थात $\overline{Q}$ का उपर के गेट से और Q को नीचे के गेट से जोड़ते है।
लॉजिक सिम्बोल ऑफ लेवल क्लॉक्ड़ JK फ्लिप– फ्लॉप :–



सत्य तालिकाः–

CLK	J	K	Q
0	X	X	पूर्व अवस्था
1	0	0	पूर्व अवस्था
1	0	1	0 (रिसेट)
1	1	0	1 (सेट)
1	1	1	टॉगल अवस्था

उक्त सत्य तालिका के **प्रथम परिणाम** से स्पष्ट है कि यदि क्लॉक लो होती है और K तथा L के कोई भी मान हो तो Q अपनी पूर्व अवस्था में ही रहता है। क्योकि इस स्थिती में प्रारंभ के दोनों NAND गेट का आउटपुट 1 होता है अर्थात R और S दोनों के मान 1 होगें।

सत्य तालिका के द्वितीय परिणाम को में क्लॉक हाई है तो J तथा K के मान लो है तो भी Q अपनी पूर्व अवस्था में ही रहता है क्योकि इस स्थिती में भी R तथा S दोनों के मान 1 होगें।

सत्य तालिका के तृतीय परिणाम में यदि क्लॉक हाई हो तथा J=0 हो तथा K=1 होतो Q=0 रीसेट होगा। क्योंकि इस स्थिती में R=1 होगा एंव S का मान Q पर निर्भर करेगा। यदि Q=0 हो तो S=1 हो जाएगा। अतः R=1 और S=1 होगा इस स्थिती में फ्लिप – फ्लॉप अपनी पूर्व की स्थिती में ही रहेगा। अर्थात Q=0 ही मिलेगा। किन्तु यदि पहले Q=1 हुआ तो S=0 हो जाएगा। अतः R=1 और S=0 अर्थात रीसेट की स्थिती प्राप्त होगी। इस कारण Q=0 हो जाएगा। अर्थात यदि उक्त फ्लिप– फ्लॉप की पूर्व की अवस्था कुछ भी हो यदि J=0 तथा K=1 एंव क्लॉक हाई हो तो Q=0 ही प्राप्त होगा।

सत्य तालिका के चतुर्थ परिणाम का अवलोकन करने पर अर्थात यदि क्लॉक हाई हो तथा J=1 एंव K=0 होतो Q=1 अर्थात फ्लिप– फ्लॉप सेट हो जाएगा। क्योंकि इस स्थिती में S=1 होगा एंव R का मान Q पर निर्भर करेगा। यदि Q=0 होतो $R=\overline{1}$ हो जाएगा। अतः R=1 और S=1 होगा इस स्थिती में फ्लिप– फ्लॉप अपनी पूर्व की स्थिती में ही रहेगा। अर्थात Q=0 ही होगा एंव $\overline{Q}$=1 होगा। किन्तु यदि पहले Q=1 हुआ तो R=0 हो जाएगा। अतः R=0 और S=1 अर्थात सेट की स्थिती प्राप्त होगी एंव Q=1 हो जाएगा। अर्थात यदि उक्त फ्लिप– फ्लॉप की पूर्व की अवस्था कुछ भी हो यदि J=1 तथा K=0 एंव क्लॉक हाई हो तो Q=1 ही प्राप्त होगा अर्थात फ्लिप– फ्लॉप सेट रहेगा।

सत्य तालिका के पॉचवे परिणाम यदि क्लॉक हाई हो एंव J=1 हो तथा K=1 होतो इस स्थिती में R व S के मान Q और $\overline{Q}$ पर निर्भर करेंगे यदि फ्लिप– फ्लॉप कि पूर्व की अवस्था Q=0 हुई तो अगली अवस्था Q=1 प्राप्त होगी। यदि पूर्व में Q=1 हो तो फ्लिप– फ्लॉप की नई अवस्था Q=0 होगी। इस स्थिती को टॉगल अवस्था (अपनी स्थिती को बदलना) कहते है।

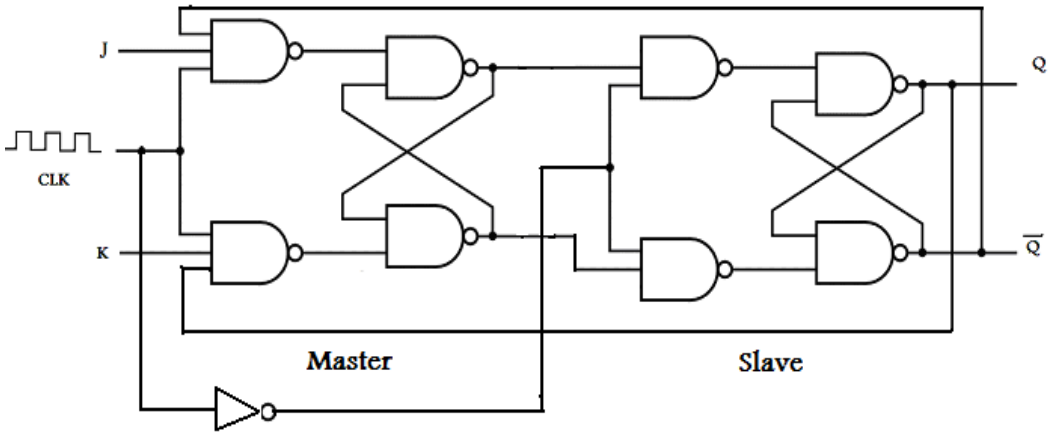
रेस अराउण्ड प्राब्लमः– यह समस्या उपरोक्त लेवल क्लॉक्ड़ JK फ्लिप– फ्लॉप में होती है। उक्त लेवल क्लॉक्ड़ JK फ्लिप– फ्लॉप की सत्य तालिका के अध्ययन से ज्ञात होता है कि यदि क्लॉक हाई हो एंव J=1 एंव K=1 होतो टॉगल अवस्था प्राप्त होती है। अर्थात क्लॉक हाई मिलते ही इस लेवल क्लॉक्ड़ JK फ्लिप– फ्लॉप का आउटपुट Q बदल जाता है। यदि पल्स की चौड़ाई डिवाईस के प्रोपेगेशन डिले से अधिक हो तो आउटपुट एक से अधिक बार टॉगल हो जाएगा, इस स्थिती में फ्लिप– फ्लॉप के सही मान का पता नहीं लगाया जा सकता है। इस प्रकार लेवल क्लॉक्ड़ फ्लिप– फ्लॉप का एक ही पल्स में एक से अधिक बार टॉगल होने की समस्या को रेस अराउन्ड़ समस्या कहते है।

रेस अराउन्ड प्रॉब्लम को दूर किये जाना – यह समस्या पल्स की चौड़ाई प्रोपेगेशन डिले से अधिक होने के कारण होती है। इस समस्या को मास्टर–स्लेव JK फ्लिप– फ्लॉप के उपयोग कर दूर किया जा सकता है।

मास्टर–स्लेव JK फ्लिप– फ्लॉप :–

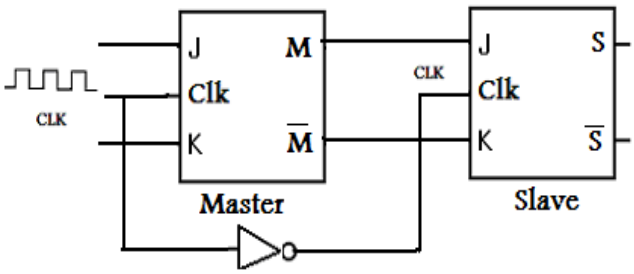
लॉजिक सर्किट डायग्राम – यह दो फ्लिप– फ्लॉप मास्टर और स्लेव के संयोजन से बनाया जाता है। जो कार्य मास्टर करता है स्लेव उसे कॉपी अर्थात दोहराता है।

लेवल क्लॉकड़ मास्टर स्लेव फ्लिप– फ्लॉप :–



यह दो फ्लिप– फ्लॉप मास्टर और स्लेव के संयोजन से बनाया जाता है। जो कार्य मास्टर करता है स्लेव उसे दोहराता है।

लेवल क्लॉकड़ मास्टर स्लेव फ्लिप– फ्लॉप का लॉजिक सिम्बोलः–

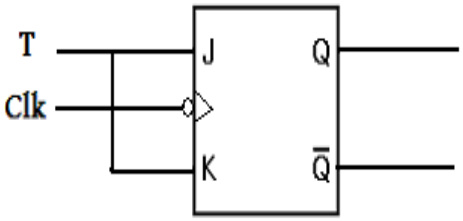


उच्च वोल्टेज पर अर्थात पॉजिटिव क्लॉक पर मास्टर कार्य करता है तथा निम्न वोल्टेज निगेटिव क्लॉक पर स्लेव कार्य करता है। जब क्लॉक हाई होती है तो मास्टर अपनी अवस्था परिवर्तित कर सकता है। किन्तु उस समय स्लेव कोई कार्य नहीं करता है। किन्तु क्लॉक लो होने पर स्लेव वही कार्य करता है जो पूर्व में मास्टर ने किया था। एंव मास्टर लो क्लॉक पर कुछ नहीं करता है। इस कारण रेस अराउन्ड समस्या अर्थात एक ही पल्स पर एक से अधिक बार टॉगल होने की समस्या समाप्त हो जाती है।

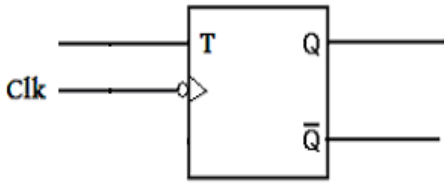
सत्य तालिकाः–

CLK	J	K	Q
1	X	X	पूर्व अवस्था
0	0	0	पूर्व अवस्था
0	0	1	0 (रिसेट)
0	1	0	1 (सेट)
0	1	1	टॉगल अवस्था

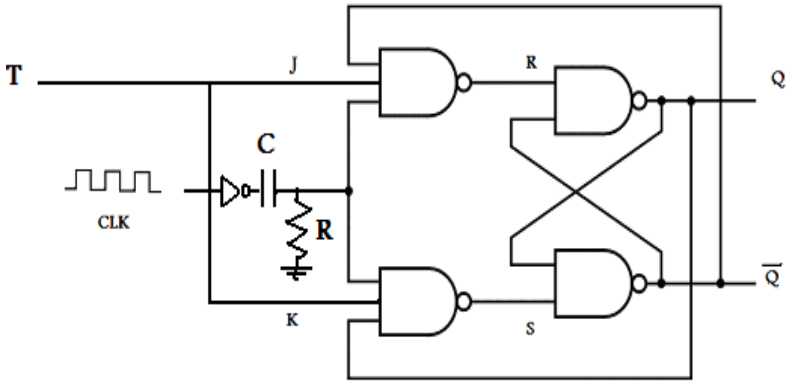
T(टॉगल) फ्लिप– फ्लॉप :– यदि JK फ्लिप– फ्लॉप में दोनों इनपुट को समान अर्थात कॉमन कर एक ही इनपुट दिया जावे जिसे **T** कहते है तो **T** फ्लिप– फ्लॉप बनता है।



उक्त लॉजिक सर्किट से स्पष्ट है कि इस फ्लिप– फ्लॉप में केवल दो ही इनपुट एक क्लॉक और दूसरा **T** होता है। इसे निम्न प्रकार से भी बनाया जा सकता है।



लॉजिक सर्किटः–



सत्य तालिकाः–

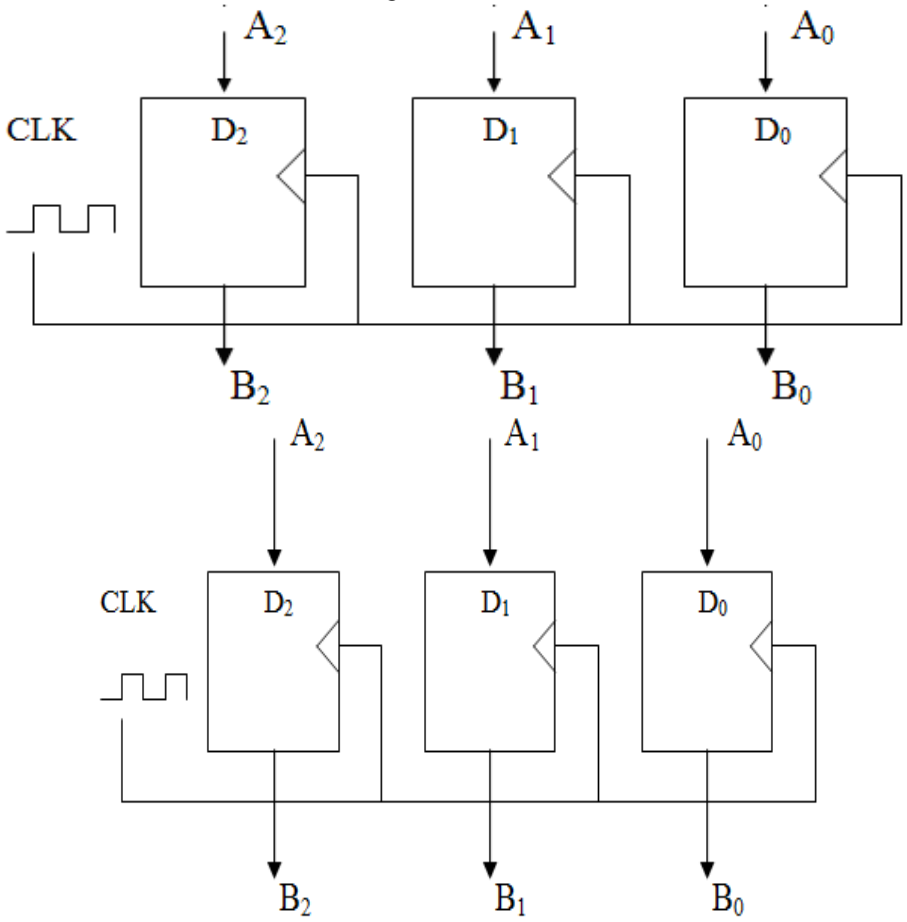
CLK	T	Q
0	X	पूर्व अवस्था
1	X	पूर्व अवस्था
↑	X	पूर्व अवस्था
↓	0	पूर्व अवस्था
↓	1	टॉगल अवस्था

रजिस्टरः– रजिस्टर साधारणतः फ्लिप– फ्लॉप का समूह होता है। जिसका उपयोग बाईनरी नम्बर को स्टोर किये जाने के लिये किया जाता है। फ्लिप– फ्लॉप 1 बिट की डिजिटल इंफार्मेशन को स्टोर कर सकते है। अतः फ्लिप– फ्लॉप को 1 बिट रजिंस्टर भी कहा जा सकता है। बाईनरी सूचनाओं को स्टोर किये जाने के लिये फ्लिप– फ्लॉप श्रंखला समूह (अरे) की आवश्यकता होगी। बाईनरी वर्ड में बिट्स की संख्या जितनी होगी उतने फ्लिप– फ्लॉप की (एक बाईनरी बिट के लिये एक फ्लिप– फ्लॉप) आवश्यकता होती है। उसे रजिस्टर कहते है।

डाटा को सीरियल अर्थात एक समय में एक बिट या पैरेलल फार्म अर्थात सभी बिट्स को एक साथ प्रवेशित या पुनः प्राप्त किया जा सकता है। इस आधार पर रजिस्टर को मुख्यतः दो भागों में बॉटा जा सकता है–

1. बफर रजिस्टर
2. शिफ्ट रजिस्टर

बफर रजिस्टरः– इस रजिस्टर में जो संख्या इनपुट पर दी जाती है वही इसमें लोड़ होती है।



बाहय इनपुट A_0, A_1, A_2 फ्लिप– फ्लॉप इनपुटों पर लगाए गये है एवं एक ही क्लॉक सारे फ्लिप– फ्लॉप को दिया जाता है। क्लॉक की पॉजेटीव एज पर सारे इनपुट फ्लिप– फ्लॉप पर लोड़ हो जाते है। क्लॉक की धनात्मक ऐज पर निम्न आउटपुट प्राप्त होता है।

$A_2A_1A_0 = B_2B_1B_0$

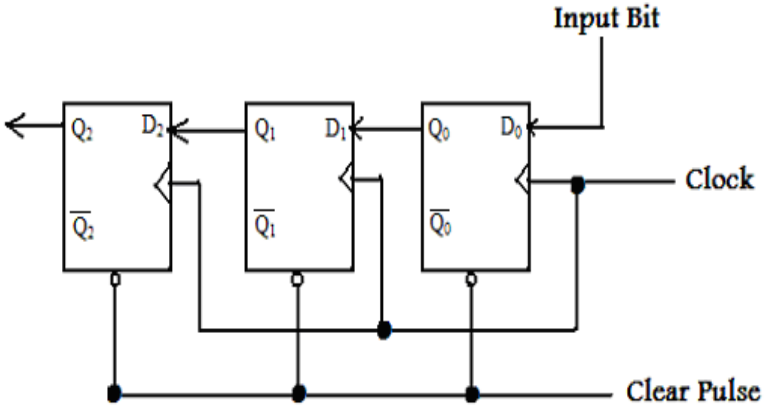
शिफ्ट रजिस्टरः–

इस प्रकार के रजिस्टर में पहले से स्टोर बाइनरी संख्या हर क्लॉक पल्स पर एक बिट लेफ्ट या राईट में शिफ्ट हो जाती है। शिफ्ट होने की दिशा के आधार पर दो प्रकार के शिफ्ट रजिस्टर होते है।

1. लेफ्ट शिफ्ट रजिस्टर
2. राईट शिफ्ट रजिस्टर

<u>लेफ्ट शिफ्ट रजिस्टरः–</u> इस प्रकार के रजिस्टर में प्रत्येक क्लॉक पल्स के पश्चात बाइनरी संख्या एक बिट लेफ्ट में शिफ्ट हो जाती है।

लॉजिक सर्किटः–



दाहिने साईड के फ्लिप– फ्लॉप के आउटपुट को अगले फ्लिप– फ्लॉप के इनपुट पर फीड़ किया गया है। इनपुट बिट को दाहिने फ्लिप– फ्लॉप पर दिया जाता है। प्रारंभ में क्लियर पल्स देकर सारे फ्लिप– फ्लॉप के आउटपुट को 0 कर दिया जाता है। अर्थात क्लियर पल्स दिये जाने के पश्चात फ्लिप– फ्लॉप का आउटपुट 000 होगा। यदि बाईनरी इनपुट बिट 1 होतो क्लॉक की धनात्मक ऐज पर प्रथम फ्लिप– फ्लॉप पर पहुॅच जावेगा। एंव $Q_2Q_1Q_0 = 001$ प्राप्त होगा। यदि बाईनरी इनपुट को लगातार 1 रखा जाता है और द्वितीय तथा तृतीय क्लॉक पल्स आती है तो आउटपुट निम्नानुसार होगें।

द्वितीय क्लॉक पल्स पर $Q_2Q_1Q_0 = 011$

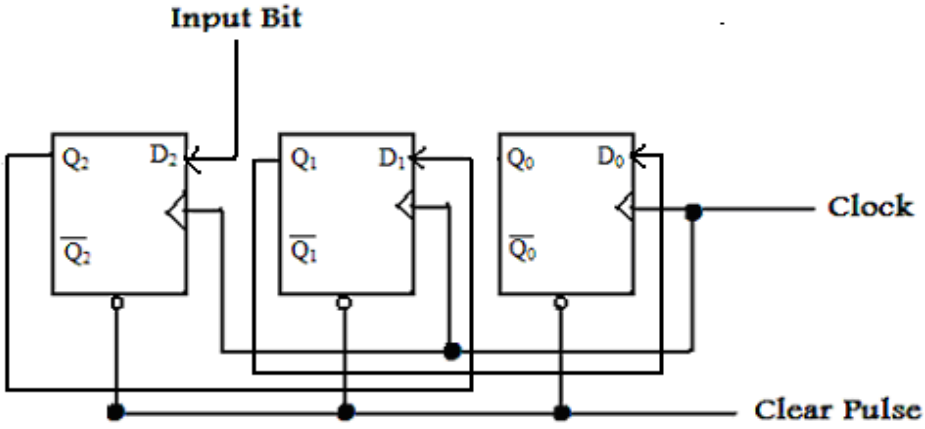
तृतीय क्लॉक पल्स पर $Q_2Q_1Q_0 = 111$

अब प्रत्येक क्लॉक पल्स पर यही आउटपुट प्राप्त होगा जब तक इनपुट पर 0नही दिया जाता है या स्टार्ट पल्स क्लियर नहीं की जाती है।

उक्त परिणाम से स्पष्ट है कि बाईनरी इनपुट बिट बायें शिफ्ट हो रही है। इसलिये इसे लेफ्ट शिफ्ट रजिस्टर कहते है।

राईट शिफ्ट रजिस्टरः– इस प्रकार के रजिस्टर में प्रत्येक क्लॉक पल्स के पश्चात बाइनरी संख्या एक बिट राईट अर्थात दाहिने साईड़ शिफ्ट हो जाती है।

लॉजिक सर्किटः–



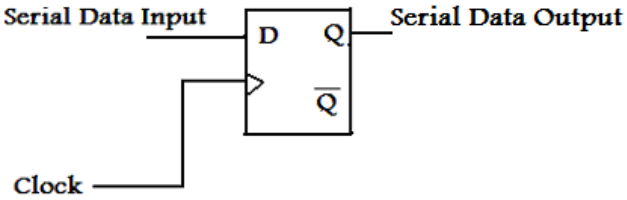
सबसे बाऐं साईड के फ्लिप– फ्लॉप के आउटपुट को दाहिने साईड़ के अगले फ्लिप– फ्लॉप के इनपुट पर फीड़ किया गया है। इनपुट बिट को बाऐं फ्लिप– फ्लॉप पर दिया जाता है। प्रारंभ में क्लियर पल्स देकर सारे फ्लिप– फ्लॉप के आउटपुट को 0 कर दिया जाता है। अर्थात क्लियर पल्स दिये जाने के पश्चात फ्लिप– फ्लॉप का आउटपुट 000 होगा। यदि बाईनरी इनपुट बिट 1 हो तो क्लॉक की धनात्मक ऐज पर प्रथम फ्लिप– फ्लॉप पर पहुॅच जावेगा। एंव $Q_2Q_1Q_0 = 100$ प्राप्त होगा। यदि बाईनरी इनपुट को लगातार 1 रखा जाता है और द्वितीय तथा तृतीय क्लॉक पल्स आती है तो आउटपुट निम्नानुसार होगें।

द्वितीय क्लॉक पल्स पर $Q_2Q_1Q_0 = 110$
तृतीय क्लॉक पल्स पर $Q_2Q_1Q_0 = 111$
अब प्रत्येक क्लॉक पल्स पर यही आउटपुट प्राप्त होगा जब तक इनपुट पर 0 नही दिया जाता है या स्टार्ट पल्स क्लियर नहीं की जाती है।
उक्त परिणाम से स्पष्ट है कि बाईनरी इनपुट बिट दाग्रे शिफ्ट हो रही है। इसलिये इसे राईट शिफ्ट रजिस्टर कहते है।
शिफ्ट रजिस्टरों का दूसरा वर्गीकरण :– डाटा को रजिस्टर में दो प्रकार या तो सीरियल प्रकार से या पेरेलल प्रकार से शिफ्ट किया जा सकता है।एंव इसी तरह दो प्रकार से डाटा को आउट किया जा सकता है। अर्थात इनपुट देने के प्रकार व आउटपुट लेने के प्रकार के आधार पर चार प्रकार के शिफ्ट रजिस्टर होते है।

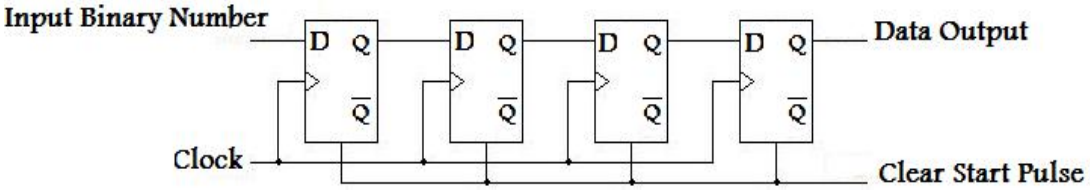
1. सीरियल इन सीरियल आउट
2. सीरियल इन पेरेलल आउट
3. पेरेलल इन सीरियल आउट
4. पेरेलल इन पेरेलल आउट रजिस्टर
5.

1. सीरियल–इन–सीरियल आउट रजिस्टर
इसमें इनपुट डाटा एक एक बिट करके दिया जाता है और आउटपुट डाटा भी एक एक बिट करके प्राप्त किया जाता है। इस ब्लॉक डायग्राम निन्नानुसार है।

इस रजिस्टर को बनाने के लिये ऐज ट्रिगर्ड JK या D फ्लिप– फ्लॉप का उपयोग किया जा सकता है। JK फ्लिप– फ्लॉप में दो इनपुट का उपयोग किया जाता है जबकी D फ्लिप– फ्लॉप में केवल एक इनपुट D होने से फ्लिप– फ्लॉप के मध्य कनेक्शनस की संख्या कम हो जाती है।
निम्न लॉजिक सर्किट डायग्राम में एक D फ्लिप– फ्लॉप का उपयोग कर **एक बिट सीरियल–इन–सीरियल आउट रजिस्टर** दर्शाया गया है। जो बिट इनपुट D पर उपस्थित होती है वहीं बिट क्लाक की धनात्मक ऐज पर आउटपुट Q पर प्रदर्शित होती है।

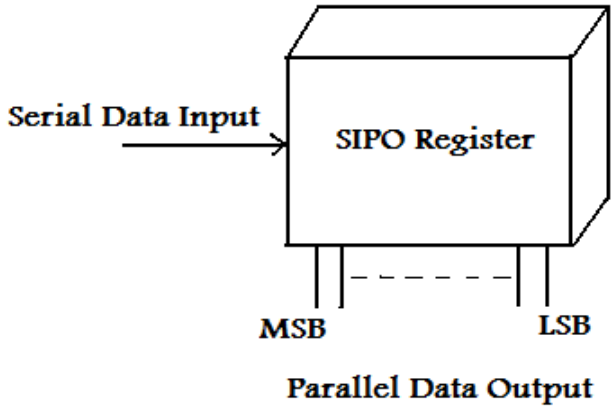


इसी प्रकार **4 बिट सीरियल–इन–सीरियल आउट रजिस्टर** का लॉजिक सर्किट निम्न प्रकार होगा–

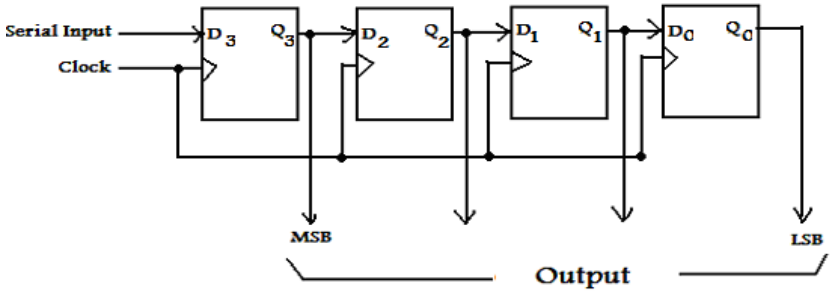


उक्त सर्किट से स्पष्ट है कि जो भी बिट इनपुट पर होगी उसे आउटपुट पर चार क्लॉक पल्स के पश्चात पहुँचेगी।
सिरियल इन पेरेलल आउट रजिस्टरः– इस रजिस्टर में इनपुट डाटा को सीरियल फैशन में शिफ्ट किया जाता है जबकि आउटपुट डाटा को पैरेलल फार्म में प्राप्त किया जाता है। इसे बनाने के लिये प्रत्येक फ्लिप–

फ्लॉप के आउटपुट को आउटपुट पिन से जोड़ दिया जाता है। इसका ब्लॉक डायग्राम निम्नानुसार बनाया जा सकता है।



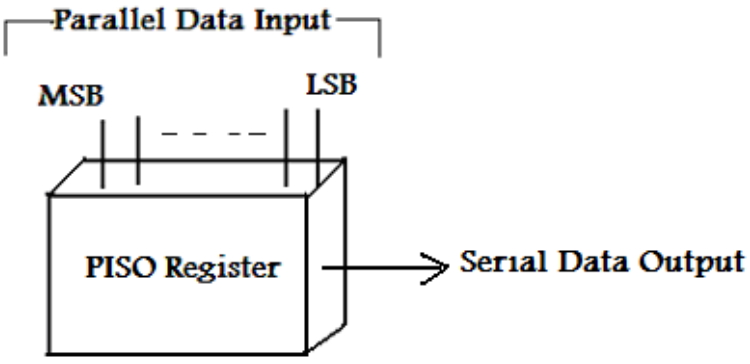
4 बिट सीरियल–इन–पेरेलल आउट रजिस्टर का लॉजिक सर्किट निम्नानुसार बनाया जा सकता है।



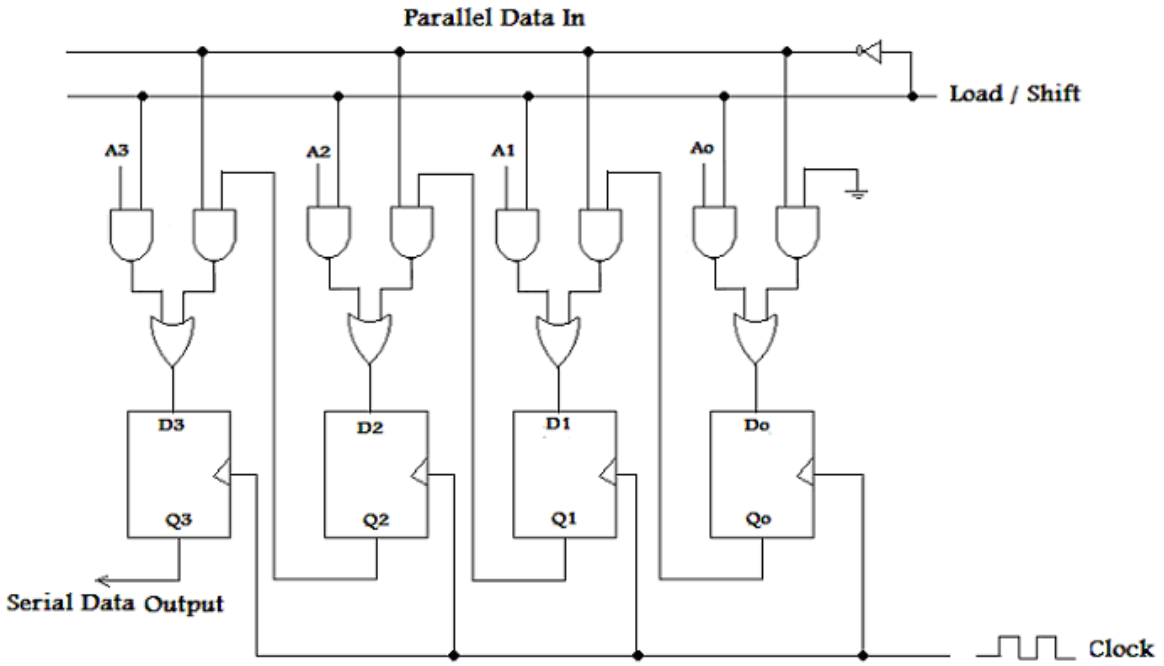
सीरियल इनपुट बिट को D_3 पर दिया जाता है इसके आउटपुट Q_3 को अगले फ्लिप – फ्लॉप के इनपुट D_2 पर फीड़ किया जाता है। शेष कनेक्शन भी इसी प्रकार किये जाते है। चुँकि उक्त लॉजिक सर्किट में धनात्मक ऐज ट्रिगर्ड डी फ्लिप– फ्लॉप का उपयोग किया गया है अतः प्रत्येक क्लॉक की धनात्मक ऐज एक फ्लिप– फ्लॉप का आउटपुट अगले फ्लिप– फ्लॉप पर शिफ्ट हो जावेगा। तथा आउटपुट को समान्तर में प्राप्त किया जाता है।

<u>पैरेलल इन सीरियल आउट रजिस्टरः–</u>

इस रजिस्टर में डाटा को समान्तर फीड़ किया जाता है तथा सीरियल फैशन में प्राप्त किया जाता है। इसका ब्लॉक डायग्राम निम्नानुसार होगा।

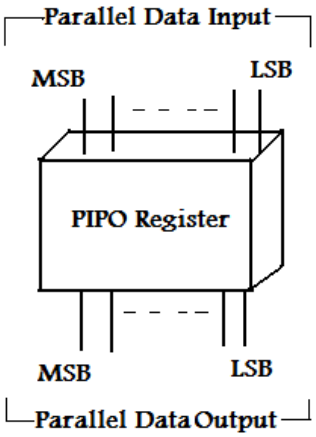


4 बिट पेरेलल–इन–सीरियल आउट रजिस्टर का लॉजिक सर्किट निम्नानुसार होगा।



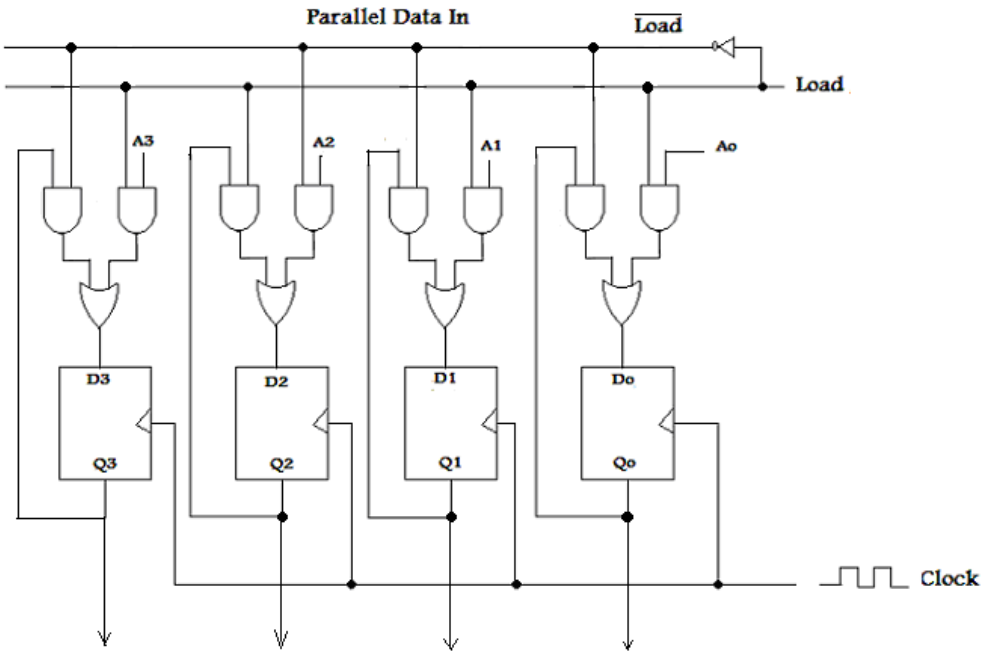
Load = इनपुट दिये जाकर बाईनरी इनपुट संख्या A3A2A1A0 को Q3Q2Q1Q0 पर लोड़ किया जाता है। इसके पश्चात Load =0 करके प्रत्येक क्लॉक पर इसे Q3 पर एक–एक करके प्राप्त किया जाता है।

<u>पैरेलल इन पेरेलल आउट रजिस्टरः–</u> इस रजिस्टर में इनपुट को पेरेलल में फीड़ किया जाता है तथा आउटपुट भी पेरेलल में प्राप्त किया जाता है।



4 बिट पेरेलल–इन–पेरेलल आउट रजिस्टर का लॉजिक सर्किट निम्नानुसार होगा।

निम्न सर्किट डायग्राम अनुसार यदि Load=0 होगा तो फ्लिप– फ्लॉप की पुरानी अवस्था अर्थात Q ही मिलेगा। बाईनरी इनपुट संख्या A3A2A1A0 को Load=1 देकर फ्लिप– फ्लॉप पर लोड़ किया जाता है। एंव निम्न सर्किट डायग्राम अनुसार आउटपुट पेरेलल में प्राप्त किया जाता है। यह बफर रजिस्टर के समान होता है।



काउन्टर :– काउन्टर डिजिटल सिस्टम में उपयोग में आने वाला बहुत उपयोगी सब सिस्टम है जो कि क्लॉक पल्स द्वारा संचालित होता है। जिसका उपयोग क्लॉक सायकल को काउन्ट किये जाने के लिये किया जाता है। काउन्टर का उपयोग किसी इंस्ट्रुमेन्ट में समय और पीरियड़ के मापन या फ्रिक्वेंसी के मापन के लिये किया जा सकता है।

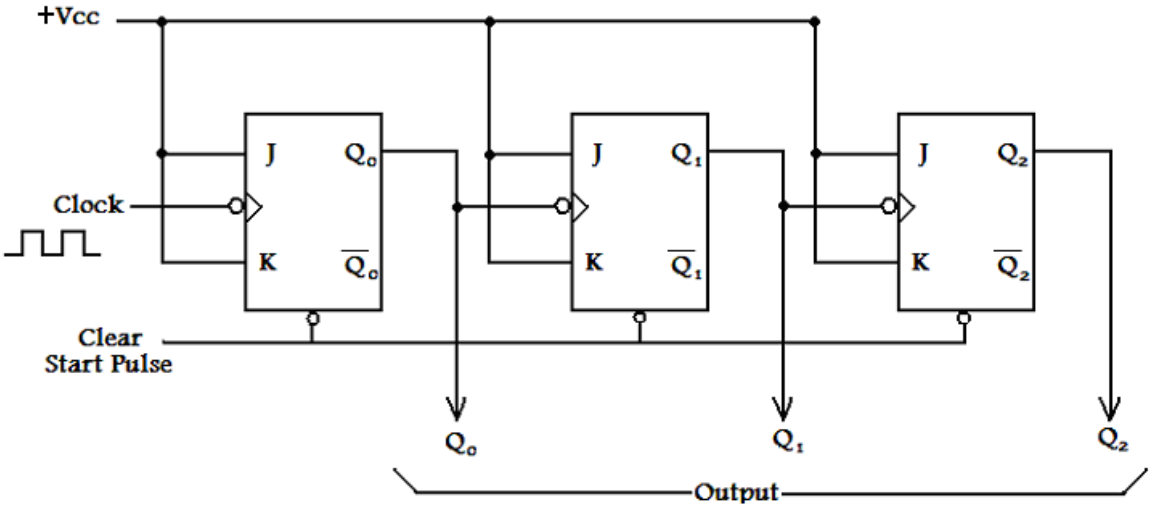
काउन्टर आधारभूत रूप से दो प्रकार होते है।

1. एसिन्क्रोनस काउंटर
2. सिंन्क्रोनस काउंटर

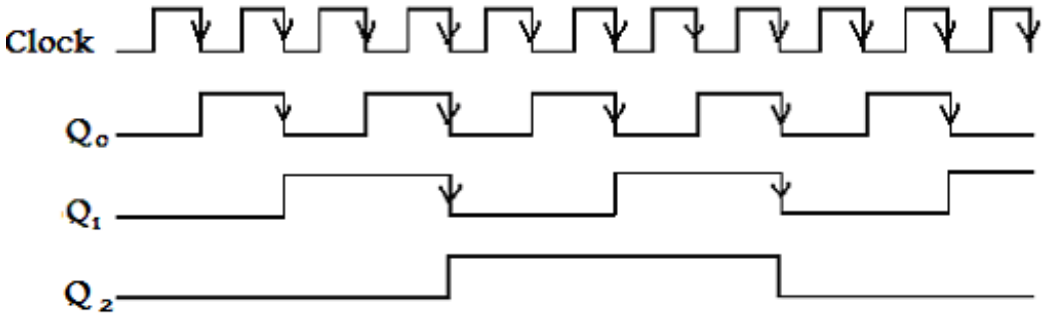
1. **एसिंक्रोनस काउन्टर (रिपल काउन्टर) :–** बाईनरी रिपल काउन्टर क्लॉकड़ JK फ्लिप– फ्लॉप से बनाया जा सकता है। तथा प्रत्येक JK फ्लिप– फ्लॉप को J=1 तथा K=1 देकर टॉगल मोड में कार्य करवाया जाता है। सिर्फ एक फ्लिप– फ्लॉप को क्लॉक दी जाती है। इसके पश्चात इस JK फ्लिप– फ्लॉप के आउटपुट को अगले JK फ्लिप– फ्लॉप के क्लॉक के इनपुट पर फीड़ किया जाता है। इस काउन्टर को एसिंक्रोनस काउन्टर या रिपल काउन्टर कहते है।

तीन बिट बाईनरी रिपल अप काउन्टर या मोड–8 एसिन्क्रोनस रिपल अप काउन्टर :– यह अप काउन्टर है इस कारण क्लियर स्र्टाट पल्स देकर सभी फ्लिप– फ्लॉप के आउटपुट को 000 कर दिया जाता है। प्रथम JK फ्लिप– फ्लॉप क्लॉक के इनपुट J=1 तथा K=1 पर क्लॉक की ऋणात्मक ऐज पर प्रथम फ्लिप– फ्लॉप टॉगल करता है। तथा प्रथम फ्लिप– फ्लॉप का आउटपुट द्वितीय फ्लिप– फ्लॉप को टॉगल करता है इसी प्रकार द्वितीय फ्लिप– फ्लॉप का आउटपुट तृतीय फ्लिप– फ्लॉप को टॉगल करता है। इस प्रकार ट्रिगर एक फ्लिप– फ्लॉप से दूसरे फ्लिप– फ्लॉप की ओर पानी की लहर (रिपल) की तरह चलती है। इस कारण इसे रिपल काउन्टर भी कहते है। इस कारण सम्पूर्ण सर्किट का प्रोपेगेशन डिले प्रत्येक एक के प्रोपेगेशन डिले, के योग के बराबर होता है। अर्थात एक फ्लिप– फ्लॉप का प्रोपेगेशन डिले 5 नेनो सेकण्ड है तो इस कारण तीन फ्लिप–फ्लॉप के काउन्टर का सम्पूर्ण प्रोपेगेशन डिले 15 नेनो सेकण्ड़ होगा। इस प्रकार लिस्ट सिगनिफिकेन्ट बिट Q_0 होती है और मोस्ट सिगनिफिकेन्ट बिट Q_2 होती है। क्लियर स्र्टाट पल्स दिये जाने के बाद काउन्टर का आउटपुट Q_2 Q_1 Q_0 =000 होगा। उसके पश्चात प्रत्येक क्लॉक पल्स पर निम्न टाईमिंग डायग्राम के अनुसार फ्लिप– फ्लॉप का आउटपुट बदलने पर काउन्टर का आउटपुट बदलता है।

लॉजिक सर्किटः–



टाईमिंग डायग्राम :–



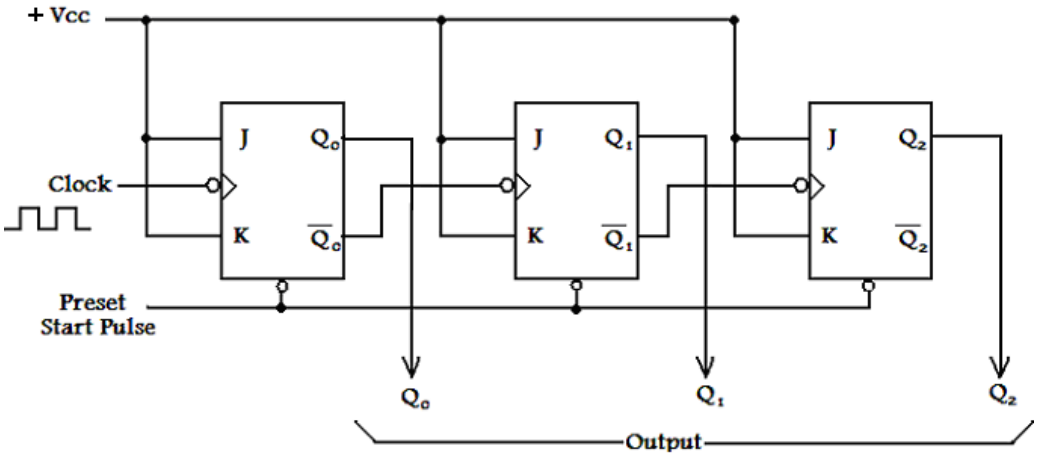
काउन्ट की अवस्था :–

Pulse No.	Q_2	Q_1	Q_0	State of Count
0	0	0	0	0
1	0	0	1	1
2	0	1	0	2
3	0	1	1	3
4	1	0	0	4
5	1	0	1	5
6	1	1	0	6
7	1	1	1	7
8	0	0	0	0

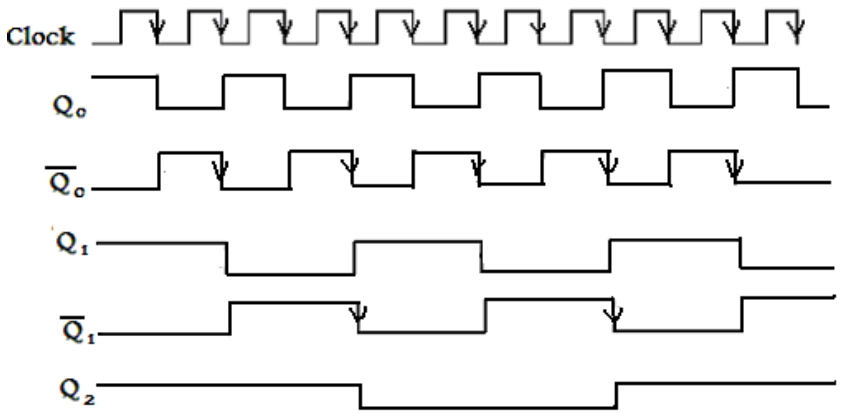
तीन बिट बाईनरी रिपल डाउन काउन्टर या मोड–8 एसिंक्रोनस रिपल डाउन काउन्टर :– यह एक डाउन काउन्टर है। इस डाउन काउन्टर में अप काउन्टर की तुलना में इतना परिवर्तन होता है कि, स्टार्ट पल्स प्रिसेट पर दी जाती है तथा क्लॉक पर काम्प्लीमेन्ट (समपूरक) आउटपुट को जोड़ेगें। प्रिसेट स्टार्ट पल्स देकर

सभी फ्लिप– फ्लॉप के आउटपुट को 111 कर दिया जाता है । प्रथम JK फ्लिप– फ्लॉप क्लॉक के इनपुट J=1 तथा K=1 पर क्लॉक की ऋणात्मक ऐज पर प्रथम फ्लिप– फ्लॉप टॉगल करता है। तथा प्रथम फ्लिप– फ्लॉप का काम्प्लीमेन्ट आउटपुट, द्वितीय फ्लिप – फ्लॉप को टॉगल करता है इसी प्रकार द्वितीय फ्लिप– फ्लॉप का काम्प्लीमेन्ट आउटपुट तृतीय फ्लिप– फ्लॉप को टॉगल करता है। इस प्रकार लीस्ट सिगनिफिकेंट बिट Q_0 होती है और मोस्ट सिगनिफिकेन्ट बिट Q_2 होती है। प्रीसेट स्टार्ट पल्स दिये जाने के बाद काउन्टर का आउटपुट $Q_2\ Q_1\ Q_0 = 111$ होगा। उसके पश्चात प्रत्येक क्लॉक पल्स पर निम्न टाईमिंग डायग्राम के अनुसार फ्लिप– फ्लॉप का आउटपुट बदलने पर काउन्टर का आउटपुट बदलता है।

लॉजिक सर्किटः–



टाईमिंग डायग्राम :–



काउन्ट की अवस्था :–

Pulse No.	Q_2	Q_1	Q_0	State of Count
0	1	1	1	7

1	1	1	0	6
2	1	0	1	5
3	1	0	0	4
4	0	1	1	3
5	0	1	0	2
6	0	0	1	1
7	0	0	0	0
8	1	1	1	7

सिंक्रोनस काउन्टर या पेरेलल काउन्टरः– इस काउन्टर में एक ही क्लॉक को पैरेलल में सभी फ्लिप– फ्लॉप के क्लॉक इनपुट पर जोड़ दिया जाता है। क्लॉक के समान्तर में फीड़ किये जाने के कारण सभी फ्लिप– फ्लॉप एक साथ टॉगल करते है। इस कारण प्रोपेगेशन डिले कम हो जाता है और कार्य करने की गति बढ़ जाती है।

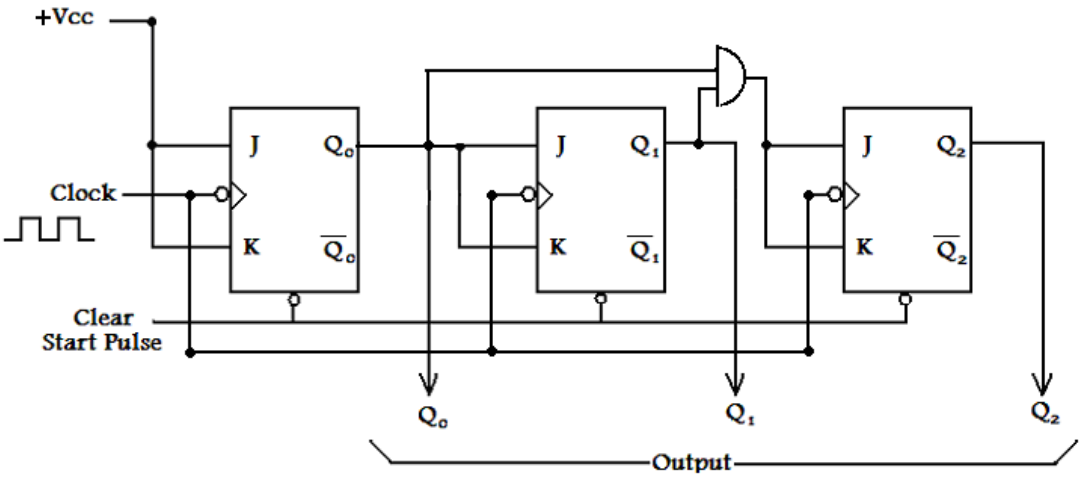
तीन बिट बाईनरी सिन्क्रोनस अप काउन्टर या मोड–8 सिन्क्रोनस अप काउन्टर :– यह एक अप काउन्टर है इस कारण क्लियर स्र्टाट पल्स देकर सभी फ्लिप– फ्लॉप के आउटपुट को 000 कर दिया जाता है।

काउन्ट की अवस्था :–

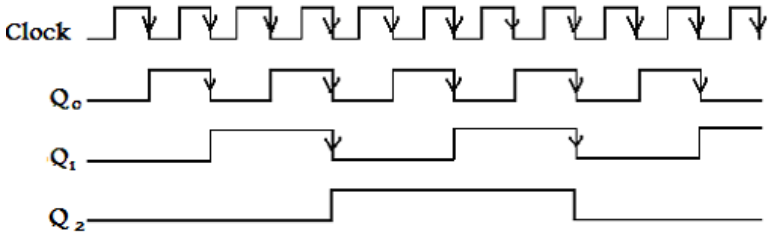
Pulse No.	Q_2	Q_1	Q_0	State of Count
0	0	0	0	0
1	0	0	1	1
2	0	1	0	2
3	0	1	1	3
4	1	0	0	4
5	1	0	1	5
6	1	1	0	6
7	1	1	1	7
8	0	0	0	0

उक्त अवस्था तालिका से स्पष्ट है कि $\mathbf{Q_0}$ प्रत्येक क्लॉक पल्स पर टॉगल करता है तथा $\mathbf{Q_1}$ तब टॉगल करता है $\mathbf{Q_0 = 1}$ हो। इसलिये $\mathbf{Q_0}$ के दोनों इनपुट J तथा K पर हाई वोल्टेज अर्थात 1 दिया जाता है और $\mathbf{Q_1}$ फ्लिप– फ्लॉप के इनपुट J और K को फ्लिप– फ्लॉप $\mathbf{Q_0}$ से जोड़ देते है। एंव उक्त सत्य तालिका से स्पष्ट है कि $\mathbf{Q_2}$ तब ही टॉगल करता है जबकि $\mathbf{Q_0}$ और $\mathbf{Q_1}$ दोनों ही 1 हो। इसलिये $\mathbf{Q_0}$ और $\mathbf{Q_1}$ दोनों के ही आउटपुट को एक एण्ड़ गेट पर दिया जाकर एण्ड़ गेट के आउटपुट को तृतीय फ्लिप– फ्लॉप के दोनों इनपुट J और K से जोड़ दिया जाता है। क्लियर स्र्टाट पल्स दिये जाने के बाद काउन्टर का आउटपुट $Q_2\ Q_1\ Q_0 = 000$ होगा। निम्न टाईमिंग डायग्राम के अनुसार फ्लिप– फ्लॉप का आउटपुट बदलने पर काउन्टर का आउटपुट बदलता है।

लॉजिक सर्किटः–



टाईमिंग डायग्राम :–



तीन बिट बाईनरी सिन्क्रोनस डाउन काउन्टर या मोड–8 सिन्क्रोनस डाउन काउन्टर :– यह एक डाउन काउन्टर है इस कारण प्रीसेट स्टार्ट पल्स देकर सभी फ्लिप– फ्लॉप के आउटपुट को 111 कर दिया जाता है ।

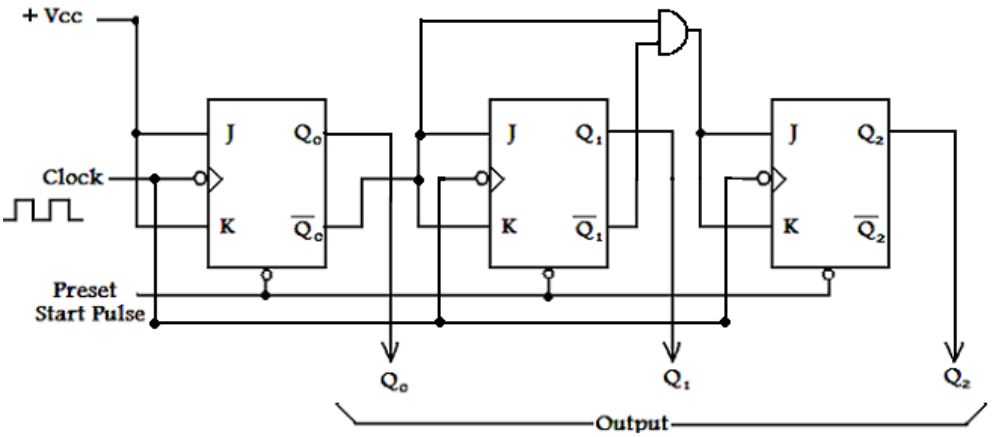
काउन्ट की अवस्था :–

Pulse No.	Q_2	Q_1	Q_0	State of Count
0	1	1	1	7
1	1	1	0	6
2	1	0	1	5
3	1	0	0	4
4	0	1	1	3
5	0	1	0	2
6	0	0	1	1
7	0	0	0	0
8	1	1	1	7

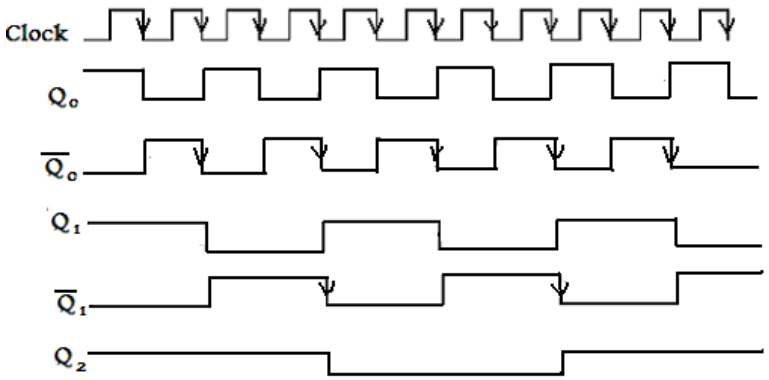
उक्त अवस्था तालिका से स्पष्ट है कि Q_0 प्रत्येक क्लॉक पल्स पर टॉगल करता है तथा Q_1 तब टॉगल करता है $Q_0 = 0$ हो अर्थात $\overline{Q_0} = 1$ हो। इसलिये Q_0 के दोनों इनपुट J तथा K पर हाई वोल्टेज अर्थात 1 दिया जाता है और Q_1 फ्लिप– फ्लॉप के इनपुट J और K को फ्लिप– फ्लॉप $\overline{Q_0}$ से जोड़ देते है। एंव उक्त सत्य तालिका से स्पष्ट है कि Q_2 तब ही टॉगल करता है जबकि Q_0 और Q_1 दोनों ही 0 हो। इसलिये Q_0 और Q_1 दोनों के ही आउटपुट को एक एण्ड़ गेट पर दिया जाकर एण्ड़ गेट के आउटपुट को तृतीय फ्लिप– फ्लॉप के दोनों इनपुट J और K से जोड़ दिया जाता है। प्रिसेट स्टार्ट पल्स दिये जाने के बाद काउन्टर का

आउटपुट Q_2 Q_1 Q_0 =111 होगा। निम्न टाईमिंग डायग्राम के अनुसार फ्लिप–फ्लॉप का आउटपुट बदलने पर काउन्टर का आउटपुट बदलता है।

लॉजिक सर्किटः–



टाईमिंग डायग्राम :–



रिंग काउन्टरः– इस काउन्टर में एक बिट 0 या 1 घड़ी की दिशा में या इसके विपरित दिशा में घुमती है। प्रत्येक क्लॉक पल्स पर यह बिट बायें या दाऐं शिफ्ट हो जाती है। यदि घुमने वाली बिट यदि 0 है तो शेष बिट 1 होगी यदि घुमने वाली बिट यदि 1 है तो शेष बिट 0 होगी। अर्थात यह एक विशेष प्रकार के शिफ्ट रजिस्टर होते है।

रिंग काउन्टर (लेफ्ट शिफ्ट का उपयोग कर) :– रिंग काउन्टर अर्थात एक बिट जिसे रोटेट करवाना है यदि वह 1 है तो एक फ्लिप– फ्लॉप पर $\overline{PR}$ शेष सारे फ्लिप– फ्लॉप पर $\overline{CLR}$ स्टॉट पल्स देते है। जैसे ही रीसैट पल्स देते है सबसे दाहिने फ्लिप– फ्लॉप पर 1 तथा शेष सभी फ्लिप– फ्लॉप पर 0 हो जाता है।

प्रथम अवस्था यह होगी

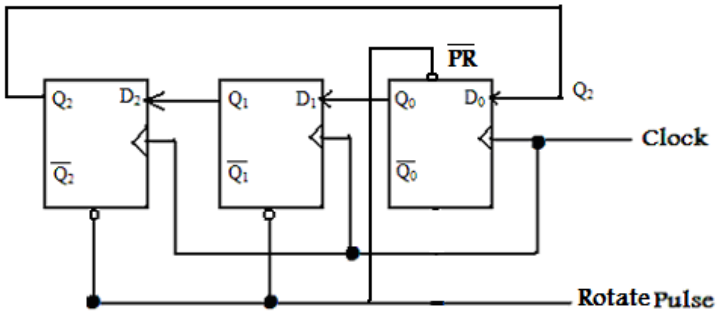
Q_2	Q_1	Q_0
0	0	1

अब जैसे जैसे प्रथम, द्वितीय, तृतीय क्लाक पल्स आवेगी निम्नानुसार अवस्थाऐं प्राप्त होगी।

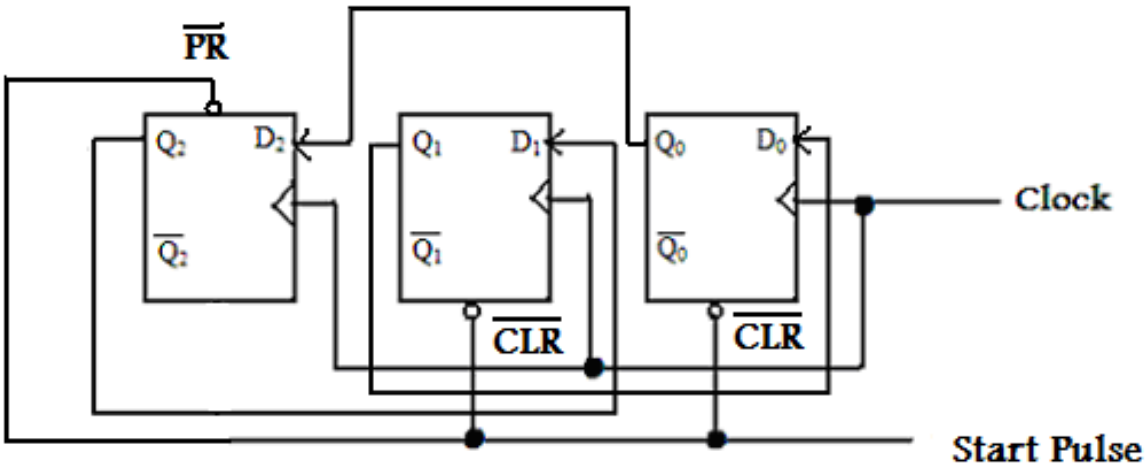
0	1	0
1	0	0
0	0	1

अर्थात एक बिट 1 घड़ी की दिशा में घूर्णन करती है। इसी प्रकार यदि 0 बिट को घुमाना हो तो अंतिम फ्लिप– फ्लॉप पर $\overline{CLR}$ स्टॉट पल्स देते है और शेष सभी फ्लिप– फ्लॉप पर $\overline{PR}$ पल्स देते है। तो आउटपुट निम्न प्रकार से होगा।

1 1 0
1 0 1
0 1 1
1 1 0



रिंग काउन्टर (राईट शिफ्ट का उपयोग कर) :– रिंग काउन्टर अर्थात एक बिट जिसे रोटेट करवाना है यदि वह 1 है तो एक फ्लिप– फ्लॉप पर $\overline{PR}$ शेष सारे फ्लिप– फ्लॉप पर $\overline{CLR}$ स्टॉट पल्स देते है। जैसे ही रोटेट पल्स देते है सबसे दाहिने फ्लिप– फ्लॉप पर 1 तथा शेष सभी फ्लिप– फ्लॉप पर 0 हो जाता है।



प्रथम अवस्था यह होगी

Q_2	Q_1	Q_0
1	0	0

अब जैसे जैसे प्रथम, द्वितीय, तृतीय पल्स आवेगी निम्नानुसार अवस्थाऐं प्राप्त होगी।

0 1 0
0 0 1
1 0 0

अतः 1 घड़ी के विपरित दिशा में घुमता है।

इस प्रकार यदि 0 को घड़ी के विपरित दिशा में घुमाना हो तो सबसे बायें फ्लिप– फ्लॉप पर $\overline{CLR}$ तथा शेष फ्लिप– फ्लॉप पर $\overline{PR}$ देने पर

प्रथम अवस्था 0 1 1 होगी

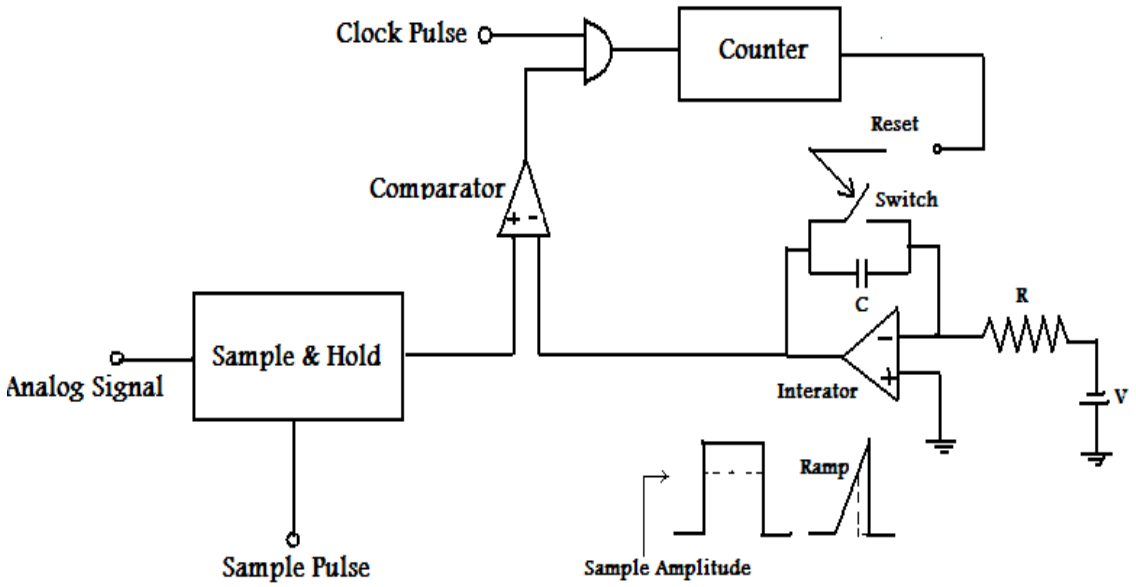
अब जैसे जैसे प्रथम, द्वितीय, तृतीय पल्स आवेगी निम्नानुसार अवस्थाऐं प्राप्त होगी।

1 0 1

1 1 0
0 1 1

ऐनालॉग से डिजिटल कनवर्टर

ऐनालॉग से डिजिटल कनवर्टर :– काउन्टर क्लॉक पल्स को काउन्ट करता किन्तु उस अंतराल के लिये जो कि ऐनालॉग सेम्पल के एम्प्लीट्यूड़ के समानुपाती होता है। इस प्रकार ऐनालॉग सेम्पल का एम्प्लीट्यूड़ काउन्टर की काउन्टिंग के अनुसार क्वान्टाईज्ड़ हो जाता है अर्थात डिजिटल आउटपुट प्राप्त होता है।



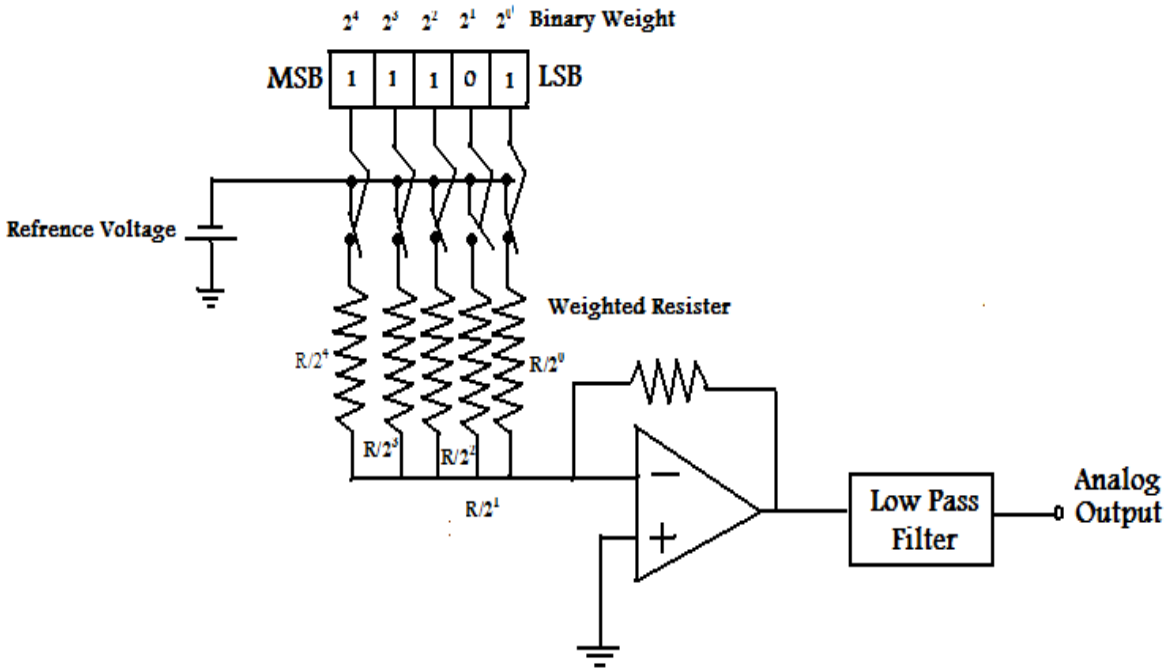
सबसे पहले काउन्टर को क्लियर पल्स देकर शुन्य कर दिया जाता है। कैपेसेटर C के एक्रास लगा स्विच को इंटीग्रेटर के शुन्य पर सेट किये जाने के लिये खोला और बंद किया जाता है। अब इंटीग्रेटर का आउटपुट रैखिक रूप से समय के साथ बढता है अर्थात रैम्प प्राप्त होता है। किन्तु जब तक यह सेम्पल एम्प्लीट्यूड़ से कम होता है तो कम्परेटर का आउटपुट हाई होता है इस कारण से AND गेट से क्लॉक पल्स काउन्टर को जाती रहती है और काउन्टर उसे काउन्ट करता है। किन्तु जैसे ही रैम्प अर्थात इंटीग्रेटर का आउटपुट सेम्पल एम्प्लीट्यूड़ से अधिक हो जाता है तो कम्परेटर का आउटपुट लो हो जाता है और AND गेट डिसेबल हो जाता है उस समय काउन्टर काउन्टिंग करना बंद कर देता है। इस प्रकार काउन्टर में संग्रहित काउन्ट, रैखिक रूप से गेट के ऑन रहने के समय अर्थात सेम्पल एम्प्लीट्यूड के समानुपाती होते है। अर्थात काउन्टर की काउन्ट या रिडिंग सेम्पल एम्प्लीट्यूड़ का डिजिटल रिप्रेजेन्टेशन होती है।

डिजिटल से ऐनालॉग कनवर्टर :– यह डिजिटल इनपुट को ऐनालॉग आउटपुट में बदलता है। यह दो प्रकार के होते है।

1. वेटेड रजिस्टर डिजिटल से ऐनालॉग कनवर्टर
2. R-2R लेडर डिजिटल से ऐनालॉग कनवर्टर

वेटेड रजिस्टर डिजिटल से ऐनालॉग कनवर्टर :– इसमें रजिस्टेंस का मान बाईनरी वेट के तुल्य होता है। प्रत्येक बाईनरी बिट के लिये एक स्विच होता है जो कि यदि बाईनरी बिट 1 होती है तो ऑन होता है अन्यथा 0 बिट के लिये ऑफ होता है। इन स्विच के सिरिज में वेटेड़ रजिस्टर कनेक्ट होते है। एक रिफ्रेन्स वोल्टेज लाईन भी कनेक्ट होती है। इस प्रकार यदि किसी बिट का मान 1 है तो स्विच ऑन हो जाता है और वह रिफ्रेन्स वोल्टेज लाईन से कनेक्ट हो जाता है। इस प्रकार डिजिटल इनपुट का वह बाईनरी नम्बर स्विच के द्वारा उस वेटेड़ रजिस्टर के रिफ्रेन्स वोल्टेज से कनेक्ट होने पर करन्ट बहती है। इस करन्ट को OP-AMP द्वारा वोल्टेज में परिवर्तित कर दिया जाता है। यह आउटपुट वोल्टेज बाईनरी नम्बर के समानुपाती

होता है। नम्बर बदलने पर आउटपुट वोल्टेज का मान भी बदलता है और जिसे लो पास फिल्टर से पास किया जाकर स्मूथ कर लिया जाता है। और ऐनॉलाग सिग्नल प्राप्त होता हे।



मइक्रोप्रोसेसर

माइक्रोप्रोसेसर :–माइक्रोप्रोसेसर बहुउद्देशीय, प्रोग्रामेबल, क्लॉक द्वारा संचालित, रजिस्टर पर आधारित इलेक्ट्रॉनिक डिवाईस होती है जो कि स्टोरेज डिवाईस, तथा इनपुट डिवाईस से प्राप्त बाईनरी डाटा को रीड़ करता है और इंस्ट्रक्शनस के अनुसार प्रोसेस कर आउटपुट के रूप में इंफार्मेशन प्रोवाईड करता है। इस प्रकार प्रोग्रामेबल मशीन के मुख्यतः तीन कम्पोनेन्ट्स को माइक्रोप्रोसेसर, मेमोरी, और इनपुट/आउटपुट डिवाईस से रिप्रेजेन्ट किया जा सकता है। माइक्रोप्रोसेसर, माइक्रोकम्प्यूटर की सेन्ट्रल प्रोसेसिंग युनिट है। यह माइक्रो कम्प्यूटर का हदय होता है। माइक्रोप्रोसेसर के चार प्राथमिक ऑपरेशनस होते है।

1. मेमोरी को रीड़ करना– मेमोरी से डाटा या इंस्ट्रक्शनस को रीड़ करना।
2. मेमोरी को राइट करना– डाटा या इंस्ट्रक्शनस को मेमोरी में राईट करना।
3. इनपुट/आउटपुट को रीड़ करना–इनपुट डिवाइस से डाटा को प्राप्त करना।
4. इनपुट/आउटपुट राईट– आउटपुट डिवाईस को डाटा भेजना।

यह सभी ऑपरेशनस माइक्रोप्रोसेसर और अन्य पेरिफेरल डिवाईस सह मेमोरी के बीच संचार प्रकिया के भाग है। पेरीफेरल डिवाइस या मेमोरी लोकेशन्स से संचार के लिये माइक्रोप्रोसेसर को निम्न स्टेपस को परफार्म किया जाता है।

1. पेरिफेरल डिवाईस को या मेमोरी लोकेशन को ऐड्रेसेस के द्वारा आइडेन्टिफाई करना।
2. बाईनरी इंफार्मेशन डाटा या इंस्ट्रक्शनस को स्थानांतरित करना।
3. टाईमिंग या सिन्क्रोनाईजिंग सिग्नलस प्रोवाईड़ करना।

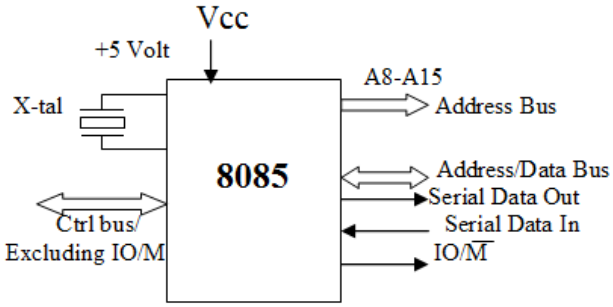
8085 माईक्रोप्रोसेसर उक्त फंक्शनस को परफार्म किये जाने के लिये तीन कम्यूनिकेशनस लाईन जिसे बस कहते है का उपयोग करता है। यह बसेस है डाटा बस, एड्रेस बस, और कन्ट्रोल बस।

इंटेल 8085:–इंटेल 8085 माइक्रोप्रोसेसर 8 बिट माइक्रोप्रोसेसर है। यह सिंगल लार्ज स्केल इंटीग्रेशन, 40 पिन की आई.सी. पैकेज है। इंटेल 8085 माइक्रोप्रोसेसर के ऑपरेशन के लिये सिंगल +5 वोल्ट डी.सी. की आवश्यकता होती है। इसकी क्लॉक स्पीड़ लगभग 3 मेगा हर्टज एवं क्लॉक सायकल 320 नैनो सेकंड़, एवं क्लॉक साईकल का समय 200 नैनो सेकंड़ होता है। इसमें 80 बेसिक इंस्ट्रक्शनस एवं 246 ऑपरेषन कोड्स होते है।

इंटेल 8085 माइक्रोप्रोसेसर के मुख्यतः तीन भाग होते है। अर्थमेटिक लॉजिक युनिट, टाईमिंग एवं कन्ट्रोल यूनिट, और रेजिस्टर्स।

इंटेल 8085 माइक्रोप्रोसेसर आर्किटेक्चर

ऐड्रेस बस इन इंटेल 8085 माइक्रोप्रोसेसरः– इस प्रोसेसर में 8 पिन मेमोरी ऐड्रेसेस की मोस्ट सिग्निफिकेंट बिट को ट्रांसमिट किये जाने के लिये होती है। और ऐड्रेसेस की लिस्ट सिग्निफिकेन्ट डिजिट्स 8 बिट्स, 8 लाईन पर ट्रांसमिट होती है जिसमें डेटा ट्रांसमिट होता है। इस प्रकार डेटा और ऐड्रेसेस का ट्रांसमिशन शेयर्ड लाईनस के सेटस के द्वारा होता है। जिसे ऐड्रेस मल्टीप्लेक्सिंग से जाना जाता है। इस प्रकार 8085 में 16 बिट के ऐड्रेस को ट्रांसमिट किये जाने की योग्यता होती है। अर्थात कुल 2^{16}=65536 मेमोरी लोकेशन को 8085 के द्वारा ऐड्रेस किया जा सकता है। प्रत्येक मेमोरी लोकेशन के लिये 8085 से 8 बिट या 1 बाईट का डाटा मेमोरी लोकशन में पेरेलल ट्रांसफर किया जाता है। इस प्रकार 8085 से मेमोरी के 64K (1 K= 1024) बाईट्स ऐड्रेसेस किये जा सकते है।



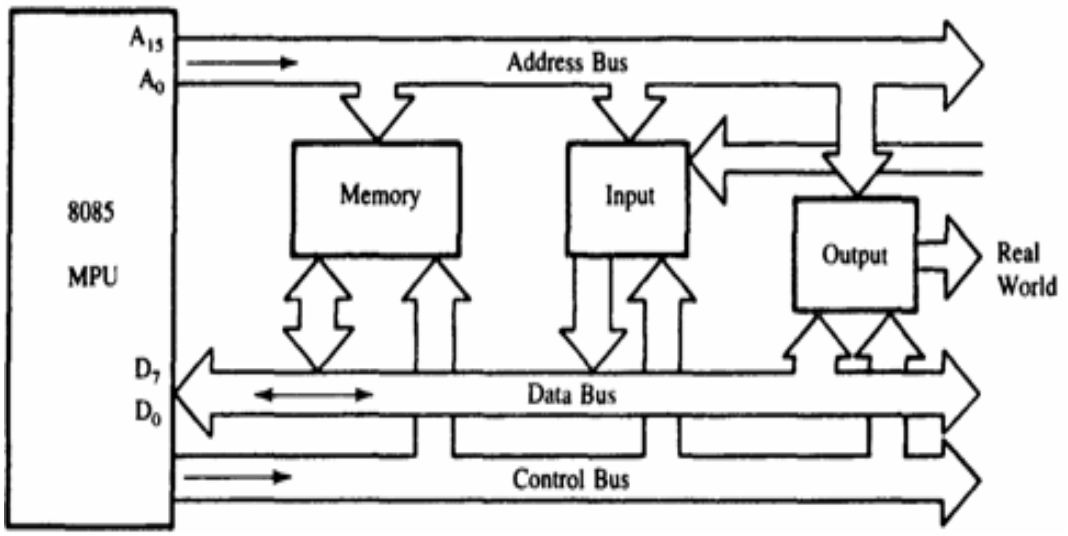
इस प्रोसेसर में ऐड्रेस बस 16 लाईन का समूह होता है जिसे A_0 से A_{15} द्वारा आईडेन्टिफाई किया जाता है। ऐड्रेस बस यूनीडायरेक्शनल होती है। जिसमें बिट्स एक ही दिशा में माइक्रोप्रोसेसर से पेरिफेरल डिवाईस की और फ्लो होती है।

डाटा बस इन इंटेल 8085 माइक्रोप्रोसेसर :– इस प्रोसेसर में डाटा बस के लिए 8 लाईन का उपयोग डाटा के फ्लो के लिये किया जाता है। यह बस या लाईनस बाईडायरेक्शनल होती है। अर्थात इस बस में डाटा का फ्लो दोनों दिशाओं, माइक्रोप्रोसेसर से पेरिफेरल डिवाईस और मेमोरी की और या उसके विपरीत होता है। इस माइक्रोप्रोसेसर में इस डाटा बस का द्वितीय उपयोग बाईनरी इनफार्मेशन के स्थानांतरण के लिये भी होता है। अर्थात ऐड्रेसेस की लिस्ट सिग्निफिकेन्ट डिजिट्स 8 बिट्स, 8 लाईन पर ट्रांसमिट होती है जिसमें डेटा ट्रांसमिट होता है। इस प्रकार डेटा और ऐड्रेसेस का ट्रांसमिशन शेयर्ड लाईनस के सेटस के द्वारा होता है। जिसे ऐड्रेस मल्टीप्लेक्सिंग से जाना जाता है। किसी समय पर डाटा और एड्रेसेस का ट्रांसमिशन अलग–अलग बिन्दु से होता है। इस मल्टीप्लेक्सिंग के कारण इस प्रोसेसर की बस मल्टीप्लेक्सड़ बस भी कहलाती है।

8 बिट डाटा बस की चौड़ाई होने के कारण, यह माइक्रोप्रोसेसर को 8 बिट के डाटा रेन्ज 00 से FF तक (2^8= 256 नम्बरस) के डाटा को चलाने के लिये माइक्रोप्रोसेसर को योग्य बनाती है। इस कारण 8085 माइक्रोप्रोसेसर 8–बिट माइक्रोप्रोसेसर कहलाता है। इस प्रोसेसर में इस डाटा बस को D_0 से D_7 तक आईडेन्टिफाईड किया जा सकता है। जबकि माइक्रोप्रोसेसर 8086 में 16 बिट की डाटा लाईनस् होती है। इस कारण 8086 माइक्रोप्रोसेसर 16 –बिट माइक्रोप्रोसेसर कहलाता है।

कन्ट्रोल बस इन इंटेल 8085 माइक्रोप्रोसेसरः– इस बस में वह लाईनस सम्मिलित होती है जो सिंक्रोनाईजेशन सिग्नलस ले जाती है। माइक्रोप्रोसेसर इस लाईनस का उपयोग तीसरे फंक्शनस टाईमिंग सिग्नल को प्रोवाईड करने के लिये करता है। यहॉ बस शब्द का संबध कंट्रोल सिग्नलस से है न कि डाटा बस या ऐड्रेस बस के समान लाईनस के समूह से है, यह विशिष्ट लाईनस है जो कि पल्स को प्रोवाईड करती है जो कि माइक्रोप्रोसेसर ऑपरेशन को इंडिकेट करती है। माइक्रोप्रोसेसर के प्रत्येक ऑपरेशन जैसे मेमोरी को रीड करना तथा इनपुट/आउटपुट को राईट करना के बाद कंट्रोल सिग्नलस जनरेट किये जाते है। इस कंट्रोल सिग्नल से उस डिवाईस को आईडेन्टीफाईड़ किया जा सकता है जिससे माइक्रोप्रोसेसर को कम्यूनिकेट करना है। जैसे माइक्रोप्रोसेसर को मेमोरी के किसी लोकेशन से इंस्ट्रक्शन रीड़ किये जाने के लिये, ऐड्रेस बस में 16

बिट ऐड्रेसेस भेजे जाते है जिससे मेमोरी लोकेशनस आईडेन्टीफाईड़ हो जाती है। तब माइक्रोप्रोसेसर द्वारा मेमोरी को रीड़ किये जाने के कंट्रोल सिग्नल जो कि पल्स के रूप में होते है भेजे जाते है। यह पल्स मेमोरी चिप को ऐक्टीवेट कर देती है और उस मेमोरी लोकेशन का कंटेन्ट 8 बिट की डाटा बस के द्वारा माइक्रोप्रोसेसर में आता है।



The 8085 Bus Structure

इंटरनल डाटा ऑपरेशन ऑफ 8085:– इंटेल 8085 में निम्न ऑपरेशनस होते है।

1. 8 बिट डाटा को स्टोर करना।
2. अर्थमैटिक और लॉजिकल ऑपरेशन परफार्म करना।
3. कंडिशन का परीक्षण करना।
4. इंस्ट्रक्शन के संपादन का अनुक्रम तय करना।
5. डाटा के एक्जीक्यूशन के समय डाटा को अस्थाई रूप से रीड/राईट मेमोरी लोकेशन पर स्टोर करना जिसे स्टेक कहते है।

माइक्रोप्रोसेसर के द्वारा उक्त ऑपरेशन को परफार्म किये जाने के लिये रजिस्टर, ए.एल.यू. और कन्ट्रोल लॉजिक, और इंटरनल बस की आवश्यकता होती है। उक्त चित्र में इंटेल 8085 माइक्रोप्रोसेसर का इंटरनल आर्किटेक्चर दर्शाया गया है।

रजिस्टर्स इन 8085– माइक्रोप्रोसेसर में रजिस्टर का उपयोग अस्थाई स्टोरेज और डाटा और इंस्ट्रक्शन के मेनिप्यूलेशन के लिये किया जाता है। डाटा रजिस्टर में तब तक रखे रहते है जब तक की वो मेमोरी या इनपुट/आउटपुट डिवाईस में नहीं भेज दिये जाते। अत्यधिक स्पीड़ वाले कम्प्यूटर में रजिस्टर अधिक होते है इस कारण डाटा का स्थानांतरण कम होता है। इंटेल 8085 में निम्न रजिस्टर होते है –

1. छः– 8 बिट रजिस्टर जो सामान्य उद्देश्य के लिये होते है। यह B, C, D, E, H और I है।
2. एक–8 बिट ऐक्यूमूलेटर (ACC) अर्थात रजिस्टर A
3. एक–16 बिट प्रोग्राम काउन्टर, PC
4. एक–16 बिट स्टैक प्वांइटर, SP
5. इंस्ट्रक्शन रजिस्टर
6. स्टेटस रजिस्टर
7. टेमप्रेररी रजिस्टर

उक्त रजिस्टर के अतिरिक्त इंटेल 8085 में ऐड्रेस बफर और डाटा/ऐड्रेस बफर भी होते है।

1. **सामान्य रजिस्टरः–** 8085 में छः सामान्य उद्देश्य के रजिस्टर होते है जो कि प्रथम ऑपरेशन, प्रोग्राम के संपादन (एक्सिक्यूशन) के समय 8 बिट डाटा को स्टोर करते है। इस रजिस्टर को B,C,D,E,H,L से आईडेन्टिफाईड किया जा सकता है। यह रजिस्टर 16 बिट ऑपरेशन के लिये जोड़े के रूप में भी संयुक्त हो सकते है। इस प्रोसेसर में वैध रजिस्टर पेयर –BC, DE, HL होते है।

यह रजिस्टर प्रोग्रामेबल होते है अर्थात प्रोग्रामर इसका उपयोग इंस्ट्रक्शनस के द्वारा डाटा को लोड़ और कॉपी करने के लिये कर सकता है।

2. **ऐक्यूमुलेटर :–** यह 8 बिट का रजिस्टर होता है जो कि ए.एल.यू. का पार्ट होता है। इस रजिस्टर का उपयोग 8 बिट डाटा को स्टोर करने के लिये और अर्थमैटिक ऑर लॉजिकल ऑपरेशन को परफार्म करने के लिये उपयोग किया जाता है। ऑपरेशन के रिजल्ट को इस एक्यूमुलेटर रजिस्टर में स्टोर किया जाता है। एक्यूमुलेटर रजिस्टर को लेटर A से आईडेन्टिफाईड किया जा सकता है।

3. फ्लैगस– ए.एल.यू. में पॉच फ्लिप–फ्लॉप होते है जो कि ऑपरेशन के परिणाम के आधार पर सेट और रिसेट होते है। माइक्रोप्रोसेसर इसका उपयोग तीसरे ऑपरेशन डाटा कंडीशन का परीक्षण करने के लिये करता है। उदाहरणार्थ यदि दो नम्बर का योग ऐक्यूमलेटर में 8 बिट से अधिक होता है तो फ्लिप–फ्लॉप के उपयोग द्वारा कैरी को इंडीकेट किया जाता है। इस फ्लिप–फ्लॉप को कैरीफ्लैग (CY) कहते है जो 1 पर सेट होता है। जब अर्थमैटिक ऑपरेशन का परिणाम शुन्य होता है, तब जिस फ्लिप–फ्लॉप के द्वारा इसे 1 पर सेट करता है उसे Zero flage (Z) कहते है। 8085 में पॉच फ्लैगस होते है जो कि पॉच विभिन्न प्रकार के परिणाम या डाटा कंडिशन को इंडिकेट करते है। यह पॉच फ्लैगस है–Zero(Z), Carry (CY), Sign(S), Parity (P) और Auxiliary Carry (AC)। चित्र में एक्यूमेलटर के पास फ्लैग रिजिस्टर दर्शाये गये है। इसमें 8 बिट रजिस्टर की तरह उपयोग नहीं किया जाता है। 8 में से 5 बिट पोजिशनस का उपयोग पॉच फ्लिप–फ्लॉप के आउट पुट को स्टोर करने के लिये किया जाता है। यह फ्लैगस 8 बिट रजिस्टर में स्टोर किये जाते है जिसे प्रोग्रामर इन रजिस्टर से फ्लैग को ऐक्सेस कर तथा इस फ्लैगस का परिक्षण कर डाटा कंडिशन का परिक्षण, इंस्ट्रक्शनस के द्वारा कर सकता है। यह फ्लैगस माइक्रोप्रोसेसर की डिसिजन मेकिंग प्रक्रिया के समय महत्वपूर्ण होते है। फ्लैग की कंडिशन सेट अथवा रिसेट को साफ्टवेयर के इंस्ट्रक्शन के द्वारा परिक्षण किया जाता है।

Carry (CY):- कैरी स्टेटस फ्लैग दर्शाता है कि डाटा के अर्थमैटिक ऑपरेशन जैसे जोड़ और घटाव या तुलना के समय कैरी है। इस स्थिति में कैरी फ्लैग 1 पर सेट होगा। अन्यथा यह शून्य होगा।

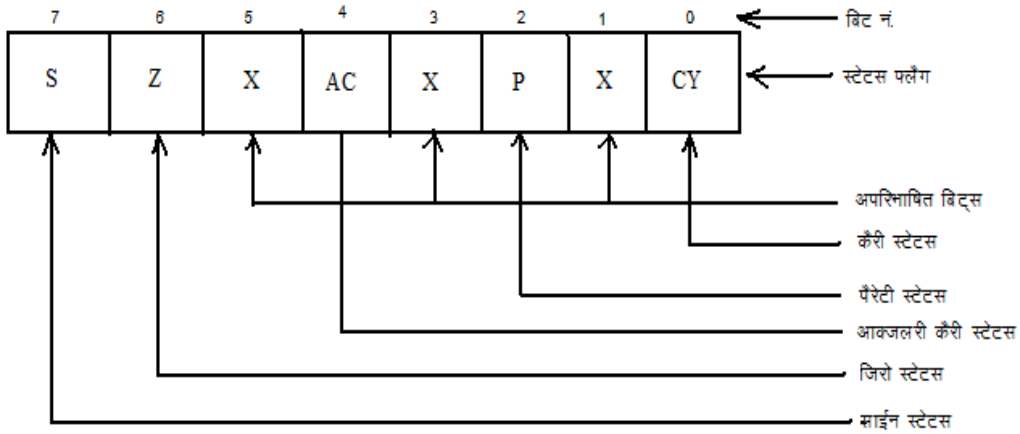
Zero(Z) :- Zero स्टेटस फ्लैग Z, 1 पर सेट होगा जबकि अर्थमैटिक और लॉजिकल ऑपरेशन का परिणाम शुन्य होता है। यदि परिणाम अशुन्य है तो यह 0 पर सेट होता है।

Sign(S):- अर्थमैटिक और लॉजिकल ऑपरेशन के परिणाम में यदि मोस्ट सिगनिफिकेन्ट बिट 1 है तो साईन स्टेटस फ्लैग 1 पर सेट होता है। अन्यथा 0 पर।

Parity (P) :- यदि ऑपरेशन में 1 कि संख्या सम है तो पैरिटी स्टैटस फ्लैग 1 पर सेट होता है। यदि ऑपरेशन में 1 कि संख्या विषम है तो यह 0 पर सेट होता है।

Auxiliary Carry (AC):- यदि अर्थमैटिक या लॉजिकल ऑपरेशन यदि कैरी बिट 3 से बिट 4 को जावे तो आक्जिलरी कैरी स्टैटस फ्लैग सेट होता है।

PSW:- इस प्रकार 8 बिट के रजिस्टर में 5 बिट फ्लैग स्टेटस को दर्शाते है जबकि 3 बिटस अपरिभाषित है। यह 8 बिट का संयोजन प्रोग्राम स्टेटस वर्ड कहलाता है। PSW और एक्यूमलेटर स्टैक ऑपरेशन के लिये 16 बिट यूनिट की तरह होते है।



4. प्रोग्राम काउन्टर, (PC):– यह 16 बिट का रजिस्टर होता है जो कि माइक्रोप्रोसेसर के चौथे ऑपरेशन, इंस्ट्रक्शनस के संपादन के अनुक्रम को तय करता है। यह रजिस्टर अगले इंस्ट्रक्शन के ऐड्रेसेस को रखता है। यह रजिस्टर मेमोरी प्वाइंटर है। मेमोरी ऐड्रेसेस 16 बिट के होते है इस कारण यह रजिस्टर भी 16 बिट का होता है। माइक्रोप्रोसेसर इस रजिस्टर का उपयोग निर्देशों के संपादन के अनुक्रम को तय करने के लिये करता है। प्रोग्राम काउन्टर का कार्य उस मेमोरी ऐड्रेस को अपने में स्टोर रखना है जो कि अगले इंस्ट्रक्शन का मेमोरी लोकेशन ऐड्रेस है। जिसे अगली 8 बिट मेमोरी ऐड्रेस के रूप में मेमोरी को फैच किया जाना है। जब एक बाईट मशीन कोड़ को फैच किया जाता है तो प्रोग्राम काउन्टर उसमें 1 जोड़ देता है जो कि अगले मेमोरी लोकेशन को प्वांइट करता है।

5. **स्टैक प्वांइटर (SP)** :– यह भी 16 बिट का रजिस्टर होता है जिसका उपयोग मेमोरी प्वाइंटर के रूप में किया जाता है। इसलिये इसे स्टैक प्वाइंटर रजिस्टर भी कहते है। यह रीड और राईट मेमोरी लोकेशन को प्वाइंट करता है जिसे स्टैक कहते है।

6. **इंस्ट्रक्शन रजिस्टर और डिकोडरः–** इंस्ट्रक्शन रजिस्टर और डिकोडर ए.एल.यू. के ही भाग होते है। जब मेमोरी से इंस्ट्रक्शन फैच किये जाते हे तो यह इंस्ट्रक्शन रजिस्टर में लोड़ेड हो जाते है। डिकोडर, इंस्ट्रक्शनस को डिकोड करता है और इंवेटस के फ्लो की सिक्वेन्स को स्थापित करता है। इंस्ट्रक्शनस रजिस्टर प्रोग्रामेबल नहीं होता है और किसी भी इंस्ट्रक्शन के द्वारा एसेस नहीं किया जा सकता है।

7. टेम्परेरी रजिस्टर :– दो अतिरिक्त रजिस्टर जिसे टेम्परेरी रजिस्टर W और Z कहते है। इस रजिस्टर का उपयोग कुछ इंस्ट्रक्शन के संपादन के समय 8 बिट डाटा को होल्ड़ करने के लिये किया जाता है। यह इंटरनली होते है यह प्रोग्रामर के लिये एवेलेबल नहीं होते है।

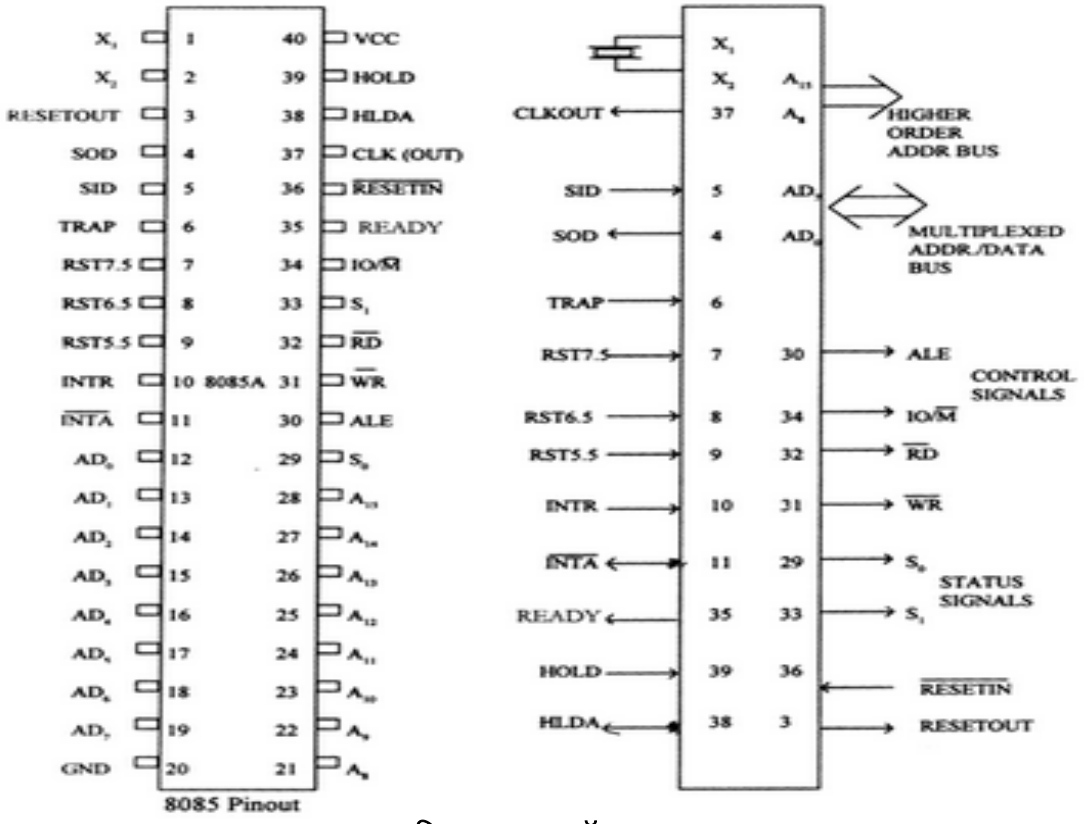
ए.एल.यू.:– अर्थमैटीक एंव लॉजिक यूनिट द्वारा निम्न अर्थमैटिक एंव लॉजिकल ऑपरेशन परफार्म किये जाते है।

1. जोड 2. सबट्रेक्शन 3. लॉजिकल AND 4. लॉजिकल OR 5.लॉजिकल EXOR 6. काम्प्लीमेन्ट (लॉजिकल NOT) 7. इंक्रीमेन्ट (ऐडिशन ऑफ 1) 8. डिक्रीमेन्ट (सब्ट्रेक्शन ऑफ 1) 9. लेफट शिफट 10. क्लियर

इसमें एक्यूमलेटर, टेम्परेरी रजिस्टर, अर्थमेटिक और लॉजिक सर्किट और पॉच फ्लैगस होते है। टैम्परेरी रजिस्टर का उपयोग अर्थमैटिक और लॉजिकल ऑपरेशन के समय डाटा को होल्ड़ करना है। परिणाम को ऐक्यूमुलेटर में स्टोर किया जाता है।

टाईमिंग एण्ड कंट्रोल यूनिट :– टाईमिंग एण्ड़ कंट्रोल यूनिट सी.पी.यू. का सेक्शन होता है। यह टाईमिंग और कन्ट्रोल सिग्नल जनरेट करता है जो इंस्ट्रक्शन को संपादित किये जाने के लिये आवश्यक होता है। यह सी. पी.यू. और अन्य पेरिफेरल्स सह मेमोरी मे मध्य डाटा के फ्लो को नियंत्रित करता है। यह माइक्रोप्रोसेसर और इससे संबंधित पेरिफैरेल्स के मध्य सम्पूर्ण ऑपरेशन को नियंत्रित करता है।

पिन डायग्राम ऑफ इंटेल 8085 माइक्रोप्रोसेसरः– इंटेल 8085A जिसे सामान्यतः इंटेल 8085 से भी जाना जाता है।



पिन डायग्राम ऑफ 8085

इंटेल 8085 के सभी सिग्नलस को 6 ग्रुप में बॉटा गया है। 1.ऐड्रेस बस 2.डाटा बस 3.कंट्रोल और स्टेटस सिग्नल 4.पॉवर सप्लाई और फ्रिक्वेंसी सिग्नल 5.एक्सटरनल इंनिशीऐटेड़ सिग्नल 6.सिरियल इनपुट/आउटपुट पोर्टस

ऐड्रेस बसः–इंटेल 8085 में 8 सिंगनल लाईनस A_8-A_{15} होती है जो कि युनिडाईरेक्शनल होती है जिसे हाई आर्डर ऐड्रेस बस भी कहते है।

मल्टीप्लैक्सड़ ऐड्रेस/डाटा बस :– इंटेल 8085 में सिंग्नल लाईन AD_0-AD_7 मल्टीप्लैक्सड़ ऐड्रेस/डाटा बस होती है। जो दो उद्देश्यों के लिये लो ऑर्डर ऐड्रेस बस और डाटा बस के लिये उपयोग की जाती है। इंस्ट्रक्शनस के एक्जीक्युशन के समय साईकल के पहले भाग के लिये लो आर्डर ऐड्रेस बस का उपयोग किया जाता है। साईकल के बाद के भाग के लिये यह लाईन डाटा बस के लिये उपयोग की जाती है। इसे बस की मल्टीप्लेक्सिग भी कहते है। लैच के उपयोग के द्वारा इसे ऐड्रेस बस को सिग्नल से पृथक किया जा सकता है।

कंट्रोल और स्टेटस सिग्नल– यह ग्रुप ऑफ सिग्नल सह दो कंट्रोल सिग्नल ($\overline{RD}$ और $\overline{WR}$) और तीन स्टेटस सिग्नल (IO/M, $\overline{S_1}$ और S_0) जो कि ऑपरेशन के नेचर को आइडेन्टीफाई करते है और एक स्पेशल सिग्नल (ALE) जो ऑपरेशन के प्रारंभ को इंडिकेट करता है।

ALE ऐड्रेस लेच इंनेबल – यह पॉजिटिव पल्स है जो कि ऑपरेशन के प्रारंभ में प्रत्येक समय जनरेट होती है। जो कि यह इंडिकेट करती है कि लाईन AD_0-AD_7 में ऐड्रेस बिट है। इस सिग्नल के उपयोग द्वारा मल्टीप्लेक्सड़ बस में लो आर्डर ऐड्रेस को लेच किया जाता है। अर्थात इस सिग्नल के द्वारा लाईन मल्टीप्लैक्सड़ लाईन से ऐड्रेस लाईन के लिये पृथक सेट A_0-A_7 जनरेट किया जाता है।

$\overline{RD}$–रिड–यह रिड़ कन्ट्रोल सिग्नल लो होने पर यह इंडिकेट करता है कि सिलेक्टेड़ इनपुट/आउटपुट या मेमोरी डिवाईस रीड़ की जा चुकी है और डाटा, डेटा बस में मिलता है।

$\overline{WR}$ –राईट– राईट कंट्रोल सिग्नल लो होने पर यह दर्शाता है कि डाटा बस में उपस्थित डाटा को सिलेक्टेड़ मेमोरी या इनपुट/आउटपुट लोकेशन पर राईट किया जा चुका है।

IO/$\overline{M}$ – इस स्टेटस सिग्नल का उपयोग इनपुट/आउटपुट ऑपरेशन और मेमोरी ऑपरेशन को अलग करने के लिये किया जाता है। जब यह हाई होता है तो यह इनपुट/आउटपुट ऑपरेशन

को दर्शाता है अर्थात यह दर्शाता है कि एड्रेसेस इनपुट/आउटपुट डिवाईसेस के लिये है। जब यह लो होता है तो यह मेमोरी ऑपरेशन को दर्शाता है अर्थात एड्रेसेस मेमोरी के लिये है।

S_1 और S_0 – यह स्टेटस सिगनल माइक्रोप्रोसेसर भेजे जाते है जो कि विभिन्न ऑपरेशन को निन्नानुसार दर्शाते है।

S1	S0	Operation
0	0	HALT
0	1	WRITE
1	0	READ
1	1	FETCH

पावर सप्लाई और क्लॉक फ्रिक्वेंसी–

1.इसे ऑपरेट के लिये Vcc= +5 V पावर सप्लाई का उपयोग किया जाता है।

2.Vss : ग्राउन्ड रिफरेन्स के लिये उपयोग किया जाता है ।

3. X_1, X_2 : क्रिस्टल को इन दो बिन्दुओं के बीच कनेक्ट किया जाता है। इसमें आंतरिक रूप से फ्रिक्वेंसी को दो भागों में विभाजित किया जाता है। इस प्रकार सिस्टम 3 मेगा हर्टज से ऑपरेट होता है इसलिये क्रिस्टल की फिक्वेंसी 6 मेगा हर्टज होना चाहिए।

4. CLK(Out)- इस सिग्नल का उपयोग अन्य डिवाईस के लिये सिस्टम क्लॉक के लिये किया जाता है।

एक्सटरनल इंनिशिएटेड़ सिंगनल सह इंटरप्टस– इंटेल 8085 माइक्रोप्रोसेसर प्रोग्राम के संपादन में पॉच इंटरप्ट सिग्नल का उपयोग किया जाता है।

1. INTR (Input)- यह इंटरपप्ट रिक्वेस्ट सिग्नल है। जब यह हाई होता है तो प्रोग्राम काउन्टर उसके कन्टेन्ट में इंक्रीमेन्ट नहीं करता है। इस प्रकार माइक्रोप्रोसेसर उसके सामान्य इंस्ट्रक्शन्स के संपादन के अनुक्रम को सस्पेन्ड़ कर देता है।

2. INTA(Outpur)- इंटरप्ट ऐक्नॉलेजमेन्ट है जिसे माइक्रोप्रोसेसर INTR के प्राप्त होने पर भेजता है।

3. RST 5.5, 6.5, 7.5 और TRAP (Inputs):- रिर्स्टाट इंटरप्ट–यह दिशा सहित इंटरप्ट रिक्वेस्ट है जो कि प्रोग्राम कंट्रोल को स्थानांतरित की जाती है, जो की मेमोरी की किसी निश्चित लोकशन के लिये होती है। इसकी प्राथमिकता INTR इंटरप्ट से अधिक होती है। जब इंटरप्ट को स्वीकृत कर लिया जाता है तो मेमोरी से अगला इंस्ट्रक्शन किसी निश्चित लोकेशन से संपादित किया जाता है।

Line	Location from which next instruction isPicked up
TRAP	0024
RST 5.5	002C
RST 6.5	0034
RST 7.5	003C

रिसेट इन (इनपुट):– यह प्रोग्रम काउन्टर को जीरो पर रिसेट कर देता है। यह इंटरप्ट इनेबल और HLDA फ्लिप–फ्लॉप दोनों को भी रिसेट करता है। यह इंस्ट्रक्शन रजिस्टर को छोड़कर किसी अन्य फ्लेग और रजिस्टर को प्रभावित नहीं करता है। जब रिसेट एप्लाई होता है तो जब तक सी.पी.यू. रिसेट कंडिशन में होता है।

रिसेट आउट (आउटपुट):– यह दर्शाता है कि सी.पी.यू. रिसेट स्थिती में है।

X_1और X_2 (इनपुट):- यह टर्मिनल होते है जिसमें एक्सटर्नल क्रिस्टल ऑसिलेटर को कनेक्ट किया जाता है जो कि माइक्रोंप्रोसेसर की आंतरिक सर्किटरी को ड्राइव करता है जो कि माइक्रोप्रोसेसर के ऑपरेशन के लिये उपयुक्त क्लॉक जनरेट करता है।

Clock **आउटपुट**– यह क्लॉक आउटपुट यूजर के लिये होता है जिसका उपयोग अन्य डिजिटल आई.सी. के लिये किया जा सकता है। इसकी फ्रिक्वेंसी, प्रोसेसर की ऑपरेट फ्रिक्वेंसी के समान होती है।

SID (**इनपुट**)– यह सिरियल इनपुट के लिये डाटा लाईन होती है। इस लाईन का डाटा एक्यूमलेटर के 7 वी बिट में लोडेड होता है जबकी RIM इंस्ट्रक्शन कार्यान्वित होता है।

SOD(**आउटपुट**)– यह सिरियल आउटपुट के लिये डाटा लाईन होती है। एक्यूमुलेटर की 7 वी बिट का आउटपुट SOD लाईन पर होता है जबकि SIM इंस्ट्रक्शन कार्यान्वित होता है।

000

लेखक – उ.नि. रे. राजेन्द्र कुमार स्वामी

बिषय सामग्री का संदर्भ –

Digital Electronics

1- Digital Principles and Applications- Leach and Malvino
2- Modern Digital Electronics – R.P.Jain
3- Digital Computers- Rashid Sheikh
4- Digital Electronics- William H.Gothmann
5- Microprocessor Architecture, Programming and Application with the 8085- Ramesh S. Gaonkar
6- Introduction to Microprocessor – Aditya P Mathur
7- Fundamental to Microprocessors and Microcomputers – B.Ram
8- Hand Book of Electronics – Gupta & Kumar
9- Internet Web Sites
10- Basic Electronics- B.L.Theraja