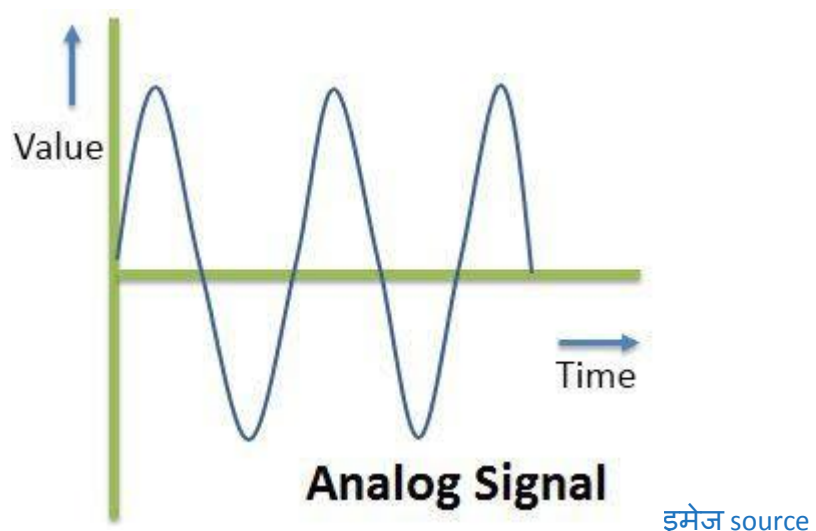


Unit-01

Analog Signal

Analog Signal नियमित electrical signal हैं जो समयनुसार बदलते रहता हैं, यह परिवर्तन Non-Electric Signal का अनुसरण करता हैं टेलीफ़ोन पर की जाने वाली बातचीत में हमारी आवाज को एक स्थान से दूसरे स्थान तक भेजना Analog Signal के माध्यम से होता है। **साइन वेव(Sine Wave)** भी एनालॉग सिगनल का एक अच्छा उदाहरण है,

Human Voice, Music, Video आदि एनालाग सिग्नल के उदाहरण है। एनालाग सिग्नल की मुख्य कमी यह है कि इसमें High Quality Transmission नहीं हो पाता है तथा गलतियों की संभावना अधिक होती है।



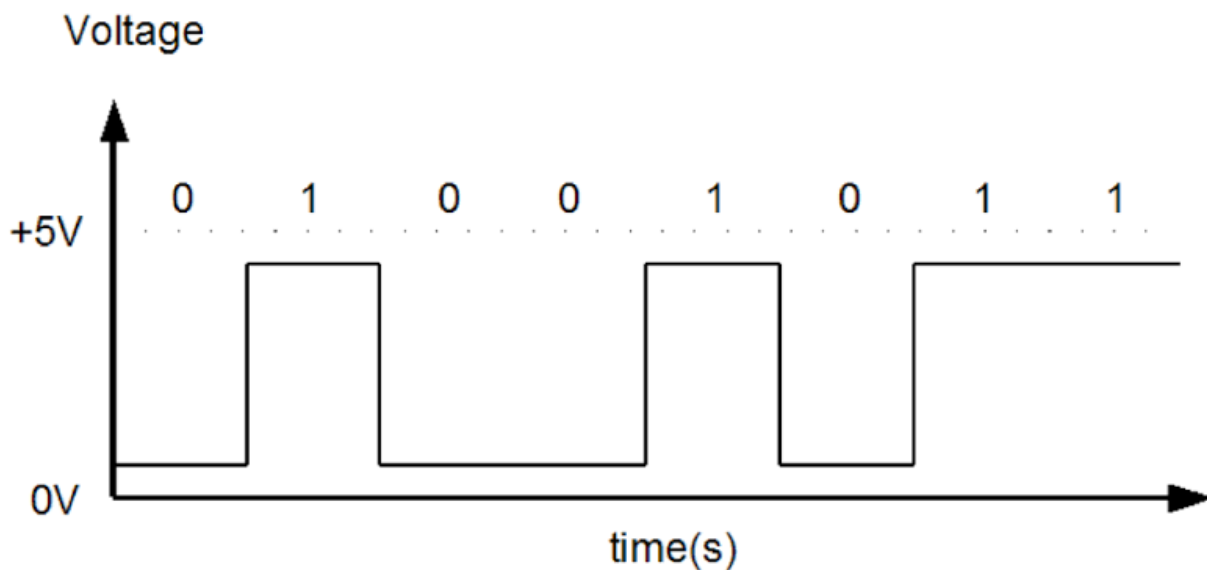
Advantage Of Analog Signal

1. Analog Signal की कोई निश्चित सीमा नहीं है।
2. Analog Signal एक तरंग के रूप में डेटा संचारित करता है।
3. एनालॉग सिगनल से ही हमारे मोबाइल, एक दूसरे से जुड़े हैं।
4. एनालॉग सिगनल के कारण सूचना को, electromagnetic तरंगों एक माध्यम से पूरी धरती में और धरती के बाहर(Satellite) तक पहुँचाना सम्भव हुआ

Digital Signal

ये Irregular (अनियमित) होते हैं ये Signal अलग-अलग steps में change होते रहते हैं , विभिन्न levels या values के साथ यह pulse या digit रखता हैं. प्रत्येक pulse का मान स्थिर रहता है परन्तु एक digit से अगली digit में suddenly change होता हैं

Digital Signal के दो Amplitude होते हैं जिन्हे हम nodes कहते हैं जिसका value दो संभव value 0 या 1, उच्च या निम्न, सत्य या असत्य में से एक होता हैं , डिजिटल सिगनल में मानों की संख्या सीमित होती है।



Advantage Of Digital Signal

1. Digital System को Design करना आसान होता है, क्योंकि इनमें Switching Circuit प्रयोग किये जाते हैं, जिनमें Signal का वास्तविक मान महत्वपूर्ण नहीं होता है, बल्कि उनकी रेंज(High/Low) महत्वपूर्ण होती है।
2. Digital format में information को store करना आसान होता है।
3. इनमें Accuracy अधिक होती है, और Switching Circuit की संख्या अधिक कर accuracy को बढ़ाया जा सकता है।
4. Operation को Computer की सहायता से Program किया जा सकता है।
5. Digital System में शोर का प्रभाव कम होता है।
6. Digital Circuit को IC(Integrated Circuit) पर बनाया जाता है।

7. Signal का Digital Communication करने पर उनमे होने वाली Error का Detection और Correction संभव है।
8. Signal का Digital Communication अधिक Secure होता है।
9. VLSI(Very Large Scale Integration) तकनीकी सहायता से Transceivers small हलके व मोबाइल हो गए है।
10. डिजिटल परिपथ(Circuit) अधिक Reliable होते है।

Difference Between Analog and Digital Signal

विशेषताएं	Analog	Digital
signal का प्रकार	continuous signal	binary signal
Accuracy	digital से कम accuracy होती हैं	इसकी Accuracy उच्च होती हैं
Speed	इसकी speed निम्न होती हैं	इसकी Speed उच्च होती हैं
data Transmission	उच्च Quality का डाटा transmission perform नहीं करता	उच्च Quality का डाटा transmission perform करता हैं
Access Time	लगभग 1 Sec	कुछ sec
Integration Method	Continuous	Numerical

Unit-03

Logic gates

logic gates किसी भी digital system के बेसिक building blocks होते हैं। logic gates एक electronic circuit होता है। इसमें एक या एक से अधिक input होते हैं और केवल एक output होता है। input और output के बीच की relationship एक विशेष logic पर आधारित होती है।

इसमें इनपुट और आउटपुट binary form में होते हैं अर्थात् इसमें इनपुट/आउटपुट की वैल्यू केवल 0 (low) तथा 1 (high) होगी।

Types of logic gates

सामान्यतया तीन प्रकार के logic gates होते हैं –

1. AND gate
2. OR gate
3. NOT gate

और भी कई प्रकार के logic gates होते हैं जो इन तीन logic gates के base पर होते हैं। जैसे :-

NAND GATE , NOR GATE, XOR gate.

AND gate

AND GATE में दो या दो से अधिक inputs होते हैं पर output केवल एक होता है। AND GATE का output तभी high होता है जब सभी input high होते हैं। AND GATE दो या दो से अधिक inputs पर multiplication operation perform करता है। AND GATE के operation को (.) से प्रदर्शित करते हैं।

Ex :- $X = A \text{ AND } B$

$X = A.B$

लॉजिक डायग्राम:- यहाँ दो input A और B है और एक output $X = A.B$ है,



TRUTH TABLE :-

INPUT A	INPUT B	OUTPUT X
0	0	0
0	1	0
1	0	0
1	1	1

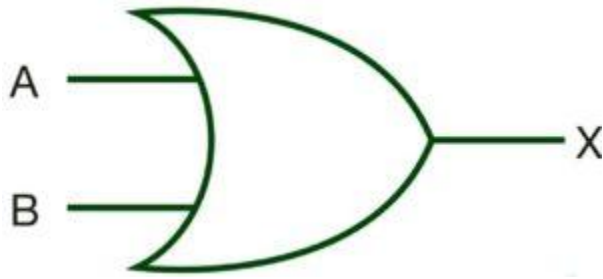
OR GATE

OR GATE में दो या दो से अधिक input signals होते हैं और केवल एक output signal होता है। यह OR GATE इसलिये कहलाता है क्योंकि किसी भी एक input या सभी input के high होने पर output high होता है। OR GATE में दो या दो से अधिक input पर addition (जोड़) operation परफॉर्म किया जाता है। OR GATE operation को (+) symbol से प्रदर्शित किया जाता है।

Ex :- $X = A \text{ OR } B$

$X = A + B$

लॉजिक डायग्राम :-



truth table:-

INPUT A	INPUT B	OUTPUT X
0	0	0
0	1	1
1	0	1
1	1	1

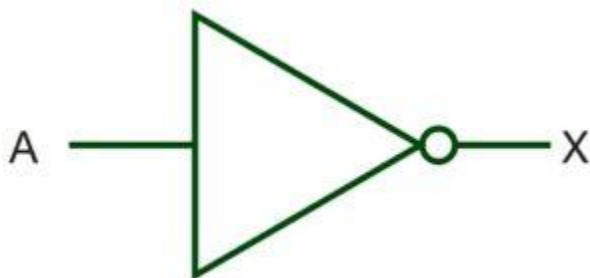
NOT GATE

NOT GATE में केवल एक input और एक output होता है क्योंकि इसकी output स्टेट input स्टेट के opposite (विपरीत) होती है। अर्थात input तब high होता है जब output low होता है। NOT GATE को complement, negation या inverter भी कहा जाता है। NOT GATE operation को bar या ($\bar{}$) symbol से प्रदर्शित किया जाता है।

Ex :- $X = \text{NOT } A$

$X = A^{\bar{}}$

लॉजिक डायग्राम:-



truth table:-

INPUT A	OUTPUT X
0	1
1	0

UNIVERSAL GATE

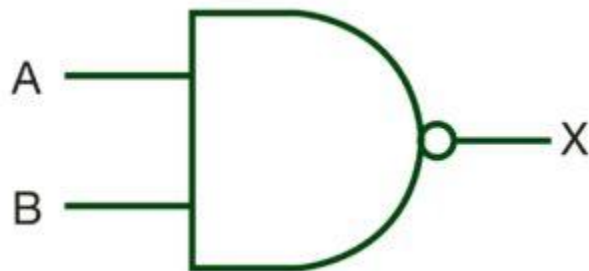
NAND GATE

एक AND GATE को NOT GATE के साथ जोड़ने पर एक नया GATE प्राप्त होता है जिसे NAND GATE कहते हैं। यह NOT और AND operation perform करता है। NAND GATE जो है वह AND GATE का compliment (\neg) होता है।

Ex :- $X = A \text{ NAND } B$

$$X = (AB)^{\neg}$$

लॉजिक डायग्राम :-



TRUTH TABLE :-

INPUT A	INPUT B	OUTPUT X
0	0	1
0	1	1
1	0	1
1	1	0

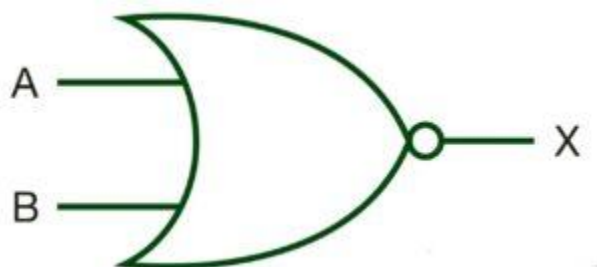
NOR GATE

यह NOT और OR GATE का combination होता है। एक NOT GATE और एक OR GATE को जोड़ने से जो GATE प्राप्त होता है उसे NOR GATE कहते हैं। NOR GATE जो है वह OR GATE का compliment होता है।

Ex :- $X = A \text{ NOR } B$

$$X = (A+B)^{-}$$

लॉजिक डायग्राम :-



TRUTH TABLE :-

INPUT A	INPUT B	OUTPUT X
0	0	1
0	1	0
1	0	0
1	1	0

SSI, MSI, LSI, VLSI

डिजिटल ICs परिपथ (circuit) की जटिलता (complexity) के आधार पर वर्गीकृत किये गये हैं। किसी लॉजिक फंक्शन को सम्पन्न करने के लिए आवश्यक gate की कितनी संख्या की आवश्यकता है। यह ICs का gate equivalency वर्गीकरण कहलाता है।

Name *	Signification *	Year *	Transistors number ^(M) *	Logic gates number ^(K) *
SSI	small-scale integration	1964	1 to 10	1 to 12
MSI	medium-scale integration	1968	10 to 500	13 to 99
LSI	large-scale integration	1971	500 to 20,000	100 to 9,999
VLSI	very large-scale integration	1980	20,000 to 1,000,000	10,000 to 99,999
ULSI	ultra-large-scale integration	1984	1,000,000 and more	100,000 and more

1:- SSI – SSI का पूरा नाम small scale integration है। इसमें 12 से कम लॉजिक gates की आवश्यकता होती है। इसे 1964 में प्रस्तावित किया गया था। इसमें 1 से लेकर 10 transistors की जरूरत होती है।

2:- MSI – MSI का पूरा नाम medium scale integration है। इसमें 12 या अधिक परन्तु 100 से कम लॉजिक गेट्स की आवश्यकता होती है। इसमें 12 से कम लॉजिक gates की आवश्यकता होती है। इसे 1968 में develop किया गया था। इसमें 10 से लेकर 500 transistors की जरूरत होती है।

3:- LSI – LSI का पूरा नाम large scale integration है इसमें 100 या अधिक परन्तु 1000 से कम लॉजिक gates की आवश्यकता होती है। इसे 1971 में प्रस्तावित किया गया था। इसमें 500 से लेकर 2000 transistors की जरूरत होती है।

4:- VLSI – इसका पूरा नाम very large scale integration है इसमें 1000 या अधिक लॉजिक gates की आवश्यकता होती है। इसे 1980 में develop किया गया था, इसमें 20,000 से 1 लाख transistors की जरूरत होती है।

CMOS(Complementary metal-oxide-semiconductor)

CMOS का पूरा नाम Complementary Metal Oxide Semi-conductor होता है। C.M.O.S. एक तरह की मेमोरी Chip होती है जो कि motherboard पर लगी होती है और एक

बैटरी की मदद से चल रही है। इस बैटरी को CMOS Cell भी कहते हैं। CMOS को Real Time Clock (RTC) नाम से भी जानते हैं।

CMOS (Complementary Metal Oxide Semi-conductor) chip में हमारे कंप्यूटर की date और time की सेटिंग्स save रहती हैं। जब कंप्यूटर को हम बंद कर देते हैं तब भी सेव रहते हैं, क्योंकि C.M.O.S. chip एक बैटरी की मदद से चल रही होती है। C.M.O.S. Cell काफी सालों तक खराब नहीं होती है और लगातार हमारे C.M.O.S chip को current देती रहती है ताकि हमारे कंप्यूटर की date और time save रहे। अगर आपके कंप्यूटर या लैपटॉप की date और time खराब हो रहे हैं, तो समझ लीजिये कि आपकी C.M.O.S. battery खराब हो चुकी है और उसे बदलने की जरूरत है।

TTL(Transistor-Transistor Logic)

TTL जैसा कि ऊपर उल्लिखित है, TTL *ट्रांजिस्टर-ट्रांजिस्टर लॉजिक का संछिप्त रूप है*। यह एक logic family है, जोकि bipolar process technology के साथ implement होती है। TTL में इस process का उपयोग NPN ट्रांजिस्टर, PN जंक्शन डायोड तथा diffused resistors को एक single monolithic structure पर जोड़ने के लिए किया जाता है, जिससे desired logic function को प्राप्त किया जा सके। NAND gate इस logic family का basic building block है।

ट्रांजिस्टर-ट्रांजिस्टर लॉजिक सबसे लोकप्रिय और व्यापक रूप से इस्तेमाल किया जाने वाला आईसी लॉजिक परिवार है। इसे T 2 L परिवार के रूप में भी जाना जाता है और 1964 में टेक्सास इंस्ट्रूमेंट्स द्वारा पेश किया गया था।

C.M.O.S. में BIOS (Basic Input Output System) की सेटिंग भी सेव रहती है, ताकि जब भी कंप्यूटर चालू हो तब जो सेटिंग्स user ने BIOS के लिए set की होंगी, वो सब में save रहेगी और वही सेटिंग्स C.M.O.S. में से load हो जाएगी।

Boolean algebra in hindi:- (बूलियन अलजेब्रा क्या है?)

boolean algebra एक गणितीय लॉजिक है जिसमें केवल दो values होती हैं सत्य तथा असत्य.

boolean algebra का प्रयोग डिजिटल सर्किटों को analyze तथा simplify करने के लिए किया जाता है. यह केवल बाइनरी संख्याओं (0 तथा 1) का ही प्रयोग करता है. जहाँ 1 सत्य को प्रस्तुत करता है तथा 0 असत्य को प्रस्तुत करता है.

कंप्यूटर आसान तथा जटिल ऑपरेशनों को boolean algebra का प्रयोग करके परफॉर्म करते हैं.

boolean algebra की खोज 1854 में George boole ने की थी.

boolean algebra का प्रयोग लॉजिक गेट्स की संख्या को कम करने के लिए किया जाता है.

उदाहरण के लिए माना हमारे पास वेरिएबल Y है तथा प्रस्तुत करता है बाहर बारिश हो रही है या नहीं.

* सत्य, बाहर बारिश हो रही है.

* असत्य, बाहर बारिश नहीं हो रही है.

real world में और भी चीजें होती हैं जैसे बाहर तेज बारिश हो रही है, लगातार बारिश हो रही है, बारिश के साथ धूप भी आई है.

लेकिन boolean algebra में केवल दो चीजें होती हैं या तो बारिश हो रही है या नहीं हो रही है.

boolean algebra के द्वारा चीजें आसान हो जाती हैं.

Boolean laws in hindi(बूलियन के नियम):-

boolean laws 6 प्रकार के होते हैं:-

- 1:- commutative law
- 2:- associative law
- 3:- AND
- 4:- OR
- 5:- inversion
- 6:- distributive

इसे भी पढ़ें:- [DMA क्या है?](#)

1:- commutative law:- (विनिमय का नियम)

commutative law निम्नलिखित कंडीशन को संतुष्ट करता है:-

- 1) $A.B=B.A$
- 2) $A+B=B+A$

विनिमय का नियम यह कहता है कि वेरिएबल के क्रम को बदल भी दिया जाएँ तो वह लॉजिक सर्किट के आउटपुट को नहीं बदल सकता है.

2:- associative law:-

$$(A+B)+C=A+(B+C)$$
$$(A.B).C=A.(B.C)$$

यह नियम यह कहता है कि combine किये हुए इनपुट वेरिएबल का क्रम इसके आउटपुट पर कोई प्रभाव नहीं डालेगा.

3:- distributive law:-

distributive law निम्नलिखित कंडीशन को संतुष्ट करता है.

$$A.(B+C)=A.B+A.C$$

4:- AND laws:-

- 1) $A \cdot 0 = 0$
- 2) $A \cdot 1 = A$
- 3) $A \cdot A = A$
- 4) $A \cdot A^{\bar{}} = 0$

5:- OR laws:-

- 1) $A + 0 = A$
- 2) $A + 1 = 1$
- 3) $A + A = A$
- 4) $A + A^{\bar{}} = 1$

6:- inversion law:-

यह नियम NOT ऑपरेशन का प्रयोग करता है. inversion law का मतलब है कि किसी वेरिएबल का double complement का आउटपुट स्वयं वेरिएबल होगा.

$$A^{\bar{\bar{}}} = A$$

DE morgan's theorems (डि मॉर्गन की प्रमेय):-

इसकी equation निम्नलिखित है:-

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

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

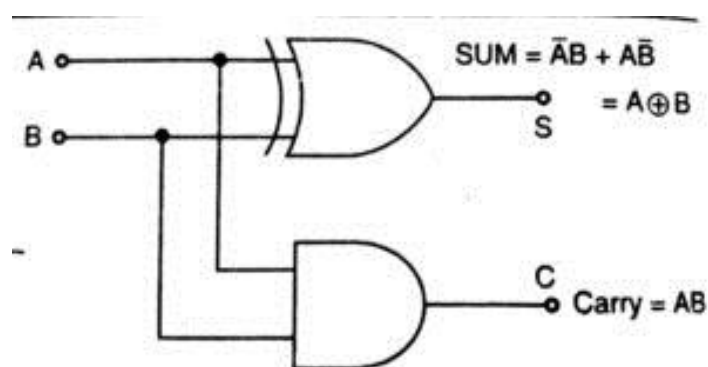
इसकी पहली equation यह दर्शाती है कि NOR गेट जो है वह bubbled AND गेट के समानुपाती होगा.

दूसरी equation यह दर्शाती है कि NAND गेट जो है वह bubbled OR गेट के समानुपाती होगा.

Half adder (हाफ ऐडर क्या है?)

half adder का प्रयोग डिजिटल परिपथों (digital circuits) में गणितीय कार्य के लिए किया जाता है. दो 1 bit संख्याओं को जोड़ने के लिए प्रयुक्त किया जाने वाला लॉजिक परिपथ 'half adder' कहलाता है.

जैसे हम दो binary संख्याओं को जोड़ते हैं उसी प्रकार half adder में भी जोड़ने की क्रिया की जाती है. दो बाइनरी संख्याओं को जोड़ने पर carry उत्पन्न हो सकती है. उदाहरण के लिए 1 एवं 1 को जोड़ने पर योग 0 तथा carry 1 प्राप्त होती है. half adder परिपथ में दो input तथा दो output टर्मिनल होते हैं. आउटपुट टर्मिनल में से एक पर इनपुट में प्रयुक्त संख्याओं का योग (sum) तथा दूसरे पर carry प्राप्त होती है.



उपर चित्र में एक half adder का लॉजिक परिपथ प्रदर्शित किया गया है. इसमें एक ex-OR तथा दूसरा AND gate का प्रयोग किया गया है. Ex-OR की आउटपुट, इनपुट संख्याओं A तथा B का योग ($\bar{A}B + A\bar{B}$) होता है. तथा AND का आउटपुट योग में प्राप्त carry होती है.

half adder truth table

नीचे आपको half adder की truth table दी गयी है.

INPUTS		OUTPUTS	
A	B	SUM	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

truth table से स्पष्ट है कि AND gate की output केवल उस स्थिति में HIGH होती है जब दोनों input A तथा B, HIGH होती है. Ex-OR gate केवल तब HIGH आउटपुट देता है जब दोनों इनपुट में से कोई एक HIGH (दोनों नहीं) होती है.

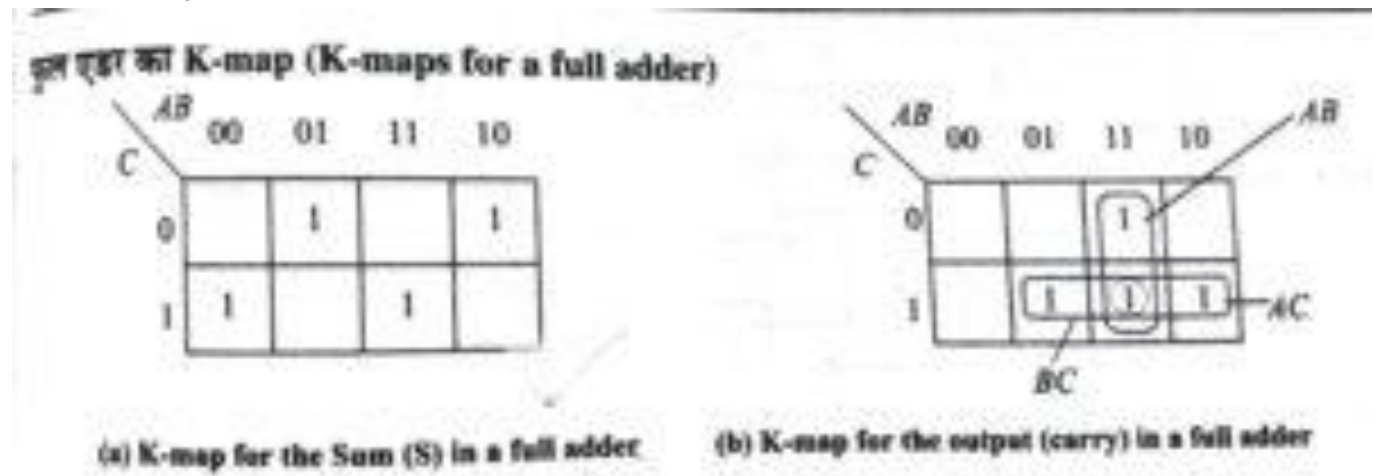
Full adder (फुल ऐडर क्या है?)

half adder में केवल दो इनपुट टर्मिनल होते हैं तथा यदि जोड़ी जाने वाली संख्या में multi bit संख्याएँ हैं तब half adder में निम्न क्रम के bit (lower order bits) से प्राप्त carry को नहीं जोड़ा जा सकता है इसके लिए full adder का प्रयोग किया जाता है. full adder एक समय में 3 bit का योग (sum) कर सकता है. तीसरा bit निम्न क्रम bit के योग से प्राप्त carry होता है.

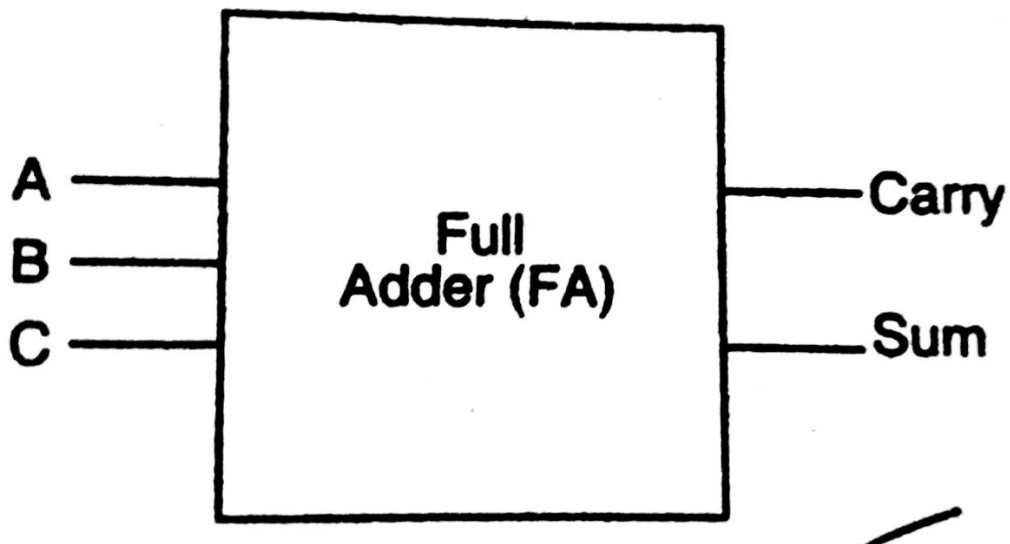
full adder truth table

INPUTS			OUTPUTS	
A	B	C	Carry	Sum
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

K-map for sum and carry in a full adder (फुल ऐडर में sum तथा carry के लिए K-map)



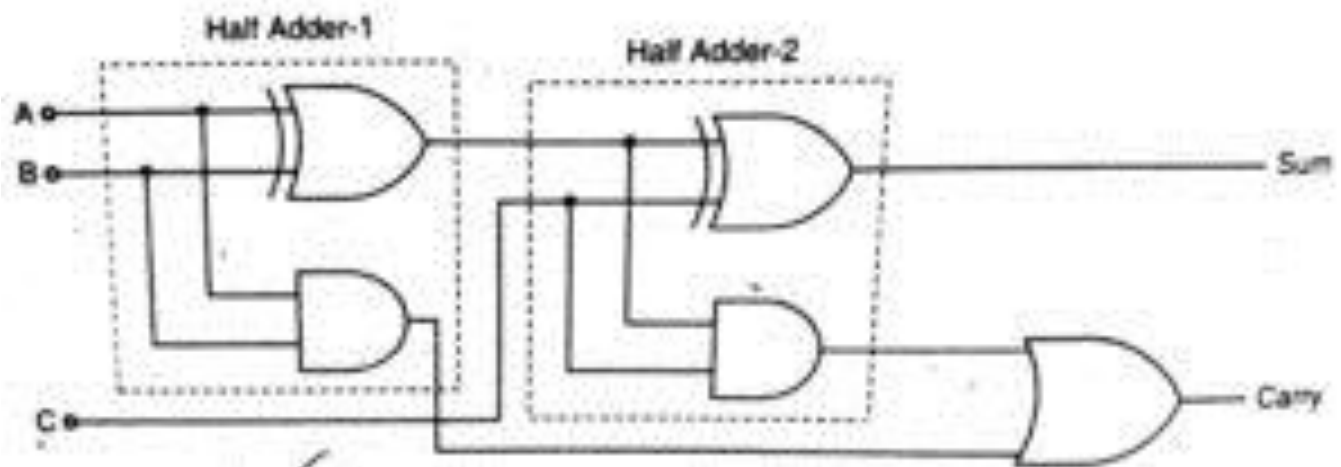
full adder symbol (फुल ऐडर का संकेत)



दो हाफ ऐडर तथा एक OR गेट का प्रयोग कर फुल ऐडर बनाना

(Designing of Full Adder using Half Adder and one OR gate)

दो half adder तथा एक OR gate का प्रयोग करके भी एक full adder बनाया जा सकता है। नीचे आपको चित्र दिया गया है।



half adder-1 को A तथा B इनपुट दी जाती है। इससे प्राप्त sum-1, दूसरे half adder को इनपुट किया जाता है। half adder-2 की दूसरी इनपुट C है। half adder-2 से प्राप्त sum फुल ऐडर की sum आउटपुट के बराबर होता है। पहले हाफ ऐडर से प्राप्त carry तथा दूसरे हाफ ऐडर से प्राप्त carry को OR gate की इनपुट में दिया जाता है। OR gate की आउटपुट full adder की carry होती है।

Half subtractor (हाफ सब्सट्रेक्टर क्या है?)

एक 1-bit संख्या से दूसरी 1-bit संख्या को घटाने के लिए प्रयोग किया गया लॉजिक परिपथ half subtractor कहलाता है।

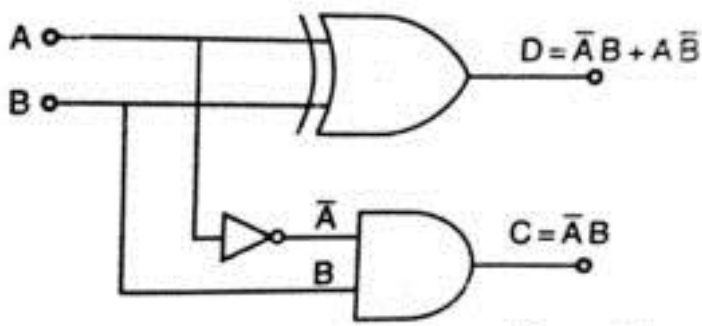
माना दो संख्याएं A तथा B हैं। A को B से घटाने पर अंतर D प्राप्त होता है एवं घटाने की प्रक्रिया में C बोर्रो (borrow) है। इस सम्पूर्ण प्रक्रिया को नीचे truth table में दिखाया गया है।

borrow तब उत्पन्न होता है जब subtrahend (B), minuend (A) से छोटा हो. जैसे truth table की दूसरी लाइन (row) में $A < B$, अर्थात् 1 को शून्य से घटाने पर borrow (1) की आवश्यकता होगी तथा अंतर $10 - 1 = 1$ होगा.

truth table of half **subtractor**

INPUTS		OUTPUTS	
A	B	D (Difference)	C (Borrow)
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

half सबट्रेक्टर का लॉजिक परिपथ (circuit)



यहाँ आपको half subtractor का लॉजिक परिपथ दिया गया है,

Full subtractor – फुल सबट्रेक्टर क्या है?

Half subtractor केवल 1-bit संख्याओं को ही घटा सकता है. 1 से अधिक bit (multibit) की संख्याओं को घटाने के लिए प्रयोग किया जाना वाला परिपथ full subtractor कहलाता है.

full subtractor में 3 इनपुट ABC तथा दो आउटपुट difference (D) एवं borrow होती हैं.

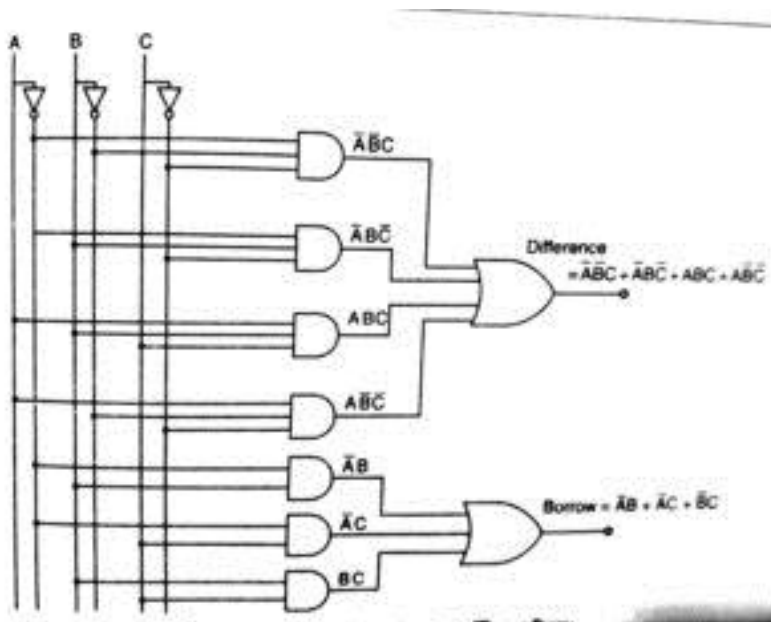
full subtractor truth table

INPUTS			OUTPUTS	
A	B	C	Difference	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

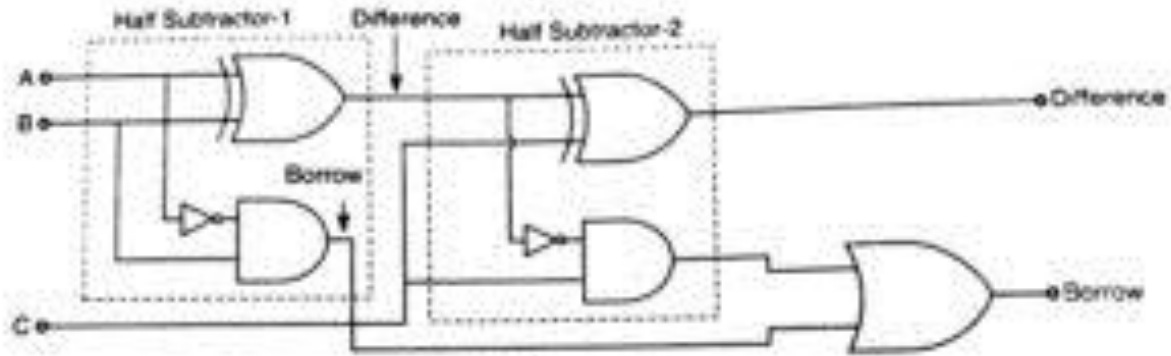
[इमेज](#)

full सबट्रेक्टर का logic circuit (लॉजिक परिपथ)

यह इसका लॉजिक परिपथ का चित्र है:-



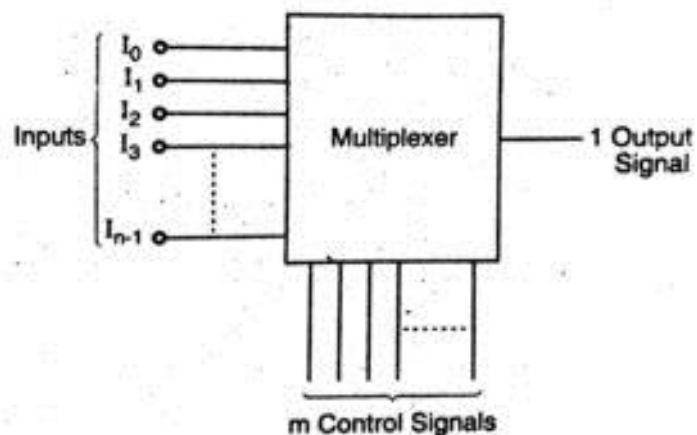
दो half subtractor की सहायता से full subtractor बनाना



Unit-06

Multiplexer in hindi (मल्टीप्लेक्सर क्या है?)

multiplexer एक ऐसा परिपथ (circuit) है जिसमें अनेक इनपुट तथा केवल एक आउटपुट होती है. यह एक data selector परिपथ है. इसमें control signal को प्रयुक्त कर, किसी भी इनपुट को आउटपुट पर प्राप्त किया जा सकता है.



चित्र में एक multiplexer का block diagram दिया गया है. परिपथ (circuit) में n इनपुट सिग्नल, m कंट्रोल सिग्नल तथा केवल एक आउटपुट सिग्नल है. n इनपुट सिग्नलों में से आउटपुट पर कोई एक सिग्नल प्राप्त करने के लिए कंट्रोल सिग्नल की संख्या निम्न समीकरण से प्राप्त की जा सकती है:-

$$[2^m = n]$$

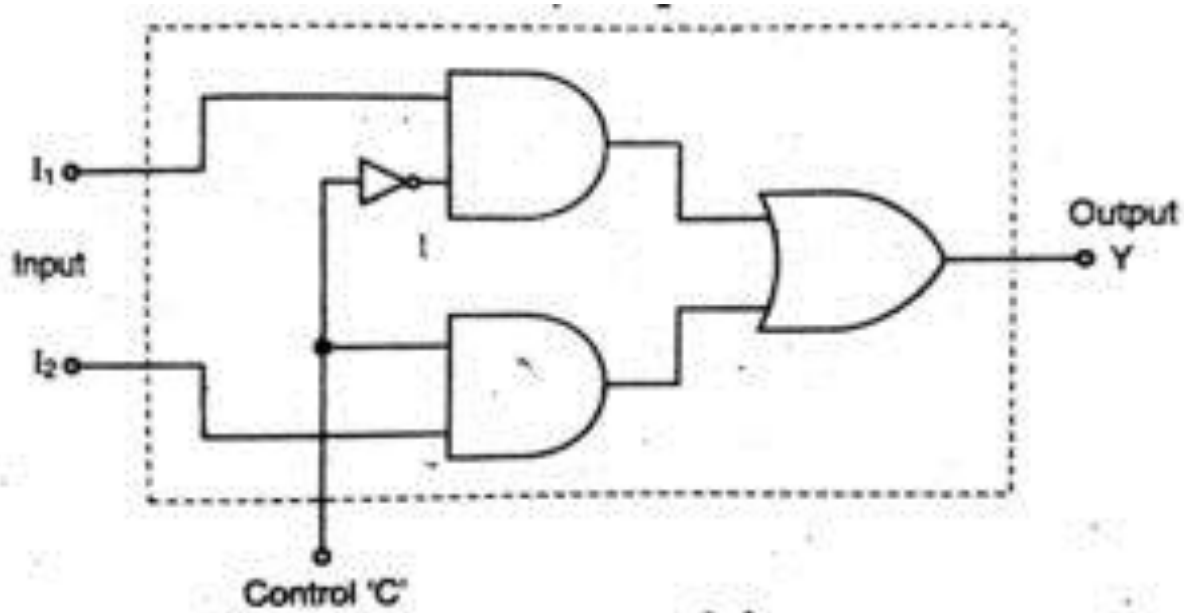
परिपथ की आउटपुट (जो एक selected इनपुट होती है) कंट्रोल टर्मिनल पर प्रयोग किये गये digital code पर निर्भर करती है. multiplexer में प्राय एक स्ट्रॉब (or enable) इनपुट भी होती है जो एक्टिव LOW होने पर परिपथ में वांछित (desired) operation करती है.

2 to 1 multiplexer

इसमें 2 इनपुट सिग्नल होते हैं. इस circuit में किसी एक इनपुट सिग्नल को select किया जाता है तथा उसी selected इनपुट को, आउटपुट पर ट्रान्सफर किया जाता है. इनपुट selection का कार्य control signal करता है.

नीचे आपको 2 to 1 multiplexer का चित्र दिया गया है. चित्र में प्रदर्शित multiplexer में दो इनपुट लाइनें I_1 , I_2 एक कंट्रोल इनपुट C तथा एक आउटपुट लाइन Y है. multiplexer के लॉजिक परिपथ से आउटपुट Y का equation निम्न प्रकार लिखा जाता है:-

$$Y = C'I_1 + CI_2$$



1) यदि $C=0$ है तब $C\bar{I}_2 = 0$

अतः $Y = I_1$

अर्थात् केवल इनपुट I_1 को select किया गया है.

2) यदि $C=1$ है तब $C\bar{I}_1 = 0$

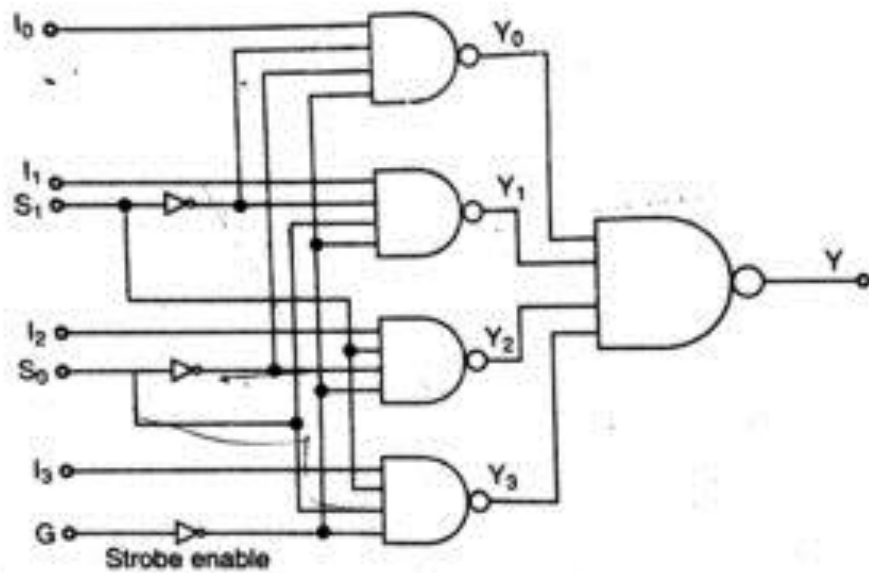
अतः $Y = I_2$

अर्थात् केवल इनपुट I_2 को select किया गया है.

इस प्रकार multiplexer में control signal की सहायता से केवल एक इनपुट को select कर उसे आउटपुट पर transfer किया जाता है.

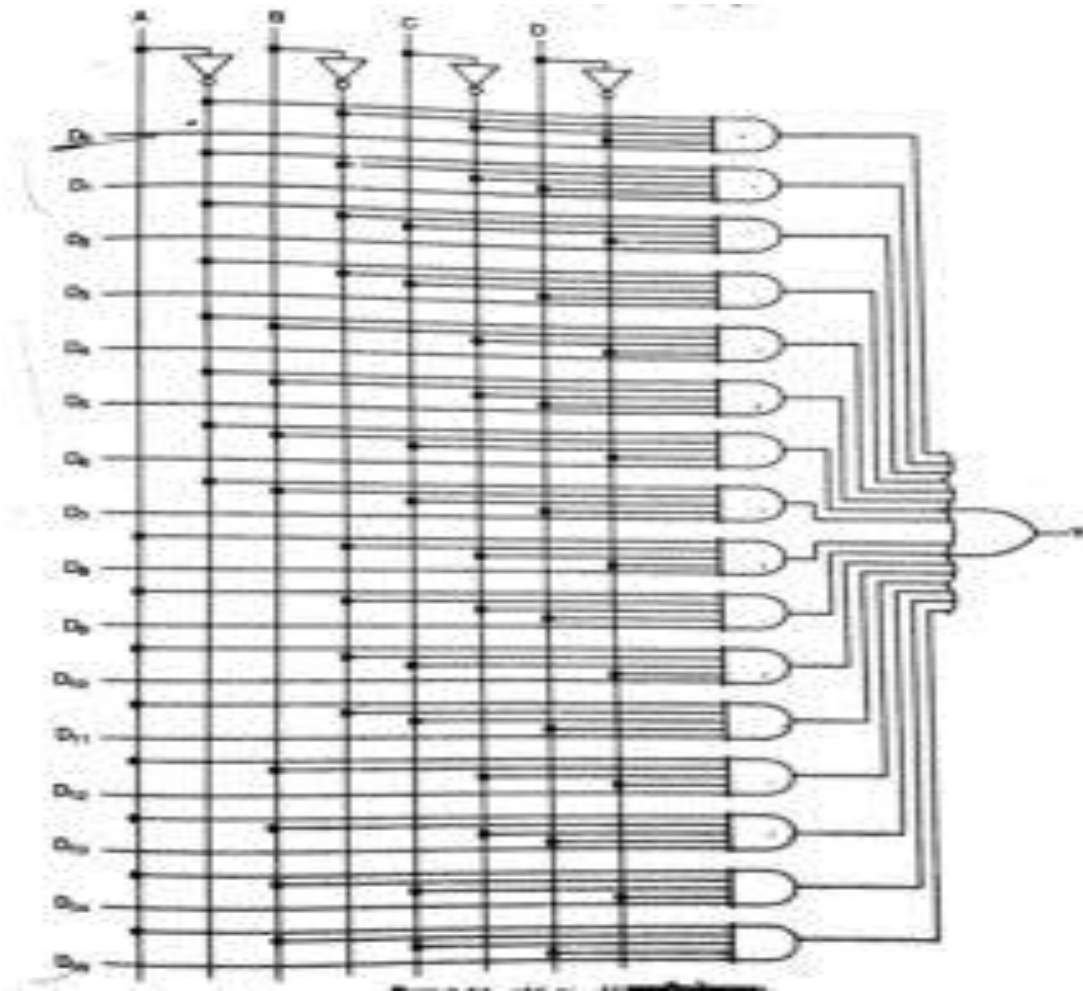
4 to 1 multiplexer

इसमें 4 input signals होते हैं. नीचे चित्र में 4 to 1 multiplexer का लॉजिक परिपथ प्रदर्शित किया गया है. इसमें चार इनपुट सिग्नल I_0, I_1, I_2, I_3 तथा दो control signal अर्थात् selected input S_1, S_2 एवं परिपथ को enable करने के लिए एक स्ट्रोब इनपुट (G) प्रयुक्त की गयी है.



16 to 1 multiplexer

यह एक data selector परिपथ है क्योंकि परिपथ का output bit, इनपुट में से कोई selected bit होता है. इसमें 16 input bits होते हैं.



ऊपर चित्र में एक 16 to 1 multiplexer का परिपथ (circuit) प्रदर्शित किया गया है इस चित्र में मल्टीप्लेक्सर के इनपुट bits $D_0, D_1, D_2, D_3, \dots, D_{15}$ हैं. output bit, इनपुट में से कौन सा bit होगा यह control signal ABCD पर निर्भर करता है.

उदाहरण के लिए-

यदि control signal-

$ABCD = 0000$ है

तब सबसे ऊपर वाला AND गेट enable होता है तथा अन्य सभी gates disable होते हैं. अतः इनपुट बिट D_0 आउटपुट को ट्रान्सफर हो जायेगा. अर्थात् $Y = D_0$

यदि D_0 का status 'LOW' है तब आउटपुट Y भी LOW होगी तथा यदि D_0 'HIGH' है तब Y भी HIGH होगी. अतः Y का मान D_0 पर निर्भर करेगा.

यदि control signal-

ABCD = 1111 है तो

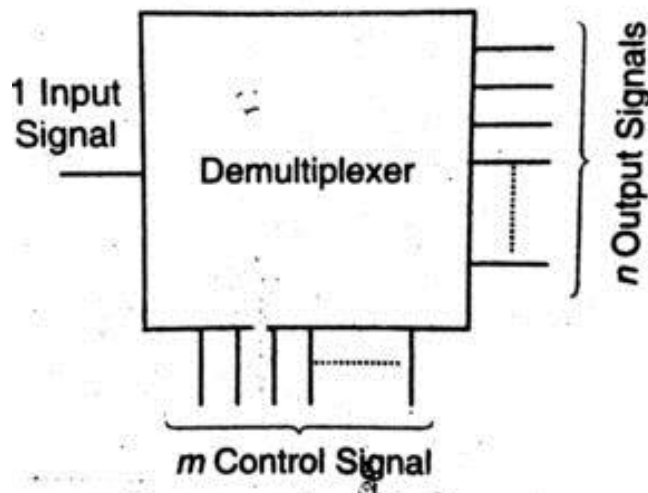
सबसे नीचे वाले AND gate के अतिरिक्त सभी अन्य गेट्स disable हो जायेंगे. तथा केवल D_{15} आउटपुट को ट्रान्सफर होता है.

अतः $Y = D_{15}$.

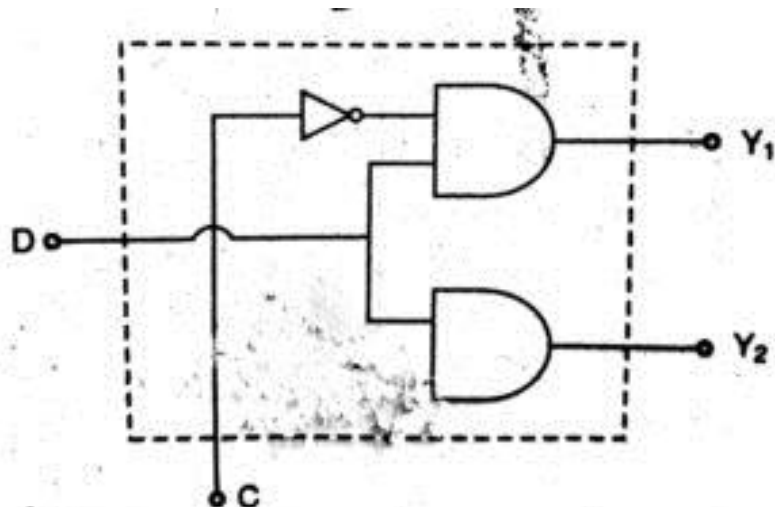
Demultiplexer (डिमल्टीप्लेक्सर क्या है?)

demultiplexer एक ऐसा लॉजिक परिपथ (logic circuit) है जिसमें एक इनपुट तथा अनेक आउटपुट होती है. इसमें control signal को प्रयोग कर input किसी भी output line पर प्राप्त की जा सकती है.

नीचे चित्र में demultiplexer का block diagram दिया गया है. इस परिपथ में एक input signal, m control signal तथा n output signal हैं.



1 to 2 demultiplexer



ऊपर चित्र में एक 1 to 2 demultiplexer का परिपथ प्रदर्शित किया गया है. यदि परिपथ की इनपुट D, control signal C तथा output Y1, Y2 है तब demultiplexer की आउटपुट निम्न प्रकार होगी-

$$Y1 = C^{\neg}D \text{ तथा } Y2 = CD.$$

1) यदि $C = 0$ तब $Y1 = D$

$$Y2 = 0.$$

अर्थात् इनपुट सिग्नल, आउटपुट लाइन Y1 पर प्राप्त होगा.

2) यदि $C = 1$ तब $Y1 = 0$

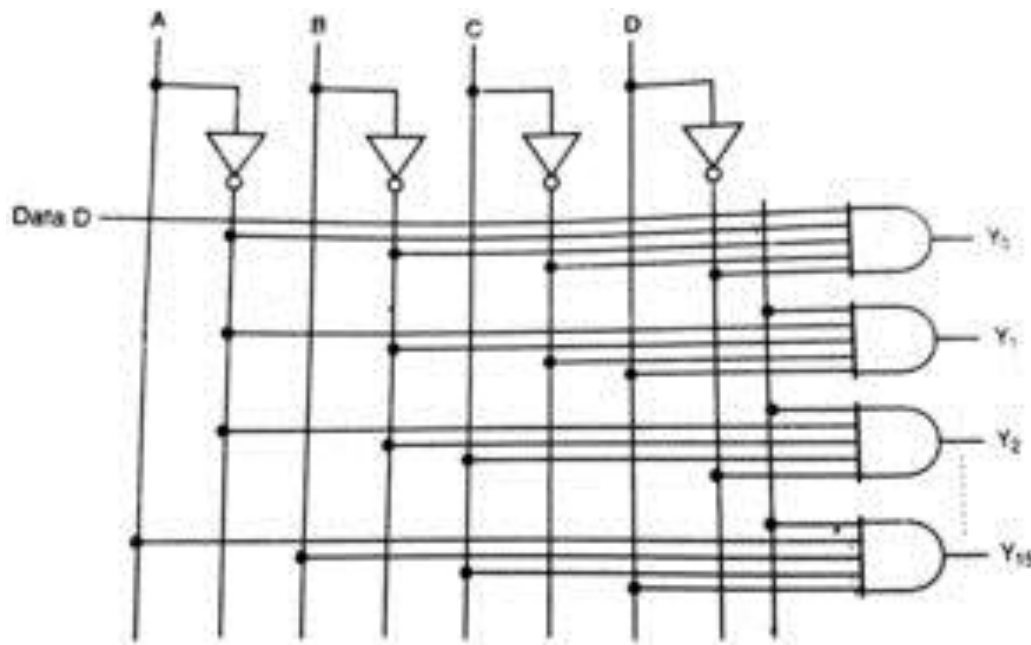
$$Y2 = D.$$

अर्थात् इनपुट सिग्नल, आउटपुट लाइन Y2 पर प्राप्त होता है.

इस प्रकार demultiplexer में एक समय में केवल एक ही आउटपुट ही select होती है किसी भी condition में दोनों आउटपुट एक साथ select नहीं होती है.

1 to 16 demultiplexer

इसमें एक input bit, 4 control input तथा 16 आउटपुट लाइन होती है. परिपथ की आउटपुट कंट्रोल इनपुट पर निर्भर करती है.



ऊपर चित्र में एक 1 to 16 डिमल्टीप्लेक्सर का परिपथ प्रदर्शित किया गया है। इसमें एक input bit D, चार control input A, B, C, D तथा $Y_0, Y_1, Y_2, \dots, Y_{15}$ हैं।

उदाहरण के लिए-

यदि $ABCD = 0000$

तब केवल उपर वाला AND गेट enable होता है तथा अन्य सभी AND गेट्स disable होते हैं।

इसलिए data bit केवल आउटपुट Y_0 पर प्राप्त होगा। अर्थात् $Y_0 = D$ ।

इस प्रकार यदि कंट्रोल इनपुट $ABCD = 1111$ है

तब केवल सबसे नीचे वाले AND गेट के अतिरिक्त अन्य सभी disable होंगे, तथा

$Y_{15} = D$ ।

इस प्रकार डिमल्टीप्लेक्सर, इनपुट डेटा बिट को 16 आउटपुट lines में से किसी एक selected लाइन पर ट्रान्सफर करता है। तथा यह selected आउटपुट लाइन, control signal ABCD पर निर्भर करती है।

Encoder (एनकोडर क्या है) :-

एक **एनकोडर** एक डिजिटल सर्किट है जो एक **डिकोडर** का उलटा Operation करता है। एक **एनकोडर** में 2^n (या उससे कम) इनपुट लाइनें और n आउटपुट लाइनें होती हैं। आउटपुट लाइनें इनपुट मान के अनुरूप बाइनरी कोड उत्पन्न करती हैं। **एनकोडर** का एक उदाहरण octal-to-binary encoder है, इसमें आठ इनपुट हैं, प्रत्येक octal अंकों के लिए एक, और तीन आउटपुट जो Connected बाइनरी नंबर उत्पन्न करते हैं। यह माना जाता है कि किसी भी समय केवल एक इनपुट का मान 1 है; अन्यथा, सर्किट कोई मतलब नहीं है।

Inputs								Outputs		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_2	A_1	A_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

एनकोडर को या गेट्स के साथ Executed किया जा सकता है जिनके इनपुट सीधे truth table से निर्धारित होते हैं। आउटपुट A_2 , 1 यदि इनपुट octal अंक 1 या 3 या 5 या 7 है। अन्य दो आउटपुट के लिए समान Terms लागू होती हैं। इन Terms को बूलियन फ़ंक्शंस द्वारा define किया जा सकता है: **एनकोडर** को तीन या गेट्स के साथ Executed किया जा सकता है।

$$A0 = D1 + D3 + D5 + D7$$

$$A1 = D2 + D3 + D6 + D7$$

$$A2 = D4 + D5 + D6 + D7$$

Decoder – डिकोडर क्या है?

Decoder एक ऐसा लॉजिक परिपथ है जो उसके इनपुट पर दिए गये संकेतों (signals) के प्रत्येक combination की पहचान कर सकता है. इसमें n-input lines लेने पर 2^n output lines होती हैं.

उदाहरण के लिए:- यदि डिकोडर की input में दो बाइनरी लाइनें हैं तब डिकोडर की output में चार लाइनें होंगी.

दूसरे शब्दों में कहें तो, "डिकोडर एक circuit होता है जो कि एक code को signals के समूह में बदल देता है." इसे decoder इसलिए कहा जाता है क्योंकि यह encoder का उल्टा होता है.

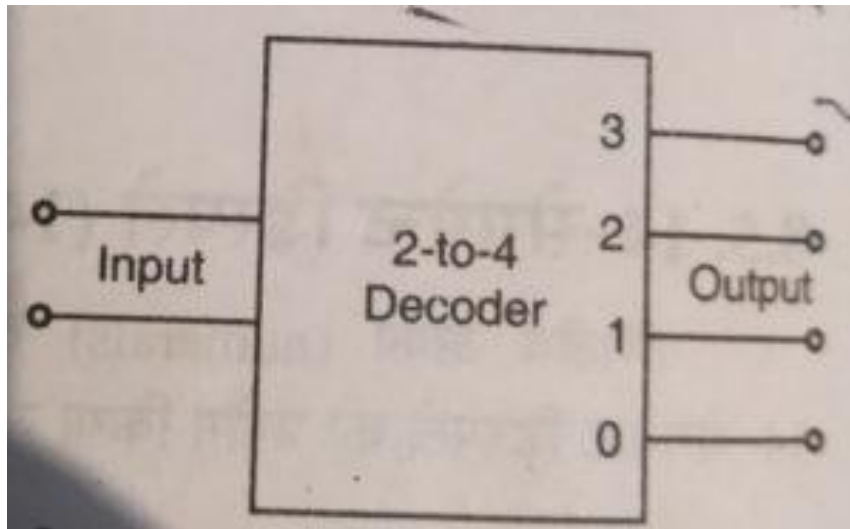
Types of Decoder in Hindi (डिकोडर के प्रकार)

इसके प्रकार निम्नलिखित होते हैं:-

2 to 4 डिकोडर

इसमें दो input lines होती हैं. दो lines से इनपुट सिग्नलों के चार combination बनाये जा सकते हैं – 00, 01, 10, और 11. इसमें से प्रत्येक combination के लिए 4 output lines में से कोई एक लाइन HIGH होती है और बाकी सारी लाइनें low पर होती हैं.

उदाहरण के लिए:- यदि इनपुट 11_2 है तब आउटपुट लाइन '3' लॉजिक '1' पर होती है तथा बची हुई सारी लाइनें लॉजिक '0' पर होती हैं. यह क्रिया decoding कहलाती है.

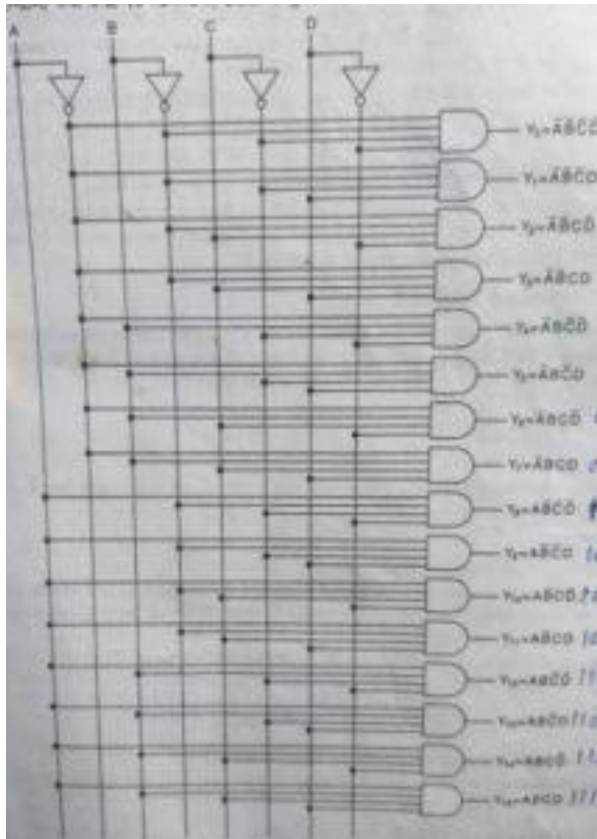


image

ऊपर चित्र में आपको 2 to 4 डिकोडर का symbol दिया गया है. इसे 1 out of 4 डिकोडर भी कहते हैं. क्योंकि इसमें चार output lines में से केवल एक लाइन HIGH पर होती है.

4 to 16 डिकोडर

इसमें 4 input lines होती है तथा 16 output lines होती है.



ऊपर चित्र में इसका लॉजिक परिपथ दिखाया गया है. जिसमें 4 control input (A, B, C, D) तथा 16 output lines हैं.

इस decoder को कोई data input नहीं दिया जाता है बल्कि इनको input केवल control bits के रूप में (ABCD) दी जाती है. इस डिकोडर में input के किसी एक combination के लिए 16 output lines में से कोई एक लाइन 'HIGH' पर होती है और बाकी की बची हुई lines 'LOW' पर होती हैं. इसलिए इस डिकोडर को 1 of 16 decoder भी कहा जाता है

Unit-07

Flip Flop (फ्लिप फ्लॉप क्या है?)

“flip flop एक प्रकार का circuit होता है जिसकी दो states (0 या 1) होती हैं. तथा इसका प्रयोग state information को स्टोर करने के लिए किया जाता है.”

फ्लिप-फ्लॉप एक sequential logic circuit है। जो एक bit store करने के लिए सक्षम होता है। flip flop का output स्थिर (stable) होता है। और यह सिर्फ दो value को ही carry करता है high voltage या low voltage जैसे :- एक output या तो 0 होगा या फिर 1 होगा।

Flip Flop एक बाइनरी स्टोरेज डिवाइस है. यह binary bit 0 या 1 स्टोर कर सकता है. यह एक computer का सबसे छोटा storage unit होता है जैसे :- यह एक single bit (0 या 1) store करता है। external inputs की संख्या flip flop के type के ऊपर निर्भर करती है।

फ्लिप फ्लॉप को stable multi-vibrator भी कहते हैं. इसे सर्वप्रथम William Eccels तथा F.W. Jordan ने 1918 में विकसित किया था.

सामान्यतया दो आउटपुट लाइन flip flops में उपलब्ध होते हैं जो एक दूसरे के विपरीत होते हैं।

Flip Flops के दो main application होते हैं register और counter memory.

Flip Flops में information को store करने का एक operation होता है जिसे triggering कहते हैं। operation को initiate करने के लिए triggering की requirement होती है। triggering दो प्रकार के होते हैं –

1. Edge
2. Level

Difference between Flipflop and Latch

ज्यादातर लोग flip flops और latch में confuse होते हैं ,flip flop और latch दोनों circuit होते हैं और दोनों ही information को store करने का काम करते हैं। Flip Flop और latch में एक अंतर यह होता है कि latch का input जब भी change करते हैं तो output भी change हो जाता है।

जबकि Flip Flop जो है वह latch और clock का combination होता है यह latch और clock से मिलकर बना होता है. यह clock output पर control operation रखता है इसलिए output नहीं change होता है और यह clock के ऊपर निर्भर करता है। Flip flop में clock का उपयोग इसलिए करते हैं ताकि जब input change करे तब flip flop की functionality या internal logic पर कोई फर्क ना पड़े।

Types of Flip Flops

यहाँ चार प्रकार के Flip Flops दिए हैं –

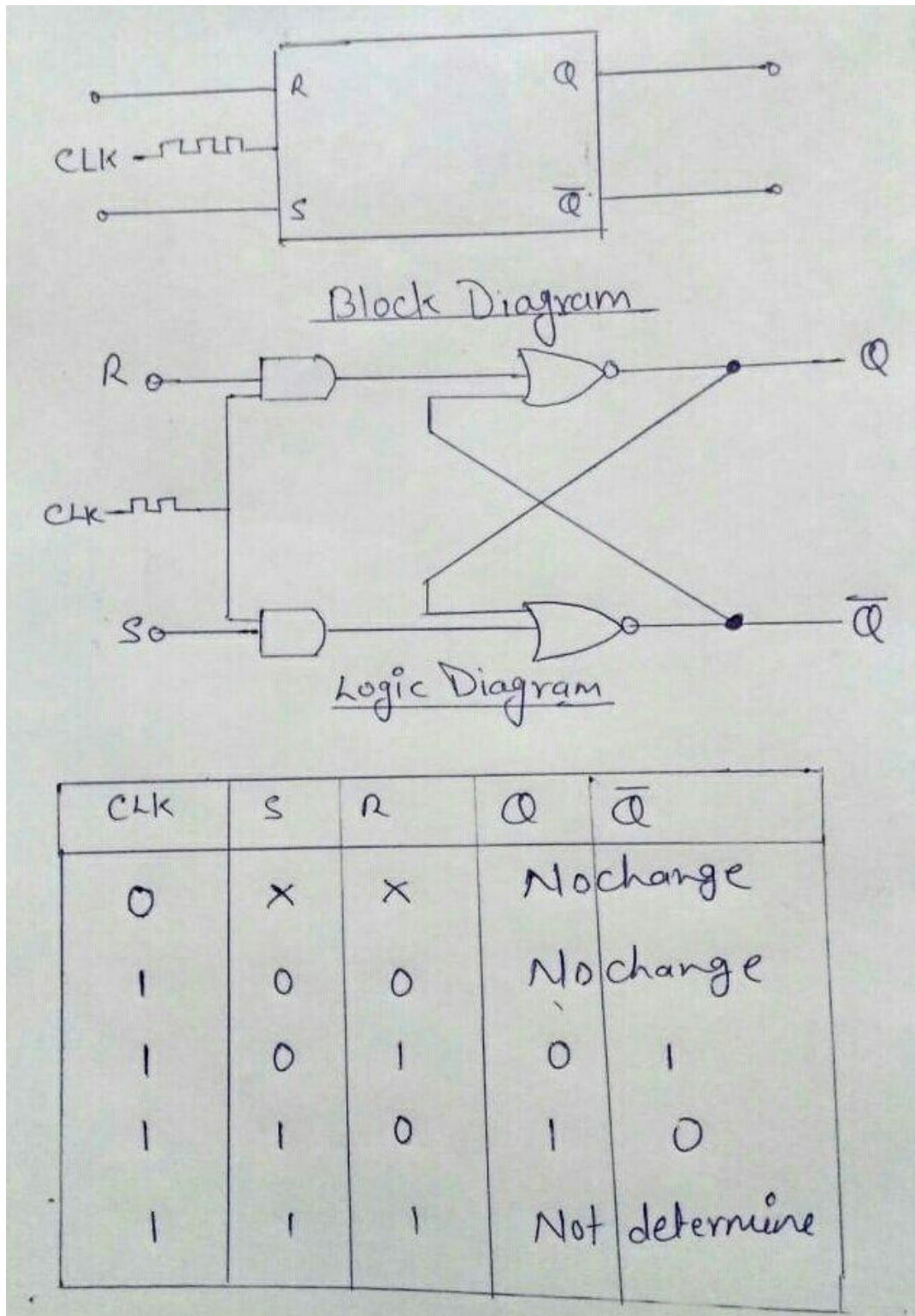
1. R S Flip Flop
2. J K Flip Flop
3. D Flip Flop
4. T Flip Flop

1. R S Flip Flop

R S Flip Flops बेसिक या fundamental Flip Flops होते हैं क्योंकि सभी Flip Flops R S Flip Flop से ही generate (उत्पन्न) होते हैं. जहाँ R S Flip Flop में दो external inputs R (reset) और S (set) होते हैं। जब R active ($R=1$) होता है तो output reset $Q=0$ होता है और जब S active ($S=1$) होता है तो output set $Q=1$ होता है।

R S Flip Flops को Bistable multivibrator भी कहते हैं। एक R S Flip Flop बनाने के लिए दो NAND या NOR gate का use करते हैं।

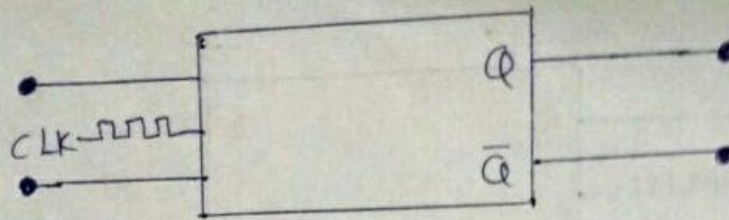
यहाँ हमने दो NOR GATE का उपयोग करके flip flop बनाया है-



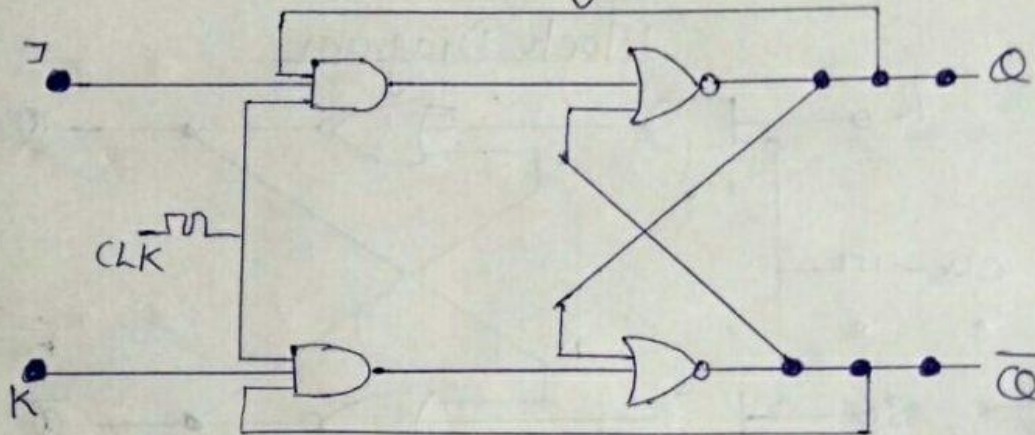
2. JK Flip Flop

JK Flip Flops, RS Flip Flop की एक कमी को पूरा करता है। RS Flip Flop का मुख्य disadvantage यह है कि इसमें जब clock triggered होती है तो दोनों inputs high नहीं होने चाहिए. अर्थात् हम RS Flip Flop में input का एक set (1,1) use नहीं कर सकते हैं। पर JK Flip Flop में हम यह input के set (1,1) को use कर सकते हैं।

इस फ्लिप फ्लॉप का नाम Jack Bilby के नाम पर पड़ा. jack bilby ने IC की खोज की थी.



Block Diagram

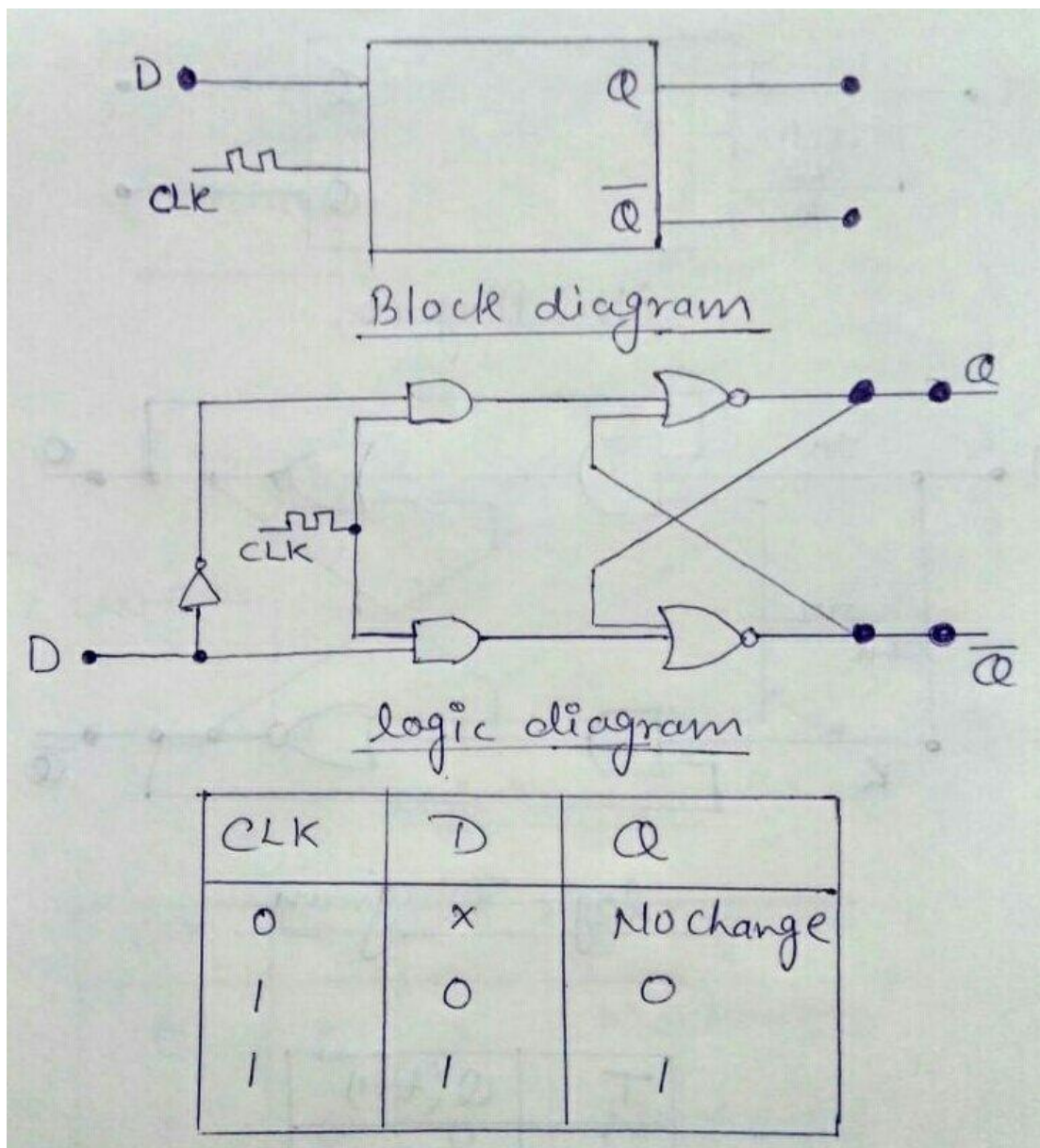


Q	J	K	$Q(t+1)$	
0	0	0	0	Nochange
0	0	1	0	reset
0	1	0	1	set
0	1	1	1	Complement
1	0	0	1	Nochange
1	0	1	0	reset
1	1	0	1	set
1	1	1	0	Complement

3. D Flip Flop

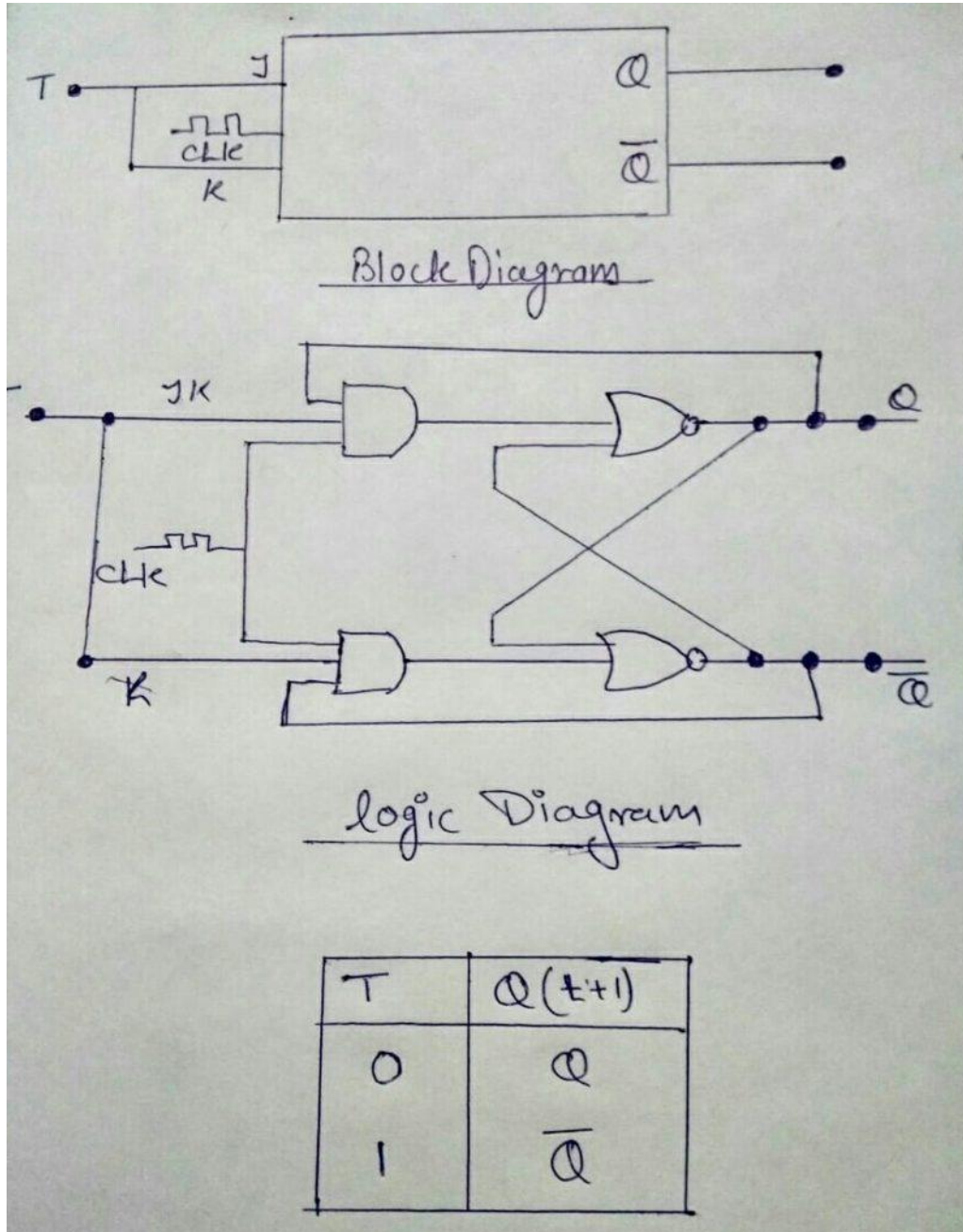
D Flip Flop का प्रयोग delay create करने के लिए करते हैं यदि किसी circuit के अंदर delay की जरूरत पड़ती है तो।

D Flip Flop में जैसा input देते हैं वैसा ही output प्राप्त होता है जैसे:- यदि 1 input दिए तो output भी 1 आएगा या 0 input दिए तो output भी 0 आएगा। D Flip Flop को delayed Flip Flop भी कहा जाता है। यह single input Flip Flop होता है। इसमें कोई race condition भी नहीं होती है।



4. T Flip Flop

T Flip Flop एक single input Flip Flop होता है। यह JK Flip Flop का एक simpler version होता है। T Flip Flop को toggle Flip Flop भी कहते हैं। T Flip Flop का व्यापक रूप से counters में use किया जाता है, यह एक sequential circuit है जो प्रत्येक clock pulse के बाद state change करती है।



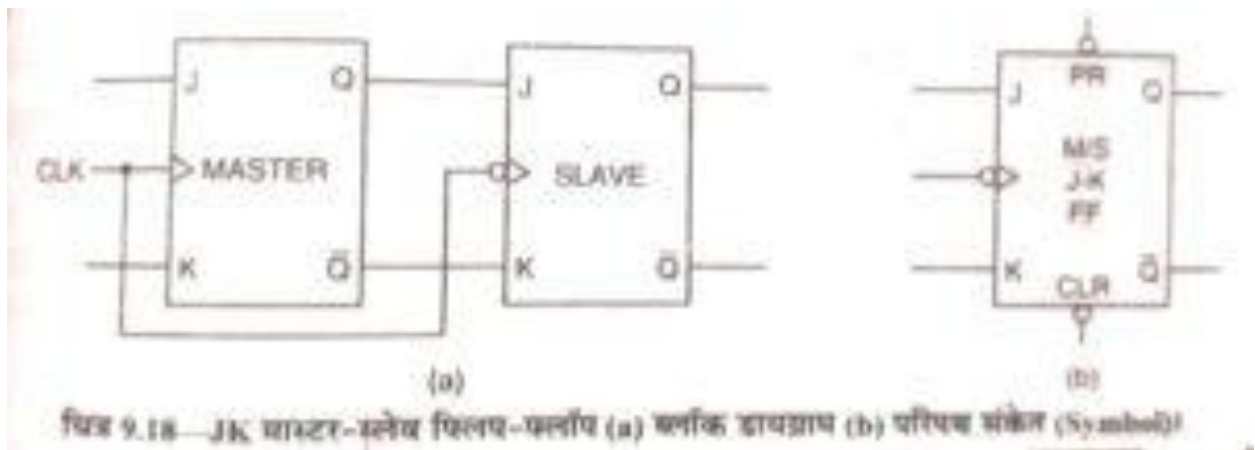
Jk master-slave flip flop

jk master-slave flip flop पढ़ने से पहले हम race around condition के बारे में पढ़ेंगे.

race around condition

JK फ्लिप-फ्लॉप में $J=K=1$ होने पर फ्लिप-फ्लॉप की आउटपुट संदिग्ध और अनिश्चित (ambiguous) होती है. यह race around condition के कारण होता है. इस condition में AND gates को फीडबैक किये गये आउटपुट की state में परिवर्तन के कारण clock pulse की अवधि (duration) में फ्लिप-फ्लॉप में दोलन की स्थिति (oscillatory condition) उत्पन्न हो जाती है. इसलिए जब clock pulse (CP) समाप्त होती है तब आउटपुट कुछ भी (arbitrary) हो सकती है. तथा यह केवल फ्लिप फ्लॉप में propagation delay तथा clock pulse की अवधि पर निर्भर करती है.

JK master-slave flip flop (JK मास्टर-स्लेव फ्लिप-फ्लॉप)



उपर दिए गये चित्र में JK master slave flip flop का ब्लॉक डायग्राम दिया गया है. यह परिपथ flip flop को race around conditions से रोकने का उपाय है. इस परिपथ में दो फ्लिप-फ्लॉप प्रयुक्त किये जाते हैं. एक मास्टर (master) तथा दूसरा slave.

master एक positive edge triggered तथा slave एक negative edge triggered फ्लिप-फ्लॉप होता है. इसलिए master flip flop, slave flip flop से पहले इनपुट J तथा K पर प्रचालित (operate) हो जाता है.

यदि $j = 1$ तथा $K = 0$ है तब master पॉजिटिव clock edge पर SET हो जाता है तथा इसकी (master flip flop की) 'HIGH' आउटपुट (Q), slave flip flop की J इनपुट को drive करती है। इसी प्रकार जब नेगेटिव clock edge आती है तब master की भांति slave SET हो जाता है।

यदि $J = 0$ तथा $K = 1$ तब master flip flop, क्लॉक पल्स की leading edge पर RESET हो जाता है। master की 'HIGH' Q' आउटपुट slave की K इनपुट को drive करती है। इसलिए clock pulse की trailing edge पर slave RESET हो जाता है। slave, इस क्रिया को master की भांति करता है।

यदि master की इनपुट $J=K=1$ अर्थात् J तथा K दोनों HIGH हैं तब यह पॉजिटिव clock edge पर टॉगल (toggle) करता है। तथा slave, नेगेटिव clock edge पर toggle करता है।

इस प्रकार प्रत्येक स्थिति में slave, master की copy करता है अर्थात् यदि मास्टर SET होता है तो slave भी SET होता है यदि मास्टर RESET होता है तब slave भी RESET होता है।

Unit-08

Counter एक डिजिटल डिवाइस है जिसका प्रयोग किसी विशेष event (घटना) को store करने तथा display करने के लिए किया जाता है।

दूसरे शब्दों में कहें तो, "काउंटर एक digital circuit है जिसका प्रयोग pulses को count करने के लिए किया जाता है।"

digital computers और अन्य डिजिटल सिस्टम में किसी निश्चित समय अंतराल में हुए किसी event को record करने के लिए काउंटिंग की आवश्यकता होती है। सामान्यतः एक electronic काउंटर का प्रयोग किसी निश्चित समय अंतराल में input line में आने वाली pulses की counting के लिए होता है। काउंटर में memory होना बहुत आवश्यक है क्योंकि इसे अपनी पिछली अवस्थाओं (states) को याद रखना होता है। काउंटर synchronous या asynchronous हो सकते हैं।

Types of Counter

1. Asynchronous Counter or Ripple Counter
2. Synchronous Counter or Parallel Counter
3. Mod N Counter
4. Decade counter

1.Asynchronous Counter or Ripple Counter

इस Counter को Serial Counter भी कहते हैं क्योंकि ये flip-flop serial में संयोजित रहते हैं तथा न्यूनतम bit वाले (lower significant bit) वाले flip-flop की छोड़कर बाकी प्रत्येक flip-flop अपने से पहले वाले flip-flop का output होता है

संरचना (structure) की दृष्टि से यह Counter अति simple होता है जिससे इसके निर्माण में कम hardware की आवश्यकता होती है इस counter की speed भी कम होती है

for example :- माना प्रत्येक flip-flop की संरचना (propagation delay time; t_p) 20 n sec है तो 4 flip-flop प्रयुक्त counter का सम्पूर्ण (propagation delay time; t_p) 80 n sec हो जायेगा

चुकी इस प्रकार का counter में flip-flop trigger होता है जो जल की तरंग की भाँति गति करता है तो इस प्रकार के counter को ripple counter भी कहते हैं

ii.Synchronous or Parallel Counter

इसमें सभी flip-flop प्रयुक्त clock pulse के द्वारा एक साथ (in synchronism) trigger होता है

अतः इस Counter के स्थिर होने का समय केवल flip-flop के (Propagation delay time ; t_p) के बराबर होता है

चुकी इसमे flip-flop समान्तर में संयोजित रहते हैं इस लिए इसे समान्तर counter भी कहते हैं इस काउन्टर की गति तो उच्च होती है परन्तु जटिल होने के कारण अधिक hardware की आवश्यकता होती है

Decade counter

ऐसा काउन्टर जिसकी N वैल्यू 10 हो उस काउन्टर को decade काउन्टर कहते हैं. यहाँ 4 फ्लिप फ्लॉप का होना जरूरी है क्योंकि $2^4=16>10$ और $n=4$ है. इसमें 10 के काउंट पर दूसरे और चौथे फ्लिप फ्लॉप कि आउटपुट 1 है तो इन दोनों कि आउटपुट को NAND gate के इनपुट पर दे दिया जाता है. और NAND gate की आउटपुट को सारे फ्लिप फ्लॉप पर apply कर दिया जाता है और सारे फ्लिप फ्लॉप जीरो पर reset हो जाते हैं.

Unit-09

SHIFT REGISTER (शिफ्ट रजिस्टर क्या है?)

शिफ्ट रजिस्टर flip-flop से मिलकर बना होता है इसमें फ्लिप फ्लॉप इस प्रकार chain में लगे हुए होते हैं कि प्रत्येक फ्लिप फ्लॉप की आउटपुट अगले फ्लिप फ्लॉप की इनपुट होती है. और इस प्रकार व्यवस्थित होने के कारण ये एक ऐसा circuit बनता है जो इसमें स्टोर bit array को एक position shift करता है.

शिफ्ट रजिस्टर के सारे फ्लिप फ्लॉप एक ही clock द्वारा trigger होते हैं. n शिफ्ट रजिस्टर bit array को left या right shift कर सकते हैं.

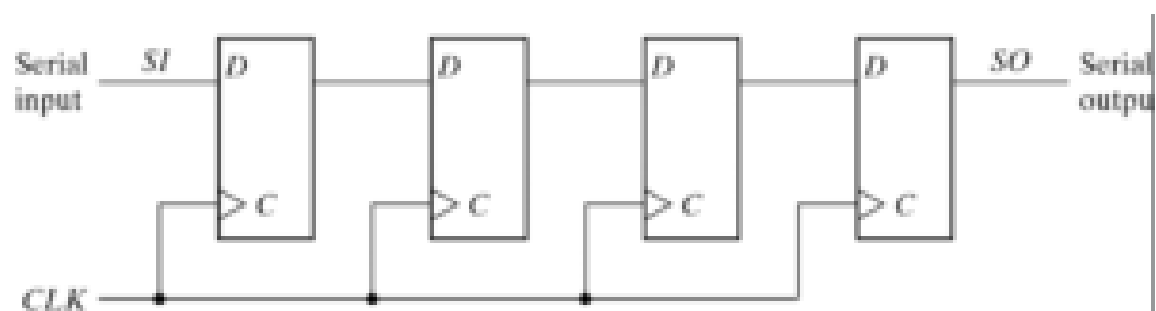


Fig. 6-3 4-Bit Shift Register

चित्र

Shift left register

shift register में clock pulse apply करके data को left या right शिफ्ट किया जा सकता है. shift left रजिस्टर में, जैसा कि नाम से ही पता चल रहा है, data को एक स्थान बांये शिफ्ट किया जाता है.

माना कि हमारे पास data $D = D_1 D_2 D_3 D_4$ है अब इसमें शिफ्ट left करने पर D_4 D_3 के स्थान पर, D_3 D_2 के स्थान पर और D_2 D_1 के स्थान पर आ जाता है. यहाँ पर D_1 lost हो जाता है और D_4 के स्थान पर इनपुट पर जो bit है वो आ जाती है. इनपुट bit 1 या 0 हो सकती है.

माना हमारे पास data $D = 1101$ है. इस पर left शिफ्ट करने पर यह 1010 हो जाती है (जब इनपुट data bit 0 हो). इसमें LSB के स्थान पर इनपुट किया गया data आ जाता है.

Shift right register

shift right रजिस्टर में data को दांये शिफ्ट किया जाता है. शिफ्ट right operation में data 1 स्थान दांये शिफ्ट होता है.

माना हमारे पास data $D=0001$ है अब अगर इनपुट data bit 1 हो तो यह data शिफ्ट right operation के बाद $D=1000$ हो जायेगा. इसमें MSB के स्थान पर इनपुट किया गया data आ जाता है.

शिफ्ट रजिस्टर में data को इनपुट करने की दो विधियाँ होती हैं-

- 1). serial
- 2). parallel

इसी प्रकार data को out करने की भी दो विधियाँ होती हैं-

- 1). serial
- 2). parallel

data in और data out करने की इन विधियों के आधार पर shift register चार प्रकार के होते हैं-

- 1). serial in serial out (SISO)
- 2). serial in parallel out (SIPO)
- 3). parallel in parallel out (PIPO)
- 4). parallel in serial out (PISO)

serial in serial out (SISO)

इस रजिस्टर में data को in तथा out, serial में किया जाता है. इसमें एक समय पर एक bit को left या right शिफ्ट किया जाता है. serial in serial out (SISO) रजिस्टर में यदि चौथे फ्लिप फ्लॉप की आउटपुट पहले फ्लिप फ्लॉप की इनपुट में दे दी जाये तो यह एक रिंग काउंटर की तरह काम करेगा और हम इसे रिंग काउंटर की तरह प्रयोग कर सकते हैं.

serial in parallel out (SIPO)

इस रजिस्टर में data को serial में in किया जाता है. एक bit एक समय में load होती है और data parallel में out होता है. data parallel में out होने के लिए सभी bits को एक समय पर उपलब्ध होना होता है तो जब तक सभी bits आउटपुट के लिए उपलब्ध न हों तब तक data को स्टोर किया जाता है.

parallel in parallel out (PIPO)

इस रजिस्टर में data parallel में in किया जाता है और same clock pulse में parallel में out कर दिया जाता है.

IC 7495–

IC 7495 में 4 बिट shift register का प्रयोग होता है. यह parallel in parallel out शिफ्ट रजिस्टर होता है. इसमें शिफ्ट लेफ्ट शिफ्ट राईट serial input भी हो सकता है.

parallel in serial out (PISO)

इस रजिस्टर में data parallel में load किया जाता है और एक bit एक समय में serial में out कर दी जाती है.