



महाराष्ट्र राज्य तंत्रशिक्षण मंडळ, मुंबई  
(स्वायत्त) (ISO 9001:2015) (ISO/IEC 27001:2013)

अभियांत्रिकी आणि तंत्रज्ञान पदविका

शिक्षण पुस्तिका  
(Learning Material)

# **DIGITAL TECHNIQUES**

## **(22320)**

अणुविद्युत अभियांत्रिकी गट

मराठी-इंग्रजी (द्विभाषिक) माध्यम  
(अभियांत्रिकी व तंत्रज्ञानातील तृतीय सत्र पदविका)



शिक्षण पुस्तिका  
(Learning Material)

# **DIGITAL TECHNIQUES**

## **(22320)**

अणुविद्युत अभियांत्रिकी गट

मराठी-इंग्रजी (द्विभाषिक) माध्यम  
(अभियांत्रिकी व तंत्रज्ञानातील तृतीय सत्र पदविका)



महाराष्ट्र राज्य तंत्रशिक्षण मंडळ, मुंबई  
(स्वायत्त) (ISO 9001:2015) (ISO/IEC 27001:2013)

## मार्गदर्शक

**अकोले किशोर प्रल्हाद**

प्रभारी विभागप्रमुख, अणुविद्युत अभियांत्रिकी

## लेखक

**गवळी विक्रम संपत**

अधिव्याख्याता, अणुविद्युत अभियांत्रिकी

**पाटील प्राजक्ता तानाजी**

अधिव्याख्याता, अणुविद्युत अभियांत्रिकी

**चव्हाण शामली विजय**

अधिव्याख्याता, अणुविद्युत अभियांत्रिकी

**एस. एस. पाटील**

अधिव्याख्याता, अणुविद्युत अभियांत्रिकी

**खोत शुभांगी प्रशांत**

अधिव्याख्याता, विद्युत अभियांत्रिकी





# महाराष्ट्र राज्य तंत्र शिक्षण मंडळ

(स्वायत्त) (ISO ९००१:२०१५) (ISO/IEC २७००१:२०१३)

शासकीय तंत्रनिकेतन इमारत, ४ था मजला, ४९, खेरवाडी, वांद्रे (पूर्व), मुंबई - ४०० ०५१

दू.क्र.: ०२२-६२५४२१००/१०१/१०२



संकेतस्थळ : [www.msbte.org.in](http://www.msbte.org.in)

ई-मेल : [director@msbte.com](mailto:director@msbte.com)

## प्रास्ताविक

महाराष्ट्र राज्यातील पदविका स्तरावरील तंत्रशिक्षणाशी संबंधित बाबींचे नियमन करण्यासाठी महाराष्ट्र राज्य तंत्रशिक्षण मंडळ वचनबद्ध असून विद्यार्थ्यांच्या सर्वांगीण विकासाकरिता वेळोवेळी प्रयत्नशील आहे. तंत्रज्ञान, उद्योग, समाज आणि जागतिकीकरण यामध्ये सतत घडून येणा-या बदलांच्या अनुषंगाने तांत्रिक शिक्षणाची भविष्यातील निकड वेधून पदविका स्तरावरील अभ्यासक्रम, परीक्षा पद्धती व शैक्षणिक सामुग्री ह्यांमध्ये अद्ययावत बदल करण्यात महाराष्ट्र राज्य तंत्रशिक्षण मंडळ अग्रगण्य आहे. विद्यार्थी हा शिक्षण क्षेत्राच्या केंद्रस्थानी असून त्यांची निकड व समस्या संवेदनशीलपणे हाताळल्यास भारत देशाचे 'ज्ञान महासत्ता' बनण्याचे स्वप्न पूर्णत्वास जाईल ह्याचा मला विश्वास आहे.

शहर आणि ग्रामीण भागातील शैक्षणिक सोयीसुविधांमधील दरी अनेक वेळा दिसून येत असून ग्रामीण भागातील विद्यार्थ्यांचे इंग्रजी भाषेतील ज्ञान व संवाद कौशल्याबाबतही ही वस्तुस्थिती प्रकर्षाने जाणवते. केवळ इंग्रजी भाषेतील संवाद कौशल्याअभावी ग्रामीण भागातील विद्यार्थी तंत्रशिक्षणापासून वंचित राहू नये, ह्या दृष्टिकोनातून महाराष्ट्र राज्य तंत्रशिक्षण मंडळाने शैक्षणिक वर्ष २०२१-२२ पासून प्रथम वर्ष पदविका अभ्यासक्रमाकरिता तांत्रिक शिक्षण मराठी-इंग्रजी द्विभाषिक माध्यमात इच्छुक विद्यार्थ्यांना उपलब्ध करून दिले आहे. मात्र असे करताना कोणत्याही परिस्थितीत गुणवत्तेशी तडजोड केली जाऊ नये ह्या दृष्टीने प्रमुख विषयांसाठीच्या शैक्षणिक सामुग्रीची निर्मिती करण्यात आली आहे.

राष्ट्रीय शिक्षण धोरण २०२० मध्ये प्रादेशिक भाषांमध्ये सर्वांना शिक्षणाची कल्पना मांडण्यात आली आहे. त्यास अनुसरून मराठी-इंग्रजी द्विभाषिक माध्यमाचा पर्याय द्वितीय व तृतीय वर्षाकरिताही उपलब्ध करून देण्यात आला आहे. तसेच त्याकरिता शैक्षणिक सामुग्रीही विद्यार्थी व अध्यापकांना उपलब्ध करून देण्यात येत आहे.

महाराष्ट्र राज्यातील अनुभवी अध्यापकांकरवी ही शैक्षणिक सामुग्री तयार करण्यात आली असून व्यावहारिक मराठी भाषा, इंग्रजी भाषेतील तांत्रिक शब्दावलीचा उपयोग आणि संदर्भ पुस्तके लक्षात घेऊन या सामुग्रीची निर्मिती करण्यात आलेली आहे. सदर सामुग्रीची पुनर्तपासणी सुकाणू समितीमार्फत करण्यात आलेली असल्याने ही शैक्षणिक सामुग्री अधिक समृद्ध झालेली आहे. त्यामुळे विद्यार्थ्यांना तांत्रिक शिक्षण समजून घेणे अधिक सुकर होईल. तसेच व्यावहारिक मराठी भाषेच्या उपयोगाने विद्यार्थ्यांना विषयाचे सखोल आकलन होईल व इंग्रजी भाषेतील तांत्रिक शब्दावलीच्या वापरामुळे विद्यार्थ्यांचा उद्योग जगतातील वावर सुलभ होईल. त्यामुळे महाराष्ट्र राज्य तांत्रिक क्षेत्रातील वैश्विक मनुष्यबळाच्या निर्मितीत अग्रेसर राहील व त्यायोगे राष्ट्रनिर्मितीकरीता निश्चितच हातभार लागेल असा मला विश्वास आहे.

अभियांत्रिकी पदविका अभ्यासक्रमातील प्रमुख विषयांची मराठी-इंग्रजी द्विभाषिक शैक्षणिक सामुग्री बनविण्यासाठी अध्यापक व सुकाणू समितीचे सदस्य हे कौतुकास पात्र असून मी त्यांचे अभिनंदन करतो.

( डॉ. विनोद म. मोहितकर )

संचालक,

महाराष्ट्र राज्य तंत्रशिक्षण मंडळ, मुंबई



## अनुक्रमणिका

अ. क्र.	युनिटचे नाव	पान क्र.
1	संख्या प्रणाली आणि कोड (Number System and Codes)	1-29
2	लॉजिक गेट्स (तार्तिक द्वार) लॉजिक फॅमिलीज (Logics Gates and Logic Families)	30-65
3	संयुक्त तार्किक मंडल (Combinational Logic Circuits)	66-119
4	अनुक्रमिक लॉजिक सर्किट्स (Sequential Logic Circuits)	120-181
	डेटा रूपांतरण आणि पीएलडीएस (Data converters and PLDs)	182-221



**युनिट 1**  
**संख्या प्रणाली आणि कोड**  
**(Number systems and Codes)**

**विषय निष्पत्ती (Course Outcome):**

डिजिटल सिस्टमचे कार्य समजण्याकरिता संख्या प्रणाली आणि कोड वापरणे.

**युनिट निष्पत्ती (Unit Outcomes):**

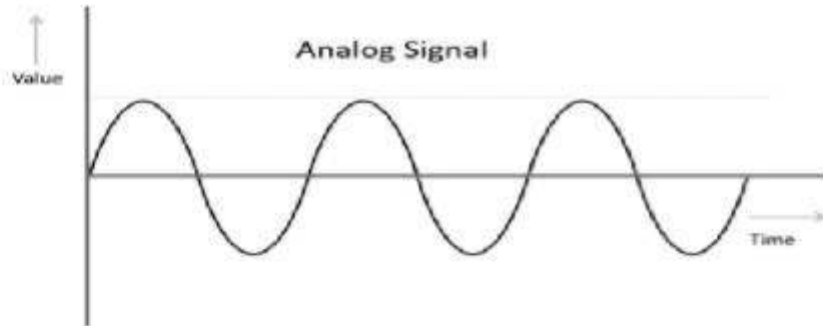
- 1.a दिलेल्या संख्येला निर्दिष्ट (specified) संख्या प्रणालीमध्ये रूपांतरित करणे.
- 1.b दिलेल्या बायनरी नंबरवर बायनरी अंकगणितीय ऑपरेशन करणे.
- 1.c दिलेल्या सांकेतिक संख्येला निर्दिष्ट (specified) सांकेतिक संख्ये मध्ये रूपांतरित करणे.
- 1.d BCD कोड चा वापर करून दिलेले BCD सांकेतिक संख्येची बेरीज करणे.

**1.1 परिचय (Introduction):-**

**अॅनालॉग (Analog) आणि डिजिटल (Digital) संदेश (Signal)**

**1.1.1 अॅनालॉग सिग्नल (Analog signal):**

अॅनालॉग सिग्नल हा एक सतत (Continuous) तरंग आहे जो वेळेनुसार बदलतो.



**आकृती 1.1 अॅनालॉग सिग्नल**

एक अॅनालॉग सिग्नल आकृती 1.1 मध्ये एक साइन वेव्ह (sine wave) आहे. अॅनालॉग सिग्नलचे वर्णन विपुलता (Amplitude), कालावधी (Time) किंवा वारंवारता (Frequency) आणि फेज (Phase) वापरून वर्णन केले जाते. विपुलता सिग्नलची कमाल उंची दर्शवते. वारंवारता सिग्नल ज्या दराने बदलत आहे ते दर्शवते. फेज वेळेच्या संदर्भात सिग्नलची स्थिती दर्शवते. अॅनालॉग सिग्नलवर बाह्य नोईज(Noise) सिग्नल चा परिणाम होऊ शकतो नोईज सिग्नल मुळे त्यात अधिक विरूपण (distortion) येते आणि त्याची गुणवत्ता कमी होते. अॅनालॉग सिग्नलमधील विपुलतेची मर्यादा (Amplitude Range) निश्चित केलेली नसते.



### अॅनालॉग सिग्नलचे फायदे:

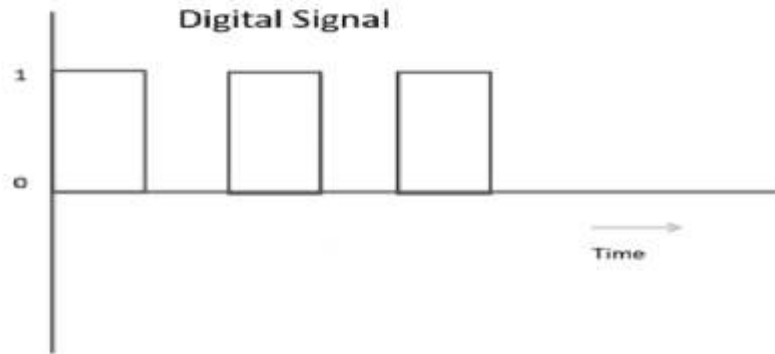
- i) प्रक्रिया करणेस सोपे असतात.
- ii) ऑडिओ आणि व्हिडिओ ट्रान्समिशनसाठी सर्वात योग्य असतात.
- iii) डिजिटल सिग्नलपेक्षा कमी बँडविड्थ (Bandwidth) वापरतात.
- iv) ध्वनीसिग्नल अॅनालॉग सिग्नल ने अधिक अचूक प्रतिनिधित्व प्रदान करतात.

### अॅनालॉग सिग्नलचे तोटे:

- i) अॅनालॉगमध्ये डिजिटलपेक्षा कमी गुणवत्तेचे सिग्नल असतात.
- ii) अॅनालॉग सिग्नल बाह्य प्रभावांबद्दल (Noise) संवेदनशील असतात.
- iii) डिजिटल सिस्टम मध्ये अॅनालॉग सिग्नल वर सरळ प्रक्रिया करता येत नाही.
- iv) अॅनालॉग सिग्नल संक्रमित(transmit) करणे खूप कठीण आहे.
- v) अॅनालॉग सिग्नल संग्रहित करणे खूप कठीण आहे.

### 1.1.2 डिजिटल सिग्नल (Digital signal):

अॅनालॉग सिग्नलप्रमाणेच डिजिटल सिग्नलमध्येही माहिती असते, परंतु ती अॅनालॉग सिग्नलपेक्षा वेगळी असते. डिजिटल सिग्नल हा स्वतंत्र वेळ (discrete time) सिग्नल आहे. आकृती 1.2 मध्ये डिजिटल सिग्नल दर्शविला आहे.



आकृती 1.2 डिजिटल सिग्नल

डिजिटल सिग्नल बायनरी स्वरूपात माहिती किंवा डेटा वाहून नेतो म्हणजे डिजिटल सिग्नल 0 आणि 1 बिटच्या स्वरूपात माहितीचे प्रतिनिधित्व (Representation) करतो. डिजिटल सिग्नलचे वर्णन बिट रेट (bit rate) आणि बिट इंटर्वलने (Bit Interval) केले जाते.

बिट इंटर्वल एक बिट पाठवण्यासाठी लागणारा वेळ दर्शवतो. दुसरीकडे, बिट रेट (bit rate) बिट इंटर्वलच्या वारंवारता दर्शवतो. डिजिटल सिग्नल हा बाह्य नॉइज सिग्नल ला अधिक प्रतिकारक्षम असतो, त्यामुळे त्यात कमी विरूपण (distortion) असते.

डिजिटल सिग्नल प्रसारित करणे सोपे आहे आणि अॅनालॉग सिग्नलच्या तुलनेत ते अधिक विश्वासार्ह (Reliable) आहे. डिजिटल सिग्नलमध्ये विपुलता (Amplitude) 0 आणि 1 या बिट्स करिता मर्यादित असते.

**डिजिटल सिग्नलचे फायदे:**

- डिजिटल डेटा सहजपणे कॉम्प्रेस (compress) केला जाऊ शकतो.
- डिजिटल फॉर्ममधील कोणतीही माहिती एन्क्रिप्ट (Encrypt) केली जाऊ शकते.
- डिजिटल सिग्नल वापरणारे उपकरण अधिक सामान्य आणि कमी खर्चिक आहे.

**डिजिटल सिग्नलचे तोटे:**

- सॅम्पलिंगमुळे माहितीचे नुकसान होऊ शकते.
- राउंड-ऑफ त्रुटी (error) निर्माण करतात.
- यासाठी अधिक बँडविड्थ आवश्यक आहे.

### 1.1.3 अॅनालॉग सिग्नल आणि डिजिटल सिग्नलमधील तुलना:

तक्ता 1.1. अॅनालॉग सिग्नल आणि डिजिटल सिग्नलमधील तुलना

अॅनालॉग सिग्नल	डिजिटल सिग्नल
अॅनालॉग सिग्नल हा एक सतत (Continuous) तरंग आहे जो वेळेनुसार बदलतो.	डिजिटल सिग्नल ही एक वेगळी तरंग आहे जी बायनरी स्वरूपात माहिती घेऊन जाते.
अॅनालॉग सिग्नल साइन वेव्हद्वारे दर्शविला जातो.	डिजिटल सिग्नल स्क्वेअर वेव्हद्वारे दर्शविला जातो.
अॅनालॉग सिग्नलचे वर्णन विपुलता (Amplitude), कालावधी (Time) किंवा वारंवारता (Frequency) आणि फेज (Phase) वापरून वर्णन केले जाते.	डिजिटल सिग्नलचे वर्णन बिट रेट (bit rate) आणि बिट इंटरवलने (Bit Interval) केले जाते.
अॅनालॉग सिग्नलमधील विपुलतेची मर्यादा (Amplitude Range) निश्चित केलेली नसते.	डिजिटल सिग्नलमध्ये 0 आणि 1 मर्यादित संख्या असतात
अॅनालॉग सिग्नल विरूपणासाठी अधिक प्रवण (Prone) आहे.	डिजिटल सिग्नलमध्ये विरूपण (Distortion) कमी होण्याची शक्यता असते.
अॅनालॉग सिग्नल लहरीच्या स्वरूपात डेटा प्रसारित करतो.	डिजिटल सिग्नल बायनरी (0 आणि 1) स्वरूपात डेटा प्रसारित करतो.
ध्वनी संदेश हे अॅनालॉग सिग्नलचे सर्वोत्तम उदाहरण आहे.	संगणकात वापरले जाणारे सिग्नल हे डिजिटल सिग्नल असतात.

## 1.2 संख्या प्रणाली (Number Systems):

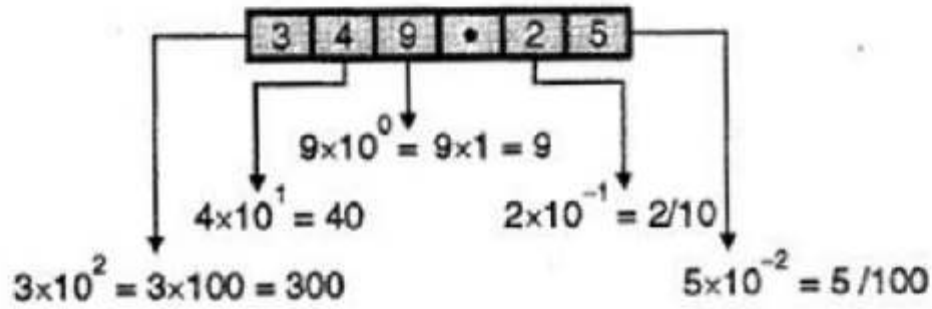
संख्या प्रणाली ही संख्या दर्शविण्याचा किंवा व्यक्त करण्याचा एक मार्ग आहे. अंक किंवा इतर चिन्हे सुसंगत रीतीने वापरून दिलेल्या संचाच्या संख्येचे प्रतिनिधित्व करण्यासाठी हे गणितीय नोटेशन आहे. डिजीटल सिस्टीम मध्ये फक्त काही अंक आणि चिन्हे असतात जे संख्येत असलेल्या स्थानावर अवलंबून भिन्न मूल्ये दर्शवतात म्हणून संख्या प्रणाली प्रत्येक संख्येचे एक विशेष प्रतिनिधित्व (Unique Representation) प्रदान करते. संख्येतील कोणत्याही अंकाचे मूल्य पुढील मुद्द्याद्वारे निर्धारित केले जाऊ शकते:

- अंक (Number),
- दशांश (decimal) बिंदूच्या आधी आणि नंतरच्या संख्येतील अंकाची स्थिती
- संख्या प्रणालीचा बेस (Base of number system)

### रेडिक्स किंवा बेस (Radix or Base):

स्थानात्मक अंक प्रणालीमध्ये बेस ही संख्या दर्शवण्यासाठी वापरल्या जाणाऱ्या शून्य अंकासह विशेष (Unique) अंकांची संख्या असते. उदाहरणार्थ, डेसिमल प्रणालीसाठी बेस (Base) 10 आहे, कारण ते दहा अंक म्हणजे 0 ते 9 पर्यंत चे वापरतात ज्यामध्ये प्रत्येक अंकात 10 भिन्न मूल्ये (Value) असू शकतात (0, 1, 2... ....9).

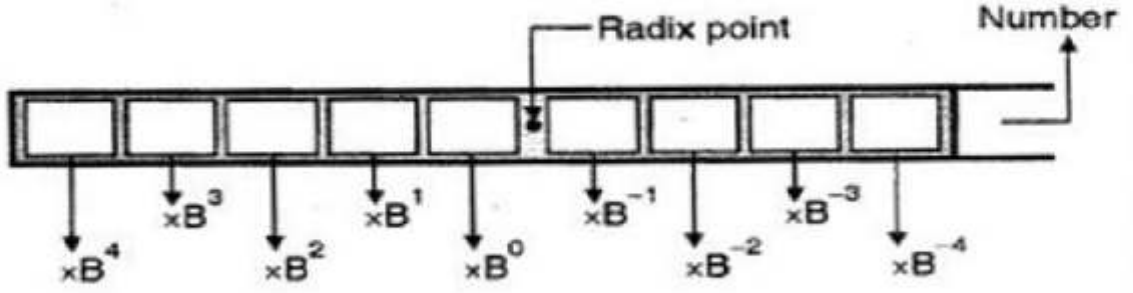
अंकाचे सर्वात मोठे मूल्य हे नेहमी बेसपेक्षा एक ने कमी असते. उदाहरणार्थ, डेसिमल प्रणालीतील (Decimal Number) सर्वात मोठा अंक 9 असतो. आकृती 1.3 मध्ये डेसिमल प्रणाली मध्ये प्रत्येक अंकाला बेस 10 च्या पॉवरने गुणाकार केलेला आहे.



आकृती 1.3 डेसिमल प्रणाली

म्हणूनच, सर्व संख्या प्रणाली साठी एक नियम लागू करतात. एका सामान्य संख्येसाठी आकृती 1.4 मध्ये दाखवल्याप्रमाणे प्रत्येक अंकाला बेस च्या काही पॉवरने गुणाकार करतात.





Where B = Base or Radix

आकृती 1.4 बेस B

### संख्या प्रणालीचे प्रकार (Types of Number System)

संख्या प्रणालीचे विविध प्रकार आहेत, चार सर्वात सामान्य संख्या प्रणाली प्रकार आहेत:

1. दशांश संख्या प्रणाली: बेस-10 (Decimal number system)
2. बायनरी नंबर सिस्टम: बेस-2 (Binary number system)
3. अष्टक संख्या प्रणाली: बेस-8 (Octal number System)
4. हेक्साडिसीमल संख्या प्रणाली: बेस-16 (Hexadecimal Number System)

#### 1.2.1 डेसिमल संख्या प्रणाली:

0 पासून 9 पर्यंत अंक असलेली संख्या प्रणाली आहे. डेसिमल ला दशांश संख्या प्रणाली म्हणून ओळखले जाते कारण एकूण दहा अंकांचा समावेश आहे.

**डेसिमल संख्या प्रणालीची वैशिष्ट्ये:**

- i) डेसिमल संख्या प्रणालीचा बेस 10 आहे.
- ii) डेसिमल बिंदूच्या डावीकडे असलेल्या संख्येच्या भागाला पूर्णांक भाग असे म्हणतात. त्याचप्रमाणे डेसिमल बिंदूच्या उजवीकडे असलेल्या संख्येच्या भागाला अपूर्णांक भाग असे म्हणतात.
- iii) या संख्या प्रणालीमध्ये अंकांची प्रत्येक जागा 10 च्या भिन्न पॉवरचे प्रतिनिधित्व करते. याला भारित मूल्ये (weighted values) असे म्हणतात.
- iv) डेसिमल बिंदूच्या डाव्या बाजूस क्रमिक स्थान असलेल्या अंकांचे भार  $10^1$ ,  $10^2$ ,  $10^3$  ..... इत्यादी असते. त्याचप्रमाणे डेसिमल बिंदूच्या उजव्या बाजूस क्रमिक स्थिती असलेल्या अंकांचे भार  $10^{-1}$ ,  $10^{-2}$ ,  $10^{-3}$  ..... इत्यादी असते. याचा अर्थ असा की, प्रत्येक स्थानाचे विशिष्ट भार असतो, जे बेस 10 ची पॉवर आहे.

तक्ता 1.2 डेसिमल संख्या प्रणाली दर्शवितो.

## तक्ता 1.2 डेसिमल संख्या प्रणाली

$10^3$	$10^2$	$10^1$	$10^0$		$10^{-1}$	$10^{-2}$	$10^{-3}$
=1000	=100	=10	=1	.	=0.1	=0.01	=0.001
Most Significant Digit				Decimal point			Least Significant Digit

मोस्ट सिग्निफिकंट डीजीट (MSD): सर्वात जास्त भार (weight) असलेल्या डाव्या अंकाला एखाद्या संख्येचा सर्वात महत्वाचा अंक असे म्हणतात.

लीस्ट सिग्निफिकंट डीजीट (LSD): सर्वात कमी भार (weight) असलेल्या सर्वात उजव्या अंकाला एखाद्या संख्येचा कमीत कमी लक्षणीय अंक असे म्हणतात.

### उदाहरण:-

1) 10 च्या पॉवरच्या दृष्टीने डेसिमल संख्या 1358.246 दर्शवा.

**उत्तर:** या संख्येचा पूर्णांकाचा भाग 1358 व या संख्येचा अपूर्णाकीय भाग 0.246 आहे.

8, 5, 3 आणि 1 या अंकांचे भार अनुक्रमे  $10^0$ ,  $10^1$ ,  $10^2$  आणि  $10^3$  असे आहे. त्याचप्रमाणे 2, 4 आणि 6 या अंकांचे भार अनुक्रमे  $10^{-1}$ ,  $10^{-2}$  आणि  $10^{-3}$  असे आहे.

$$1358.246 = (1 \times 10^3) + (3 \times 10^2) + (5 \times 10^1) + (8 \times 10^0) + (2 \times 10^{-1}) + (4 \times 10^{-2}) + (6 \times 10^{-3}).$$

### 1.2.2 बायनरी संख्या प्रणाली (Binary number system)

आधुनिक डिजिटल प्रणाली डेसिमल संख्येवर प्रक्रिया करत नाहीत तर ते बायनरी नंबर सिस्टमसह कार्य करतात. ज्याचा बेस 2 आहे आणि त्याचे घटक किंवा अंक फक्त 0 आणि 1 असतात. या प्रणालीचा बेस 2 असल्याने बायनरी नंबर सिस्टिम म्हणून ओळखली जाते. या प्रणालीतील 0 व 1 हे अंक बिट्स (Bits) म्हणून ओळखले जातात. डिजिटल इलेक्ट्रॉनिक्समध्ये या नंबर सिस्टिमचा मोठ्या प्रमाणात वापर होतो.

#### बायनरी संख्या प्रणालीची वैशिष्ट्ये:

- या संख्या प्रणालीचा बेस 2 आहे.
- संख्येचा जो भाग बायनरी पॉईंटच्या डावीकडे असतो, तो पूर्णांक भाग म्हणून ओळखतात. त्याचप्रमाणे संख्येचा जो भाग बायनरी पॉईंटच्या उजवीकडे असतो, त्याला अपूर्णांक भाग असे म्हणतात.
- या संख्याप्रणालीत बायनरी पॉईंटच्या डाव्या बाजूस क्रमिक स्थिती असलेल्या अंकांचे भार  $2^0$ ,  $2^1$ ,  $2^2$ ,  $2^3$  ..... इत्यादी असते. त्याचप्रमाणे बायनरी पॉईंटच्या उजवीकडे

क्रमिक स्थाने असलेल्या अंकाचे भार  $2^{-1}$ ,  $2^{-2}$ ,  $2^{-3}$ ..... इत्यादी असते, म्हणजे प्रत्येक स्थानाचे विशिष्ट भार असतो, जे बेस 2 ची पॉवर असते.

तक्ता 1.3 बायनरी संख्या प्रणाली दर्शवितो.

तक्ता 1.3: बायनरी संख्या प्रणाली

$2^3$	$2^2$	$2^1$	$2^0$		$2^{-1}$	$2^{-2}$	$2^{-3}$
=8	=4	=2	=1	.	=0.5	=0.25	=0.125
Most Significant Digit				Binary point			Least Significant Digit

बायनरी नंबरमधील डाव्या बाजूस सर्वात जास्त भार असलेल्या बायनरी बिटला मोस्ट सिग्निफिकंट बिट (MSB) असे म्हणतात, तर सर्वात कमी भार असलेल्या उजव्या बिटला लीस्ट सिग्निफिकंट बिट (LSB) म्हणतात. डेसिमल संख्या बायनरी नंबर सिस्टम मध्ये दर्शविण्यासाठी 1 आणि 0 च्या खूप लांब स्ट्रिंगची (Long string) आवश्यकता असते.

तक्ता 1.4 0 ते 15 डेसिमल संख्या बायनरी नंबरमध्ये दर्शवितो.

तक्ता 1.4: 0 ते 15 डेसिमल संख्या बायनरी नंबरमध्ये दाखवणारा तक्ता

डेसिमल (Base 10)	बायनरी (Base 2)
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

**उदाहरण:-**

1) 2 च्या पॉवरने बायनरी नंबर 1101.011 दर्शवा.

$$1101.011 = (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) + (0 \times 2^{-1}) + (1 \times 2^{-2}) + (1 \times 2^{-3})$$

### 1.2.3 अष्टक संख्या प्रणाली (Octal number System)

ऑक्टल संख्या प्रणालीमध्ये, बेस 8 असतो आणि तो संख्यांचे प्रतिनिधित्व करण्यासाठी 0 ते 7 पर्यंतच्या अंकांचा वापर करतो. मोठ्या बायनरी संख्या लहान करण्यासाठी ऑक्टल नंबर सिस्टम वापरतात. डिजिटल सर्किट्स आणि संगणक अंतर्गतपणे बायनरी सिस्टम प्रमाणे काटेकोरपणे कार्य करतात, ते ऑक्टल सिस्टम वापरत नाहीत. ऑक्टल सिस्टमचा वापर फक्त युजरची सोय म्हणून केला जातो.

**ऑक्टल संख्या प्रणालीची वैशिष्ट्ये**

- या संख्या प्रणालीचा बेस 8 आहे.
- ऑक्टल प्रणालीतील अंकाचे सर्वात मोठे मूल्य 7 आहे.
- अंकाचा जो भाग ऑक्टल बिंदूच्या (Octal point) डावीकडे असतो, तो पूर्णांक भाग म्हणून ओळखतात. त्याचप्रमाणे ऑक्टल बिंदूच्या उजवीकडेच्या भागाला अपूर्णांक भाग असे म्हणतात.
- या संख्याप्रणालीत ऑक्टल बिंदूच्या डाव्या बाजूस क्रमिक स्थान असलेल्या अंकांचे भार  $8^0$ ,  $8^1$ ,  $8^2$ ,  $8^3$ ... इत्यादी असतात. त्याचप्रमाणे ऑक्टल बिंदूच्या उजवीकडे क्रमिक स्थान असलेल्या अंकांचे भार  $8^{-1}$ ,  $8^{-2}$ ,  $8^{-3}$ .... इत्यादी असतात, म्हणजे प्रत्येक स्थानाचे विशिष्ट भार असतात, जे बेस 8 ची पॉवर असते.

तक्ता 1.4 ऑक्टल संख्या प्रणाली दर्शवितो.



तक्ता 1.4 ऑक्टल संख्या प्रणाली

$8^3$	$8^2$	$8^1$	$8^0$		$8^{-1}$	$8^{-2}$	$8^{-3}$
=512	=64	=16	=1	.	=1/8	=1/64	=1/512
Most Significant Digit				Octal point			Least Significant Digit

**उदाहरण**

3) ऑक्टल संख्या 1457.236 ला 8 च्या पॉवर च्या दृष्टीने दर्शवा.

$$1457.236 = (1 \times 8^3) + (4 \times 8^2) + (5 \times 8^1) + (7 \times 8^0) + (2 \times 8^{-1}) + (3 \times 8^{-2}) + (6 \times 8^{-3})$$

**1.2.4 हेक्साडेसीमल संख्या प्रणाली (Hexadecimal Number System)**

हेक्साडेसीमल प्रणालीमध्ये संख्या बेस 16 सह दर्शविल्या जातात. हेक्स प्रणालीमध्ये प्रथम संख्यांचे डेसिमल प्रणालीप्रमाणे म्हणजे 0 ते 9 पर्यंतचे अंक वापरतात आणि पुढील 10 ते 15 पर्यंतच्या संख्या A ते F पर्यंतच्या अक्षरांचा वापर करून दर्शविल्या जातात. हेक्साडेसीमल संख्येचा आकार बायनरी नंबरपेक्षा खूपच लहान आहे ज्यामुळे त्यांना लिहिणे आणि लक्षात ठेवणे सोपे असते. मायक्रोप्रोसेसर/मायक्रो कंट्रोलर मध्ये या संख्या मोठ्या प्रमाणावर वापरतात.

हेक्साडेसीमल संख्या

0	1	2	3	4	5	6	7
8	9	A	B	C	D	E	F

डेसिमल संख्या

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15

**हेक्साडेसीमल संख्या प्रणालीची वैशिष्ट्ये:**

- या संख्या प्रणालीचा बेस 16 आहे.
- हेक्साडेसीमल प्रणालीतील अंकाचे सर्वात मोठे मूल्य 15 आहे आणि ते F द्वारे दर्शवितात.
- हेक्साडेसीमल बिंदूच्या डावीकडे असलेल्या संख्येचा भागाला पूर्णांक भाग म्हणून ओळखतात त्याचप्रमाणे संख्येचा उजवीकडील भागाला अपूर्णांक भाग असे म्हणतात.
- या संख्याप्रणालीत हेक्साडेसीमल बिंदूच्या डाव्या बाजूस क्रमिक स्थान असलेल्या अंकांचे भार  $16^0$ ,  $16^1$ ,  $16^2$ ,  $16^3$  ... इत्यादी असते. त्याचप्रमाणे हेक्साडेसीमल बिंदूच्या उजवीकडे क्रमिक स्थान असलेल्या अंकांचे भार  $16^{-1}$ ,  $16^{-2}$ ,  $16^{-3}$  ... इत्यादी असते, म्हणजे प्रत्येक स्थानाचे विशिष्ट भार असतात, जे बेस 16 ची पॉवर असते.

हेक्साडेसीमल संख्या प्रणालीतील अंकाच्या विविध स्थानांसाठी भारित मूल्ये तक्ता 1.6 मध्ये दाखविली आहेत.

तक्ता 1.6 हेक्साडेसीमल संख्या प्रणाली

$16^3$	$16^2$	$16^1$	$16^0$		$16^{-1}$	$16^{-2}$	$16^{-3}$
=4096	=256	=16	=1	.	=1/16	=1/256	=1/4096
Most Significant Digit				Hexa Decimal point			Least Significant Digit

उदाहरण

4) 16 च्या पॉवर मध्ये हेक्साडेसीमल संख्या 6DE दर्शवा.

$$6DE = (6 \times 16^2) + (13 \times 16^1) + (14 \times 16^0)$$

तक्ता 1.7 बायनरी, डेसिमल, ऑक्टल आणि हेक्साडेसिमल यांच्यातील संबंध

डेसिमल (Base 10)	बायनरी (Base 2)	ऑक्टल (Base 8)	हेक्साडेसिमल (Base 16)
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B

12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

### 1.2.5 संख्या प्रणालींचे रूपांतरण

दिलेल्या संख्येचा विस्तार करून आणि सर्व संज्ञांची (Power) भर घालून एका संख्येचे एका संख्या प्रणालीतून दुसऱ्या संख्या प्रणालीत रूपांतर केले जाते. जर दिलेल्या संख्यामध्ये बेस बिंदूचा समावेश असेल, तर संख्या पूर्णांक भाग आणि अपूर्णांक भागामध्ये विभक्त करतात आणि नंतर प्रत्येक भागाचे स्वतंत्रपणे रूपांतरण केले जाते.

संख्या प्रणालीचे डेसिमल प्रणालीत रूपांतर करण्याची सर्वसाधारण प्रक्रिया खाली दिली आहे.

**स्टेप्स:-**

1. दिलेल्या संख्येची नोंद करा.
2. वेगवेगळ्या स्थानांची संबंधित भार (Weight) लिहा.
3. दिलेल्या संख्येतील प्रत्येक अंकाला गुणात्मक (Product) संख्या मिळविण्यासाठी संबंधित भारासह गुणाकार करा.
4. डेसिमल समतुल्य मिळविण्यासाठी सर्व गुणात्मक संख्या जोडा.

बायनरी, ऑक्टल आणि हेक्साडेसिमल संख्या प्रणालीचे त्याच्या डेसिमल समतुल्य (Equivalent) रूपांतरण खालील उदाहरणांवरून दिसून येते.

उदाहरण: बायनरी नंबर  $(1011.01)_2$  चे डेसिमल समतुल्य मध्ये रूपांतरीत करा.

$$\begin{aligned}
 (1011.01)_2 &= (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) + (0 \times 2^{-1}) + (1 \times 2^{-2}) \\
 &= 8 + 0 + 2 + 1 + 0 + \frac{1}{4} \\
 &= 11.25
 \end{aligned}$$

$$(1011.01)_2 = (11.25)_{10}$$

उदाहरण: ऑक्टल संख्या  $(314)_8$  ला त्याच्या डेसिमल समतुल्य मध्ये रूपांतरित करा.

$$\begin{aligned}
 (314)_8 &= (3 \times 8^2) + (1 \times 8^1) + (4 \times 8^0) \\
 &= 192 + 8 + 4 \\
 &= 204
 \end{aligned}$$

$$(314)_8 = (204)_{10}$$

उदाहरण: हेक्साडेसिमल संख्या  $(18)_{16}$  त्याच्या डेसिमल समतुल्य मध्ये रूपांतरित करा.

$$(18)_{16} = (1 \times 16^1) + (8 \times 16^0)$$

$$= 16 + 8$$

$$= 24$$

$$(18)_{16} = (24)_{10}$$

**डेसिमल पासून बायनरी, ऑक्टल हेक्साडेसिमल संख्या प्रणालीमध्ये रूपांतरण:**

डेसिमल संख्या प्रणालीचे इतर कोणत्याही प्रणालीत रूपांतर करण्याची सामान्य प्रक्रिया खाली दिली आहे:

स्टेप्स:

1. दिलेल्या संख्येची नोंद करा. दिलेल्या संख्येला बेस 'r' (बायनरी साठी 2, ऑक्टल साठी 8, हेक्साडेसिमल साठी 16) ने बाकी 0 येईपर्यंत भागा. समतुल्य(Equivalent) शोधण्यासाठी उर्वरित मूल्यांची खालून वरपर्यंत उलट्या क्रमाने यादी करा.
2. जर दिलेल्या डेसिमल संख्येमध्ये डेसिमल बिंदू असेल, तर आपल्याला प्रथम पूर्णांक आणि अपूर्णांक भाग वेगळे करावे लागतात. नंतर त्यांचे योग्य मूल्यांत रूपांतर करा व रूपांतरित भाग एकत्र करून पूर्ण रूपांतरित संख्या प्राप्त करा.
3. पूर्णांक भागासाठी: डेसिमल संख्येचा पूर्णांक भाग बेस 'r' द्वारे विभाजित करा आणि उर्वरित भाग खाली नोंदवा, बाकी 0 येईपर्यंत भागाचे विभाजन करणे सुरू ठेवा, प्रत्येक स्टेपवरील उर्वरित भागाची नोंद घ्या. समतुल्य शोधण्यासाठी उर्वरित मूल्यांची खालून वरपर्यंत उलट्या क्रमाने यादी करा.
4. अपूर्णांक भागासाठी:
  - i) डेसिमल संख्येच्या अपूर्णांक भागाला बेस 'r' ने गुणाकार करा. गुणाकारात निर्माण झालेली कॅरी एमएसडी (MSD) म्हणून लिहून ठेवा.
  - ii) आता गुणात्मक संख्येच्या अपूर्णांक संख्येला आधीच्या स्टेपासून बेस 'r' ने गुणाकार करा आणि कॅरीला एमएसडीची पुढची बिट म्हणून नोंदवा.
  - iii) वरील पायऱ्यांची शेवटपर्यंत पुनरावृत्ती करा, शेवटची कॅरी एमएसडीचे (LSD) प्रतिनिधित्व करते, आता रूपांतरित संख्या मिळविण्यासाठी वरपासून खालपर्यंत मूल्यांची यादी करा.



उदाहरण: डेसिमल  $(35)_{10}$  बायनरी मध्ये रूपांतरित करा.

2	35	
2	17	1
2	8	1
2	4	0
2	2	0
2	1	0
	0	1

Ans:  $(35)_{10} = (10011)_2$

उदाहरण: डेसिमल  $(965.125)_{10}$  ऑक्टल मध्ये रूपांतरित करा.

Integer part

8	965	
8	120	5
8	15	0
8	1	7
	0	1

Fractional part

Decimal	Product	Integer part
$0.125 \times 8$	1.00	1

Ans:  $(965.125)_{10} = (1705.1)_8$

उदाहरण: डेसिमल  $(2545)_{10}$  हेक्साडेसिमल मध्ये रूपांतरित करा.

Decimal = 2545

	Remainder
16   2545	1
16   159	15
16   9	9
0	

Hexadecimal = 9 F 1

### 1.3 बायनरी अंकगणित (Binary Algebra)

बायनरी अंकगणित हा सर्व संगणक आणि इतर अनेक डिजिटल प्रणालींचा आवश्यक भाग आहे. बेरीज, वजाबाकी, गुणाकार आणि भागाकार यांसारखी अंकगणितीय ऑपरेशन्स बायनरी अंकगणितात केली जातात. बायनरी नंबरवर ही ऑपरेशन्स केली जातात तेव्हा काही नियम लागू करावे लागतात.

#### 1.3.1 बायनरी बेरीज (Binary addition)

बायनरी बेरीज ही डेसिमल-प्रणालीतील बेरीजप्रमाणेच केली जाते. बायनरी बेरीज तक्ता 1.8 मध्ये दिलेल्या चार मूलभूत नियमांचे पालन करते.

तक्ता 1.8 बायनरी बेरीज

Case	A+B	Sum	Carry
1	0+0	0	0
2	0+1	1	0
3	1+0	1	0
4	1+1	0	1

दिलेल्या तक्ता मधील चौथ्या केसमध्ये बायनरी बेरीज (1+1 = 10) म्हणजे बेरीज 0 आणि कॅरी 1 अशी ऑडिशन लिहिली जाते.

उदाहरण - बायनरी ऑडिशन

उदाहरण:  $(5)_{10} + (4)_{10}$

$(1011)_2 + (1101)_2$

$$\begin{array}{r} (5)_{10} + (4)_{10} \\ 0 \ 1 \ 0 \ 1 \\ + \ 0 \ 1 \ 0 \ 0 \\ \hline 1 \ 0 \ 0 \ 1 \end{array}$$

#### 1.3.2 बायनरी वजाबाकी (Binary Subtraction):

वजाबाकी आणि बॉरो (Borrow) हे दोन शब्द बायनरी वजाबाकीसाठी वारंवार वापरले जातात. बायनरी सबट्रक्शन तक्ता 1.9 मध्ये दिलेल्या चार मूलभूत नियमांचे पालन करते.

तक्ता 1.9 बायनरी वजाबाकी

Case	A - B	Difference	Borrow
1	0 - 0	0	0
2	0 - 1	1	1
3	1 - 0	1	0
4	1 - 1	0	0

बायनरी सबट्रक्शन करताना काही वेळा पुढील अंकातून बॉरो घेणे आवश्यक असते. जेव्हा आपण 0 मधून 1 वजा करण्याचा प्रयत्न करतो तेव्हा फक्त बॉरो घेणे आवश्यक असते. दिलेल्या तक्तात केस 2 मध्ये पुढील अंकातून एक 1 बॉरो घेतल्यामुळे तो अंक 0 बनतो आणि नवीन अंक  $(10)_2$  तयार होतो, जो वजा केला जात असलेल्या ठिकाणी राहतो. खालील उदाहरणे बायनरी सबट्रक्शन स्पष्ट करतात.

उदाहरण:  $(14)_{10} - (9)_{10}$

उदाहरण:  $(1011011)_2 - (10010)_2$

$$(14)_{10} - (9)_{10}$$

$$\begin{array}{r} 1\ 1\ \overset{1}{\cancel{0}} \\ -1\ 0\ 0\ 1 \\ \hline 0\ 1\ 0\ 1 \end{array}$$

$$\begin{array}{r} 1\ 0\ 1\ 1\ 0\ 1\ 1 \\ -\quad\quad 1\ 0\ 0\ 1\ 0 \\ \hline 1\ 0\ 0\ 1\ 0\ 0\ 1 \end{array}$$

### 1.3.3 बायनरी गुणाकार (Binary Multiplication)

बायनरी मल्टिप्लीकेशन हा डेसिमल गुणाकारासारखाच असतो. बायनरी मल्टिप्लीकेशन तक्ता 1.10 मध्ये दिलेल्या चार मूलभूत नियमांचे पालन करते.

तक्ता 1.10 बायनरी मल्टिप्लीकेशन

Case	A x B	Multiplication
1	0 x 0	0
2	0 x 1	0
3	1 x 0	0
4	1 x 1	1

उदाहरण- बायनरी मल्टिप्लीकेशन

उदाहरण:  $(101)_2 \times (11)_2$

$$\begin{array}{r}
 101 \\
 \times 11 \\
 \hline
 101 \\
 101 \\
 \hline
 1111
 \end{array}$$

उदाहरण:  $(26)_{10} \times (12)_{10}$

$$\begin{array}{r}
 (26)_{10} \quad 11010 \\
 (12)_{10} \quad \times 1100 \\
 \hline
 00000 \\
 00000 \\
 11010 \\
 11010 \\
 \hline
 100111000 \quad (312)_{10}
 \end{array}$$

### 1.3.4 बायनरी भागाकार (Binary Division):-

बायनरी डिविजन ही डेसिमल भागाकारासारखेच असते. याला दीर्घ विभाजन प्रक्रिया असे म्हणतात. तो बिट्स चे EX-OR करतो. खालील उदाहरण बायनरी डिविजन स्पष्ट करते.

उदाहरण:  $(11100)_2$  भागिले  $(100)_2$

$$\begin{array}{r}
 111 \\
 100 \overline{) 11100} \\
 \underline{- 100} \phantom{0} \\
 0110 \\
 \underline{- 100} \phantom{0} \\
 0100 \\
 \underline{- 100} \phantom{0} \\
 000
 \end{array}$$

### 1.4. 1-चे पूरक (1's complement) व 2-चे पूरक (2's complement) वापरून वजाबाकी

#### 1.4.1 1-चे पूरक (1's complement):

1 चे कॉम्प्लिमेंट मूळ संख्येची ऋण (Negative) संख्या दर्शवितो.

बायनरी संख्येचे 1 चे कॉम्प्लिमेंट दिलेल्या बायनरी संख्येमधील 0 ला 1 ने आणि 1 ला 0 ने बदलून मिळवतात.

## 1 च्या कॉम्प्लिमेंट पद्धतीचा वापर:

- i) संगणक
- ii) ॲडर्स
- iii) सबट्रक्टर
- iv) डिजिटल प्रणाली

उदाहरण: दिलेल्या बायनरी संख्यांचे 1-चे कॉम्प्लिमेंट काढा.

a) 101010

b) 110101

a)101010

b)110101

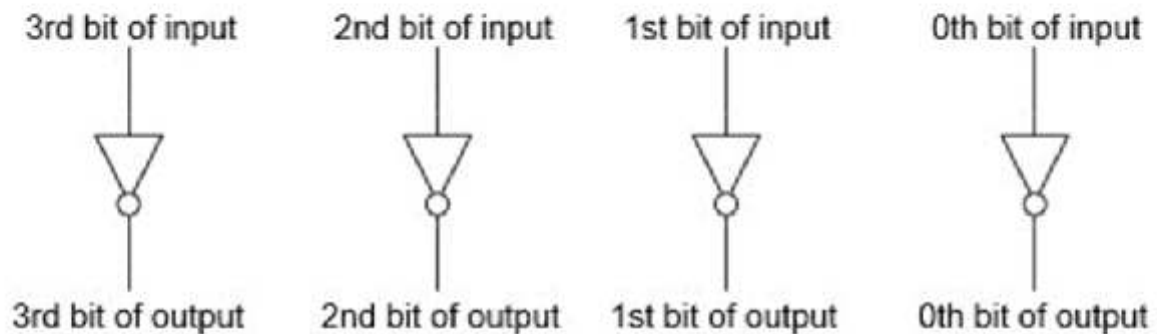
↓↓↓↓↓

↓↓↓↓↓

010101

001010

बायनरी संख्या 1-चे कॉम्प्लिमेंट मध्ये रूपांतरित करण्यासाठी एक साधा अल्गोरिदम आहे. बायनरी संख्येचे 1-चे कॉम्प्लिमेंट मिळविण्यासाठी, दिलेली संख्या उलट (Invert) करा. बायनरी नंबर इनपुटच्या प्रत्येक बिटसाठी फक्त नॉट गेट वापरून 1's कॉम्प्लिमेंट तयार करतात. आकृती 1.5 मध्ये 4-Bit 1's compliment लॉजिकची (Logic) अंमलबजावणी दिली आहे.



आकृती 1.5 4-बिट 1-चे कॉम्प्लिमेंट लॉजिक

$(A)_2 - (B)_2$  वजाबाकीसाठी खालील स्टेप्स आहेत जिथे A हा मिन्युएंड आहे आणि B सबट्राहेंड आहे.

स्टेप 1: सबट्राहेंड B चे 1 चे कॉम्प्लिमेंट काढा.

स्टेप 2: A मध्ये B चे 1 चे कॉम्प्लिमेंट जोडा.

स्टेप 3: जर अंतिम कॅरी 1 असेल, तर स्टेप 2 मध्ये मिळालेल्या बेरजेच्या निकालात जोडा.

जर कॅरी 1 असेल, तर वजाबाकी पोझिटिव्ह असेल, म्हणून ते मूळ(True) स्वरूपात आहे.

स्टेप 4: जर स्टेप 2 मध्ये उत्पादित कॅरी 0 असेल, तर मिळालेला परिणाम निगेटिव्ह आणि 1 च्या कॉम्प्लिमेंट स्वरूपात असेल, म्हणून ते मूळ स्वरूपात रूपांतरित करा.

मोठ्या संख्येतून लहान संख्या वजा करायची उदाहरणे:

i) 1-चे कॉम्पलिमेंट पद्धतीचा वापर करून  $(9)_{10} - (4)_{10}$  करा.

उत्तर;  $(9)_{10} = (1001)_2$

$(4)_{10} = (0100)_2$

स्टेप 1:  $(0100)_2$  चे 1 चे कॉम्पलिमेंट  $= (1011)_2$

स्टेप 2:  $(9)_{10}$  आणि  $(4)_{10}$  चे कॉम्पलिमेंट जोडा.

$$\begin{array}{r} \text{मिनुएन्ड-} \\ \text{सबट्राहेंडचे 1 ची कॉम्पलिमेंट} \\ \text{कॅरी} \end{array} \quad \begin{array}{r} 1001 \\ + 1011 \\ \hline 10100 \end{array}$$

स्टेप 3: आलेल्या उत्तरात कॅरी जोडा.

0101

उत्तर पोजिटीव्ह आणि मूळ स्वरूपात आहे.

उत्तर  $(0101)_2$  आहे.

ii) 1-चे कॉम्पलिमेंट पद्धतीचा वापर करून  $(1011.001)_2 - (110.10)_2$  करा.

उत्तर:

स्टेप 1:  $(0110.100)_2$  1-चे कॉम्पलिमेंट  $(1001.011)_2$

$$\begin{array}{r} \text{मिनुएन्ड-} \\ \text{सबट्राहेंडचे 1-चे कॉम्पलिमेंट-} \\ \text{कॅरी} \end{array} \quad \begin{array}{r} 1011.001 \\ \underline{1001.011} \\ 10100.100 \end{array}$$

स्टेप 3: अंतिम कॅरी आलेल्या उत्तरात जोडा

0100.101

उत्तर पोजिटीव्ह आणि मूळ स्वरूपात आहे

उत्तर  $(0100.101)_2$  आहे.

#### 1.4.2 2-चे कॉम्पलिमेंट (2's complement)

2-चे कॉम्पलिमेंट (2's complement) हे बायनरी संख्यांवरील गणितीय क्रिया आहे. हे संगणकीय चिन्हांकित (Signed) संख्येच्या प्रतिनिधित्वाची पद्धत म्हणून वापरले जाते. चिन्हांकित संख्यांचे प्रतिनिधित्व करण्यासाठी इतर प्रणालींच्या तुलनेत 2-चे कॉम्पलिमेंटचा हा फायदा असा आहे की, बेरीज, वजाबाकी आणि गुणाकाराची मूलभूत अंकगणितीय क्रिया चिन्हांकित केलेल्या बायनरी संख्यांसाठी समान असतात. या गुणधर्मांमुळे प्रणाली विशेषतः उच्च-अचूकता (Highly accurate) अंकगणितासाठी लागू करणे सोपे होते.

**बायनरी संख्येचे 2-चे कॉम्पलिमेंट:**

बायनरी संख्येची 2-चे कॉम्पलिमेंट संख्या 1-चे कॉम्पलिमेंट संखेच्या लीस्ट सीग्निफिकंट बिट (LSB) मध्ये 1 जोडून प्राप्त केली जाते.

2-चे कॉम्पलिमेंट = 1-चे कॉम्पलिमेंट + 1

2-चे कॉम्पलिमेंट चे उदाहरण खालीलप्रमाणे आहे.

**उदाहरण 1:**  $(110)_2$  2-चे कॉम्पलिमेंट काढा

उत्तर:  $110 \rightarrow 001$  1-चे कॉम्पलिमेंट

2-चे कॉम्पलिमेंट = 001

$$\begin{array}{r} + 1 \\ 010 \end{array}$$

**उदाहरण 2:** प्रत्येक 3-बिट बायनरी संख्येची 2-चे कॉम्पलिमेंट काढा.

तक्ता 1.11 मध्ये दाखविल्याप्रमाणे दिलेल्या बायनरी संख्येमधील 0 ला 1 ने आणि 1 ला 0 ने बदलून 1-चे कॉम्पलिमेंट नंतर बदललेल्या संख्येच्या एलएसबी मध्ये 1 मिळवून 2-चे कॉम्पलिमेंट काढले जाते

तक्ता 1.11 3 बिट बायनरी साठी 1-चे आणि 2-चे कॉम्पलिमेंट संख्या

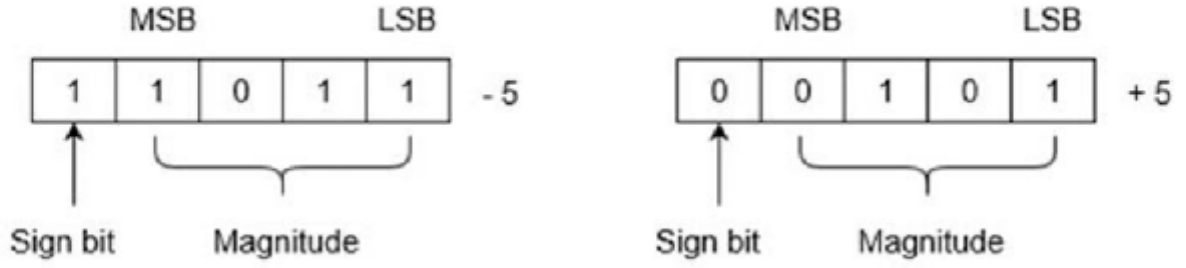
Binary number	1's complement	2's complement
000	111	000
001	110	111
010	101	110
011	100	101
100	011	100
101	010	011
110	001	010
111	000	001

**2-चे कॉम्पलिमेंट बायनरी संख्यांचा वापर**

बायनरी संख्यांच्या 2-चे कॉम्पलिमेंटचे विविध उपयोग आहेत. मुख्यतः चिन्हांकित केलेल्या बायनरी संख्येच्या प्रतिनिधित्वामध्ये आणि बायनरी संख्यांसाठी विविध अंकगणितीय ऑपरेशन्स, उदा., बेरीज, वजाबाकी मध्ये 2-चे कॉम्पलिमेंट विशेषतः उपयुक्त आहे.

उदाहरण - 5-Bit register वापरलेला आहेत तर -5 आणि +5 चे प्रतिनिधित्व आकृती 1.5 प्रमाणे असेल.

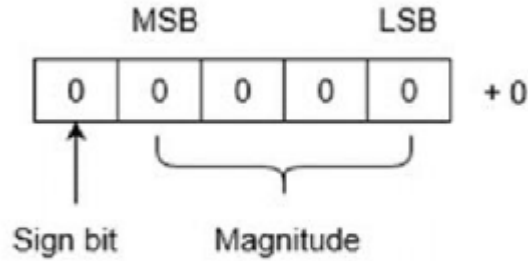




आकृती 1.5 -5, +5 चे प्रतिनिधित्व

- (i) +5 = 0 0101  
(ii) -5 = 1 0101

या प्रणालीचा फायदा असा आहे की 2-चे कॉम्प्लिमेंट प्रतिनिधित्वामध्ये शून्य (0) नेहमी पॉझिटिव्ह आहे असे मानले जाते म्हणून ते आकृती 1.6 मध्ये दाखविल्याप्रमाणे पॉझिटिव्ह संख्येचे स्पष्ट प्रतिनिधित्व करते.



आकृती 1.6 धन संख्येचे चे प्रतिनिधित्व

वजाबाकीच्या 2-चे कॉम्प्लिमेंट पद्धतीच्या मदतीने आपण दोन बायनरी संख्या सहजपणे वजा करू शकतो.

ऑपरेशन खालील स्टेप्सद्वारे केले जाते:

स्टेप 1: सुरुवातीला, सबट्राहेंडची (subtrahend) 2-चे कॉम्प्लिमेंट शोधतात.

स्टेप 2: नंतर ते मिनुएन्ड(minuend) मध्ये जोडले जाते.

स्टेप 3: बेरीजचे अंतिम कॅरी 1 असल्यास, ते काढून टाकले जाते आणि उत्तर पॉझिटिव्ह असते.

स्टेप 4: बेरीजचे अंतिम कॅरी 0 असल्यास उत्तर निगेटिव्ह असते.

मोठ्या संख्येतून लहान संख्या वजा करायची उदाहरणे:

2-चे कॉम्प्लिमेंट पद्धतीचा वापर करून  $(54)_{10} - (22)_{10}$  शोधा

$$(54)_{10} = (110110)_2$$

$$(22)_{10} = (10110)_2$$



**उत्तर:**

सबट्राहेंड मधील बिट्सची संख्या 5 आहे तर मिनुएन्ड ची संख्या 6 आहे. सबट्राहेंडच्या सहाव्या ठिकाणी '0' घेऊन सबट्राहेंड मधील बिट्सची संख्या मिनुएन्ड च्या समान करतात. आता, 010110 ची 2-चे कॉम्पलिमेंट (101101+1) म्हणजे 101010 आहे. हे मिनुएन्ड सह ऑडिशन करतात.

$$\begin{array}{r} \text{मिनुएन्ड} \quad \quad \quad 1\ 1\ 0\ 1\ 1\ 0 \\ \text{सबट्राहेंडचे २ ची कॉम्पलिमेंट} \quad 1\ 0\ 1\ 0\ 1\ 0 \\ \hline \quad \quad \quad 1\ 1\ 0\ 0\ 0\ 0\ 0 \end{array}$$

कॅरी काढून टाकल्यानंतर, वजाबाकी  $(100000)_2$  मिळेल.

ii)  $(1010.11)_2 - (1001.01)_2$

**उत्तर:**

1001.01 ची 2-चे कॉम्पलिमेंट 0110.11

$$\begin{array}{r} \text{मिनुएन्ड} \quad \quad \quad 1\ 0\ 1\ 0 . 1\ 1 \\ \text{सबट्राहेंडचे २ ची कॉम्पलिमेंट} \quad + \quad \underline{0\ 1\ 1\ 0 . 1\ 1} \\ \text{Carry over} \quad \quad \quad 1\ 0\ 0\ 0\ 1 . 1\ 0 \end{array}$$

कॅरी काढून टाकल्यानंतर, वजाबाकी  $(1.10)_2$  मिळेल.

**1.5 बीसीडी, ग्रे, एक्सएस-3, आस्की (ASCII) कोड (BCD, Gray, Excess-3, ASCII code):**

### 1.5.1 बायनरी कोडेड डेसिमल किंवा बीसीडी (BCD)

ही डेसिमल संख्यांना त्यांच्या बायनरी समतुल्यमध्ये रूपांतरित करण्याची एक प्रक्रिया आहे.

हा बायनरी एन्कोडिंगचा एक प्रकार आहे जिथे डेसिमल संख्येतील प्रत्येक अंक बिट्सच्या स्वरूपात दर्शविला जातो. हे एन्कोडिंग एकतर 4-बिट किंवा 8-बिट मध्ये केले जाऊ शकते. ही एक जलद आणि कार्यक्षम प्रणाली आहे जी बायनरी प्रणालीच्या तुलनेत डेसिमल संख्यांना बायनरी संख्यांमध्ये रूपांतरित करते. हे सामान्यतः डिजिटल डिस्प्लेमध्ये वापरले जातात.

डेसिमल संख्येचे बीसीडी समतुल्य पूर्णांक आणि अपूर्णांक भागांमधील प्रत्येक डेसिमल अंकाला त्याच्या चार बिट बायनरी समतुल्यासह बदलून लिहिले जाते. बीसीडी कोड 8421 बीसीडी कोड म्हणून ओळखला जातो, ज्यामध्ये 8,4,2 आणि 1 चार-बिट गटांमधील वेगवेगळ्या बिट्सच्या भारांचे प्रतिनिधित्व करतात, एमएसबीपासून प्रारंभ करून आणि एलएसबीच्या

दिशेने पुढे जातात. दिलेल्या डेसिमल अंकाचे प्रतिनिधित्व करणाऱ्या चार बिट गटातील प्रत्येक बिटचा एक नियुक्त भार आहे.

डेसिमल अंकाचे बीसीडी मध्ये मर्यादित स्थान-मूल्य असते. उदाहरणार्थ, बायनरीमध्ये 0.2 हे .001100... आहे आणि बीसीडीमध्ये 0.0010 आहे. हे अपूर्णकीय चुका टाळते. तक्ता 1.12 मध्ये डेसिमल अंकाचे BCD करिता सत्य सारणी (Truth Table) दाखवली आहे.

तक्ता 1.12 BCD करिता सत्य सारणी (Truth Table)

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

बीसीडी प्रणालीमध्ये, दिलेली डेसिमल संख्येच्या आतील प्रत्येक डेसिमल अंकासाठी चार बिट्स मध्ये विभागली जाते. प्रत्येक डेसिमल अंकाचे त्याच्या थेट बायनरी स्वरूपात रूपांतर होते.

उदाहरण: डेसिमल संख्या  $(123)_{10}$  त्याच्या बीसीडी समतुल्य मध्ये रूपांतरित करा.

दुथ टेबल वरून

1 -> 0001

2 -> 0010

3 -> 0011

म्हणून BCD नंबर-> 0001 0010 0011

अशा प्रकारे डेसिमल संख्यांचे त्यांच्या समतुल्य बीसीडी मध्ये रूपांतर केले जाते.

बीसीडी म्हणजे डेसिमल संख्येच्या प्रत्येक अंकाचे बायनरी प्रतिनिधित्व आहे.

### बीसीडी कोडचे फायदे:

- i) हे डेसिमल प्रणालीसारखेच आहे.
- ii) फक्त 0 ते 9 या डेसिमल संख्यांच्या समतुल्य बायनरी नंबर लक्षात ठेवण्याची गरज आहे.

### बीसीडी कोडचे तोटे

- i) बीसीडीची बेरीज आणि वजाबाकीचे वेगवेगळे नियम आहेत.
- ii) बीसीडीचे अंकगणित अधिक गुंतागुंतीचे असते.
- iii) डेसिमल संख्या दर्शविण्यासाठी बीसीडीला बायनरीपेक्षा जास्त बिट्सची आवश्यकता असते. त्यामुळे बीसीडी बायनरीपेक्षा कमी कार्यक्षम आहे.
- iv) दिलेल्या डेसिमल संख्येचे बीसीडी प्रतिनिधित्व अतिरिक्त बिट्स वापरते, ज्यामुळे ते जड-भारित (weighted) होते.

### 1.5.2 ग्रे कोड (Gray Code)

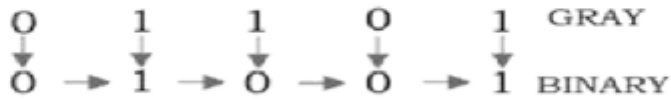
याला चक्रीय (Cyclic) कोड 'किंवा' परावर्तित बायनरी कोड (Reflective Binary code) म्हणून देखील ओळखले जाते. ग्रे कोडमध्ये एका स्टेपवरून दुसऱ्या स्टेपवर जाताना, कोड गुपमधील फक्त एक बिट बदलते. म्हणजे दोन लगतचे कोड नंबर एकमेकांपासून फक्त एका बिटने भिन्न असतात. ग्रे कोड हा एकक अंतर कोड मध्ये सर्वात लोकप्रिय आहे, परंतु तो अंकगणितीय क्रियांसाठी योग्य नाही. रोटरी आणि ऑप्टिकल एनकोडरमध्ये ग्रे कोड वापरले जातात.

### ग्रे ते बायनरी कोड रूपांतरण (Gray to Binary code conversion)

स्टेप्स:

1. ग्रे कोडला बायनरीमध्ये रूपांतरित करण्यासाठी, दिलेल्या ग्रे कोड नंबरचा एमएसडी (MSD) खाली आणा, कारण, पहिला ग्रे कोड नंबरचा अंक किंवा एमएसडी बायनरी संख्येसारखाच आहे.
2. क्रमिक द्वितीय बायनरी बिट (Sequential second binary bit) प्राप्त करण्यासाठी, पहिल्या बिट किंवा एमएसडी दरम्यान EX-OR ऑपरेशन करा
3. क्रमिक तिसरे बायनरी बिट (Sequential third binary bit) प्राप्त करण्यासाठी, दुसऱ्या बिट किंवा एमएसडीच्या दरम्यान EX-OR ऑपरेशन करा. अश्याप्रकारे पुढील बायनरी बिट प्राप्त करा.

उदाहरण: (01101)<sub>Gray</sub> चा बायनरी कोड

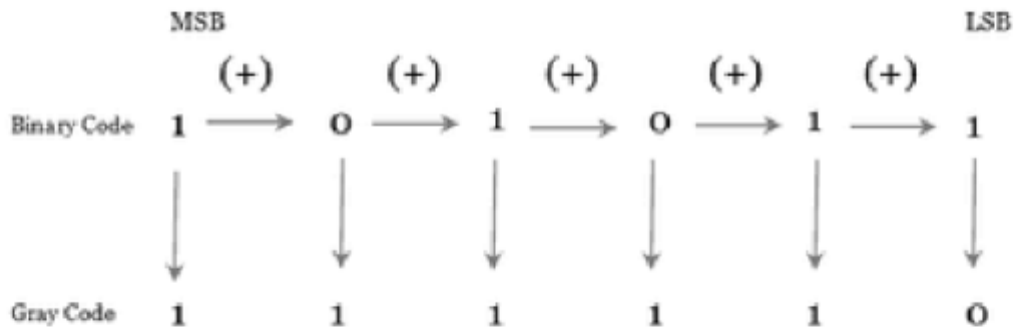


### बायनरी ते ग्रे कोड रूपांतरण (Binary to Gray code conversion)

स्टेप्स:

1. बायनरीचे ग्रे कोडमध्ये रूपांतर करण्यासाठी दिलेल्या बायनरी नंबरचा एमएसडी खाली आणा, कारण, ग्रे कोड नंबरचा पहिला अंक किंवा एमएसडी बायनरी नंबरसारखाच असतो.
2. दिलेल्या बायनरीसाठी समतुल्य ग्रे कोडेड क्रमांक तयार करण्यासाठी क्रमिक ग्रे कोडेड बिट्स मिळविण्यासाठी, पहिल्या बिट किंवा बायनरीच्या एमएसडीच्या ते दुसऱ्या अंकाच्या दरम्यान EX-OR ऑपरेशन करा आणि ग्रे कोडच्या पहिल्या बिटच्या पुढे रिजल्ट लिहा.
3. जोपर्यंत सर्व बायनरी बिट्स त्यांच्या आधीच्या बिट्ससह EX-OR केले जात नाहीत तोपर्यंत वरील स्टेप ची पुनरावृत्ती करा.

उदाहरण: (101011)<sub>2</sub> चा ग्रे कोड



0 ते 15 पर्यंतच्या डेसिमल संख्यांसाठी तक्ता 1.13 4 बिट बायनरी कोड आणि 4 बिट ग्रे कोड दर्शवितो.

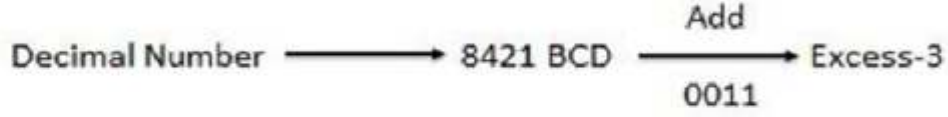
तक्ता 1.13 डेसिमल बायनरी ग्रे कोड रूपांतरण

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

### 1.5.3 एक्सएस-3 कोड (Excess-3 code)

हा डेसिमल संख्या व्यक्त करण्यासाठी वापरला जाणारा नॉन-वेटेड कोड आहे.

एक्सएस-3 मध्ये सांकेतिक शब्द 8421 बीसीडी कोड शब्दांपासून 8421 मधील प्रत्येक कोड शब्दांमध्ये  $(0011)_2$  किंवा  $(3)_{10}$  जोडून तयार केले गेले आहेत.



तक्ता 1.14 डेसिमल बीसीडी, एक्सएस-3 कोड मधील संबंध दर्शवितो.

तक्ता 1.14 डेसिमल बीसीडी, एक्सएस-3 कोड रूपांतरण

डेसिमल	बीसीडी	एक्सएस-3
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

उदा., 15 चे रूपांतर एक्सएस-3 कोडमध्ये करण्यासाठी, खाली दाखवल्याप्रमाणे प्रत्येक अंकात प्रथम 3 समाविष्ट केले जाईल.

$$\begin{array}{r}
 15 \\
 + 33 \\
 \hline
 48
 \end{array}$$

आता याचे BCD रूपांतरण

$$(48)_{10} = (0100 \ 1000)$$

**बीसीडी ते एक्सएस-3 रूपांतरण (BCD to Excess-3 code conversion)**

स्टेप्स:

- स्टेप 1 - बी.सी.डी.ला डेसिमल मध्ये रूपांतरित करा.
- स्टेप 2 - या डेसिमल संख्येमध्ये  $(3)_{10}$  जोडा.



स्टेप 3 - एक्सएस-3 कोड मिळविण्यासाठी बायनरीमध्ये रूपांतरित करा.

उदाहरण -  $(0110)_{BCD}$  ला एक्सएस-3 मध्ये रूपांतरित करा.

स्टेप 1 - डेसिमल मध्ये परिवर्तित करा

$$(0110)_{BCD} = 6$$

स्टेप 2 - डेसिमलमध्ये 3 जोडा

$$6+3 = 9$$

स्टेप 3 - एक्सएस-3मध्ये रूपांतरित करा

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

परिणाम:

$$(0110)_{BCD} = (1001)_{XS-3}$$

#### 1.5.4 आस्की (ASCII) कोड:

ASCII कोड (American Standard Code for Information Interchange) हा एक 7-बिट कोड आहे आणि तो  $2^7 = 128$  संभाव्य कॅरेक्टर्सचे प्रतिनिधित्व करतो.

हा कोड सर्व स्टॅंडर्ड कीबोर्ड कॅरेक्टर्स (standard keyboard) तसेच रिटर्न इ. सारख्या नियंत्रण कार्यांचे प्रतिनिधित्व करण्यासाठी पुरेसा आहे. तक्ता 1.15 मध्ये ASCII कोडची यादी दर्शविली आहे. प्रत्येक कॅरेक्टरसाठी, कॅरेक्टर च्या स्तंभात पहिले तीन बिट्स ( $b_7, b_6, b_5$ ) आणि ओळीत उरलेले चार बिट्स ( $b_4, b_3, b_2, b_1$ ) असे 7 बायनरी बिट्स दर्शवितो. ASCII चा समतुल्य डेसिमल नंबर हा या बायनरी नंबर वरून शोधला जातो.

उदाहरणार्थ, K साठी ASCII कोड  $(1001011)_2$  आहे.

तक्ता 1.15 आस्की(ASCII) कोड

USASCII code chart

<div> <div>7 6 5 4 3 2 1</div> <div>Bits</div> </div>					<div> <div>0 0 0 0 0 0 0 0</div> <div>0 0 0 0 1 0 1 1</div> </div>							
<div> <div>Column</div> <div>Row</div> </div>					0	1	2	3	4	5	6	7
0	0	0	0	0	0	NUL	DLE	SP	@	P	\	p
0	0	0	1	1	1	SOH	DC1	!	A	Q	a	q
0	0	1	0	2	2	STX	DC2	"	B	R	b	r
0	0	1	1	3	3	ETX	DC3	#	C	S	c	s
0	1	0	0	4	4	EOT	DC4	\$	D	T	d	t
0	1	0	1	5	5	ENQ	NAK	%	E	U	e	u
0	1	1	0	6	6	ACK	SYN	&	F	V	f	v
0	1	1	1	7	7	BEL	ETB	'	G	W	g	w
1	0	0	0	8	8	BS	CAN	(	H	X	h	x
1	0	0	1	9	9	HT	EM	)	I	Y	i	y
1	0	1	0	10	10	LF	SUB	*	J	Z	j	z
1	0	1	1	11	11	VT	ESC	+	K	[	k	{
1	1	0	0	12	12	FF	FS	,	L	\	l	
1	1	0	1	13	13	CR	GS	-	M	]	m	}
1	1	1	0	14	14	SO	RS	.	N	^	n	~
1	1	1	1	15	15	SI	US	/	O	_	o	DEL

## 1.6 बीसीडी अंकगणित (BCD Arithmetic)

### 1.6.1 बीसीडी बेरीज (BCD Addition)

1. नियमित बायनरी संख्या म्हणून बीसीडी अंक जोडा.
2. जर बेरीज 9 किंवा त्यापेक्षा कमी असेल आणि कोणतीही कॅरी उत्पन्न झाली नसेल, तर तो वैध (Valid) बीसीडी अंक आहे.
3. जर बेरीज कॅरी तयार करते, तर बेरीज अवैध आहे आणि अंक  $6_{10}$  किंवा  $(0110)_2$  अंकात जोडणे आवश्यक आहे.
4. बेरीज नऊ पेक्षा जास्त असल्यास, बेरीज अवैध आहे आणि अंक  $6_{10}$  किंवा  $(0110)_2$  अंकात जोडणे आवश्यक आहे.
5. प्रत्येक बीसीडी अंकासाठी पुनरावृत्ती करा.

**प्रकरण 1: बेरीज 9 च्या बरोबर किंवा त्यापेक्षा कमी आहे आणि कॅरी = 0 आहे**

Example:  $(3)_{10} + (4)_{10}$  in BCD

Decimal		BCD
$(3)_{10}$	→	0011
$(4)_{10}$	→ +	0100
$(7)_{10}$		<u>0111</u>

Sum is a valid BCD Number with carry 0

**प्रकरण 2: बेरीज 9 पेक्षा जास्त आहे परंतु कॅरी = 0 आहे**

Example:  $(6)_{10} + (8)_{10}$  in BCD

Decimal		BCD
$(6)_{10}$	→	0110
$(8)_{10}$	→ +	1000
$(14)_{10}$		<u>1110</u>

Sum is a invalid BCD Number with carry 0

Hence requires a correction of 6

		1110	
	+	0110	
		111	
final carry 1		<u>0100</u>	
	0001	0100	
final answer	1	4	Valid BCD

वरील केसमध्ये ऑडिशनमध्ये 6 हा क्रमांक जोडला जातात. हे बीसीडी च्या सहा अवैध (Invalid) अवस्था वगळण्यासाठी केले जाते ज्यात उत्तर 10 ते 15 पर्यंत आहे.



### प्रकरण 3: बेरीज 9 पेक्षा कमी किंवा समान आहे परंतु (कॅरी) = 1 आहे

Example:  $(9)_{10} + (8)_{10}$  in BCD

Decimal		BCD
$(9)_{10}$	→	1001
$(8)_{10}$	→	+ 1000
$(17)_{10}$		1 0001

Sum is a valid BCD Number with carry 1. Hence invalid BCD

Hence requires a correction of 6

	0001	0001
	+ 0000	0110
	0001	0111
final answer	1	7

Valid BCD

#### स्व-अध्ययन (Self Learning) प्रश्न:

- 1) डेसिमल संख्या प्रणाली स्पष्ट करा.
- 2) बायनरी संख्या सिस्टम स्पष्ट करा.
- 3) ऑक्टल संख्या प्रणाली स्पष्ट करा.
- 4) हेक्साडेसिमल संख्या प्रणाली स्पष्ट करा.
- 5) ऑक्टल संख्या  $(317)_8$  ला त्याच्या डेसिमल समतुल्य मध्ये रूपांतरित करा.
- 6) बायनरी नंबर 1011.011 चे डेसिमल समतुल्य मध्ये रूपांतरित करा.
- 7) हेक्साडेसिमल संख्या  $(20)_{16}$  त्याच्या डेसिमल समतुल्य मध्ये रूपांतरित करा.
- 8) सोडवा  $(1011011)_2 - (10010)_2$
- 9) 1 च्या कॉम्प्लिमेंट पद्धतीचा वापर करून  $(8)_{10} - (4)_{10}$  करा.
- 10) 2-चे कॉम्प्लिमेंट पद्धतीचा वापर करून  $(52)_{10} - (12)_{10}$  शोधा.
- 11) बीसीडी बेरजेचे नियम लिहा.
- 12) दिलेले बायनरी अंक ग्रे कोड मध्ये लिहा.  
a. 110100      b. 1111
- 13) बीसीडी अंकगणित करा.    1)  $17+58$     2)  $9 - 2$ .

#### लघु प्रकल्प (Micro project):

1. बीसीडी संख्याचे हेक्साडेसिमल संख्या (BCD to Hex Number) मध्ये रूपांतरण करणाऱ्या IC चा अभ्यास करा.

## Unit 2

### लॉजिक गेट्स (तार्किक द्वार) अँड लॉजिक फॅमिलीज

#### Logic Gates and Logic Families

#### विषय निष्पत्ती (Course Outcome):

बुलियन एक्सप्रेशनचा वापर करून लॉजिक सर्किट्स विकसित करणे.

#### घटक निष्पत्ती (Unit Outcome):

1. दिलेल्या NAND/NOR गेटचा सार्वत्रिक गेट म्हणून वापर करून मूलभूत गेट्स विकसित करणे.
2. बुलियन नियम वापरून दिलेली एक्सप्रेशन सिम्प्लिफाय करणे.
3. दिलेल्या बुलियन एक्सप्रेशनचा वापर करून लॉजिक सर्किट्स विकसित करणे.
4. दिलेल्या डिजिटल लॉजिक फॅमिलीच्या ठळक वैशिष्ट्यांची तुलना करणे.

#### 2.1 प्रस्तावना

जॉर्ज बुलियन याने गणितशास्त्र व लॉजिक यांच्यात संबंध आहे असे मत ई. स. १८५४ मध्ये मांडले. यासाठी त्याने काही सांकेतिक तर्क मांडले. त्याला बुलियनचे गणितशास्त्र (Boolean Algebra) असे म्हणतात. या तर्कानुसार प्रत्येक राशीला दोन किमती असतात. सत्य (true) किंवा असत्य (false) या द्विस्थितीय तत्वांचा वापर करून अनेक तार्किक (logical) समस्या सोडविल्या जावू शकतात. याच तत्वाचा वापर करून संगणकाची रचना निर्माण झाली. या रचनेमुळे अनेक कार्ये सोपी झाली, वेग व अचूकतासुद्धा वाढली.

## 2.2 मूलभूत तार्किक कार्य (Basic logic function)

### 2.2.1 धन (Positive) आणि ऋण (Negative) तर्क (Logic)-

कोणत्याही डिजिटल सर्किटमध्ये धन आणि ऋण तर्क वापरतात. जर कमी (Low) इनपुटला तर्क 0 तर जास्त (High) इनपुटला तर्क 1 मानले तर त्या तर्क पद्धतीला धन तर्क (Positive Logic) पद्धत असे म्हणतात. तसेच कमी इनपुटला तर्क 1 तर जास्त इनपुटला तर्क 0 मानले तर त्या तर्क पद्धतीला ऋण तर्क (Negative Logic) पद्धत असे म्हणतात.

### 2.2.2 लॉजिक गेट्स (Logic Gates) -

लॉजिक गेट्स हे तार्किक सर्किट असून याला एका पेक्षा जास्त इनपुट व एक आउटपुट असते. आउटपुटला फक्त दोनच स्थिती असतात. जास्त किंवा कमी आउटपुट हा इनपुटच्या स्थितीवर अवलंबून असतो.

### 2.2.3 सत्य सारणी (Truth Table)-

लॉजिक गेटच्या इनपुटच्या स्थिती व त्यानुसार मिळणारे आउटपुट यामधील संबंध दाखविणाऱ्या तक्त्यास सत्य सारणी असे म्हणतात.

### 2.2.4 तार्किक वाक्य (Logical Statement)-

द्विमान तर्क वापरून लॉजिक गेट साठी केलेली व्याख्या ही तार्किक वाक्य असते.

## 2.3 तार्किक द्वार (Logic Gates) -

सर्व डिजिटल इलेक्ट्रॉनिक्स उपकरणे जसे कॉम्प्युटर्स, कॅल्कुलेटर्स, इत्यादी तार्किक कार्यपद्धतीनुसार कार्य करतात म्हणून या उपकरणात वापरण्यात येणाऱ्या सर्किटला लॉजिक गेट म्हणतात. लॉजिक गेट द्विमान अंकपद्धतीनुसार कार्य करतात. याला एका पेक्षा जास्त इनपुट व एक आउटपुट असते. आउटपुटला फक्त दोनच स्थिती असतात, सत्य किंवा असत्य. या दोन अवस्थांवर कार्य करणाऱ्या सर्किटला व्दिस्थितीय सर्किट असे म्हणतात. उदाहरणार्थ बटन हे एकतर सुरु किंवा बंद असते. याच्या व्यतिरिक्त बटनाची कोणतीही अवस्था नसते.

लॉजिक गेट हे स्विच, डायोड, ट्रान्जिस्टर किंवा इंटिग्रेटेड सर्किट पासून तयार करतात. इंटिग्रेटेड सर्किटची विश्वसनीयता, उपलब्धता, किंमत व उपयोगिता जास्त असल्याने लॉजिक गेट साठी याचा वापर जास्त करतात.

### 2.3.1 लॉजिक गेटचे वर्गीकरण (Classification of Logic Gates)-

लॉजिक गेटचे खालील दोन प्रकार असतात

**A. मूलभूत गेट्स (Basic Gates)** – AND, OR व NOT ही मूलभूत गेट्स आहेत.

**B. साधित गेट्स (Derived Gates)** – NOR, NAND व EX -OR ही साधित गेट्स आहेत.

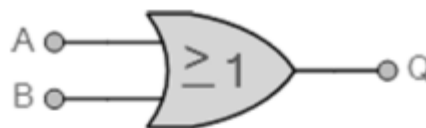
NAND, NOR गेटला **सार्वत्रिक गेट (Universal Gates)** असे म्हणतात कारण या गेट्सचा वापर करून दुसरे लॉजिक गेट्स तयार करता येतात.

### 2.3.2 मूलभूत गेट्स (Basic Gates)-

#### 2.3.2.1 OR गेट (OR Gate)

OR गेटला दोन किंवा जास्त इनपुट तर एक आउटपुट असते. जर A आणि B ही इनपुट असतील तर Q हे आउटपुट असते.

**चिन्ह (Symbol)**



आकृती 2.1- OR गेट

**बुलियन समीकरण (Boolean's Equation)**

$$Q = A + B$$

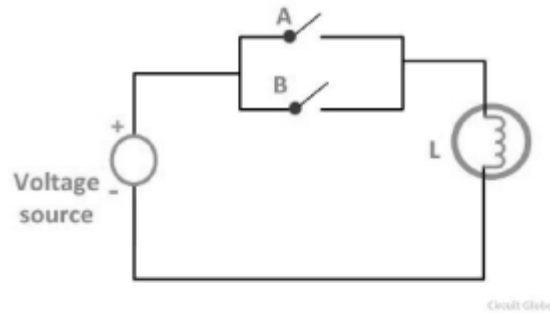
## सत्य सारणी (Truth Table)-

तक्ता 2.1- OR गेटची सत्य सारणी

इनपुट		आउटपुट
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

सत्य सारणीवरून हे स्पष्ट होते की जर सर्व इनपुट किंवा कोणतेही इनपुट जास्त असतील तर आउटपुट Q जास्त असेल, तर जर सर्व इनपुट कमी असतील, तर आउटपुट Q कमी असेल.

## स्विच सर्किट (Switch Circuit)



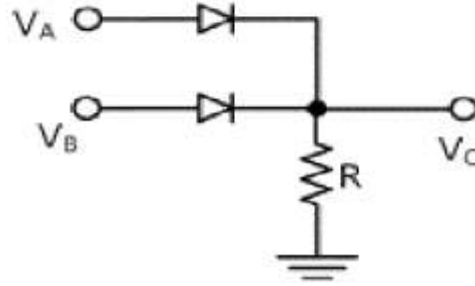
आकृती 2.2- स्विचचा वापर करून OR गेटची रचना

आकृती 2.2 प्रमाणे दोन स्विचेस A आणि B समांतर (Parallel) मध्ये जोडून त्याच्या एकसर (Series) बल्ब व बॅटरी जोडलेली आहे. स्विच बंद म्हणजे तर्क 0 व स्विच सुरु म्हणजे तर्क 1 वापरतात. स्विच सुरु



केल्यास सर्किट मधून विद्युतप्रवाह (करंट) वाहून बल्ब सुरु होतो. परंतु दोन्ही स्विच ओपन (open) असल्यास सर्किट मधून विद्युतप्रवाह (करंट) वाहत नसल्यामुळे बल्ब बंद राहतो.

### डायोड सर्किट (Diode Circuit)

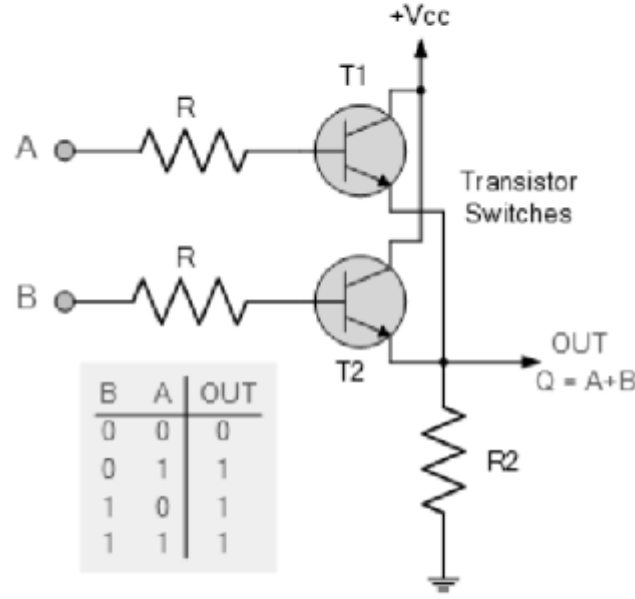


### आकृती 2.3- डायोडचा वापर करून OR गेटची रचना

दोन डायोड (Diode) दोन स्वतंत्र बॅटरीना जोडून OR गेटची कार्यप्रणाली स्पष्ट करता येते.

1.  $V_A = 0$   $V_B = 0$  असल्यास दोन्ही डायोड रिवर्स बायस (Reverse Biased) राहतात त्यामुळे विरोधक (Resistor) R चा ड्रॉप शून्य राहतो त्यामुळे  $V_C = 0$  होते.
2.  $V_A = 0$   $V_B = 1$  असल्यास एक डायोड रिवर्स बायस (Reverse Biased) व दुसरा डायोड फॉरवर्ड बायस (Forward Biased) होतो त्यामुळे विरोधक R मधून विद्युतप्रवाह वाहतो त्यामुळे  $V_C = 1$  होते.
3.  $V_A = 1$   $V_B = 0$  असल्यास एक डायोड रिवर्स बायस (Reverse Biased) व दुसरा डायोड फॉरवर्ड बायस (Forward Biased) होतो त्यामुळे विरोधक R मधून विद्युतप्रवाह वाहतो त्यामुळे  $V_C = 1$  होते.
4.  $V_A = 1$   $V_B = 1$  असल्यास दोन्ही डायोड फॉरवर्ड बायस (Forward Biased) राहतात त्यामुळे त्यामुळे विरोधक R मधून विद्युतप्रवाह वाहतो त्यामुळे  $V_C = 1$  होते.

## ट्रान्जिस्टर सर्किट (Transistor Circuit)



### आकृती 2.4- ट्रान्जिस्टरचा वापर करून OR गेटची रचना

दोन ट्रान्जिस्टर दोन स्वतंत्र बॅटरीना जोडून OR गेटची कार्यप्रणाली स्पष्ट करता येते.

ट्रान्जिस्टर समांतर(Parallel) जोडलेले आहेत आणि त्यांचे बेस इनपुट म्हणून वापरले जातात. दोन्ही ट्रान्जिस्टरचा बेस इनपुटप्रमाणे कार्य करतो आणि दोन्ही ट्रान्जिस्टरच्या एमिटर(Emitter) पैकी एकाचा वापर आउटपुट काढण्यासाठी केला जातो. सुरुवातीला, दोन्ही स्वीच बंद अवस्थेत असल्यामुळे कोणत्याही ट्रान्जिस्टर बेसला(Base) वीजपुरवठा होत नाही. दोन्ही ट्रान्जिस्टरच्या बेस टू एमिटर जंक्शन आणि बेस टू कलेक्टर जंक्शनमध्ये 0.7V पेक्षा कमी व्होल्टेज आहे, जो डायोडचा व्यावहारिक थ्रेशोल्ड व्होल्टेज (Threshold Voltage) आहे. दोन्ही जंक्शन रिव्हर्स बायसमध्ये आहेत म्हणून दोन्ही ट्रान्जिस्टर बंद होतात आणि त्यांच्या कटऑफ(Cutt Off) स्थितीत जातात. त्यामुळे ट्रान्जिस्टर ओपन स्विचसारखे काम करतात. म्हणून आउटपुटवर कमी व्होल्टेज मिळते, ज्यामुळे LED बंद होतो.

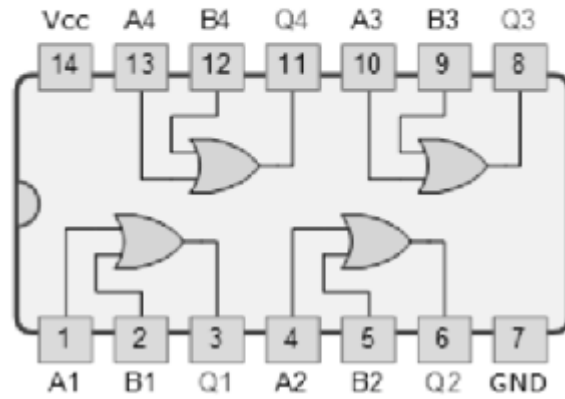
जेव्हा स्विच 1 बंद करतो तेव्हा पहिल्या ट्रान्जिस्टरच्या बेसला धन विद्युतदाब मिळतो. पहिल्या ट्रान्जिस्टरच्या बेस टू एमिटर जंक्शन ( Base to Emitter) आणि बेस टू कलेक्टर (Base to Collector) जंक्शनमध्ये थ्रेशोल्ड व्होल्टेजपेक्षा जास्त विद्युतदाब (Voltage) असते त्यामुळे दोन्ही जंक्शन फॉरवर्ड बायसमध्ये असतात. पहिला ट्रान्जिस्टर सॅच्युरेशन (Saturation) अवस्थेत पोहोचतो आणि शॉर्ट सर्किट प्रमाणे काम करतो तर दुसरा ट्रान्जिस्टर अद्याप वीज पुरवठा नसल्यामुळे कटऑफ (Cutoff) स्थितीत

आहे. त्यामुळे सर्व विद्युतप्रवाह पहिल्या ट्रान्झिस्टरमधून आउटपुटमध्ये जातो आणि एलईडी दिवे प्रकाशित होतात.

त्याचप्रमाणे, जर आपण स्विच 1 उघडला आणि स्विच 2 बंद केला तर पहिला ट्रान्झिस्टर कटऑफमध्ये जाईल तर दुसरा ट्रान्झिस्टर सॅच्युरेशन (Saturation) मध्ये जाईल. आता दुस-या ट्रान्झिस्टरमधून विद्युतप्रवाह वाहतो आणि आउटपुट जास्त(1) होतो

जेव्हा आपण दोन्ही स्विच बंद करतो, तेव्हा दोन्ही ट्रान्झिस्टरच्या जंक्शनमध्ये थ्रेशोल्ड व्होल्टेजपेक्षा जास्त विद्युतदाब असते त्यामुळे दोन्ही जंक्शन फॉरवर्ड बायसमध्ये असतात. त्यामुळे दोन्ही ट्रान्झिस्टर सॅच्युरेशन अवस्थेत असतात आणि शॉर्ट सर्किटसारखे काम करतात. विद्युतप्रवाहाला आता शॉर्ट सर्किट मार्ग मिळतो आणि दोन्ही ट्रान्झिस्टरमधून प्रवाहित होतो ज्यामुळे LED प्रकाशित होतो.

### Quad 2-input Logic OR Gate (7432)

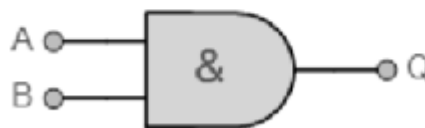


आकृती 2.5- Pin Diagram

### 2.3.2.2. AND गेट

या गेटला दोन किंवा जास्त इनपुट तर एक आउटपुट असते. जर A आणि B इनपुट असतील तर Y आउटपुट असते.

चिन्ह



आकृती 2.6- AND गेट

## बुलियन समीकरण

$$Q = A.B$$

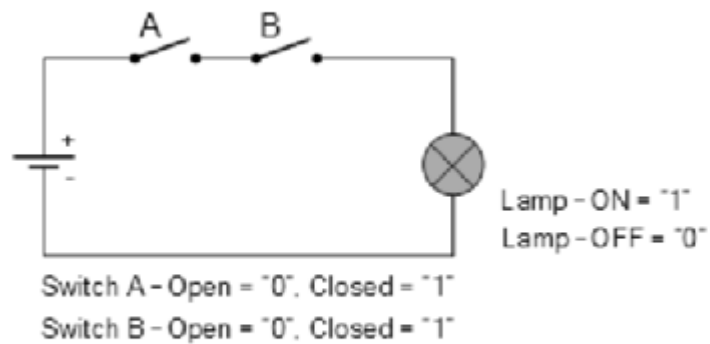
सत्य सारणी -

तक्ता 2.2- AND गेटची सत्य सारणी

इनपुट		आउटपुट
A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

सत्य सारणीवरून हे स्पष्ट होते की दोन्ही इनपुट जास्त असतानाच आउटपुट जास्त असते.

## स्विच सर्किट (Switch Circuit)



आकृती 2.7- स्विचचा वापर करून AND गेटची रचना

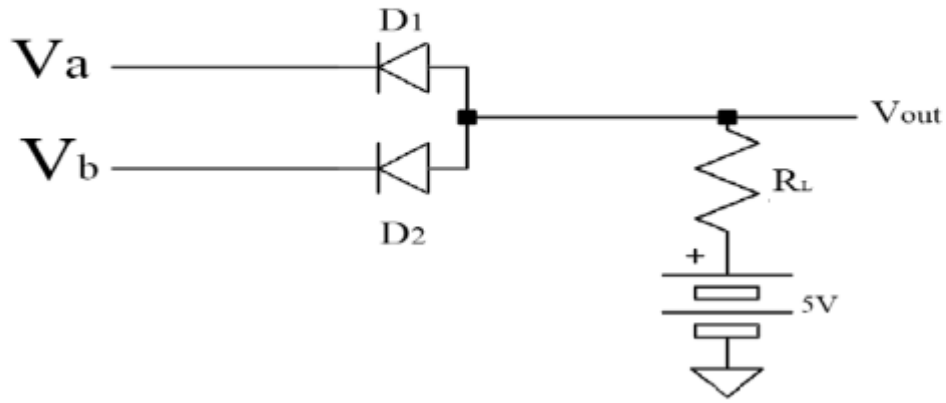
आकृती 2.7 प्रमाणे दोन स्विचेस A आणि B एकसर (Series) मध्ये जोडून त्याच्या एकसर बल्ब व बॅटरी जोडलेली दिसते.

त्यामुळे जेव्हा दोन्ही स्विच दाबले जात नाहीत तेव्हा याचा अर्थ इनपुट तर्क "0 आणि 0" असा होतो, अशा परिस्थितीत LED प्रकाशित होणार नाही.

जेव्हा कोणतेही एक स्विच दाबला जातो, तेव्हा LED प्रकाशित होणार नाही, याचा अर्थ तर्क "0 आणि 1" किंवा "1 आणि 0" असा होतो.

जेव्हा दोन्ही स्विच दाबले जातात तेव्हा याचा अर्थ तर्क "1 आणि 1" असा होतो. तेव्हाच, LED प्रकाशित होतील.

### डायोड सर्किट(Diode Circuit)



### आकृती 2.8- डायोडचा वापर करून AND गेटची रचना

जर  $V_A$  जास्त असेल आणि  $V_B$  कमी असेल, तर डायोड D1 रिव्हर्स बायस्ड बनतो म्हणून ओपन स्विच म्हणून काम करतो. तसेच, डायोड D2 फॉरवर्ड बायस्ड बनतो म्हणून बंद(Close) स्विच म्हणून काम करतो. त्यामुळे आउटपुट कमी मिळते.

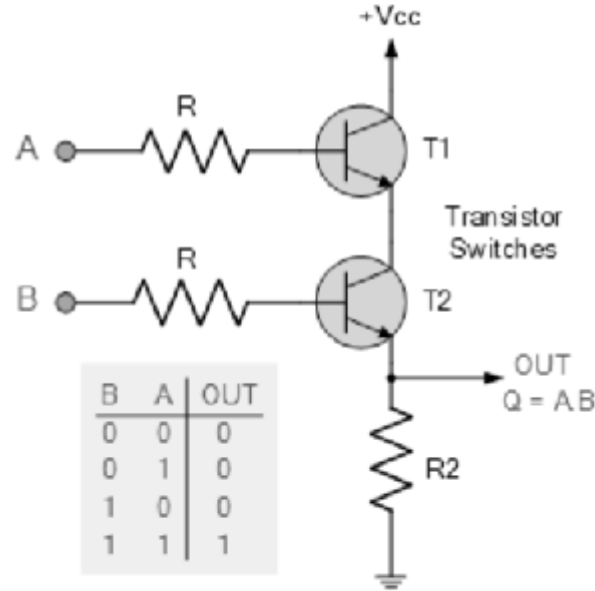
$V_A$  कमी आणि  $V_B$  जास्त असल्यास, डायोड D1 फॉरवर्ड बायस्ड बनतो म्हणून बंद स्विच म्हणून काम करतो. तसेच, डायोड D2 रिव्हर्स बायस्ड बनतो म्हणून ओपन स्विच म्हणून काम करतो. आउटपुट कमी मिळते.

जर सर्व इनपुट कमी असतील तर सर्व डायोड फॉरवर्ड बायस्ड होतात आणि ओपन स्विच म्हणून काम करतात. त्यामुळे आउटपुट कमी मिळते.



जेव्हा सर्व इनपुट्स जास्त असतात तेव्हा सर्व डायोड रिव्हर्स बायस्डमध्ये बनतात म्हणून ओपन स्विच म्हणून काम करतात. त्यामुळे आउटपुट जास्त मिळते.

### ट्रान्झिस्टर सर्किट(Transistor Circuit)



### आकृती 2.9- ट्रान्झिस्टरचा वापर करून AND गेटची रचना

या मध्ये ट्रान्झिस्टर एकसर (Series) जोडलेले असतात आणि त्यांचे बेस इनपुट म्हणून वापरले जातात. दोन्ही ट्रान्झिस्टरचे बेस इनपुट्सप्रमाणे काम करतात आणि आउटपुट काढण्यासाठी ट्रान्झिस्टरपैकी एकाचा वापर केला जातो. सुरुवातीला, दोन्ही स्विच बंद अवस्थेत असतात त्यामुळे कोणत्याही ट्रान्झिस्टर बेसला वीजपुरवठा मिळत नाही. दोन्ही ट्रान्झिस्टरच्या बेस टू एमिटर जंक्शन आणि बेस टू कलेक्टर जंक्शनमध्ये 0.7V पेक्षा कमी विद्युतदाब असतो, जो डायोडचा थ्रेशोल्ड विद्युतदाब (Threshold Voltage) आहे.

दोन्ही जंक्शन रिव्हर्स बायसमध्ये आहेत म्हणून दोन्ही ट्रान्झिस्टर बंद होतात आणि त्यांच्या कटऑफ स्थितीत जातात. म्हणून, ट्रान्झिस्टर ओपन स्विचसारखे कार्य करतात म्हणून आउटपुटला कमी विद्युतदाब मिळतो त्यामुळे एलईडी बंद होतात.

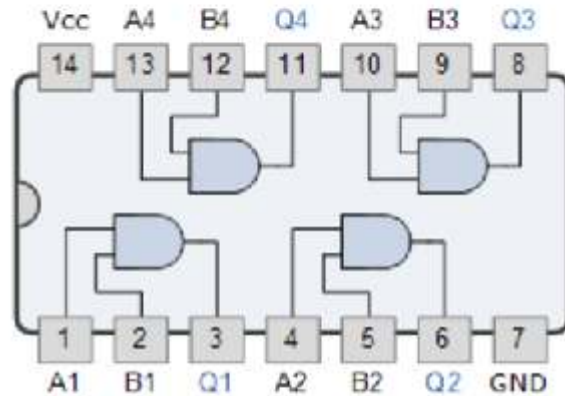
पुढच्या बाबतीत जेव्हा स्विच 1 ऑन करतो तेव्हा पहिल्या ट्रान्झिस्टरच्या बेसला विद्युतदाबाचे पॉसिटीव्ह मूल्य मिळते परंतु त्याचा एमीटर दुसऱ्या ट्रान्झिस्टर कलेक्टरशी जोडलेले असतो . दुसरा

ट्रान्झिस्टर अजूनही त्याच्या कटऑफ स्थितीत असल्याने, पहिल्या ट्रान्झिस्टरचा एमीटर डिस्कनेक्ट होतो . दोन्ही ट्रान्झिस्टरच्या बेस टू एमिटर जंक्शन आणि बेस टू कलेक्टर जंक्शनमध्ये थ्रेशोल्ड विद्युतदाबापेक्षा कमी विद्युतदाब असतो आणि ते पुन्हा त्यांच्या कटऑफ स्थितीत पोहोचतात. सर्व विद्युतप्रवाह पुन्हा ट्रान्झिस्टरद्वारे अवरोधित केला जातो आणि आउटपुटवर कमी व्होल्टेज मिळते, जे LED बंद करते.

जेव्हा स्विच 1 उघडला जातो आणि स्विच 2 बंद केला जातो, तेव्हा पहिल्या ट्रान्झिस्टरद्वारे विद्युत प्रवाह अवरोधित केला जातो आणि आउटपुट कमी होते.

जेव्हा आपण दुसरा स्विच दाबतो, तेव्हा दोन्ही ट्रान्झिस्टरच्या जंक्शनमध्ये थ्रेशोल्ड व्होल्टेजपेक्षा जास्त विद्युतदाब असतो त्यामुळे दोन्ही जंक्शन फॉरवर्ड बायस असतात. म्हणून, दोन्ही ट्रान्झिस्टर सॅच्युरेशन स्थितीत जातात आणि शॉर्ट सर्किटसारखे कार्य करतात. विद्युतप्रवाहाला आता शॉर्ट सर्किट मार्ग मिळतो आणि पहिल्या ट्रान्झिस्टरच्या कलेक्टर पासून दुसऱ्या ट्रान्झिस्टरच्या एमीटर कडे वाहतो जो आउटपुटमध्ये उच्च असतो आणि LED प्रकाशित होतात.

#### Quad 2-input AND Gate (7408)

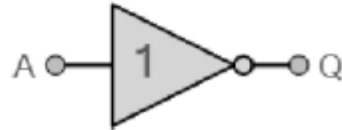


आकृती 2.10- Pin Diagram

### 2.3.2.3 NOT गेट

या गेटला इन्वर्टर असेही म्हणतात. या गेटला एकच इनपुट व एक आउटपुट असते.

चिन्ह



आकृती 2.11- NOT गेट

बुलियन समीकरण

$$Q = \bar{A}$$

सत्य सारणी-

तक्ता 2.3- NOT गेटची सत्य सारणी

इनपुट	आउटपुट
A	Q
0	1
1	0

सत्य सारणीवरून हे स्पष्ट होते की इनपुट जास्त असल्यास आउटपुट कमी असते तसेच इनपुट कमी असल्यास आउटपुट जास्त असते.

## स्विच सर्किट (Switch Circuit)

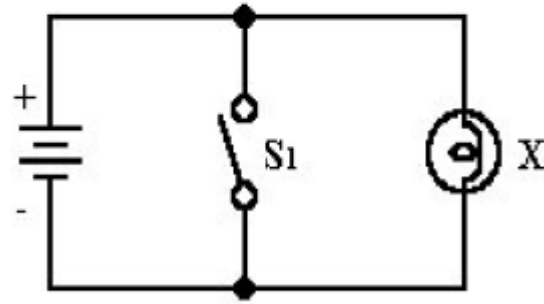
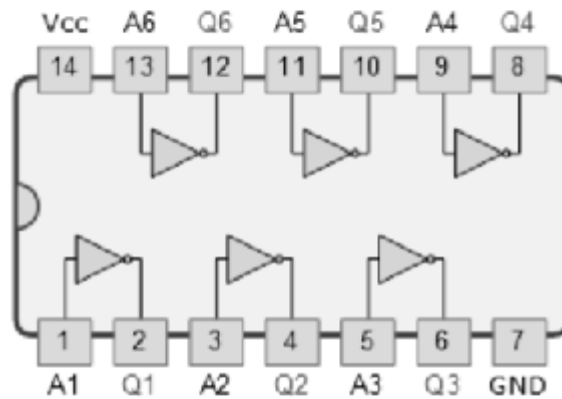


Figure 2.12- Switch Circuit

आकृती 2.12- स्विचचा वापर करून NOT गेटची रचना

स्विचिंग सर्किट स्पष्ट करते की स्विच S1 उघडल्यावर दिवा प्रकाशित होईल आणि S1 स्विच बंद झाल्यावर तो बंद होईल.

## Logic NOT Gate or Inverter (7404)



आकृती 2.13- Pin Diagram

### 2.3.3 साधित गेट्स (Derived Gates)

#### 2.3.3.1 NAND गेट (NAND Gate)

NAND गेट किंवा "NotAND" गेट हे दोन मूलभूत लॉजिक गेट्सचे संयोजन आहे, AND गेट आणि एकसर जोडलेले नॉट गेट. NAND गेट आणि NOR गेटला सार्वत्रिक गेट (युनिव्हर्सल गेट्स) म्हटले जाऊ शकते

कारण या गेट्सच्या संयोजनाचा वापर कोणत्याही मूलभूत ऑपरेशन्स पूर्ण करण्यासाठी केला जाऊ शकतो. म्हणून, NAND गेट आणि NOR गेटचे संयोजन इन्व्हर्टर, OR गेट किंवा AND गेट तयार करू शकते.

एकजरी इनपुट जास्त असल्यास किंवा दोन्ही इनपुट कमी असल्यास NAND गेटचे आउटपुट जास्त असते.

**चिन्ह**



आकृती 2.14- NAND गेट

**बुलियन समीकरण**

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

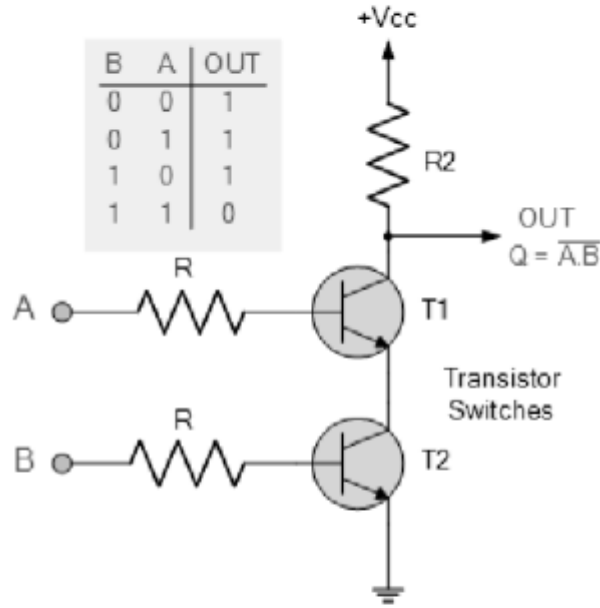
**सत्यसारणी -**

तक्ता 2.4- NAND गेटची सत्य सारणी

इनपुट		आउटपुट
A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0



## ट्रान्जिस्टर सर्किट (Transistor Circuit)

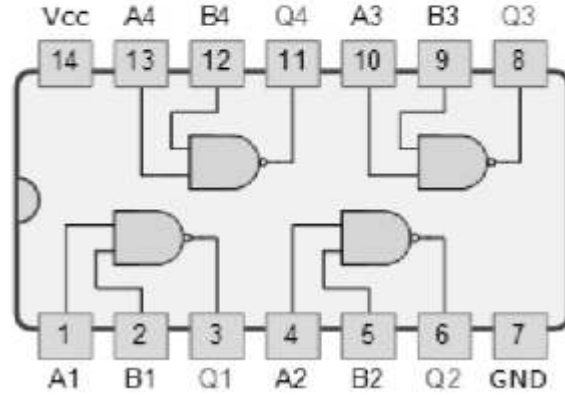


आकृती 2.15- ट्रान्जिस्टरचा वापर करून NAND गेटची रचना

जेव्हा NAND गेटचे कोणतेही एक इनपुट 0 विद्युतदाबाशी जोडलेले असते, तेव्हा संबंधित ट्रान्जिस्टर बंद स्थितीत असतो. त्यामुळे रेझिस्टरमधून कोणताही विद्युतप्रवाह वाहत नाही आणि विरोधकामध्ये विद्युतदाब ड्रॉप होत नाही. मग आउटपुट विद्युतदाब हे पुरवठा विद्युतदाब(Supply Voltage) +5 व्होल्ट म्हणजेच जास्त तर्क पातळीच्या (Logic Level) बरोबरीचे असते.

जेव्हा NAND गेटचे दोन्ही इनपुट पुरवठा विद्युतदाब +5 व्होल्टशी जोडलेले असतात, तेव्हा ट्रान्जिस्टर T1 आणि T2 चालू स्थितीत असतात. त्यामुळे विद्युतप्रवाह विरोधक, ट्रान्जिस्टरमधून वाहतो. मग आउटपुट विद्युतदाब VCE 0 व्होल्ट होते व कमी तर्क मिळतो.

### Quad 2 input NAND gate (7400)

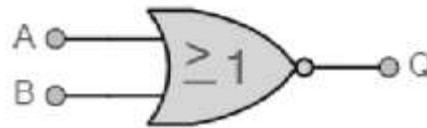


आकृती 2.16- Pin Diagram

### 2.3.3.2 NOR गेट ( NOR Gate)

NOR गेट ("not OR गेट" ) हे एक लॉजिक गेट आहे जे उच्च आउटपुट (1) फक्त त्याचे सर्व इनपुट कमी असल्यास मिळते अन्यथा कमी आउटपुट (0) उत्पन्न करते. म्हणून NOR गेट हे OR गेटच्या उलट काम करते. त्याचे सर्किट OR गेटला NOT गेटशी जोडून तयार केले जाते. OR गेट प्रमाणे, NOR गेटमध्ये कितीही इनपुट असू शकतात परंतु फक्त एक आउटपुट असते.

चिन्ह



आकृती 2.17- NOR गेट

बुलियन समीकरण

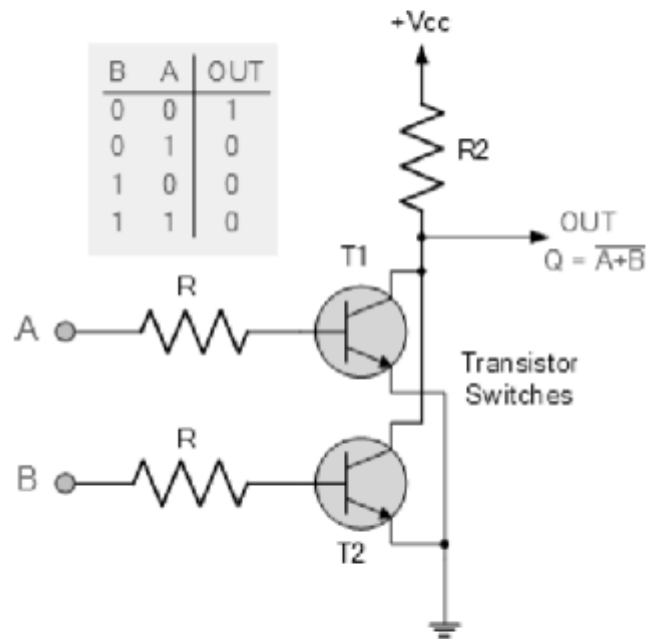
$$Q = \overline{A+B}$$

## सत्यसारणी-

तक्ता 2.5- NOR गेटची सत्य सारणी

इनपुट		आउटपुट
A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

## ट्रान्जिस्टर सर्किट (Transistor Circuit)



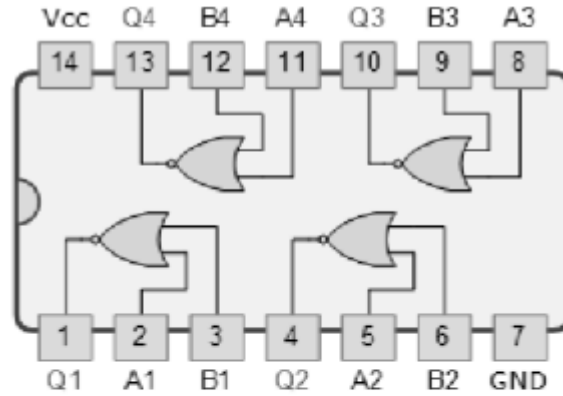
आकृती 2.18- ट्रान्जिस्टरचा वापर करून NOR गेटची रचना

जेव्हा NOR गेटचे दोन्ही इनपुटना 0 विद्युतदाब दिला जातो, तेव्हा ट्रान्जिस्टर T1 आणि T2 बंद स्थितीत असतात. त्यामुळे विरोधकामधून कोणताही विद्युतप्रवाह वाहत नाही आणि विरोधकामध्ये

विद्युतदाब ड्रॉप होत नाही. मग आउटपुट विद्युतदाब हा +5 व्होल्ट म्हणजेच जास्त तर्क पातळीच्या (Logic Level) बरोबरीचा असतो.

इनपुटपैकी कोणतेही एक +5 V शी जोडलेले असल्यास, ट्रान्झिस्टर चालू स्थितीत असतील. त्यामुळे विद्युतदाब ड्रॉप जास्त असेल. त्यामुळे सर्किटचे आउटपुट व्होल्टेज 0 V असेल म्हणजेच ग्राउंड व्होल्टेजच्या म्हणजेच कमी तर्क पातळीच्या बरोबरीचे असेल.

#### Quad 2-input Logic NOR Gate (7402)



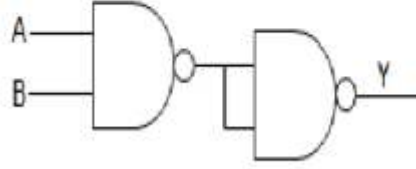
आकृती 2.19-Pin Diagram

#### 2.3.4 NAND व NOR याचा सार्वत्रिक गेट ( Universal Gate) म्हणून वापर

सार्वत्रिक गेट (Universal Gate) हे असे लॉजिक गेट आहे जे इतर कोणत्याही प्रकारचे लॉजिक गेट न वापरता कोणतेही बुलियन फंक्शन लागू करू शकते. NOR गेट आणि NAND गेट ही युनिव्हर्सल गेट आहेत. याचा अर्थ असा की फक्त NOR गेट्स किंवा फक्त NAND गेट्स वापरून कोणतीही लॉजिकल बुलियन एक्सप्रेसन (Boolean Expression) तयार करू शकतो.

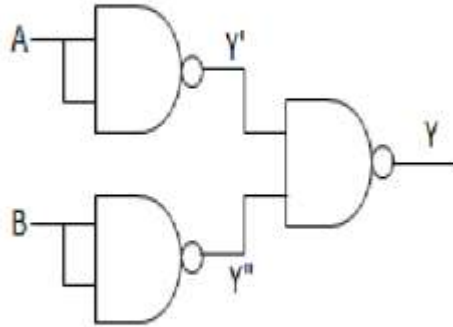
#### 2.3.4.1 NAND गेटचा सार्वत्रिक गेट म्हणून वापर

AND गेट खालील पद्धतीमध्ये दोन NAND गेट्स वापरून तयार केले जाऊ शकते.



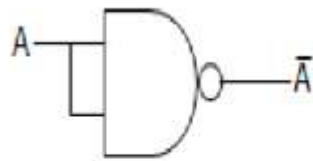
आकृती 2.20

OR गेट खालीलप्रमाणे NAND गेट वापरून तयार केले जाऊ शकते.



आकृती 2.21

NOT गेट खालीलप्रमाणे NAND गेट वापरून तयार केले जाऊ शकते

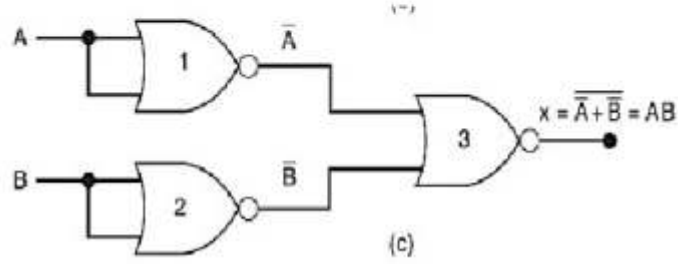


आकृती 2.22



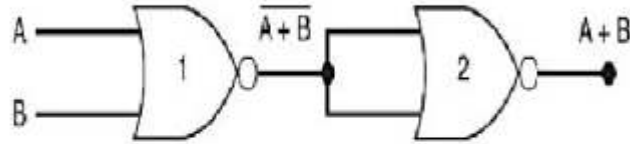
#### 2.3.4.2. NOR गेटचा सार्वत्रिक गेट म्हणून वापर

AND गेट खालील पद्धतीमध्ये दोन NOR गेट्स वापरून तयार केले जाऊ शकते.



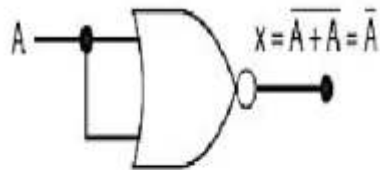
आकृती 2.23

OR गेट खालीलप्रमाणे NOR गेट वापरून तयार केले जाऊ शकते.



आकृती 2.24

NOT गेट खालीलप्रमाणे NOR गेट वापरून तयार केले जाऊ शकते

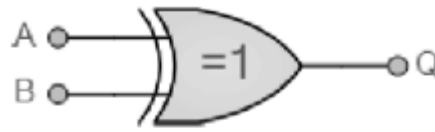


आकृती 2.25

### 2.3.5 XOR गेट

XOR गेट (याला EOR, किंवा EXOR गेट असेही म्हणतात) - ज्याला एक्सक्लुझिव्ह गेट (Exclusive Gates) म्हणून उच्चारले जाते - हे एक डिजिटल लॉजिक गेट आहे जे गेटच्या इनपुटपैकी फक्त एक जास्त असल्यास जास्त आउटपुट परिणाम येतो. दोन्ही इनपुट कमी असल्यास (म्हणजे LOW किंवा 0) किंवा दोन्ही इनपुट जास्त असल्यास, आउटपुट कमी देते .

चिन्ह



आकृती 2.26- XOR गेट

बुलियन समीकरण

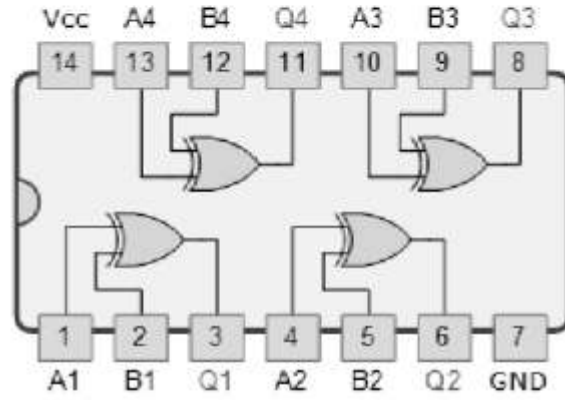
$$Q = A \oplus B$$

सत्यसारणी-

तक्ता 2.6- XOR गेटची सत्य सारणी

इनपुट		आउटपुट
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

### Quad 2 input EX-OR gate (7486)

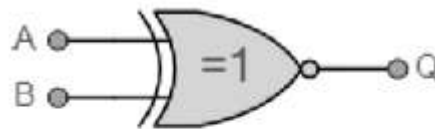


आकृती 2.27- Pin Diagram

### 2.3.6 XNOR गेट (XNOR Gate)

XNOR गेट (ज्याला ENOR, EXNOR किंवा NXOR म्हणून देखील ओळखले जाते) - आणि एक्सक्लुझिव्ह NOR म्हणून उच्चारले जाते. हे एक डिजिटल लॉजिक गेट आहे ज्याचे कार्य एक्सक्लुझिव्ह OR (XOR गेट) चे तार्किक पूरक आहे. तार्किकदृष्ट्या, XNOR गेट हे एक NOT गेट व त्यानंतर XOR गेट आहे.

चिन्ह



आकृती 2.28- XNOR गेट

बुलियन समीकरण

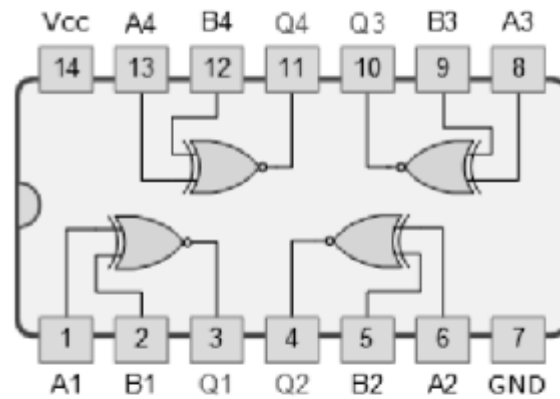
$$Q = \overline{A \oplus B}$$

सत्य सारणी -

तक्ता 2.7- XNOR गेटची सत्य सारणी

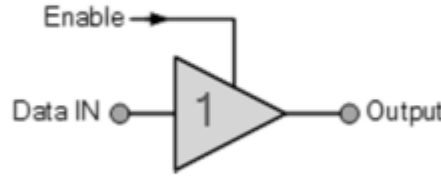
इनपुट		आउटपुट
A	B	Q
0	0	1
0	1	0
1	0	0
1	1	1

Quad 2-input Ex-NOR Gate(74266)



आकृती 2.29- Pin Diagram

### 2.3 ट्राय-स्टेट लॉजिक (Tri State Logic)



आकृती 2.30- ट्राय-स्टेट लॉजिक

डिजिटल इलेक्ट्रॉनिक्स मध्ये थ्री-स्टेट, ट्राय-स्टेट, किंवा 3-स्टेट लॉजिक आउटपुट किंवा इनपुट पिनला 0 आणि 1 लॉजिक स्तरांव्यतिरिक्त, सर्किटमधून आउटपुट प्रभावीपणे काढून टाकून, उच्च प्रतिबाधा (Hi impedance state) स्थिती गृहित धरण्यास अनुमती देते. हे एकापेक्षा अधिक सर्किट्सला समान आउटपुट लाइन सामायिक करण्याची अनुमती देते.

थ्री-स्टेट आउटपुट 7400 आणि 4000 मालिकेतील अनेक रजिस्टर्स, बस ड्रायव्हर्स आणि फ्लिप-फ्लॉपमध्ये तसेच इतर प्रकारांमध्ये लागू केले जातात, परंतु अनेक एकात्मिक सर्किट्समध्ये अंतर्गत देखील. मायक्रोप्रोसेसर, कॉम्प्युटर मेमरी आणि पेरिफेरल्समधील अंतर्गत आणि बाह्य बसेसचे इतर विशिष्ट उपयोग आहेत. अनेक उपकरणे OE (आउटपुट एनाबल) नावाच्या सक्रिय इनपुटद्वारे नियंत्रित केली जातात जी आउटपुट उच्च-इम्पीडन्स स्थितीत ठेवली जातात किंवा त्यांचे संबंधित भार (एकतर 0-किंवा 1-स्तरीय) चालवावी की नाही हे ठरवते.

तक्ता 2.8- ट्राय-स्टेट लॉजिक

Enable	IN	OUT
0	0	Hi-Z
0	1	Hi-Z
1	0	0
1	1	1

## 2.5 बुलियन अलजेब्रा (Boolean Algebra)

बुलियन अलजेब्रा हे एक गणितशास्त्र आहे जे आपण डिजिटल गेट्स आणि सर्किट्सचे विश्लेषण करण्यासाठी वापरतो. आवश्यक लॉजिक गेट्सची संख्या कमी करण्यासाठी तसेच जटिल बुलियन एक्सप्रेशन कमी आणि सुलभ करण्यासाठी हे "बुलियनचे नियम" वापरू शकतो. म्हणून बुलियन अलजेब्रा ही तर्कशास्त्रावर आधारित गणिताची एक प्रणाली आहे ज्याचे स्वतःचे नियम आहेत, जे बुलियन एक्सप्रेशन परिभाषित करण्यासाठी आणि कमी करण्यासाठी वापरले जातात.

बुलियन अलजेब्रामध्ये वापरल्या जाणाऱ्या व्हेरिएबल्समध्ये फक्त दोन संभाव्य मूल्यांपैकी एक असते, एक तर्क "0" आणि एक तर्क "1" परंतु एक्सप्रेशनमध्ये जास्त संख्येने व्हेरिएबल्स असू शकतात जे सर्व एक्सप्रेशनचे इनपुट दर्शवण्यासाठी स्वतंत्रपणे लेबल केलेले असू शकतात, उदाहरणार्थ, व्हेरिएबल्स A, B, C इत्यादी,  $A + B = C$  चे तार्किक एक्सप्रेशन देतात, परंतु प्रत्येक व्हेरिएबल फक्त 0 किंवा 1 असू शकते.

### 2.5.1 बुलियन अलजेब्राचे नियम (Laws of Boolean algebra)

#### A. कॉम्यूटेटिव्ह लॉ (Commutative Law)

दोन स्वतंत्र व्हेरिएबल लागू करण्याचा क्रम महत्वाचा नाही.

$$A \cdot B = B \cdot A$$

$$A + B = B + A$$

#### B. असोसिएटिव्ह लॉ (Associative Law)

हा नियम एक्सप्रेशनमधून कंस काढण्याची आणि व्हेरिएबल्सचे पुनर्गठन करण्यास परवानगी देतो.

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

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



### C. डिस्ट्रीब्युटिव्ह लॉ (Distributive Law)

हा नियम एक्सप्रेशनमधून गुणाकार किंवा गुणांक काढण्याची परवानगी देतो.

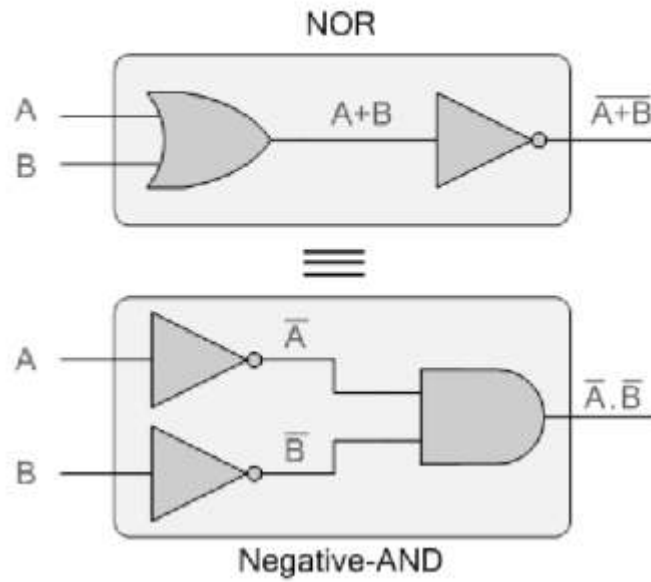
$$A (B + C) = A.B + A.C$$

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

### D. डी मॉर्गनची थेरम (De Morgan's Theorem)

**पहिला नियम-** बेरजेचा कॉम्प्लिमेंट हा प्रत्येक कॉम्प्लिमेंटच्या गुणाकाराइतका असतो.

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



आकृती 2.31- डी मॉर्गनचा पहिला नियम

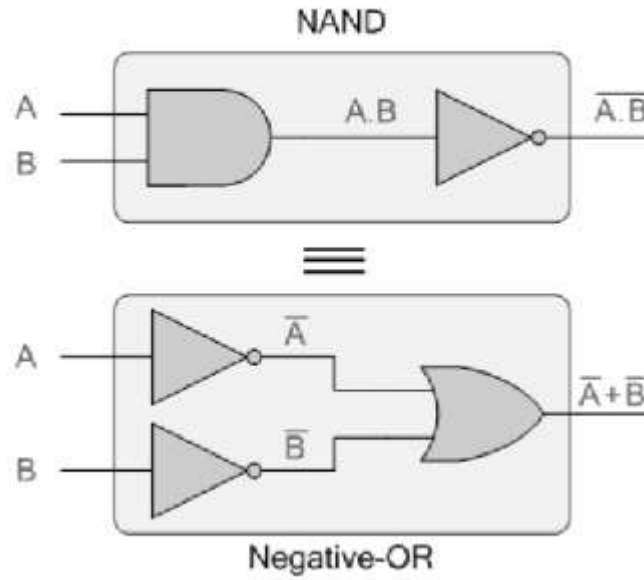
डी मॉर्गनच्या पहिल्या नियमाची सत्यता दर्शविणारा ट्रुथ टेबल

तक्ता 2.9- डी मॉर्गनच्या पहिल्या नियमाची सत्य सारणी

A	B	$\overline{A+B}$	$\overline{A}$	$\overline{B}$	$\overline{A} \cdot \overline{B}$
0	0	1	1	1	1
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	0

**दुसरा नियम-** गुणाकाराचा कॉम्प्लिमेंट हा प्रत्येक कॉम्प्लिमेंटच्या बेरजे इतका असतो.

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



**आकृती 2.32- डी मॉर्गनचा दुसरा नियम**

डी मॉर्गनच्या दुसऱ्या नियमाची सत्यता दर्शविणारा दुथ टेबल

**तक्ता 2.10- डी मॉर्गनच्या दुसऱ्या नियमाची सत्य सारणी**

A	B	$\overline{A \cdot B}$	$\overline{A}$	$\overline{B}$	$\overline{A} + \overline{B}$
0	0	1	1	1	1
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	0	0	0

## 2.6 डुआलिटी थेरम (Duality Theorem)

हा नियम असे सांगतो की बूलियन फंक्शनशी संबंधित ड्युअल फंक्शन हे लॉजिकल AND ऑपरेटरची लॉजिकल OR ऑपरेटर बरोबर किंवा एकची शून्यासह अदलाबदल करून प्राप्त होते.

तक्ता 2.11- डुआलिटी थेरम

Group1	Group2
$x + 0 = x$	$x.1 = x$
$x + 1 = 1$	$x.0 = 0$
$x + x = x$	$x.x = x$
$x + x' = 1$	$x.x' = 0$
$x + y = y + x$	$x.y = y.x$
$x + y+zy+z = x+yx+y + z$	$x.y.zy.z = x.yx.y.z$

प्रत्येक ओळीमध्ये, दोन बुलियन समीकरणे आहेत आणि ती एकमेकांशी ड्युअल आहेत. आपण ड्युअल थेरम वापरून ग्रुप 1 आणि ग्रुप 2 ची ही सर्व बुलियन समीकरणे सत्यापित करू शकतो.

## 2.7 लॉजिक फॅमिलीज (Logic Families)

डिजिटल सिस्टिम्स जसे डिजिटल मोजमापन करणारी उपकरणे , कॉम्प्युटर किंवा मायक्रो प्रोसेसरसाठी शंभरपेक्षा जास्त लॉजिक गेट्स स्वीचच्या कार्यासाठी वापरली जातात. अनेक सुसंगत असलेल्या, सारख्या लॉजिक लेवल व पुरवठ्यावर कार्य करणाऱ्या परंतु वेगवेगळे लॉजिक कार्य करणारी आय. सी. एका लहान चिपवर तयार करणे आता शक्य झाले. या आय. सी. ला डिजिटल लॉजिक फॅमिली असे म्हणतात.

### 2.7.1 डिजिटल लॉजिक फॅमिलीजचे प्रकार

#### 2.7.1.1. युनीपोलार लॉजिक फॅमिलीज

##### युनीपोलार लॉजिक फॅमिलीजचे प्रकार

1. PMOS

2. NMOS

3. CMOS - CMOS हे कॉम्प्लिमेंटरी मेटल ऑक्साईडचे संक्षिप्त रूप आहे. हे सेमीकंडक्टर (Semiconductor) आणि इंटीग्रेटेड सर्किट्सच्या (Integrated Circuit) बनवण्याचे तंत्रज्ञान आहे. जे विविध अनुप्रयोगांमध्ये (Applications) वापरले जातात. CMOS तंत्रज्ञान वापरून विविध प्रकारचे इंटीग्रेटेड सर्किट्स वापरून तयार केले जातात जसे की मायक्रोप्रोसेसर (Microprocessor), मायक्रोकंट्रोलर (Microcontroller), मेमरी चिप्स (Memory Chips) आणि इतर अनेक डिजिटल लॉजिक सर्किट्स. डेटा कन्व्हर्टर, इमेज सारख्या स्टॅटिक ऍनालॉग सर्किट्समध्ये सेन्सर्स (Sensors) मध्ये हे तंत्रज्ञान मोठ्या प्रमाणावर वापरले जाते.

### 2.7.1.2. बायपोलार लॉजिक फॅमिलीज (Bipolar Logic Families)

#### A. बायपोलार लॉजिक फॅमिलीजचे प्रकार

1. RTL - रेसिस्टर-ट्रान्जिस्टर लॉजिक
2. DTL - डायोड- ट्रान्जिस्टर लॉजिक
3. TTL - ट्रान्जिस्टर- ट्रान्जिस्टर लॉजिक

### 2.7.1.3 लॉजिक फॅमिलीजची वैशिष्ट्ये

#### 1. व्होल्टेज आणि करंट पॅरामीटर्स

0V आणि +5V (TTL साठी) च्या इनपुट व्होल्टेज स्तरांना अनुक्रमे लॉजिक 0 आणि 1 स्तर म्हणतात. तथापि, व्यावहारिकदृष्ट्या नेहमी या मूल्यांशी जुळणारे व्होल्टेज स्तर प्राप्त होत नाहीत. म्हणून सर्वात वॉर्सट -केस इनपुट व्होल्टेज परिभाषित करणे आवश्यक आहे.

#### A. व्होल्टेज पॅरामीटर्स

#### 1. $V_{IL}(\max)$ – वॉर्सट केस लो लेवल इनपुट व्होल्टेज (worst case low level input voltage)

हे इनपुट व्होल्टेजचे कमाल मूल्य आहे जे लॉजिक 0 स्तर म्हणून मानले जाईल. जर इनपुट व्होल्टेज  $V_{IL}(\text{कमाल})$  पेक्षा जास्त असेल, तर ते कमी (0) इनपुट स्तर मानले जाणार नाही.

**2.  $V_{IH}(\max)$  – वॉस्ट केस हाय लेवल इनपुट व्होल्टेज (worst case high level input voltage)**

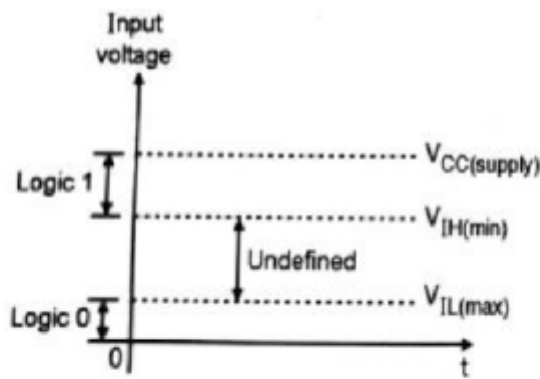
हे इनपुट व्होल्टेजचे किमान मूल्य आहे जे लॉजिक 1 स्तर म्हणून मानले जाईल. जर इनपुट व्होल्टेज  $V_{IH}(\min)$  पेक्षा कमी असेल, तर ते उच्च (1) इनपुट स्तर मानले जाणार नाही.

**3.  $V_{OL}(\max)$  – वॉस्ट केस लो लेवल आउटपुट व्होल्टेज (worst case low level output voltage)**

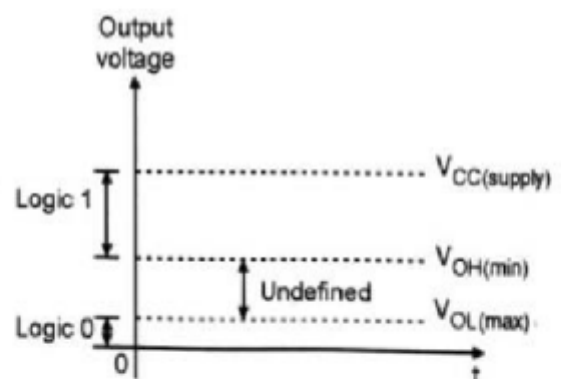
हे आउटपुट व्होल्टेजचे कमाल मूल्य आहे जे लॉजिक 0 पातळी म्हणून मानले जाईल. जर इनपुट व्होल्टेज  $V_{OL}(\max)$  पेक्षा जास्त असेल, तर ते कमी (0) आउटपुट पातळी म्हणून मानले जाणार नाही.

**4.  $V_{OH}(\max)$  – वॉस्ट केस हाय लेवल आउटपुट व्होल्टेज (worst case high level output voltage)**

हे आउटपुट व्होल्टेजचे किमान मूल्य आहे जे लॉजिक 1 स्तर म्हणून मानले जाईल. जर आउटपुट व्होल्टेज  $V_{OH}(\min)$  पेक्षा कमी असेल, तर ते उच्च (1) आउटपुट स्तर मानले जाणार नाही.



(a) Input voltage parameters



(b) Output voltage parameters

Voltage parameters

**आकृती 2.33- व्होल्टेज पॅरामीटर्स**

**B. करंट पॅरामीटर्स**

### 1. $I_{IL}$ – Low-level input Current (लो लेवल इनपुट करंट)

निम्न पातळीचे इनपुट व्होल्टेज लागू केले जाते तेव्हा प्रवाहित होणारा विद्युतप्रवाह म्हणजे लो लेवल इनपुट करंट होय.

### 2. $I_{OL}$ – Low-level output Current (लो लेवल आउटपुट करंट )

निम्न पातळीचे इनपुट व्होल्टेज लागू केले जाते तेव्हा प्रवाहित होणारा जास्तीत जास्त विद्युतप्रवाह म्हणजे हाय लेवल इनपुट करंट होय.

### 3. फॅन -इन अँड फॅन -आऊट (Fan-in and Fan-out)

#### A. फॅन -इन

फॅन इन हे गेटच्या असलेल्या इनपुटची संख्या म्हणून परिभाषित केले आहे. उदाहरणार्थ, दोन इनपुट गेटमध्ये फॅन-इन 2 च्या समान असेल.

#### B. फॅन –आऊट

एका लॉजिक गेट मुळे दुसरे सारखे किती गेट त्यावर कार्य करतात, त्या संख्येला फॅन आऊट म्हणतात.

TTL साठी फॅन-इन व फॅन-आऊट ची संख्या 10 असते.

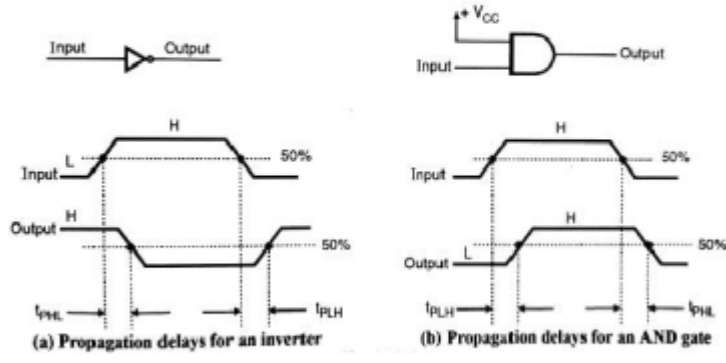
### 4. अपव्यय शक्ती (Power Dissipation)

डिजिटल आय सी मध्ये खर्च झालेल्या एकूण शक्तीला पॉवर डिसिपेशन असे म्हणतात. हा फॅक्टर कमी असणे आवश्यक असते अन्यथा सर्किटमध्ये जास्तीची उष्णता निर्माण होते. TTL गेटमध्ये 10mw शक्ती खर्च होते.

### 5. प्रसार विलंब (Propagation Delay)

इनपुटच्या स्थितीतील बदलाच्या प्रतिसादात तर्कशास्त्राचे आउटपुट त्याची स्थिती त्वरित बदलत नाही. त्या दोन घटनांमध्ये एक वेळ विलंब(Time Delay) आहे, ज्याला प्रसार विलंब (प्रोपॅगेशन डीले) म्हणतात.





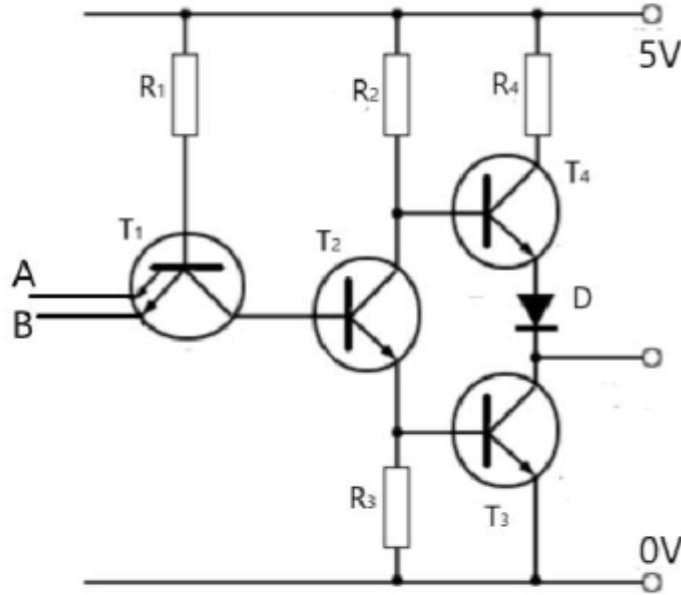
आकृती 2.34- प्रोपॅगेशन डीले

## 6. फिगर ऑफ मेरिट (Figure of Merit)

प्रोपॅगेशन डीले आणि पॉवर डिसिपेशन यांच्या गुणाकारास फिगर ऑफ मेरिट असे म्हणतात.

फिगर ऑफ मेरिट = प्रोपॅगेशन डीले( $t_p$ ) X पॉवर डिसिपेशन( $P_D$ )

## 2.8 TTL NAND गेट



आकृती 2.35- टोटम-पोल आउटपुटसह मूलभूत 2-इनपुट TTL NAND गेट.

आकृती 2.35 TTL NAND गेटचे सर्किट दर्शवित आहे.

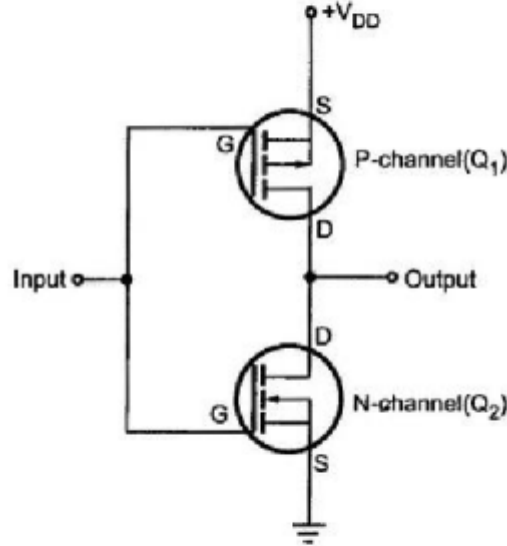
ट्रान्झिस्टर T1 ट्रान्झिस्टर 2 मध्ये दोन इनपुट देण्यासाठी दोन एमीटर आहेत. जोडल्याप्रमाणे बेस व्होल्टेज 5V वर असेल. दोन्ही इनपुट लॉजिक 1 असल्यास (सामान्यतः याचा अर्थ सुमारे 5V देखील असतो), बेस आणि एमिटरमधील संभाव्य फरक शून्य किंवा जवळपास असेल. त्यामुळे, कोणताही विद्युतप्रवाह वाहणार नाही आणि ट्रान्झिस्टर बंद आहे. तर, कलेक्टर व्होल्टेज देखील सुमारे 5V च्या समान असेल. म्हणून, ट्रान्झिस्टर T2 च्या एमीटरद्वारे विद्युत प्रवाह चालवू शकते. हे नंतर ट्रान्झिस्टर T2 च्या कलेक्टर व्होल्टेजला अनुमती देईल. एमिटरमधून वाहणाऱ्या विद्युतप्रवाहामुळे, रेझिस्टर R3 वर व्होल्टेज ड्रॉप होईल. इच्छित व्होल्टेज ड्रॉप सुमारे 0.7V असेल. पाहिल्याप्रमाणे, हे ट्रान्झिस्टर T3 चे इनपुट आहे. म्हणून, ट्रान्झिस्टर चालू आहे. सॅचुरेशन मुळे, कलेक्टर व्होल्टेज सुमारे 0.2V पर्यंत खाली येईल जे तर्क 0 आहे. ट्रान्झिस्टर T4 साठी, ट्रान्झिस्टर T3 च्या संपूर्ण व्होल्टेजने एमिटर व्होल्टेज बनलेले असल्याचे निरीक्षण करा अधिक डायोड डी वर व्होल्टेज ड्रॉप सुमारे 0.7V असेल. त्यामुळे उत्सर्जक संभाव्यता  $0.7+0.2=0.9V$  असेल. आता ट्रान्झिस्टर T4 चे बेस व्होल्टेज, हे T3 च्या बेस-एमिटरमधील व्होल्टेज असेल आणि संपूर्ण ट्रान्झिस्टरचे व्होल्टेज (म्हणजे) एमिटर-कलेक्टरमधील व्होल्टेज. हे देखील सुमारे 0.9V च्या समान असेल. म्हणून एमिटर व्होल्टेज आणि कलेक्टर व्होल्टेज समान आहेत. तर ट्रान्झिस्टर T4 देखील बंद केले जाईल. दोन्ही इनपुट 1 असताना आउटपुट शून्य आहे.

#### तक्ता 2.10- NAND गेटची सत्य सारणी

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

## 2.9 CMOS इन्व्हर्टर (Inverter)

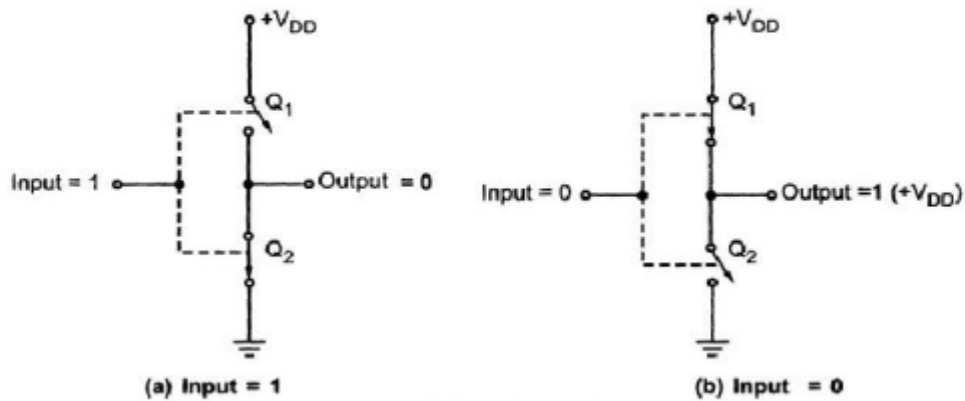
कॅपेसिटिव्ह लोड्सच्या स्विचिंगला गती देण्यासाठी CMOS सर्किटमध्ये NMOS आणि PMOS दोन्ही वापरले जातात. हे कमी उर्जा वापरते आणि उच्च विद्युतदाबावर (High Voltage) कार्य करू शकते. खालील आकृतीमध्ये CMOS इन्व्हर्टरच्या कार्याचे मूलभूत तत्त्व दाखवले आहे.



आकृती 2.36- CMOS इन्व्हर्टर

यामध्ये दोन MOSFETs अशा प्रकारे असतात की P-चॅनेल डिव्हाइसचा सोर्स (Source)  $+V_{DD}$  शी जोडलेला असतो आणि N-चॅनेल डिव्हाइसचा सोर्स ग्राउंडशी जोडलेला असतो. दोन उपकरणांचे गेट्स (Gates) सामान्य इनपुट म्हणून एकत्र जोडलेले आहेत आणि ड्रेन्स (Drain) सामान्य आउटपुट म्हणून एकत्र जोडलेले आहेत.

जेव्हा इनपुट जास्त असते, तेव्हा  $Q_1$  (P-चॅनेल) चे गेट  $Q_1$  च्या सोर्सच्या सापेक्ष  $0V$  वर असते म्हणजेच  $V_{GS1} = 0V$ . अशा प्रकारे,  $Q_1$  बंद आहे. दुसरीकडे,  $Q_2$  (N-चॅनेल) चे गेट त्याच्या सोर्सच्या सापेक्ष  $+V_{DD}$  वर आहे म्हणजे  $V_{GS2} = +V_{DD}$ . अशा प्रकारे,  $Q_2$  चालू आहे. हे आकृती 2.37 मध्ये दाखवल्याप्रमाणे  $V_{out} \approx 0V$  तयार करेल.



आकृती 2.37- CMOS इन्व्हर्टरचे कार्य

इनपुट कमी असताना, Q1 (P-चॅनेल) चे गेट त्याच्या सोर्सच्या सापेक्ष नकारात्मक विभवांतरावर असते तर Q2 मध्ये  $V_{GS} = 0\text{ V}$  असते. अशा प्रकारे, Q1 चालू असतो आणि Q2 बंद असतो. आकृती 2.37 मध्ये दर्शविल्याप्रमाणे हे अंदाजे +VDD आउटपुट व्होल्टेज तयार करते.

**तक्ता 2.11- इन्व्हर्टरची सत्यसारणी**

A	Q1	Q2	Output
0	ON	OFF	1
1	OFF	ON	0

### 2.9.1 TTL च्या तुलनेत CMOS चे फायदे-

1. खर्चिक शक्ती (power dissipation) कमी असते
2. फॅन आऊट 50 पेक्षा जास्त असते.
3. CMOS आई सी (IC) 3 ते 18V पर्यंत कार्य करू शकते.
4. तापमान स्थिरता जास्त असते.

### 2.9.2 TTL च्या तुलनेत CMOS चे तोटे

1. प्रसारित प्रवलंबन (Propagation Delay) TTL पेक्षा जास्त असते.
2. CMOS आई सी ची हाताळणी काळजीपूर्वक करावी लागते.

### 2.9.3 TTL व CMOS यांची तुलना

**तक्ता 2.11- TTL व CMOS यांची तुलना**

TTL	CMOS
बायपोलार फॅमिलीचा घटक	युनिपोलार फॅमिलीचा घटक
प्रोपॅगेशन डीले 10 nsec असतो	प्रोपॅगेशन डीले 50 nsec असतो
फॅन आऊट 10 असतो	फॅन आऊट 50 असतो
पुरवठा विद्युतदाब 5V लागतो	पुरवठा विद्युतदाब 3V ते 18V लागतो
पॉवर डिसिपेशन 10mW असते	पॉवर डिसिपेशन 10nW असते

## स्व: अध्ययन (Self learning)

1. लॉजिक गेटची व्याख्या लिहून त्याचे प्रकार लिहा.
2. धन आणि ऋण तर्क स्पष्ट करा.
3. OR गेटचे चिन्ह काढून त्याची सत्य सारणी काढा.
4. OR गेटचे स्विच सर्किट काढून स्पष्टीकरण लिहा.
5. AND गेटचे चिन्ह काढून त्याची सत्य सारणी काढा.
6. NOT गेट थोडक्यात स्पष्ट करा.
7. NAND गेटचे चिन्ह काढून त्याची सत्य सारणी काढा.
8. NAND गेटला युनिवर्सल गेट का म्हणतात ते उदाहरणासह स्पष्ट करा.
9. NOR गेटचे चिन्ह काढून माहिती द्या.
10. NOR गेट वापरून AND OR व NOT गेट तयार करा.
11. डिमॉर्गनचे दोन्ही नियम स्पष्ट करा.
12. TTL NAND गेटचे सर्किट काढून त्याचे कार्य स्पष्ट करा.
13. व्याख्या लिहा – A. फॅन इन B. फॅन आऊट C. प्रोपॅगेशन डीले
14. TTL व CMOS यांची तुलना करा.

## लघु प्रकल्प (Micro Project)

डिजिटल आयसी टेस्टर सर्किट तयार करा व तपासा.

## युनिट ३

**युनिट चे नाव -** संयुक्त तार्किक मंडल (Combinational Logic Circuits)

**विषय निष्पत्ती (Course Outcomes)-** सोपे संयुक्तिक तर्कशास्त्र मंडळ तयार करणे

**युनिट निष्पत्ती (Unit Outcomes)-**

3a) दिलेल्या लॉजिकल एक्सप्रेशनवरून गुणात्मक पदचे बेरजीय रूप (SOP) / बेरजीय राशीचे गुणात्मक रूप (POS) मध्ये लॉजिक सर्किट्स विकसित करणे

3b) के मॅप वापरून दिलेली तार्किक एक्सप्रेशन लहान करणे

3c) दिलेल्या बेरीज (Adder) आणि वजाबाकीची (Subtractor) रचना करण्यासाठी IC7483 वापरणे

3d) दिलेल्या इनपुट आउटपुटसाठी मक्स डीमक्स ट्री (MUX DeMUX Tree) काढणे

3e) दिलेल्या उपयोगीतेकरिता कॉम्प्योनंट चे स्पेसिफिकेशन्स लिहिणे

3f) दिलेल्या प्रकारचे कोड कन्वर्टर विकसित करणे

**परिचय -**

डिजिटल सर्किट्स दोन मुख्य प्रकार आहेत- (१) एकत्रित किंवा संयुक्त (Combinational) मंडल. (२) एक्स्स (Sequential) मंडल. संयुक्त मंडलमध्ये मिळणारे आउटपुट हे त्यावेळी दिलेल्या इनपुट वर फक्त अवलंबून असते. अश्या सर्किट्स ना स्मरण शक्ती (मेमरी) नसते. परंतु एक्स्स मंडलामध्ये मिळणारे आउटपुट हे त्यावेळच्या इनपुट वर तसेच पूर्वीच्या इनपुट किंवा आउटपुट वर देखील अवलंबून असते. अशा सर्किट्स ना पूर्वीची माहिती साठवण्यासाठी मेमरी असते. अश्या सर्किट्स चा संयुक्त सर्किट हा एक उपघटक असू शकतो.

संयुक्त लॉजिक सर्किट्स तयार करण्यासाठी मलिटप्लेक्सर्स व डेमलिटप्लेक्सर्स वापरतात. मलिटप्लेक्सर्स व डेमलिटप्लेक्सर्स तसेच एनकोडर्स व डीकोडर्ससाठी आयसीचा वापर केल्यास सिस्टिम ची विश्वासार्हता (reliability) वाढते व किंमत कमी होते

### 3.1 स्टॅंडर्ड बुलियन प्रतिनिधित्व (Standard Boolean Representation)-

**LO! शिकण्याचे उद्दिष्ट!:** दिलेल्या लॉजिकल एक्सप्रेशनसाठी स्टॅंडर्ड एसओपी (SOP) फॉर्ममध्ये लॉजिक सर्किट्स विकसित/डिझाइन करणे.

एक स्टॅंडर्ड गुणात्मक पद (standard product) किंवा "मीन- टर्म" (min term) हे फंक्शनसाठी सर्व स्वतंत्र इनपुट व्हेरिएबल्सचे गुणात्मक पद आहे जे 1 च्या आउटपुटसह सत्य सारणीच्या पंक्तीशी (ट्रुथ टेबल) संबंधित आहे. सामान्यतः, मिनिमायझेशन स्टेपच्या आधी फंक्शन दोन स्टॅंडर्ड फॉर्मपैकी एकांमध्ये लिहावे लागते.

दोन स्टॅंडर्ड प्रकार आहेत -

१. गुणात्मक राशीचे बेरजीय रूप (Sum of Products SOP)

२. बेरजीय राशीचे गुणात्मक रूप (Product of Sum POS)

सम ऑफ प्रॉडक्ट दिलेल्या फंक्शन साठी प्राप्त करणे-

**SOP फॉर्म:** गुणात्मक पद फॉर्मची बेरीज हा बुलियन बीजगणितातील एक्सप्रेशनचा एक प्रकार आहे ज्यामध्ये इनपुटच्या भिन्न गुणात्मक पद संज्ञा एकत्रित केल्या जातात.

"मीन- टर्म" (Minterm): एक "मीन- टर्म" एक बुलियन एक्सप्रेशन आहे परिणामी एका सेलच्या आउटपुटसाठी 1 आणि सत्य सारणीतील इतर सर्व सेलसाठी 0s.



उदाहरण -  $F(A,B) = A+B$  साठी

सिस्टम आकृती-



आकृती 3.1- सिस्टम आकृती

$F(A,B)$  चे सर्व संभाव्य मूल्य पाहण्यासाठी सत्य सारणी-

तक्ता क्रमांक 3.1- सत्य सारणी

Input		Output
A	B	$F(A,B)=A+B$
0	0	0
0	1	1
1	0	1
1	1	1

सत्य सारणीमध्ये इनपुट आणि आउटपुटचा संच असतो. जर 'n' इनपुट व्हेरिएबल्स असतील, तर शून्य आणि एकसह  $2^n$  संभाव्य संयोजन असतील. त्यामुळे प्रत्येक आउटपुट व्हेरिएबलचे मूल्य इनपुट व्हेरिएबल्सच्या संयोजनावर अवलंबून असते. तर, प्रत्येक आउटपुट व्हेरिएबलमध्ये इनपुट व्हेरिएबल्सच्या काही संयोजनासाठी '1' आणि इनपुट व्हेरिएबल्सच्या इतर संयोजनासाठी '0' असेल.

म्हणून, आपण प्रत्येक आउटपुट व्हेरिएबल खालील दोन प्रकारे व्यक्त करू शकतो.

- गुणात्मक राशींचे बेरजीय रूप (SOP form)
- बेरजीय राशींचे गुणात्मक रूप (POS form)

### 3.1.1 गुणात्मक राशींचे बेरजीय रूप (SOP form)-

कॅनोनिकल एसओपी फॉर्म(SOP) म्हणजे गुणात्मक ची कॅनोनिकल बेरीज फॉर्म. या फॉर्ममध्ये, प्रत्येक गुणात्मक पदामध्ये सर्व शब्दशः समाविष्ट आहेत. तर, या गुणात्मक पद पद किमान पद(min terms) आहेत. म्हणून, कॅनोनिकल एसओपी फॉर्मला किमान पद फॉर्मची बेरीज असेही म्हणतात.

प्रथम, ज्यासाठी आउटपुट व्हेरिएबल 1 आहे ते किमान पद ओळखा आणि नंतर त्या आउटपुट व्हेरिएबलशी संबंधित बुलियन एक्सप्रेशन फंक्शन मिळविण्यासाठी त्या मिन टर्मचे लॉजिकल OR करा. हे बुलियन फंक्शन सम (sum) ऑफ मिन टर्मच्या स्वरूपात असेल.

एकापेक्षा जास्त आउटपुट व्हेरिएबल असल्यास इतर आउटपुट व्हेरिएबल्ससाठी देखील हीच पद्धत वापरतात.

उदाहरण: तक्ता क्रमांक 3.2 सत्य सारणीचा विचारात घ्या-

तक्ता क्रमांक 3.2 - सत्य सारणी

Input			Output
A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

येथे, इनपुटच्या चार संयोजनांसाठी आउटपुट F '1' आहे. संबंधित किमान पद आहेत या चार किमान पद पैकी लॉजिकल OR करून, आउटपुट F चे बुलियन फंक्शन मिळेल.

म्हणून, आउटपुटचे बुलियन फंक्शन आहे,  $F = \bar{A}BC + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$ . हे आउटपुट " F " चे SOP फॉर्म आहे.

हे फंक्शन खालील दोन नोटेशनमध्ये देखील दर्शवू शकतो.

$$F = m_3 + m_5 + m_6 + m_7$$

$$F = \sum m(3, 5, 6, 7)$$

उदाहरण-

खालील लॉजिक एक्सप्रेशनला कॅनॉनिकल SOP फॉर्ममध्ये रूपांतरित करा:

$$Y = AB + A\bar{C} + BC$$

$$\begin{aligned} Y &= AB(C + \bar{C}) + A\bar{C}(B + \bar{B}) + BC(A + \bar{A}) \\ &= ABC + AB\bar{C} + A\bar{B}\bar{C} + \bar{A}B\bar{C} + ABC + \bar{A}BC \\ &= ABC + AB\bar{C} + A\bar{B}\bar{C} + \bar{A}BC \end{aligned}$$

LO2 शिकण्याचे उद्दिष्ट 2: दिलेल्या तार्किक एक्सप्रेशनसाठी स्टॅंडर्ड POS फॉर्ममध्ये लॉजिक सर्किट्स विकसित/डिझाइन करणे.

**POS फॉर्म:** बेरजीय राशींचे गुणात्मक रूप हे बुलियन बीजगणितातील एक्सप्रेशनचे एक प्रकार आहे ज्यामध्ये इनपुटच्या भिन्न बेरीज संज्ञा एकत्रितपणे गुणात्मक रूपात तयार केल्या जातात.

“मॅक्स”टर्म(max term): मॅक्सटर्म ही एक बुलियन एक्सप्रेशन आहे ज्यामुळे एका सेलच्या आउटपुटसाठी "0" आणि सत्य सारणीमधील इतर सर्व सेलसाठी "1"s येतो.

लिटर्ल्स(literals): अप्रूक्(uncomplimented) किंवा पूरक(complimented) स्वरूपातील व्हेरिएबलला लिटर्ल्स म्हणतात.

स्पष्टीकरण-

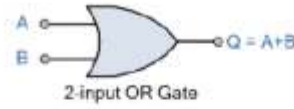
बेरीज एक्सप्रेशन गुणात्मक पद म्हणजे एक किंवा अधिक व्हेरिएबल्सच्या बेरजेने बनलेले बूलियन एक्सप्रेशन आहेत, एकतर त्याच्या सामान्य सत्य स्वरूपात किंवा पूरक स्वरूपात किंवा दोन्हीचे संयोजन, जे नंतर आणि एकत्र केले जातात.

जर अनेक व्हेरिएबल (variable) चे बूलियन फंक्शन गुणात्मक राशी (POS) मध्ये व्यक्त केले असेल, तर प्रत्येक पदाला कमाल पद (max term) म्हणतात, हे व्हेरिएबल लॉजिक "0" म्हणून घेतले जाते.

OR फंक्शनला बेरीज संज्ञा म्हणून संबोधले जाते. OR फंक्शन हे बेरीज चे गणितीय समतुल्य आहे जे अधिक(+) चिन्हाने दर्शविले जाते. अशा प्रकारे

2-इनपुट किंवा गेटमध्ये A+B च्या बूलियन एक्सप्रेशनद्वारे दर्शविलेले आउटपुट टर्म असते कारण ते A आणि B ची तार्किक बेरीज असते.

OR गेट (बेरीज)-



आकृती. 3.2- OR गेट

ही तार्किक बेरीज सामान्यतः बूलियन ऑडिशन म्हणून ओळखली जाते कारण OR फंक्शन दोन किंवा अधिक इनपुट व्हेरिएबल्स किंवा स्थिरांकांची (constants) बेरीज टर्म तयार करते. अशा प्रकारे 2-इनपुट किंवा गेटसाठी बूलियन समीकरण असे दिले आहे:

$Q = A+B$ , म्हणजे  $Q = A \text{ OR } B$  दोन्ही समान आहे.

बेरीज टर्मसाठी (Sum terms) हे इनपुट व्हेरिएबल्स एकतर "सत्य" किंवा "असत्य", "1" किंवा "0" असू शकतात किंवा पूरक (compliment) स्वरूपाचे असू शकतात. या बेरीज पद maxterm म्हणून ओळखल्या जातात.

उदाहरण-

खालील एक्सप्रेशन POS मधून स्टॅंडर्ड (Standard POS) फॉर्ममध्ये रूपांतरित करा

$$F = (\bar{A}+B)(\bar{A}+C)$$

$$F = (\bar{A}+B+C\bar{C})(\bar{A}+C+B\bar{B})$$

$$F = (\bar{A}+B+C)(\bar{A}+B+\bar{C})(\bar{A}+B+C)(\bar{A}+\bar{B}+C)$$

$$F = (\bar{A}+B+C)(\bar{A}+B+\bar{C})(\bar{A}+\bar{B}+C)$$

### 3.1.2 पीओएस (POS) मधून स्टॅंडर्ड पीओएस (standard POS) फॉर्म / कॅनॉनिकल फॉर्ममध्ये रूपांतरण:

POS फॉर्म स्टॅंडर्ड POS फॉर्ममध्ये रूपांतरित केला जाऊ शकतो एक्सप्रेशन मध्ये असलेल्या टर्म आणि नसलेल्या वेरियेबल व त्याचा कॉम्प्लिमेंट च्या AND केलेल्या टर्म यांचे OR करून.

POS ला स्टॅंडर्ड POS फॉर्ममध्ये रूपांतरित करण्यासाठी स्टेप्स:

स्टेप 1: प्रत्येक बेरीज टर्ममध्ये गहाळ असलेले लिटरल शोधा.

Step 1: Find the missing literal in each sum term if any.

स्टेप 2: एक्स्प्रेसशन मध्ये असलेल्या टर्म आणि नसलेल्या वेरियेबल व त्याचा कॉम्प्लिमेंट च्या AND केलेल्या टर्म यांचे OR करा

Step 2: OR each sum term having missing literals with terms form by ANDing the literal and its complement.

स्टेप 3: विस्तृत नियम लागू करून पदाचा विस्तार करा आणि शब्दशः पुनर्क्रमित करा

Step 3: Expand the term by applying distributive law and reorder the literal

स्टेप 4: पुनरावृत्ती केलेल्या गुणात्मक पद पद कमी करा. कारण  $A + A = A$  (बूलियन कायदा).

Step 4: Reduce the repeated product terms. Because  $A + A = A$  (Boolean law).

**Example:**

$$f(A,B,C) = (A + B) \cdot (B + C) \cdot (A + C)$$

**Step 1: Find the missing literals in each sum term.**

$$f(A,B,C) = (A + B) \cdot (B + C) \cdot (A + C)$$

Literal B is missing  
 Literal A is missing  
 Literal C is missing

**Step 2: OR the sum term with missing literal . its complement.**

$$f(A,B,C) = (A + B) + (C \cdot \bar{C}) + (B + C) + (A \cdot \bar{A}) + (A + C) + (B \cdot \bar{B})$$

Missing literals and their complements

**Step 3: Expands the term and reorder the literals.**

$$f(A,B,C) = (A + B) + (C \cdot \bar{C}) + (B + C) + (A \cdot \bar{A}) + (A + C) + (B \cdot \bar{B})$$

Expand & Reorder:

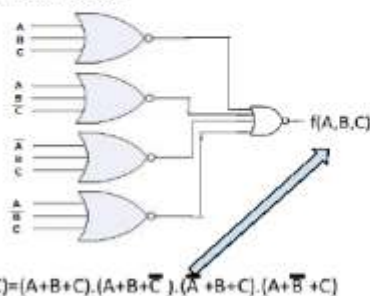
$$f(A,B,C) = (A+B+C) \cdot (A+B+\bar{C}) \cdot (A+B+C) \cdot (\bar{A}+B+C) \cdot (A+B+C) \cdot (A+\bar{B}+C)$$

**Step 4: Omit repeated sum terms.**

$$f(A,B,C) = (A+B+C) \cdot (A+B+\bar{C}) \cdot (\bar{A}+B+C) \cdot (A+\bar{B}+C)$$

NOR गेट वापरून प्राली-

Realization using NOR gate



आकृती 3.3- NOR गेट वापरून प्राली

**Maxterms:**

1.  $f(A,B,C) = (A+B+C).(A+\overline{B}+C).(A+\overline{B}+\overline{C}).(\overline{A}+\overline{B}+\overline{C})$   
 $= M_0 + M_2 + M_3 + M_7$   
 $= \Pi M(0,2,3,7)$
2.  $f(A,B,C) = (A+B+\overline{C}).(A+\overline{B}+\overline{C}).(\overline{A}+B+\overline{C}).(\overline{A}+\overline{B}+\overline{C})$   
 $= M_1 + M_3 + M_5 + M_7$   
 $= \Pi M(1,3,5,7)$
3.  $f(A,B,C) = (A+B+C).(A+\overline{B}+C).(A+\overline{B}+\overline{C}).(\overline{A}+\overline{B}+C)$   
 $= M_0 + M_2 + M_3 + M_6$   
 $= \Pi M(0,2,3,6)$

स्टॅंडर्ड SOP मधून स्टॅंडर्ड POS फॉर्ममध्ये रूपांतर:

- दिलेल्या स्टॅंडर्ड SOP एक्सप्रेशनमधील गुणात्मक पद संज्ञांची बायनरी मूल्ये समतुल्य स्टॅंडर्ड POS एक्सप्रेशनमध्ये उपस्थित नाहीत.
- बायनरी मूल्ये जी SOP एक्सप्रेशनमध्ये दर्शविली जात नाहीत ती समतुल्य POS एक्सप्रेशनमध्ये उपस्थित आहेत.

**Step 1:** SOP एक्सप्रेशनमधील प्रत्येक गुणात्मक पद पदाचे मूल्यमापन करा, म्हणजेच, गुणात्मक पद पद दर्शविणाऱ्या बायनरी संख्या निश्चित करा.

**Step 2:** Step 1 मधील मूल्यमापनात समाविष्ट नसलेल्या सर्व बायनरी संख्या निश्चित करा.

**Step 3:** स्टेप 2 वरून प्रत्येक बायनरी नंबरसाठी समतुल्य बेरीज टर्म लिहा आणि POS फॉर्ममध्ये व्यक्त करा.

उदाहरण-

**SOP form-**  $F = \sum A, B, C (0, 2, 3, 5, 7) = A' B' C' + A B' C' + A B' C + ABC' + ABC$

$$001 = (A + B + C) \quad 100 = (A + B' + C') \quad 110 = (A + B' + C')$$

**POS form-**  $F = \prod A, B, C (1, 4, 6) = (A + B + C) * (A + B' + C') * (A + B' + C')$

त्याचप्रमाणे स्टॅंडर्ड POSचे स्टॅंडर्ड SOPमध्ये रूपांतर करणे देखील शक्य आहे.

उदाहरण-

**POS form-**  $F = \prod A, B, C (2, 3, 5) = A B' C' + A B' C + ABC'$

$$000 = A' * B' * C' \quad 001 = A' * B' * C \quad 100 = A * B' * C'$$

$$110 = A * B * C' \quad 111 = A * B * C$$

**SOP form-**  $F = \sum A, B, C (0, 1, 4, 6, 7) = (A' * B' * C') + (A' * B' * C) + (A * B' * C') + (A * B * C') + (A * B * C)$

### 3.2 के-मॅप(Kmap) वापरून एसओपी(SOP)पीओएस(POS) फॉर्ममध्ये दिलेली लॉजिक फंक्शन्स लहान करणे-

अनेक डिजिटल सर्किट्समध्ये आणि व्यावहारिक समस्यांमध्ये कमीतकमी व्हेरिएबल्ससह लहान एक्सप्रेशन शोधणे आवश्यक आहे. हे कोणत्याही बुलियन बीजगणित प्रमेयांचा वापर न करता के-मॅप वापरून 3, 4 व्हेरिएबल्सचे बुलियन एक्सप्रेशन अगदी सहजपणे लहान करू शकते. के-मॅपमध्ये समस्यांच्या गरजेनुसार SOP आणि POS असे दोन प्रकार असू शकतात.

के-मॅप (K-map) हा टेबलासारखा आहे पण तो सत्य सारणी पेक्षा जास्त माहिती देतो. हे के-मॅपचे ग्रिड 0 ने भरलेले आहे

के-मॅप वापरून एक्सप्रेशन सोडवण्याच्या स्टेप्स -

1. व्हेरिएबल्सच्या संख्येनुसार के-मॅप निवडा.
2. समस्यांमध्ये दिल्याप्रमाणे किमान पद (minterms) किंवा कमाल पद (maxterms) ओळखा.
3. SOP साठी 1's के-मॅप च्या सेलमध्ये किमान पद (minterms) (अन्यत्र 0's) मध्ये ठेवा.
4. POS साठी के-मॅप च्या सेलमध्ये कमाल पद (maxterms) (अन्यत्र 1) च्या सेलमध्ये 0 टाका.
5. 2, 4, 8.. (1 वगळता) सारख्या दोनच्या बळावर एकूण संज्ञा असलेले आयताकृती गट बनवा आणि एका गटात जास्तीत जास्त घटक समाविष्ट करण्याचा प्रयत्न करा.
6. स्टेप 5 मध्ये तयार केलेल्या गटांमधून गुणात्मक पद शोधा आणि SOP फॉर्मसाठी त्यांची बेरीज करा.

के-मॅप (K-map) पद्धत वापरून खालील फंक्शन कमी करा.

$$1) F(A,B,C) = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + \bar{A}B + AC$$

AB \ C	00	01	11	10
0	1	1	1	
1		1	1	1

लहान केलेले फंक्शन आहे:

$$F(A,B,C) = B + AC + \bar{A}\bar{C}$$

$$F(A,B,C) = \bar{A}B + B\bar{C} + BC + A\bar{B}\bar{C}$$

AB \ C	00	01	11	10
0		1	1	1
1		1	1	

लहान केलेले फंक्शन आहे:

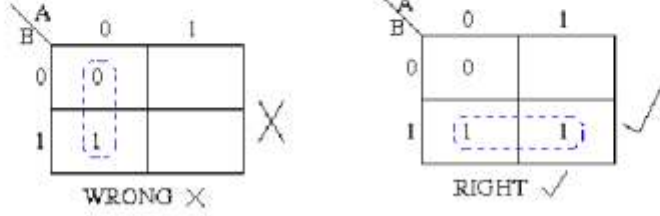
$$F(A,B,C) = B + A\bar{C}$$



### 3.2.1 के-मॅप (K-map) - सरलीकरणाचे नियम:

के-मॅप (K-map) समीप असलेल्या पेशींचा एकत्रित समूह करून एक्स्प्रेसशनच्या सरलीकरणासाठी खालील नियम वापरतो-

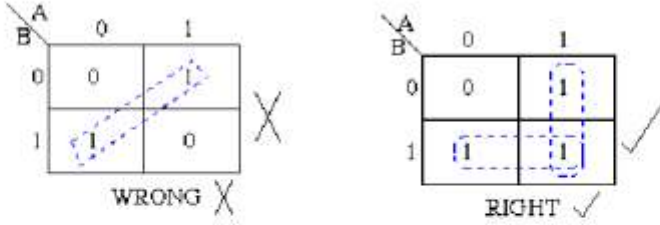
1) गटांमध्ये शून्य असलेल्या कोणत्याही सेलचा समावेश असू शकत नाही



चूक

बरोबर

2) गट क्षैतिज(horizontal) किंवा अनुलंब(vertical) असू शकतात, परंतु कर्णा( diagonal) नसतात.



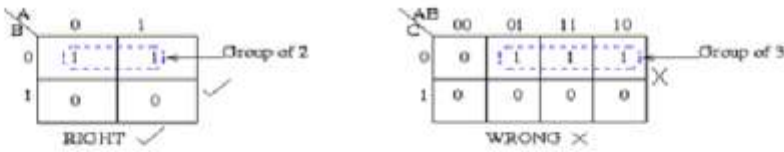
चूक

बरोबर

3) गटांमध्ये 1, 2, 4, 8 किंवा सर्वसाधारणपणे  $2^n$  सेल असणे आवश्यक आहे.

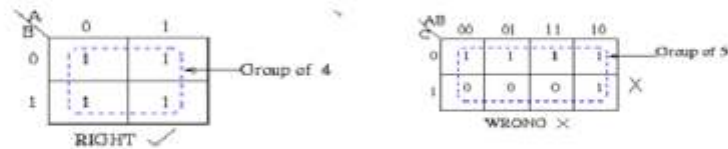
म्हणजेच  $n = 1$  असल्यास, एका गटात  $2^1 = 2$  पासून दोन 1 असतील.

जर  $n = 2$ , गटांमध्ये  $2^2 = 4$  पासून चार 1 असतील.



बरोबर

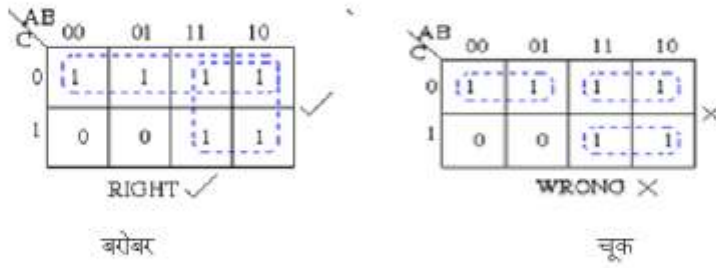
चूक



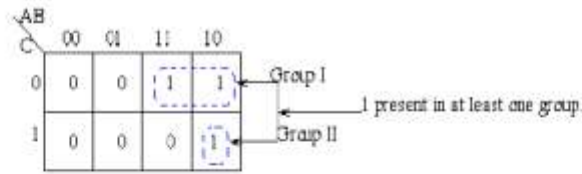
बरोबर

चूक

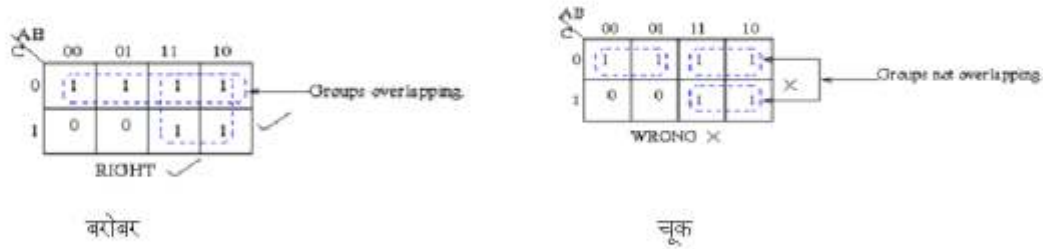
4) प्रत्येक गट शक्य तितका मोठा असावा.



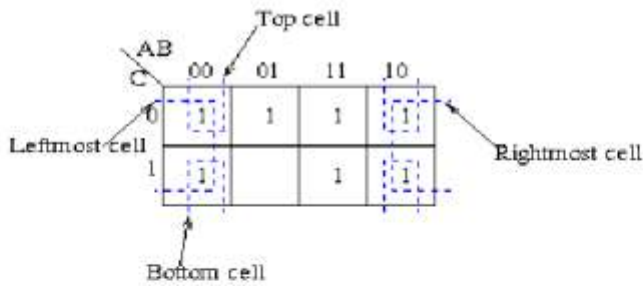
5) 1 असलेली प्रत्येक सेल किमान एका गटात असणे आवश्यक आहे.



6) गट ओव्हरलॅप होऊ शकतात.



7) ग्रुप फोल्ड होऊ शकतो. एका ओळीतील सर्वात डावीकडील सेल सर्वात उजव्या सेलसह गटबद्ध केला जाऊ शकतो आणि स्तंभातील शीर्ष सेल तळाशी असलेल्या सेलसह गटबद्ध केला जाऊ शकतो.



8) शक्य तितक्या कमी गट असावेत, जोपर्यंत हे मागील कोणत्याही नियमांचा विरोध करत नाही.

AB \ C	00	01	11	10
0	1	1	1	1
1	0	0	1	1

RIGHT ✓

बरोबर

AB \ C	00	01	11	10
0	1	1	1	1
1	0	0	1	1

WRONG ✗

चूक

### 3.3.2 SOP फॉर्मसाठी के-मॅप (K-map):-

1) 3 व्हेरिएबल्सचा के-मॅप (K-map)-

$$F(A, B, C) = \sum m(1, 3, 6, 7)$$

अंतिम एक्सप्रेशन

$$F(A, B, C) = AB + \bar{A}C$$

(ह्या गटाचा उपयोग नाही कारण ह्यातील 1 आपण आधीच्या गटात घेतले आहेत)

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

no need of this group as we've already covered those 1's

Groups of two elements in one group

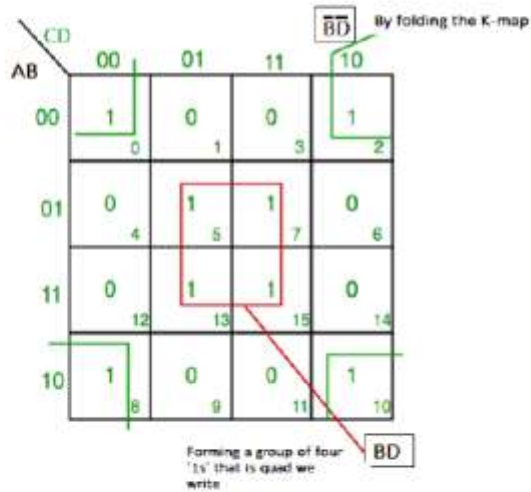
AB

(एका गटातील दोन घटकांचा समूह)

2) 4 व्हेरिएबल्ससाठी के-मॅप

$$F(A,B,C,D)=\sum m(0,2,5,7,8,10,13,15)$$

(के-मॅप फोल्ड करून)



(चार 1 चा एक गट तयार करणे)

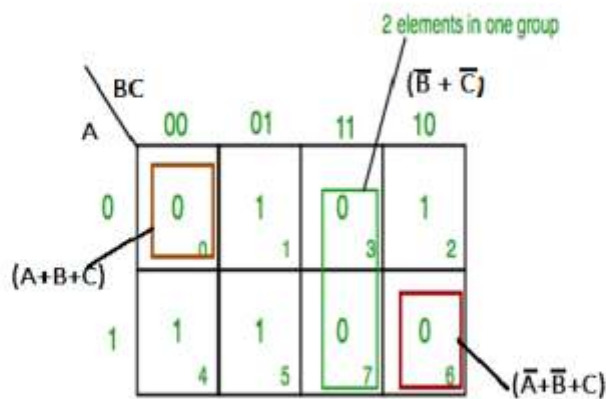
अंतिम एक्सप्रेशन

$$F(A, B, C, D) = BD + \bar{B}\bar{D}$$

POS फॉर्म के-मॅप (K-map)

3 व्हेरिएबल्सचा के-मॅप (K-map)-

(एका गटात 2 घटक)



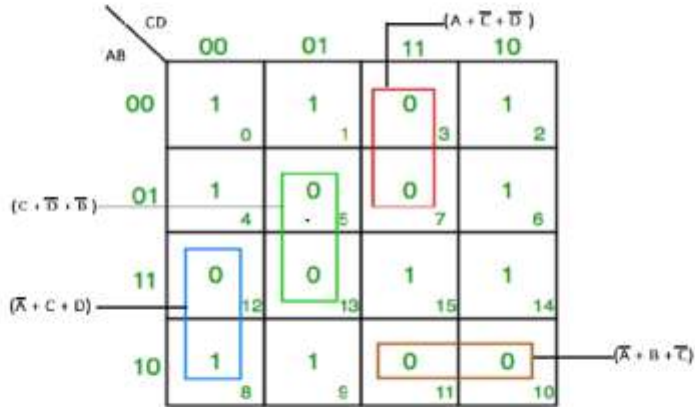
$$F(A, B, C) = \pi M (1, 3, 6, 7)$$

या तीन पदांचे गुणात्मक पद घ्या: अंतिम एक्सप्रेशन

$$F(A, B, C) = (\bar{A} + \bar{B} + C) (\bar{B} + \bar{C}) (A + B + C)$$

4 व्हेरिएबल्ससाठी के-मॅप

$$F(A, B, C, D) = \pi M(3, 5, 7, 8, 10, 11, 12, 13)$$



अंतिम एक्सप्रेशन-

$$F(A, B, C, D) = (A + \bar{C} + \bar{D}) (C + \bar{D} + \bar{B}) (\bar{A} + C + D) (\bar{A} + B + \bar{C})$$

### 3.3 के-मॅप वापरून गणितीय मंडल (arithmetic circuit) डिझाइन करणे-

LO3 शिकण्याचे उद्दिष्ट 3: के मॅप वापरून अरीथमेटिक सर्किट्स विकसित/डिझाइन करणे.

गणितीय मंडल हे कोणत्याही प्रोसेसरच्या ALU मध्ये वापरल्या जाणाऱ्या युनिटपैकी एक आहे. ही अंकगणितीय सर्किट एकापेक्षा जास्त बिट संख्यांच्या बेरीज आणि वजाबाकीसाठी वापरली जातात.

#### 3.3.1 हाफ अॅडर -

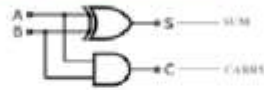
हाफ अॅडर हे एक लॉजिक सर्किट असून त्यामध्ये एका वेळी २ बायनरी डिजिट्स (bits) ची बेरीज केली जाते. हाफ अॅडर साठी एक एक्स ऑर आणि एक अँड गेट ची आवश्यकता असते, एक्स ऑर गेट च्या आउटपुट ला बेरीज (SUM) व अँड गेट च्या आउटपुट ला हातचा (CARRY) म्हणतात.

SUM आणि CARRY साठी बुलियन एक्सप्रेशन आहेत,

$$\text{SUM} = A \oplus B$$

$$\text{CARRY} = A \cdot B$$

#### HALF ADDER CIRCUIT

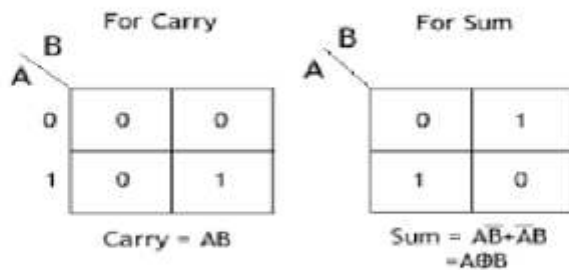


आकृती 3.4- हाफ अॅडर

तक्ता क्रमांक 3.3-सत्य सारणी

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

डिझाइन केलेले हाफ अॅडर सर्किट:-

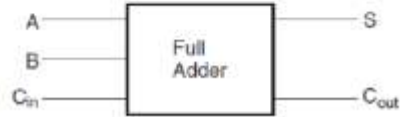




### 3.3.2 फुल अॅडर-

ज्या लॉजिक मंडलमध्ये एकावेळी ३ बायनरी डिजिट्स (bits) ची बेरीज केली जाते, त्याला 'फुल अॅडर' म्हणतात. फुल अॅडर ला हातचा (CARRY) व बेरीज (SUM) अशी दोन आउटपुट असतात

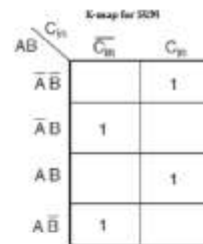
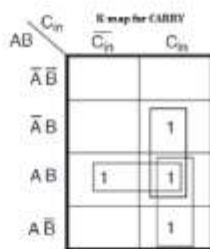
प्रकार १- यामध्ये फुल अॅडर तयार करण्यासाठी ३ अँड गेट्स, १ ऑर गेट व एक एक्स ऑर गेट वापरतात. ऑर गेट च्या आउटपुटला कॅरी व एक्स ऑर गेट च्या आउटपुटला बेरीज म्हणतात



आकृती 3.5- फुल अॅडर

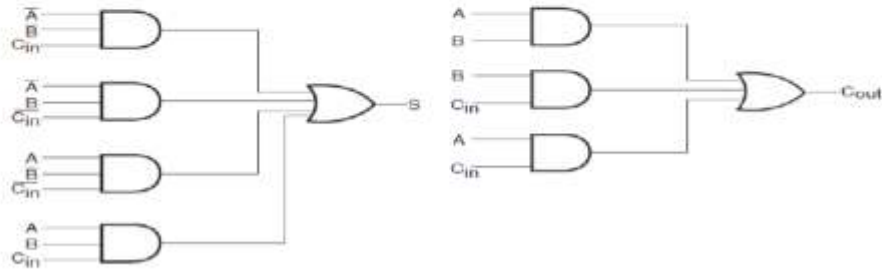
तक्ता क्रमांक 3.4-सत्य सारणी

A	B	C <sub>in</sub>	SUM(S)	C <sub>out</sub>
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

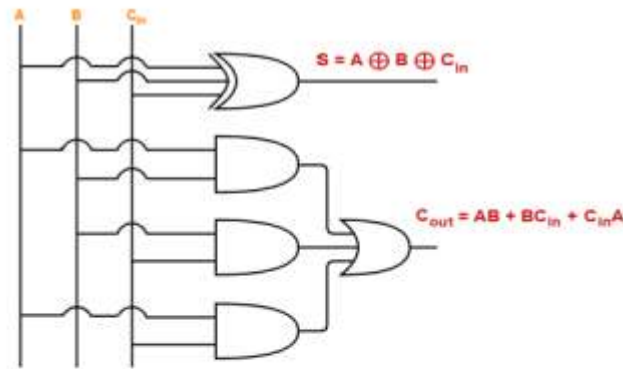


$$C_{out} = AB + AC_{in} + BC_{in}$$

आकृती 3.6(a) बेरीज आणि कॅरी दोन्हीसाठी फुल अॅडर सर्किटची सरलीकृत अंमलबजावणी दर्शवते

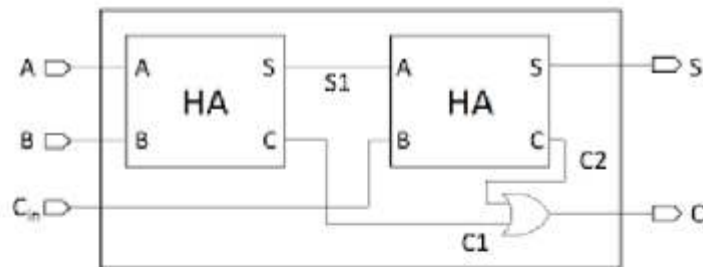


आकृती.3.6(a)- फुल अॅडर



आकृती.3.6(b)- फुल अॅडर

प्रकार २- दोन हाफ अॅडर्स वापरून फुल अॅडर:



आकृती.3.7- दोन हाफ अॅडर्स वापरून फुल अॅडर

### 3.3.3 अर्थ वजाबाकी करणारे मंडळ (हाफ सबट्रॅक्टर)-

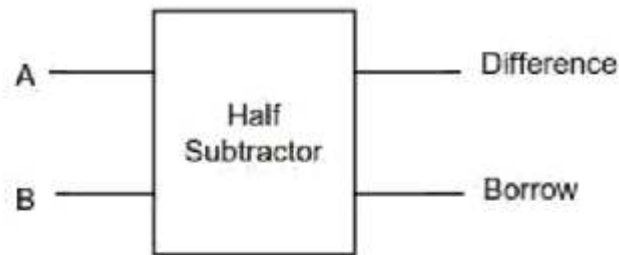
वजाबाकी दशांश संख्या प्रणालीप्रमाणेच स्तंभानुसार केली जाते. पहिल्या स्टेप मध्ये, LSBs वजा करतो आणि नंतर MSB च्या दिशेने पुढे जातो. जेथे सबट्राहेंड (वजाबाकीचा बिट) मिन्युएंडपेक्षा मोठा असेल, '1' असलेल्या पुढील समीप उच्च बिट स्थानावरून (higher significant digit) उधार(borrow) घेतो.

खाली उदाहरण दिले आहे, या उदाहरणात (1100) मधून (1001) वजा करण्याच्या वेगवेगळ्या स्टेप्स आहेत. येथे, दुसऱ्या MSB पोजिशनमधून 1 बॉरो घेतला आहे, त्या स्थितीत 0 सोडला आहे. बॉरो प्रथम तिसऱ्या MSB पोजिशनवर आणून ते 10 केले जाते. या पोजिशनमधील 10 पैकी 1 ला LSB पोजिशनवर 10 करण्यासाठी नेले जाते आणि तिसऱ्या MSB पोजिशनमध्ये 1 सोडला जातो.

बायनरी वजाबाकीच्या स्टेप्स खालील आकृतीमध्ये दिल्या आहेत.

<b>Step 1</b>	$\begin{array}{r} 1 \quad 1 \quad 0 \quad 0 \\ 1 \quad 0 \quad 0 \quad 1 \\ \hline \end{array}$	<b>Step 2</b>	$\begin{array}{r} 1 \quad 1 \quad 0 \quad 0 \\ 1 \quad 0 \quad 0 \quad 1 \\ \hline \end{array}$
<b>Step 3</b>	$\begin{array}{r} 1 \quad 1 \quad 0 \quad 0 \\ 1 \quad 0 \quad 0 \quad 1 \\ \hline \end{array}$	<b>Step 4</b>	$\begin{array}{r} 1 \quad 1 \quad 0 \quad 0 \\ 1 \quad 0 \quad 0 \quad 1 \\ \hline \end{array}$
	0 1 1		0 0 1 1

एक अर्ध वजाबाकी मुख्यतः एक डिफरन्स (DIFFERENCE) आऊटपुट आणि बॉरो (BORROW) आऊटपुट तयार करण्यासाठी एक बायनरी अंक दुसऱ्यामधून वजा करण्यासाठी वापरला जातो. वजाबाकी करण्यासाठी '1' घेतले गेले आहे की नाही हे येथे BORROW आऊटपुट निर्दिष्ट करते.



आकृती.3.8- हाफ सबट्रॅक्टर

तक्ता क्रमांक 3.5-सत्य सारणी

Input		Output	
A	B	DIFFERENCE	BORROW
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

For Difference

	B	0	1
A	0	0	1
	1	1	0

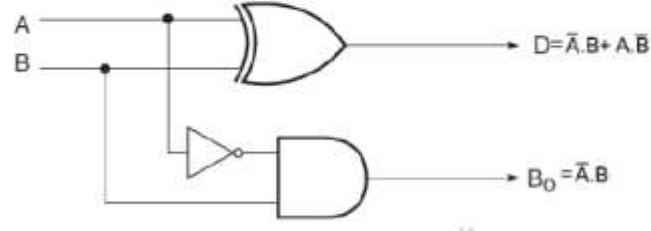
$$\text{Difference} = A\bar{B} + \bar{A}B \\ = A \oplus B$$

For Borrow

	B	0	1
A	0	0	1
	1	0	0

$$\text{Borrow} = \bar{A}B$$

डिझाइन केलेले हाफ सबट्रॅक्टर सर्किट:-



आकृती.3.9- हाफ सबट्रॅक्टर

### 3.3.4 पूर्ण वजाबाकी करणारे मंडळ (फुल सबट्रॅक्टर) :-

पूर्ण वजाबाकी दोन बिट्सची वजाबाकी करतो, एक मिन्युएंड (minuend) आणि दुसरा सबट्राहेन्ड (subtrahend). पूर्ण वजाबाकीमध्ये '1' मागील समीप असलेल्या लोअर मिन्युएंड बिटद्वारे उधार घेतला जातो. त्यामुळे पूर्ण वजाबाकीच्या इनपुटवर तीन बिट विचारात घेतले जातात. दोन आउटपुट आहेत, ते म्हणजे डिफरन्स आउटपुट डी आणि बोरो आउटपुट बो. BORROW आउटपुट सूचित करते की minuend bit ला पुढील minuend bit मधून '1' उधार घेणे आवश्यक आहे. आकृतीसंपूर्ण वजाबाकीचे सत्य सारणी दर्शवते. दोन आउटपुटसाठी के-नकाशे आकृतीमध्ये दर्शविले आहेत. जर आपण DIFFERENCE आउटपुट D आणि BORROW आउटपुट Bo ची तुलना फुल अॅड्सह केली तर असे दिसून येईल की DIFFERENCE आउटपुट D SUM आउटपुट प्रमाणेच आहे. पुढे, बोरो आउटपुट CARRY साखे आहे. अर्ध्या-वजाबाकीच्या बाबतीत, "A" इनपुट फूक आहे तशाच गोष्टी पूर्ण वजाबाकीमध्ये केल्या जातात. खालील आकृती 3.10 संपूर्ण वजाबाकीचे ब्लॉक डायग्राम दर्शवते. पूर्ण वजाबाकीचे इनपुट A, B आणि Bin आहेत. पूर्ण वजाबाकीचे आउटपुट D आणि Bo आहेत.



आकृती.3.10- फुल सबट्रॅक्टर

### तक्ता क्रमांक 3.7- सत्य सारणी

Minuend (A)	Subtrahend (B)	Borrow In (Bin)	Difference (D)	Borrow Out (Bo)
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

AB \ B <sub>in</sub>	$\overline{B}_{in}$	B <sub>in</sub>
$\overline{A}\overline{B}$		1
$\overline{A}B$	1	
AB		1
A $\overline{B}$	1	

$$D = \overline{A}.\overline{B}.B_{in} + \overline{A}.B.\overline{B}_{in} + A.\overline{B}.\overline{B}_{in} + A.B.B_{in}$$

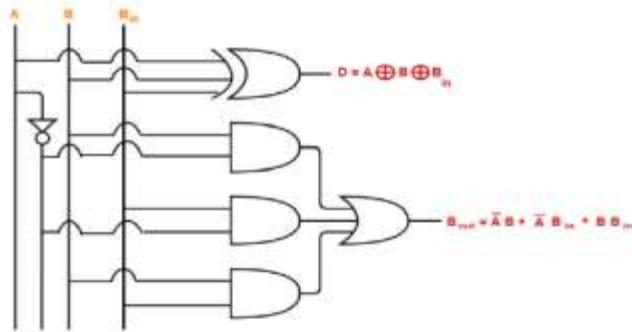
AB \ B <sub>in</sub>	$\overline{B}_{in}$	B <sub>in</sub>
$\overline{A}\overline{B}$		1
$\overline{A}B$	1	1
AB		1
A $\overline{B}$		

$$B_o = \overline{A}.\overline{B}.B_{in} + \overline{A}.B.\overline{B}_{in} + \overline{A}.B.B_{in} + A.B.B_{in}$$

$$\begin{aligned} D &= \overline{A}\overline{B}B_{in} + \overline{A}B\overline{B}_{in} + A\overline{B}\overline{B}_{in} + AB B_{in} \\ &= B_{in}(\overline{A}\overline{B} + AB) + \overline{B}_{in}(\overline{A}B + A\overline{B}) \\ &= B_{in}(A \odot B) + \overline{B}_{in}(A \oplus B) \\ &= B_{in} \oplus (A \oplus B) \end{aligned}$$

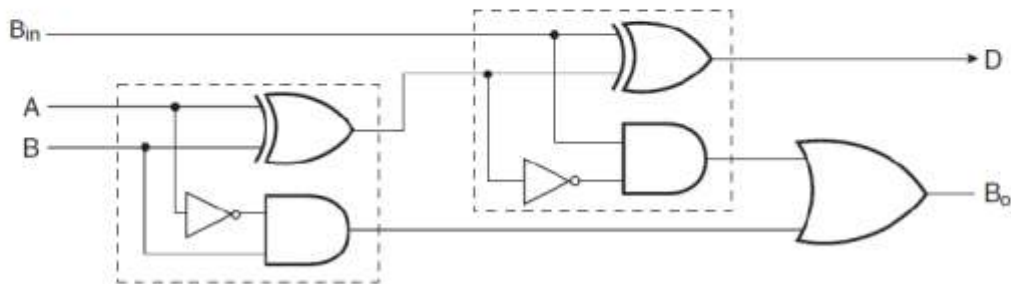
$$B_{out} = \overline{A}B + \overline{A}B_{in} + BB_{in}$$

डिज़ाइन केलेले फुल सबट्रैक्टर सर्किट:-



आकृती.3.11- फुल सबट्रैक्टर

दोन हाफ सबट्रैक्टर वापरून फुल सबट्रैक्टर:



आकृती 3.12- दोन हाफ सबट्रैक्टर वापरून फुल सबट्रैक्टर

### 3.3.5 के-मॅप वापरून कोड कन्व्हर्टर सर्किट डिझाइन करणे-

LO4 शिकण्याचे उद्दिष्ट 4: के-मॅप वापरून ग्रे टू बायनरी(Gray to Binary)/ बायनरी टू ग्रे(Binary to Gray) कोड कन्व्हर्टर सर्किट डिझाइन करणे

#### ग्रे कोड:

- ग्रे कोड सिस्टीम ही एक बायनरी संख्या प्रणाली आहे ज्यामध्ये संख्यांची प्रत्येक सलग जोडी फक्त एका बिटमध्ये भिन्न असते.
- हे ॲप्लिकेशनमध्ये वापरले जाते ज्यामध्ये हार्डवेअरद्वारे व्युत्पन्न केलेल्या बायनरी संख्यांचा सामान्य क्रम एका क्रमांकावरून दुसऱ्या क्रमांकावर संक्रमणादरम्यान त्रुटी किंवा अस्पष्टता निर्माण करू शकतो.

उदाहरणार्थ, प्रणालीची अवस्था 3(011) वरून 4(100) 011 — 001 — 101 — 100 मध्ये बदलू शकते.

- त्यामुळे प्रणाली सुरुवातीच्या स्थितीपासून अंतिम स्थितीत बदलत असताना चुकीची स्थिती वाचण्याची उच्च शक्यता असते.
- ग्रे कोड ही समस्या दूर करतो कारण दोन संख्यांमधील कोणत्याही संक्रमणादरम्यान फक्त एक बिट त्याचे मूल्य बदलतो.

#### बायनरी- ग्रे कोड कन्व्हर्टर:-

बायनरी ग्रे कोडमध्ये रूपांतरित करणे -

A, B, C, D हे बायनरी संख्या दर्शविणारे बिट असू द्या, जेथे D हा LSB आणि A हा MSB आहे आणि

G1 G2, G3, G4, बायनरी संख्यांच्या ग्रे कोडचे प्रतिनिधित्व करणारे बिट असू द्या, जेथे G4 हा LSB आहे आणि G1 हा MSB आहे.

1. MSB समान ठेवला आहे. बायनरीचा MSB 0 असल्याने, ग्रे कोडचा MSB देखील 0 असेल (प्रथम ग्रे बिट)
2. पुढे, पहिल्या आणि दुसऱ्या बायनरी बिटचा XOR घ्या. पहिला बिट 0 आहे आणि दुसरा बिट 1 आहे. बिट्स भिन्न आहेत त्यामुळे परिणामी ग्रे बिट 1 असेल (दुसरा ग्रे बिट)
3. पुढे, दुसऱ्या आणि तिसऱ्या बायनरी बिटचा XOR घ्या. दुसरा बिट 1 आहे आणि तिसरा बिट 0 आहे. हे बिट पुन्हा वेगळे आहेत त्यामुळे परिणामी ग्रे बिट 1 असेल (तिसरा ग्रे बिट)
4. पुढे, तिसऱ्या आणि चौथ्या बायनरी बिटचा XOR घ्या. तिसरा बिट 0 आहे आणि चौथा बिट 0 आहे. हे सारखेच असल्याने परिणामी ग्रे बिट 0 असेल (चौथा ग्रे बिट)
5. शेवटी, चौथ्या आणि पाचव्या बायनरी बिटचा XOR घ्या. चौथा बिट 0 आहे आणि पाचवा बिट 1 आहे. हे बिट वेगळे आहेत त्यामुळे परिणामी ग्रे बिट 1 असेल (पाचवा ग्रे बिट)
6. त्यामुळे 01001 च्या बायनरी ते ग्रे कोड रूपांतरणाचा परिणाम पूर्ण झाला आहे आणि समतुल्य ग्रे कोड 01101 आहे.



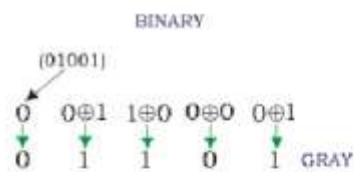
के-मॅप वापरून बायनरी ते ग्रे कोड कन्व्हर्टरची रचना:-

सत्य सारणी:-

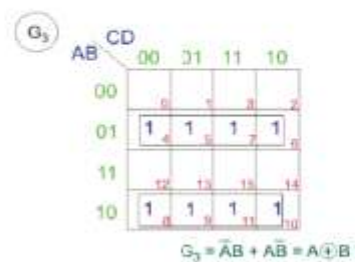
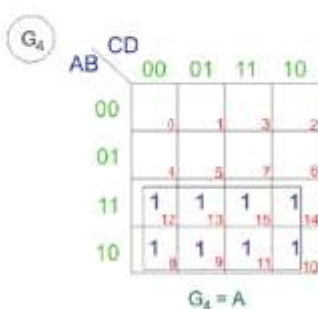
तक्का क्रमांक 3.7- सत्य सारणी

Decimal Number	Four bit Binary Number				Four bit Gray Code			
	A	B	C	D	A	B	C	D
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

खाली दिलेले एक उदाहरण:



ग्रे कोड आउटपुटसाठी के-मॅप:-



**G<sub>2</sub>**

AB \ CD	00	01	11	10
00	0	1	3	2
01	1	1		
11	1	1	1	1
10			1	1

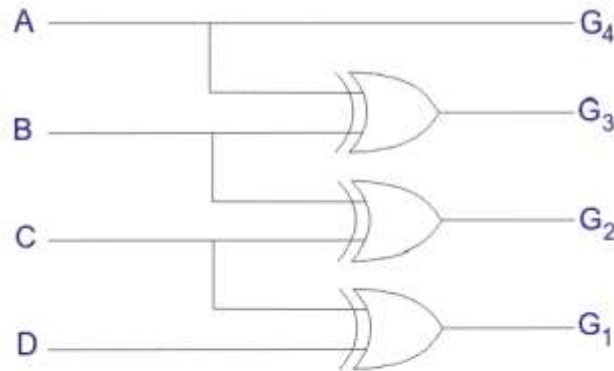
$G_2 = BC + \bar{B}C = B \oplus C$

**G<sub>1</sub>**

AB \ CD	00	01	11	10
00	0	1	3	2
01		1		1
11	1	1	1	1
10		1		1

$G_1 = \bar{C}D + CD = C \oplus D$

बायनरी ते ग्रे कोड कन्वर्टर डिझाइन:-



आकृती.3.13- बायनरी ते ग्रे कोड कन्वर्टर

#### ग्रे ते बायनरी कोड कनवर्टर:

ग्रे ते बायनरी कोड कन्वर्टरमध्ये, इनपुट हा ग्रे कोड असतो आणि आउटपुट हा त्याचा समतुल्य बायनरी कोड असतो.

ग्रे कोड ते बायनरी रूपांतरण:

ग्रे कोड ते बायनरी रूपांतरण ही पुन्हा एक अतिशय सोपी प्रक्रिया आहे.

खालील स्टेप्स मुळे या प्रकारच्या रूपांतरणावर तुमची कल्पना स्पष्ट होऊ शकते.

1. बायनरी क्रमांकाचा **MSB** दिलेल्या ग्रे कोडच्या **MSB** सारखा असेल.
2. आता जर दुसरा ग्रे बिट **0** असेल तर दुसरा बायनरी बिट आधीच्या किंवा पहिल्या बिट सारखाच असेल. ग्रे बिट **1** असल्यास दुसरा बायनरी बिट बदलेल. जर इकडे(ग्रे बिट) **1** असेल तर तिकडे(बायनरी बिट) **0** असेल आणि इकडे(ग्रे बिट) **0** असेल तर तिकडे(बायनरी बिट) **1** असेल.
3. ग्रे कोड ते बायनरी रूपांतरण करण्यासाठी सर्व बिट्ससाठी स्टेप केली जाते.

खाली दिलेले एक उदाहरण स्पष्ट करेल.

0	1	1	0	1	GRAY
↓	↓	↓	↓	↓	
0	→ 1	→ 0	→ 0	→ 1	BINARY

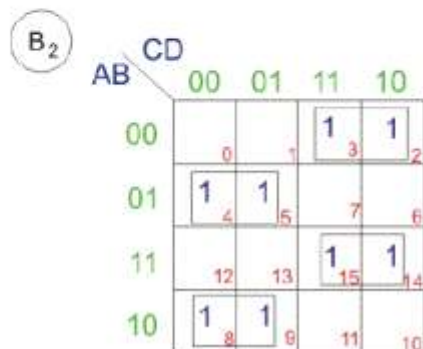
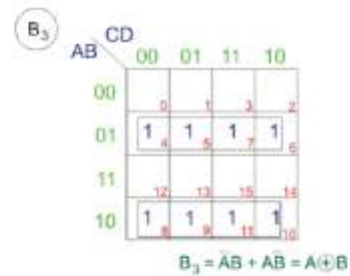
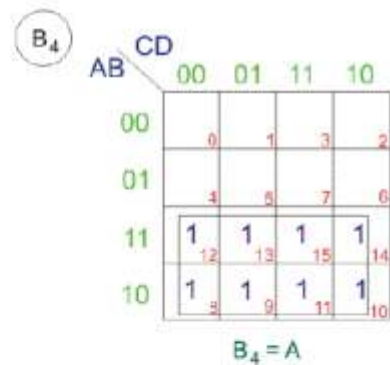
4 बिट ग्रे ते बायनरी कोड कन्वर्टर डिज़ाइन करण्यासाठी, प्रथम एक ग्रे कोड रूपांतरण तक्ता काढावा लागेल, खाली दर्शविल्याप्रमाणे:

सत्य सारणी:-

तक्ता क्रमांक 3.8-सत्य सारणी

Decimal Number	Four bit Gray Code				Four bit Binary Number			
	A	B	C	D	A	B	C	D
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	1	0	0	1	0
3	0	0	1	0	0	0	1	1
4	0	1	1	0	0	1	0	0
5	0	1	1	1	0	1	0	1
6	0	1	0	1	0	1	1	0
7	0	1	0	0	0	1	1	1
8	1	1	0	0	1	0	0	0
9	1	1	0	1	1	0	0	1
10	1	1	1	1	1	0	1	0
11	1	1	1	0	1	0	1	1
12	1	0	1	0	1	1	0	0
13	1	0	1	1	1	1	0	1
14	1	0	0	1	1	1	1	0
15	1	0	0	0	1	1	1	1

के-मॅप:-



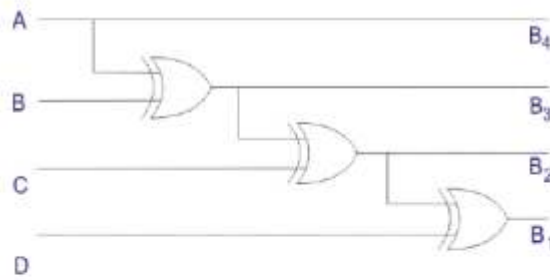
$$\begin{aligned}
 B_2 &= \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}BC \\
 &= \bar{A}(\bar{B}\bar{C} + \bar{B}C) + \bar{A}(B\bar{C} + BC) \\
 &= \bar{A}(\bar{B}\bar{C} + \bar{B}C) + \bar{A}(B\bar{C} + BC) \\
 &= \bar{A}(\bar{B} \oplus C) + \bar{A}(B \oplus C) = A \oplus B \oplus C
 \end{aligned}$$

**B<sub>1</sub>**

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

$$B_1 = \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}C\bar{D} + \bar{A}B\bar{C}\bar{D} + \bar{A}BCD + A\bar{B}\bar{C}D + A\bar{B}C\bar{D} + AB\bar{C}\bar{D} + ABCD = A \oplus B \oplus C \oplus D$$

ग्रे ते बायनरी कनवर्टर डिझाइन-



आकृती.3.14- ग्रे ते बायनरी कनवर्टर

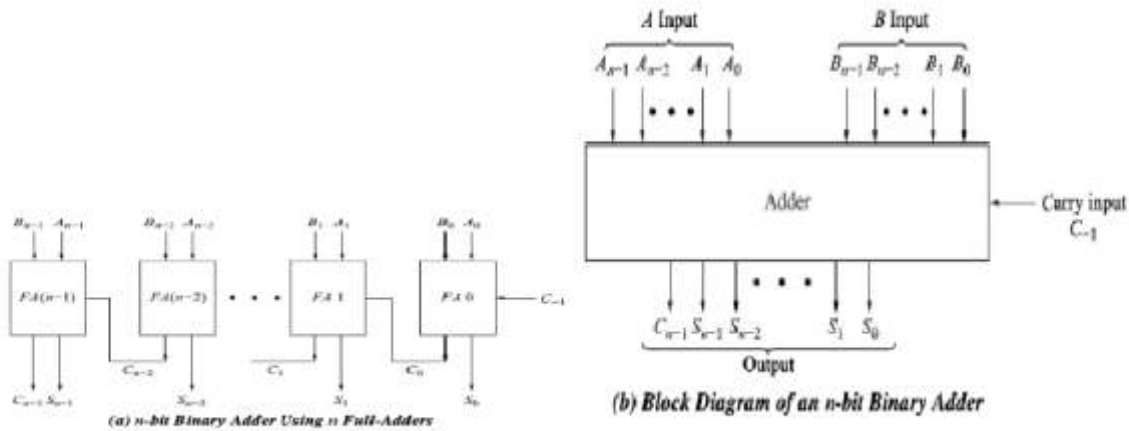
### 3.4 IC 7483 वापरून अॅडर आणि सबट्रॅक्टर तसेच बीसीडी अॅडर -

बीसीडी (BCD):- हे थेट बायनरीमध्ये कोड केलेल्या संख्येचे दशांश प्रतिनिधित्व आहे, डिजिट नुसार डिजिट. उदाहरणार्थ, संख्या (9527)<sub>10</sub>

(1001 0101 0010 0111) BCD म्हणून दर्शविली जाते.

बीसीडी अॅडर BCD ADDER:

हे एक सर्किट आहे जे दोन बीसीडी डिजिट जोडते आणि बीसीडीमध्येही डिजिट ची बेरीज तयार करते. दोन एन-बिट (n bit) बायनरी संख्या जोडण्यासाठी अॅडर सर्किटमध्ये एन-बिट फुल अॅडर सर्किट्स असतात, ते दोन एन-बिट बायनरी संख्या इनपुट म्हणून स्वीकारतात आणि बेरीज म्हणून (n+1) बिट बायनरी संख्या तयार करतात. आकृती(a) फुल ऍडर्स वापरून एन-बिट फुल अॅडर आणि आकृती(b) त्याचा ब्लॉक डायग्राम दाखवते



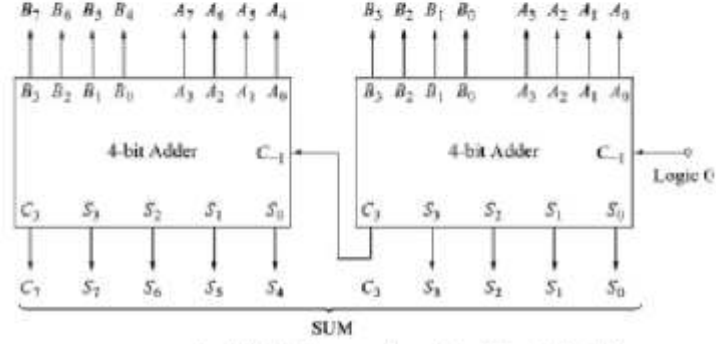
आकृती. 3.15 - फुल ऍडर्स वापरून एन-बिट फुल अॅडर

आकृती.3.16- एन-बिट फुल अॅडर ब्लॉक डायग्राम

ॲड्सचे कॅस्केडिंग:-

जोडायच्या संख्येची बिट लांबी ॲडर्स कॅस्केड करून वाढवता येते. आकृतीदोन 4-बिट ॲडर्स वापरून 8-बिट ॲडर दाखवते.

Here  $A_7A_6\cdots A_1A_0$  and  $B_7B_6\cdots B_1B_0$  are the two 8-bit numbers to be added and  $C_7S_7S_6\cdots S_1S_0$  is the sum.



An 8-bit Adder as a Cascade of Two 4-bit Adders

आकृती 3.17-8-बिट ॲडर

### 3.4.1 IC- 7483 वापरून 4-बिट (1- डिजिट) BCD ॲडर-

1. 4-बिट बायनरी ॲडर IC (7483) BCD संख्या जोडण्यासाठी वापरला जाऊ शकतो.
2. यामध्ये, जर 4-बिट बेरीज आउटपुट वैध BCD डिजिट नसेल, किंवा कॅरी C3 व्युत्पन्न झाला असेल, तर योग्य परिणाम मिळविण्यासाठी बेरीजमध्ये दशांश 6 (0 1 1 0) जोडणे आवश्यक आहे.
3. बीसीडी ॲडरला कॅस्केड करून अनेक अंकी 0 ते 9 अशी संख्या जोडली जाऊ शकते.

बीसीडी बेरीजचे नियम:

1. बायनरी बेरीजचे नियम वापरून दोन संख्या जोडा.
2. जर 4 बिट बेरीज 9 पेक्षा जास्त असेल किंवा कॅरी जन्मत झालो असेल तर बेरीज अवैध आहे. बेरीज दुरुस्त करण्यासाठी बेरीजमध्ये 0110 म्हणजेच (6)<sub>10</sub> जोडा. या जोडणीतून कॅरी व्युत्पन्न झाल्यास ते पुढील उच्च क्रमाच्या BCD अंकात जोडा.
3. जर 4 बिट बेरीज 9 पेक्षा कमी किंवा 9 च्या समान असेल तर बेरीज योग्य स्वरूपात असेल.

बीसीडी ॲडरची रचना:

- प्रारंभिक जोडणीसाठी 4 बिट बायनरी ॲडर वापरला जातो. म्हणजे दोन 4 बिट संख्यांची बायनरी जोडणी. ( $C_{in} = 0$  सह)
- लॉजिक सर्किट समजण्यासाठी जर बेरीज 9 पेक्षा जास्त असेल तर किंवा कॅरी 1 असेल तर, हे डिजिटल सर्किट उच्च 1 देईल अन्यथा त्याचे आउटपुट 0 असेल.
- बेरीजमध्ये (0110)<sub>2</sub> जोडण्यासाठी आणखी एक 4-बिट ॲडर 9 पेक्षा जास्त आहे किंवा कॅरी 1 आहे.

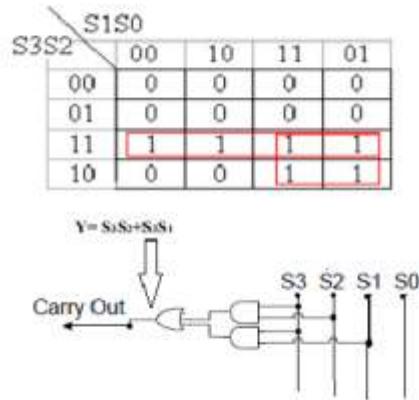
सत्य सारणी:-

अवैध(इनवॅलिड) बीसीडी तपासण्यासाठी बीसीडी ॲडरसाठी कॉम्बिनेशनल सर्किटच्या डिझाइनसाठी

तक्ता क्रमांक 3.8- सत्य सारणी

INPUT				OUTPUT
S3	S2	S1	S0	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

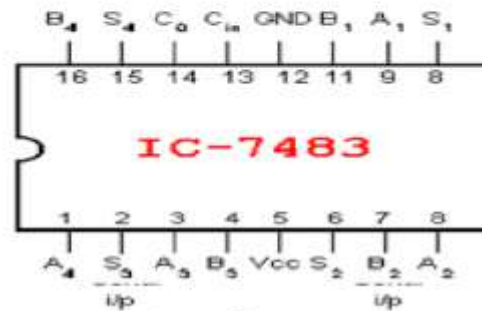
K-map:- आउटपुटच्या कमी झालेल्या बुलियन एक्सप्रेशनसाठी:



आकृती.3.18-बीसीडी ॲडरसाठी कॉम्बिनेशनल सर्किट

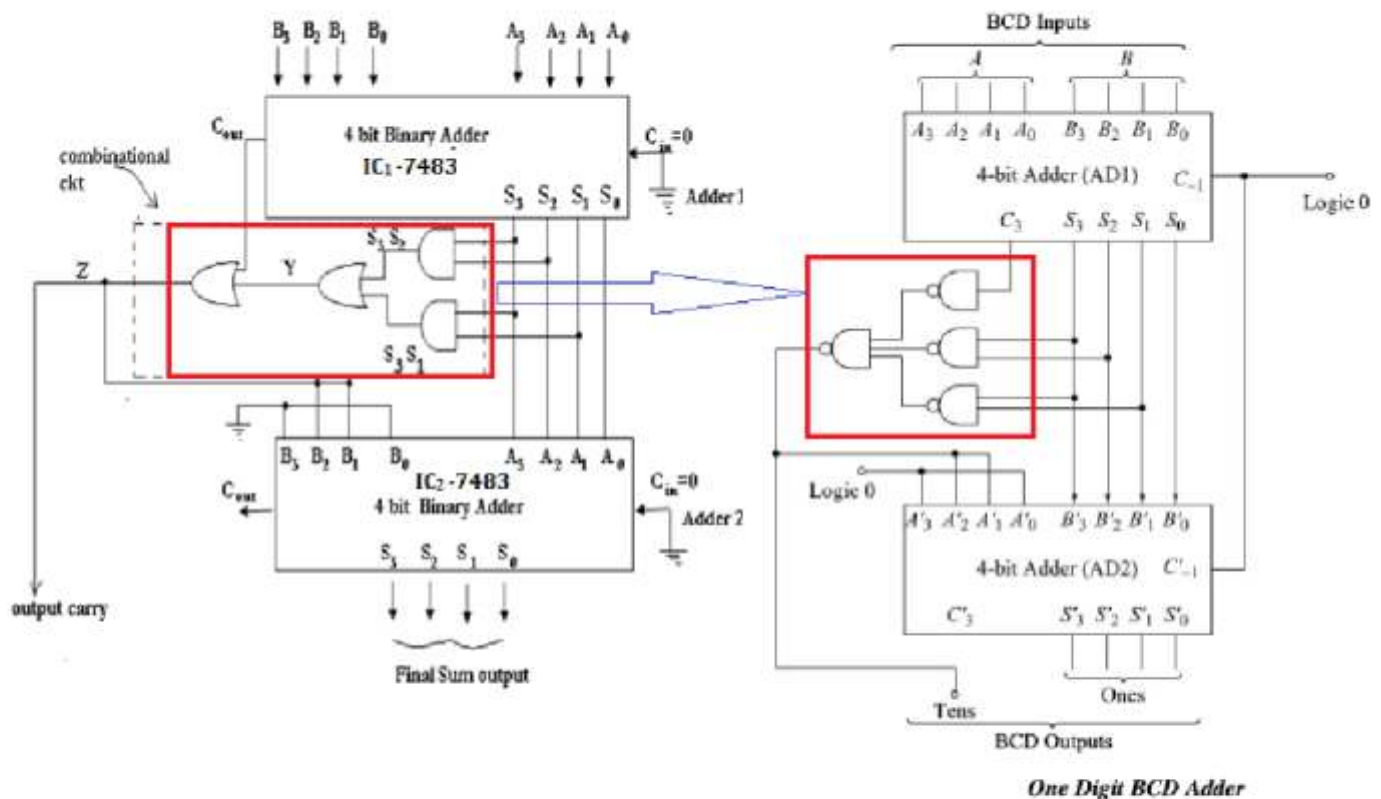


IC 7483 चा पिन डायग्राम:-



आकृती.3.19- पिन डायग्राम

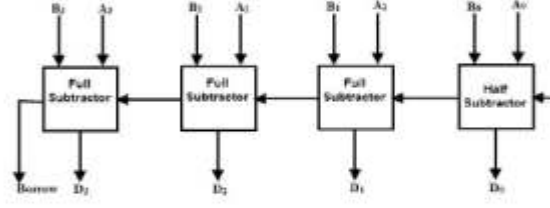
IC 7483 वापरून एक अंकी BCD अँडर डिझाइन केले:



**आकृती.3.20- एक अंकी BCD अॅडर**

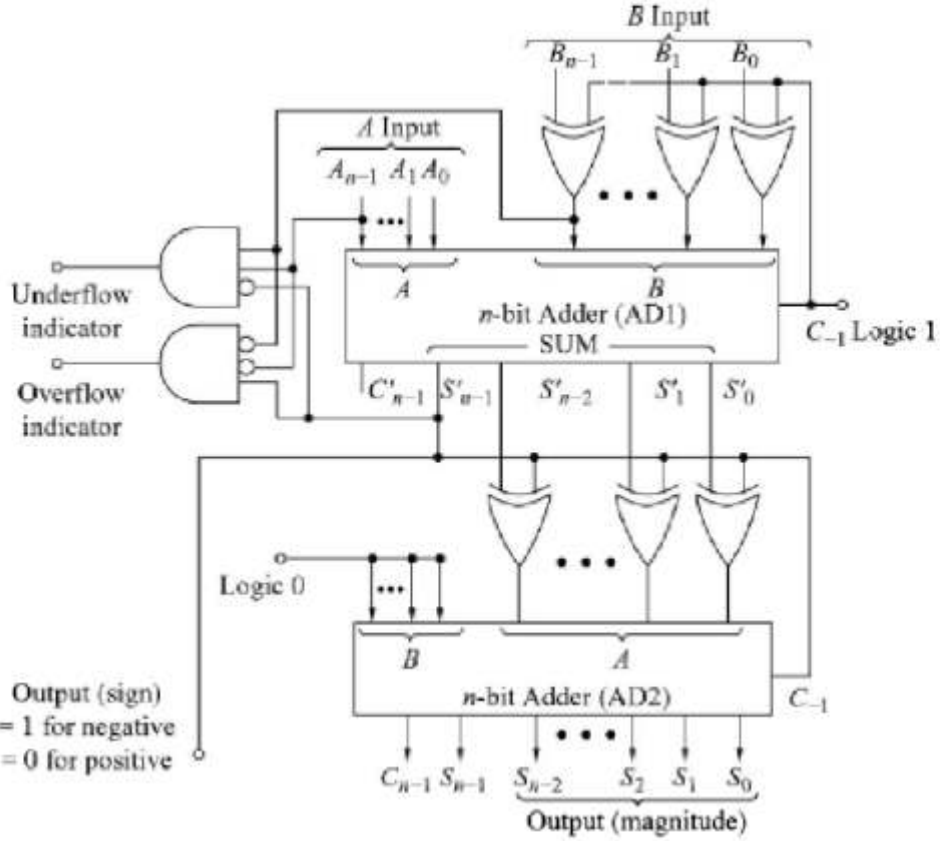
### 3.4.2 4 बिट सबट्रॅक्टर हाफ आणि फुल सबट्रॅक्टर वापरून

आवृत्ती 3.21 एक अर्धा वजाबाकी(हाफ सबट्रॅक्टर) आणि तीन पूर्ण वजाबाकी(फुल सबट्रॅक्टर) जोडून तयार केलेला 4 बिट समांतर(पॅरेलल) बायनरी वजाबाकी दाखवते. या वजाबाकीमध्ये, 4 बिट माइन्युएंड A3A2A1A0 4 बिट सबट्राहेंड B3B2B1B0 ने वजा केला जातो आणि फरक आउटपुट D3D2D1D0 देतो. प्रत्येक वजाबाकीचे BORROW आउटपुट पुढील आधीच्या वजाबाकीशी BORROW इनपुट म्हणून जोडलेले आहे.



आकृती 3.21-4 बिट समांतर(पैरेलल) बायनरी वजाबाकी

अॅडर वापरून वजाबाकी:-



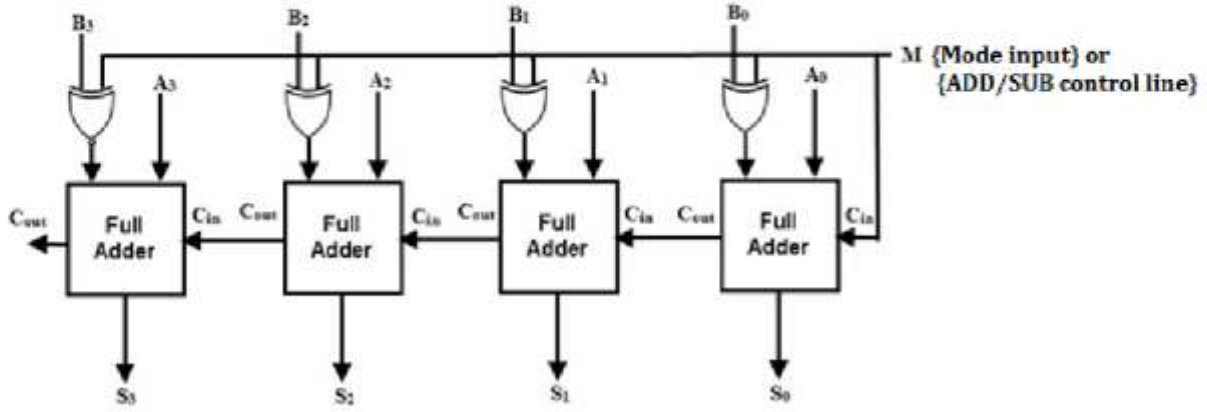
*n-bit Subtractor Circuit Using n-bit Adders*

आकृती.3.22(a)- एन-बिट सबट्रॅक्टर सर्किट

A आणि B या दोन संख्या एकाच चिन्हाचे किंवा विरुद्ध चिन्हाचे असू शकतात. जर दोन संख्या विपरीत चिन्हाचे असतील तर ओव्हरफ्लो किंवा अंडरफ्लोची समस्या उद्भवते. ओव्हरफ्लो होतो जेव्हा वजाबाकी ऑपरेशन सर्वात मोठ्या संभाव्य संख्येपेक्षा (Largest possible number) मोठी संख्या तयार करते जी दुसरीकडे n-बिट्सद्वारे दर्शविली जाऊ शकते, तेव्हा अंडरफ्लो होतो जेव्हा उत्पादित परिणाम सर्वात लहान संख्येपेक्षा (smallest number) लहान असतो ज्याला n-बिट्स द्वारे दर्शविले जाऊ शकते. खालील आकृती एन-बिट अॅडर्सचा वापर करून एन-बिट सबट्रॅक्टर सर्किट दाखवते जेथे n-बिट नंबरमधील वजाबाकी झाल्यावर हा ओव्हरफ्लो किंवा अंडरफ्लो तपासला जाईल.

### 3.4.3 ADD/SUB कंट्रोल लाइन किंवा मोड इनपुट (M) वापरून 4-बिट बायनरी अॅडर / वजाबाकी:-

बेरीज आणि वजाबाकी दोन्हीची क्रिया एका सामान्य बायनरी अॅडरद्वारे केली जाऊ शकते. अशा बायनरी सर्किटची रचना खालील आकृतीमध्ये दर्शविल्याप्रमाणे प्रत्येक पूर्ण अॅडरसह Ex-OR गेट जोडून केली जाऊ शकते. आकृती 3.22(b) 4 बिट समांतर बायनरी अॅडर/सबट्रॅक्टर दाखवते ज्यामध्ये A<sub>3</sub>A<sub>2</sub>A<sub>1</sub>A<sub>0</sub> आणि B<sub>3</sub>B<sub>2</sub>B<sub>1</sub>B<sub>0</sub> असे दोन 4 बिट इनपुट आहेत. मोड इनपुट कंट्रोल लाइन M पूर्ण अॅडरच्या लीस्ट सिग्निफिकंट बिटच्या कॅरी इनपुटसह जोडलेली आहे. ही नियंत्रण रेषा ऑपरेशनचा प्रकार ठरवते, बेरीज किंवा वजाबाकी.

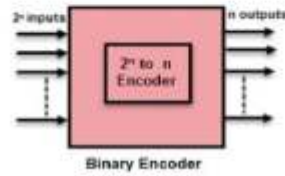


आकृती.3.22(b) 4 बिट समांतर बायनरी अॅडर/सबट्रॅक्टर

### 3.5 एनकोडर्स / डीकोडर्स-

#### 3.5.1 एनकोडर्स(Encoders)-

आवश्यकता - ऐक्टिव्ह इनपुट सिग्नलचे विशिष्ट कोड च्या स्वरूपातील आउटपुट मध्ये रूपांतर करणाऱ्या सर्किटला 'एनकोडर' म्हणतात. एनकोडिंग क्रियेमध्ये नेहमीच्या परिचित किन्हांचे रूपांतर विशिष्ट कोड मध्ये केले जाते. डिजिटल सर्किट्स मध्ये बायनरी किंवा BCD इनपुट ची आवश्यकता असते. अशावेळी एनकोडरचा उपयोग होतो. ह्यासाठी डेसिमल ते BCD एनकोडर ची आवश्यकता असते.



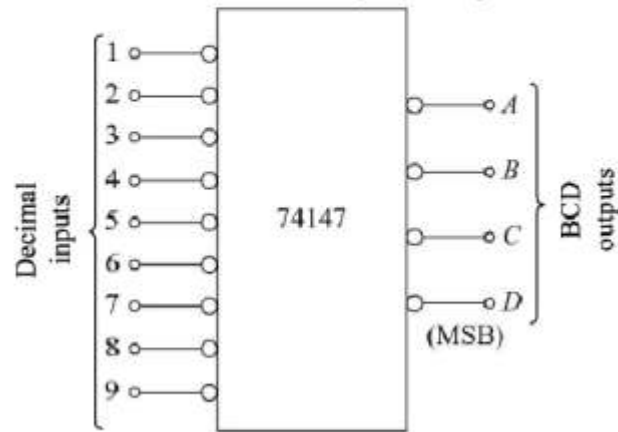
आकृती 3.23- एनकोडर

#### एनकोडरचे प्रकार:-

दशांश- ते BCD प्राधान्य एनकोडर(Decimal to BCD Priority encoder)-

या प्रकारचा एनकोडर 4-बिट बीसीडी आउटपुटमध्ये दशांश अंक एनकोड करण्याप्रमाणे सामान्य दशांश ते बीसीडी एनकोडरप्रमाणेच कार्य करतो. तथापि, त्यात प्राधान्य प्रदान करण्याच्या अतिरिक्त कार्यक्षमतेचा समावेश आहे. म्हणजे बीसीडी आउटपुट इतर सर्व इनपुट्सकडे दुर्लक्ष करून इनपुटवर दिसणाऱ्या दशांश अंकाच्या सर्वोच्च प्राधान्याशी संबंधित आहे.

तक्ता 3.10 दशांश ते BCD एनकोडरचे सत्य सारणी दर्शवते. या एनकोडरमध्ये 1 ते 9 पर्यंत दशांश अंकांचे प्रतिनिधित्व करणारे नऊ ऐक्टिव्ह लो (Active low) इनपुट आहेत. ऐक्टिव्ह इनपुटच्या सर्वोच्च क्रमानुसार, तो उलटा BCD कोड तयार करतो.



**Block Diagram of 74147  
Decimal-to-BCD Priority Encoder**

आकृती 3.24- प्राधान्य एनकोडर

तक्ता क्रमांक 3.10-सत्य सारणी

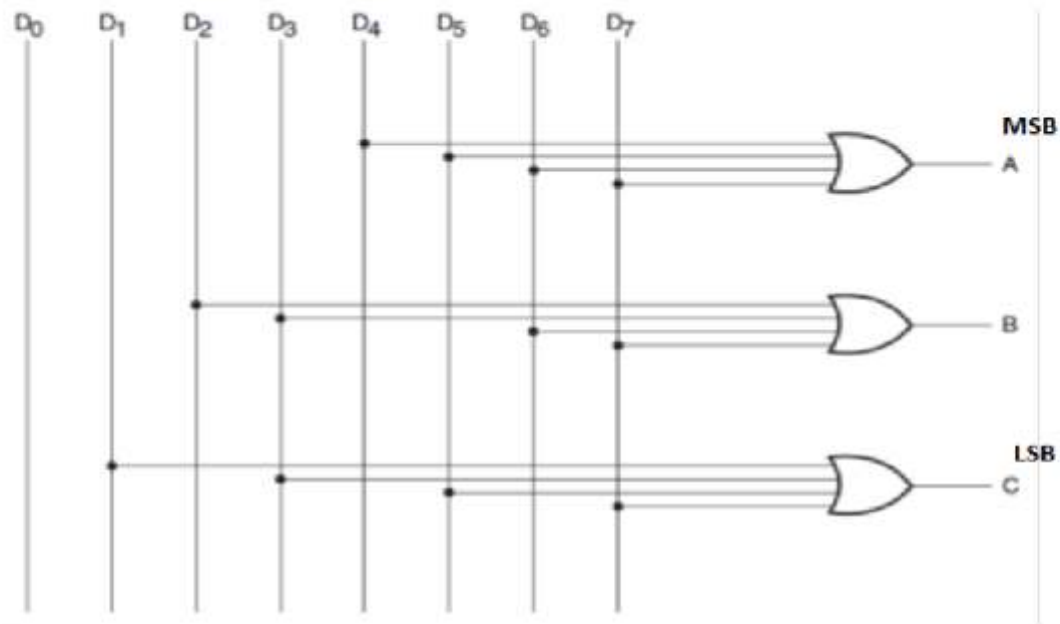
Active-low decimal inputs									Active-low BCD outputs			
1	2	3	4	5	6	7	8	9	D	C	B	A
1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	0
x	0	1	1	1	1	1	1	1	1	1	0	1
x	x	0	1	1	1	1	1	1	1	1	0	0
x	x	x	0	1	1	1	1	1	1	0	1	1
x	x	x	x	0	1	1	1	1	1	0	1	0
x	x	x	x	x	0	1	1	1	1	0	0	1
x	x	x	x	x	x	0	1	1	1	0	0	0
x	x	x	x	x	x	x	0	1	1	0	1	1
x	x	x	x	x	x	x	x	0	1	1	1	0

तक्ता 3.10 10-ओळी(10 line) दशांश ते चार-ओळ(4 line) बीसीडी एनकोडरचे लॉजिक चिन्ह आणि सत्य सारणी दर्शवते जे उच्च-ऑर्डरसाठी प्राधान्य एनकोडिंग प्रदान करते. डिजिट 9 ला सर्वोच्च प्राधान्य आहे. दर्शविलेल्या कार्यात्मक(active) सारणीमध्ये, सर्वोच्च प्राधान्य असलेली इनपुट लाइन ज्यावर LOW असते ती इतर इनपुट लाइनच्या तर्क स्थितीकडे दुर्लक्ष करून एनकोड केलेली असते.

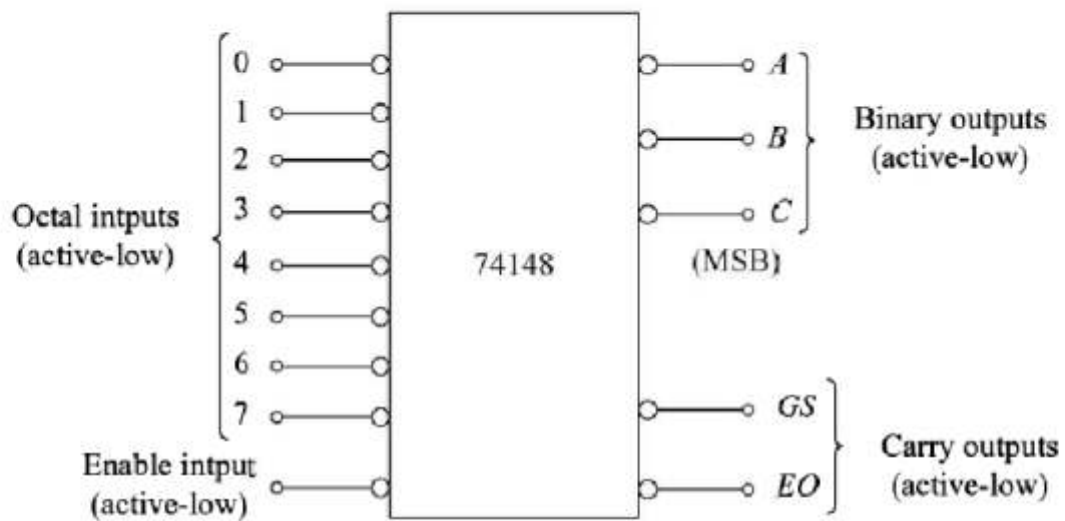
• ऑक्टल ते बायनरी प्राधान्य एनकोडर(Octal to Binary Priority Encoder)-

ऑक्टल-टू-बायनरी एनकोडरचे उदाहरण घेऊ. अशा एनकोडरमध्ये आठ इनपुट रेषा असतील, प्रत्येक ऑक्टल अंक दर्शविते आणि तीन आउटपुट रेषा तीन-बिट बायनरी समतुल्य दर्शवितात. अशा एनकोडरचे सत्य सारणी यात दिलेली आहे की D0 ते D7 0 ते 7 अष्टक अंक दर्शवितात. A, B आणि C बायनरी अंकांचे प्रतिनिधित्व करतात.

आठ इनपुट ओळींमध्ये  $2^8 = 256$  संभाव्य जोड्या असतील. तथापि, ऑक्टल-टू-बायनरी एनकोडरच्या बाबतीत, या 256 संयोगांपैकी फक्त आठचाच अर्थ असेल. इनपुट व्हेरिएबल्सचे उरलेले कॉम्बिनेशन म्हणजे 'डोन्ट केअर' इनपुट कॉम्बिनेशनस. तसेच, एकावेळी इनपुट ओळींपैकी एकच लॉजिक '1' स्थितीत असते. सत्य सारणीद्वारे वर्णन केलेल्या ऑक्टल-टू-बायनरी एनकोडरची हार्डवेअर अंमलबजावणी आकृती 3.25 दाखवते. या सर्किटमध्ये अशी कमतरता आहे की जेव्हा सर्व इनपुट लाइन लॉजिक '0' स्थितीत असतात तेव्हा ते सर्व 0s आउटपुट अनुक्रम तयार करते. सर्व 0s इनपुट क्रम दर्शविण्यासाठी अतिरिक्त ओळ ठेवून यावर मात केली जाऊ शकते. D2, D4 आणि D7 या इनपुट लाईन्स एकाच वेळी लॉजिक '1' स्थितीत आहेत असे गृहीत धरूया. अशा स्थितीत, फक्त D7 एनकोड केले जाईल आणि आउटपुट 111 असेल. अशा प्राधान्य एनकोडरचे सत्य सारणी नंतर सुधारित केले जाईल, याचा अर्थ असा होतो की, जर D7 = 1, तर, इतर इनपुटच्या तर्क स्थितीकडे दुर्लक्ष करून, आउटपुट 111 आहे. D7 फक्त एनकोड केले जाईल.



आकृती 3.25- ऑक्टल ते बायनरी प्राधान्य एनकोडर



**Block Diagram of 74148 Octal-to-Binary Priority Encoder**

आकृती 3.26- ऑक्टल ते बायनरी प्राधान्य एनकोडर



Truth Table of 74148

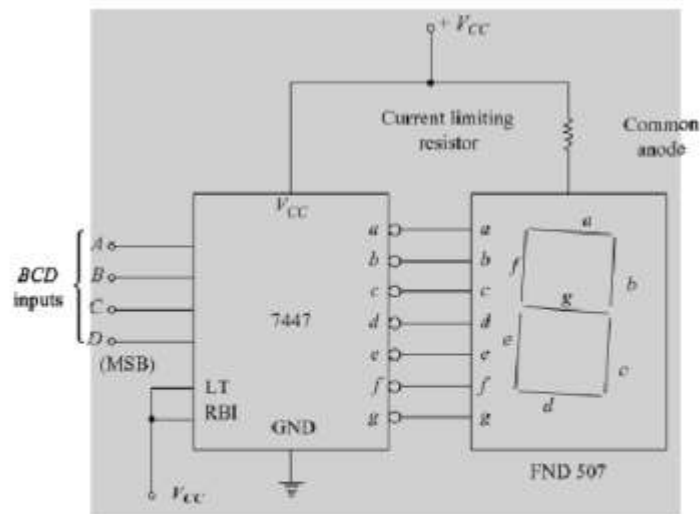
EI	Inputs								Outputs				
	0	1	2	3	4	5	6	7	C	B	A	GS	EO
1	X	X	X	X	X	X	X	X	1	1	1	1	1
0	0	1	1	1	1	1	1	1	1	1	1	0	1
0	X	0	1	1	1	1	1	1	1	1	0	0	1
0	X	X	0	1	1	1	1	1	1	0	1	0	1
0	X	X	X	0	1	1	1	1	1	0	0	0	1
0	X	X	X	X	0	1	1	1	0	1	1	0	1
0	X	X	X	X	X	0	1	1	0	1	0	0	1
0	X	X	X	X	X	X	0	1	0	0	1	0	1
0	X	X	X	X	X	X	X	0	0	0	1	0	1
0	X	X	X	X	X	X	X	X	0	0	0	0	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0

### 3.5.2 डीकोडर (Decoder) :-

आवश्यकता- दशमान संख्या पद्धत हि सर्वात जास्त परिचित असणारी संख्या पद्धत आहे, परंतु अनेकदा उल्लेख्य डेटा बायनरी किंवा बीसीडी प्रकारचा असतो. तसेच डिजिटल सिस्टिमचे आउटपुट बायनरी किंवा बीसीडी प्रकारचे असते. म्हणून, त्याचे रूपांतर दशमान पद्धतीमध्ये करणे आवश्यक असते. यासाठी डीकोडर चा उपयोग होतो. त्याचे कार्य एनकोडरच्या विरुद्ध प्रकारचे असते. बीसीडी चे रूपांतर दशमान मध्ये करण्यासाठी तसेच इतर विविध प्रकारचे डीकोडर वापरतात.

BCD ते 7-सेगमेंट डिकोडर:-

सेव्हन सेगमेंट डिस्प्ले (Seven segment display) हे डिजिटल सिस्टीममध्ये वापरले जाणारे सर्वात लोकप्रिय डिस्प्ले डिवाइस आहे. हे उपकरण वापरून डेटा प्रदर्शित करण्यासाठी डेटा BCD मधून 7-सेगमेंट कोडमध्ये रूपांतरित करावा लागेल. डीकोडर / ड्रायव्हर सर्किटमध्ये बीसीडी डेटासाठी 4 इनपुट लाइन आणि 7-सेगमेंट डिस्प्ले चालविण्यासाठी 7 आउटपुट लाइन आहेत. डिकोडरच्या "a to g" वरील आउटपुट टर्मिनल्स अनुक्रमे डिस्प्लेच्या "a ते g" शी जोडले जावेत. जर आउटपुट कमी सक्रिय असतील तर 7-सेगमेंट LED कॉमन ऍनोड (common anode) प्रकारातील असणे आवश्यक आहे, तर आउटपुट जास्त सक्रिय असल्यास 7-सेगमेंट LED कॉमन कॅथोड (common cathode) प्रकारातील असणे आवश्यक आहे



A 7447 Driving a 7-Segment LED Display

आकृती 3.27- सेव्हन सेगमेंट डिस्प्ले



एनकोडर आणि डीकोडर मधील तुलना:-

पॅरामीटर	एन्कोडर	डीकोडर
इनपुट	मूळ संदेश सिग्नल (active input signal)	कोडेड बायनरी इनपुट
आउटपुट	कोडेड बायनरी आउटपुट	मूळ संदेश (active signal)
इनपुट लाईन	$2^n$	$n$
आउटपुट लाईन	$n$	$2^n$
ऑपरेशन	सोपे	गुंतागुंतीचे (complex)
मूलभूत तर्क घटक	OR गेट	AND गेट व NOT गेट
उपयोग	ई-मेल, व्हिडिओ एनकोडर	मायक्रोप्रोसेसर, मेमरी चिप्स इ.

### 3.6.1 मल्टीप्लेक्सर (MUX)-

-हे एक कॉम्बिनेशनल सर्किट आहे जे अनेक इनपुट लाइन्समधून बायनरी माहिती निवडते आणि एका आउटपुट लाइनवर निदेशित करते.

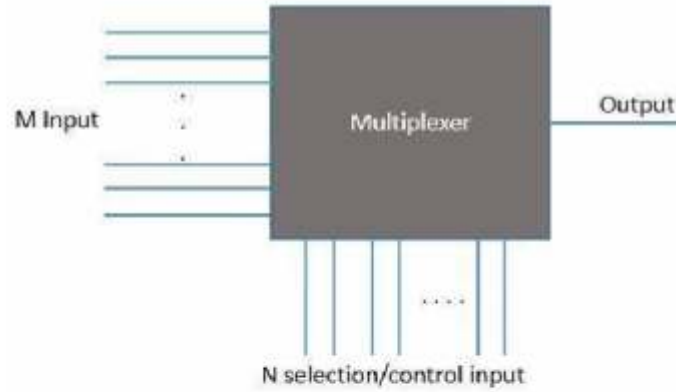
-विशिष्ट इनपुट लाइनची निवड सर्किटमधील निवड रेषांच्या संचाद्वारे नियंत्रित केली जाते.

-MUX ला अनेक ते एक, डेटा निवडकर्ता म्हणतात.

मल्टिप्लेक्सिंग म्हणजे थोड्या संख्येने चॅनेल किंवा लाइन वर मोठ्या संख्येने माहिती युनिट्स प्रसारित करणे. त्याचे संक्षिप्त रूप MUX असे आहे.

- $2^N$  इनपुट लाइन आणि 'N' कंट्रोल लाइन आहेत ज्यांचे बिट कॉम्बिनेशन कोणते इनपुट निवडले आहे हे निर्धारित करतात.

$M = 2^N$  म्हणजे  $N = \log M$  जेथे M = डेटा इनपुटची संख्या आणि N = कंट्रोल लाइनची संख्या



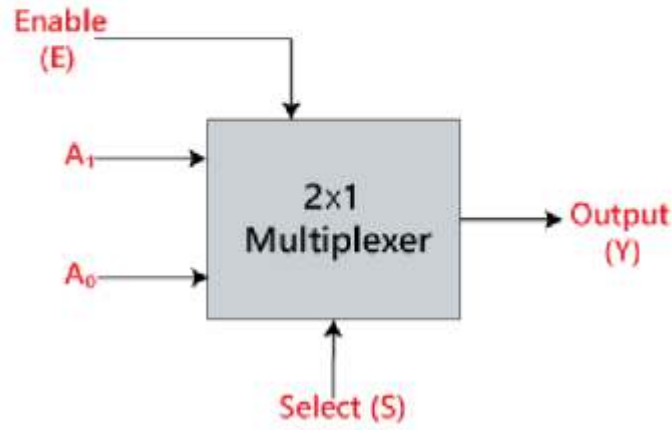
आकृती 3.28- मल्टीप्लेक्सर

मल्टीप्लेक्सरचे प्रकार-

मल्टीप्लेक्सर्सचे अनेक प्रकार आहेत आणि काही खालीलप्रमाणे सूचीबद्ध आहेत:

- 2:1 मल्टीप्लेक्सर
- 4 :1 मल्टीप्लेक्सर
- 8:1 मल्टीप्लेक्सर
- 16:1 मल्टीप्लेक्सर
- 2:1 मल्टीप्लेक्सर-

2:1 मल्टिप्लेक्सरमध्ये, फक्त दोन इनपुट आहेत, म्हणजे, A0 आणि A1, 1 निवड लाइन, म्हणजे, S0 आणि सिंगल आउटपुट, म्हणजे, Y. निवड लाइन S0 वर उपस्थित असलेल्या इनपुटच्या संयोजनाच्या आधारवर, या 2 इनपुटपैकी एक आउटपुटशी कनेक्ट केले जाईल. 2:1 मल्टिप्लेक्सरचे ब्लॉक आकृती डायग्राम आणि सत्य सारणी खाली दिलेली आहे.



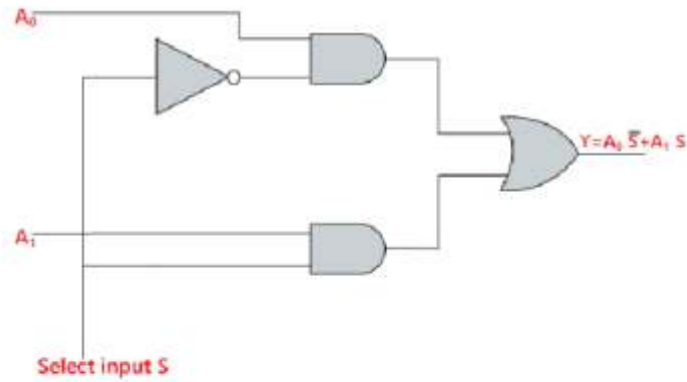
आकृती 3.29-2x1 मल्टिप्लेक्सर

तक्ता क्रमांक 3.11-सत्य सारणी

INPUTS	Output
$S_0$	Y
0	$A_0$
1	$A_1$

लॉजिकल एक्सप्रेशन  $Y = S_0' \cdot A_0 + S_0 \cdot A_1$

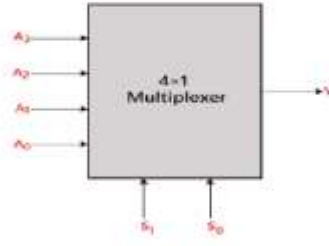
वरील एक्सप्रेशनचे लॉजिकल सर्किट आकृती 3.30 दिले आहे:



आकृती 3.30-2x1 मल्टिप्लेक्सर

4:1 मल्टीप्लेक्सर:

4:1 मल्टिप्लेक्सरमध्ये एकूण चार इनपुट आहेत, म्हणजे,  $A_0$ ,  $A_1$ ,  $A_2$ , आणि  $A_3$ , 2 निवड रेषा, म्हणजे,  $S_0$  आणि  $S_1$  आणि सिंगल आउटपुट, म्हणजे, Y, च्या संयोजनाच्या आधारावर सिलेक्शन लाईन्स  $S_0$  आणि  $S_1$  वर उपस्थित असलेले इनपुट, या 4 इनपुटपैकी एक आउटपुटशी जोडलेले आहे. 4:1 मल्टिप्लेक्सरचे ब्लॉक डायग्राम आणि सत्य सारणी खाली दिलेली आहे.



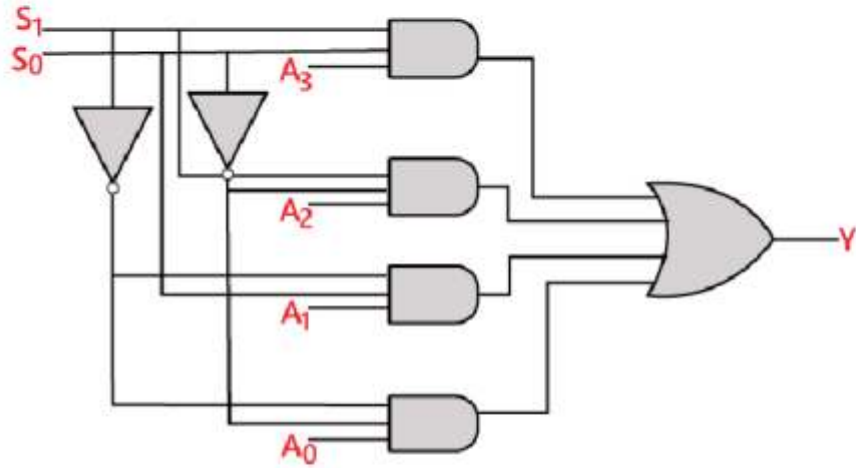
आकृती 3.31-4×1 मल्टिप्लेक्स

तक्ता क्रमांक 3.12-सत्य सारणी

INPUTS		Output
S <sub>1</sub>	S <sub>0</sub>	Y
0	0	A <sub>0</sub>
0	1	A <sub>1</sub>
1	0	A <sub>2</sub>
1	1	A <sub>3</sub>

लॉजिकल एक्सप्रेशन  $Y = S_1' S_0' A_0 + S_1' S_0 A_1 + S_1 S_0' A_2 + S_1 S_0 A_3$

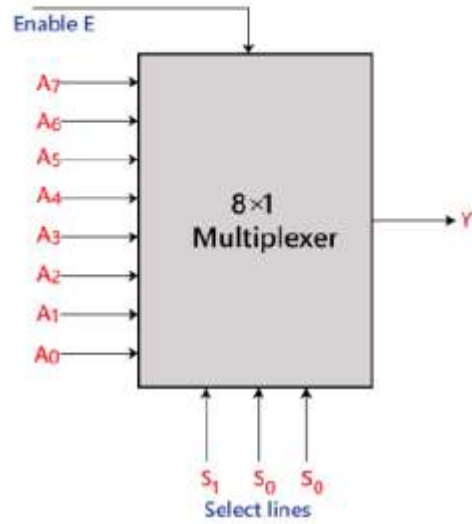
वरील एक्सप्रेशनचे लॉजिकल सर्किट आकृती 3.32 दिले आहे:



आकृती 3.32-4×1 मल्टिप्लेक्स

8:1 मल्टीप्लेक्स;

8 :1 मल्टिप्लेक्समध्ये एकूण आठ इनपुट आहेत, म्हणजे, A<sub>0</sub>, A<sub>1</sub>, A<sub>2</sub>, A<sub>3</sub>, A<sub>4</sub>, A<sub>5</sub>, A<sub>6</sub> आणि A<sub>7</sub>, 3 निवड रेषा, म्हणजे, S<sub>0</sub>, S<sub>1</sub> आणि S<sub>2</sub> आणि एकल आउटपुट, म्हणजे, Y. S<sub>0</sub>, S<sub>1</sub> आणि S<sub>2</sub> सिलेक्शन लाईन्सवर उपस्थित असलेल्या इनपुट्सच्या संयोजनाच्या आधारावर, या 8 इनपुटपैकी एक आऊटपुटशी जोडला जातो. 8:1 मल्टिप्लेक्सचा ब्लॉक डायग्राम आणि सत्य सारणी खाली दिलेली आहे.



आकृती 3.33 – 8:1 मल्टिप्लेक्सर

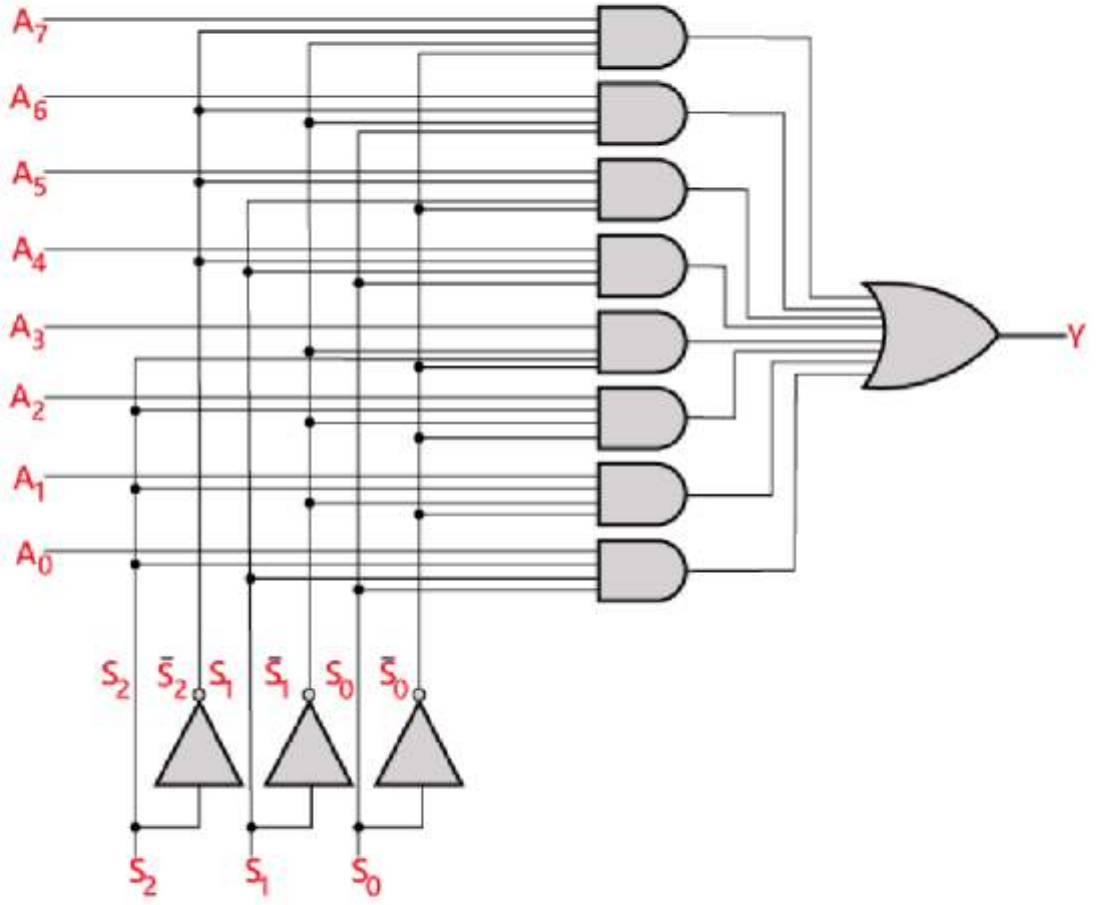
तक्ता क्रमांक 3.14-सत्य सारणी

INPUTS			Output
$S_2$	$S_1$	$S_0$	$Y$
0	0	0	$A_0$
0	0	1	$A_1$
0	1	0	$A_2$
0	1	1	$A_3$
1	0	0	$A_4$
1	0	1	$A_5$
1	1	0	$A_6$
1	1	1	$A_7$

लॉजिकल एक्सप्रेशन

$$Y = S_0' \cdot S_1' \cdot S_2' \cdot A_0 + S_0 \cdot S_1' \cdot S_2' \cdot A_1 + S_0' \cdot S_1 \cdot S_2' \cdot A_2 + S_0 \cdot S_1 \cdot S_2' \cdot A_3 + S_0' \cdot S_1' \cdot S_2 \cdot A_4 + S_0 \cdot S_1' \cdot S_2 \cdot A_5 + S_0' \cdot S_1 \cdot S_2 \cdot A_6 + S_0 \cdot S_1 \cdot S_2 \cdot A_7$$

वरील एक्सप्रेशनचे लॉजिकल सर्किट आकृती 3.34 दिले आहे:



आकृती 3.34

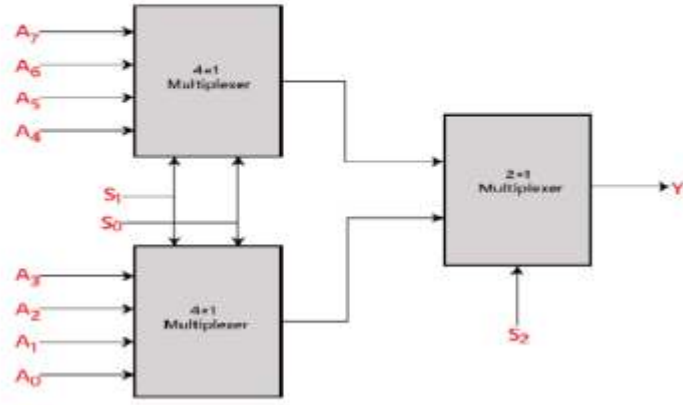
मल्टिप्लेक्सर्सचे कॅस्केडिंग (MUX tree)

4:1 आणि 2:1 मल्टिप्लेक्सर वापरून 8:1 मल्टीप्लेक्सर

लोअर ऑर्डर मल्टीप्लेक्सर वापरून 8:1 मल्टीप्लेक्सर तयार करू शकतो. 8:1 मल्टिप्लेक्सर तयार करण्यासाठी, दोन 4:1 मल्टिप्लेक्सर आणि एक 2:1 मल्टिप्लेक्सरची गरज आहे. 4:1 मल्टिप्लेक्सरमध्ये 2 निवड रेखा, 4 इनपुट आणि 1 आउटपुट आहे. 2:1 मल्टिप्लेक्सरमध्ये फक्त 1 निवड ओळ आहे.

8 डेटा इनपुट मिळविण्यासाठी, दोन 4:1 मल्टिप्लेक्सर्सची आवश्यकता आहे. 4x1 मल्टिप्लेक्सर एक आउटपुट तयार करतो. म्हणून, अंतिम आउटपुट मिळविण्यासाठी, 2:1 मल्टिप्लेक्सरची आवश्यकता आहे. 4:1 आणि 2:1 मल्टिप्लेक्सर वापरून 8:1 मल्टिप्लेक्सरचा ब्लॉक आकृती डायग्राम आकृती 3.35 दिला आहे.

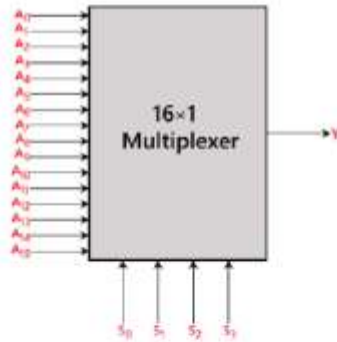




आकृती 3.35

### 16:1 मल्टीप्लेक्सर-

16:1 मल्टीप्लेक्सरमध्ये एकूण 16 इनपुट आहेत, म्हणजे,  $A_0, A_1, \dots, A_{15}$ , 4 निवड ओळी, म्हणजे,  $S_0, S_1, S_2$ , आणि  $S_3$  आणि सिंगल आउटपुट, म्हणजे  $Y$ . च्या आधारावर  $S_0, S_1$  आणि  $S_2$  सेलेक्शन लाईन्सवर उपस्थित असलेल्या इनपुटचे संयोजन, या 16 इनपुटपैकी एक आउटपुटशी कनेक्ट केले जाईल. ब्लॉक डायग्राम आणि 16:1 चे सत्य सारणी.



आकृती 3.36-16x1 मल्टीप्लेक्सर

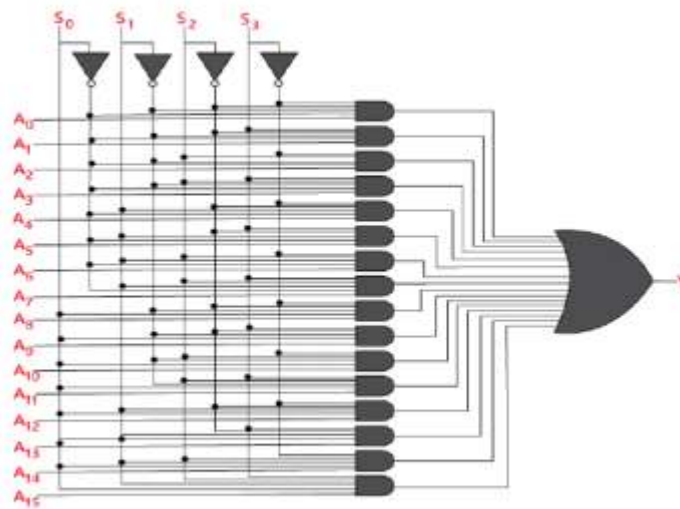
तक्ता क्रमांक 3.15-सत्य सारणी

INPUTS				Output
S <sub>0</sub>	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	Y
0	0	0	0	A <sub>0</sub>
0	0	0	1	A <sub>1</sub>
0	0	1	0	A <sub>2</sub>
0	0	1	1	A <sub>3</sub>
0	1	0	0	A <sub>4</sub>
0	1	0	1	A <sub>5</sub>
0	1	1	0	A <sub>6</sub>
0	1	1	1	A <sub>7</sub>
1	0	0	0	A <sub>8</sub>
1	0	0	1	A <sub>9</sub>
1	0	1	0	A <sub>10</sub>
1	0	1	1	A <sub>11</sub>
1	1	0	0	A <sub>12</sub>
1	1	0	1	A <sub>13</sub>
1	1	1	0	A <sub>14</sub>
1	1	1	1	A <sub>15</sub>

लॉजिकल एक्सप्रेसन

$$Y = A_0.S_0'.S_1'.S_2'.S_3' + A_1.S_0'.S_1'.S_2'.S_3 + A_2.S_0'.S_1'.S_2.S_3' + A_3.S_0'.S_1'.S_2.S_3 + A_4.S_0'.S_1.S_2'.S_3' + A_5.S_0'.S_1.S_2'.S_3 + A_6.S_0'.S_1.S_2.S_3' + A_7.S_0'.S_1.S_2.S_3 + A_8.S_0.S_1'.S_2'.S_3' + A_9.S_0.S_1'.S_2'.S_3 + A_{10}.S_0.S_1'.S_2.S_3' + A_{11}.S_0.S_1'.S_2.S_3 + A_{12}.S_0.S_1.S_2'.S_3' + A_{13}.S_0.S_1.S_2'.S_3 + A_{14}.S_0.S_1.S_2.S_3' + A_{15}.S_0.S_1.S_2.S_3$$

वरील एक्सप्रेसनचे लॉजिकल सर्किट आकृती 3.37 मध्ये दिले आहे:



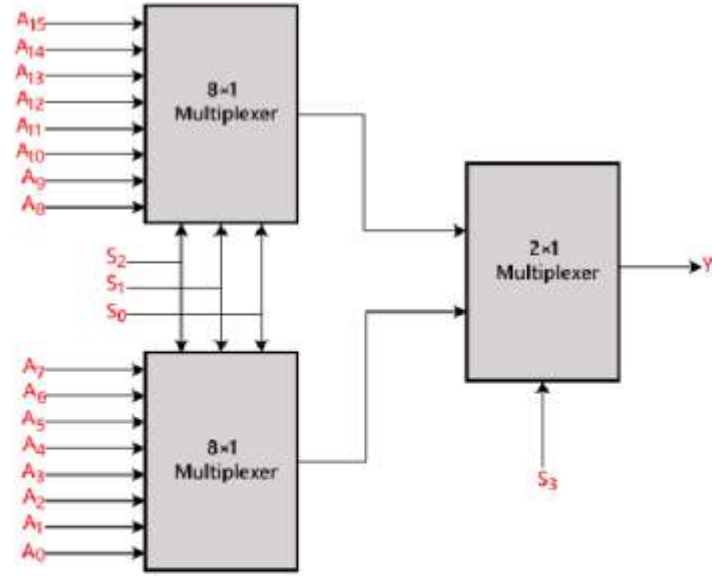
आकृती 3.37-16 ते 1 मल्टीप्लेक्स

### मल्टिप्लेक्सर्सचे कॅस्केडिंग (MUX Tree)

8:1 आणि 2:1 मल्टिप्लेक्सर वापरून 16:1 मल्टिप्लेक्सर-

लोअर ऑर्डर मल्टीप्लेक्सर वापरून 16:1 मल्टीप्लेक्सर तयार करू शकतो. 8:1 मल्टिप्लेक्सर तयार करण्यासाठी, दोन 8:1 मल्टिप्लेक्सर आणि एक 2:1 मल्टिप्लेक्सर आवश्यक आहे. 8:1 मल्टिप्लेक्सरमध्ये 3 निवड रेखा, 4 इनपुट आणि 1 आउटपुट आहे. 2:1 मल्टिप्लेक्सरमध्ये फक्त 1 निवड ओळ आहे.

16 डेटा इनपुट मिळविण्यासाठी, दोन 8 : 1 मल्टिप्लेक्सर्सची आवश्यकता आहे. 8:1 मल्टिप्लेक्सर एक आउटपुट तयार करतो. म्हणून, अंतिम आउटपुट मिळविण्यासाठी, 2:1 मल्टिप्लेक्सरची आवश्यकता आहे. 8:1 आणि 2:1 मल्टिप्लेक्सर वापरून 16:1 मल्टिप्लेक्सरचा ब्लॉक डायग्राम खाली दिला आहे.



आकृती 3.38- MUX- टी

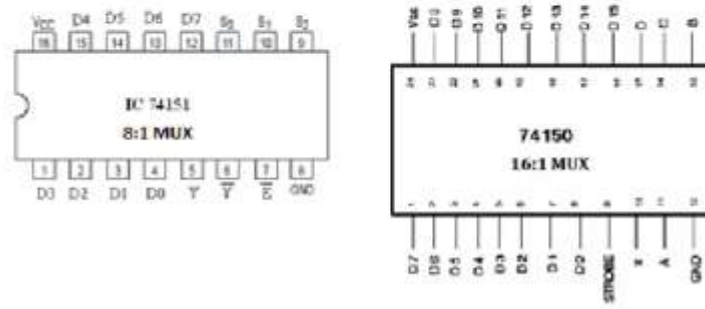
मल्टीप्लेक्सरचे फायदे-

1. बँडविड्थ प्रभावीपणे वापरली जाते.
2. एकाधिक सिग्नल प्रसारित करण्यासाठी एकच लाईन वापरली जाते.

मल्टीप्लेक्सरचे तोटे-

1. मल्टिप्लेक्सर्स वापरून अॅनालॉग किंवा डिजिटल ट्रान्समिशनमध्ये, ट्रान्सपोर्ट सिग्नलचे सिंक्रोनाइझेशन अवघड होते.

मल्टीप्लेक्सर आयसी:-



आकृती 3.39- मल्टीप्लेक्सर आयसी

उपयोग

1. टेलिफोन नेटवर्क, सॅटेलाइट सिस्टीम, टेलिमेट्री यांसारख्या कम्युनिकेशन सिस्टीममध्ये मल्टीप्लेक्सर्सचा वापर केला जातो.
2. रेडिओ आणि टेलिव्हिजन सिग्नलचे प्रसारण मल्टीप्लेक्सशिवाय अशक्य होते.
3. कॉम्प्युटरमधील डेटा राउटिंगमध्ये मल्टीप्लेक्सर देखील वापरला जातो.
4. मल्टीप्लेक्सरचा वापर स्विच सेटिंग कम्प्रेटर आणि फंक्शन जनरेटर म्हणून केला जातो.

### 3.6.2 डी-मल्टीप्लेक्सर(DeMUX)-

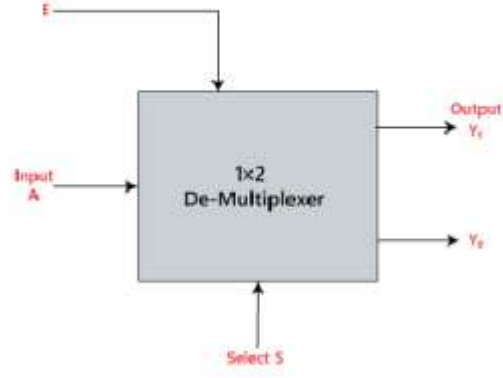
डी-मल्टीप्लेक्सर हे एक कॉम्बिनेशनल सर्किट आहे ज्यामध्ये फक्त 1 इनपुट लाइन आणि  $2^N$  आउटपुट लाइन आहेत. फक्त, मल्टीप्लेक्सर एक सिंगल-इनपुट आणि मल्टी-आउटपुट कॉम्बिनेशनल सर्किट आहे. माहिती सिंगल इनपुट लाईन्समधून प्राप्त होते आणि आउटपुट लाईन्सवर निर्देशित केली जाते. निवड ओर्लीच्या मूल्यांच्या आधारावर, इनपुट यापैकी एका आउटपुटशी कनेक्ट केले जाईल. डी-मल्टीप्लेक्सर मल्टीप्लेक्सरच्या विरुद्ध आहे.

एनकोडर आणि डीकोडरच्या विपरीत, N निवड रेषा आणि  $2^N$  आउटपुट आहेत. तर, इनपुटचे एकूण  $2^N$  संभाव्य संयोजन आहे. डी-मल्टीप्लेक्सरला डी-मक्स देखील मानले जाते.

डी-मल्टीप्लेक्सरचे विविध प्रकार आहेत जे खालीलप्रमाणे आहेत:

1:2 डी-मल्टीप्लेक्सर:

1 : 2 डी-मल्टीप्लेक्सरमध्ये, फक्त दोन आउटपुट आहेत, म्हणजे, Y0, आणि Y1, 1 निवड रेषा, म्हणजे, S0, आणि सिंगल इनपुट, म्हणजे, A. निवड मूल्यांच्या आधारावर, इनपुट असेल आउटपुटपैकी एकाशी कनेक्ट केलेले. 1x2 मल्टीप्लेक्सरचे ब्लॉक डायग्राम आणि सत्य सारणी खाली दिलेली आहे.



आकृती 3.40-1:2 डी-मल्टीप्लेक्सर

तक्ता क्रमांक 3.16-सत्य सारणी

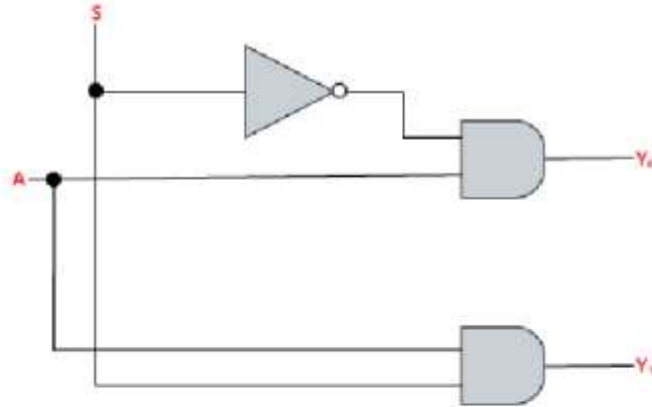
INPUTS	Output	
$S_0$	$Y_1$	$Y_0$
0	0	A
1	A	0

लॉजिकल एक्सप्रेसन

$$Y_0 = S_0' \cdot A$$

$$Y_1 = S_0 \cdot A$$

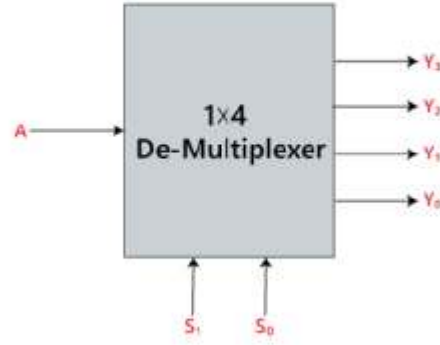
वरील एक्सप्रेसनचे लॉजिकल सर्किट आकृती 3.41 मध्ये दिले आहे:



आकृती 3.41-1x2 डी-मल्टीप्लेक्सर

1:4 डी-मल्टीप्लेक्सर:

1 : 4 डी-मल्टीप्लेक्सरमध्ये, एकूण चार आउटपुट आहेत, म्हणजे,  $Y_0$ ,  $Y_1$ ,  $Y_2$ , आणि  $Y_3$ , 2 निवड रेषा, म्हणजे,  $S_0$  आणि  $S_1$  आणि एकल इनपुट, म्हणजे, A, च्या संयोजनाच्या आधारावर इनपुट जे सिलेक्शन लाईन्स  $S_0$  आणि  $S_1$  वर उपस्थित आहेत, इनपुट आउटपुटपैकी एकाशी कनेक्ट केले जातील. 1:4 मल्टीप्लेक्सरचे ब्लॉक डायग्राम आणि सत्य सारणी खाली दिलेली आहे.



आकृती 3.42 - 1:4 डी-मल्टीप्लेक्सर

तक्ता क्रमांक 3.17- सत्य सारणी

INPUTS		Output			
$S_1$	$S_0$	$Y_3$	$Y_2$	$Y_1$	$Y_0$
0	0	0	0	0	A
0	1	0	0	A	0
1	0	0	A	0	0
1	1	A	0	0	0

लॉजिकल एक्सप्रेशन

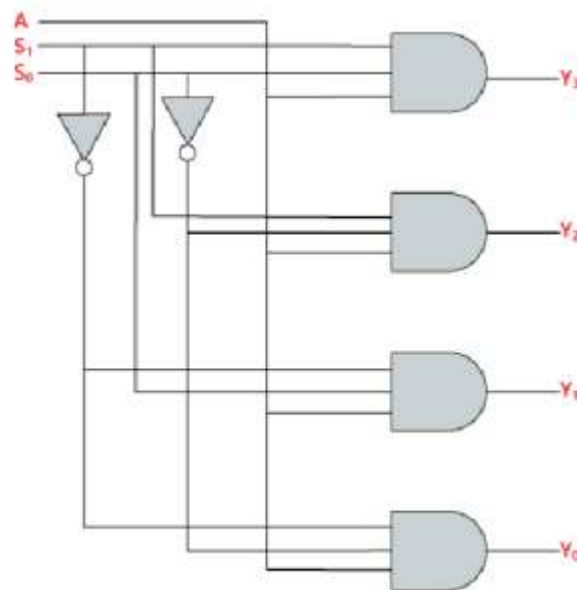
$$Y_0 = S_1' S_0' A$$

$$Y_1 = S_1' S_0 A$$

$$Y_2 = S_1 S_0' A$$

$$Y_3 = S_1 S_0 A$$

वरील एक्सप्रेशनचे लॉजिकल सर्किट आकृती 3.43 मध्ये दिले आहे:

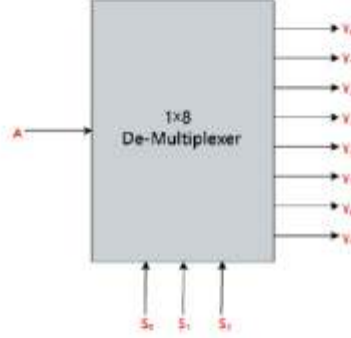


आकृती 3.43-1:4 डी-मल्टीप्लेक्सर



### 1:8 डी-मल्टीप्लेक्सर-

1 : 8 डी-मल्टीप्लेक्सरमध्ये, एकूण आठ आउटपुट आहेत, म्हणजे, Y0, Y1, Y2, Y3, Y4, Y5, Y6, आणि Y7, 3 निवड ओळी, म्हणजे, S0, S1 आणि S2 आणि एकल इनपुट, म्हणजे, A. सिलेक्शन लाईन्स S0, S1 आणि S2 वर उपस्थित असलेल्या इनपुटच्या संयोजनाच्या आधारावर, इनपुट यापैकी एका आउटपुटशी कनेक्ट केले जाईल. ब्लॉक डायग्राम आणि 1:8 डी-मल्टीप्लेक्सरचे सत्य सारणी खाली दिलेली आहे.



आकृती 3.4-1:8 डी-मल्टीप्लेक्सर

तक्ता क्रमांक 3.18-सत्य सारणी

INPUTS			Output							
S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>	Y <sub>7</sub>	Y <sub>6</sub>	Y <sub>5</sub>	Y <sub>4</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>
0	0	0	0	0	0	0	0	0	0	A
0	0	1	0	0	0	0	0	0	A	0
0	1	0	0	0	0	0	0	A	0	0
0	1	1	0	0	0	0	A	0	0	0
1	0	0	0	0	0	A	0	0	0	0
1	0	1	0	0	A	0	0	0	0	0
1	1	0	0	A	0	0	0	0	0	0
1	1	1	A	0	0	0	0	0	0	0

लॉजिकल एक्सप्रेशन

$$Y_0 = S_0' \cdot S_1' \cdot S_2' \cdot A$$

$$Y_1 = S_0 \cdot S_1' \cdot S_2' \cdot A$$

$$Y_2 = S_0' \cdot S_1 \cdot S_2' \cdot A$$

$$Y_3 = S_0 \cdot S_1 \cdot S_2' \cdot A$$

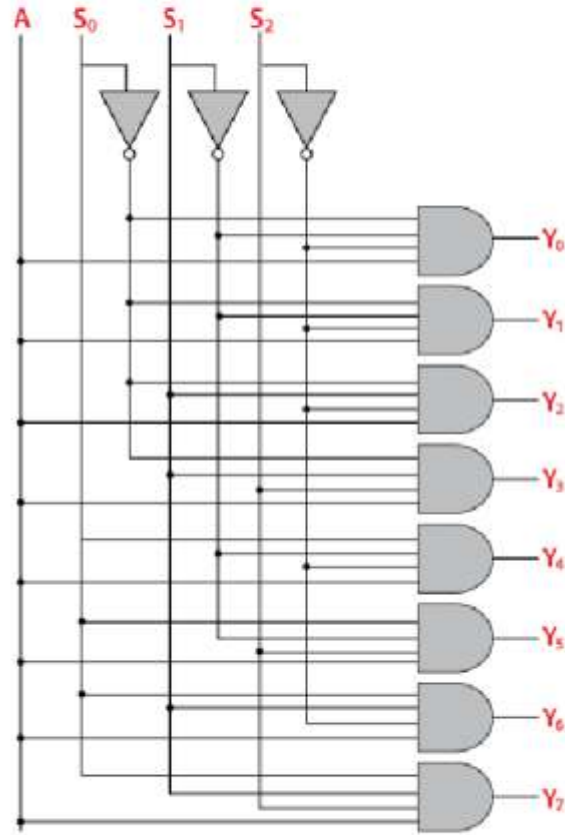
$$Y_4 = S_0' \cdot S_1' \cdot S_2 \cdot A$$

$$Y_5 = S_0 \cdot S_1' \cdot S_2 \cdot A$$

$$Y_6 = S_0' \cdot S_1 \cdot S_2 \cdot A$$

$$Y_7 = S_0 \cdot S_1 \cdot S_2 \cdot A$$

वरील एक्सप्रेशनचे लॉजिकल सर्किट आकृती 3.45 मध्ये दिले आहे:



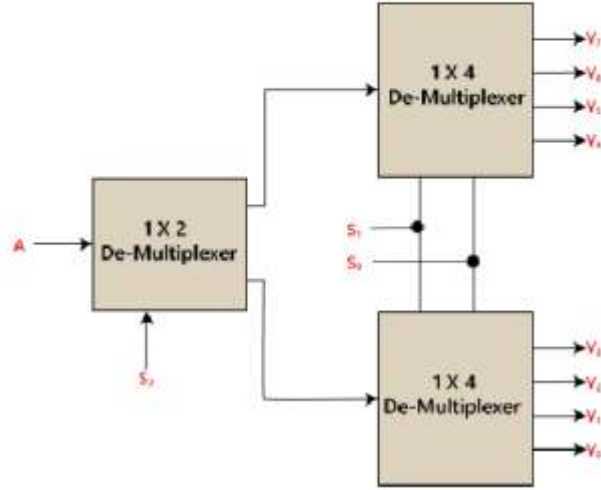
आकृती 3.45-1:8 डी-मल्टीप्लेक्सर

डी मक्स ट्री (DeMUX tree)-

1:8 डी-मल्टीप्लेक्सर 1:4 आणि 1:2 डी-मल्टीप्लेक्सर वापरून-

लोअर ऑर्डर डी-मल्टीप्लेक्सर वापरून 1:8 डी-मल्टीप्लेक्सर तयार करू शकतो. 1×8 डी-मल्टीप्लेक्सर तयार करण्यासाठी, दोन 1:4 डी-मल्टीप्लेक्सर आणि एक 1:2 डी-मल्टीप्लेक्सरची गरज आहे. 1:4 मल्टीप्लेक्सरमध्ये 2 निवड रेखा, 4 आउटपुट आणि 1 इनपुट आहे. 1:2 डी-मल्टीप्लेक्सरमध्ये फक्त 1 निवड ओळ आहे.

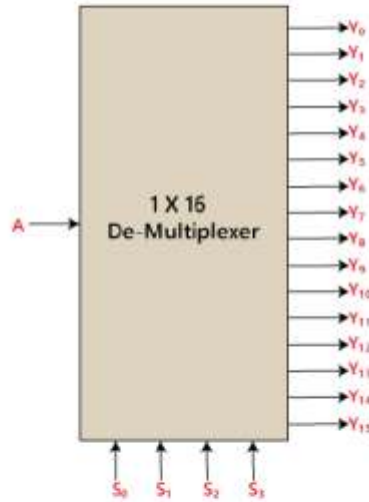
8 डेटा आउटपुट मिळविण्यासाठी, दोन 1:4 डी-मल्टीप्लेक्सरची आवश्यकता आहे. 1:2 डी-मल्टीप्लेक्सर दोन आउटपुट तयार करतो. तर, अंतिम आउटपुट मिळविण्यासाठी, 1:2 डी-मल्टीप्लेक्सरचे आउटपुट 1:4 डी-मल्टीप्लेक्सर दोन्हीचे इनपुट म्हणून पास करावे लागतील. 1:4 आणि 1:2 डी-मल्टीप्लेक्सर वापरून 1×8 डी-मल्टीप्लेक्सरचा ब्लॉक डायग्राम खाली दिला आहे.



आकृती 3.46-1:8 डी-मल्टीप्लेक्सर

### 1 : 16 डी-मल्टीप्लेक्सर

1:16 डी-मल्टीप्लेक्समध्ये, एकूण 16 आउटपुट आहेत, म्हणजे,  $Y_0, Y_1, \dots, Y_{15}$ , 4 निवड ओळी, म्हणजे,  $S_0, S_1, S_2$ , आणि  $S_3$  आणि एक इनपुट, म्हणजे,  $A$ . आधारावर  $S_0, S_1$  आणि  $S_2$  या निवड ओळींवर उपस्थित असलेल्या इनपुटच्या संयोजनापैकी, इनपुट यापैकी एका आउटपुटशी कनेक्ट केले जाईल. 1:16 डी-मल्टीप्लेक्सचे ब्लॉक डायग्राम आणि सत्य सारणी खाली दिलेली आहे.



आकृती 3.47-1:16 डी-मल्टीप्लेक्सर

तक्ता क्रमांक 3.19-सत्य सारणी

INPUTS				OUTPUTS															
S <sub>3</sub>	S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>	Y <sub>15</sub>	Y <sub>14</sub>	Y <sub>13</sub>	Y <sub>12</sub>	Y <sub>11</sub>	Y <sub>10</sub>	Y <sub>9</sub>	Y <sub>8</sub>	Y <sub>7</sub>	Y <sub>6</sub>	Y <sub>5</sub>	Y <sub>4</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	A	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	A	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0	0	0	A	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	A	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	A	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	A	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	A	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	A	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	A	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	A	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	A	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	A	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

लॉजिकल एक्सप्रेशन

$$Y_0 = A.S_0'.S_1'.S_2'.S_3'$$

$$Y_1 = A.S_0'.S_1'.S_2'.S_3$$

$$Y_2 = A.S_0'.S_1'.S_2.S_3'$$

$$Y_3 = A.S_0'.S_1'.S_2.S_3$$

$$Y_4 = A.S_0'.S_1.S_2'.S_3'$$

$$Y_5 = A.S_0'.S_1.S_2'.S_3$$

$$Y_6 = A.S_0'.S_1.S_2.S_3'$$

$$Y_7 = A.S_0'.S_1.S_2.S_3$$

$$Y_8 = A.S_0.S_1'.S_2'.S_3'$$

$$Y_9 = A.S_0.S_1'.S_2'.S_3$$

$$Y_{10} = A.S_0.S_1'.S_2.S_3'$$

$$Y_{11} = A.S_0.S_1'.S_2.S_3$$

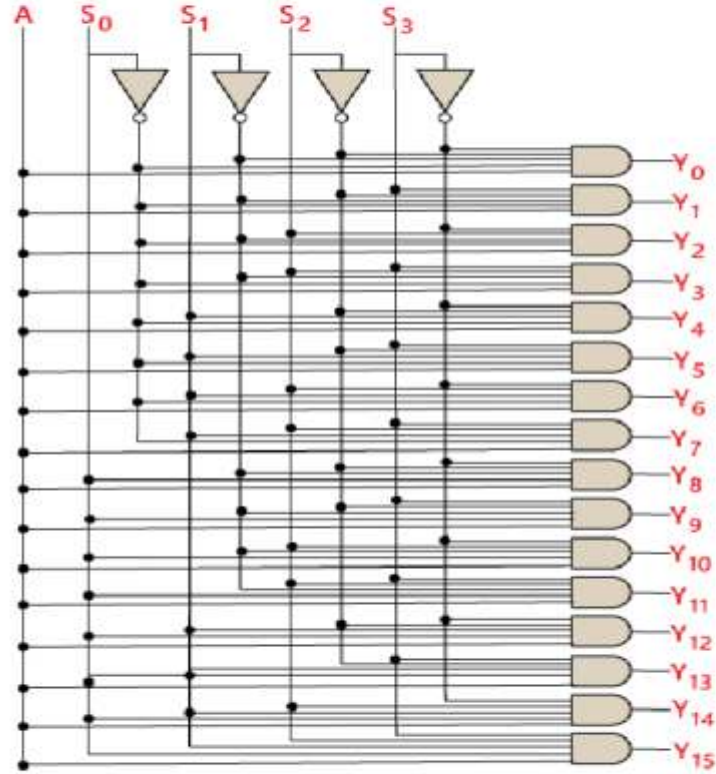
$$Y_{12} = A.S_0.S_1.S_2'.S_3'$$

$$Y_{13} = A.S_0.S_1.S_2'.S_3$$

$$Y_{14} = A.S_0.S_1.S_2.S_3'$$

$$Y_{15} = A.S_0.S_1.S_2'.S_3$$

वरील एक्सप्लेनचे लॉजिकल सर्किट आकृती 3.48 मध्ये दिले आहे:



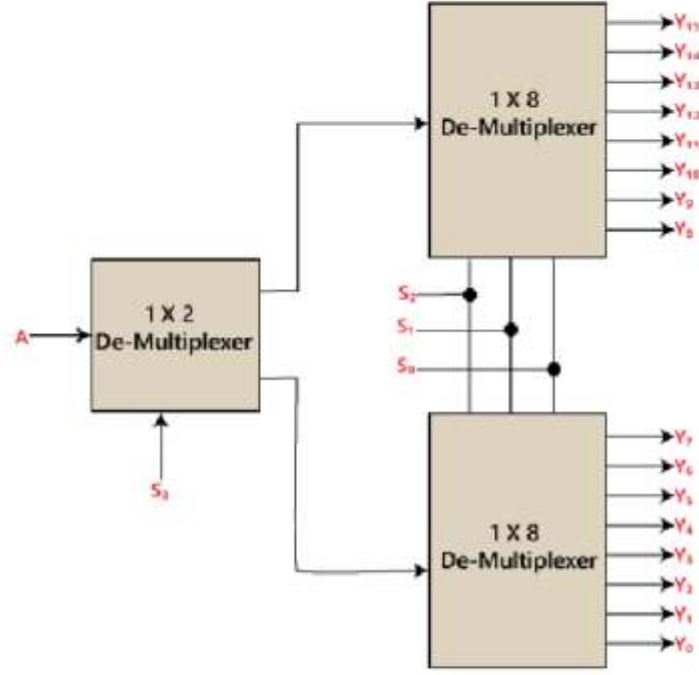
आकृती 3.48-1:16 डी-मल्टीप्लेक्सर

डी मक्स ट्री(DeMUX Tree)-

1:16 डी-मल्टीप्लेक्सर 1:8 आणि 1:2 डी-मल्टीप्लेक्सर वापरून-

लोअर ऑर्डर डी-मल्टीप्लेक्सर वापरून 1:16 डी-मल्टीप्लेक्सर तयार करू शकतो. 1:16 डी-मल्टीप्लेक्सर तयार करण्यासाठी, दोन 1:8 डी-मल्टीप्लेक्सर आणि एक 1:2 डी-मल्टीप्लेक्सरची गरज आहे. 1:8 मल्टीप्लेक्सरमध्ये 3 निवड रेषा, 1 इनपुट आणि 8 आउटपुट आहेत. 1:2 डी-मल्टीप्लेक्सरमध्ये फक्त 1 निवड ओळ आहे.

16 डेटा आउटपुट मिळविण्यासाठी, दोन 1:8 डी-मल्टीप्लेक्सरची आवश्यकता आहे. 1:8 डी-मल्टीप्लेक्सर आठ आउटपुट तयार करतो. तर, अंतिम आउटपुट मिळविण्यासाठी, एका इनपुटमधून दोन आउटपुट तयार करण्यासाठी 1:2 डी-मल्टीप्लेक्सरची आवश्यकता आहे. मग हे आउटपुट दोन्ही डी-मल्टीप्लेक्सरमध्ये इनपुट म्हणून पास करून. 1:8 आणि 1:2 डी-मल्टीप्लेक्सर वापरून 1:16 डी-मल्टीप्लेक्सरचा ब्लॉक डायग्राम खाली दिला आहे.



आकृती 3.49-1x16 डी-मल्टीप्लेक्सर

डिमल्टीप्लेक्सर्सचे फायदे:-

1. ऑडिओ/व्हिडिओ सिग्नल प्रसारित करण्यासाठी मल्टीप्लेक्सर्स आणि डिमल्टीप्लेक्सर्सचे संयोजन आवश्यक आहे.
2. ते बँकिंग क्षेत्रांसारख्या सुरक्षा प्रणालींमध्ये डीकोडर म्हणून देखील वापरले जातात.
3. डी मक्स चे मक्स सह संयोजन संवाद प्रणालीची कार्यक्षमता वाढवते.

डिमल्टीप्लेक्सर्सचे तोटे:-

1. बँडविड्थचा अपव्यय होऊ शकतो.
2. सिग्नलच्या सिंक्रोनाइझेशनमुळे विलंब होऊ शकतो.

उपयोग -

डिमल्टीप्लेक्सरचा वापर अनेकांपैकी एक सिग्नल निवडण्यासाठी किंवा सक्षम करण्यासाठी केला जात असल्याने, ते मायक्रोप्रोसेसर किंवा संगणक नियंत्रण प्रणालींमध्ये मोठ्या प्रमाणावर वापरले जातात जसे की

- एंड्रेस अवलंबून असते, मेमरी चिप्सच्या विविध लाईन सक्षम करते
- सिंक्रोनस डेटा ट्रान्समिशन सिस्टम
- बुलियन फंक्शन अंमलबजावणी
- डेटा संपादन प्रणाली
- कॉम्बिनेशनल सर्किट डिझाइन
- स्वयंचलित चाचणी उपकरणे प्रणाली



### 3.7 बफर आयसी –

बफर(Buffer):

बफरमध्ये फक्त एकच इनपुट आणि एकच आउटपुट असतो

#### 3.7.1 ट्राय स्टेट बफर (Tri-State):

हे 3-स्टेट लॉजिक आउटपुट पोर्टला 0 आणि 1 लॉजिक स्तरांव्यतिरिक्त, सर्किटमधून आउटपुट प्रभावोपणे काढून टाकून, उच्च प्रतिबाधा(Hi-Z) स्थिती गृहीत करण्यास अनुमती देते.

ट्राय-स्टेट बफर हा बफर सर्किटचा प्रकार आहे ज्याचा वापर लॉजिक सिग्नलला त्याच्या इनपुटपासून आउटपुटपर्यंत नियंत्रित करण्यासाठी केला जाऊ शकतो. ट्राय-स्टेट बफर हे एक कॉम्बिनेशनल डिव्हाईस आहे ज्याचे आउटपुट बाह्य "नियंत्रण" किंवा "सक्षम" (EN) सिग्नल इनपुटद्वारे इलेक्ट्रॉनिक पद्धतीने "चालू" किंवा "बंद" केले जाऊ शकते जे त्यांना बस(BUS)-ओरिएंटेड सिस्टममध्ये वापरण्याची परवानगी देते.

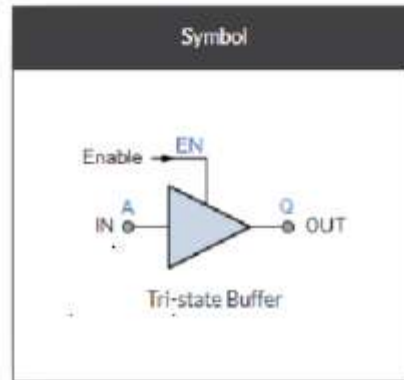
त्यांच्या नावाप्रमाणे, ट्राय-स्टेट बफरसाठी "Q" वरील आउटपुट तीन संभाव्य अवस्थांपैकी एक घेऊ शकते, लॉजिक "0", लॉजिक "1", आणि उच्च प्रतिबाधा(Hi-Z), म्हणजेच एक ओपन सर्किट, स्टॅंडर्ड "0" आणि "1" स्थितींपेक्षा.

बफर सक्षम किंवा नियंत्रण सिग्नल एकतर लॉजिक "0" किंवा लॉजिक "1" लेव्हल सिग्नल असू शकतात ज्यात आउटपुट इनव्हर्टिंग आणि नॉन-इनव्हर्टिंग आहे कारण डिजिटल सिग्नल त्यातून जातो.

♣ सक्षम सिग्नल उच्च असल्यास, लॉजिक "1", बफर गेटचा इनपुट सिग्नल थेट त्याच्या आउटपुटवर जातो.

♣ सक्षम सिग्नल कमी असल्यास, लॉजिक "0" असल्यास, बफर गेटचे आउटपुट ओपन सर्किट सारखे कार्य करते, म्हणजेच उच्च प्रतिबाधा, (Hi-Z).

एँक्टिव्ह "हाय" ट्राय स्टेट बफर (Active "HIGH" Tri-state Buffer)-



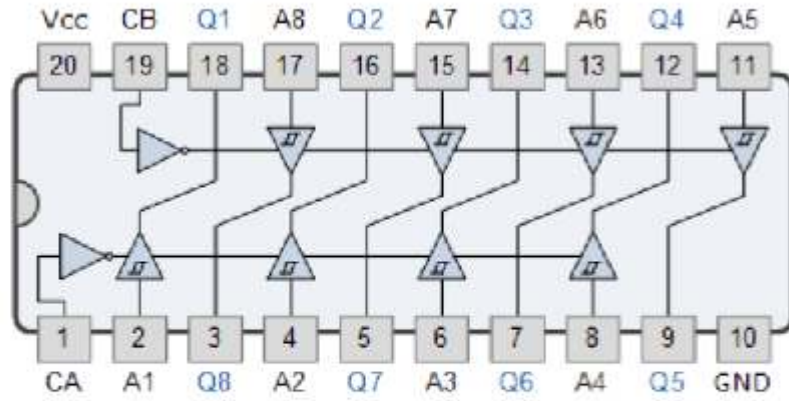
आकृती 3.50- ट्राय स्टेट बफर

तक्ता क्रमांक 3.20-सत्य सारणी

Truth Table		
Enable	IN	OUT
0	0	Hi-Z
0	1	Hi-Z
1	0	0
1	1	1

• ट्राय-स्टेट बफरचे चिन्ह वरील स्टँडर्ड बफर चिन्हासारखेच आहे परंतु नियंत्रण कार्य सक्षम/अक्षम करण्याचे प्रतिनिधित्व करणारे दुसरे इनपुट जोडावे. जेव्हा सक्षम (EN) इनपुट लॉजिक लेव्हल "1" (पॉसिटिव्ह लॉजिक साठी) वर असते, तेव्हा ते इनपुट सिग्नल, A ला थेट Q वर आउटपुटवर जाण्यास अनुमती देणारे सामान्य बफर म्हणून कार्य करते. ते लॉजिक "0" असो. किंवा तर्क "1". जेव्हा सक्षम इनपुट लॉजिक "0" वर असते, तेव्हा ट्राय-स्टेट बफर त्याच्या तिसऱ्या स्थितीत सक्रिय होतो आणि ओपन सर्किट स्थिती निर्माण करून त्याचे आउटपुट अक्षम किंवा "बंद" करते. ही तिसरी अट लॉजिक "1" उच्च (High) किंवा लॉजिक "0" कमी (Low) मध्ये नाही, परंतु त्याऐवजी एक आउटपुट स्थिती देते जी खूप उच्च प्रतिबाधावर (High-Z) आहे, अधिक सामान्यतः Hi-Z असे लिहिले जाते. अशा प्रकारे ट्राय-स्टेट बफरमध्ये दोन लॉजिक स्टेट इनपुट असतात, "0" किंवा "1" परंतु तीन वेगवेगळ्या आउटपुट स्टेट्स तयार करू शकतात, "0", "1" किंवा "Hi-Z" त्यामुळे त्याला "ट्राय" किंवा "3-स्टेट" डिव्हाइस असे म्हणतात. लक्षात घ्या की ही तिसरी अवस्था तर्कशास्त्र पातळी "0" किंवा "1" च्या बरोबरीची नाही, परंतु तिचे आउटपुट इलेक्ट्रिकली डिस्कनेक्ट झाल्यामुळे उच्च प्रतिबाधा स्थिती आहे. ट्राय-स्टेट बफर्स एकात्मिक स्वरूपात क्वाड (QUAD), हेक्स (HEX) किंवा ऑक्टल (OCTAL) बफर/ड्रायव्हर्स जसे की TTL 74LS244 म्हणून उपलब्ध आहेत.

74LS244 ऑक्टल ट्राय-स्टेट बफर



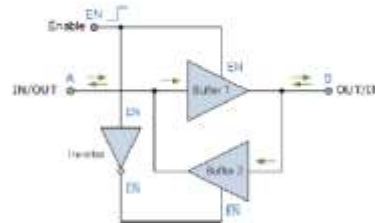
आकृती 3.51-74LS244 ऑक्टल ट्राय-स्टेट बफर

लक्षात घ्या की आठ बफर चारच्या दोन गटांमध्ये कॉन्फिगर केले आहेत आणि पहिला गट (A1 ते A4) सक्षम इनपुट, CA द्वारे नियंत्रित केला जात आहे आणि दुसरा गट (A5 ते A8) सक्षम इनपुट, CB द्वारे नियंत्रित केला जात आहे.

### 3.7.2 द्वि-दिशात्मक बफर (Bidirectional Buffer) नियंत्रण-

द्वि-दिशात्मक बफर किंवा ट्रान्झिस्टर सर्किट म्हटल्या जाणाऱ्या उत्पादनासाठी ट्राय-स्टेट बफर "बॅक-टू-बॅक" (उलटा समांतर) जोडणे देखील शक्य आहे. अतिरिक्त इन्व्हर्टर वापरून, एक ट्राय-स्टेट बफर "सक्रिय-उच्च (Active High) बफर" म्हणून आहे, तर दुसरा "सक्रिय-निम्न (Active Low) बफर" म्हणून कार्य करतो, दाखवल्याप्रमाणे.

एकाच बसवर अनेक ट्राय स्टेट बफर-



आकृती 3.52- द्वि-दिशात्मक बफर

येथे, दोन ट्राय-स्टेट बफर समांतरपणे जोडलेले आहेत परंतु सक्षम नियंत्रण इनपुटसह "A" ते "B" पर्यंत उलट आहेत, EN अधिक दिशात्मक नियंत्रण सिग्नलसारखे कार्य करते.

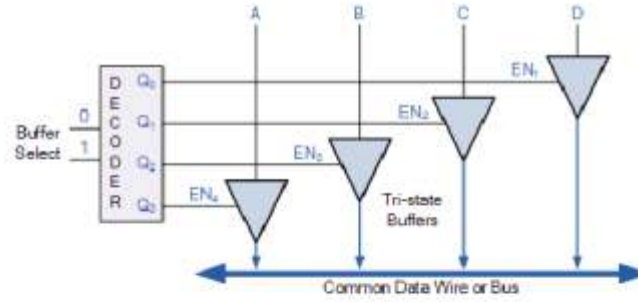
- तर या सोप्या उदाहरणात, जेव्हा सक्षम इनपुट उच्च असेल, (EN "1") डेटाला बफर 1 द्वारे A ते B मध्ये जाण्याची परवानगी आहे आणि जेव्हा सक्षम इनपुट कमी असेल, (EN "0") डेटा बफर 2 द्वारे B पासून A मध्ये जातो.

- अशा प्रकारे सक्षम इनपुट "EN" दिशा नियंत्रण म्हणून कार्य करते ज्यामुळे डेटा या नियंत्रण इनपुटच्या तर्क स्थितीनुसार दोन्ही दिशेने प्रवाहित होऊ शकतो. या प्रकारच्या ऍप्लिकेशनमध्ये TTL 74LS245 सारख्या द्वि-दिशात्मक स्विचिंग क्षमतेसह त्रि-राज्य बफर

ट्राय-स्टेट बफर नियंत्रण

- ट्राय-स्टेट बफर एकाधिक डिवाइसेसना एक सामान्य आउटपुट वायर किंवा बस सामायिक करण्याची परवानगी देऊ शकतात फक्त एक ट्राय-स्टेट डिवाइस कोणत्याही वेळी वायर बस चालवते तर इतर सर्व बफर त्यांच्या हाय-झेड स्थितीत राहतात. सर्किट आकृती 3.53 विचारात घ्या.

एकाच बसवर अनेक त ट्राय-स्टेट बफर



आकृती 3.53- ट्राय-स्टेट बफर नियंत्रण

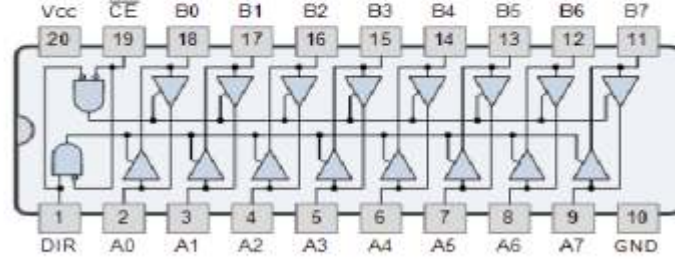
- प्रत्येक ट्राय-स्टेट बफरमधील आउटपुट एका सामान्य वायर बसशी जोडलेले असतात परंतु त्यांचे सक्षम इनपुट बायनरी डीकोडरशी जोडलेले असतात. डीकोडर त्याच्या सक्षम सिग्नलमुळे, कोणत्याही वेळी फक्त एक ट्राय-स्टेट बफर सक्रिय असेल याची हमी देतो. हे सक्रिय बफरचा डेटा थेट सामान्य बसमध्ये पास करण्यास अनुमती देते, तर इतर गैर-सक्षम बफरचे आउटपुट प्रभावीपणे डिस्कनेक्ट केलेले असतात आणि त्यांच्या उच्च-प्रतिबाधा स्थितीत असतात. अशा प्रकारे कोणता बफर सामान्य रेषेशी जोडलेला आहे हे डीकोडर निवडलेल्या इनपुटच्या बायनरी मूल्यावर अवलंबून असेल.

- म्हणून, कोणत्याही वेळी एकापेक्षा जास्त ट्राय-स्टेट बफर "सक्रिय स्थितीत" असू शकत नाहीत. तुमच्या लक्षात आले असेल की वरील एकाच आउटपुट लाइनला जोडलेल्या वेगवेगळ्या डेटा इनपुटचे संभाव्य संयोजन 4:1 लाइन मल्टिप्लेक्सरसारखे आहे आणि तुमचे म्हणणे बरोबर आहे, मल्टीप्लेक्सर सर्किट्स ट्राय-स्टेट बफर वापरून सहजपणे तयार केले जाऊ शकतात.

- कोणताही ट्राय-स्टेट बफर घटक वापरलेल्या ट्राय-स्टेट बफरवर अवलंबून, त्यांच्या सक्षम (EN) इनपुटला थेट +Vcc किंवा Gnd शी कनेक्ट करून सामान्य डिजिटल बफरमध्ये सहजपणे रूपांतरित केले जाऊ शकते. अशा प्रकारे, आउटपुट कायमस्वरूपी सक्षम केले जाते त्यामुळे "A" वर उपस्थित असलेले कोणतेही इनपुट सिग्नल बफरमधून थेट "Q" वरील आउटपुटवर जाईल.

- आतापर्यंत पाहिले आहे की, वायर किंवा बस(BUS)मध्ये युनि-डायरेक्शनल पद्धतीने माहिती पाठवण्यासाठी ट्राय-स्टेट बफरचा वापर करू शकतो. परंतु आम्ही त्यांचा वापर दोन्ही दिशांना डेटा पाठवण्यासाठी, म्हणजे डेटा पाठवण्यासाठी आणि सामान्य वायर बसमधून डेटा प्राप्त करण्यासाठी कसा करू शकतो.

### 74LS245 (द्वि-दिशात्मक बफर) Bidirectional Buffer-



आकृती 3.54-74LS245 (द्वि-दिशात्मक बफर)

TTL 74LS245 हा एक ऑक्टल बस ट्रान्सीव्हर (ट्रान्समीटर/रिसीव्हर) आहे जो दोन डेटा बस किंवा इनपुट/आउटपुट यंत्रादरम्यान असिंक्रोन्स द्वि-मार्ग संवादासाठी डिझाइन केलेला आहे. ट्रान्सीव्हर टर्मिनल्स A पासून टर्मिनल B पर्यंत डेटा प्रसारित करण्यास किंवा दिशा-नियंत्रण (DIR) इनपुट, (पिन 1) वर तर्क पातळीवर अवलंबून रिव्हर्स करण्यास परवानगी देतो.

- उदाहरणार्थ, जर दिशा-नियंत्रण इनपुट लॉजिक लेव्हल “1” वर जास्त असेल, तर डेटा टर्मिनल A मधून टर्मिनल B मध्ये जाईल. जर डायरेक्शन-कंट्रोल इनपुट लॉजिक लेव्हल “0” वर कमी असेल, तर डेटा पास होईल टर्मिनल B पासून टर्मिनल A पर्यंत उलट दिशेने. लॉजिक स्तर “1” वर उच्च (High) धरल्यावर, आउटपुट चिप-सक्षम (CE) इनपुट, (pin 19) डिव्हाइस अक्षम करण्यासाठी वापरले जाऊ शकते जेणेकरून टर्मिनल्स, आणि म्हणून कोणतेही कनेक्ट केलेल्या डेटा बसेस प्रभावीपणे एकमेकांपासून वेगळ्या केल्या जातात.

### स्वयं अध्याय-

- १) संयुक्त तार्किक (लॉजिक) मंडल चे विश्लेषण करा.
- २) लॉजिक गेट वापरून फुल अँडरचा लॉजिक सर्किट डायग्राम काढा आणि एका उदाहरणासह कार्य स्पष्ट करा.
- ३) के-मॅप रिडक्शन तंत्राचा वापर करून, बेरीजसाठी एक्स्प्रेसन शोधा आणि अर्ध्या योजनांचे आउटपुट काढा. फक्त NAND गेट वापरून आउटपुट समीकरणे लक्षात घ्या.
- ४) मलिटप्लेक्सर आवश्यकता स्पष्ट करा.
- ५) 4 : 1 मलिटप्लेक्सर गेट्स वापरून तार्किकआकृतीकाढा , त्याचे कार्य व सत्य सारणी लिहा.
- ५) गेट्स वापरलेल्या 1 :4 डिमलिटप्लेक्सरची तार्किकआकृतीकाढून त्याचे कार्य व सत्य सारणी लिहा.
- ६) डिमलिटप्लेक्सर्स उपयोग लिहा.
- ७) दशमान ते बीसीडी एन्कोडरआकृतीकाढून कार्य व सत्य सारणी लिहा .
- ८) आय. सी. 74147 वापरलेल्या प्राधान्यक्रम एन्कोडर चे कार्य व सत्य सारणी लिहा.
- ९) अँड गेट्स वापरलेल्या बीसीडी ते दशमान डीकोडरचे कार्य तार्किक आकृतीकाढून स्पष्ट करा.
- १०) आय. सी. 7445 कशासाठी वापरतात? डीकोडर आय.सी. ची उदाहरणे लिहा .

**मायक्रो-प्रोजेक्ट- 4** बिट अँडर सर्किट योग्य आय सी वापरून ब्रेड बोर्ड वर तयार करणे , व ते सर्किट वेगवेगळ्या इनपुट्स देऊन तपासून घेणे .



## युनिट-4

### अनुक्रमिक लॉजिक सर्किट्स (Sequential Logic Circuits)

विषय निष्पत्ती (Course Outcome):-

सोपे अनुक्रमिक सर्किट तयार करणे. (Build simple sequential circuits)

युनिट निष्पत्ती (Unit outcomes):-

1. दिलेल्या डिजिटल सर्किट्ससाठी संबंधित ट्रिगरिंग तंत्राचा वापर करणे.  
(Use relevant triggering techniques for the given digital circuits).
2. विशिष्ट प्रकारचे गणक (counter) तयार करण्यासाठी दिलेल्या फ्लिप-फ्लॉपचा वापर करणे.  
(Use the given flip-flop to construct the specific type of counter).
3. सिंक्रोनस काउंटर डिझाइन करण्यासाठी दिलेल्या फ्लिप-फ्लॉपचे उत्तेजना सारणीचा वापर करणे.  
(Use excitation table of the given flip-flop to design synchronous counter).
4. IC7490 वापरून निर्दिष्ट मोड्युलो-N गणक डिझाइन करणे.  
(Design the specified modulo-N counter using IC7490)
5. दिलेल्या फ्लिप-फ्लॉपचा वापर करून रिंग/ट्विस्टेड रिंग काउंटर तयार करणे.  
(Construct ring/twisted ring counter using the given flip-flop)

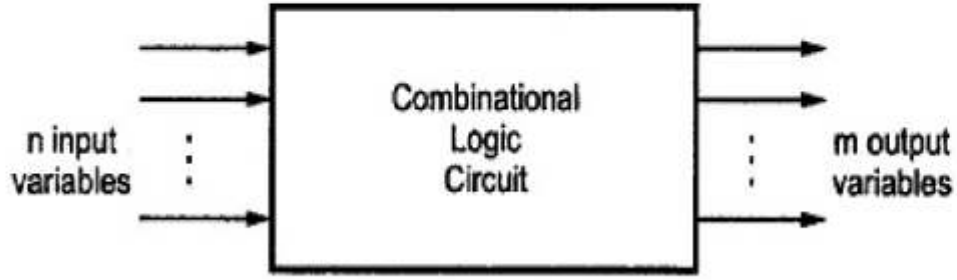
#### 4.1 परिचय:-

बेसिक लॉजिक गेट्स, कॉम्बिनेशनल आणि सिक्वेन्शियल लॉजिक सर्किट्सचे ज्ञान स्वतंत्र गेट्स वापरून, तसेच डिजिटल ICs उपकरणांच्या कार्याचा अर्थ लावण्यासाठी आणि त्यांची देखभाल करण्यास सक्षम करेल. डिजिटल सर्किट आधारित ऍप्लिकेशन्स विकसित करण्यात मदत होईल.

#### 4.2 लॉजिक सर्किट्सचे प्रकार (Types of logic circuits):

##### 4.2.1 कॉम्बिनेशनल लॉजिक सर्किट (Combinational Logic circuit)

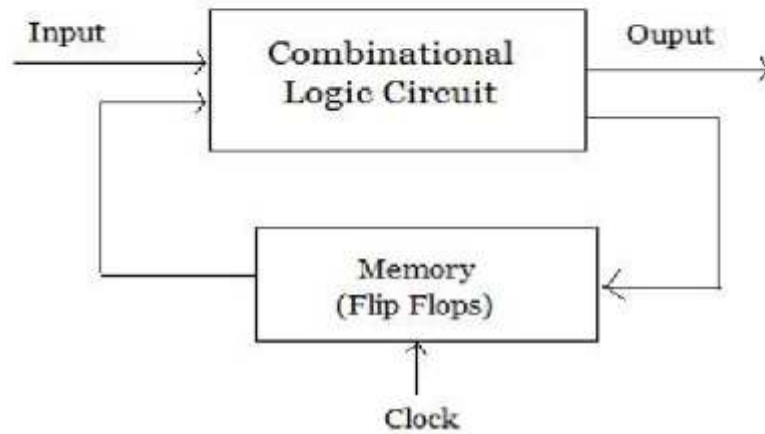
यात लॉजिक गेट्स असतात ज्यांचे आउटपुट कोणत्याही वेळी केवळ इनपुटच्या सध्याच्या संयोजनातून (combination) निर्धारित केले जातात आणि त्यांना कोणतीही मेमरी नसते.



आकृती 4.1 कॉम्बिनेशनल लॉजिक सर्किट

#### 4.2.2 अनुक्रमिक लॉजिक सर्किट (Sequential Logic circuit)

ज्यांचे आउटपुट कोणत्याही वेळी इनपुट आणि मागील आउटपुटच्या सध्याच्या संयोजनातून निर्धारित केले जातात. म्हणजे अनुक्रमिक सर्किट मागील आउटपुटचे मूल्य संचयित करण्यासाठी मेमरी घटक वापरतात



आकृती 4.2 अनुक्रमिक लॉजिक सर्किट

**4.2.3 फ्लिपफ्लॉप (Flip-flop)** - एक बायस्टेबल (bistable) इलेक्ट्रॉनिक सर्किट आहे ज्यामध्ये दोन स्थिर अवस्था असतात आणि 1 बिट माहिती संचयित करू शकणारे मेमरी म्हणून ओळखले जाऊ शकते.

**4.2.3.1 कॉम्बिनेशनल आणि सिक्वेन्शियल सर्किट्समधील फरक** (Comparison between combinational and sequential circuits)



### तक्ता क्र. 4.1

Sr. No.	कॉम्बिनेशनल सर्किट (Combinational circuits)	सिक्वेन्शियल सर्किट (Sequential circuits)
1	हे एक डिजिटल लॉजिक सर्किट आहे ज्याचे आउटपुट कोणत्याही वेळी एकाच वेळी दिलेल्या इनपुटवर अवलंबून असते.	हे एक डिजिटल लॉजिक सर्किट आहे ज्याचे आउटपुट सध्याच्या इनपुटवर तसेच मागील स्थितीवर अवलंबून असते.
2	यात कोणताही मेमरी घटक नाही.	यात किमान एक मेमरी घटक आहे.
3	मेमरीच्या अनुपस्थितीमुळे, हे डिझाइन करणे सोपे आहे.	मेमरीच्या उपस्थितीमुळे डिझाइन करणे कठीण आहे.
4	हे अधिक जलद आहे. कारण सर्व इनपुट एकाच वेळी दिलेल्या इनपुट आहेत.	ते तुलनेने कमी वेगाचे आहे कारण त्यात दुर्यम इनपुट देखील आहेत, जे विलंबानंतर दिले जातात.
5	यात अधिक हार्डवेअरची आवश्यकता आहे.	यात कमी हार्डवेअरची आवश्यकता आहे.

**4.2.4 क्लॉक सिग्नल (Clock signal)** - क्लॉक सिग्नल हा उच्च (high) आणि कमी (low) व्होल्टेजसह एक नियतकालिक (periodic) सिग्नल आहे आणि त्याची ऑनटाइम आणि ऑफटाइम समान असणे आवश्यक नाही. क्लॉक सिग्नल आकृती 4.3 मध्ये दर्शविला आहे.

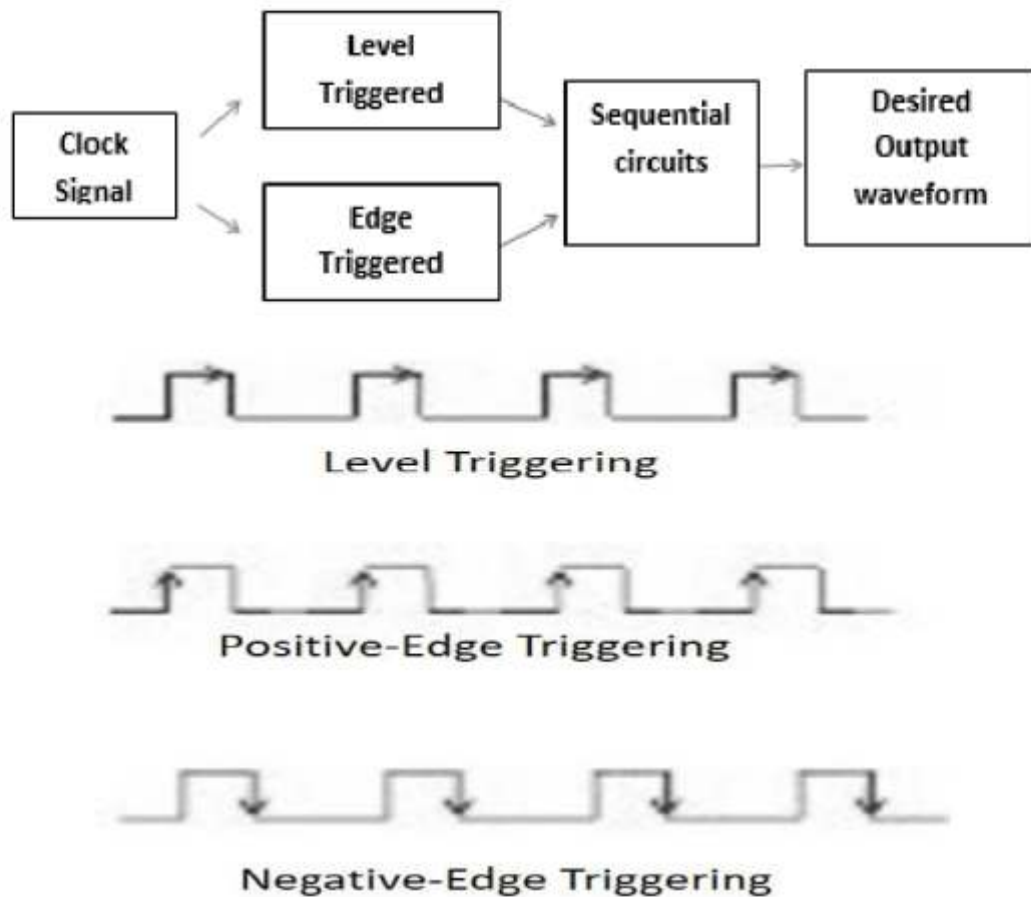


आकृती 4.3

i) सर्किटचे वेगवेगळे भाग ताळमेळ राखण्यासाठी (synchronize), क्लॉक सिग्नलचा वापर केला जातो.

ii) डिजिटल सर्किटचे काही भाग वापरात नसताना प्रभावीपणे बंद करून वीज वाचवण्यासाठी गेटेड (gated clock) क्लॉक सिग्नलचा वापर केला जातो.

#### 4.3 ट्रिगरिंगचे प्रकार (Types of triggering)



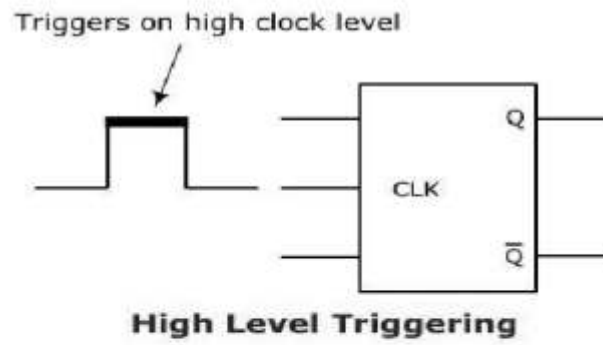
#### आकृती 4.4 ट्रिगरिंगचे प्रकार

**4.3.1 क्लॉक -** क्लॉक सिग्नल उच्च आणि कमी व्होल्टेज असलेली स्क्वेअरवेव्ह, आकृती 4.4 नुसार असते.

**4.3.2 लेव्हल ट्रिगरिंग (Level triggering) -** जेव्हा क्लॉक सिग्नल एकतर उच्च किंवा कमी असतो तेव्हा इनपुट सिग्नलचा नमुना (sample) घेतला जातो.

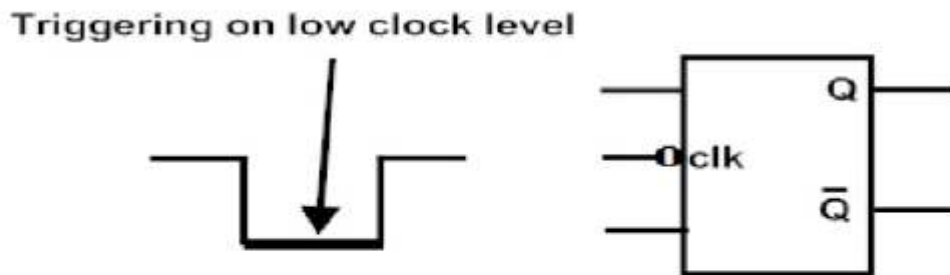
**4.3.3 एज ट्रिगरिंग (edge triggering)** - जर क्लॉकचा पल्स वाढणाऱ्या कडेवर (rising edge) किंवा कमी होणाऱ्या कडेवर (falling edge) ट्रिगर केले जात असेल तर त्याला एज ट्रिगरिंग म्हणतात. इनपुट सिग्नलचा नमुना, क्लॉक सिग्नलच्या rising एज किंवा falling एज वर घेतला जातो.

**4.3.4 उच्च स्तरीय ट्रिगरिंग (high level triggering)** - जेव्हा फ्लिप-फ्लॉपला त्याच्या उच्च स्थितीत प्रतिसाद देणे आवश्यक असते, तेव्हा उच्च स्तर ट्रिगरिंग पद्धत वापरली जाते. हे प्रामुख्याने क्लॉकच्या इनपुटमधून सरळ लीड वरून ओळखले जाते. हे आकृती 4.5 मध्ये दर्शविले आहे.



आकृती 4.5 उच्च स्तरीय ट्रिगरिंग

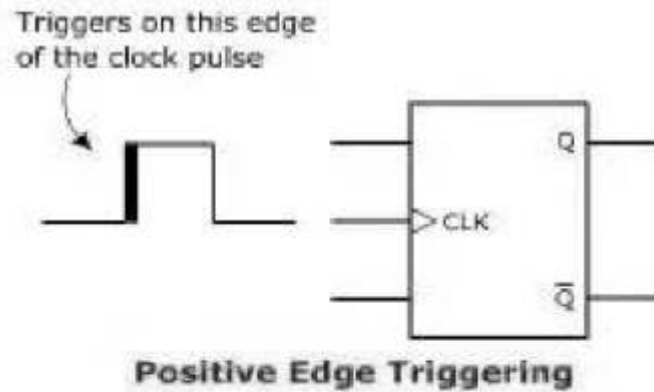
**4.3.5 निम्न पातळी ट्रिगरिंग (low level triggering)** - जेव्हा फ्लिप-फ्लॉपला त्याच्या निम्न स्थितीत प्रतिसाद देणे आवश्यक असते, तेव्हा निम्न स्तर ट्रिगरिंग पद्धत वापरली जाते. हे प्रामुख्याने क्लॉकच्या इनपुटमधून वर्तुळ आकारसह (bubble) लीडवरून ओळखले जाते. हे आकृती 4.6 मध्ये दर्शविले आहे.



आकृती 4.6 निम्न पातळी ट्रिगरिंग

#### 4.3.6 रायजिंग एज ट्रिगरिंग (Positive edge triggering) -

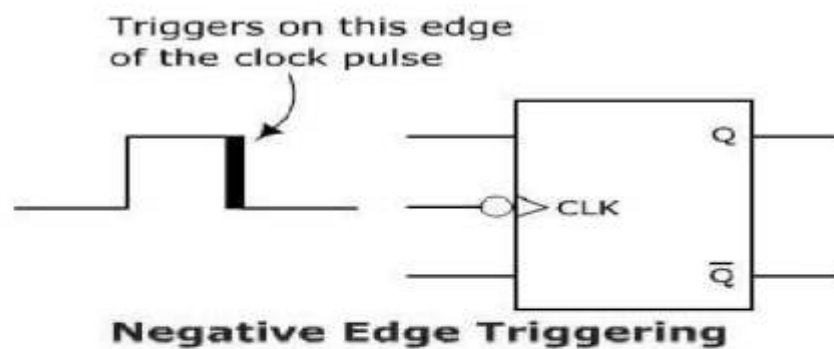
जेव्हा फ्लिप-फ्लॉपला कमी ते जास्त संक्रमण (transition) अवस्थेत प्रतिसाद देणे आवश्यक असते, तेव्हा पॉझिटिव्ह एज ट्रिगरिंग पद्धत वापरली जाते. हे प्रामुख्याने त्रिकोणासह क्लॉक इनपुट लीडवरून ओळखले जाते.



आकृती 4.7 सकारात्मक एज ट्रिगरिंग

#### 4.3.7 फॉलिंग एज (negative edge) ट्रिगरिंग -

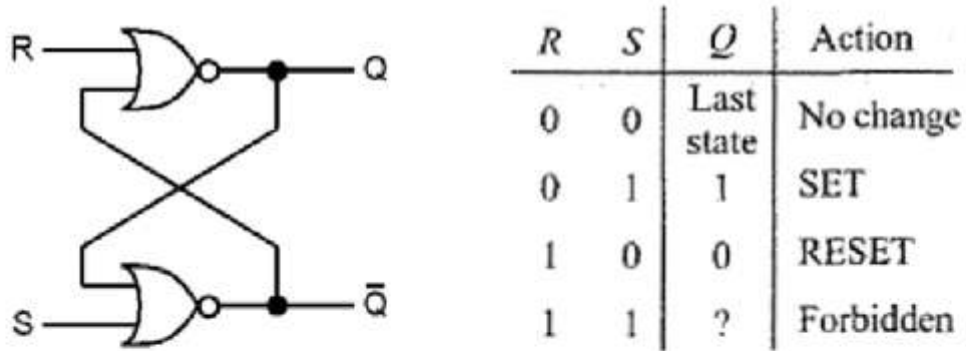
जेव्हा फ्लिप-फ्लॉपला जास्त ते कमी संक्रमण अवस्थेत प्रतिसाद देणे आवश्यक असते, तेव्हा फॉलिंग एज ट्रिगरिंग पद्धत वापरली जाते. हे प्रामुख्याने त्रिकोण व बबलसह क्लॉक इनपुट लीडवरून ओळखले जाते.



आकृती 4.7 फॉलिंग एज ट्रिगरिंग

**4.4.1 फ्लिप-फ्लॉप (Flip-Flop)** - फ्लिप-फ्लॉप एक बायस्टेबल (bistable) इलेक्ट्रॉनिक सर्किट आहे ज्यामध्ये दोन स्थिर अवस्था असतात - म्हणजे, त्याचे आउटपुट एकतर 0 V किंवा +5 Vdc असते. फ्लिप-फ्लॉपला मेमरी डिव्हाइस म्हणून देखील ओळखले जाते जे 1 बिट माहिती संचयित (संग्रह) करू शकते (1-bit memory cell).

#### 4.4.2 नॉर गेट्स वापरून R-S ल्याच (R-S Latch using NOR Gates)



आकृती 4.8 नॉर गेट्स वापरून R-S ल्याच

कार्यपद्धत:- 1.  $S=0, R=0$

नॉर गेटच्या इनपुटवरील 0 चा त्याच्या आउटपुटवर कोणताही परिणाम होत नसल्यामुळे, फ्लिप-फ्लॉप फक्त त्याच्या सध्याच्या स्थितीत राहतो; म्हणजे, Q अपरिवर्तित राहतो. आउटपुट मागील स्थितीतच राहते म्हणजेच त्यात मागील डेटा राहतो. आउटपुट मध्ये कोणताही बदल होत नाही. (No change;NC).

2.  $S=0, R=1$

R इनपुट NOR गेटचे उच्च इनपुट मुळे, त्याचे आउटपुट 0 राहते. त्यामुळे S इनपुट नॉर गेटचे दोन्ही इनपुट 0 राहते. यामुळे फ्लिप-फ्लॉपचे आउटपुट 0 होईल. (RESET state).

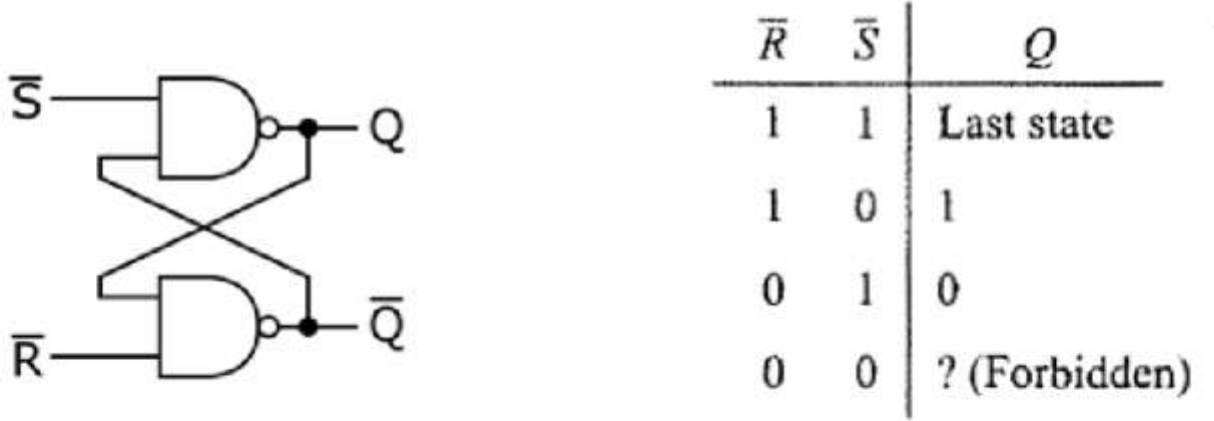
3.  $S=1, R=0$  - S इनपुट NOR गेटचे उच्च इनपुट मुळे, त्याचे आउटपुट 0 होते. त्यामुळे R इनपुट नॉर गेटचे दोन्ही इनपुट 0 होते. यामुळे फ्लिप-फ्लॉपचे आउटपुट  $Q = 1$  होईल. (SET state)..

4.  $S=1, R=1$  - दोन्ही गेटचे आउटपुट 0 राहते. हे अशक्य आहे कारण Q व Q' हे आउटपुट्स पूरक (कॉम्प्लिमेंटरी) म्हणजेच नेहमी एकमेकांच्या विरुद्ध स्थितीत असतात म्हणून  $S=1, R=1$  ही अवस्था



टाळली जाते. अशा प्रकारचे इनपुट दिल्यास आउटपुट स्थिर मिळत नाही म्हणून या क्रियेला निषिद्ध (forbidden state) म्हणतात.

#### 4.4.3 न्यांड गेट्स वापरून R-S ल्याच (RS latch using NAND gates)



आकृती 4.9 न्यांड गेट्स वापरून R-S ल्याच

1-  $S'=1, R'=1$  - आउटपुट मागील स्थितीतच राहते. म्हणजे त्यात मागील डेटा असतो. आउटपुट मध्ये कोणताही बदल होत नाही (no change: NC).

2-  $S'=0, R'=1$  -  $S'$  इनपुट न्यांड गेटचे 0 इनपुट व 1 दुसऱ्या न्यांड गेटचे, कारण त्याचे आउटपुट 1 आहे. त्यामुळे  $R'$  इनपुटसह न्यांड गेटचे दोन्ही इनपुट 1 होतात. यामुळे फ्लिप - फ्लॉपचे आउटपुट 1 येईल. **सेट (SET state)**.

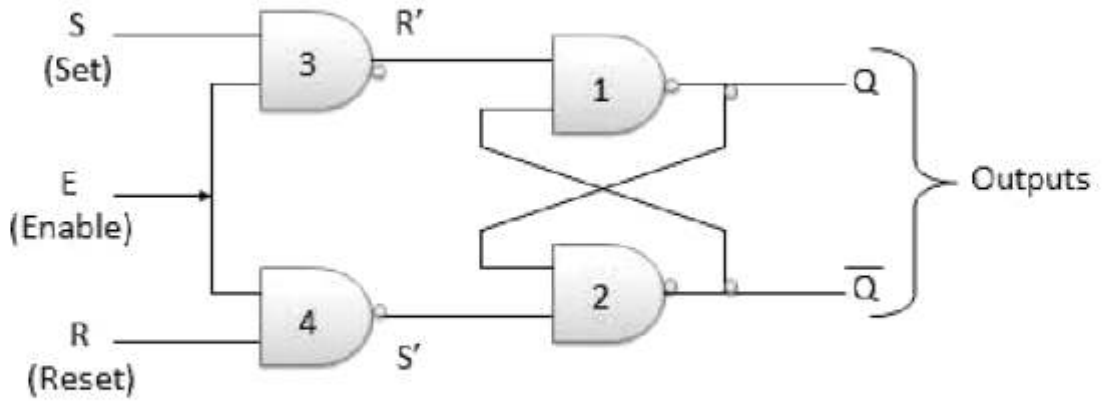
3-  $S'=1, R'=0$  -  $R'$  इनपुटसह न्यांड गेटचे 0 इनपुट व 1 दुसऱ्या न्यांड गेटचे, कारण त्याचे आउटपुट 1 आहे. त्यामुळे  $S'$  इनपुटसह न्यांड गेटचे दोन्ही इनपुट 1 होतात. यामुळे फ्लिप - फ्लॉपचे आउटपुट 0 येईल. **रीसेट (RESET state)**.

4-  $S'=0, R'=0$  - दोन्ही गेटचे आउटपुट 1 राहते. हे अशक्य आहे कारण  $Q$  व  $Q'$  हे आउटपुट्स पूरक (कॉम्प्लिमेंटरी) म्हणजेच नेहमी एकमेकांच्या विरुद्ध स्थितीत असतात म्हणून  $S'=0, R'=0$  ही अवस्था टाळली जाते. अशा प्रकारचे इनपुट दिल्यास आउटपुट स्थिर मिळत नाही म्हणून या क्रियेला निषिद्ध (forbidden state) म्हणतात.



**4.4.4 मेमरी सेल - फ्लिप-फ्लॉप** हे एक बायस्टेबल सर्किट आहे. म्हणजेच त्याच्या दोन स्थिर अवस्था आहेत. याला 1-बिट मेमरी सेल असेही म्हणतात. हे बायनरी बिट एकतर '0' किंवा '1' संचयित करण्यासाठी वापरले जाऊ शकते. एक फ्लिप-फ्लॉप 1-बिट इ. संचय (store) करतो.

**4.4.5 S-R फ्लिप-फ्लॉप** - हे अतिरिक्त इन्याबल (enable) इनपुटसह न्यांड गेट्स वापरून S-R लॅच आहे. याला लेव्हल ट्रिगर्ड SR-FF असेही म्हणतात. यासाठी, इन्याबल इनपुट (E) सक्रिय (active) केले तरच सर्किट सक्रिय होईल. थोडक्यात हे सर्किट  $E = 1$  असल्यास S-R लॅच म्हणून कार्य करेल व  $E = 0$  असल्यास आउटपुटमध्ये कोणताही बदल होणार नाही.



आकृती 4.10 S-R फ्लिप-फ्लॉप

**कार्यपद्धत:-** 1-  $S = R = 0, E=1$ : कोणताही बदल नाही (नो चेंज **NC**).

जर  $S = R = 0$  असेल तर न्यांड गेट्स 3 आणि 4 चे आउटपुट 1 व्हायला भाग पाडले जाईल. त्यामुळे  $R'$  आणि  $S'$  दोन्ही 1 बरोबर असतील.  $S'$  आणि  $R'$  हे न्यांड गेट्स वापरून मूळ S-R लॅचचे इनपुट असल्याने, आउटपुटच्या स्थितीत कोणताही बदल होणार नाही.

2-  $S = 0, R = 1, E = 1$

$S = 0$  पासून, न्यांड-3 चे आउटपुट म्हणजे  $R' = 1$  आणि  $E = 1$  पासून न्यांड-4 चे आउटपुट म्हणजे  $S' = 0$ . म्हणून  $Q_{n+1} = 0$  आणि  $Q_{n+1}$  बार = 1. ही Reset स्थिती आहे.

3-  $S = 1, R = 0, E = 1$

न्यांड-3 चे आउटपुट म्हणजे  $R' = 0$  आणि न्यांड-4 चे आउटपुट म्हणजे  $S' = 1$ . म्हणून S-R न्यांड लॅचचे आउटपुट  $Q_{n+1} = 1$  आणि  $Q_{n+1}$  बार = 0 आहे. ही SET स्थिती आहे.

4-  $S = 1, R = 1, E = 1$  - न्यांड गेट्स 3 आणि 4 दोन्हीचे आउटपुट 0 आहे म्हणजे  $S' = R' = 0$ . त्यामुळे मूळ न्यांड लॅचमध्ये forbidden state निर्माण होईल.

तक्ता क्र. 4.2 सत्य सारणी S-R फ्लिप-फ्लॉप

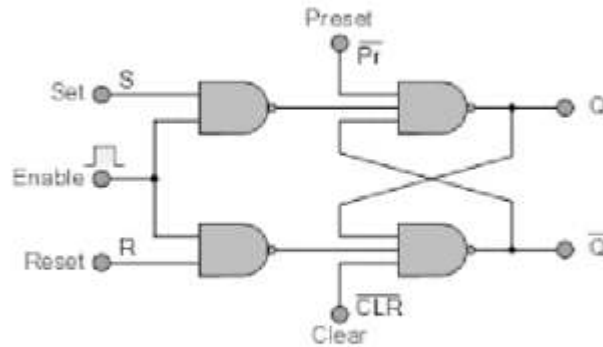
Inputs			Outputs		Comments
E	S	R	$Q_{n+1}$	$\overline{Q}_{n+1}$	
1	0	0	$Q_n$	$\overline{Q}_n$	No change
1	0	1	0	1	Rset
1	1	0	1	0	Set
1	1	1	x	x	Indeterminate

**प्रिसेट (Preset):** हे फ्लिप-फ्लॉप आउटपुट 1 वर सेट करण्यासाठी असिंक्रोनस (asynchronous) इनपुट आहे.

**क्लियर (Clear):** हे फ्लिप-फ्लॉप आउटपुट 0 वर रीसेट करण्यासाठी वापरलेले असिंक्रोनस (asynchronous) इनपुट आहे.

#### 4.4.5.1 प्रिसेट आणि क्लियर R-S फ्लिप-फ्लॉप

R-S फ्लिप-फ्लॉप जेव्हा फ्लिप-फ्लॉप सर्किटवर प्रथम पॉवर सप्लाय दिला जातो, तेव्हा आउटपुटची प्रारंभिक तार्किक स्थिती पूर्णपणे अनिश्चित असू शकते. ते सेट स्थितीत असू शकते, ( $Q = 1$ ) किंवा रीसेट स्थिती, ( $Q = 0$ ) असू शकते. तर अनेक ऍप्लिकेशन्समध्ये आउटपुट पूर्वनिर्धारित स्थितीत सेट करणे आवश्यक असू शकते, अशावेळी डेटा स्वीकारण्यासाठी सेट किंवा रीसेट वापरले जाते. प्रिसेट (PR) आणि क्लियर (CLR), हे दोन अतिरिक्त असिंक्रोनस इनपुट वापरून फ्लिप-फ्लॉप आउटपुट सेट किंवा रीसेट केले जाते.



आकृती 4.11 प्रिसेट आणि क्लियर R-S फ्लिप-फ्लॉप

जेव्हा क्लिअर इनपुट "0" असेल आणि प्रिसेट इनपुट "1" असेल तेव्हा हे अतिरिक्त इनपुट फ्लिप-फ्लॉप क्लिअर करतात ( $Q = 0$ ).

त्याचप्रमाणे, जेव्हा प्रिसेट इनपुट "0" असेल आणि क्लिअर इनपुट "1" असेल तेव्हा फ्लिप-फ्लॉप लॉजिक "1" स्थितीवर प्रिसेट केले जाऊ शकते.

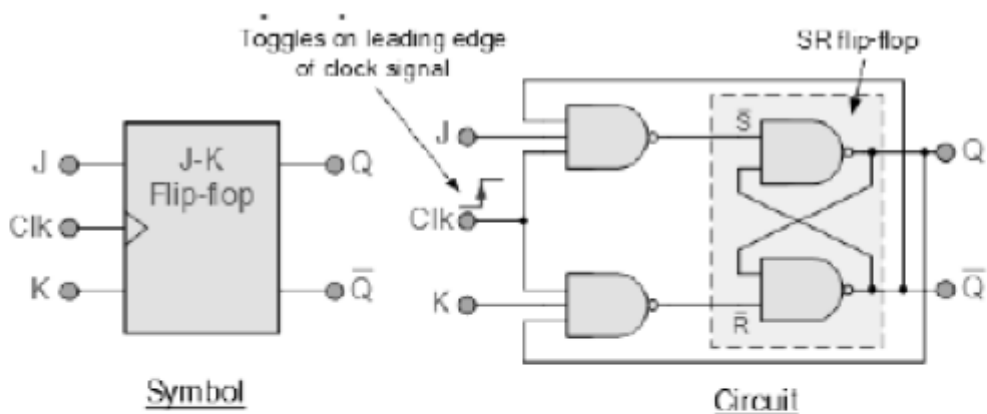
परंतु काही डिजिटल सिस्टीम्समध्ये सुरुवातीला सर्व फ्लिप-फ्लॉप प्रिसेट ( $Q=1$ ) किंवा क्लिअर ( $Q=0$ ) करणे आवश्यक असते. यासाठी प्रिसेट व क्लिअर इनपुट्सचा उपयोग होतो.

प्रिसेट आणि क्लिअर दोन्ही इनपुट एकाच वेळी सक्रिय लो ( $PR = CLR = 0$ ) केले जाऊ नयेत. कारण यामुळे अनिश्चित स्थिती येते.

**4.5 J-K फ्लिप-फ्लॉप (J-K Flip-flop)** - S-R फ्लिप-फ्लॉप सर्किटचे अनुक्रमिक लॉजिक सर्किट्समध्ये बरेच फायदे आणि उपयोग आहेत परंतु एका फ्लिप-फ्लॉप मध्ये निषिद्ध स्थिती निर्माण होऊ शकते. ते टाळण्या करिता J-K फ्लिप-फ्लॉप विकसित करण्यात आला.

J-K फ्लिप-फ्लॉप हे मूलतः एक गेटेड S-R फ्लिप-फ्लॉप आहे ज्यामध्ये क्लॉक इनपुट सर्किटरी जोडली जाते. हे अवैध आउटपुट स्थिती प्रतिबंधित करते जी दोन्ही इनपुट S आणि R "1" असतात तेव्हा उद्भवू शकतात.

#### 4.5.1 मूलभूत J-K फ्लिप-फ्लॉप (Basic JK Flip-flop)



आकृती 4.12 मूलभूत J-K फ्लिप-फ्लॉप

मागील S-R फ्लिप-फ्लॉपचे S आणि R दोन्ही इनपुट आता अनुक्रमे J आणि K इनपुट नावाच्या दोन इनपुट्सने बदलले आहेत.

**कार्यपद्धत:-**

यामध्ये दोन न्यांड गेट्स व एक R-S फ्लिप-फ्लॉप वापरून लेव्हल क्लॉक प्रकारचा J-K फ्लिप-फ्लॉप केला आहे.

क्लॉक=1 दिली असता, फ्लिप-फ्लॉपचे कार्य नियंत्रित केले जाते. CLK=0 असताना न्यांड गेट्स अकार्यक्षम होतात. त्यामुळे सर्किट क्रियाशील नसते. त्यामुळे आऊटपुट पूर्वीच्या स्थितीत (No change) (Q) राहते. आऊटपुट मध्ये कोणताही बदल होत नाही (नो चेंज NC)

1. CLK=0 असताना, दोन्ही न्यांड गेट्स अकार्यक्षम (डिसएबल्ड) असल्यामुळे, J व K च्या किमती काहीही असल्या (डोन्ट केअर), आऊटपुट (Q) पूर्वीच्या स्थितीत राहते (नो चेंज NC).

2. CLK=1, J=0 व K=0 असल्यास, दोन्ही न्यांड गेट्स डिसएबल्ड असल्यामुळे आऊटपुट (Q) पूर्वीच्या स्थितीत राहते नो चेंज (NC).

3. CLK=1, J=0, व K=1 असल्यास, खालील न्यांड गेट कार्यक्षम होऊन ते R-1 पल्स देते. त्यामुळे फ्लिप-फ्लॉपचे आऊटपुट (Q) पूर्वी 0 असल्यास तसेच राहते व पूर्वी 1 असल्यास बदलून 0 होते. फ्लिप-फ्लॉप रिसेट होतो.  $Q=0$  व  $Q'=1$ .

4. CLK=1, J=1 व K=0 असल्यास, वरील गेट S-1 पल्स देते. त्यामुळे फ्लिप-फ्लॉपचे आऊटपुट पूर्वी 1 असल्यास 1 राहते व पूर्वी 0 असल्यास ते बदलून 1 होते. फ्लिप-फ्लॉप सेट होतो.  $Q=1$  व  $Q' = 0$ .

5. CLK=1, J=1 व K=1 असतील तर फ्लिप-फ्लॉप सेट किंवा रिसेट करता येतो.

जर  $Q=1$  असेल, तर खालील न्यांड गेट रिसेट पल्स पाठवते. म्हणून  $Q=0$  होतो.

जर  $Q=0$  असेल, तर वरील न्यांड गेट सेट पल्स पाठवते. म्हणून  $Q=1$  होतो.

अशा प्रकारे, J=1 व K=1 असताना पूर्वीच्या स्थितीचा पूरक ( $Q'$ ) मिळतो. याला टॉगल (toggle) असे म्हणतात.



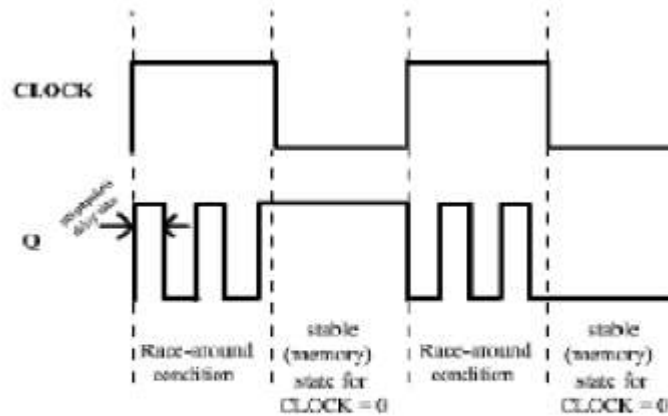
### तक्ता क्र. 4.3 सत्य सारणी J-K फ्लिप-फ्लॉप

Clk	J	K	Q	Q'	State
1	0	0	Q	Q'	No change in state
1	0	1	0	1	Resets Q to 0
1	1	0	1	0	Sets Q to 1
1	1	1	-	-	Toggles

#### 4.5.2 J-K फ्लिप-फ्लॉपचे तोटे-(रेस आराऊंड स्थिती) (Race around condition)

हे सर्किट क्लॉक S-R फ्लिप-फ्लॉपवर एक सुधारणा आहे. परंतु या फ्लिप-फ्लॉप मध्ये "रेस around " स्थिती येऊ शकते. समजा की क्लॉक = 1, J=K=1 आहे व J-K फ्लिप-फ्लॉप रीसेट स्थितीत आहे म्हणजे Q = 0. आता J=K=1 लागू केल्यावर, आउटपुट मागील मूल्याला पूरक बनते, म्हणजे Q = 1 होतो. परंतु गेट्सच्या प्रसार विलंबाच्या कालावधीनंतर (propagation delay), आउटपुट पुन्हा पूरक होते आणि Q = 0 होते. हे क्लॉक = 1 आणि J=K=1 असेपर्यंत चालू राहते. या घटनेला 0 to 1, 1 to 0, 0 to 1 आणि याप्रमाणे आउटपुट बदलत असे म्हणतात. याला रेस-आराऊंड कंडिशन म्हणतात.

रेस-आराऊंड कंडिशन- J-K फ्लिप-फ्लॉपसाठी J, K आणि क्लॉक =1 असल्यास फ्लिप-फ्लॉपची स्थिती टॉगल होत राहते ज्यामुळे फ्लिप-फ्लॉपचे आउटपुट निश्चित राहत नाही.



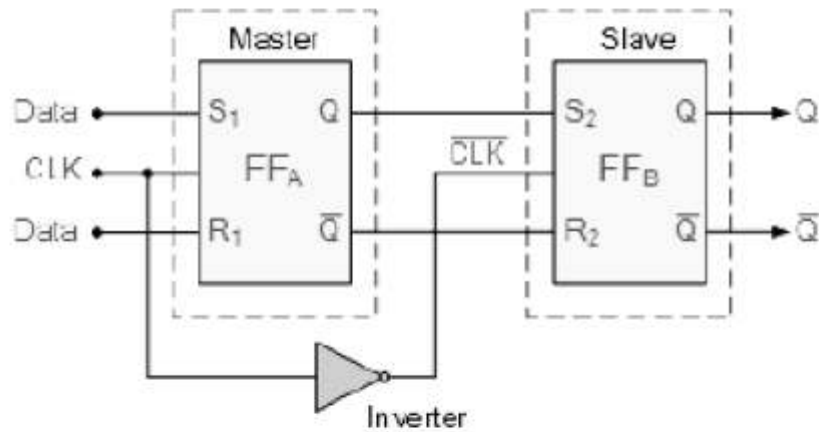
आकृती 4.13 रेस आराऊंड कंडिशन

#### 4.5.3 रेस-अराउंड कंडिशन टाळण्याच्या पद्धती

1. फ्लिप-फ्लॉपच्या प्रसार विलंबापेक्षा क्लॉक width कमी करून रेस-अराउंड कंडिशन स्थिती टाळू शकतो. हे एज ट्रिगरिंगद्वारे प्राप्त केले जाऊ शकते.
2. रेस-अराउंड कंडिशन टाळण्याकरीता मास्टर स्लेव् फ्लिप-फ्लॉप वापरता येतो.

**4.5.3.1 मास्टर-स्लेव् J-K फ्लिप-फ्लॉप (Master-slave) -** मास्टर-स्लेव् फ्लिप-फ्लॉप कॉन्फिगरेशनमध्ये एकत्र जोडलेले दोन S-R फ्लिप-फ्लॉप वापरून time delay बाबतच्या सर्व समस्या दूर करते. एक फ्लिप-फ्लॉप "मास्टर" सर्किट म्हणून कार्य करते, जे क्लॉक पल्सच्या रायजिंग एज वर ट्रिगर होते तर दुसरे "स्लेव्" सर्किट म्हणून कार्य करते, जे क्लॉक पल्सच्या फॉलिंग एज वर ट्रिगर होते. "स्लेव्" फ्लिप-फ्लॉपमधील Q आणि Q बार मधील आउटपुट "मास्टर" च्या इनपुटमध्ये परत दिले जातात आणि "मास्टर" फ्लिप-फ्लॉपचे आउटपुट "स्लेव्" फ्लिप-फ्लॉपच्या दोन इनपुटशी कनेक्ट केले जातात.

**4.5.4 बेसिक मास्टर-स्लेव् कॉन्फिगरेशन -** जेव्हा क्लॉक पल्स उच्च होते, तेव्हा S आणि R इनपुटमधील डेटा मास्टर फ्लिप-फ्लॉप, FFA द्वारे दिला जातो.

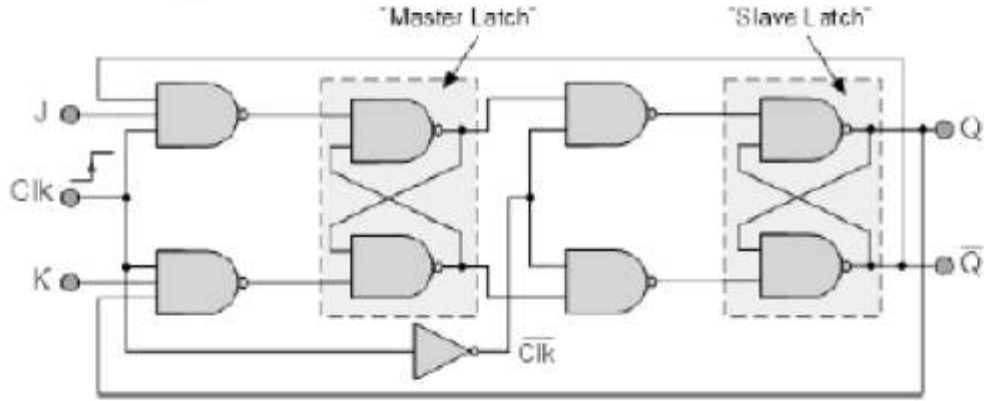


आकृती 4.14 बेसिक मास्टर-स्लेव् J-K फ्लिप-फ्लॉप

स्लेव् फ्लिप-फ्लॉप, FFB ला इन्व्हर्टर वापरून क्लॉक सिग्नल दिला आहे. त्या मुळे दोन्ही फ्लिप-फ्लॉप चे कार्य वेग वेगळ्या वेळेत होत असते.



जेव्हा सुरुवातीच्या क्लॉकची पल्स "0" वर येते, तेव्हा मास्टर अक्षम होतो आणि बाह्य डेटा इनपुटला त्याच्या आउटपुटमध्ये माहिती देण्यापासून वंचित करतो, तर स्लेव् फ्लिप-फ्लॉप आता सक्षम होतो आणि अशा प्रकारे लॅच केलेली माहिती त्याच्या आउटपुट Q वर पाठवते.



आकृती 4.15 मास्टर-स्लेव् J-K फ्लिप-फ्लॉप

लॉजिक लेव्हल "1" वर क्लॉक इनपुट "हाय" आहे. "स्लेव्" फ्लिप-फ्लॉपचे क्लॉक इनपुट हे "मास्टर" क्लॉक इनपुटचे व्यस्त (पूरक) असल्याने, "स्लेव्" S-R फ्लिप-फ्लॉप टॉगल होत नाही. जेव्हा क्लॉकचे इनपुट लॉजिक "0" वर जाते तेव्हा "मास्टर" फ्लिप-फ्लॉपचे आउटपुट गेट केलेल्या "स्लेव्" फ्लिप-फ्लॉपद्वारे जातात. जेव्हा क्लॉक "लो" असते, तेव्हा "मास्टर" फ्लिप-फ्लॉपचे आउटपुट लॅच केले जातात आणि त्याच्या इनपुटमधील कोणतेही अतिरिक्त बदल दुर्लक्षित केले जातात. गेट केलेला "स्लेव्" फ्लिप-फ्लॉप आता "मास्टर" द्वारे पास केलेल्या इनपुटच्या स्थितीला प्रतिसाद देतो. नंतर क्लॉक पल्सच्या "लो-टू-हाय" संक्रमणावर "मास्टर" फ्लिप-फ्लॉपचे इनपुट "स्लेव्" फ्लिप-फ्लॉपच्या गेट केलेल्या इनपुटमधून दिले जातात आणि "हाय-टू-लो" संक्रमणावर तेच इनपुट "स्लेव्" च्या आउटपुटवर येते ज्यामुळे या प्रकारचा फ्लिप-फ्लॉप एज किंवा पल्स ट्रिगर होतो.

**कार्यपद्धत:-**

1  $J = K = 0$  (कोणताही बदल नाही) - जेव्हा क्लॉक = 0, तेव्हा स्लेव् सक्रिय होतो आणि मास्टर निष्क्रिय असतो. परंतु S आणि R इनपुट बदलले नसल्यामुळे, स्लेव् आउटपुट देखील बदलत नाही. म्हणून  $J = K = 0$  असल्यास आउटपुट बदलणार नाहीत.

2  $J = 0$  आणि  $K = 1$  क्लॉक = 1 - मास्टर सक्रिय, स्लेव निष्क्रिय. म्हणून मास्टरचे आउटपुट  $Q_1 = 0$  आणि  $Q_1$  बार = 1 बनतात. म्हणजे  $S = 0$  आणि  $R = 1$ . क्लॉक = 0 - स्लेव सक्रिय, मास्टर निष्क्रिय. त्यामुळे स्लेवचे आउटपुट  $Q = 0$  आणि  $Q$  बार = 1 बनतात. पुन्हा क्लॉक = 1 - मास्टर सक्रिय, स्लेव निष्क्रिय. त्यामुळे बदललेल्या आउटपुटसह  $Q = 0$  आणि  $Q$  बार = 1 मास्टरला परत दिले तरी, त्याचे आउटपुट  $Q_1 = 0$  आणि  $Q_1$  बार = 1 असेल. म्हणजे  $S = 0$  आणि  $R = 1$ . म्हणून क्लॉक = 0 आणि सक्रिय स्लेवचे आउटपुट  $Q = 0$  आणि  $Q$  बार = 1 राहतील. अशा प्रकारे आपल्याला मास्टर स्लेवकडून एक स्थिर आउटपुट मिळेल.

3  $J = 1$  आणि  $K = 0$  क्लॉक = 1 - मास्टर सक्रिय, स्लेव निष्क्रिय. म्हणून मास्टरचे आउटपुट  $Q_1 = 1$  आणि  $Q_1$  बार = 0 बनतात. म्हणजे  $S = 1$  आणि  $R = 0$ . क्लॉक = 0 - स्लेव सक्रिय, मास्टर निष्क्रिय. म्हणून स्लेवचे आउटपुट  $Q = 1$  आणि  $Q$  बार = 0 बनतात. पुन्हा क्लॉक = 1 - नंतर हे दाखवले जाऊ शकते की स्लेवचे आउटपुट  $Q = 1$  आणि  $Q$  बार = 0 वर स्थिर झाले आहेत.

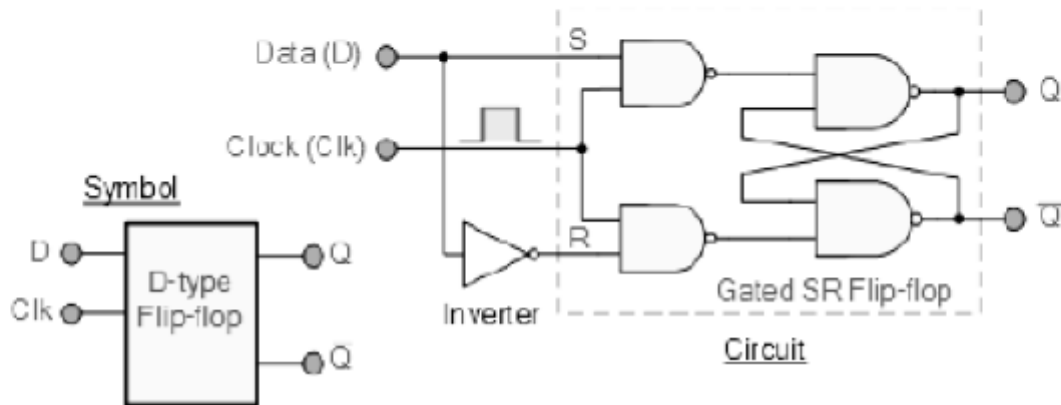
4  $J = K = 1$  क्लॉक = 1 - मास्टर सक्रिय, स्लेव निष्क्रिय. मास्टरचे आउटपुट टॉगल होतील. त्यामुळे  $S$  आणि  $R$  देखील पूरक होतील. क्लॉक = 0 - स्लेव सक्रिय, मास्टर निष्क्रिय. स्लेवचे आउटपुट टॉगल होतील. हे बदललेले आउटपुट मास्टर इनपुटवर परत केले जातात. परंतु क्लॉक = 0 असल्यामुळे, मास्टर अद्याप निष्क्रिय आहे. त्यामुळे ते या बदललेल्या आउटपुटला प्रतिसाद देत नाही. हे एका पेक्षा अधिक टॉगलिंग टाळते. मास्टर स्लेव फ्लिप-फ्लॉप रेस टाळेल.

तक्ता क्र. 4.4 सत्य सारणी मास्टर-स्लेव J-K फ्लिप-फ्लॉप

Inputs			Outputs		Comments
E	J	K	$Q_{n+1}$	$\overline{Q}_{n+1}$	
1	0	0	$Q_n$	$\overline{Q}_n$	No change
1	0	1	0	1	Rset
1	1	0	1	0	Set
1	1	1	$\overline{Q}_n$	$Q_n$	Toggle

#### 4.5.5 D फ्लिप-फ्लॉप (D flip-flop):-

D फ्लिप-फ्लॉप हे क्लॉक केलेल्या फ्लिप-फ्लॉप्सपैकी सर्वात महत्वाचे आहे कारण ते हे सुनिश्चित करते की इनपुट S आणि R कधीही एकाच वेळी एकसारखे नसतात. एका साध्या S-R फ्लिप-फ्लॉपसाठी दोन इनपुट आवश्यक असतात, एक आउटपुट "सेट" करण्यासाठी आणि एक आउटपुट "रीसेट" करण्यासाठी. D-टाइप फ्लिप-फ्लॉप एका गेटेड S-R फ्लिप-फ्लॉपपासून तयार केला जातो आणि एकच D डेटा इनपुट देण्यासाठी S आणि R इनपुटमध्ये इन्व्हर्टर जोडला असतो. यात फक्त एक इनपुट आहे. इनपुट डेटा काही काळानंतर आउटपुटवर येतो. इनपुट आणि आउटपुट दरम्यान या डेटा विलंबामुळे, त्याला विलंब फ्लिप-फ्लॉप (delay flip-flop) म्हणतात. इन्व्हर्टरमुळे S आणि R एकमेकांचे पूरक असतील. त्यामुळे  $S = R = 0$  किंवा  $S = R = 1$ , ही इनपुट स्थिती कधीही दिसणार नाही.



आकृती 4.16 D फ्लिप-फ्लॉप

अशा प्रकारे या इनपुटला "डेटा" इनपुट म्हणतात. जर हा डेटा इनपुट हाय असेल तर फ्लिप-फ्लॉप "सेट" होईल आणि जेव्हा ते लो असेल तेव्हा फ्लिप-फ्लॉप बदलेल आणि "रीसेट" होईल. तथापि, फ्लिप-फ्लॉपच्या लॅचिंग सर्किटरीमधून डेटा इनपुट करण्यासाठी "क्लॉक" किंवा "enable" नावाचा अतिरिक्त इनपुट वापरला जातो. परिणाम असा आहे की जेव्हा क्लॉक इनपुट सक्रिय असते तेव्हा D इनपुट स्थिती आउटपुट Q वर कॉपी केली जाते.

जोपर्यंत क्लॉक इनपुट उच्च असेल तोपर्यंत "D फ्लिप-फ्लॉप" डेटा टर्मिनलवर कोणतीही लॉजिक लेव्हल लागू केली जाईल आणि ती आउटपुट वर कॉपी करेल. एकदा क्लॉक इनपुट कमी झाले की

फ्लिप-फ्लॉपचे "सेट" आणि "रीसेट" इनपुट दोन्ही लॉजिक लेव्हल "1" वर जातात त्यामुळे ते स्थिती बदलणार नाही आणि क्लॉक संक्रमण होण्यापूर्वी त्याच्या आउटपुटवर जो काही डेटा उपस्थित होता तो संग्रहित केला जाईल. दुसऱ्या शब्दांत आउटपुट लॉजिक "0" किंवा लॉजिक "1" वर "लॅच केलेले" आहे.

तक्ता क्र. 4.5 सत्य सारणी D- फ्लिप-फ्लॉप

Clock	D	Q	Q'	Description
↓ » 0	X	Q	Q'	Memory no change
↑ » 1	0	0	1	Reset Q » 0
↑ » 1	1	1	0	Set Q » 1

लक्षात ठेवा: ↓ आणि ↑ क्लॉक पल्सची दिशा दर्शवतात कारण असे गृहीत धरले जाते की D-फ्लिप फ्लॉप्स एज ट्रिगर आहेत

**कार्यपद्धत:-**

- 1 क्लॉक = 0, त्याच अक्षम आहे. त्यामुळे आउटपुटमध्ये कोणताही बदल होत नाही
- 2 जर क्लॉक = 1 आणि D = 0 असेल तर S = 0 आणि R = 1. त्यामुळे, पुढील स्थिती  $Q_{n+1} = 0$  आणि  $Q_{n+1}$  बार = 1 आहे. ही रीसेट स्थिती आहे.
- 3 जर क्लॉक = 1 आणि D = 1, असेल तर S = 1 आणि R = 0. हे लॅच सेट करेल आणि  $Q_{n+1}=1$  आणि  $Q_{n+1}$  बार = 0

#### 4.5.6 टॉगल फ्लिप-फ्लॉप (T फ्लिप-फ्लॉप)

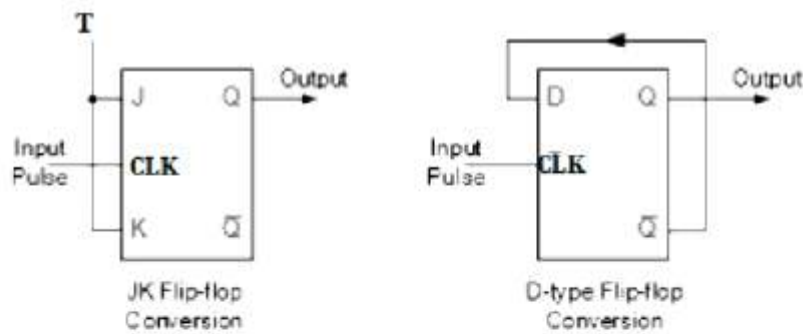
क्लॉक इनपुट लागू केल्यावर व T = 1 असेल तर टॉगल फ्लिप-फ्लॉप आउटपुट स्थिती बदलते, आणि T = 0 असताना बदलत नाही. क्लॉक "0" ते "1" चे संक्रमण आउटपुट टॉगल करेल. टॉगल T- फ्लिप-फ्लॉप हे फ्रिक्वेन्सी डिव्हायडर आणि डिजिटल काउंटरसह अनेक डिजिटल सर्किट्सचे मूलभूत बिल्डिंग ब्लॉक आहे.



टॉगल T- फ्लिप-फ्लॉप, J-K फ्लिप-फ्लॉपपासून तयार केले जाऊ शकतात. J आणि K इनपुट दाखवल्याप्रमाणे एकत्र जोडलेले असतात व त्यांना हाय (logic1) सिग्नल दिलेला असतो. क्लॉक इनपुट नुसार आउटपुट टॉगल होईल.

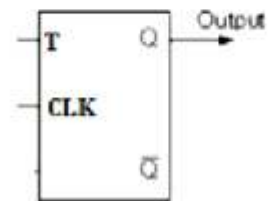
आकृती 4.17 मध्ये दर्शविल्याप्रमाणे त्यात फक्त T ने दर्शविलेले इनपुट आहे.

आकृती 4.17 T फ्लिप-फ्लॉप व सत्य सारणी



Truth Table

Inputs		Outputs		Comments
E	T	$Q_{n+1}$	$\overline{Q}_{n+1}$	
1	0	$Q_n$	$\overline{Q}_n$	No change
1	1	$\overline{Q}_n$	$Q_n$	Toggle



कार्यपद्धत:- 1 - T = 0, J = K = 0 - आउटपुट Q आणि Q बार बदलणार नाही (नो चेंज)

2 T = 1, J = K = 1 - क्लॉक सिग्नलच्या प्रत्येक रायजिंग एजला आउटपुट टॉगल करेल.

T-फ्लिप-फ्लॉप व्यावसायिकरित्या उपलब्ध नाही परंतु J इनपुटला K इनपुट आणि दोन्ही लॉजिक लेव्हल "1" शी जोडून J-K फ्लिप-फ्लॉप (किंवा D-फ्लिप-फ्लॉप) वरून तयार केले जाऊ शकते. J आणि K हाय सह, फ्लिप-फ्लॉप प्रत्येक वेळी त्याच्या क्लॉक इनपुटवर ट्रिगर झाल्यावर स्थिती बदलते. या आउटपुटला टॉगल आउटपुट म्हणतात कारण आउटपुट "0" असल्यास "1" होते आणि "1" असल्यास "0" होते, म्हणजे ते टॉगल होते.

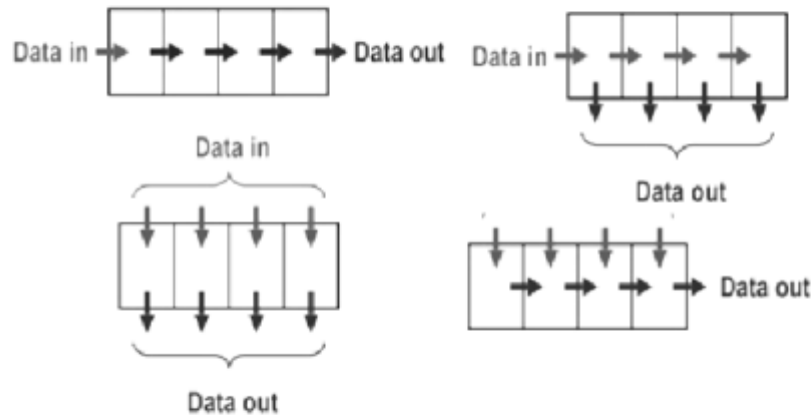
#### 4.5.8 ॲप्लिकेशन्स-

- D फ्लिप-फ्लॉप डिजिटल सिग्नल प्रोसेसिंग सिस्टममध्ये वापरल्या जाणाऱ्या विलंब-लाइन तयार करण्यासाठी वापरला जाऊ शकतो
- इव्हेंट डिटेक्टर
- R-S फ्लिप-फ्लॉपचा वापर स्विच डिबाउन्सर म्हणून केला जाऊ शकतो.
- डेटा सिंक्रोनायझर्स
- कॉउंटर्स (counters)
- वारंवारता विभाजक (frequency divider).

**4.6 शिफ्ट रजिस्टर्स (Shift Registers)** - एक प्रकारचे रजिस्टर आहे जो डेटा शिफ्ट करण्यास सक्षम आहे.

**बायडायरेक्शनल शिफ्ट रजिस्टर (Bidirectional shift register)** - एक प्रकारचे शिफ्ट रजिस्टर आहे जो डेटा बिट डावीकडे आणि/किंवा उजवीकडे शिफ्ट करण्यास सक्षम आहे.

**युनिव्हर्सल शिफ्ट रजिस्टर (Universal shift register):** एक प्रकारचे शिफ्ट रजिस्टर आहे जो डेटा उजवीकडे किंवा डावीकडे शिफ्ट तसेच डेटा समांतर लोड करण्यास सक्षम आहे.



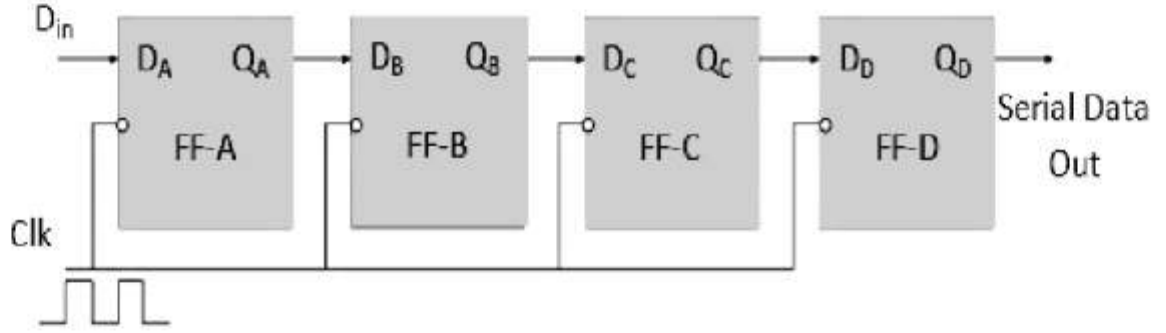
आकृती 4.18 शिफ्ट रजिस्टर्स

शिफ्ट रजिस्टर्स हे मुळात एक प्रकारचे रजिस्टर असतात ज्यात डेटा ट्रान्सफर करण्याची क्षमता असते. "शिफ्ट" रजिस्टर हे सर्वसाधारणपणे स्टोरेज डिव्हाइसेस असतात जे विशिष्ट संख्येच्या फ्लिप-फ्लॉपला मालिकेत एकत्र जोडून तयार केले जातात. रजिस्टरद्वारे संचयित करता येणारा डेटा (बिट्सची संख्या)



फ्लिप-फ्लॉपच्या संख्येच्या प्रमाणात असतो. प्रत्येक फ्लिप-फ्लॉप एका वेळी फक्त एक बिट संचयित करण्यास सक्षम असतो.

#### 4.6.1 सिरीयल-इन-सीरियल-आउट (Serial-In-Serial out Shift Register) (SISO)



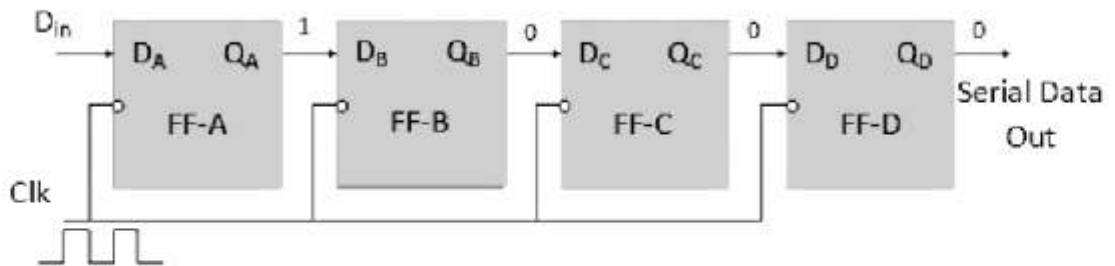
आकृती 4.19 सिरीयल-इन टू सीरियल-आउट

सर्व फ्लिप-फ्लॉप सुरुवातीला रीसेट स्थितीत असतात म्हणजे  $Q_A = Q_B = Q_C = Q_D = 0$ .

आपण असे गृहीत धरू की डेटा 1 1 1 1 रजिस्टरमध्ये लोड करायचा आहे.

हा डेटा  $D_{in}$  बिट वर लागू केला जावा, ज्यामध्ये LSB बिट प्रथम लागू केला जाईल. FFA चे  $D_A$  इनपुट हे डेटा इनपुट  $D_{in}$  शी जोडलेले आहे. FFA चे आउटपुट म्हणजेच  $Q_A$  पुढील फ्लिप-फ्लॉपच्या इनपुटशी जोडलेले आहे, म्हणजे  $D_B$  या नुसार.

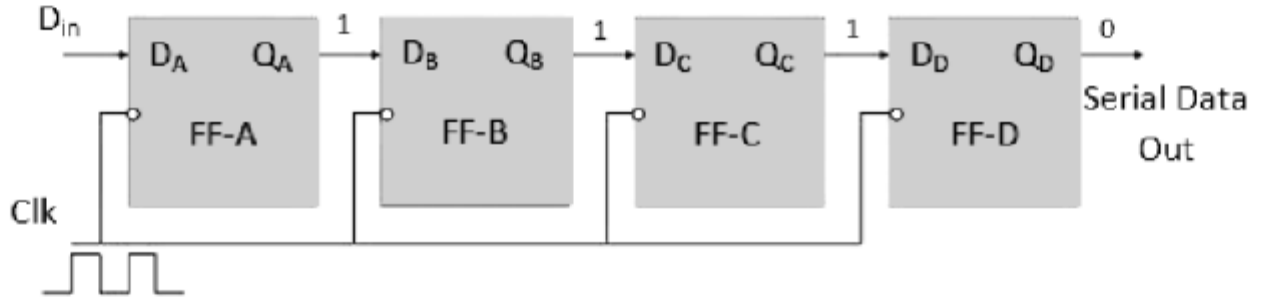
**कार्यपद्धत:-** क्लॉक सिग्नल लागू करण्यापूर्वी प्रविष्ट करायच्या क्रमांकाचा LSB बिट  $D_{in}$  वर उपस्थित असावा. तर  $D_{in}=D_A=1$  - क्लॉकच्या पहिल्या फॉलिंग एजवर, FFA सेट केला जातो आणि रजिस्टरमध्ये संग्रह केला जातो. आकृती 4.20 मध्ये दाखवल्याप्रमाणे  $Q_A Q_B Q_C Q_D = 1000$ .



आकृती 4.20

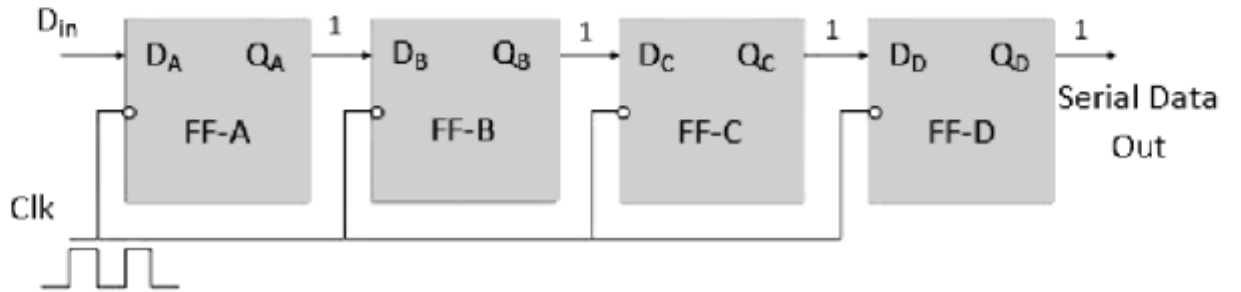
$D_{in} = 1$  - पुढील फॉलिंग क्लॉक एजला, FF-B सेट होईल आणि आउटपुट QA QB QC QD = 1100 मध्ये बदलेल.

पुढील फॉलिंग क्लॉक एज येताच, FF-C सेट केला जाईल आणि आउटपुट QA QB QC QD = 1110 वर जाईल



आकृती 4.21

त्याचप्रमाणे  $D_{in}=1$  आणि चौथ्या फॉलिंग क्लॉक एजवर, रजिस्टरमध्ये आउटपुट QA QB QC QD = 1111 होईल.

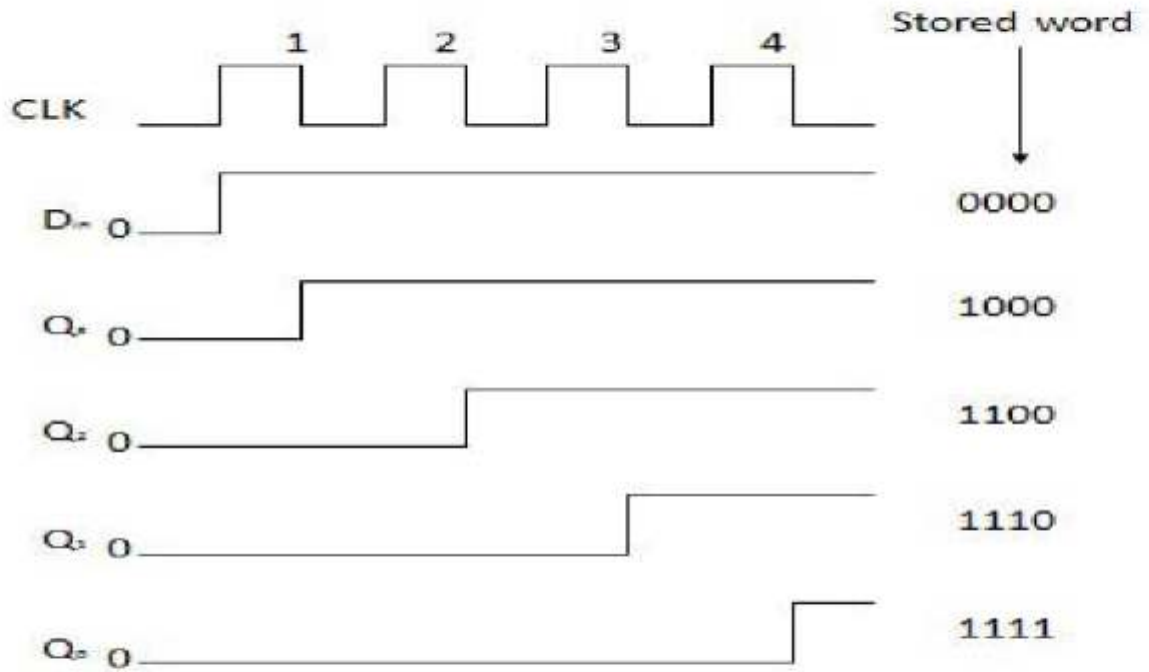


आकृती 4.22

तक्ता. 4.7 सत्य सारणी

	CLK	$D_{in}=Q_A$	$Q_A=D_B$	$Q_B=D_C$	$Q_C=D_D$	$Q_D$
Initially			0	0	0	0
(i)	↓	1	1	0	0	0
(ii)	↓	1	1	1	0	0
(iii)	↓	1	1	1	1	0
(iv)	↓	1	1	1	1	1

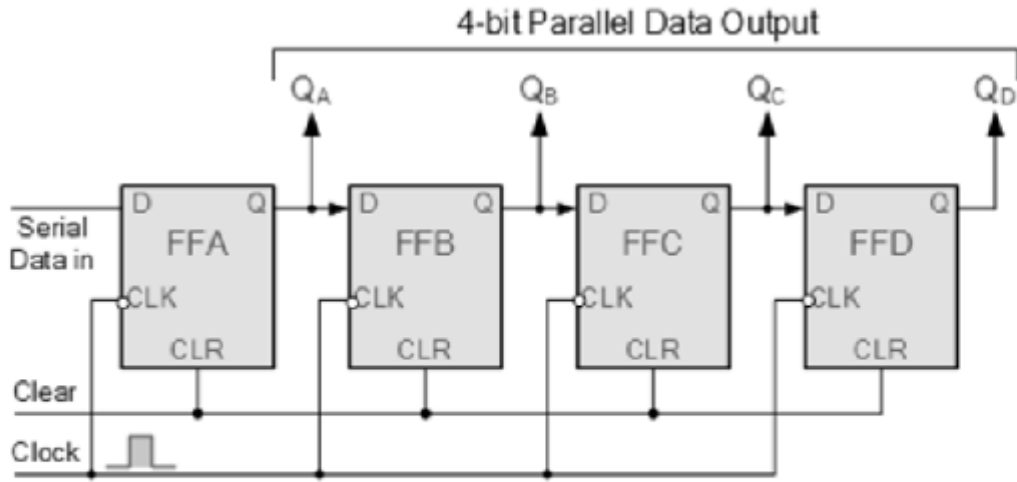
→ Direction of data travel



आकृती 4.23 टाइमिंग डायग्राम

#### 4.6.2 सिरियल-इन-पॅरलल-आउट शिफ्ट रजिस्टर (Serial-In-Parallel out Shift Register) (SIPO)

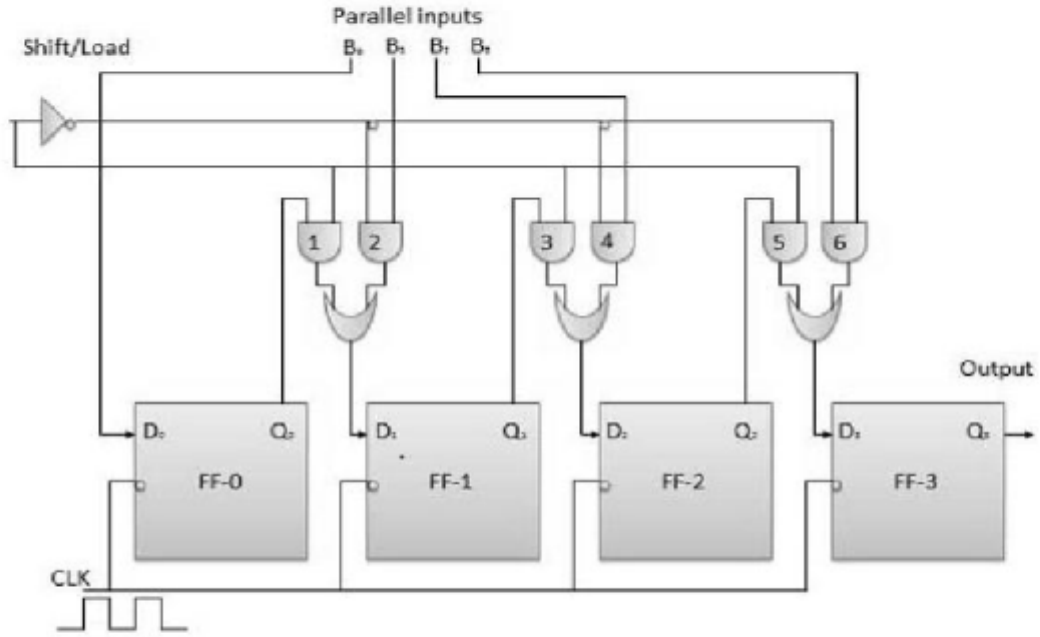
अशा प्रकारच्या ऑपरेशन्समध्ये, डेटा क्रमाने प्रविष्ट केला जातो आणि समांतर पद्धतीने बाहेर येतो. डेटा लोडिंग पूर्ण होताच, सर्व फ्लिप-फ्लॉपमध्ये त्यांचा आवश्यक डेटा असतो. आउटपुट सक्षम केले असता सर्व लोड केलेला डेटा एकाच वेळी सर्व आउटपुट लाइनवर उपलब्ध केला जाईल. चार बिट लोड करण्यासाठी 4 क्लॉक आवश्यक आहेत. त्यामुळे SIPO मोडच्या इनपुट ऑपरेशनची गती SISO मोड सारखीच आहे.



आकृती 4.24 सिरियल-इन टू पॅरलल-आउट शिफ्ट रजिस्टर

#### 4.6.3 समांतर इनपुट सिरीयल आउटपुट (Parallel-In-Serial-out Shift Register) (PISO)

या प्रकारच्या शिफ्ट रजिस्टरमध्ये डेटा बिट्स समांतर पद्धतीने एंटर केले जातात. आकृती 4.25 हे चार बिट समांतर इनपुट सिरीयल आउटपुट शिफ्ट रजिस्टर आहे.



आकृती 4.25 समांतर इनपुट सिरीयल आउटपुट

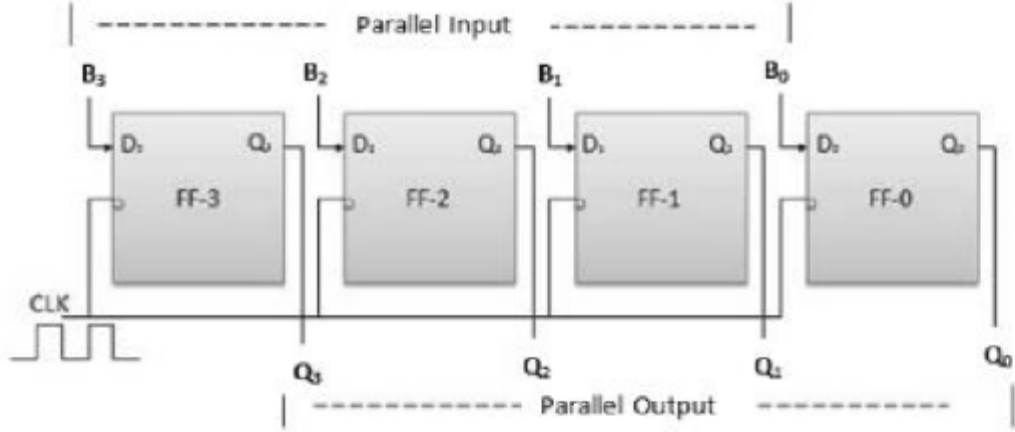
आपण लक्षात घेऊ की, मागील फ्लिप-फ्लॉपचे आऊटपुट हे कॉम्बिनेशनल सर्किटद्वारे पुढील इनपुटशी जोडलेले आहे. बायनरी इनपुट शब्द B0,B1,B2,B3 कॉम्बिनेशनल सर्किटमध्ये लागू केला जातो. हे सर्किट दोन मोडमध्ये काम करू शकते लोड मोड व शिफ्ट मोड.

**लोड मोड (Load Mode)**- जेव्हा शिफ्ट/लोड बार लाईन कमी असते म्हणजे लॉजिक "0", तेव्हा अँड गेट 2,4 आणि 6 सक्रिय होतात. ते संबंधित फ्लिप-फ्लॉपवर B1, B2, B3 बिट देतील. क्लॉकच्या फॉलिंग एजवर, बायनरी इनपुट B0,B1,B2,B3 संबंधित फ्लिप-फ्लॉपमध्ये लोड केले जाईल. अशा प्रकारे समांतर लोडिंग होते.

**शिफ्ट मोड (Shift Mode)**- जेव्हा शिफ्ट/लोड बार लाईन लॉजिक "1" असते, तेव्हा अँड गेट 2,4 आणि 6 निष्क्रिय होतात. त्यामुळे डेटाचे समांतर लोडिंग अशक्य होते. पण अँड गेट 1,3 आणि 5 सक्रिय होतात. त्यामुळे क्लॉक पल्स वापरून डेटाचे स्थानांतर (transfer) डावीकडून उजवीकडे होते. अशा प्रकारे सिरीयल आउट ऑपरेशनमध्ये होते.

**4.6.4 समांतर इनपुट समांतर आउटपुट (Parallel-In-Parallel-out Shift Register) (PIPO) -** या मोडमध्ये, 4 बिट बायनरी इनपुट B0, B1, B2, B3 हे चार फ्लिप-फ्लॉपच्या डेटा इनपुट अनुक्रमे D0, D1, D2, D3 वर लागू केले जाते.

फॉलिंग क्लॉकएज लागू होताच, इनपुट बायनरी बिट्स फ्लिप-फ्लॉपमध्ये एकाच वेळी लोड केले जातील. लोड केलेले बिट्स एकाच वेळी आउटपुटला दिसतील. सर्व बिट्स लोड करण्यासाठी फक्त एक क्लॉक आवश्यक आहे.



आकृती 4.26 समांतर इनपुट समांतर आउटपुट

#### 4.6.5 बायडीरेक्शनल शिफ्ट रजिस्टर (Bidirectional Shift Register)

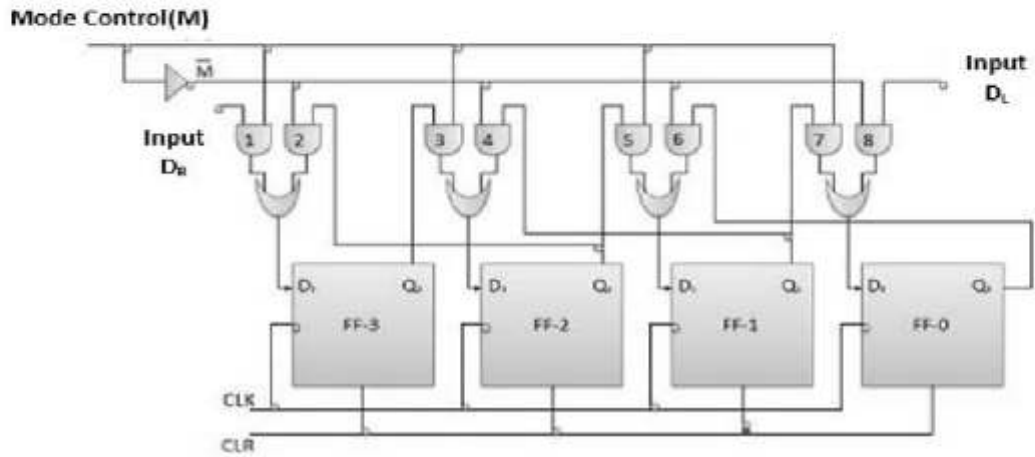
जर बायनरी संख्या एका स्थानाने डावीकडे शिफ्ट केली तर ती मूळ संख्येचा 2 ने गुणाकार केल्याच्या समतुल्य आहे. त्याचप्रमाणे जर बायनरी संख्या एका स्थानाने उजवीकडे शिफ्ट केली तर ती मूळ संख्येला 2 ने भागल्यास समान आहे.

म्हणून जर आपल्याला दिलेल्या बायनरी संख्येचा गुणाकार आणि भागाकार करण्यासाठी शिफ्ट रजिस्टर वापरायचे असेल, तर आपल्याला डेटा डावीकडे किंवा उजवीकडे शिफ्ट करता आला पाहिजे.

अशा रजिस्टरला (आकृती 4.27) बायडीरेक्शनल शिफ्ट रजिस्टर म्हणतात.

मोड सिलेक्ट इनपुट (M) सह, दोन सिरीयल इनपुट आहेत. i) सीरियल राइट शिफ्ट डेटा इनपुट DR ii) सीरियल लेफ्ट शिफ्ट डेटा इनपुट DL.





आकृती 4.27 बायडीरेकशनल शिफ्ट रजिस्टर

**कार्यपद्धतः--**  $M = 1$ : राईट शिफ्ट: जेव्हा  $M = 1$  असल्यास, अँड गेट्स 1,3,5 आणि 7 सक्रम आहेत तर उर्वरित अँड गेट्स 2,4,6 आणि 8 अक्षम होतात.

क्लॉक पल्स आल्यावर DR इनपुट वरील डेटा FF3 पासून FF0 वर उजवीकडे शिफ्ट केला जातो. अशाप्रकारे आपल्याला सिरियल राईट शिफ्ट ऑपरेशन मिळते.

$M = 0$ :- लेफ्ट शिफ्ट ऑपरेशन:

जेव्हा मोड कंट्रोल  $M$  0 शी कनेक्ट केले जाते तेव्हा अँड गेट्स 2,4,6 आणि 8 सक्रम केले जातात तर 1,3,5 आणि 7 अक्षम केले जातात.

क्लॉक पल्स आल्यावर DL इनपुट वरील डेटा FF0 पासून FF3 वर डावीकडे शिफ्ट केला जातो. अशाप्रकारे सिरियल राईट शिफ्ट ऑपरेशन मिळते.

**4.6.6 युनिव्हर्सल शिफ्ट रजिस्टर (Universal Shift Register)** - शिफ्ट रजिस्टर जे डेटा दोन्ही दिशांना शिफ्ट करू शकते तसेच समांतर लोड करू शकते, त्याला युनिव्हर्सल शिफ्ट रजिस्टर म्हणून ओळखले जाते. शिफ्ट रजिस्टर खालील ऑपरेशन करण्यास सक्षम आहे.

समांतर लोडिंग: डावीकडे स्थलांतर, उजवीकडे स्थलांतर.

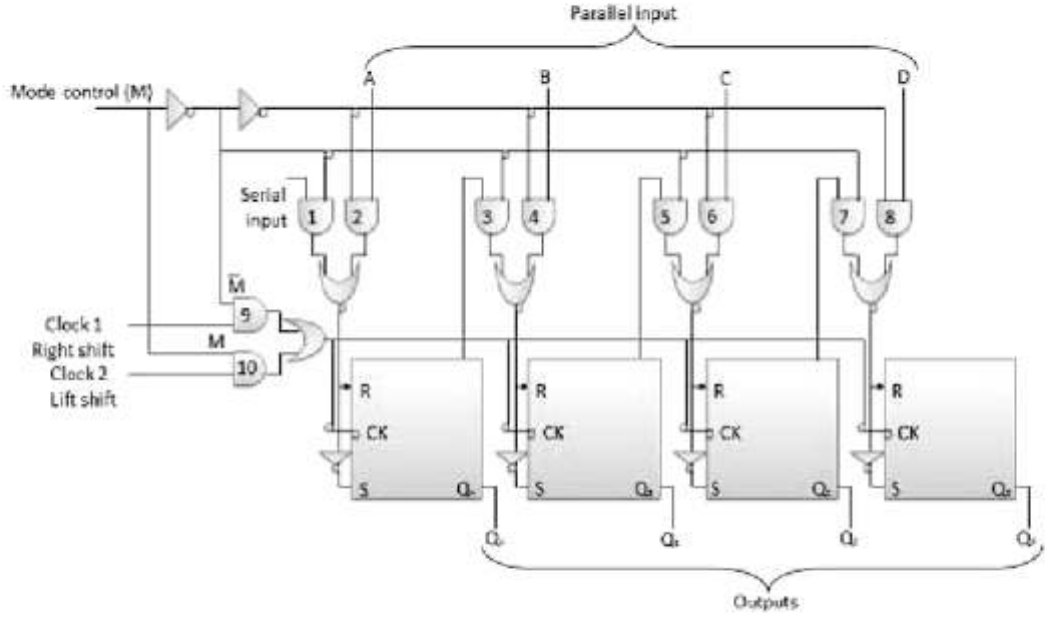
मोड कंट्रोल इनपुट समांतर लोडिंग ऑपरेशनसाठी लॉजिक 1 शी कनेक्ट केलेले आहे तर सीरियल शिफ्टिंगसाठी ते 0 शी कनेक्ट केलेले आहे.

मोड कंट्रोल पिन ग्राउंड जोडलेला असताना, युनिव्हर्सल शिफ्ट रजिस्टर बायडीरेकशनल शिफ्ट रजिस्टर म्हणून काम करते.

**समांतर लोडिंग (Paralle loading)**- जेव्हा मोड कंट्रोल M लॉजिक 1 शी कनेक्ट केले जाते, आणि गेट्स 2,4,6,8 सक्षम केले जातील आणि गेट्स 1,3,5,7 अक्षम केले जातील. 4 बिट बायनरी डेटा समांतर लोड केला जाईल. क्लॉक-2 इनपुट फ्लिप-फ्लॉपवर लागू केले जाईल. M=1 असल्याने, आणि गेट 10 सक्षम केले जाईल आणि 9 अक्षम केले जाईल. D इनपुट समांतर डेटा Q आउटपुटमध्ये जातील.

**लेफ्ट शिफ्ट (Left Shift)**- जेव्हा मोड नियंत्रण लॉजिक 0 शी कनेक्ट केले जाते, आणि गेट्स 1,3,5,7 सक्षम केले जातील आणि 2,4,6,8 अक्षम केले जातील. डेटा क्रमाने शिफ्ट केला जाईल. क्लॉक इनपुट फ्लिप-फ्लॉपच्या क्लॉक-1 वर लागू केले जाईल. आकृती 4.28 मध्ये दर्शविलेल्या अँड गेट-1 वर इनपुट लागू केले जाते. M=0 असल्याने, आणि गेट 9 सक्षम केले जाईल आणि गेट 10 अक्षम केले जाईल. डेटा QA वरून QD वर क्रमाने उजवीकडे शिफ्ट होईल.

**उजवीकडे शिफ्ट (Right Shift)**- जेव्हा मोड नियंत्रण M लॉजिक 1 शी कनेक्ट केले जाते, आणि गेट्स 2,4,6,8 सक्षम केले जातील आणि गेट्स 1,3,5,7 अक्षम केले जातील. हे रजिस्टरचे समांतर लोडिंग आणि डाव्या शिफ्ट ऑपरेशनला परवानगी देते. प्रत्येक फ्लिप-फ्लॉपचे आउटपुट मागील फ्लिप-फ्लॉपच्या समांतर इनपुटशी जोडून लेफ्ट शिफ्ट ऑपरेशन पूर्ण केले जाते आणि D-इनपुटवर क्रमाक्रमाने इनपुट लागू केले जाते.



आकृती 4.28 युनिव्हर्सल शिफ्ट रजिस्टर

1. शिफ्ट रजिस्टर हे कॅल्क्युलेटर, संगणक आणि डेटा-प्रोसेसिंग सिस्टममध्ये आढळणारे डिजिटल मेमरी सर्किट आहे.
2. शिफ्ट रजिस्टर डेटा डावीकडे आणि/किंवा उजवीकडे तसेच क्रमिक आणि समांतर ऑपरेशनमध्ये शिफ्ट करण्यास वापरतात.

**4.6.7 काउंटर्स (Counters)** - क्लॉक इनपुटची संख्या मोजण्यासाठी वापरल्या जाणाऱ्या सर्किटला काउंटर असे म्हणतात. दोन प्रकारचे काउंटर आहेत.

1. असिंक्रोनस (Asynchronous counter) काउंटर किंवा रिपल (ripple Counter) काउंटर- या काउंटरमध्ये सर्व फ्लिप-फ्लॉपना एकाच वेळी क्लॉक दिलेली नसते.
2. सिंक्रोनस काउंटर (Synchronous counter) - या काउंटरमध्ये सर्व फ्लिप-फ्लॉपना एकाच वेळी क्लॉक दिलेली असतात.

ॲप्लिकेशन्स-

1. वॉशिंग मशिनमध्ये, सेट केलेला टाइम मोजतो.
2. स्टेडियम किंवा सभागृहात प्रवेश करणाऱ्या आणि सोडणाऱ्या लोकांच्या संख्येची गणना,
3. पार्किंगमध्ये वाहने मोजण्यासाठी देखील वापरला जाऊ शकतो.
4. डिजिटल इलेक्ट्रॉनिक सर्किट्समध्ये, काउंटर म्हणून वापरले जातात

5. फ्रिक्वेन्सी काउंटर/विभाजक,

6. डिजिटल घड्याळे.

4.6.7.1 सिंक्रोनस आणि असिंक्रोनस काउंटर्स मधील फरक (Comparision between synchronous and asynchronous counters)

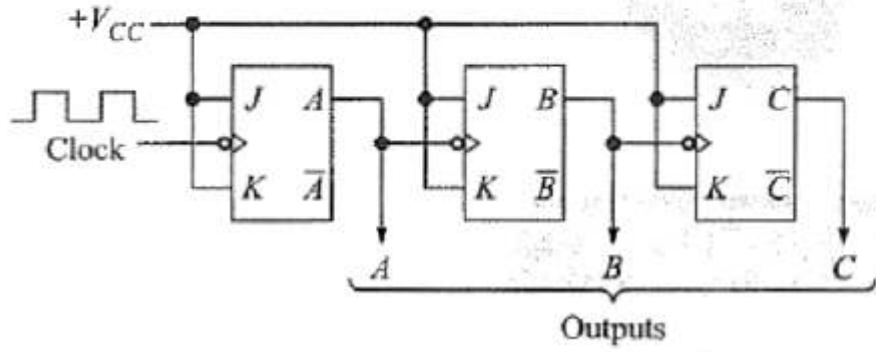
तक्ता क्र. 4.8

सिंक्रोनस	असिंक्रोनस
प्रसार विलंब (propagation delay) खूप कमी आहे.	सिंक्रोनस काउंटरच्या तुलनेत प्रसार विलंब जास्त आहे.
हे असिंक्रोनस काउंटरपेक्षा वेगवान आहेत.	यांचे कार्य सिंक्रोनस काउंटरपेक्षा हळू चालते.
डिझाइन करण्यासाठी मोठ्या प्रमाणात लॉजिक गेट्स आवश्यक आहेत.	लॉजिक गेट्सची कमी संख्या आवश्यक आहे.
जास्त किंमत.	कमी खर्च.

4.6.7.2 3 बिट असिंक्रोनस काउंटर (3-bit synchronous counter)- जेव्हा फ्लिप-फ्लॉपचे आउटपुट पुढील फ्लिप-फ्लॉपसाठी क्लॉक इनपुट म्हणून वापरले जाते, तेव्हा आपण काउंटरला रिपल काउंटर किंवा असिंक्रोनस काउंटर म्हणतो.

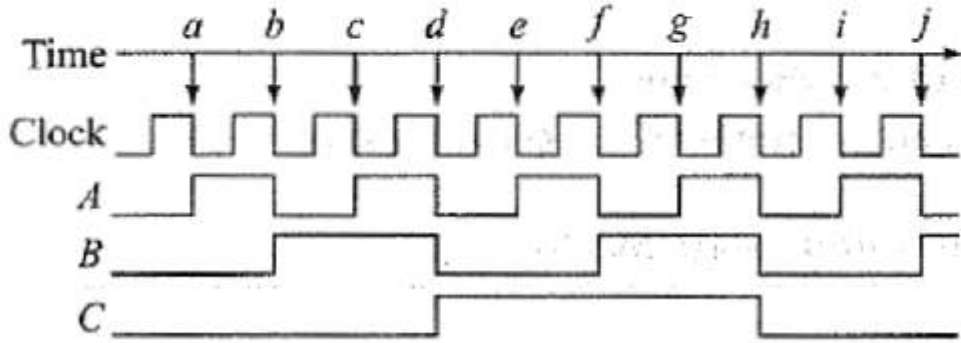
आकृती 4.29 तीन फॉलिंग एज-ट्रिगर केलेले, J-K फ्लिप-फ्लॉप कॅस्केडमध्ये जोडलेले दाखवते. क्लॉक, फ्लिप-फ्लॉप A ड्राइव्ह करते. A चे आउटपुट B आणि B चे आउटपुट फ्लिप-फ्लॉप C. सर्व J आणि K इनपुट +Vcc शी जोडलेले असतात. याचा अर्थ प्रत्येक फ्लिप-फ्लॉप त्याच्या क्लॉक इनपुटवर फॉलिंग एज-ट्रिगर नुसार बदलेल (टॉगल).





आकृती 4.29 3 बिट असिंक्रोनस काउंटर

दिलेली वेव्हफॉर्मस क्लॉक चालू असताना काउंटरची क्रिया दर्शवतात. असे गृहीत धरू की फ्लिप-फ्लॉप सुरुवातीला 0 आउटपुट तयार करण्यासाठी रीसेट केले आहेत. जर आपण A ला LSB बिट आणि C ला MSB बिट मानले, तर आपण काउंटरची स्थिती CBA = 000 आहे असे म्हणू शकतो.



आकृती 4.30 वेव्हफॉर्मस (टाईमिंग डायग्राम)

प्रत्येक क्लॉक फॉलिंग वेळी फ्लिप-फ्लॉप A संक्रमण होते, स्थिती बदलेल. हे टाइमलाइनवरील लहान खाली बाणांनी सूचित केले आहे. अशा प्रकारे टाइम रेषेवर a बिंदूवर, A वर जातो व b बिंदूवर तो परत खाली येतो, c वर तो परत उंचावर जातो, इत्यादी. लक्षात घ्या की फ्लिप-फ्लॉप A च्या आउटपुटवरील वेव्हफॉर्म ही क्लॉक वारंवारताच्या (फ्रीक्वेन्सीच्या) अर्धा आहे.

A हे B साठी क्लॉक म्हणून काम करत असल्याने, प्रत्येक वेळी A वरील वेव्हफॉर्म कमी झाल्यावर, फ्लिप-फ्लॉप B टॉगल होईल. अशा प्रकारे टाइम रेषेवर b बिंदूवर, B वर जातो; ते नंतर बिंदू d वर खाली येतो आणि बिंदू f वर पुन्हा वर टॉगल करते. लक्षात घ्या की फ्लिप-फ्लॉप B च्या आउटपुटवरील वेव्हफॉर्म A च्या अर्धा फ्रीक्वेन्सी आणि क्लॉकच्या एकचतुर्थांश फ्रीक्वेन्सी आहे. B साठी क्लॉक म्हणून C कार्य करते, प्रत्येक वेळी B वरील वेव्हफॉर्म कमी झाल्यावर, फ्लिप-फ्लॉप C

टॉगल होईल. अशा रीतीने C हा बिंदूच्या वेळी वर जातो आणि h बिंदूवर पुन्हा कमी होतो. C वरील वारंवारता B पेक्षा अर्धा आहे, परंतु ती टाइम वारंवारतेच्या फक्त एक-अष्टमांश आहे.

तीन-फ्लिप-फ्लॉप काउंटरमध्ये  $2^3 = 8$  आउटपुट (000 ते 111) आहे. पाच फ्लिप-फ्लॉपमध्ये  $2^5 = 32$  आउटपुट (00000 ते 11111) असेल, आणि असेच. सर्वात मोठी बायनरी संख्या जी n कॅस्केड फ्लिप-फ्लॉपद्वारे दर्शविली जाऊ शकते ती  $(2^n - 1)$  च्या दशांश (decimal) समतुल्य आहे.

उदाहरणार्थ, तीन-फ्लिप-फ्लॉप काउंटर  $(2^3 - 1)$ च्या कमाल दशांश संख्येपर्यंत पोहोचतो. पाच फ्लिप-फ्लॉपसाठी कमाल दशांश संख्या  $(2^5 - 1 = 31)$  आहे, तर सहा फ्लिप-फ्लॉपची कमाल संख्या 63 आहे.

तक्ता क्र. 4.9

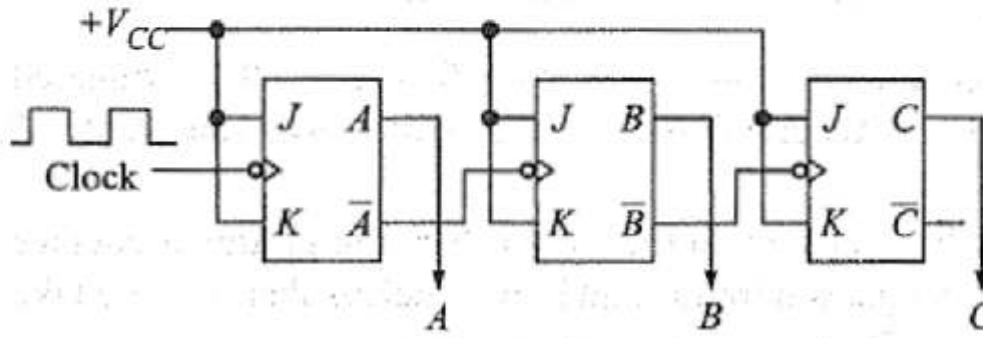
Negative clock transitions	C	B	A	State or count
---	0	0	0	0
a	0	0	1	1
b	0	1	0	2
c	0	1	1	3
d	1	0	0	4
e	1	0	1	5
f	1	1	0	6
g	1	1	1	7
h	0	0	0	0

**4.6.8 काउंटरचे मॉड्यूलस (Modulus of counter)-** काउंटर किती स्थिती मधून जातो ती संख्या. तीन-फ्लिप-फ्लॉप काउंटरला अनेकदा मॉड्यूलस-8 (किंवा मॉड-8) काउंटर म्हणून संबोधले जाते कारण त्याच्या आठ स्थिती असतात.

त्याचप्रमाणे, चार-फ्लिप-फ्लॉप काउंटर मॉड-16 काउंटर आहे, आणि सहा-फ्लिप-फ्लॉप काउंटर मॉड-64 काउंटर आहे. काउंटरचे मॉड्यूलस म्हणजे एकूण किती स्थिती मधून आउटपुट बदलते ती संख्या होय.

**4.6.9 असिंक्रोनस 3 बिट डाउन काउंटर (3-bit Asynchronous down counter) -** क्लॉक इनपुट फ्लिप-फ्लॉप A ला दिला जातो. परंतु A' आउटपुट (पूरक), B फ्लिपफ्लॉपला त्याचप्रमाणे; B' आउटपुट (पूरक), फ्लिप-फ्लॉप Cला दिला जातो.





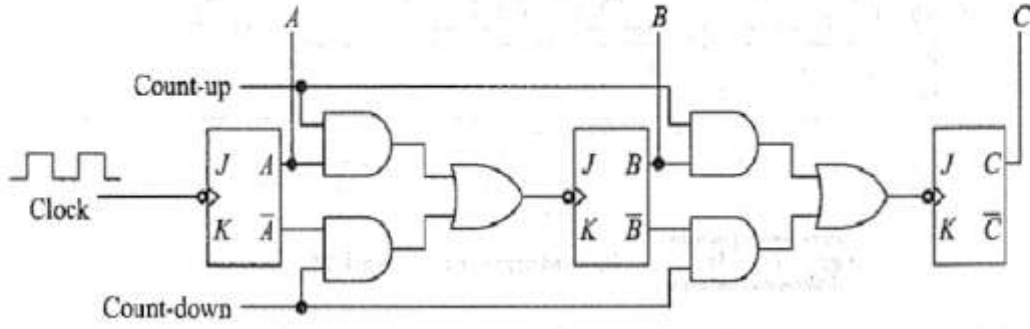
आकृती 4.31 असिंक्रोनस 3 बिट डाउन काउंटर

फ्लिप-फ्लॉप A पूर्वीप्रमाणेच प्रत्येक फॉलिंग क्लॉक संक्रमणाला टॉगल करतो. पण प्रत्येक वेळी A हाय वर गेल्यावर फ्लिप-फ्लॉप B टॉगल करेल. लक्षात घ्या की प्रत्येक वेळी A वर जातो, A खाली येतो आणि A वरील हे फॉलिंग संक्रमण B ला ट्रिगर करते. टाइम लाइन वर, B a, c, e, g आणि i बिंदूवर टॉगल करते.

त्याचप्रमाणे, फ्लिप-फ्लॉप C ला B द्वारे चालना दिली जाते आणि प्रत्येक वेळी B हाय झाल्यावर C टॉगल करेल. अशा प्रकारे C टाइम लाइन वर बिंदू a वर हाय टॉगल करतो, बिंदू e वर लो, मागे टॉगल करतो आणि बिंदू i वर पुन्हा हाय वर जातो.

काउंटरची टाइम लाइन वर बिंदू a वर ABC= 111 बनते, बिंदू b वर 110 बनते आणि बिंदू c वर 101 बनते. लक्षात घ्या की प्रत्येक क्लॉक च्या संक्रमणसह काउंटर एका संख्येने कमी होते. दुसऱ्या शब्दांत, काउंटर काउंट-डाउन मोडमध्ये कार्यरत आहे. हे मॉड-8 काउंटर आहे, कारण त्यात आठ स्वतंत्र अवस्था आहेत,

**4.6.10 3 बिट असिंक्रोनस अप-डाउन काउंटर (3-bit Asynchronous UP-Down counter))-** 3-बिट असिंक्रोनस अप-डाउन काउंटर बायनरी अनुक्रमात मोजले जाते. या काउंटरला काउंट-अप क्रमाने प्रगती करण्यासाठी, प्रत्येक फ्लिप-फ्लॉपला मागील फ्लिप-फ्लॉपच्या खऱ्या बाजूने (पूरक बाजूच्या विरुद्ध) ट्रिगर करणे आवश्यक आहे. जर काउंट-डाउन कंट्रोल लाईन लो असेल आणि काउंट-अप कंट्रोल लाईन हाय असेल तर काउंटरमध्ये काउंट-अप मोडमध्ये असेल आणि वेव्हफॉर्म नुसार प्रगती करेल.



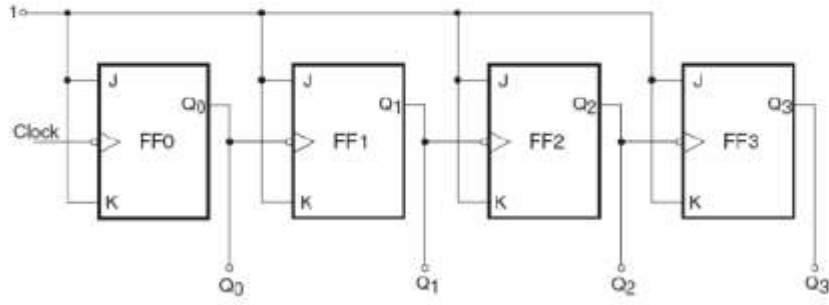
Note: The J and K inputs are all tied to  $+V_{CC}$   
The counter outputs are A, B, and C.

#### आकृती 4.32 3-बिट असिंक्रोनस अप-डाऊन काउंटर

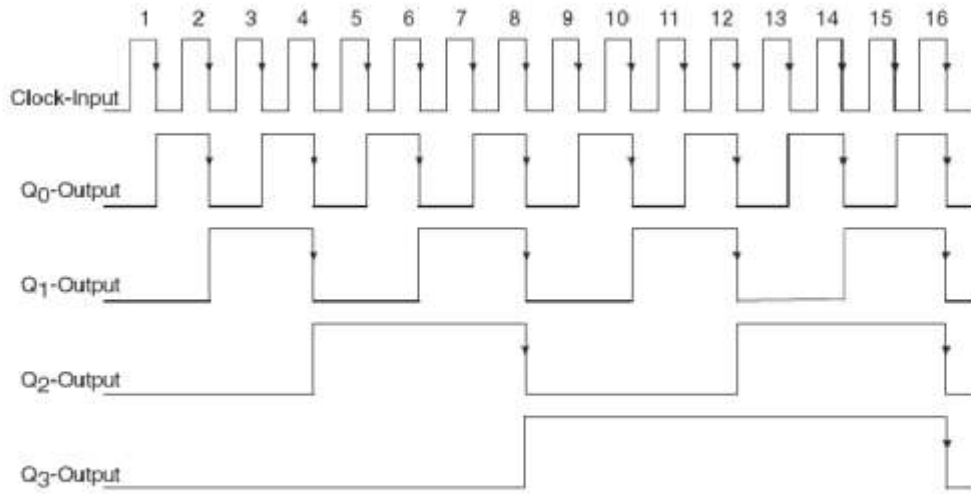
जर काउंट-डाऊन हाय असेल आणि काउंट-अप लो असेल, तर प्रत्येक फ्लिप-फ्लॉप मागील फ्लिप-फ्लॉपच्या पूरक बाजूपासून ट्रिगर केला जाईल. काउंटर काउंट-डाऊन मोडमध्ये असेल आणि प्रगती करेल.

**4.6.11 4 बिट बायनरी रिपल काउंटर (4-bit binary ripple counter)-** हे J-K फ्लिप-फ्लॉप टॉगल फ्लिप-फ्लॉप मोडमध्ये आहे आणि फॉलिंग एज-ट्रिगर आहे. पहिल्या फ्लिप-फ्लॉपचे आउटपुट दुसऱ्याचे क्लॉक इनपुटला दिले जाते आणि दुसऱ्या फ्लिप-फ्लॉपचे आउटपुट तिसऱ्याचे क्लॉक इनपुटला दिले जाते. तिसऱ्याचे आउटपुट चौथ्या फ्लिप-फ्लॉपचे क्लॉक इनपुटला दिले जाते. चार फ्लिप-फ्लॉपचे आउटपुट Q0 (LSB) फ्लिप-फ्लॉप, Q1, Q2 आणि Q3 (MSB) फ्लिप-फ्लॉप्स आहेत. आकृती 4.33 Q0, Q1, Q2 आणि Q3 आउटपुटवर दिसणाऱ्या वेव्हफॉर्म्स दाखवते.

**कार्यपद्धत:-** सुरुवातीला सर्व फ्लिप-फ्लॉप्स '0' स्थितीत केले आहेत. पहिल्या क्लॉकच्या हाय-ते-लो संक्रमण वर, टॉगलिंग क्रियेमुळे Q0 '0' वरून '1' वर जातो. सर्व फ्लिप-फ्लॉप हे फॉलिंग एज-ट्रिगर असल्याने, Q0 चे '0' ते '1' फ्लिप-फ्लॉप FF1 ट्रिगर करत नाही. FF1, FF2 आणि FF3 '0' स्थितीत राहते. तर, पहिल्या क्लॉक फॉलिंग एज-ट्रिगर वर, Q0=1, Q1=0, Q2=0 आणि Q3=0 करते. दुसऱ्या क्लॉकच्या हाय-ते-लो वर, Q0 पुन्हा टॉगल करते. म्हणजेच ते '1' वरून '0' पर्यंत जाते.



आकृती 4.33 4-बिट रिपल काउंटर व टाइमिंग डायग्राम



Q0 आउटपुटवर हे '1' ते '0' FF1 ट्रिगर करते, ज्याचे आउटपुट Q1 '0' वरून 1 जाते. Q2 आणि Q3 आउटपुट अप्रभावित राहतात. म्हणून, क्लॉक सिग्नलचे दुसऱ्या हाय-ते-लो संक्रमण क्लॉक नंतर लगेच,  $Q_0 = 0$ ,  $Q_1 = 1$ ,  $Q_2 = 0$  आणि  $Q_3 = 0$ .

त्याचप्रमाणे, त्यानंतरच्या क्लॉक संक्रमण नंतर लगेचच  $Q_0$ ,  $Q_1$ ,  $Q_2$  आणि  $Q_3$  आउटपुटची लॉजिक स्थिती तक्ता 4.10 मध्ये सारांशित केली आहे.

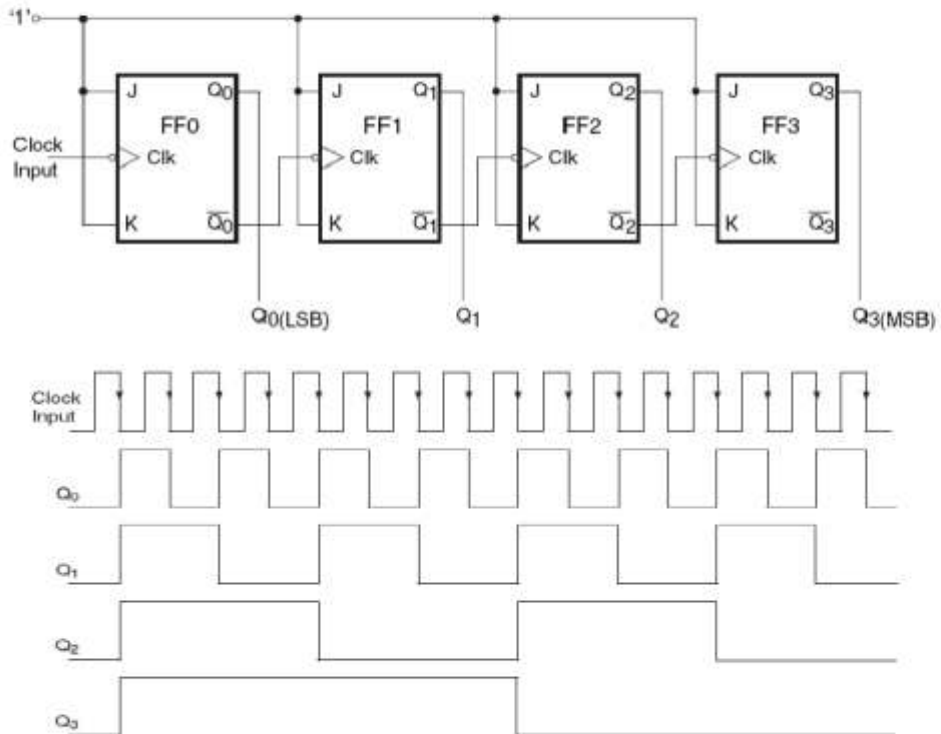
अशाप्रकारे, आपण पाहतो की काउंटर 0000 ते 1111 पर्यंत 16 भिन्न अवस्थांमधून जातो आणि नंतर, सोळाव्या क्लॉकला ते 0000 च्या मूळ स्थितीवर रीसेट होते जिथून ते सुरु झाले होते. सर्वसाधारणपणे, जर N फ्लिप-फ्लॉप असेल तर,  $2^N$  पल्स मोजू शकतो.

काउंटर प्रारंभिक स्थितीवर रीसेट करते. आपण पाहू शकतो की,  $Q_0$ ,  $Q_1$ ,  $Q_2$  आणि  $Q_3$  वारंवारता, अनुक्रमे  $f/2$ ,  $f/4$ ,  $f/8$  आणि  $f/16$  आहेत. येथे,  $f$  ही क्लॉक इनपुटची वारंवारता आहे. हे लक्षात घेतले जाऊ शकते की वारंवारता डिव्हिजन काउंटरच्या प्रमुख ॲप्लिकेशन आहे.

तक्ता क्र. 4.10

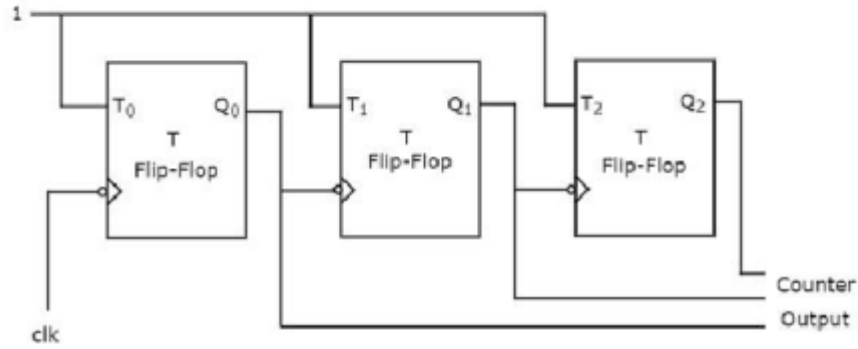
Clock	$Q_3$	$Q_2$	$Q_1$	$Q_0$
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

#### 4.6.12 4 बिट डाउन काउंटर (4-bit down counter)



आकृती 4.34. 4 बिट डाउन काउंटर व टाइमिंग डायग्राम

#### 4.6.13 T फ्लिप-फ्लॉप वापरून असिंक्रोनस बायनरी अप काउंटर (Asynchronous binary UP counter using TFF)



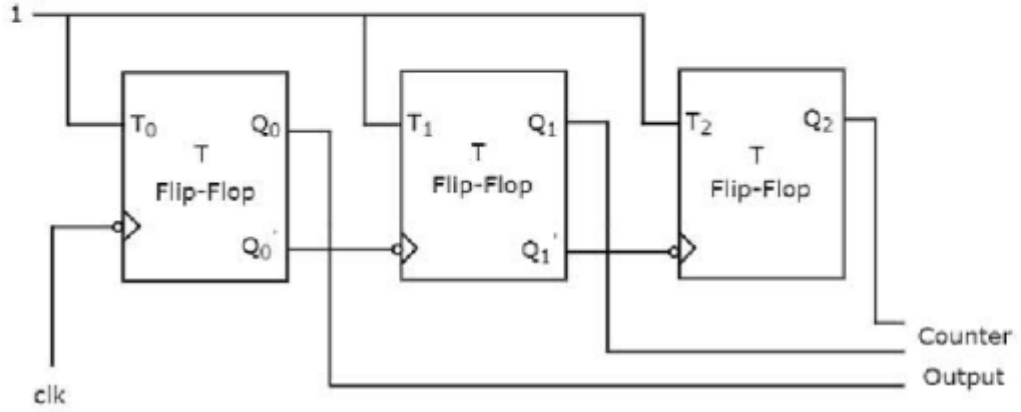
आकृती 4.34.5 T फ्लिप-फ्लॉप वापरून असिंक्रोनस बायनरी अप काउंटर

3-बिट असिंक्रोनस बायनरी अप काउंटरमध्ये तीन T फ्लिप-फ्लॉप आहेत आणि सर्व फ्लिप-फ्लॉपचे T-इनपुट '1' शी जोडलेले आहेत. हे सर्व फ्लिप-फ्लॉप फॉलिंग एज ट्रिगर केले जातात. परंतु आउटपुट असिंक्रोनस बदलतात. क्लॉक सिग्नल पहिल्या T फ्लिप-फ्लॉपवर लागू केला जातो. त्यामुळे, क्लॉक सिग्नलच्या प्रत्येक फॉलिंग एज साठी पहिल्या T फ्लिप-फ्लॉपचे आउटपुट टॉगल होते.

पहिल्या T फ्लिप-फ्लॉपचे आउटपुट दुसऱ्या T फ्लिप-फ्लॉपसाठी क्लॉक सिग्नल म्हणून लागू केले जाते. तर, दुसऱ्या T फ्लिप-फ्लॉपचे आउटपुट पहिल्या T फ्लिप-फ्लॉपच्या आउटपुटच्या प्रत्येक फॉलिंग एज साठी टॉगल होते. त्याचप्रमाणे, तिसऱ्या T फ्लिप-फ्लॉपचे आउटपुट दुसऱ्या T फ्लिप-फ्लॉपच्या आउटपुटच्या प्रत्येक फॉलिंग एज साठी टॉगल करते, कारण दुसऱ्या T फ्लिप-फ्लॉपचे आउटपुट तिसऱ्या T फ्लिप-फ्लॉपसाठी क्लॉक सिग्नल म्हणून कार्य करते.

#### 4.6.14 T फ्लिप-फ्लॉप वापरून 3 बिट असिंक्रोनस डाउन काउंटर (3-bit Asynchronous binary down counter using TFF)



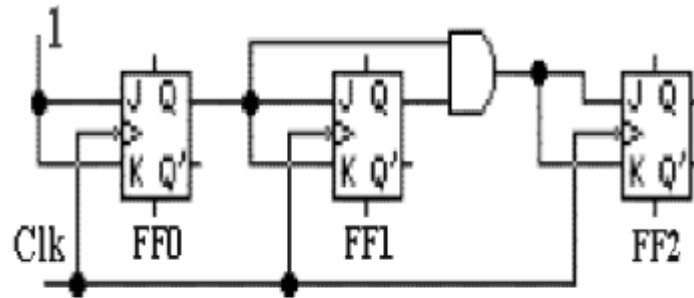


आकृती 4.35 फ्लिप-फ्लॉप वापरून 3 बिट असिंक्रोनस डाउन काउंटर

3-बिट असिंक्रोनस बायनरी डाउन काउंटरचा आकृती, 3-बिट असिंक्रोनस बायनरी अप काउंटरच्या आकृती प्रमाणे आहे. परंतु, फरक एवढाच आहे की पुढील स्टेज फ्लिप-फ्लॉपसाठी क्लॉक सिग्नल म्हणून मागील स्टेज फ्लिप-फ्लॉपचे पूरक आउटपुट कनेक्ट केला जातो.

**4.6.15 सिंक्रोनस काउंटर (Synchronous counter):** काउंटर सर्किट ज्यामध्ये सर्व फ्लिप-फ्लॉपना एकाच वेळी क्लॉक दिला आहे.

**4.6.16 J-K फ्लिप-फ्लॉप वापरून 3 बिट सिंक्रोनस काउंटर (3-bit Synchronous counter using JKFF)**

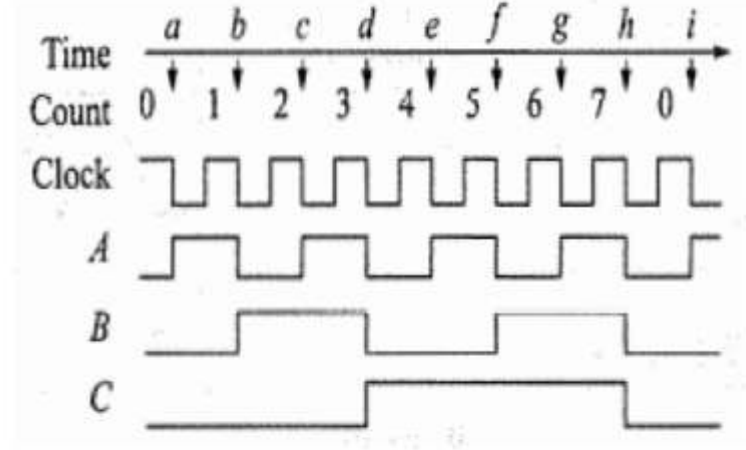


आकृती 4.36 J-K फ्लिप-फ्लॉप वापरून 3 बिट सिंक्रोनस काउंटर

क्लॉक पल्स प्रत्येक J-K फ्लिप-फ्लॉपला दिल्या जातात. दोन्ही J आणि K इनपुट टॉगल मोडमध्ये जोडले आहेत, परंतु फक्त पहिला फ्लिप-फ्लॉप FF0 (LSB) लॉजिक "1" कनेक्ट केलेला आहे. यामुळे

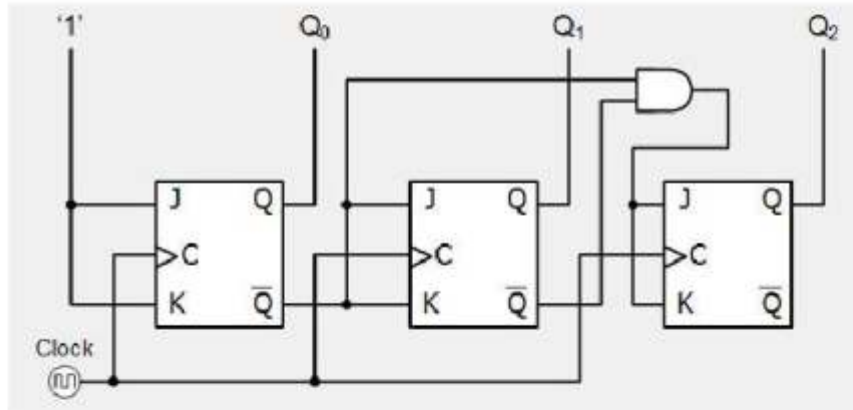


FF0 फ्लिप-फ्लॉप प्रत्येक क्लॉकच्या पल्सवर टॉगल होतो. फ्लिप-फ्लॉप FF1 चे J आणि K इनपुट, फ्लिप-फ्लॉप FF0 च्या Q0 आउटपुटशी जोडलेले आहेत, तसेच फ्लिप-फ्लॉप FF2 चे J आणि K इनपुट, फ्लिप-फ्लॉप FF0 आणि FF1 च्या आउटपुटशी अँड गेट मधून जोडलेले आहेत.



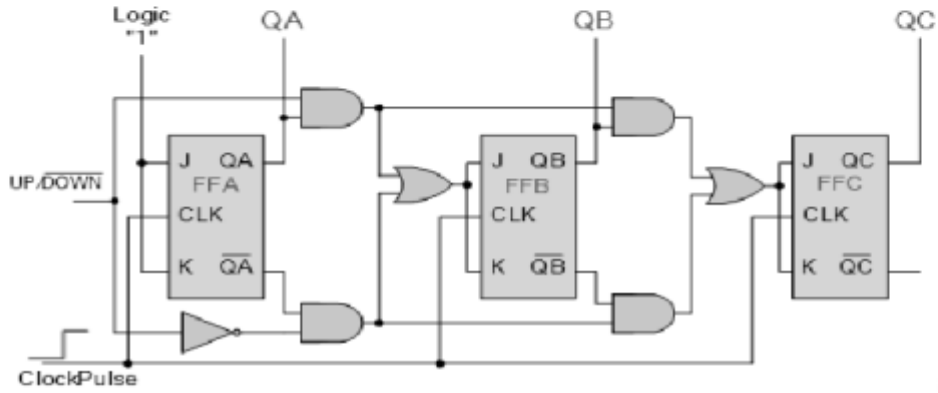
आकृती 4.36.1- 3 बिट सिंक्रोनस काउंटर वेव्हफॉर्म्स (टाईमिंग डायग्राम)

**4.6.17 J-K फ्लिप-फ्लॉप वापरून 3 बिट सिंक्रोनस डाउन काउंटर (3-bit Synchronous down counter using JKFF)**



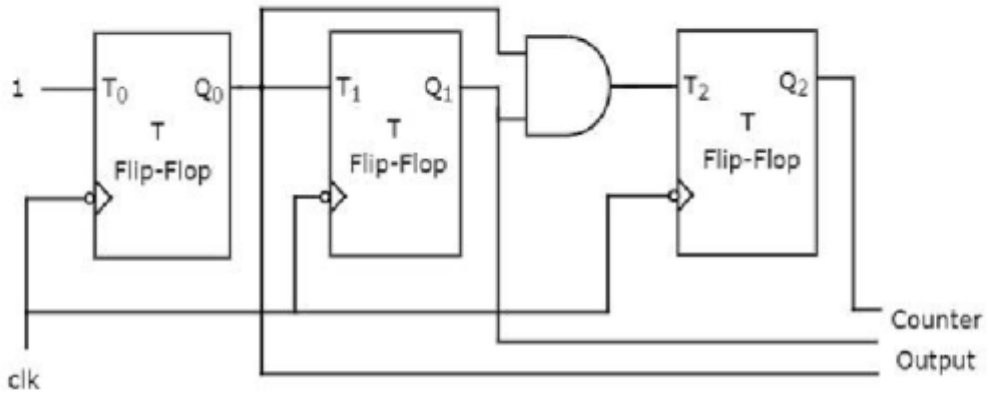
आकृती 4.37 J-K फ्लिप-फ्लॉप वापरून 3 बिट सिंक्रोनस डाउन काउंटर

#### 4.6.18 3 बिट सिंक्रोनस अप/डाउन काउंटर



आकृती 4.38 3 बिट सिंक्रोनस अप/डाउन काउंटर

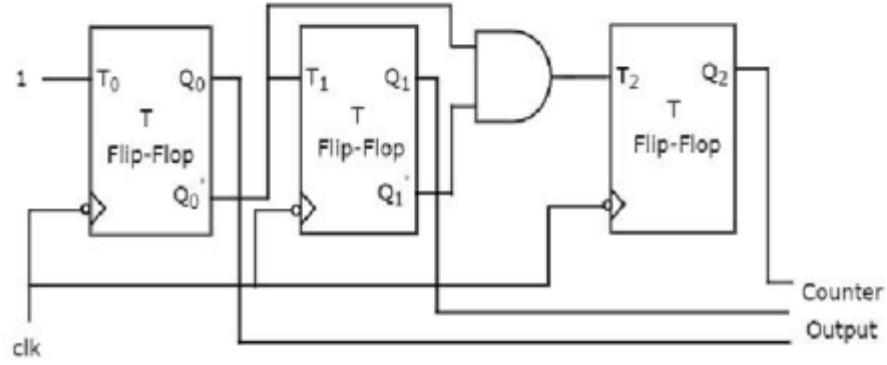
#### 4.6.19 T फ्लिप-फ्लॉप वापरून 3 बिट सिंक्रोनस अप काउंटर (3-bit Synchronous UP counter using TFF)



आकृती 4.39 फ्लिप-फ्लॉप वापरून 3 बिट सिंक्रोनस अप काउंटर

3-बिट सिंक्रोनस बायनरी अप काउंटरमध्ये तीन T फ्लिप-फ्लॉप आणि एक 2-इनपुट अँड गेट आहे. हे सर्व फ्लिप-फ्लॉप फॉलिंग एज ट्रिगर असतात आणि फ्लिप-फ्लॉपचे आउटपुट एकास वेळी बदलतात. पहिल्या फ्लिप-फ्लॉपचे T इनपुट = 1, दुसऱ्या फ्लिप-फ्लॉपचे T इनपुट = Q0 आणि तिसऱ्या फ्लिप-फ्लॉपचे T इनपुट, Q0.Q1 आहेत. पहिल्या T फ्लिप-फ्लॉपचे आउटपुट क्लॉक सिग्नलच्या प्रत्येक फॉलिंग एजला टॉगल करते. जर Q0=1 असेल तर दुसऱ्या T फ्लिप-फ्लॉपचे आउटपुट क्लॉक सिग्नलच्या प्रत्येक फॉलिंग एजला टॉगल करते.

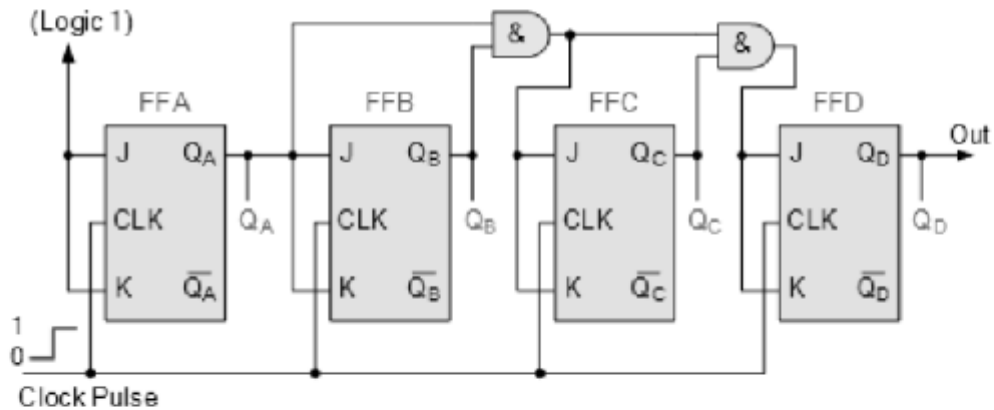
#### 4.6.20 T फ्लिप-फ्लॉप वापरून 3 बिट सिंक्रोनस डाउन काउंटर (3-bit Synchronous down counter using TFF)



आकृती 4.40 T फ्लिप-फ्लॉप वापरून 3 बिट सिंक्रोनस डाउन काउंटर

पहिल्या T फ्लिप-फ्लॉपचे आउटपुट क्लॉक सिग्नलच्या प्रत्येक फॉलिंग एजला टॉगल करते. जर  $Q_0 = 1$  असेल तर क्लॉक सिग्नलच्या फॉलिंग एजला दुसरा T फ्लिप-फ्लॉप आउटपुट टॉगल होते. जर  $Q_1$  आणि  $Q_0$  दोन्ही 1 असतील तर तिसऱ्या T फ्लिप-फ्लॉपचे आउटपुट क्लॉक सिग्नलच्या फॉलिंग एजला टॉगल होते.

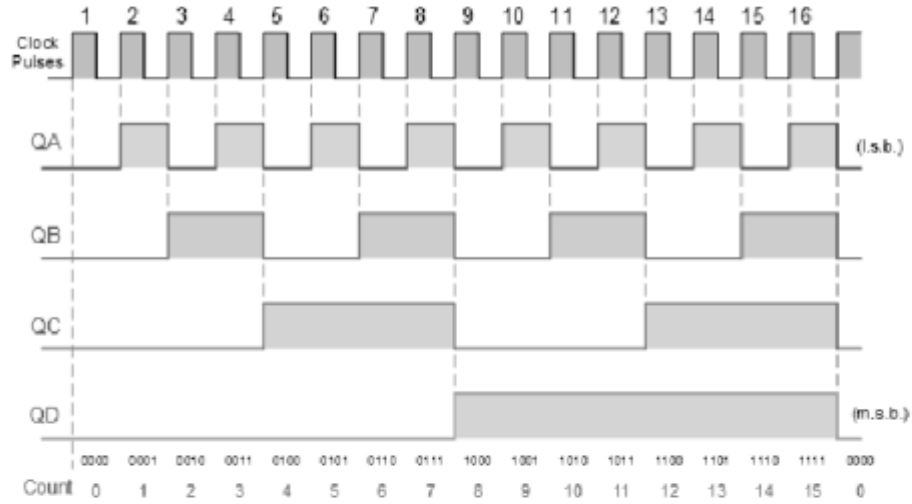
#### 4.6.21 4-बिट सिंक्रोनस अप काउंटर (4-bit Synchronous UP counter)



आकृती 4.41 4-बिट सिंक्रोनस अप काउंटर

क्लॉक पल्स प्रत्येक J-K फ्लिप-फ्लॉपला दिल्या जातात. दोन्ही इनपुट, J आणि K टॉगल मोडमध्ये आहेत. परंतु फक्त पहिला फ्लिप-फ्लॉप (FFA) लॉजिक "1" मध्ये कनेक्ट केलेला आहे, ज्यामुळे

फ्लिप-फ्लॉप प्रत्येक क्लॉक पल्सवर टॉगल करते. फ्लिप-फ्लॉप FFB चे J आणि K इनपुट फ्लिप-फ्लॉप FFA च्या QA आउटपुट शी जोडलेले आहेत, परंतु फ्लिप-फ्लॉप FFC आणि FFD चे J आणि K इनपुट हे अँड गेट्स मधून येतात.

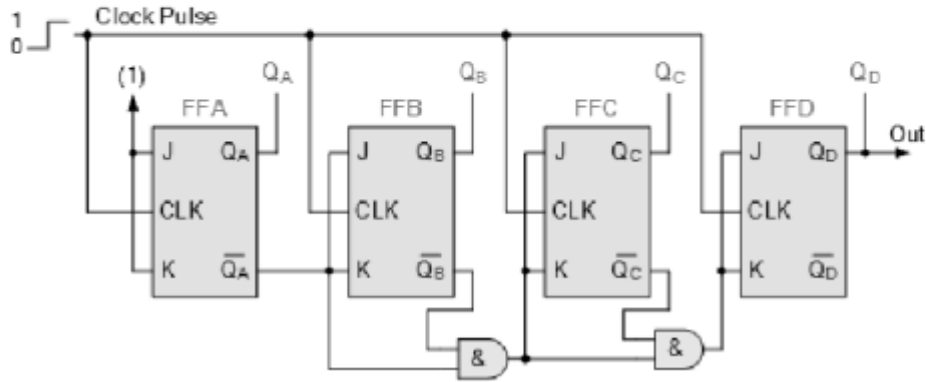


आकृती 4.42 4-बिट सिंक्रोनस काउंटर वेव्हफॉर्म

हे 4-बिट सिंक्रोनस काउंटर प्रत्येक क्लॉकच्या पल्सवर क्रमाने मोजले जाते व आउटपुट 0 (0000) ते 15 (1111) पर्यंत मोजले जातात. म्हणून, या प्रकारच्या काउंटरला 4-बिट सिंक्रोनस अप काउंटर असेही म्हणतात.

#### 4.6.22 4-बिट सिंक्रोनस डाउन काउंटर (4-bit Synchronous down counter)

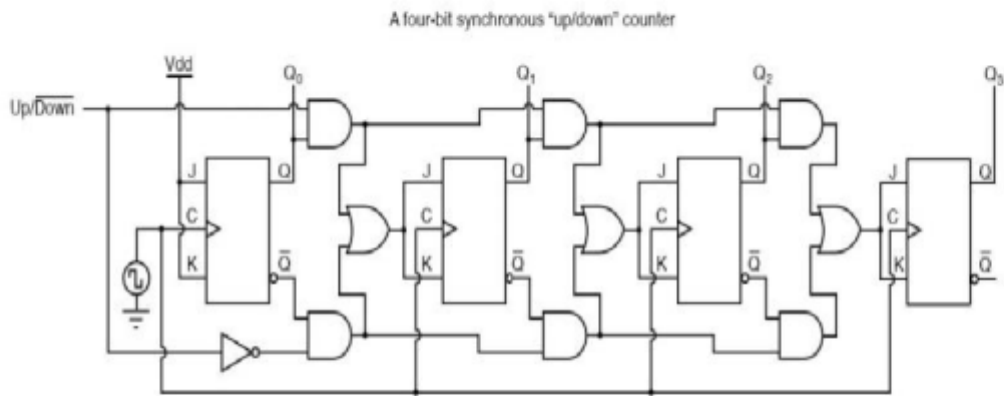
अँड गेट्सला फ्लिप-फ्लॉपच्या Q बार आउटपुटशी जोडलेले आहेत. येथे काउंटर त्याच्या सर्व आउटपुट हाय (1111) ने सुरू होते आणि ते पुन्हा पुनरावृत्ती होण्यापूर्वी प्रत्येक क्लॉकच्या पल्सला शून्य, (0000) वर मोजले जाते. म्हणून, या प्रकारच्या काउंटरला 4-बिट सिंक्रोनस डाउन काउंटर असेही म्हणतात.



आकृती 4.43 4-बिट सिंक्रोनस डाउन काउंटर

#### 4.6.23 4 बिट सिंक्रोनस अप-डाउन काउंटर (4-bit Synchronous UP-Down counter)

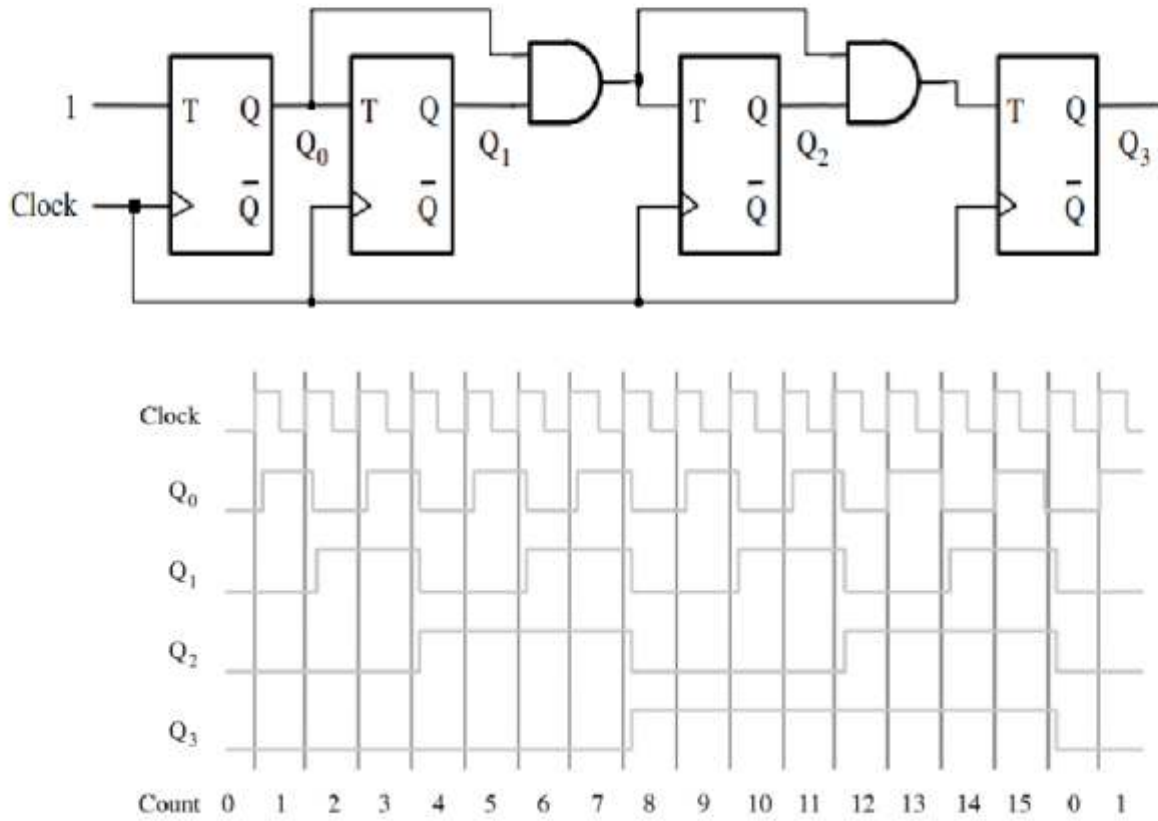
हे काउंटर "अप" आणि "डाउन" काउंट मोड निवडण्याकरीता, अँड गेट्स वापरले आहेत. "UP/DOWN" बार इंपुटला योग्य तो बिट दिला जातो. अँड गेट्सचे आउटपुट ऑर गेट्सला जोडले जातात व पुढे फ्लिप-फ्लॉपच्या J आणि K इनपुटला दिले जातात.



आकृती 4.44 4 बिट सिंक्रोनस अप-डाउन काउंटर

जर अप-डाउन नियंत्रण रेषा "हाय" असेल, तर वरचे अँड गेट्स सक्षम होतात, आणि सर्किट सिंक्रोनस "अप" काउंटर म्हणून कार्य करते, जर अप-डाउन नियंत्रण रेषा "लो" केली असेल, तर खालचे अँड गेट्स सक्षम होतील, आणि सर्किट "डाउन" काउंटर म्हणून कार्य करते.

**4.6.24 T फ्लिप-फ्लॉप वापरून 4 बिट सिंक्रोनस काउंटर (4-bit Synchronous counter using TFF)**



आकृती 4.45 T फ्लिप-फ्लॉप वापरून 4 बिट सिंक्रोनस काउंटर

**4.7 फ्लिप-फ्लॉपचे एक्सायटेशन टेबल (Excitation Table of Flip-flop)**



S-R Flip Flop

Q (t)	Q (t+1)	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

J-K Flip Flop

Q (t)	Q (t+1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

T Flip Flop

Q (t)	Q (t+1)	T
0	0	0
0	1	1
1	0	1
1	1	0

D Flip Flop

Q (t)	Q (t+1)	D
0	0	0
0	1	1
1	0	0
1	1	1

#### 4.7.1 सिंक्रोनस मॉड-6 काउंटरची रचना

मॉड 6 काउंटर 000 ते 101 पर्यंत मोजण्यासाठी डिझाइन केले जाऊ शकते.

तक्ता क्र. 4.11

$C_n$	$B_n$	$A_n$	$C_{n+1}$	$B_{n+1}$	$A_{n+1}$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	0	0	1	X	1	0	X	X	1

काउंटरच्या वर्तमान स्थितीचे कार्य म्हणून प्रत्येक फ्लिप-फ्लॉप इनपुटसाठी तर्क समीकरण मिळवणे आहे. यासाठी K-मॅप वापरले जाते.

न वापरलेल्या स्थिती 110 आणि 111 शी संबंधित मूल्ये 'x' डोन्ट केअर म्हणून दिसतात.  $J_A = K_A = 1$  हे वरील तक्त्यावरून दिसते. (तसे  $J_A$  आणि  $K_A$  साठी K-मॅप दाखवलेले नाही).

$B_n A_n$		$C_n$			
		00	01	11	10
$C_n$	0	0	0	1	0
	1	×	×	×	×

$J_C = B_n A_n$

$B_n A_n$		$C_n$			
		00	01	11	10
$C_n$	0	×	×	×	×
	1	0	1	×	×

$K_C = A_n$

$B_n A_n$		$C_n$			
		00	01	11	10
$C_n$	0	0	1	×	×
	1	0	0	×	×

$J_B = \overline{C_n} A_n$

$B_n A_n$		$C_n$			
		00	01	11	10
$C_n$	0	×	×	1	0
	1	×	×	×	×

$K_B = A_n$

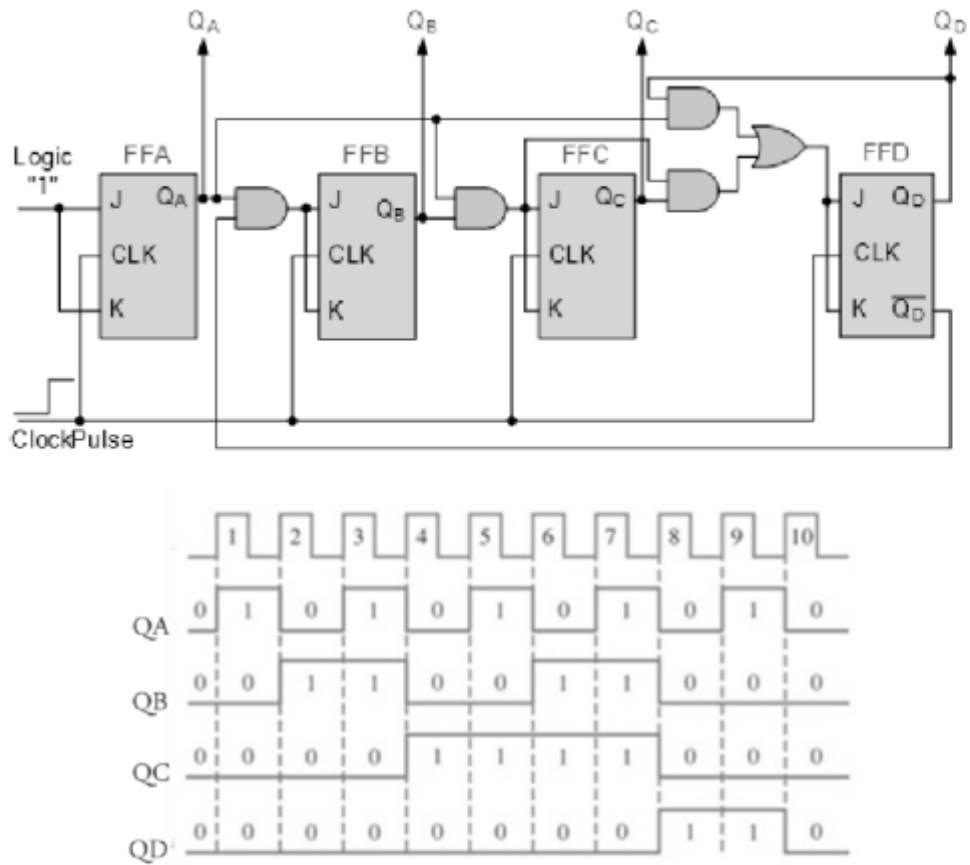
आकृती 4.46

या डिझाइन समीकरणांमधून सर्किट आकृती काढणे ही अंतिम पायरी आहे, जी आकृती 4.47 मध्ये दर्शविली आहे.



### 4.7.3 सिंक्रोनस दशक काउंटर (Synchronous decade counter)

0 ते 9 पर्यंत गणना करण्यासाठी सिंक्रोनस बायनरी काउंटर वापरून 4-बिट दशक काउंटर देखील तयार केले जाऊ शकते. बायनरी काउंटर काही अतिरिक्त तर्कशास्त्राच्या मदतीने दशकात (दशांश) काउंटरमध्ये रूपांतरित केले जाऊ शकते. इच्छित क्रम स्टेट "1001" च्या संख्येपर्यंत पोहोचल्यानंतर, काउंटर पुन्हा "0000" वर पुनर्प्रक्रिया करतो.



आकृती 4.49

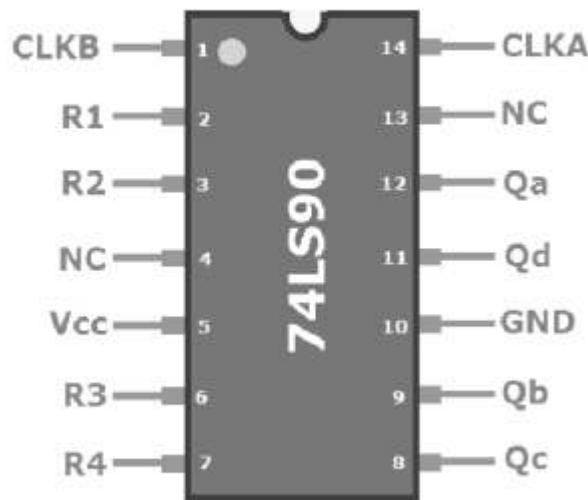
#### सिंक्रोनस काउंटरची ॲप्लिकेशन्स

- वेगवान असल्याने, सिंक्रोनस काउंटर जेथे स्विचिंग गतीची जास्त आवश्यकता असते तिथे वापरले जातात.
- अलार्म क्लॉक व डिजिटल घड्याळे.

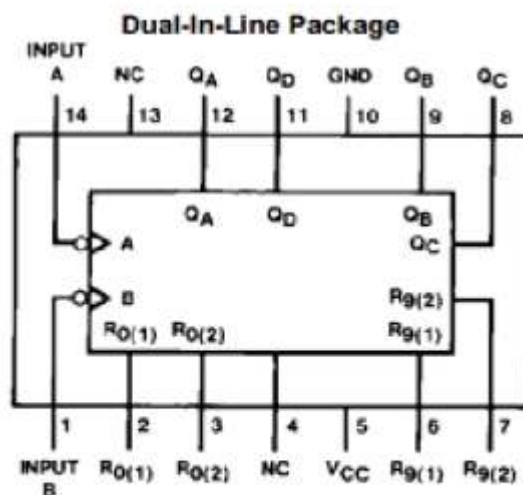
- छायाचित्र घेण्यासाठी कॅमेरामध्ये वेळ सेट करणे.
- ऑटोमोबाईलमधील फ्लॅशिंग लाइट इंडिकेटर.
- डिजिटल इलेक्ट्रॉनिक सर्किट्समध्ये, वारंवारता काउंटर/विभाजक, एडीसी इत्यादींमध्ये वापरले जातात.

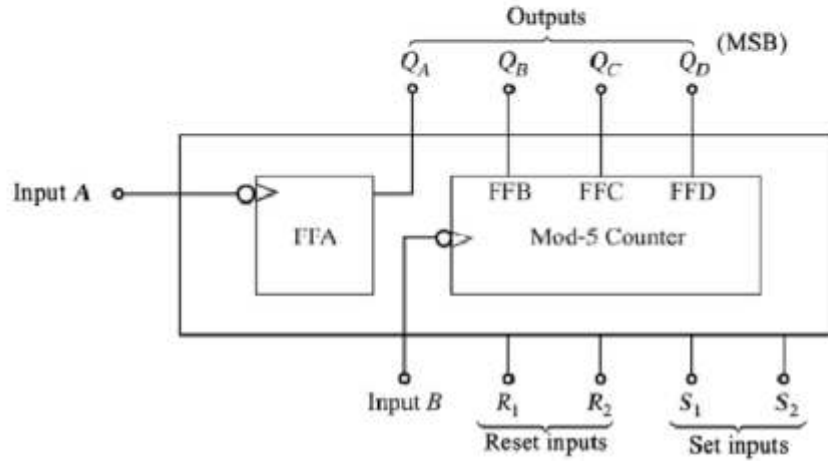
#### 4.8 IC 7490

##### पिन आकृती



IC 7490 Pin Diagram





आकृती 4.50

IC 7490 चे पिन फंक्शन

तक्ता क्र. 4.12

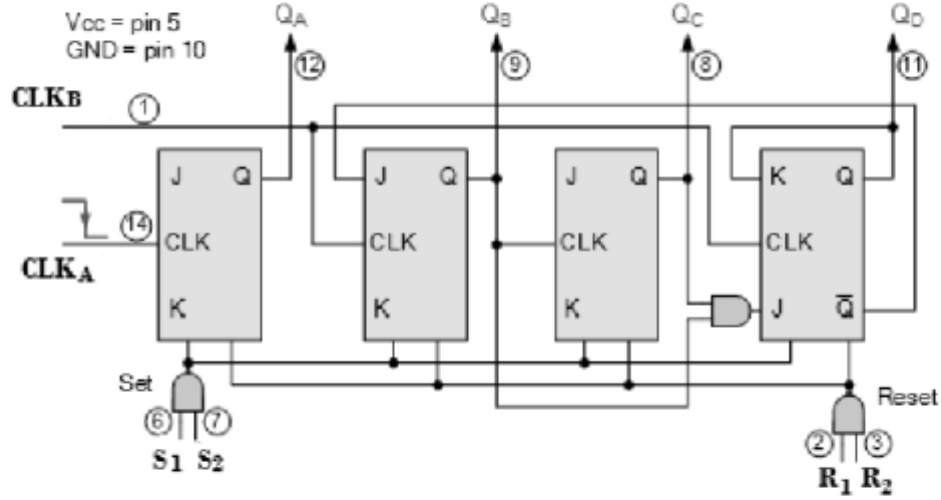
Pin No	Function	Name
1	Clock input 2	Input2
2	Reset1	R1
3	Reset2	R2
4	Not connected	NC
5	Supply voltage; 5V (4.75V – 5.25V)	Vcc
6	Reset3	R3
7	Reset4	R4
8	Output 3, BCD Output bit 2	Q <sub>C</sub>
9	Output 2, BCD Output bit 1	Q <sub>B</sub>
10	Ground (0V)	Ground
11	Output 4, BCD Output bit 3	Q <sub>D</sub>
12	Output 1, BCD Output bit 0	Q <sub>A</sub>
13	Not connected	NC
14	Clock input 1	Input1

#### 4.8.1 IC 74LS90 चे अंतर्गत तर्क आकृती (Logical diagram of IC7490)

अंतर्गत 74LS90 मध्ये चार मास्टर-स्लेव्ह J-K फ्लिप-फ्लॉप असतात. मॉड-2 (काउंट-टू-2) काउंटर आणि मॉड-5 (काउंट-टू-5) काउंटर म्हणून प्रदान करण्यासाठी कनेक्ट केले जाते. 74LS90 मध्ये CLK



A इनपुटद्वारे एक स्वतंत्र टॉगल J-K फ्लिप-फ्लॉप आहे आणि CLK B इनपुटद्वारे, तीन टॉगल J-K फ्लिप-फ्लॉप आहेत, जे असिंक्रोनस काउंटर बनवतात.



आकृती 4.51 IC 74LS90 चे अंतर्गत तर्क आकृती

QA, QB, QC आणि QD हे आउटपुट आहेत. 74LS90 मोजणी, क्लॉक सिग्नलच्या फॉलिंग एज वर ट्रिगर केला जातो, म्हणजे जेव्हा क्लॉक सिग्नल लॉजिक 1 (हाय) वरून लॉजिक 0 (लो) वर जातो.

अतिरिक्त इनपुट पिन R1 आणि R2 काउंटर "रीसेट" पिन आहेत तर इनपुट S1 आणि S2 "सेट" पिन आहेत. लॉजिक 1 शी कनेक्ट केल्यावर, रीसेट इनपुट R1 आणि R2 काउंटरला शून्य, 0 (0000) वर रीसेट करतात आणि जेव्हा सेट इनपुट S1 आणि S2 लॉजिक 1 शी कनेक्ट केलेले असतात, तेव्हा ते काउंटर कमाल 9 किंवा (1001) वर सेट करतात.

#### 4.8.2 IC 7490 चे ॲप्लिकेशन्स

1. IC 7490 चा वापर 0 - 9 बायनरी मोजणीसाठी केला जाऊ शकतो.
2. हे दशक (डिकेड/decade) काउंटर म्हणून वापरले जाऊ शकते.
3. IC 7490 स्वयंचलित कंट्रोल सर्किट्समध्ये वापरला जातो.
4. IC 74LS90 सर्व्हर, नेटवर्किंग आणि डिजिटल प्रणालींमध्ये देखील वापरतात.

5. डिजिटल क्लॉक व मॉड-6 आणि मॉड-10 काउंटरसाठी वापरले जातात.

**मॉड्यूलो -n काउंटर/मॉड-n काउंटर** - n बिट काउंटरसाठी, स्टेट्स संख्या  $2^n$  आहे.

उदाहरणार्थ 2 बिट रिपल काउंटरला  $2^2 = 4$  अवस्था म्हणतात, म्हणजे मॉड-4 काउंटर.

#### 4.8.3 कॅस्केडिंग काउंटर (Cascading counters)

काउंटर सर्किट्स, क्रम आणि वारंवारताचा विभाग करण्यासाठी कॅस्केड केले जाऊ शकतात.

प्रत्येक कॅस्केड स्टेजसह, काउंटरचे मॉड्यूलस वाढते. काउंटरचे अंतिम मॉड्यूलस एकत्रितपणे गुणाकार केलेल्या वैयक्तिक टप्प्यांच्या मॉड्यूलसच्या समान आहे.

अशा प्रकारे, 4-बिट असिंक्रोनस काउंटरमध्ये  $2 \times 2 \times 2 \times 2 = 16$  चे मॉड्यूलस असते. अंतिम टप्प्यातील आउटपुट वारंवारता, मॉड्यूलसने विभाजित केलेल्या इनपुट वारंवारते इतकी असते.

74LS90 IC काउंटरचे दशक काउंटर मिळविण्यासाठी कॅस्केडिंग आवश्यक आहे. मॉड-5 काउंटरसह मॉड-2 काउंटर कॅस्केड करून दशक काउंटर तयार होतो. मॉड-2 x मॉड-5 = मॉड-10.

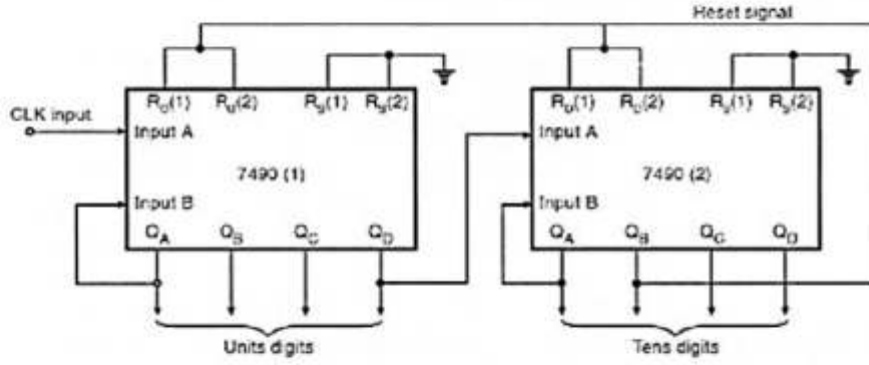
#### 4.8.4 IC 7490 वापरून दशक काउंटर (Decade counter using IC 7490)

हा एक काउंटर आहे जो 10 बायनरी अवस्था मोजतो. जेव्हा आउटपुट संख्या 10 च्या (दशांश) मूल्यापर्यंत पोहोचते, म्हणजे जेव्हा QDQCQBQA = 1010, तेव्हा ही स्थिती रीसेट इनपुटवर लागू करून काउंटर 0 वर रीसेट केले जाते.

क्लॉक-2 (पिन क्रमांक 1), QA शी (पिन क्रमांक 12) जोडलेले आहे आणि सर्व रीसेट पिन ग्राउंडशी जोडलेले आहेत. क्लॉक पल्स इनपुट पिन क्रमांक 14वर लागू केल्या जातात. व ते 0000 ते 1001 पर्यंत मोजले जाईल आणि नंतर रीसेट होईल. कारण नवव्या गणनेनंतर त्याचे सर्व फ्लिप-फ्लॉप क्लिअर होतात. अशा प्रकारे IC 7490 वापरून डिकेड काउंटर केला जाऊ शकतो.



पहिल्या IC 7490(1) चे MSB आउटपुट दुसऱ्या IC 7490(2) च्या मॉड 2 च्या क्लॉक इनपुटला कॅस्केड करून आकृती 4.54 मध्ये दाखवल्याप्रमाणे मॉड 20 काउंटर बनवू शकतो. काउंटर 0-19 स्टेड्समधून पुढे जाईल आणि 20 व्या स्थिती मध्ये रीसेट होईल.



Logic diagram of MOD-20 counter using IC 7490

### आकृती 4.54 IC 7490 वापरून मोड-20 काउंटर

#### 4.9 रिंग काउंटर (Ring counter)

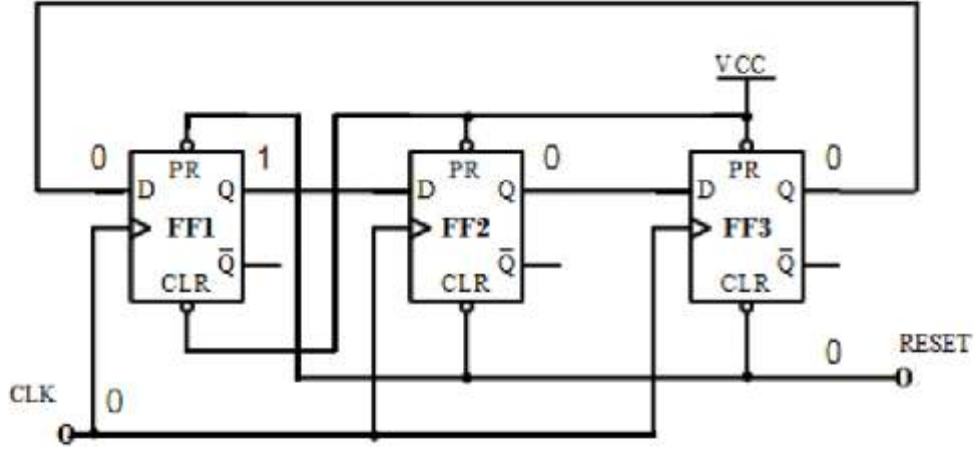
रिंग काउंटर हे शिफ्ट रजिस्टर (फ्लिप-फ्लॉपचे) कॅस्केड कनेक्शन आहे ज्यामध्ये शेवटच्या फ्लिप-फ्लॉपचे आउटपुट पहिल्याच्या इनपुटशी जोडलेले आहे. हे अशा प्रकारे सुरु केले जाते की फक्त एका फ्लिप-फ्लॉपचे आउटपुट 1 असते तर उर्वरित 0 असते. 1 बिट प्रत्येक क्लॉक पल्सला पुढे पुढे जात असतो. त्यामुळे  $n$  फ्लिप-फ्लॉप वापरल्यास स्थिती प्रत्येक  $n$  क्लॉक सायकलची पुनरावृत्ती होते.

रिंग काउंटर हा एक प्रकारचा काउंटर आहे जो शिफ्ट रजिस्टरमध्ये फ्लिप-फ्लॉप्सने बनलेला असतो, ज्यामध्ये शेवटच्या फ्लिप-फ्लॉपचे आउटपुट पहिल्याच्या इनपुटला दिले जाते, ज्यामुळे "गोलाकार" किंवा "रिंग" रचना बनते.

रिंग काउंटरचे ॲप्लिकेशन्स -

1. सतत लूपमध्ये डेटा मोजण्यासाठी रिंग काउंटर वापरले जातात.
2. वारंवारता विभाजक सर्किट्समध्ये 2 स्टेज, 3 स्टेज आणि 4 स्टेज रिंग काउंटर अनुक्रमे 2 ने विभागाने, 3 ने विभागाने आणि 4 ने विभागाने म्हणून वापरले जातात

#### 4.9.1 3 बिट रिंग काउंटर (3-bit ring counter)



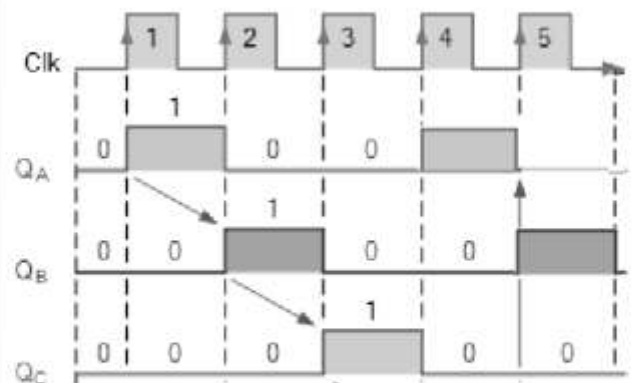
आकृती 4.55 3 बिट रिंग काउंटर

सुरुवातीला FF2 आणि FF3 रीसेट इनपुटने क्लिअर केले जाते. परंतु रीसेट लाइन FF1 च्या PR इनपुटशी कनेक्ट केल्यामुळे FF1 सेट केला जाईल. सर्किटची प्रारंभिक स्थिती नियुक्त केल्यावर, रीसेट लाइन उच्च केली जाते आणि क्लॉक पल्स लागू केल्या जातात.

प्रत्येक क्लॉक पल्सला, डेटा एक बिट उजवीकडे शिफ्ट होईल आणि 3 क्लॉक पल्सनंतर क्रम पुनरावृत्ती होईल. अशा प्रकारे, "1" प्रत्येक क्लॉक पल्सला सर्किटमध्ये फिरत राहतो. 3 बिट रिंग काउंटरचे सत्य सारणी आणि टाइम आकृती तक्ता 4.13 मध्ये दाखवली आहे.

तक्ता 4.13 3 बिट रिंग काउंटर सत्य सारणी आणि टाइमिंग डायग्राम

Clock Cycle	QA	QB	QC
1	1	0	0
2	0	1	0
3	0	0	1
4	1	0	0
5	0	1	0





तक्ता वरून पाहिले जाऊ शकते की कोणत्याही वेळी, फक्त एकच आउटपुट हाय असते आणि ती हाय स्थिती प्रत्येक क्लॉक पल्ससह जवळच्या फ्लिप-फ्लॉपकडे सरकते.

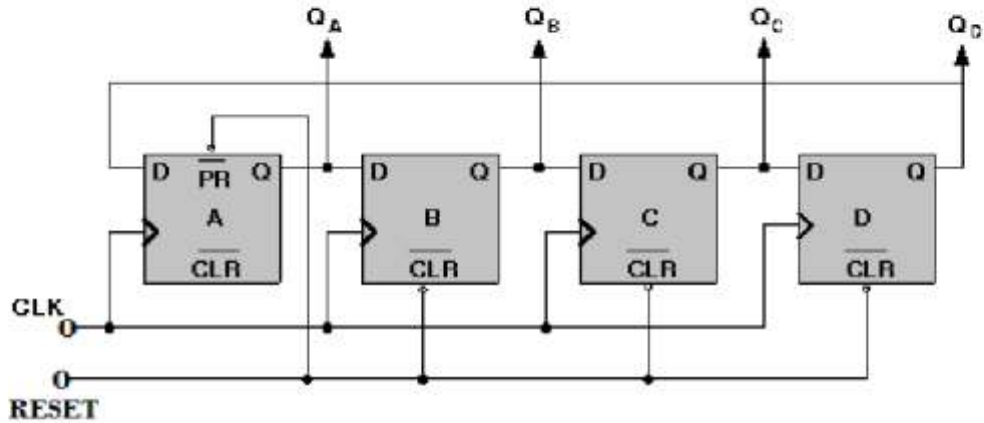
#### 4.9.2 4-बिट रिंग काउंटर (4-bit ring counter)

सिंक्रोनस रिंग काउंटरचे प्रिसेट ने एक डेटा बिट लॉजिक "1" वर सेट केला जातो व इतर सर्व बिट "0" वर रीसेट केले जातात.

फ्लिप-फ्लॉप्सचा आउटपुट लॉजिक "0" स्तरावर करण्यासाठी "क्लीअर" सिग्नल देऊन "रीसेट" केला जातो आणि पहिल्या फ्लिप-फ्लॉपच्या इनपुटवर "प्रिसेट" पल्स लागू केला जातो.

या मुळे क्लॉक पल्स लागू करण्यापूर्वी फ्लिप-फ्लॉप (FFA) लॉजिक "1" राहते.

काउंटरभोवती डेटा योग्यरित्या फिरण्याकरीता, प्रथम काउंटरला योग्य डेटा पॅटर्न "लोड" करणे आवश्यक असते. कारण प्रत्येक क्लॉकवर सर्व लॉजिक "0's" किंवा सर्व लॉजिक "1's" आउटपुट केल्यास रिंग काउंटर अवैध (invalid) होईल.



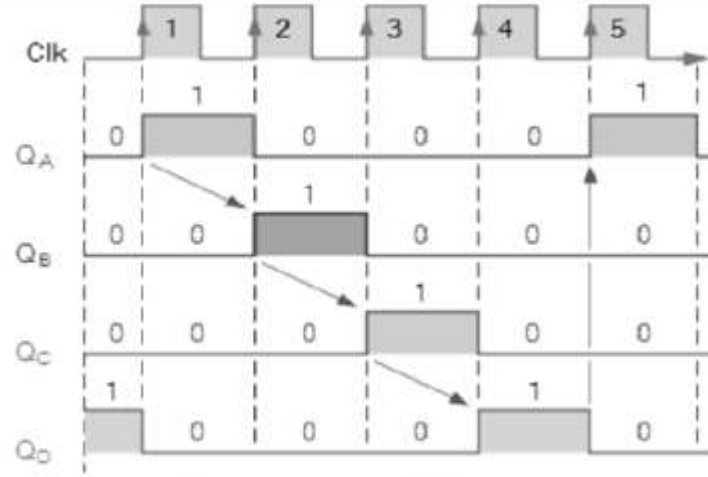
आकृती 4.56 4-बिट रिंग काउंटर

या प्रकारच्या डेटा हालचालीला "रोटेशन" असे म्हणतात. रिंग काउंटरद्वारे डावीकडून उजवीकडे डेटाच्या हालचालीचा परिणाम त्याच्या टाइम लाइन वर आकृती 4.57 प्रमाणे ग्राफिकरित्या दाखवला आहे.



तक्ता 4.14 4-बिट रिंग काउंटर सत्य सारणी

Clock Cycle	QA	QB	QC	QD
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	1	0	0	0
6	0	1	0	0



आकृती 4.57 टाईमिंग डायग्राम

वर दर्शविलेल्या रिंग काउंटरच्या उदाहरणामध्ये चार भिन्न अवस्था असल्यामुळे, प्रत्येक फ्लिप-फ्लॉप आउटपुटमध्ये वारंवारता, क्लॉकच्या एक चतुर्थांश ( $1/4$ ) आहे.

काउंटरचा "मॉड्युलो" किंवा "मॉड्युलस" म्हणजे काउंटरची स्टेट्सची संख्या आहे आणि कोणत्याही मॉड्युलो नंबर करिता रिंग काउंटर बनवता येतो. "मॉड-n" रिंग काउंटरला "n" भिन्न आउटपुट स्थिती मिळवण्यासाठी, n-फ्लिप-फ्लॉप एकत्र जोडलेले आवश्यक आहे.

उदाहरणार्थ, मॉड-8 रिंग काउंटरला आठ फ्लिप-फ्लॉपची आवश्यकता असते आणि मॉड-16 रिंग काउंटरला सोळा फ्लिप-फ्लॉपची आवश्यकता असते. तथापि, वरील उदाहरणाप्रमाणे, संभाव्य सोळा स्टेट्स पैकी फक्त चार स्टेट्स वापरली जातात, ज्यामुळे रिंग काउंटर त्यांच्या आउटपुट स्थितीच्या वापराच्या दृष्टीने अत्यंत अकार्यक्षम बनतात.

#### 4.10 ट्विस्टेड रिंग काउंटर (Twisted ring counter)

ट्विस्टेड रिंग काउंटर हे शिफ्ट रजिस्टर आहे ज्यामध्ये शेवटच्या फ्लिप-फ्लॉपचे आउटपुट पहिल्याच्या इनपुटशी जोडलेले आहे. प्रत्येक फ्लिप-फ्लॉप क्लॉक पल्समध्ये सेट आणि रीसेट होतो आणि  $n$  फ्लिप-फ्लॉप वापरल्यास हा क्रम प्रत्येक क्लॉकच्या  $2^n$  वेळा पुनरावृत्त होतो.

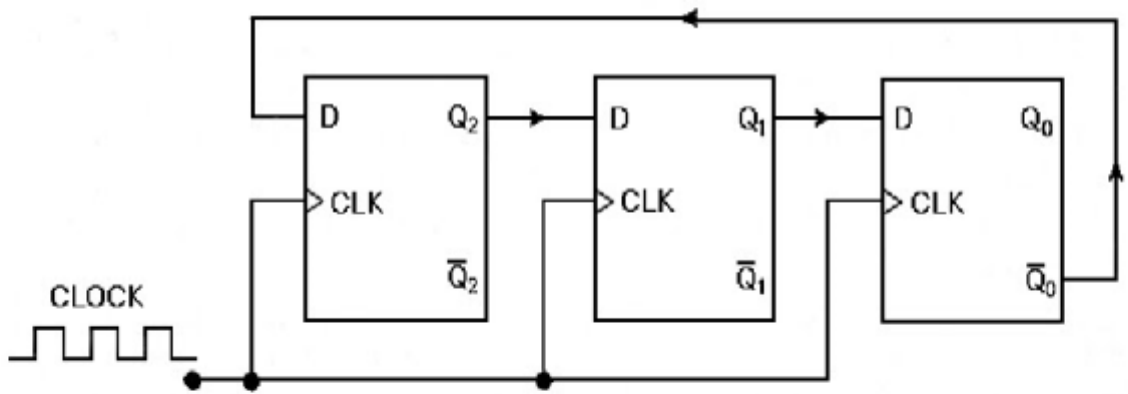
##### 4.10.1 जॉन्सन्स काउंटर (Johnson's counter)

रिंग काउंटरप्रमाणे जॉन्सन काउंटर हे शिफ्ट रजिस्टर आहे. जर रिंग काउंटरचे पूरक आउटपुट इनपुटला दिले गेले तर, जॉन्सन काउंटर तयार होईल.

रिंग काउंटर आणि जॉन्सन काउंटरमधील फरक म्हणजे शेवटच्या टप्प्याचे कोणते आउटपुट परत दिले जाते ( $Q$  की  $Q'$ ).

##### 4.10.2 3 बिट ट्विस्टेड रिंग काउंटर (3-bit twisted ring counter)

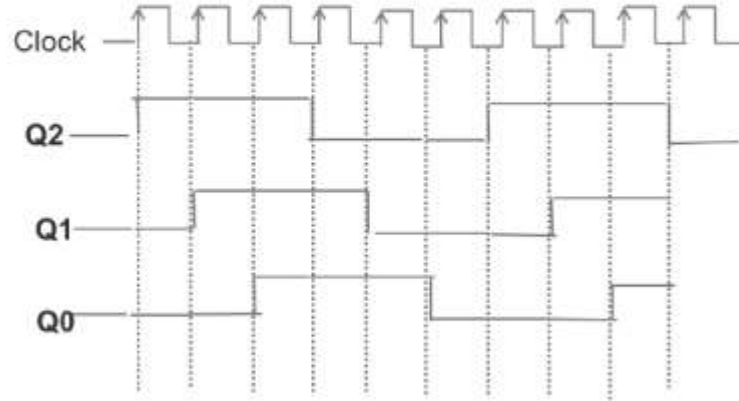
सुरुवातीला सर्व D फ्लिप-फ्लॉप क्लिअर केले आहेत. तर,  $Q_2Q_1Q_0=000$ . पहिल्या D फ्लिप-फ्लॉपच्या इनपुटला तिसऱ्या फ्लिप-फ्लॉपचे पूरक आउटपुट मिळते. तर, पहिल्या D फ्लिप-फ्लॉपचे आउटपुट दुसऱ्या फ्लिप-फ्लॉपच्या मागील पूरक आउटपुटच्या बरोबरीचे आहे. पहिल्या आणि दुसऱ्या D फ्लिप-फ्लॉपचे मागील आउटपुट एका बिटने उजवीकडे शिफ्ट जातात.



आकृती 4.58 3 बिट ट्विस्टेड रिंग काउंटर

Clock Cycle	Q2	Q1	Q0
1	1	0	0
2	1	1	0
3	1	1	1
4	0	1	1
5	0	0	1
6	0	0	0
7	1	0	0

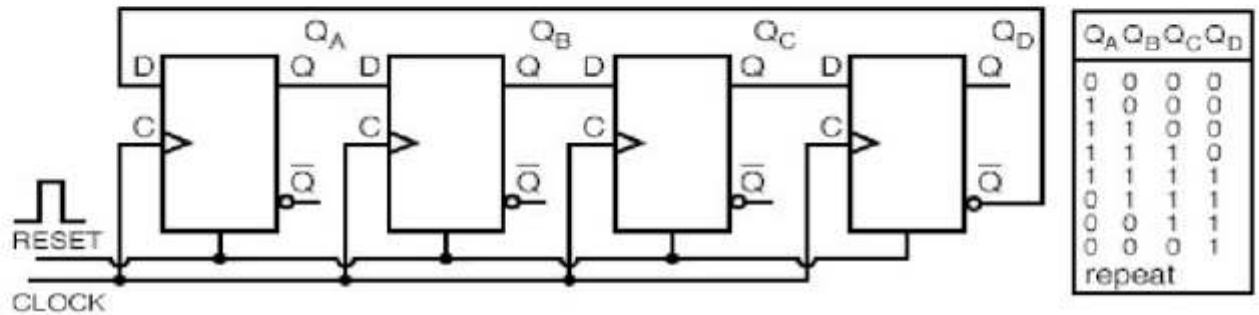
तक्ता 4.14 3 बिट ट्विस्टेड रिंग काउंटरचे सत्य सारणी



आकृती 4.59 टाईमिंग डायग्राम

जेव्हा क्लॉक रायजिंग एज येते, तेव्हा आउटपुट Q1 हे Q0 चे मूल्य असते. पहिल्या क्लॉक पल्सवर, काउंटरचे आउटपुट 100 असते. दुसऱ्या क्लॉक पल्सवर, काउंटरचे आउटपुट 110 असते. तिसऱ्या क्लॉक पल्सवर, काउंटरचे आउटपुट 111 असते. चौथ्या क्लॉक पल्सवर, काउंटरचे आउटपुट 011 आहे. पाचव्या क्लॉक पल्सवर, काउंटरचे आउटपुट 001 आहे. सहाव्या क्लॉक पल्सवर काउंटरचे आउटपुट 000 आहे.

#### 4.10.3 4 बिट ट्विस्टेड रिंग काउंटर (4-bit Twisted ring counter)

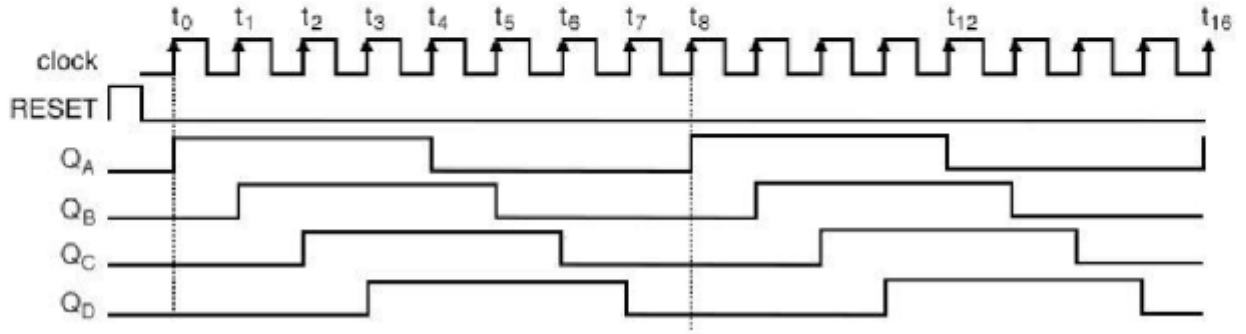


आकृती 4.60 4 बिट ट्विस्टेड रिंग काउंटर

सुरुवातीला सर्व आउटपुट क्लिअर केले जातील. आकृती 4.60 प्रमाणे, पहिल्या क्लॉकला QA QB QC वरून उजवीकडे QB QC QD मध्ये तीन 0s शिफ्ट होतात. Q<sub>D</sub> चे पूरक '1' QA मध्ये शिफ्ट होते. अशा प्रकारे, 0s च्या जागी 1s उजवीकडे शिफ्ट होतात.

4 बिट ट्विस्टेड रिंग किंवा जॉन्सनच्या काउंटरचे सत्य सारणी

Clock No.	$Q_D$	$Q_C$	$Q_B$	$Q_A$
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	1	1	1
4	1	1	1	1
5	1	1	1	0
6	1	1	0	0
7	1	0	0	0



Four stage Johnson counter waveforms

### आकृती 4.61 4- बिट जॉन्सन काउंटर वेवफॉर्म्स

#### स्व: अध्ययन (Self learning)

1. डिजिटल सर्किट्सचे प्रकार सूचीबद्ध करा.
2. डिजिटल सर्किट्समध्ये क्लॉकचे कार्य स्पष्ट करा.
3. डिजिटल सर्किट्समध्ये ट्रिगरिंगचे प्रकार सूचीबद्ध करा.
4. फ्लिप-फ्लॉपची व्याख्या लिहा. नॉर गेट वापरलेल्या R-S फ्लिप-फ्लॉपचे कार्य स्पष्ट करा.
5. फ्लिप-फ्लॉपची व्याख्या लिहा. न्यांड गेट वापरून R-S फ्लिप-फ्लॉपचे कार्य स्पष्ट करा.
6. फ्लिप-फ्लॉपची व्याख्या लिहा. फ्लिप-फ्लॉपचे प्रकार लिहा.
7. क्लॉकड R-S फ्लिप-फ्लॉपचे आकृती काढून कार्य स्पष्ट करा.
8. R-S फ्लिप-फ्लॉपचे तोटे लिहा. D फ्लिप-फ्लॉपमध्ये कसे नाहीसे केले जातात ते स्पष्ट करा.
9. प्रिसेट (PR) आणि क्लीअर (CLR)चे स्पष्टीकरण द्या.
10. J-K फ्लिप-फ्लॉपची तार्किक आकृती काढून त्याची सत्य सारणी लिहा व स्पष्ट करा.
11. D फ्लिप-फ्लॉपची तार्किक आकृती काढून त्याची सत्य सारणी लिहा.
12. रेस- अराऊंड कंडिशन स्पष्ट करा.
13. एज टिगर्ड D फ्लिप-फ्लॉपचे कार्य स्पष्ट करा.
14. एज टिगर्ड J-K फ्लिप-फ्लॉपचे कार्य स्पष्ट करा.
15. मास्टर स्लेव्ह J-K फ्लिप-फ्लॉपचे सविस्तर वर्णन करा.

16. तार्किक आकृतीच्या साहाय्याने T फ्लिप-फ्लॉप स्पष्ट करा.
17. रजिस्टरची संकल्पना स्पष्ट करा व त्यांचे मूलभूत प्रकार सांगून त्यांच्या ठोकळाकृती काढा.
18. लेफ्ट शिफ्ट रजिस्टरची तार्किक आकृती व टायमिंग आकृती काढून कार्य लिहा.
19. राईट शिफ्ट रजिस्टरचे कार्य स्पष्ट करा.
20. तार्किक आकृतीच्या साहाय्याने 3 बीट रिपल काउंटर स्पष्ट करून टायमिंग आकृती काढा.
21. तार्किक आकृतीच्या साहाय्याने डिकेड काउंटरचे कार्य स्पष्ट करा. वेव्हफॉर्मस काढा.
22. IC 7490 वापरून डिकेड काउंटरची आकृती काढून स्पष्ट करा.
23. सिंक्रोनस आणि एसिंक्रोनस काउंटरमधील फरक लिहा.
24. कॉम्बिनेशनल आणि सिक्वेन्शियल सर्किट्सचे व्याख्या आकृतीसह स्पष्टी करा.
25. कॉम्बिनेशनल आणि सिक्वेन्शियल सर्किट्समधील फरक करा.

#### **लघु प्रकल्प करणे (Micro project suggestions)**

1. IC 7490 वापरून दशक काउंटर तयार करा
2. J-K फ्लिप-फ्लॉपचे वापरून रिपल काउंटर तयार करा
3. IC 7490 वापरून बायनरी काउंटर तयार करा



## युनिट-5

### डेटारूपांतरण आणि पीएलडीस (Data converters and PLDs)

#### विषय निष्पत्ती (Course Outcome):

डिजिटल इलेक्ट्रॉनिक्स सिस्टमचा डेटा रूपांतरण आणि पीएलडी ची (PLD) चाचणी करणे.

#### युनिट निष्पत्ती (Unit outcomes):-

- 5.a दिलेल्या निर्दिष्ट (specified) डिजिटल इनपुटसाठी R-2R लॅडर आउटपुट विद्युत दाबाची गणना करणे.
- 5.b दिलेल्या विशिष्ट डिजिटल इनपुटच्या भारित (weighted) रजिस्टर DAC च्या आउटपुट विद्युत दाबाची गणना करणे.
- 5.c विविध प्रकारच्या ADC च्या कार्याचे तत्व डायग्रामसह स्पष्ट करणे.
- 5.d विविध प्रकारच्या मेमरिंचे कार्य तत्व आकृतीसह स्पष्ट करणे.
- 5.e दिलेल्या प्रकारच्या प्रोग्रामेबल लॉजिक उपकरणांचे (PLDs) कार्यतत्व मूलभूत ब्लॉक डायग्राम सह स्पष्ट करणे.

#### 5.1.1 परिचय (Introduction):

वास्तविकता जगातील सर्व संदेश हे मुळतः अॅनालॉग स्वरूपात असतात. अॅनालॉग सिग्नल हा वेळेनुसार बदलणारा (time-varying) सिग्नल आहे, जे दिलेल्या टाईम स्लॉटसाठी अमर्यादित असतात. डिजिटल सिग्नल एका लेव्हलवरून दुसऱ्या लेव्हलवर बदलतो आणि दिलेल्या टाईम स्लॉटसाठी मर्यादित असतात.

#### डेटाकन्व्हर्टरची आवश्यकता

डेटा रूपांतरण (Conversion) एका स्वरूपामधून दुसऱ्या स्वरूपात बदलण्याची किंवा रूपांतरित करण्याची प्रक्रिया आहे. प्रक्रिया आणि संप्रेषणामध्ये फक्त दोन प्रकारचे डेटा स्वरूप अॅनालॉग आणि डिजिटल डेटा आहेत.

डिजिटल सिग्नलचे स्टोरेज आणि प्रक्रिया हे जलद, नॉईस प्रतिकारक (noise immune), कार्यक्षम आणि सोपे आहे. त्यामुळे सर्व रियल वर्ल्ड ॲप्लिकेशन डिजिटल पद्धतीने केले जातात आणि साठवले जातात. त्यामुळे अॅनालॉगला ते डिजिटल आणि डिजिटलला ते अॅनालॉगमध्ये रूपांतरित करावे लागते.

### डेटाकन्व्हर्टरचे उपयोग:-

1. ट्रान्सड्यूसर (Transducer) सह एकत्र वापरले जाते.
2. ॲनालॉग सिग्नलला डिजिटल सिग्नलमध्ये रूपांतरित करण्यासाठी संगणकात (computer) वापरले जाते.
3. सेल फोनमध्ये वापरले जाते
4. मायक्रोकंट्रोलरमध्ये वापरले जाते.
5. डिजिटल सिग्नल प्रक्रियेत वापरले जाते.
6. डिजिटल स्टोरेज ऑसिलोस्कोपमध्ये वापरले जाते.
7. वैज्ञानिक उपकरणांमध्ये वापरले जाते.

### डेटाकन्व्हर्टरचे प्रकार (Types of Data converter)

1. डिजिटल टू ॲनालॉग कन्व्हर्टर
2. ॲनालॉग ते डिजिटल कन्व्हर्टर

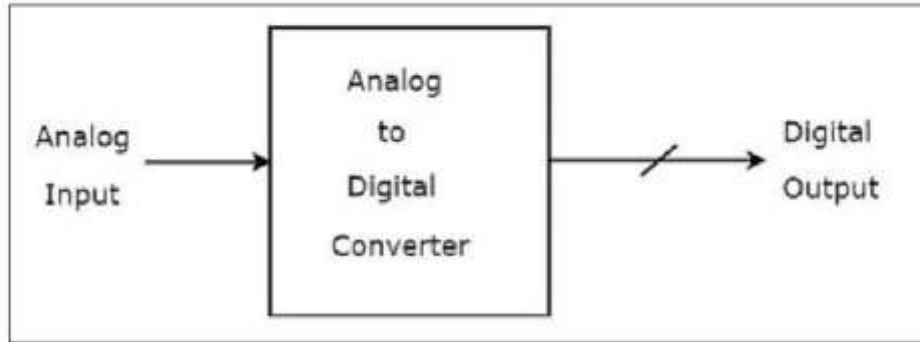
जर डिजिटल सर्किटचे आउटपुट ॲनालॉग सर्किटचे इनपुट म्हणून जोडायचे असेल, तर त्यांच्यामध्ये इंटरफेसिंग सर्किट ठेवावे लागते. डिजिटल सिग्नलला ॲनालॉग सिग्नलमध्ये रूपांतरित करणाऱ्या या इंटरफेसिंग सर्किटला डिजिटल टू ॲनालॉग कन्व्हर्टर (ADC) म्हणतात. त्याचप्रमाणे, काही ॲनालॉग टू डिजिटल कन्व्हर्टर त्यांच्या ऑपरेशनसाठी अंतर्गत ब्लॉक म्हणून डिजिटल टू ॲनालॉग कन्व्हर्टरची (DAC) आवश्यकता असू शकते.

#### 5.1.2 ॲनालॉग ते डिजिटल कन्व्हर्टर (Analog to Digital converter)

ॲनालॉग व्हॅल्यूमधून डेटा डिजिटल व्हॅल्यूमध्ये रूपांतरित करणाऱ्या डेटा कन्व्हर्टरला ॲनालॉग टू डिजिटल कन्व्हर्टर (ADC) म्हणतात. ADC ला दोन प्रकारात तयार केले जाऊ शकते. सॅम्पलिंग आणि क्वांटायझेशन. सॅम्पलिंग हे टाईम व्हेरिंग व्होल्टेज सिग्नलला डिस्क्रीट टाईम सिग्नलमध्ये रूपांतरित करते. ADC मध्ये इनपुट सिग्नल व्हॅल्यूचे विशिष्ट वेळेच्या अंतराने सॅम्पलिंग केले जाते आणि काउंटर आणि DAC च्या कॉम्बिनेशनने तयार केलेल्या ॲनालॉग व्हॅल्यूशी तुलना केली जाते. हे अतिशय महत्वाचे कन्व्हर्टर आहे. कारण डिजिटल कॉम्प्युटरवर प्रक्रिया करण्यासाठी ॲनालॉग सिग्नल्सचे डिजिटलमध्ये रूपांतर करावे लागते.

आकृती 5.2 मध्ये दर्शविलेल्या प्रमाणे, ॲनालॉग टू डिजिटल कन्व्हर्टर (ADC) मध्ये एक (single) ॲनालॉग इनपुट आणि अनेक (multiple) बायनरी (digital) आउटपुट असतात. ADC च्या बायनरी (digital) आउटपुटची संख्या दोनची पॉवर असते. (e.g.  $2^0, 2^1, 2^2, \dots, 2^n$  etc)

ADC ची ब्लॉक डायग्राम आकृती 5.2 मध्ये दर्शविले आहे-



आकृती 5.2- ADC ब्लॉक डायग्राम

अॅनालॉग टू डिजिटल कन्व्हर्टरचे प्रकार:

1. फ्लॅश ए टू डी कन्व्हर्टर (Flash A/D convertre)
2. काउंटर-टाईप ए टू डी कन्व्हर्टर (Counter type A/D converter)
3. ट्रॅकिंग-टाईप ए टू डी कन्व्हर्टर (Tracking type A/D converter)
- 4.सकसेसीव अॅप्रॉक्सिमेशन टाईप ए टू डी कन्व्हर्टर (Successive Approximation type A/D converter)
5. सिंगल स्लोप ए टू डी कन्व्हर्टर (Single Slope A/D converter)
6. ड्युअल-स्लोप ए टू डी कन्व्हर्टर (Dual slope A/D converter)

फ्लॅश ADC हे वर नमूद केलेल्या सर्व प्रकारांपैकी सर्वात वेगवान ADC आहे.

डिजिटल टू अॅनालॉग कन्व्हर्टर (DAC) डिजिटल इनपुट सिग्नलला अॅनालॉग आउटपुट सिग्नलमध्ये रूपांतरित (convert) करते. स्विच नियंत्रित करण्यासाठी 1(ON) आणि 0 (Off) च्या स्वरूपातील डिजिटल डेटा वापरला जातो, या स्विचच्या स्थितीवर (ON/OFF) आणि स्थिती (MSB 'or' LSB) आउटपुट अॅनालॉग विपुलता नुसार मोजला जातो. डिजिटल-टू-अॅनालॉग कन्व्हर्टर (DAC) हा वृद्धी (gain) आणि ऑफसेट त्रुटिवर अवलंबून आहेत.

DAC ब्लॉक डायग्राम आकृती 5.1 मध्ये दर्शविली आहे.



आकृती 5.1-DAC ब्लॉक डायग्राम

डिजिटल टू ॲनालॉग कन्व्हर्टर (DAC) मध्ये अनेक (more than one) बायनरी इनपुट आणि एक (single) आउटपुट असते. सर्वसाधारणपणे, DACच्या बायनरी इनपुटची संख्या दोनची पॉवर (e.g.  $2^0, 2^1, 2^2, \dots, 2^n$  etc) असेल.

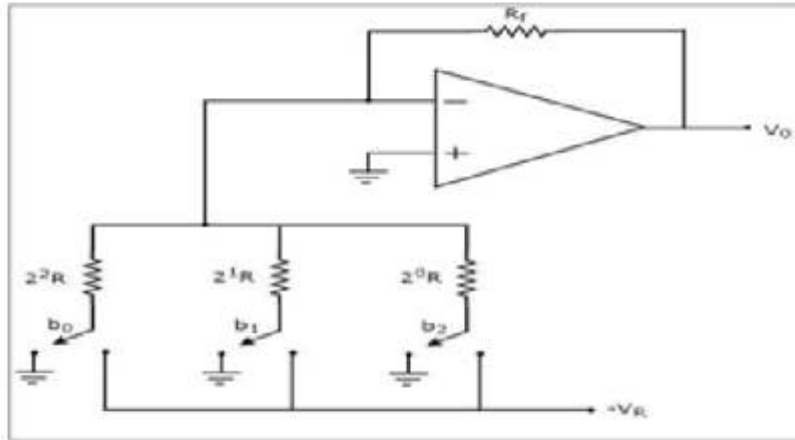
### 5.1.3 DAC चे प्रकार

1. बायनरी वेटेड रेसिस्टर DAC (Binary Weighted Resistor)
- 2 R-2R लॅडर DAC (R-2R ladder)

#### 5.1.3.1 बायनरी वेटेड रेजिस्टर (Binary Weighted Resistor) DAC

वेटेड रेजिस्टर DAC एक ॲनालॉग आउटपुट तयार करते, जे इनव्हर्टिंग ॲडर सर्किटमध्ये बायनरी वेटेड रेजिस्टर वापरून डिजिटल (बायनरी) इनपुटच्या जवळपास समान असते. म्हणून बायनरी वेटेड रेसिस्टर DAC ला वेटेड रेसिस्टर DAC म्हणतात. बायनरी वेटेड रेजिस्टरमध्ये, प्रत्येक डिजिटल स्तर (level) समतुल्य (equivalent) ॲनालॉग विद्युतदाब (voltage) किंवा विद्युतप्रवाह (current) मध्ये रूपांतरित केला जातो.

3-बिट बायनरी वेटेड रेसिस्टर DAC चे सर्किट डायग्राम आकृती 5.3 मध्ये दाखवले आहे –



आकृती 5.3 3-बिट बायनरी वेटेड रेसिस्टर DAC सर्किट

बायनरी वेटेड रेसिस्टरचे सर्किट, आकृती 5.3 मध्ये दर्शविल्याप्रमाणे Op-amp (Operational amplifier) वापरून रेसिस्टरचे नेटवर्क आणि समिंग ॲम्प्लिफायर वापरते. रेसिस्टर  $2^1R=2R$ ,  $2^2=4R$ ,  $\dots, 2^nR$  हे बायनरी वेटेड रेसिस्टरचे नेटवर्क बनवतात. प्रति (per) बिट एक वापरल्या जाणाऱ्या डिजिटल नियंत्रित स्विचची संख्या  $n$  असते.

जेव्हा आकृती 5.3 मध्ये दर्शविलेले डिजिटल स्विच ग्राउंडला जोडले जातील त्यावेळी संबंधित इनपुट बिट '0' असेल, त्याचप्रमाणे जेव्हा संबंधित इनपुट बिट '1' असेल, आकृती 5.3 मध्ये दर्शविलेले डिजिटल स्विच निगेटिव्ह रेफरन्स व्होल्टेज,  $-V_R$  शी जोडले जातील.



सर्किट 5.3 मध्ये, op-amp (operational amplifier) चे नॉन-इनव्हर्टिंग इनपुट टर्मिनल ग्राउंडला जोडलेले आहे. म्हणजे op-amp च्या नॉन-इनव्हर्टिंग इनपुट टर्मिनलवर शून्य (0V) असेल.

व्हर्च्युअल ग्राउंड (virtual ground) संकल्पनेनुसार, op-amp च्या इनव्हर्टिंग इनपुट टर्मिनल वरील व्होल्टेज त्याच्या नॉन-इनव्हर्टिंग इनपुट टर्मिनलवर असलेल्या व्होल्टेज सारखेच असते. जर, इनव्हर्टिंग इनपुट टर्मिनलच्या नोड (node) वरील व्होल्टेज शून्य (0V) असेल तर इनव्हर्टिंग इनपुट टर्मिनलच्या नोडवर, नोडल समीकरण (equation) असे असेल:

$$\frac{0+V_R b_2}{2^0 R} + \frac{0+V_R b_1}{2^1 R} + \frac{0+V_R b_0}{2^2 R} + \frac{0-V_o}{R_f} = 0 \dots (1)$$

$$\frac{V_o}{R_f} = \frac{V_R b_2}{2^0 R} + \frac{V_R b_1}{2^1 R} + \frac{V_R b_0}{2^2 R}$$

$$V_o = \frac{V_R R_f}{R} \left[ \frac{b_2}{2^0} + \frac{b_1}{2^1} + \frac{b_0}{2^2} \right] \dots (2)$$

Substitute  $R_f = R$

$$V_o = \frac{V_R R}{R} \left[ \frac{b_2}{2^0} + \frac{b_1}{2^1} + \frac{b_0}{2^2} \right]$$

$$V_o = V_R \left[ \frac{b_2}{2^0} + \frac{b_1}{2^1} + \frac{b_0}{2^2} \right] \dots (3)$$

समीकरण-3 (equation) 3-बिट बायनरी वेटेड रेसिस्टर DAC चे आउटपुट व्होल्टेज समीकरण दर्शवते. बायनरी (digital) इनपुटमध्ये बिट्सची संख्या तीन असल्याने, फिक्स्ड रेफरन्स व्होल्टेज,  $V_R$  साठी बायनरी इनपुट 000 ते 111 पर्यंत बदलून आउटपुट व्होल्टेजची सात संभाव्य (possible) व्हॅल्यूज मिळतील.

3-बिट बायनरी वेटेड रेसिस्टर DAC च्या आउटपुट व्होल्टेज समीकरणाच्या आधारे आपण N-bit बायनरी वेटेड रेसिस्टर DAC चे जनरलाइज्ड (generalized) आउटपुट व्होल्टेज समीकरण (equation) असे लिहू शकतो.

$$V_o = V_R \left[ \frac{b_{N-1}}{2^0} + \frac{b_{N-2}}{2^1} \dots + \frac{b_0}{2^{N-1}} \right] \dots (4)$$

$b_1, b_2, b_3, \dots, b_N$  ची व्हॅल्यू बदलून संबंधित अॅनालॉग आउटपुट व्होल्टेज  $V_o$  ची व्हॅल्यू मिळवू शकतो. निगेटिव्ह रेफरन्स व्होल्टेजचा वापर केल्याने पॉझिटिव्ह आउटपुट व्होल्टेज मिळते.

**फायदे:**

1. अंमलबजावणी सोपी असते (Simple implementation)
2. गणना सोपी असते. (Easy computations)

तोटे:

1. यासाठी रेसिस्टर व्हॅल्यूची विस्तृत रेंज (wide range) आवश्यक आहे.
2. डिजिटल इनपुटमध्ये प्रेझेंट (present) असलेल्या बिट्सची संख्या वाढते त्यामुळे अधिक अचूक रेसिस्टर रचना (design) करणे कठीण असते.
3. डिजिटल इनपुटमध्ये प्रेझेंट (present) असलेल्या बिट्सची संख्या वाढल्यामुळे LSB आणि MSB शी संबंधित रेसिस्टर व्हॅल्यू मधील फरक वाढेल.
4. MSB पोजिशनसाठी खूप जास्त वॅटेजचे रेसिस्टर आवश्यक असतात.
5. प्रत्येक बिट पोजिशनसाठी एक अचूक रेसिस्टर आवश्यक आहे.

**उदाहरणे:-**

1. Calculate analog output of 4 bit DAC for digital input 1101.

Assume  $V_{FS} = 10V$ .

उत्तर :

$$V_R = V_{FS}$$

$$V_o = V_R \left[ \frac{b_{N-1}}{2^0} + \frac{b_{N-2}}{2^1} + \dots + \frac{b_0}{2^{N-1}} \right]$$

$$\begin{aligned} &= 10(1 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4}) \\ &= 10(0.5 + 0.25 + 0 + 0.0625) \\ &= 8.125 \text{ Volts} \end{aligned}$$

2. For a 5 bit binary weighted resistor assume '0' = 0V and '1' = +10V. Find the output voltage for following digital input 10110.

उत्तर:

$$V_o = V_R \left[ \frac{b_{N-1}}{2^0} + \frac{b_{N-2}}{2^1} + \dots + \frac{b_0}{2^{N-1}} \right]$$

Input bits: 101001

$$\begin{aligned} V_o &= 10[(1 \times 2^{-1}) + (0 \times 2^{-2}) + (1 \times 2^{-3}) + (1 \times 2^{-4}) + (0 \times 2^{-5})] \\ &= 10[0.5 + 0.125 + 0.0625] \\ &= 6.875V \end{aligned}$$

### 5.1.3.2 R-2R लॅडर DAC

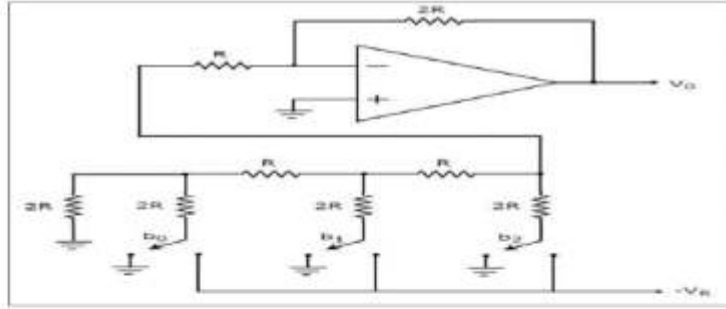
लॅडर नेटवर्कमध्ये फक्त दोन रेसिस्टर व्हॅल्यूचा संच असतो, R आणि 2R. R/2R नेटवर्क डिजिटल टू ऑनॅलॉग रूपांतरणाची सर्वात अचूक पद्धत देते. R/2R लॅडरचा Thevenin's resistance नेहमी R असतो.

**R-2R DAC चे उपयोग**



1. ऑडिओ अॅम्प्लिफायर
2. व्हिडिओ एन्कोडर
3. डेटा संपादन प्रणाली
4. कॅलिब्रेशन
5. ऑसिलोस्कोप
6. मोटरनियंत्रक

3-बिट R-2R लॅडर DAC सर्किट डायग्राम आकृती 5.4 मध्ये दर्शविले आहे-



आकृती 5.4 3-बिट R-2R लॅडर DAC सर्किट डायग्राम

बायनरी संख्येच्या बिट्समध्ये दोनपैकी फक्त एक मूल्य असू शकते 0 'or' 1. 3-बिट बायनरी इनपुट  $b_2 b_1 b_0$  असते. बिट्स  $b_2$  आणि  $b_0$  अनुक्रमे MSB आणि LSB दर्शवतात.

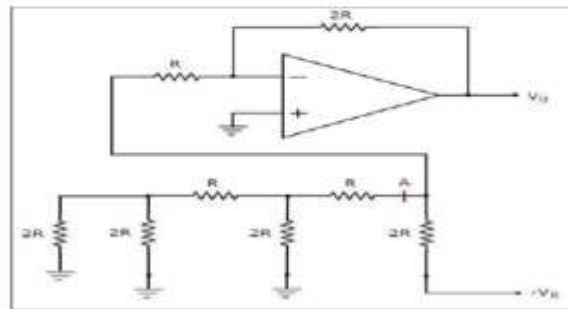
आकृती 5.4 मध्ये दर्शविलेले डिजिटल स्विच ग्राउंडशी जोडले जातील, जेव्हा संबंधित इनपुट बिट '0' असतील, डिजिटल स्विच निगेटिव्ह रेफरेंस वोल्टेज,  $-V_R$  शी जोडले जातील जेव्हा संबंधित इनपुट बिट '1' असतील.

**उदाहरण:**

Find the value of analog output voltage of R-2R Ladder DAC for a binary input,  $b_2 b_1 b_0 = 100$   
Circuit Diagram and its Simplification

उत्तर:

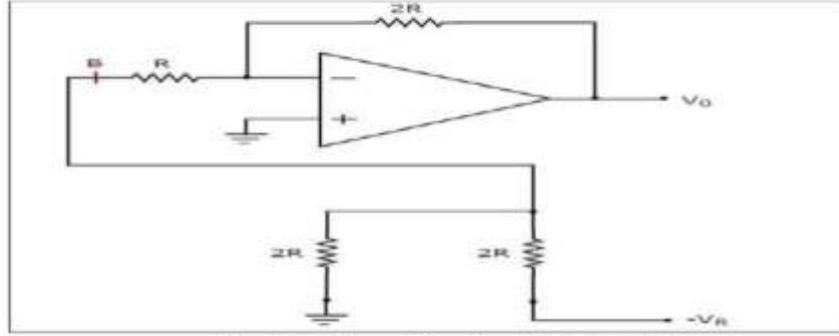
बायनरी इनपुट,  $b_2 b_1 b_0 = 100$  लागू केल्यावर 3-बिट R-2R शिडी DAC चे सर्किट आकृती -5.5 आकृतीमध्ये दाखवले आहे.



आकृती 5.5 3-बिट R-2R लॅडर DAC सर्किट डायग्राम

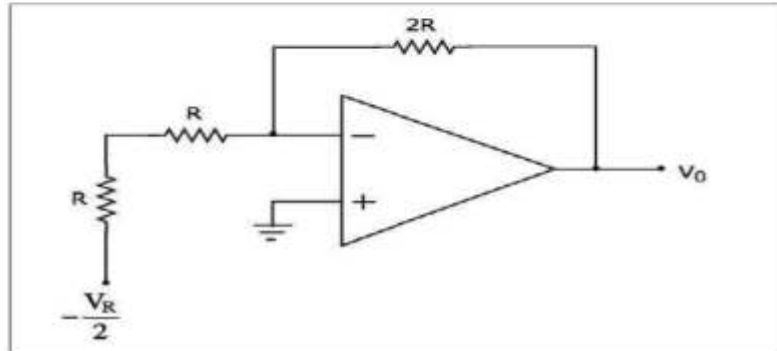
वरील सर्किटमध्ये, ग्राउंडच्या संदर्भात बिंदू A च्या डावीकडे रेसिस्टरची मालिका आणि समांतर (parallel) कॉम्बिनेशनमध्ये अस्तित्वात आहेत. तर, संपूर्ण रेसिस्टर नेटवर्क  $2R \Omega$  च्या रेसिस्टन्स असलेल्या एका रेसिस्टरने बदलू शकतो.

सिम्पलीफाईड सर्किट आकृती -5.6 मध्ये दर्शविली आहे



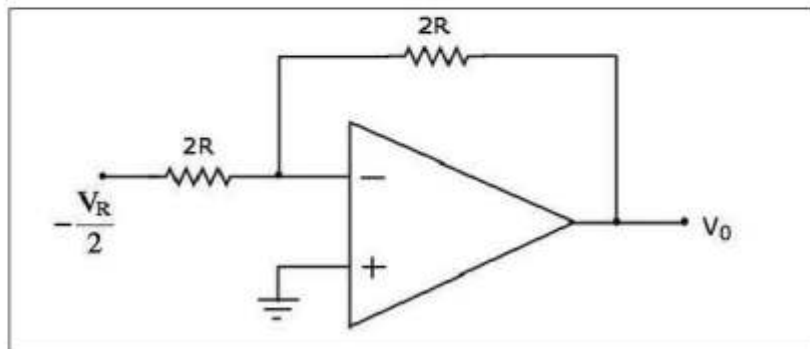
आकृती 5.6- सिम्पलीफाईड R-2R ladder

Thevenin च्या समतुल्य (equivalent) सर्किट वापरून बिंदू B च्या डावीकडे जोडलेल्या नेटवर्कचा भाग ग्राउंडच्या संदर्भात बदलू शकतो. सुधारित सर्किट आकृती 5.7 मध्ये दर्शविली आहे



आकृती 5.7 - सुधारित R-2R ladder

वरील सर्किटमध्ये, दोन रेसिस्टरचे कॉम्बिनेशन अस्तित्वात आहे. हे कॉम्बिनेशन एकाच रेसिस्टरने बदला. सिम्पलीफिकेशन नंतरचे फायनल सर्किट आकृती आकृती 5.8 मध्ये दर्शविले आहे



आकृती 5.8 – फायनल R-2R ladder

आता, वरील सर्किट डायग्राम इनव्हर्टिंग ऑप्लिफायर सारखी दिसते. यात  $-V_R/2$  व्होल्टचा इनपुट व्होल्टेज,  $2R \Omega$  चा इनपुट रेसिस्टर आणि  $2R \Omega$  चा फीडबॅक रेसिस्टर आहे. वर दर्शविलेल्या सर्किटचे आउटपुट व्होल्टेज ,

$$V_0 = -2R/2R (-V_R/2)$$

$$V_0 = V_R/2$$

म्हणून, बायनरी इनपुट  $b_2b_1b_0 = 100$  साठी 3-बिट R-2R Ladder DAC चे आउटपुट व्होल्टेज  $V_R/2$  व्होल्ट इतके आहे.

### फायदे:

1. R-2R लॅडर DAC मध्ये रेसिस्टरच्या फक्त दोनच व्हॅल्यू असतात, R आणि  $2R$ . त्यामुळे, अधिक अचूक रेसिस्टर निवडणे आणि डिझाइन करणे सोपे आहे.
2. जर डिजिटल इनपुटमध्ये अधिक संख्येने बिट्स असतील, तर आवश्यक संख्येने R-2R विभाग देखील समाविष्ट करावे लागतील.
3. यामुळे, बायनरी वेटेड रेसिस्टर DAC पेक्षा R-2R लॅडर DAC उपयोगी आहे.

### उदाहरण

1. Calculate analog output of 4 bit DAC for digital input 1101.

Assume  $V_{FS} = 5V$ .

उत्तर:

$$V_R = V_{FS}$$

$$V_O = V_{FS}(2^{N-1}b_{N-1} + 2^{N-2}b_{N-2} + \dots + 2^1b_1 + 2^0b_0)$$

$$= 5[(1 \times 2^{-1}) + (1 \times 2^{-2}) + (0 \times 2^{-3}) + (1 \times 2^{-4})]$$

$$= 5[0.5 + 0.25 + 0 + 0.0625]$$

$$= 4.0625 \text{ Volts}$$

2. For a 6 bit binary R-2R ladder, assume '0' = 0V and '1' = +10V. Find the output voltage for following digital input 101001

Solution:

$$V_O = V_{FS}(2^{N-1}b_{N-1} + 2^{N-2}b_{N-2} + \dots + 2^1b_1 + 2^0b_0)$$

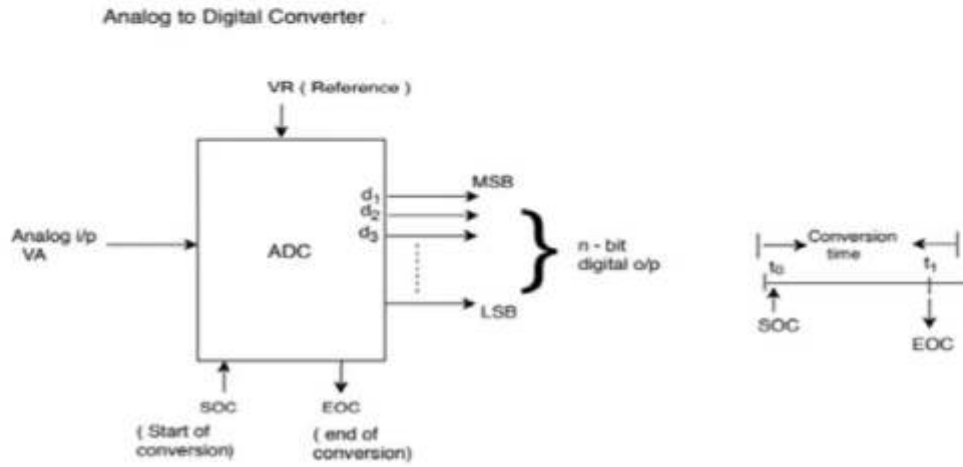
Input bits: 101001

$$V_O = 10[(1 \times 2^{-1}) + (0 \times 2^{-2}) + (1 \times 2^{-3}) + (0 \times 2^{-4}) + (0 \times 2^{-5}) + (1 \times 2^{-6})]$$

$$= 6.40625V$$

### 5.1.4 अॅनालॉग टू डिजिटल कन्वर्टर (Analog to Digital Converter)

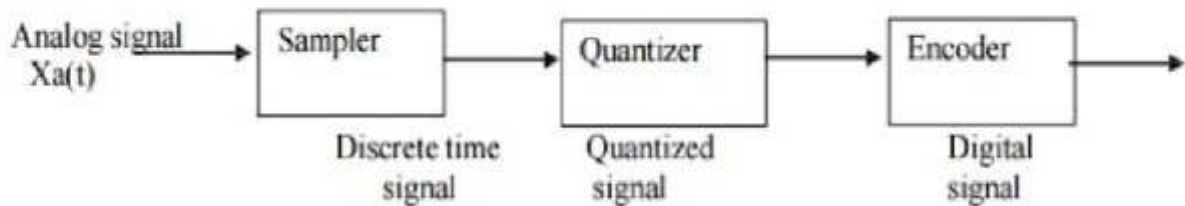
ADC ब्लॉक डायग्राम आकृती 5.9 मध्ये दर्शविली आहे



आकृती 5.9-ADC ब्लॉक डायग्राम

ADC चे उपयोग:

- 1) सेलफोन डिजिटल व्हॉइस सिग्नलवर काम करतात .आवाज हा अॅनालॉग स्वरूपात असतो, जो सेलफोन ट्रान्समीटरला फीड करण्यापूर्वी ADC द्वारे रूपांतरित केला जातो.
- 2) कॅमेरा वापरून कॅपचर (capture) केलेल्या इमेजेस आणि व्हिडिओ कोणत्याही डिजिटल उपकरणामध्ये संग्रहित केले जातात, ADC वापरून डिजिटल स्वरूपात रूपांतरित केले जातात.
- 3) क्ष-किरण (x-ray) आणि MRI सारखे वैद्यकीय इमेजेस देखील मॉडिफिकेशन पूर्वी डिजिटल स्वरूपात रूपांतरित करण्यासाठी ADC चा वापर करते .नंतर ते अधिक चांगल्या प्रकारे समजून घेण्यासाठी सुधारित केले जातात.
- 4) डिजिटल ऑसिलोस्कोपमध्ये (Digital Oscilloscope) अॅनालॉग सिग्नलला डिस्प्लेसाठी आणि इतर विविध वैशिष्ट्यांसाठी डिजिटल सिग्नलमध्ये रूपांतरित करण्यासाठी ADC देखील आहे.
- 5) एअर कंडिशनरमध्ये खोलीचे तापमान राखण्यासाठी तापमान सेन्सर असतात व त्याचे आउटपुट कॅन्व्हर्ट करण्यासाठी ADC चा वापर करतात.



आकृती 5.10-अॅनालॉग टू डिजिटल कन्वर्टर ब्लॉक डायग्राम



## ॲनालॉग टू डिजिटल रूपांतरण स्टेप्स

### 1. सॅम्पल:

सॅम्पल ब्लॉकचे काम विशिष्ट वेळेच्या अंतराने इनपुट ॲनालॉग सिग्नलचे सॅम्पल घेणे हे आहे. ते सॅम्पल कन्टीन्युअस विपुलता (continuous amplitude) च्या स्वरूपात घेतले जातात

### होल्ड:

ADC मध्ये वापरलेला दुसरा ब्लॉक म्हणजे 'होल्ड' ब्लॉक. पुढील सॅम्पल घेईपर्यंत ते फक्त सॅम्पल होल्ड करते. पुढील सॅम्पलपर्यंत होल्ड व्हॅल्यू अपरिवर्तित (unchanged) राहते.

### क्वांटाइझर (Quantizer):

हा ब्लॉक क्वांटाइझेशनसाठी वापरला जातो. ॲनालॉग किंवा कन्टीन्युअस विपुलताचे डिसक्रीट विपुलता मध्ये रूपांतरित केले जाते.

### एनकोडर (Encoder):

एनकोडर ब्लॉक डिजिटल सिग्नलला बायनरी (digital) फॉर्ममध्ये म्हणजेच बिट्समध्ये रूपांतरित करतो. डिजिटल उपकरणे बायनरी सिग्नलवर कार्य करतात म्हणून एनकोडर वापरून डिजिटल सिग्नलचे बायनरी स्वरूपात रूपांतर करणे आवश्यक आहे.

ॲनालॉग टू डिजिटल कन्व्हर्टर वापरून ॲनालॉग सिग्नलला डिजिटल स्वरूपात रूपांतरित करण्याची ही संपूर्ण प्रक्रिया आहे हे संपूर्ण रूपांतरण काही मायक्रो सेकंद मध्ये होते.

### ADC चे प्रकार:

ADC चे दोन प्रकार आहेत डायरेक्ट टाइप ADC आणि इनडायरेक्ट टाइप ADC .

डायरेक्ट टाइप ADC – चीउदाहरणे

1. काउंटरटाइप (Counter type) ADC
2. सक्सेसिव्ह ॲप्रॉक्सिमेशन टाइप (Successive Approximation type) ADC
3. फ्लॅश टाइप (Flash type) ADC

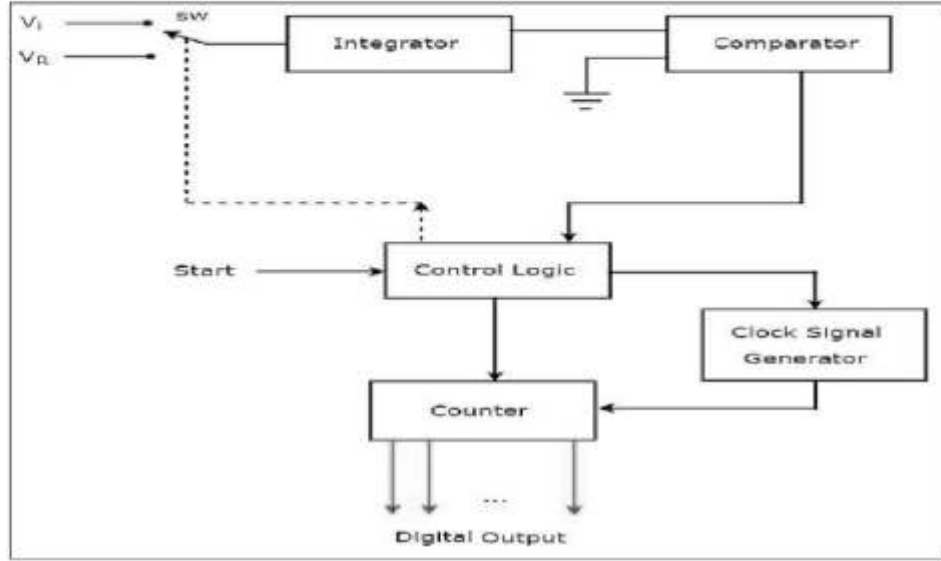
इनडायरेक्ट प्रकारच्या ADC ची उदाहरणे

1. सिंगल स्लोप (Single Slope) ADC
2. ड्युअल स्लोप (Dual Slope) ADC

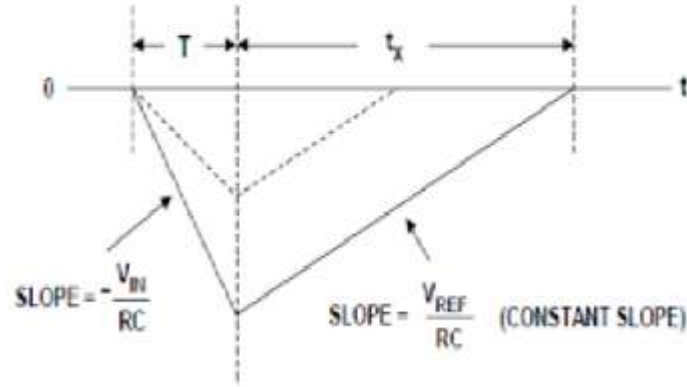
ड्युअल स्लोप ADC हे अप्रत्यक्ष प्रकारच्या ADC चे सर्वोत्तम उदाहरण आहे.

### 5.1.4.1 .ड्युअल स्लोप (Dual slope) ADC

ड्युअलस्लोप ADC (दोन) Dual स्लोप तंत्राचा वापर करून संबंधित ॲनालॉग इनपुटसाठी समतुल्य (equivalent) डिजिटल आउटपुट तयार करते. ड्युअल स्लोप ADC चे ब्लॉक डायग्राम आणि त्याचे वेव्हफॉर्म अनुक्रमे आकृती क्रमांक 5.11 आणि, 5.12 मध्ये दर्शविले आहेत.



आकृती 5.11- ड्युअल स्लोप ADC ब्लॉक डायग्राम



आकृती 5.12- ड्युअल स्लोप ADC वेव्हफॉर्म

ड्युअल स्लोप ADC मध्ये प्रामुख्याने 5 ब्लॉक्स असतात, इंटिग्रेटर, कंप्यारेटर, क्लॉक सिग्नल जनरेटर, कंट्रोल लॉजिक आणि काउंटर. जेव्हा कंट्रोल लॉजिक एसओसी (SOC, Start of Conversion) सिग्नल रीसिव्ह करतो तेव्हा काउंटर रीसेट होतो आणि क्लॉक सिग्नल जनरेटर क्लॉक पल्स पाठवतो. जेव्हा एसओसी (SOC) सिग्नल प्राप्त होतो तेव्हा कंट्रोल लॉजिक बाह्य अॅनालॉग इनपुट व्होल्टेज  $V_i$  शी कनेक्ट करण्यासाठी स्विचला जोडते. इनपुट इंटिग्रेटरला ऑपलाय केले जाते.

इंटिग्रेटरचे आऊटपुट कंप्यारेटरच्या दोन इनपुटपैकी एकाशी जोडलेले असते आणि इंटिग्रेटरचे दुसरे इनपुट ग्राउंडशी जोडलेले असते. कंप्यारेटर इंटिग्रेटरच्या आऊटपुटची तुलना शून्य व्होल्ट (ग्राउंड) बरोबर करतो आणि आऊटपुट तयार करतो, जे कंट्रोल लॉजिकवर ऑपलाय केले जाते. प्रत्येक क्लॉक पल्ससाठी काउंटर एकने वाढेल आणि त्याचे मूल्य बायनरी (digital) स्वरूपात असेल. जेव्हा ते कमाल (maximum) काउन्ट व्हॅल्यूपर्यंत पोहोचेल त्यावेळी काउंटर कंट्रोल लॉजिकसाठी ओव्हरफ्लो सिग्नल तयार करते. यावेळी काउंटरचे सर्व बिट्स



केवळ शून्य असतील. कंट्रोल लॉजिक निगेटिव्ह रेफरन्स व्होल्टेज  $-V_{ref}$  शी जोडण्यासाठी स्विचला पुश (push) करते.हे निगेटिव्ह रेफरन्स व्होल्टेज इंटिग्रेटरवर अॅपलाय केले जाते. हे कॅप्यासीटरमध्ये साठवलेले चार्ज शून्य होईपर्यंत काढून टाकते.

**फायदे:**

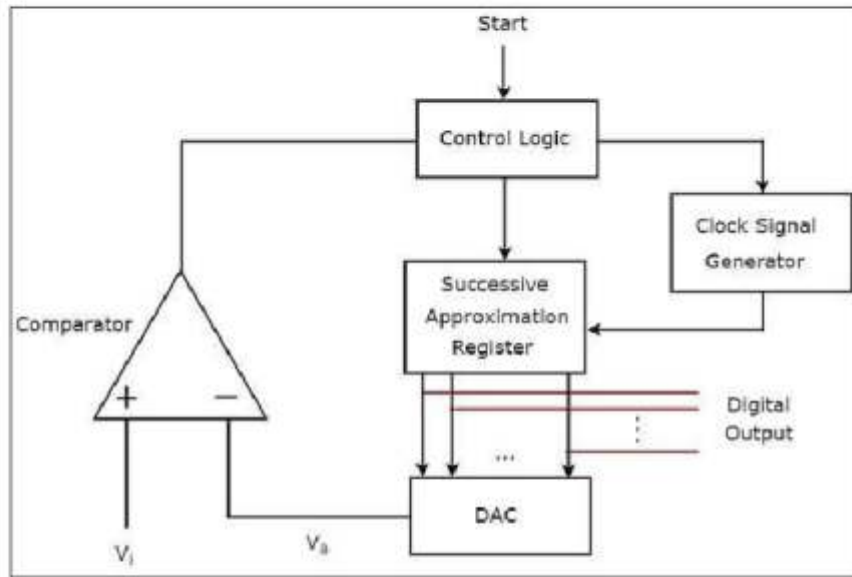
1. हे सोपे आणि तुलनेने स्वस्त आहे.
2. यात उच्च रूपांतरण अचूकता आहे.(High conversion accuracy)
3. वेळ, तापमान आणि इनपुट व्होल्टेजचा परिणाम होत नाही.
4. स्थिरतेसाठी (stability) क्रिस्टल ऑसिलेटरची आवश्यकता नाही.

**तोटे:**

1. इतर कोणत्याही ADC च्या तुलनेत याच्या रूपांतरणाची वेळ जास्त आहे.
2. याचा रूपांतरणाचा (conversion) वेग खूपच कमी आहे.

#### 5.1.4.2 सक्सेसिव्ह अॅप्रॉक्सिमेशन टाइप ADC

सक्सेसीव्ह अॅप्रॉक्सिमेशन ADC ब्लॉक डायग्राम आकृती -5.13मध्ये दर्शविले आहे-



**आकृती 5.13- सक्सेसिव्ह अॅप्रॉक्सिमेशन रेजिस्टर ब्लॉक डायग्राम**

कंट्रोल लॉजिक SAR (Successive Approximation Register) चे सर्व बिट्स रीसेट करते आणि स्टार्ट ऑफ कन्व्हर्जन सिग्नल (SOC) प्राप्त झाल्यावर SAR ला क्लॉक पल्स पाठवण्यासाठी क्लॉक सिग्नल जनरेटर एनेबल करते. SAR मध्ये उपस्थित (present) असलेला digital डेटा कॅंप्यारेटरच्या आउटपुटवर आधारित प्रत्येक क्लॉक पल्ससाठी अपडेट केला जाईल. SAR चे आउटपुट DAC च्या इनपुट म्हणून अॅपलाय केले जाते.

DAC ला प्राप्त झालेले डिजिटल इनपुट, जे SAR चे आउटपुट आहेत अॅनालॉग आउटपुटमध्ये कनव्हर्ट करते. कॅंप्यारेटर या अॅनालॉग व्हॅल्यू  $V_a$  ची बाह्य अॅनालॉग इनपुटमूल्य  $V_i$  शी तुलना

करतो.जोपर्यंत  $V_i$  हा  $V_a$  पेक्षा मोठा असेल तोपर्यंत कंप्यारेटरचे आउटपुट '1' असेल .त्याचप्रमाणे, कंप्यारेटरचे आउटपुट '0' असेल, जेव्हा  $V_i$  पेक्षा कमी किंवा समान असेल. वरील ऑपरेशन्स डिजिटल आउटपुट व्हॅलिड होईपर्यंत सुरू ठेवली जातील.

**फायदे:**

1. याची कन्व्हर्जन गती अधिक (high) आहे.
2. यात कन्व्हर्जन टाइम स्थिर (constant) असतो.
3. हे फक्त  $n$  क्लॉकच्या पल्समध्ये  $n$  बिट A दू D रूपांतरण करू शकते.

**तोटे:**

1. हा ADC महाग आहे.
2. सर्किट गुंतागुंतीचे (complex) आहे.
3. यासाठी DAC आवश्यक आहे.

डी दू डी कन्व्हर्टरस आणि ए दू डी कन्व्हर्टरस मॅनुफॅक्चरने निर्दिष्ट (specified) केलेल्या विस्तृत (wide) वैशिष्ट्यांसह उपलब्ध आहेत . रिझोल्यूशन अचूकता, रेखीयता, मोनोटोनिसिटी, रूपांतरण वेळ, सेटलिंग वेळ आणि स्थिरता ही काही महत्वाची वैशिष्ट्ये आहेत.

### 1. रिझोल्यूशन: (Resolution)

कन्व्हर्टर रिझोल्यूशन हे व्होल्टेज मधील सर्वात लहान बदल आहे जे कन्व्हर्टरच्या आउटपुटवर (किंवाइनपुट) तयार केले जाऊ शकते. उदाहरणार्थ, 8-बिट D/A कनवर्टरमध्ये  $2^8 - 1 = 255$  समान intervals असतात. म्हणून आउटपुट व्होल्टेजमधील सर्वात लहान बदल पूर्ण स्केल आउटपुट रेंजच्या  $1/255$  आहे. रिझोल्यूशन शक्य तितके कमी असावे. हे DAC ला दिलेल्या डिजिटल इनपुटमधील बिट्सच्या संख्येवर अवलंबून असते. बिट्सची संख्या जितकी जास्त तितके रिझोल्यूशन कमी असते. डिजिटल इनपुटवर 1 LSB च्या बदलामुळे अॅनालॉग आउटपुट व्होल्टेजमधील बदलाचे गुणोत्तर(ratio) म्हणून देखील म्हटले जाऊ शकते.  $n$  बिट DAC साठी खालील समीकरण वापरतात.

$$\text{Resolution} = \frac{V_{FS}}{2^n - 1}$$

$$\% \text{ Resolution} = \frac{V_{FS}}{2^n - 1} \times 100$$

### 2. रेखीयता (Linearity)

डिजिटल इनपुट आणि अॅनालॉग आउटपुट मधील संबंध (relation) रेखीय (linear) असावा.

### 3. अचूकता (Accuracy)

परिपूर्ण (Absolute) अचूकता म्हणजे actual कन्व्हर्टर आउटपुट आणि Ideal कन्व्हर्टर आउटपुट मधील कमाल तफावत (maximum deviation). सापेक्ष (Relative) अचूकता हे गेन (gain) आणि ऑफसेट एरर (offset error) काढून टाकल्यानंतरची कमाल तफावत (maximum deviation) आहे.

#### 4. निश्चितीची वेळ (Settling time)

सेटलिंग टाइम हा डिजिटल इनपुटमधील बदलानंतर, त्याच्या अंतिम व्हॅल्यूच्या निर्दिष्ट (specified) बँड[ $\pm(1/2)$  LSB] मध्ये आउटपुटला सेटल होण्यासाठी लागणारा वेळ दर्शवतो. ते शक्य तितके लहान असावे.

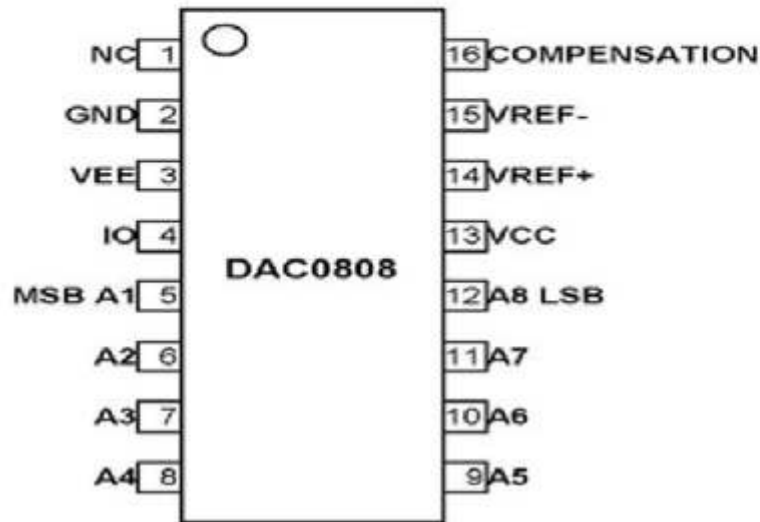
#### 5. मोनोटोनिसिटी (Monotonicity)

मोनोटोनिक DAC चे अॅनालॉग आउटपु डिजिटल इनपुट नुसार वाढते. जर DAC मोनोटोनिक असेल, तर एररही प्रत्येक आउटपुट स्तरा(level) वर  $\pm(1/2)$  LSB पेक्षा कमी असावी.

#### 6. स्थिरता (Stability)

कन्व्हर्टरचा परफॉर्मन्स हा तापमान, वेळ आणि वीज पुरवठ्यातील बदलानुसार बदलतो. त्यामुळे ऑफसेट, गेन, linearity एरर आणि मोनोटोनिसिटी यासारखे सर्वसंबंधित पैरामीटर्स पूर्ण(full) तापमान आणि वीजपुरवठा श्रेणीमध्ये निर्दिष्ट(specified)करणे आवश्यक आहे.

#### 5.1.5 DAC IC 0808 पिन कॉन्फिगरेशन:-



आकृती 5.14- DAC 0808 पिन डायग्राम

#### IC DAC 0808 पिन कॉन्फिगरेशन:

1. पिन 1 (NC)- कनेक्शन नाही
2. पिन 2 (GND)-ग्राउंड पिन
3. पिन 3 (VEE)-निगेटिव्ह (-ve) वीज पुरवठा
4. पिन 4 (IO)-इनपुट/आउटपुट सिग्नल पिन
5. Pin 5 (A1)- MSB (डिजिटल i/p बिट-1)
6. पिन 6 (A2)-digital i/p बिट-2
7. पिन (A3)-digital i/p बिट-3
8. पिन 8 (A4)-digital i/p बिट-4



9. पिन 9 (A5)-digital i/p बिट-5
10. पिन 10 (A6)-digital i/p बिट-6
11. पिन 11 (A7)-digital i/p बिट- 7
12. पिन 12 (A8)-digital i/p बिट-8
13. पिन 13 (VCC)-पॉझिटिव्ह (+ve) पॉवर सप्लाय
14. पिन 14 (V<sub>REF</sub>+) -पॉझिटिव्ह (+ve) रेफरन्स व्होल्टेज
15. पिन 15 (V<sub>REF</sub>-) -निगेटिव्ह (-ve) रेफरन्स व्होल्टेज
16. पिन 16 (कॉम्पेन्सेशन)- कॉम्पेन्सेशन कप्यासीटर पिन

#### वैशिष्ट्ये :

1. सापेक्ष अचूकता (Relative accuracy):  $\pm 0.19\%$  कमाल एरर (maximum)
2. जलद सेटलिंग वेळ: 150 ns
3. नॉन -इन्व्हर्टिंग (non-inverting) डिजिटल इनपुट TTL आणि CMOS शी सुसंगत आहेत
4. उच्च गती मल्टिप्लायिंग (multiplying) Input slew rate: 8mA/ $\mu$ s
5. वीज पुरवठा व्होल्टेज :  $\pm 4.5V$  ते  $\pm 18V$
6. लो पॉवर कन्सम्पशन (consumption): 33mW @  $\pm 5V$

#### Specifications:

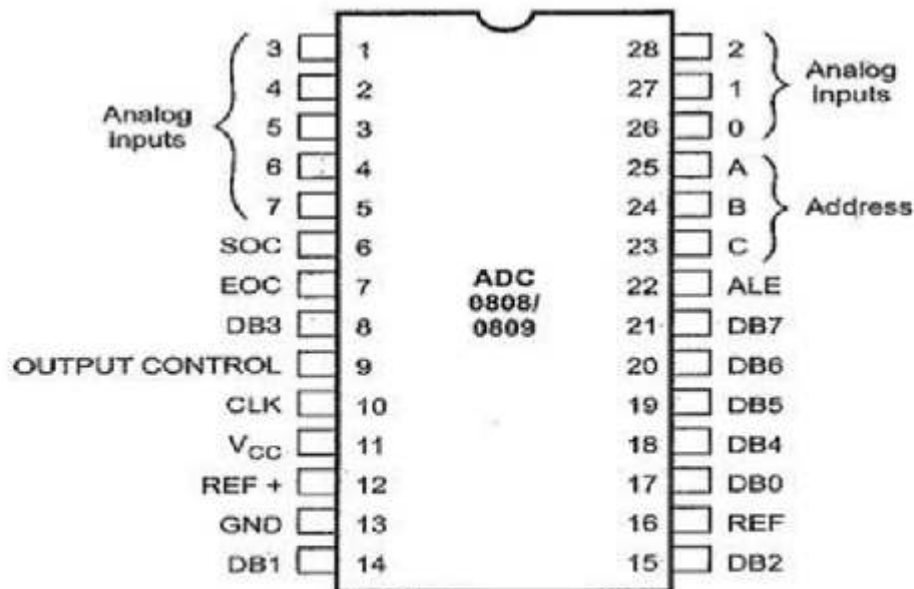
पॅरामीटर्स	तपशील
भाग क्रमांक	DAC0808
रिझोल्यूशन	8-bits
DAC चॅनेल	1
इंटरफेस	समांतर (Parallel)
ऑपरेटिंग तापमान श्रेणी(C)	0 ते 70
आर्किटेक्चर	गुणाकार DAC
आउटपुट प्रकार	अन बफर केलेले करंट (Un buffered) Current)
आउटपुट रेंज (कमाल) (V) ते (किमान) (V)	0 - 4.2
सेटलिंगवेळ ( $\mu$ s)	0.15

#### ADC चे तपशील (Specificationsof ADC):

1. अॅनालॉग इनपुट व्होल्टेज श्रेणी(**range**) - ही जास्तीत जास्त स्वीकार्य (allowable) इनपुट-व्होल्टेज (range) आहे ज्यामध्ये ADC योग्यरित्या कार्य करेल. डिजिटल आउटपुटची संपूर्ण रेंज वापरण्यासाठी सर्वात लहान आणि सर्वात मोठ्या अॅनालॉग इनपुट व्होल्टेजमधील फरक आहे. ठराविक व्हॅल्यू 1 ते 10V, 0 ते 12V,  $\pm 5V$ ,  $\pm 10V$ , आणि  $\pm 12V$  आहेत.

2. इनपुट इम्पेडन्स-ADCचे इनपुट इम्पेडन्स ADCच्या प्रकारानुसार  $1K\Omega$  (Kohm) ते  $1M\Omega$  (Mohm) पर्यंत बदलते. ADC ची इनपुट कॅप्यासिटन्स काही पिकोफरेड्स (PF) असते.
3. क्वांटायझेशन एरर- क्वांटायझेशन ही प्रक्रिया आहे त्यामुळे येणाऱ्या एररला क्वांटायझेशन एरर म्हणतात. क्वांटायझेशन एरर  $(1/2)LSB$  निर्दिष्ट (specified) केली जाते.
4. अचूकता- ADC ची अचूकता क्वांटायझेशन एरर, डिजिटल सिस्टीम नॉइज, गेन एरर, ऑफसेट एरर आणि रेखीयते (linearity)पासून विचलन(deviation) इत्यादींवर अवलंबून असते. अचूकता सर्व प्रकारच्या एररच्या बेरजेवरून निश्चित केली जाते. अचूकतेची ठराविक व्हॅल्यू  $\pm 0.001\%$ ,  $\pm 0.01\%$ ,  $\pm 0.02\%$ , आणि पूर्ण-स्केल मूल्याच्या  $\pm 0.04\%$  आहेत. परिपूर्ण अचूकता (absolute accuracy)कन्व्हर्टर आउटपुट आणि आदर्श कन्व्हर्टरआउटपुटमधील कमाल विचलन (maximum deviation)आहे. सापेक्ष अचूकता (relative accuracy) ही गेन आणि ऑफसेट एरर काढल्या नंतरचे कमाल विचलन (maximum deviation) आहे.
5. रिझोल्यूशन हे रेफरन्स व्होल्टेज आणि आउटपुट स्टेप्स संख्येचे गुणोत्तर (ratio) आहे.  $LSB$  साठी अॅनालॉग व्होल्टेज मधील हा सर्वात लहान बदल आहे.  $N$  रिझोल्यूशन = रेफरन्सव्होल्टेज /  $(2^n - 1)$  जेथे  $N$  = ADC च्या बिट्सची संख्या.
6. कनव्हरजन टाइम- मध्यम-गती ADC चा कनव्हरजन टाइम सुमारे  $50 \mu s$  आहे आणि हाय स्पीड ADC साठी, कनव्हरजन टाइम काही ns आहे. म्हणून कनव्हरजन टाइम  $50 \mu s$  ते काही ns ते मंद/मध्यम वेग ते हाय स्पीड ADC पर्यंत बदलते. डिजिटल आउटपुटचे स्वरूप युनिपोलरबायनरी, बायपोलरबायनरी, ऑफसेटबायनरी, 1's complement and 2's complement इ. असते.

#### ADC IC 0808/09 पिन कॉन्फिगरेशन



आकृती 5.15 - ADC 0808/0809 पिन बायग्राम

पिन नंबर	पिन नाव	डिस्क्रिप्शन
1 ते 5, 26, 28	अनालॉग चॅनल 1 ते 5	या 8 पिन अॅनालॉग व्होल्टेज (सेन्सर) साठी इनपुट पिन आहेत.
6	स्टार्ट	कन्व्हर्जन सुरू करण्यासाठी या पिनला लॉजिक '1' दिले जाते.
7	कन्व्हर्जनसमाप्ती (EOC)	कन्व्हर्जन पूर्ण झाल्यावर ही पिन उच्च (high) होते.
8, 14, 15, 18, 19, 20, 21	आउटपुट (2 <sup>-1</sup> ते 2 <sup>-7</sup> )	ADC चे आउटपुट देतात.
9	OUT EN (Output Enable)	आउटपुट पिनवर आउटपुट मिळविण्यासाठी ही पिन हाय केले जाते.
10	क्लॉक	(0V-5V) 20 Mhz क्लॉक
11	Vcc	5V
12	Vref(+)	रेफरन्स व्होल्टेज पिन, +5V
13	ग्राउंड	सर्किटच्या ग्राउंडला कनेक्ट असते.
16	Vref(-)	ग्राउंडला जोडलेले असते.
22	अॅड्रेस लॅच एनेबल (ALE)	ही पिन ADC चॅनेल निवडण्यासाठी तात्पुरती हाय केली जाते.
23, 24, 25	ADD A, ADD B, ADD C	चॅनल निवडण्यासाठी वापरतात.

### वैशिष्ट्ये:

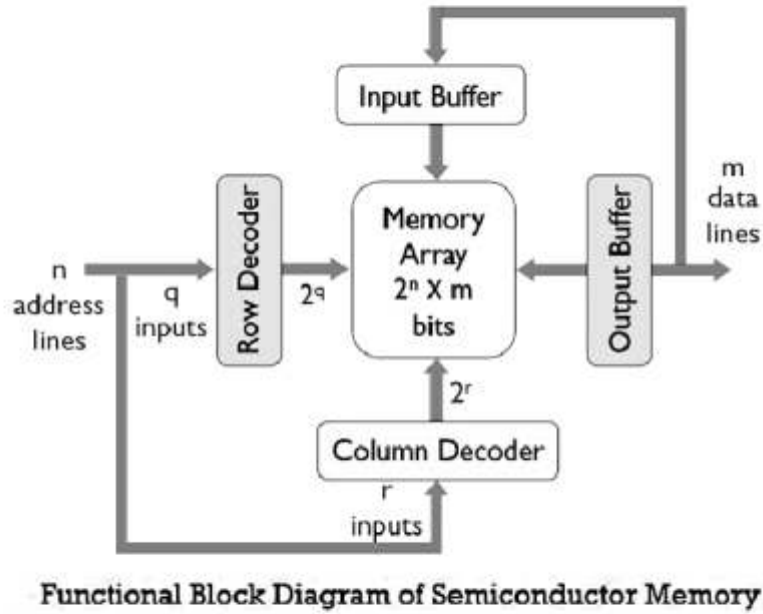
1. सर्व मायक्रोप्रोसेसर इंटरफेस करणे सोपे आहे किंवा स्वतंत्र (stand-alone) कार्य करतो.
2. 8-बिट ADC मॉड्यूल
3. 8 अॅनालॉग व्हॅल्यू अखंडपणे मोजू शकतात.
4. बाह्य ऑसिलेटर आवश्यक आहे (क्लॉक)
5. डिजिटल आउटपुट 0 ते 255 पर्यंत आहेत, ऑपरेटिंग पॉवर 15mW, कन्व्हर्जनवेळ 100us आहे
7. 28- पिन DIP, SOIC पॅकेजेसमध्ये उपलब्ध आहे.

### 5.2.1 सेमीकंडक्टर मेमरी (Semiconductor memory):

सेमीकंडक्टर मेमरी थेट CPU शी जोडलेल्या असतात आणि या मेमरीना CPU कोड आणि डेटा read आणि write म्हणून सेमीकंडक्टर मेमरीना प्राथमिक (primary) मेमरी म्हणतात. सेमीकंडक्टर मेमरीच्या IC मध्ये  $n$  नंबर अॅड्रेसलाईन्स आणि  $m$  डेटा लाईन्स असतात. त्याद्वारे  $2^n \times m$  बिट्सची एकूण मेमरी क्षमता असते. त्यात  $2^n$  मेमरी स्थाने आहेत आणि प्रत्येक स्थान  $m$ -bit पर्यंत डेटा साठवू शकतो.



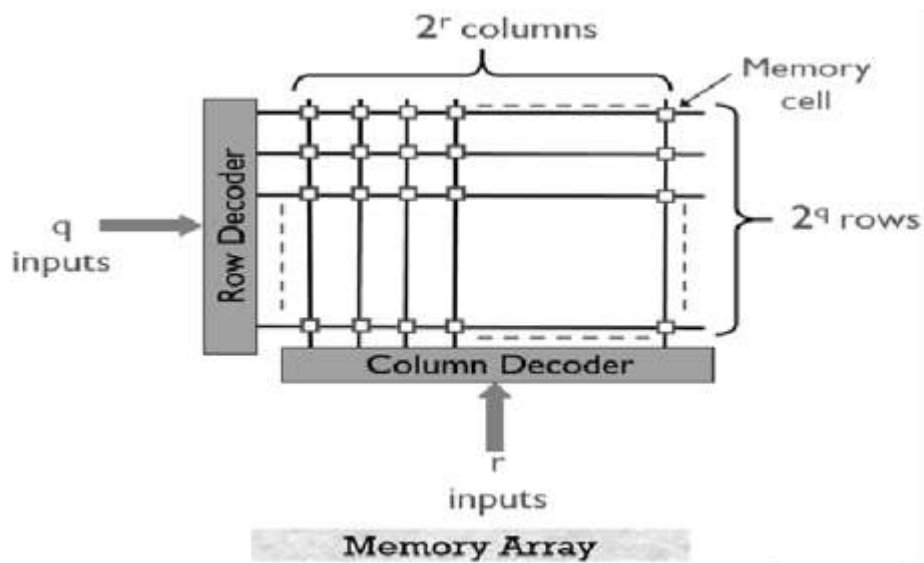
सेमीकंडक्टर मेमरीचे फंक्शनल ब्लॉक डायग्राम आकृती 5.20 मध्ये दिली आहे.



आकृती 5.20-सेमीकंडक्टर मेमरी ब्लॉक डायग्राम

ब्लॉक डायग्राम 5.21 मध्ये मेमरी अ‍ॅरे आणि I/O बफर्ससह एक row आणि column अ‍ॅड्रेस डीकोडर असतात. हे बफर विशिष्ट कालावधीसाठी डेटा ठेवतात.

अ‍ॅड्रेस लाईन्सची एकूण संख्या  $n$ ,  $q$  आणि  $r$  मध्ये विभागली गेली आहे आणि स्वतंत्रपणे row आणि column डीकोडरमध्ये इनपुट म्हणून प्रोव्हाइड केले आहेत. या दोन डीकोडरचे आउटपुट  $2^q$  क्रॉसिंग पॉइंट्स असलेले  $2^q \times 2^r$  आकाराचे मॅट्रिक्स अ‍ॅरे बनवते.



आकृती 5.21-मेमरी अ‍ॅरे

या क्रॉसिंग पॉइंट्सना मेमरी सेल म्हणतात आणि प्रत्येक मेमरी सेलमध्ये बायनरी एक-बिट साठविण्याची क्षमता असते म्हणून, प्रोसेसर मेमरी IC ला अँड्रेस पाठवतो तेव्हा row आणि column डीकोडर एक लाईन निवडतो, जे मॅट्रिक्स मधून मेमरी सेल निवडते.

प्रोसेसरने पाठवलेल्या ऍड्रेसद्वारे मेमरी सेल निवडले जातात, जनरेट झालेल्या कंट्रोल सिग्नलनुसार डेटा विशिष्ट निवडलेल्या मेमरी सेलमध्ये वाचला किंवा लिहिला जाऊ शकतो. मेमरी चिप वापरून परफॉर्म केले जाणारे दोन मूलभूत ऑपरेशन्स म्हणजे "वाचणे (Read)", ज्यामध्ये मेमरी word चे डेटा कन्टेन्ट वाचले जातात आणि "लिहा (Write)" ज्यामध्ये डेटा मेमरी वर्डमध्ये साठवला जातो, तेथे साठवलेला कोणताही डेटा बदलून डेटा रेट वाढवण्यासाठी, काही लेटेस्ट प्रकारच्या मेमरी चिप्समध्ये जसे की DDR, SDRAM मध्ये प्रत्येक वाचन किंवा लेखन ऑपरेशनसह अनेक शब्दांमध्ये अॅक्सेस केला जातो.

सेमीकंडक्टर मेमरीच्या मेमरी सेल्स रेजिस्टर आणि कॅपॅसिटर सारख्या निष्क्रिय (passive) घटकांपासून बनवल्या गेल्या होत्या. त्यानंतर डायोड्सचाही वापर करण्यात आला. परंतु नवीन तंत्रज्ञानाने, Bipolar आणि MOS ट्रान्झिस्टरने डायोड, रेजिस्टर आणि कॅपॅसिटरची जागा घेतली. मेमरी सेल CMOS आणि HMOS तंत्रज्ञान वापरून बनवले जातात ज्यात कमी उर्जा वापरासह हाय ऑपरेशनल गती असते.

सर्व सेमीकंडक्टर मेमरीमध्ये सामान्य असलेल्या काही महत्वाच्या बाबी म्हणजे capacity, organization आणि speed.

### 1. Memory capacity:

सेमीकंडक्टर मेमरी चिप (chip) साठवू शकणाऱ्या बिट्सच्या संख्येला चिप कॅपॅसिटी म्हणतात. हे Kbits (kilobits), Mbits (megabits) इत्यादी युनिट्समध्ये असतात. मेमरी चिपची मेमरी क्षमता नेहमी बिट्समध्ये दिली जाते, संगणक प्रणालीची मेमरी क्षमता बाइट्स (Bytes) मध्ये असते.

### 2. Memory Organisation:

ब्लॉक डायग्राम आकृती 5.21 मध्ये दाखवल्या प्रमाणे मेमरी चिप्स अनेक ठिकाणी organize केल्या आहेत. प्रत्येक लोकेशन 1 बिट, 4 बिट, 8 बिट किंवा अगदी 16 बिट्स होल्ड करू शकते. मेमरी चिपमधील प्रत्येक स्थानामध्ये ठेवू शकणाऱ्या बिट्सची संख्या नेहमी चिपवरील डेटा पिनच्या संख्ये इतकी असते. मेमरी चिपमध्ये असलेल्या स्थानांची संख्या ऍड्रेस पिनच्या संख्येवर अवलंबून असते. मेमरी लोकेशन्सची संख्या नेहमी अड्रेस पिनच्या संख्येच्या 2 च्या पॉवर इतकी असते.

1. मेमरी चिपमध्ये M लोकेशन्स असतात, जेथे n ही अँड्रेस पिनची संख्या असते.

2. प्रत्येक लोकेशनमध्ये m-bits असतात, जेथे m ही चिपवरील डेटा पिनची संख्या असते.

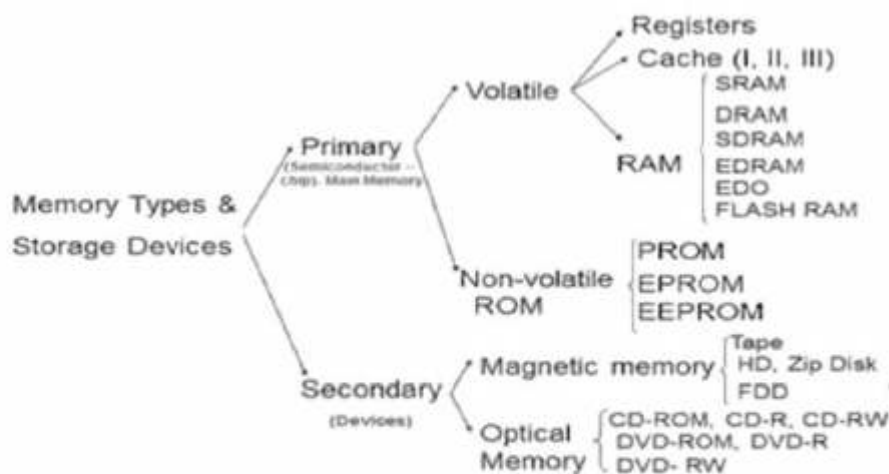
संपूर्ण चिपमध्ये  $2^n \times m$  बिट्स असतील, n ही अँड्रेस पिनची संख्या आहे आणि N ही चिपवरील डेटा पिनची संख्या आहे.

### 3. Speed:

मेमरी चिपचे सर्वात महत्वाचे वैशिष्ट्य म्हणजे त्याचा डेटा वेगाने अॅक्सेस केला जाऊ शकतो. डेटा memory मध्ये प्रवेश करण्यासाठी, अॅड्रेस हा अॅड्रेस पिनवर पाठवला जातो, READ पिन ऍक्टिव्हेट केली जाते आणि ठराविक वेळेनंतर डेटा हा डेटा पिनवर दिसतो. हा वेळ कमी असेल, मेमरी चिप अधिक महाग असते. मेमरी चिपचा वेग सामान्यतः त्याचा अॅक्सेस टाइम म्हणतात. डिझाईन आणि फॅब्रिकेशन प्रक्रियेत वापरल्या जाणाऱ्या IC तंत्रज्ञानावर अवलंबून, मेमरी चिप्सचा अॅक्सेस टाइम काही नॅनो सेकंदांपासून शेकडो नॅनो सेकंदां पर्यंत बदलतो.

### सेमीकंडक्टर मेमरीचे वर्गीकरण:

मेमरीचे तपशीलवार वर्गीकरण आकृती 5.22 मध्ये दिले आहे.



आकृती 5.22 -मेमरीचे वर्गीकरण

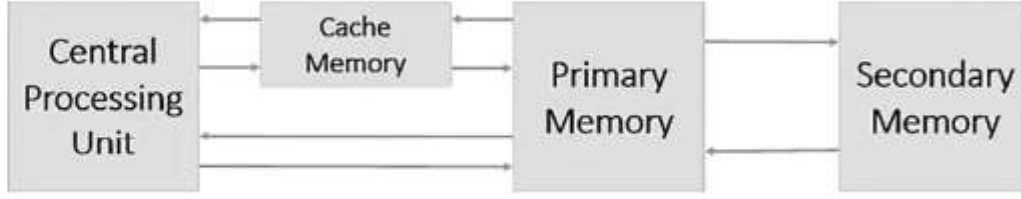
मेमरी प्रामुख्याने तीन प्रकारची असते

- 1.कॅशे मेमरी
- 2.प्राथमिक मेमरी/मुख्य मेमरी
- 3.दुय्यम मेमरी

### 1.कॅशे मेमरी:

कॅशे मेमरी ही एक अतिशय हाय स्पीड सेमीकंडक्टर मेमरी आहे त्यामुळे ती CPU चा वेग वाढवते.हे CPU आणि मुख्य मेमरी दरम्यान बफर म्हणून काम करते.हे डेटा आणि प्रोग्रामचे भाग ठेवण्यासाठी वापरले जाते जे CPU द्वारे वारंवार वापरले जातात.डेटा आणि प्रोग्राम्सचे भाग(parts) ऑपरेटिंग सिस्टमद्वारे डिस्कवरून कॅशेमेमरीमध्ये हस्तांतरित (transfer) केले जातात,तेथून CPU त्यांना अॅक्सेस करू शकते.





### आकृती 5.23-कॅशे मेमरी

#### फायदे:

1. कॅशे मेमरी मुख्य मेमरी पेक्षा वेगवान आहे.
2. ही मुख्य मेमरीच्या तुलनेत कमी प्रवेश (access) वेळ वापरते.
3. ही अल्प कालावधीत कार्यान्वित होऊ शकणारा प्रोग्राम साठवते.
4. ही तात्पुरत्या वापरासाठी डेटा साठवते.

#### तोटे:

1. कॅशे मेमरीची क्षमता मर्यादित आहे.
2. हे खूप महाग आहे.

#### कॅशे मेमरीचे उपयोग

1. भ्रमणध्वनी (mobile)
2. संगणक
3. कॅमकॉर्डर
4. अंतःस्थापित प्रणाली (Embedded systems)
5. स्मार्टकार्ड
6. इलेक्ट्रॉनिक्स संगीत वाद्ये
7. वैद्यकीय उपकरणे आणि स्पेसक्राफ्ट
8. ऑटोमोबाईल्स इ.

### 2. प्राथमिक मेमरी/ मुख्य मेमरी:

प्राथमिक मेमरी मध्ये फक्त तो डेटा आणि सूचना असतात ज्यावर संगणक सध्या (currently) कार्यरत आहे. त्याची क्षमता मर्यादित आहे आणि पॉवर बंद केल्यावर डेटा गमावला (loss) जातो. ही सेमीकंडक्टर components चे बनलेले असते. या मेमरी वेगवान नाहीत. प्रक्रिया (process) करण्यासाठी आवश्यक डेटा आणि सूचना (instructions) मुख्य मेमरी मध्ये राहतात. हे RAM आणि ROM या दोन उपश्रेणींमध्ये (subcategories) विभागलेले आहे.

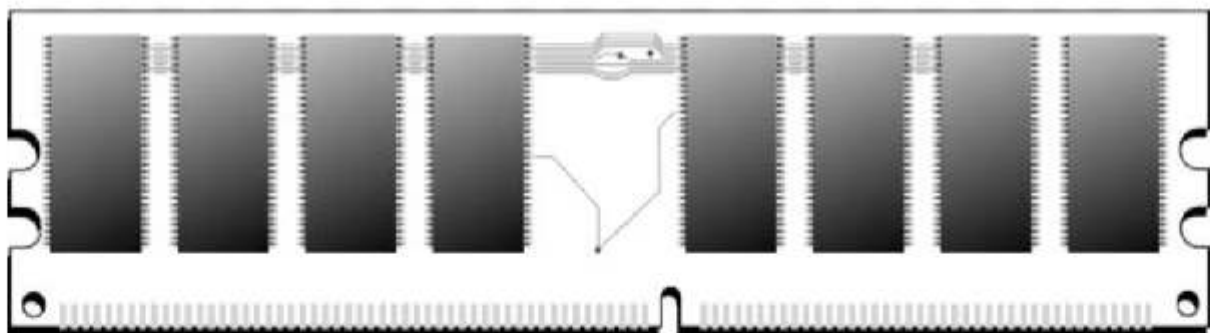
#### मुख्य मेमरीची वैशिष्ट्ये:

1. या सेमीकंडक्टर मेमरी आहेत.
2. हे मुख्य मेमरी म्हणून ओळखले जाते.
3. सहसा अस्थिर (volatile) मेमरी.
4. पॉवर बंद झाल्यास डेटा गमावला जातो.

5. ही संगणकाची कार्यरत मेमरी आहे.
6. दुय्यम (secondary) मेमरीपेक्षा जलद.
7. प्राथमिक मेमरी शिवाय संगणक चालू शकत नाही.

### Random Access Memory (RAM):

RAM-रँडम अॅक्सेस मेमरी (ही डेटा, प्रोग्राम आणि प्रोग्राम परिणाम (result) साठविण्यासाठी CPUअंतर्गत मेमरी आहे. ही Read/Write मेमरी आहे जी मशीन काम करेपर्यंत डेटा संग्रहित करते. मशीन बंद होताच डेटा delete केला जातो.



आकृती 5.24-रँडम अॅक्सेस मेमरी (RAM)

RAM मधील अॅक्सेस टाइम ऍड्रेसपासून स्वतंत्र आहे, म्हणजेच मेमरीमधील प्रत्येक स्टोरेज लोकेशन इतर लोकेशन इतकेच जवळ आहे आणि तेवढाच वेळ लागतो. RAM मधील डेटा रँडमली (randomly) अॅक्सेस केला जाऊ शकतो परंतु रँम खूप महाग आहे.

रँम अस्थिर असते, म्हणजे जेव्हा आपण संगणक बंद करतो किंवा विद्युत प्रवाहामध्ये बिघाड होतो तेव्हा त्यात साठवलेला डेटा गमावला जातो म्हणून. बँकअप अनइंटरप्टिबल पॉवर सिस्टीम (UPS) बहुतेक वेळा संगणकांसह वापरली जाते. रँम त्याच्या भौतिक (physical) आकाराच्या दृष्टीने आणि तो ठेवू शकणाऱ्या डेटाच्या प्रमाणात लहान आहे.

### RAM दोन प्रकारची असते-

- a) स्थिर रँम (Static RAM)
- b) डायनॅमिक रँम (Dynamic RAM)

#### a) स्टॅटिक रँम (SRAM):

स्टॅटिक रँममध्ये जोपर्यंत पॉवर पुरवली जात आहे तोपर्यंत मेमरी त्यातील कन्टेन्ट राखून ठेवते. अस्थिर नेचरमुळे पॉवर कमी झाल्यावर डेटा गमावला जातो. SRAM चिप्स मध्ये कॅप्यासिटर नसून 6 -ट्रान्झिस्टरचे मॅट्रिक्स वापरतात. ट्रान्झिस्टर मधील लिकेज रोखण्यासाठी पॉवरची आवश्यकता नसते, SRAM ला नियमितपणे रीफ्रेश करण्याची आवश्यकता नसते. मॅट्रिक्समध्ये अतिरिक्त जागा आहे, SRAM समान प्रमाणात स्टोरेज स्पेससाठी DRAM पेक्षा

अधिक चिप्स वापरते, ज्यामुळे उत्पादन खर्च जास्त होतो .SRAM कॅशे मेमरी म्हणून वापरली जाते आणि खूप जलद ॲक्सेस आहे.

#### **स्टॅटिक RAM ची वैशिष्ट्ये:**

- 1.लॉन्ग लाइफ
- 2.रिफ्रेश करण्याची गरज नाही
- 3.जलद
- 4.कॅशे मेमरी म्हणून वापरली जाते.
- 5.मोठा आकार
- 6.महाग
- 7.जास्त ऊर्जा वापर होतो (high power consumption)

#### **Dynamic RAM (DRAM):**

DRAM ला डेटा राखण्यासाठी (hold) सतत रिफ्रेश करणे आवश्यक आहे हे रिफ्रेश सर्किटवर . मेमरी ठेवून केले जाते जे प्रति सेकंद अनेक वेळा डेटा पुन्हा लिहिते DRAM बहुतेक सिस्टीम मेमरी साठी वापरली जाते कारण ती स्वस्त आणि लहान आहे सर्व .DRAM मेमरी सेलनी बनलेले असतात, जे एक कॅपेसिटर आणि एक ट्रान्झिस्टर बनलेले असतात.

#### **Dynamic RAM ची वैशिष्ट्ये**

1. याची लाईफ शॉर्ट असते.
2. सतत रिफ्रेश करणे आवश्यक आहे
3. SRAM च्या तुलनेत स्लो आहे.
5. आकाराने लहान
6. कमी खर्चिक
7. वीज वापर कमी होतो.

#### **Read only Memory (ROM):**

ROM ही रीड ओन्ली मेमरी.या मेमरीतून आपण फक्त वाचू शकतो लिहू शकत नाही.या प्रकारची मेमरी अस्थिर (non-volatile) असते.यात माहिती कायमस्वरूपी साठवली जाते.संगणक सुरू करण्यासाठी (boot करणे) आवश्यक अशा सूचना रॉम साठवतो.या ऑपरेशनला बूटस्ट्रॅप म्हणतात. रॉम चिप्स केवळ संगणकातच नाही तर वॉशिंग मशीन आणि मायक्रोवेव्ह ओव्हनसारख्या इतर इलेक्ट्रॉनिक वस्तूंमध्येदेखील वापरल्या जातात.



**आकृती 5.25- रीड ओन्ली मेमरी (ROM)**



रॉमचे विविध प्रकार आणि त्यांची वैशिष्ट्ये.

#### 1.MROM (मास्कडरॉम):

अगदी पहिले रॉम हार्ड-वायर्ड उपकरणे होते ज्यात डेटा किंवा सूचनांचा पूर्व-प्रोग्राम केलेला संच होता.या प्रकारचे रॉम मास्क केलेले रॉम म्हणून ओळखले जातात, जे स्वस्त आहेत.

#### 2.PROM (Programmable ROM):

PROM ही केवळ वाचनीय मेमरी आहे. युजर फक्त एकदाच मॉडिफाय करू शकतो. युजर रिक्त (blank) PROM खरेदी करतो आणि PROM प्रोग्राम वापरून इच्छित (desired) कन्टेन्ट स्टोर(store) करतो. PROM चिपच्या आत, लहान फ्यूज असतात जे प्रोग्रामिंग दरम्यान बर्न होतात. हे फक्त एकदाच प्रोग्राम केले जाऊ शकते आणि इरेज इरेज येत नाही.

#### 3.EPROM (Erasable and Programmable Read Only Memory)

EPROM काही मिनिटांपर्यंत अल्ट्रा-व्हायोलेट प्रकाशाच्या संपर्कात आणून इरेस(delete) केले जाऊ शकते. EPROM इरेजर हे करते. प्रोग्रामिंग करताना, IC मध्ये विद्युत चार्ज ट्रॅप केलेला असतो. हा चार्ज 10 वर्षांहून अधिक काळ टिकवून ठेवलेला असतो कारण चार्जमध्ये लिकेज होत नाही. हा चार्ज इरेज करण्यासाठी, अल्ट्रा-व्हायोलेट लाईट क्वार्ट्ज क्रिस्टल विंडोमधून पाठवली जाते. संपर्कात येण्यामुळे चार्ज नष्टहोतो त्यामुळे वापरताना क्वार्ट्जचे लिड स्टिकरने सील केले जाते.

#### 4.EEPROM (इलेक्ट्रीकली इरेसबल आणि प्रोग्रामेबल रीड ओन्ली मेमरी)

EEPROM प्रोग्राम केलेले आणि इलेक्ट्रिकली इरेस केले जाते. हे सुमारे दहा हजार वेळा बंद केले जाते आणि पुन्हा प्रोग्राम केले जाऊ शकते. इरेज करणे आणि प्रोग्रामिंग दोन्ही सुमारे 4 ते 10ms घेतात. EEPROM मध्ये, कोणतेही लोकेशन निवडकपणे इरेस आणि प्रोग्राम केले जाऊ शकते. संपूर्ण चिप इरेस करण्याऐवजी EEPROMs एकावेळी एक बाइट इरेस करते. याची री-प्रोग्रामिंगची प्रोसेस फ्लेक्सिबल परंतु स्लो आहे.

ROM चे फायदे:

1. अस्थिर (non-volatile)
2. रॅमपेक्षा स्वस्त
3. चाचणी करणे सोपे
4. रॅमपेक्षा अधिक विश्वासार्ह (reliable)
5. स्थिर आणि रीफ्रेश करण्याची आवश्यकता नाही.
6. कन्टेन्ट नेहमी व्हेरिफाय केले जाऊ शकतात.

रॅम आणि ROM मधील मुख्य फरक-

अनु. क्र.	रॅम (RAM)	रॉम (ROM)
1	अस्थिर	स्थिर

2	कन्टेन्ट तात्पुरते आहेत. वीज पुरवठा बंद झाल्यावर डेटा गमावला जातो.	कन्टेन्ट कायम आहे. वीज बंद असतानाही डेटा गमावला जात नाही.
3	लहान स्टोरेज क्षमतेत उपलब्ध.	जास्त स्टोरेज क्षमतेत उपलब्ध.
4	प्रोससिंग गती जास्त आहे.	प्रोससिंग गती कमी आहे.
5	युसर-डीफाइनड प्रोग्राम साठवते.	ऑपरेटिंग सिस्टमला सपोर्ट करणारे प्रोग्राम साठवते.
6	खर्च खूप जास्त आहे.	कमी खर्च
7	हे दोन प्रकारचे आहे, SRAM आणि DRAM.	हे विविध प्रकारात येते जसे की PROM, EPROM, EEPROM आणि फ्लॅश मेमरी

### SRAM आणि DRAM – मधील प्रमुख फरक

अनु. क्र.	SRAM	DRAM
1	ट्रान्झिस्टरची संख्या अधिक असते ट्रान्झिस्टर	सिंगल (1) ट्रान्झिस्टर वापरला जातो.
2	रीफ्रेश सर्किटरी आवश्यक नाही.	त्यामुळे रिफ्रेश सर्किटरीची गरज असते.
3	खूपच वेगवान	तुलनेने हळू
4	उर्जा कमी वापरतो	जास्त उर्जा वापरतो
5	कमी जागा व्यापते.	जास्त जागा व्यापते.
6	महाग	स्वस्त
7	कॅशे मेमरी म्हणून वापरली जाते.	मुख्य मेमरी म्हणून वापरली जाते.

### 2.दुय्यम मेमरी(Secondary memory)

याप्रकारच्या मेमरीला बाह्य मेमरी किंवा नॉन-व्होलाटाइल असेही म्हणतात .हे मुख्यमेमरीपेक्षाहळू आहे .ही मेमरी डेटा/माहिती कायमस्वरूपी(permanent) साठवण्यासाठी वापरली जाते.CPU चा डेटा मेमरी मध्ये थेट प्रवेश करत नाही, त्याऐवजी ते इनपुट-आउटपुटद्वारे ऍक्सेस होतात .दुय्यम (secondary) मेमरीचे कन्टेन्ट प्रथम मुख्य मेमरीमध्ये ट्रान्सफर केले

जातात आणि नंतर CPU ते access करू शकतो. उदाहरणार्थ, डिस्क, सीडी-रॉम, डीव्हीडी



### आकृती 5.26 - दुय्यम मेमरी

#### 2.दुय्यम मेमरीची (secondary memory) वैशिष्ट्ये:

- 1.या चुंबकीय (मॅग्नेटिक) आणि ऑप्टिकल मेमरी आहेत.
- 2.ती बॅकअप मेमरी म्हणून ओळखली जाते.
- 3.हीएकनॉन-अस्थिर (non-volatile) मेमरीआहे.
- 4.वीज बंद असली तरीही डेटा कायमचा साठवला जातो.
- 5.याचा वापर संगणकात डेटा साठवण्यासाठी केला जातो.
- 6.संगणक दुय्यम मेमरी शिवाय चालू शकतो.
- 7.प्राथमिक मेमरी पेक्षा हळू आहे.

#### 5.3.1 प्रोग्रामेबल लॉजिक डिव्हाइसेस (PLD):

कोणत्याही इलेक्ट्रॉनिक उपकरणामध्ये 3 गोष्टी महत्वाची भूमिका बजावतात :मेमरी, मायक्रोप्रोसेसर आणि लॉजिक डिव्हाइसेस

1. मेमरी random माहिती साठवतात.
2. मायक्रोप्रोसेसर हे सॉफ्टवेअर सूचना एक्सिक्युट (execute) करतात ज्याच्या मध्ये वर्डप्रोसेसिंग प्रोग्राम किंवा व्हिडीओ गेम चालवण्यासारखी विविध कार्ये केली जातात.
3. लॉजिक devices विशिष्ट कार्ये करतात, ज्यामध्ये डिव्हाइस-टू-डिव्हाइस इंटरफेसिंग, डेटा कम्युनिकेशन, सिग्नल प्रोसेसिंग, डेटा डिस्प्ले, टाइम आणि कंट्रोल ऑपरेशन्स आणि सिस्टमने पूर्ण करणे आवश्यक असलेले प्रत्येक कार्य करते.

#### प्रोग्रामेबल लॉजिक डिव्हाइसेसचे उपयोग:

- 1.उच्च कार्यक्षमता कॉम्प्युटिंग
- 2.नेटवर्क प्रोसेसिंग
- 3.मोठा (large) डेटा प्रोसेसिंग
- 4.जीनोमिक्स (medical field application)

#### Programmable Logic:

हा एक लॉजिक घटक आहे ज्याचे कार्य विशिष्ट कार्यासाठी मर्यादित नाही .हे लाईफ सायकलच्या वेगवेगळ्या पॉईंट्सवर प्रोग्राम करू शकते.हे सेमीकंडक्टर व्हेंडर (vendor) द्वारे



standard सेल, गेट अरे, असेंबलीपूर्वी डिझाइनरद्वारे किंवा सर्किटमध्ये युजरद्वारे प्रोग्राम केले जाते.

### **Programmable Logic Devices (PLDs):**

PLD ही सेमीकंडक्टर उपकरणे आहेत जी आवश्यक लॉजिक डिव्हाइस मिळविण्यासाठी प्रोग्राम केली जाऊ शकतात. प्रोग्रामेबिलिटी मुळे, अनेक सेमीकंडक्टर ऑप्लिकेशन्स मध्ये लॉजिक गेट्स, फ्लिप-फ्लॉप, काउंटर आणि मल्टीप्लेक्सर्स सारख्या विशेष (special) लॉजिक उपकरणांची जागा घेतली आहे. यामुळे डिझाईनचा वेळ कमी होतो आणि त्यामुळे उत्पादन बाजारात पोहोचण्यासाठी वेळ कमी होतो. त्यामध्ये AND आणि OR गेट्सच्या अरे (array) असतात, ज्याला आवश्यक लॉजिक फंक्शन समजण्यासाठी प्रोग्राम केले जाऊ शकते.

या प्रोग्राम कोडमध्ये डिव्हाइससाठीच्या सूचनेचा समावेश आहे.

1. हा प्रोग्राम आवश्यकतेनुसार बदलले, एडिट किंवा रिप्लेस केला जाऊ शकतो.
2. PLDs या IC's मध्ये कोड वापरून बनवले जातात.
3. हार्डवेअर प्रोग्रामिंग/डिस्क्रिप्शन language वापरून PLDs साठी कोड लिहितात.
4. प्रत्येक गेट ऑपरेशन नियंत्रित करण्यासाठी डिव्हाइस प्रोग्रामर PLD वर फ्यूज ब्लो होते.
5. स्वस्त सॉफ्टवेअर साधने, quick development, सिमुलेशन आणि चाचणीसाठी वापरली जातात.

### **1. फिक्स्ड लॉजिक सिस्टिम (Fixed Logic System):**

लॉजिक सिस्टिममध्ये सर्किट्स असतात ज्यांचे कॉन्फिगरेशन प्रीसेट किंवा कायम असते. त्यांच्या सूचना repeatedly ऑपरेशन्सचा एक fixed संच करतात. एकदा उत्पादित आणि प्रोग्राम केल्यानंतर, लॉजिक बदलता येत नाही. फिक्स्ड लॉजिक युजरना त्यांच्या एक्सिस्टिंग कार्यक्षमतेचा विस्तार किंवा बिल्ड करण्याची परवानगी देत नाही.

### **2. प्रोग्रामेबल लॉजिक सिस्टिम (Programmable logic system):**

हा प्रोग्राम करण्यास सोपा, परवडणारा आणि चांगल्या वैशिष्ट्यांसह सुसज्ज आहे. डिझाइन विकसित करण्यासाठी, कोड करण्यासाठी आणि चाचणी करण्यासाठी स्वस्त सॉफ्टवेअर वापरले जाते. हे डिझाइननंतर डिव्हाइसमध्ये प्रोग्राम केले जाते आणि डायरेक्ट इलेक्ट्रॉनिक सर्किटमध्ये तपासले जाते.

**प्रोग्रामेबल लॉजिक लिहिण्यासाठी वापरल्या जाणाऱ्या भाषा:**

#### **PALASM:**

सुरवातीला हार्डवेअर डिस्क्रिप्शन भाषा (HDL), PALASM बुलियन लॉजिक फंक्शन्स आणि स्टेटट्रांझिशन टेबल्सचे फ्यूजमॅपमध्ये ट्रान्सलेट करते. PALASM हे प्रोग्रामेबल अरे लॉजिक(PAL) उपकरणांमध्ये वापरले जाते.

#### **b. ABEL**

Advanced Boolean Expression Language (ABEL) ही PLDs साठी वापरली जाणारी दुसरी प्रोग्रामिंग भाषा आहे. हे सिक्वेंशियल स्टेट्स मशीन, ट्रूथ टेबलस (truth tables) आणि कॉन्करंट समीकरणांशी (concurrent equations) संबंधित आहे.

#### c. CUPL

C भाषेत लिहिलेले, युनिव्हर्सल प्रोग्रामेबल लॉजिकसाठी कंपाइलर प्रोग्रामेबल लॉजिक उपकरणांसाठी वापरल्या जाणाऱ्या जुन्या भाषांपैकी एक आहे.

#### d. Verilog

व्हेरिलॉग ही ऍबस्ट्रक्शन (abstraction) च्या रेजिस्टर-ट्रान्सफर स्तरा (level) वर डिजिटल सर्किट्ससाठी सामान्यतः वापरली जाणारी सुरवातीची भाषा आहे. ही हार्डवेअर वर्णन भाषा ॲनालॉग आणि मिक्सड-सिग्नल सर्किट्ससाठी तसेच इंटिग्रेटेड सर्किट्सच्या फॅब्रिकेशनसाठी देखील लागू केली जाऊ शकते.

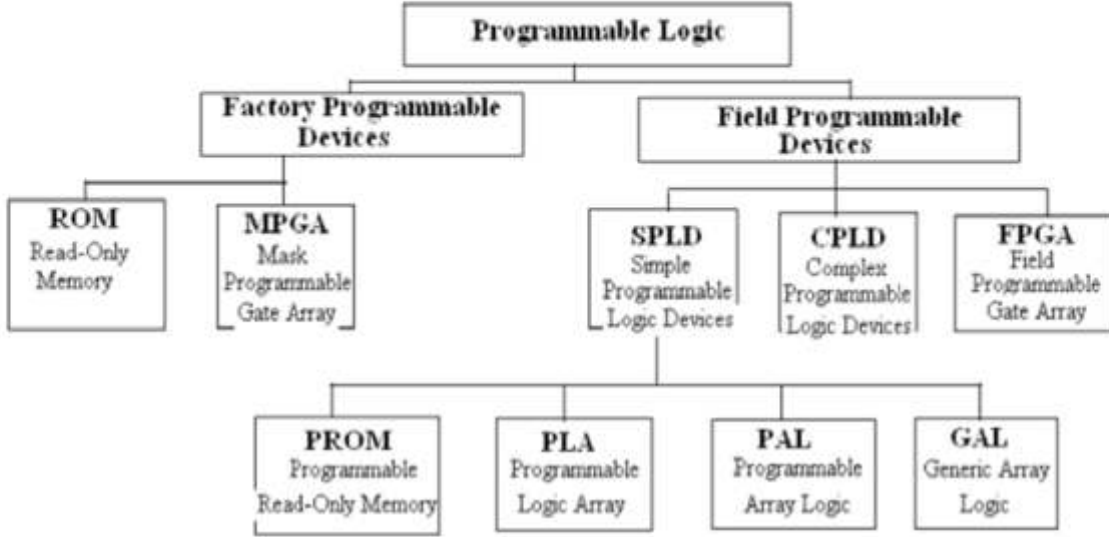
#### e. VHDL

VHDL म्हणजे VHSIC-HDL. व्हेरी हाय-स्पीड इंटिग्रेटेड सर्किट हार्डवेअर वर्णन भाषा. ही हार्डवेअर वर्णन करणारी भाषा आहे जी ॲनालॉग आणि मिश्र-सिग्नल सिस्टमशी संबंधित प्रोग्रामिंगसाठी वापरली जाते.

**प्रोग्रामेबल लॉजिक डिव्हाइसेसचे फायदे:**

- डिझाइन फ्लेक्सिबिलिटी** : PLDs ग्राहकांना डिझाइन सायकल दरम्यान अधिक फ्लेक्सिबिलिटी देतात कारण डिझाइन iterations ही फक्त प्रोग्रामिंग फाइल बदलण्याची बाब आहे आणि डिझाइन बदलांचे परिणाम (results) कार्यरत पार्ट्समध्ये त्वरित दिसू शकतात.
- सुधारित विश्वासार्हता (Improved Reliability)** कमी उर्जा आणि कमी इंटरकनेक्शन आणि पॅकेजेस मोठ्या प्रमाणात सुधारित सिस्टम रिलयायबिलिटीमध्ये ट्रान्सलेट होतात.
- लोपॉवर** : वीज वापर कमी करण्यासाठी CMOS आणि कमी पॅकेजेस एकत्र होतात.
- कमी गुंतागुंतीचे (less complex)**: PLDs कमी उर्जा वापरत असल्याने कमी जागा लागते.
- पीएलडी फील्ड**- प्रोग्राम करण्यायोग्य आहेत मॅनुफॅक्चूरिंग ऍम्बिअरॉनमेंट (manufacturing environment) च्या बाहेर प्रोग्राम केले जाऊ शकते.
- PLDs इरेस आणि री-प्रोग्राम करण्यायोग्य आहेत डिव्हाइस अपडेट करण्यास किंवा error सुधारण्यासाठी परवानगी देते आणि वेगळ्या डिझाइनसाठी डिव्हाइसचा पुनर्वापर करण्यास परवानगी देते.

**PLDs चे वर्गीकरण डायग्राम 5.27 दाखवले आहे:**



आकृती 5.27 PLDs चे वर्गीकरण

पीएलडीएसचे दोन प्रकारांमध्ये वर्गीकरण केले जाते

1. फॅक्टरी प्रोग्राम करण्या योग्य उपकरणे

2. फील्ड प्रोग्राम करण्या योग्य उपकरणे

1. फॅक्टरी प्रोग्राम करण्या योग्य उपकरणे

i) फॅक्टरी प्रोग्राम करण्या योग्य उपकरणे प्रोग्राम करण्या योग्य लॉजिक डिव्हाइसचा एक प्रकार आहे.

ii) जेव्हा एखादे उपकरण बनवतो, तेव्हा ते काही वैशिष्ट्यांनुसार बनवतो.

iii) फॅक्टरीमध्ये फॅक्टरी प्रोग्राम करण्यायोग्य उपकरणे तयार केल्यानंतर, ते डेटाशीट नुसार पॅरामीटर्सशी जुळतात याची पडताळणी करण्यासाठी त्यांची चाचणी केली जाते.

iv) या उपकरणांमध्ये विस्तृत (wide) वारंवारता श्रेणी आहे. जास्त (high) तापमानात, प्रोग्राम केल्या नंतर स्थिरतेसाठी त्यांची चाचणी केली जाते.

फॅक्टरी प्रोग्रामेबल डिव्हाइसेसचे दोन प्रकार आहेत

1. MROM

2. MPGA

1. MROM (Mask ROM)

मास्क केलेला ROM एक प्रकारची read only स्टोरेज आहे ज्याची अंतर्गत माहिती मॅनुफॅक्चरर स्वतः प्रोग्राम केलेली आहे. या उपकरणामध्ये एक सॉफ्टवेअर मास्क आहे जो डिझायनिंग स्टेज दरम्यान चिपमध्ये बर्न केला जातो. या चिप्सची मॅनुफॅक्चरिंग प्रोसेस प्रत्यक्षात सुरू होण्यापूर्वी ट्रान्झिस्टरची arrangement केली जाते. लॉन्ग टर्म sustainability आवश्यक असलेल्या सिस्टममध्ये, MROM चा वापर केला जातो. रॉमच्या डिझायनिंग दरम्यान ICs सुरुवातीला 'मास्क' केले जातात.



## 2. MPGA

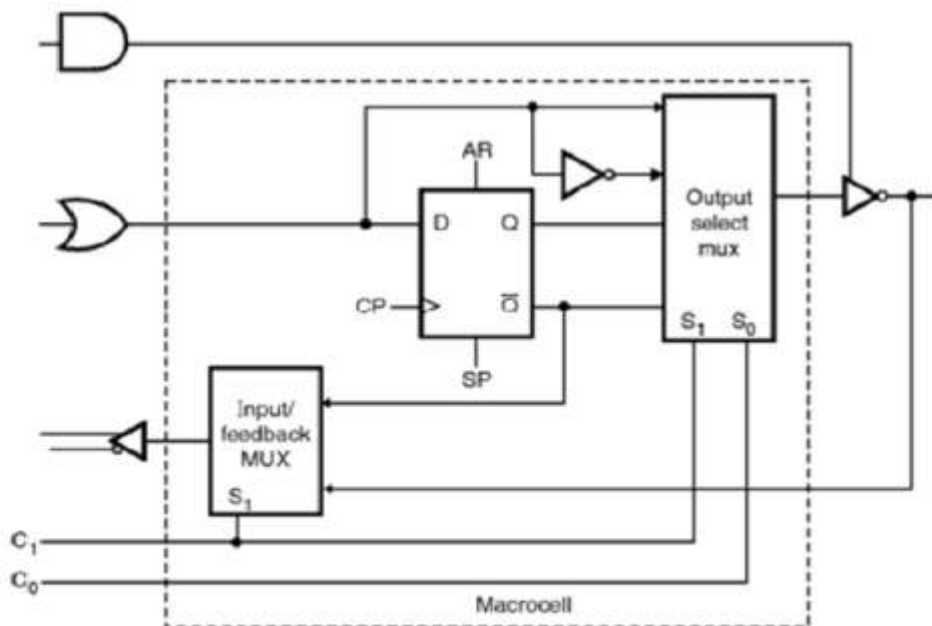
मास्क प्रोग्रामेबल गेट अरे विस्तृत (extensive) लॉजिक सर्किटरी हाताळते. MPGAs मध्ये प्री-फॅब्रिकेटेड ट्रान्झिस्टरचे अरे असतात .जे नंतर user defined लॉजिक सर्किट्समध्ये कस्टमाईज्ड केले जाऊ शकतात.हे कस्टमायझेशन सिस्टम डिझाइननुसार ट्रान्झिस्टर एकमेकांशी जोडून केले जाते. कोणतेही इच्छित(desired) लॉजिक सर्किट लागू करण्यासाठी ट्रान्झिस्टरच्या row एकमेकांशी जोडल्या जातात. हे कनेक्शन रो(row)मध्ये किंवा रो(row) मधील असू शकतात. चिप फॅब्रिकेशन दरम्यान मेटल इंटरकनेक्शन प्रोव्हाईड करून कस्टमायझेशन केले जाते.वेगवेगळ्या गरजांसाठी वेगवेगळे inter connection आवश्यक आहे.फील्ड-प्रोग्राम करण्या योग्य उपकरणे

- फील्ड प्रोग्राम करण्या योग्य उपकरणांची प्रोग्रामिंग नंतर चाचणी केली जाऊ शकत नाही.ii) वारंवारता श्रेणी (frequency range) मर्यादित आहे.
- प्रोग्रामिंग qualified प्रोग्रामर वापरून केले जाऊ शकते.
- सर्व प्रोग्रामेबल लॉजिक डिव्हाइसेसना सामान्यतः फील्ड प्रोग्रामेबल डिव्हाइसेस म्हणतात.
- CPLDs EEPROM तंत्रज्ञानावर आधारित आहेत.

## 1. SPLD

जेथे फक्त थोड्याच I/Os ची आवश्यकता असते तेथे एक साधे प्रोग्रामेबल लॉजिक डिव्हाइस म्हणून वापरले जाते. यात फक्त एक डझन किंवा त्यापेक्षा जास्त मॅक्रोसेल असतात. SPLDs हे सर्वात सरळ, स्वस्त, सर्वात लहान आणि कमीत कमी उर्जा वापरणारे फील्ड प्रोग्राम करण्या योग्य उपकरणे आहेत.

PALs, PLAs आणि GAL सारखे PLDs हे simple PLDs आहेत.

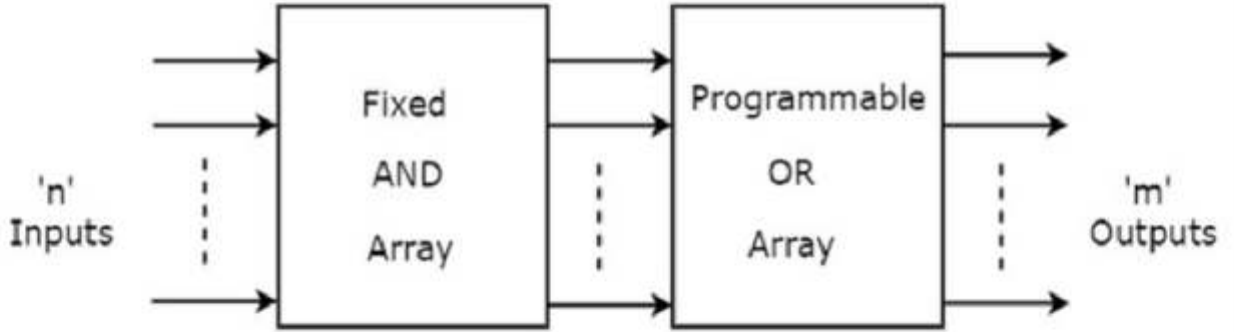


आकृती 5.28 सिम्पल प्रोग्रामेबल लॉजिक डिव्हाइसेस आर्किटेक्चर

## सिम्पल प्रोग्रामेबल लॉजिक डिवाइसेस आर्किटेक्चर

### a. PROM (Programmable ROM)

प्रोग्राम करण्यायोग्य ROM हे एक मेमरी उपकरण आहे जे कायमस्वरूपी digital माहिती साठवते. हे सामान्य रॉमपेक्षा वेगळे आहे कारण ते एकदा PROM प्रोग्रामर वापरून इलेक्ट्रिकली प्रोग्राम केले जाऊ शकते.



आकृती 5.29 PROM ची ब्लॉक डायग्राम

•AND अ‍ॅरेचे इनपुट प्रोग्राम करण्या योग्य नाहीत. हा डीकोडर 'n' min-terms उत्पन्न करतो. OR गेट्स इनपुट प्रोग्राम करण्या योग्य आहेत. AND अ‍ॅरेचे आउटपुट हे या OR अ‍ॅरे सिस्टीमचे इनपुट आहेत. PROM चे आउटपुट min-terms च्या बेरजेइतके असते.

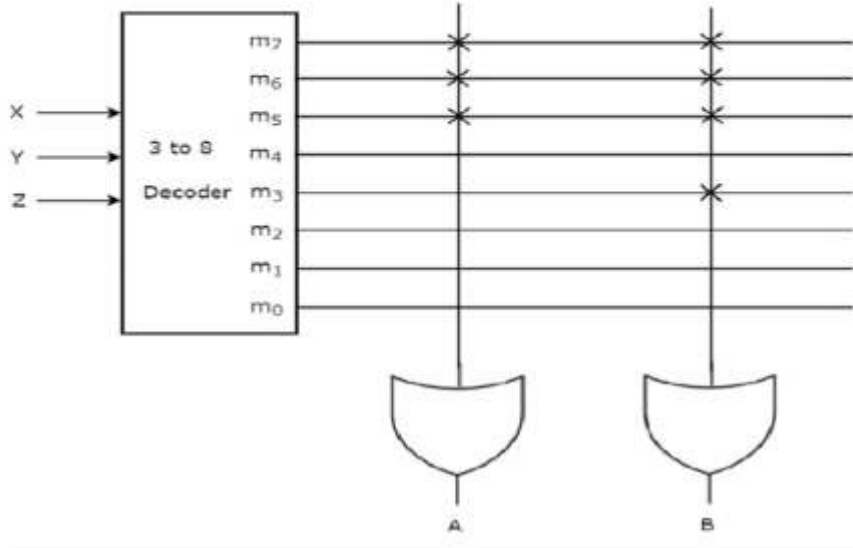
#### उदाहरण

PROM वापरून खालील बुलियन फंक्शन्स लिहू.

$$A(X,Y,Z)=\sum m(5,6,7)$$

$$B(X,Y,Z)=\sum m(3,5,6,7)$$

दिलेली दोन फंक्शन्स min-terms फॉर्म मध्ये आहेत आणि प्रत्येक फंक्शनमध्ये तीन व्हेरिएबल्स X, Y आणि Z आहेत. दोन फंक्शन्स तयार करण्यासाठी 3:8 डीकोडर आणि दोन प्रोग्रामेबल OR गेट्स आवश्यक आहेत. संबंधित PROM आकृती 5.30 मध्ये दर्शविले आहे.

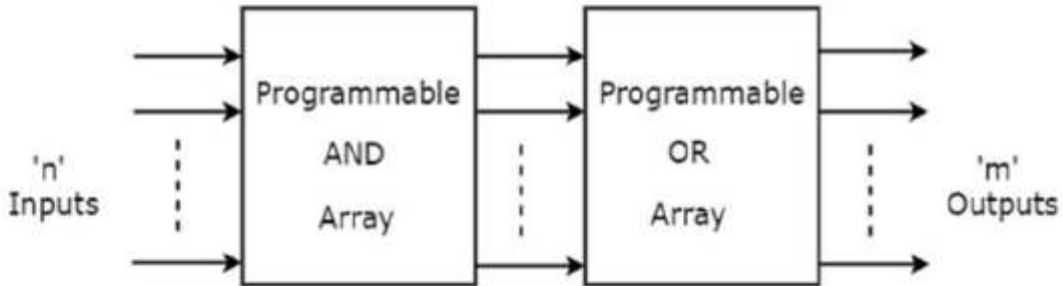


आकृती 5.30 3:8 डीकोडर

3 ते 8 डीकोडर 8 min-terms उत्पन्न करतात. प्रत्येक OR गेटद्वारे संबंधित बुलियन फंक्शन्स तयार करण्यासाठी फक्त आवश्यक किमान अटी प्रोग्राम केल्या जातात. 'X' हे चिन्ह प्रोग्रामेबल कनेक्शनसाठी वापरले जाते.

#### b.PAL (Programmable Logic Array)

Programmable Logic Array एक PLD आहे ज्यामध्ये AND आणि OR अॅरेचे दोन्ही parts प्रोग्राम करण्या योग्य आहेत, प्रोग्रामेबल AND अॅरे आणि प्रोग्रामेबल OR अॅरे आहेत. या विभागांना AND-plane आणि OR-plane म्हणतात.



आकृती 5.31 PALची ब्लॉक डायग्राम

AND गेट्सचे इनपुट प्रोग्राम करण्या योग्य आहेत. OR गेट्सचे इनपुट देखील प्रोग्राम करण्या योग्य आहेत. PAL चे आउटपुट उत्पादनाच्या अटींची बेरजेइतके असेल. PALचा वापर कॉम्बिनेशनल आणि सिक्वेन्शियल लॉजिक सर्किट्सच्या अंमलबजावणीसाठी केला जाऊ शकतो.

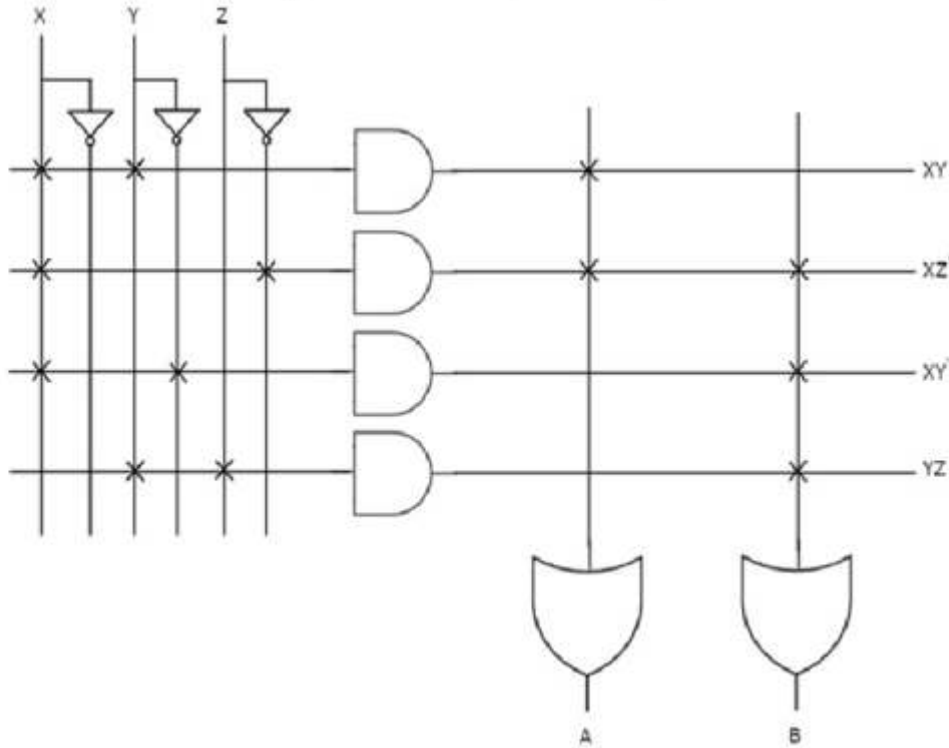
#### उदाहरण

PLA वापरून खालील बुलियन फंक्शन्स लिहू,

$$A=XY+XZ'$$

$$B=XY'+YZ+XZ'$$

दिलेली दोन फंक्शन्स उत्पादनांच्या बेरीजमध्ये आहेत. दिलेल्या बुलियन फंक्शन्स A आणि B मध्ये उपस्थित असलेल्या उत्पादन पदांची संख्या अनुक्रमे दोन आणि तीन आहेत. एक उत्पादन संज्ञा,  $Z'X$  प्रत्येक फंक्शनमध्ये सामान्य आहे. त्यामुळे, त्या दोन फंक्शन्सच्या मॅन्युफॅक्चरिंगसाठी चार प्रोग्रामेबल AND गेट्स आणि दोन प्रोग्रामेबल OR गेट्स आवश्यक आहेत. PLA सर्किट डायग्राम आकृती 5.32 मध्ये दर्शविले आहे.

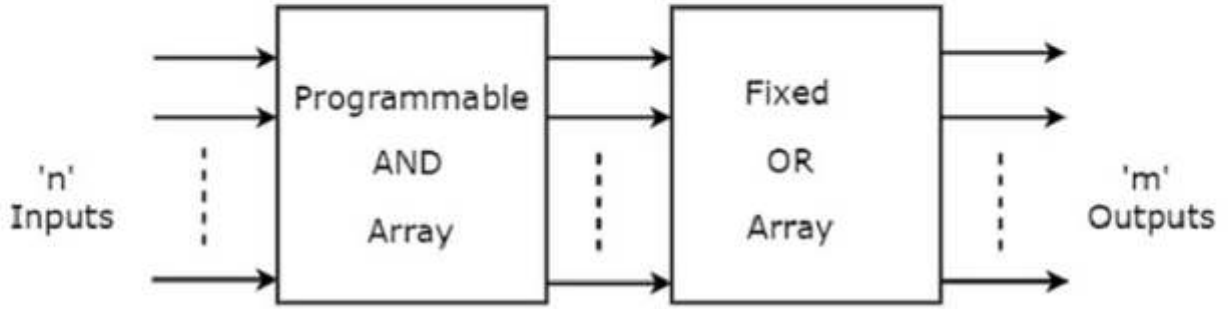


आकृती 5.32 PAL सर्किट डायग्राम

प्रोग्रामेबल AND गेट्समध्ये व्हेरिएबल्सच्या सामान्य (normal) आणि पूरक (complemented) दोन्ही इनपुटचा ऍक्सेस आहे. आकृती 5.32 मध्ये, X, X', Y, Y', Z आणि Z', इनपुट प्रत्येक AND गेटच्या इनपुटवर उपलब्ध आहेत. प्रत्येक AND गेटद्वारे एक गुणात्मक टर्म उत्पन्न करण्यासाठी फक्त आवश्यक variable प्रोग्राम केले जाते. या सर्व गुणात्मक संज्ञा प्रत्येक प्रोग्राम करण्या योग्य OR गेटच्या इनपुटवर उपलब्ध आहेत. प्रत्येक OR गेटद्वारे संबंधित बुलियन फंक्शन्स तयार करण्यासाठी केवळ आवश्यक गुणात्मक अटींचा प्रोग्राम केले जाते. 'X' हे चिन्ह प्रोग्रामेबल कनेक्शनसाठी वापरले जाते.



### c) PAL (प्रोग्राम करण्यायोग्य अँरे लॉजिक)



आकृती 5.33 PAL ब्लॉक डायग्राम

AND गेट्सचे इनपुट येथे प्रोग्राम करण्या योग्य आहेत. सर्व संभाव्य minterms उत्पन्न करण्याऐवजी (प्रोग्राम करण्या योग्य AND गेट्स वापरून) फक्त आवश्यक गुणात्मक अटी उत्पन्न करू शकतो. OR गेट्सचे इनपुट प्रोग्राम करण्या योग्य प्रकारचे नाहीत. प्रत्येक OR गेट्सच्या इनपुटची संख्या निश्चित प्रकारची आहे. PAL चे आउटपुट गुणात्माकांच्या बेरजेच्या स्वरूपात असेल. PALs च्या तुलनेत PAL चे गुणात्माक करणे सोपे आहे आणि ते खूपच कमी खर्चिक देखील आहे.

#### उदाहरण

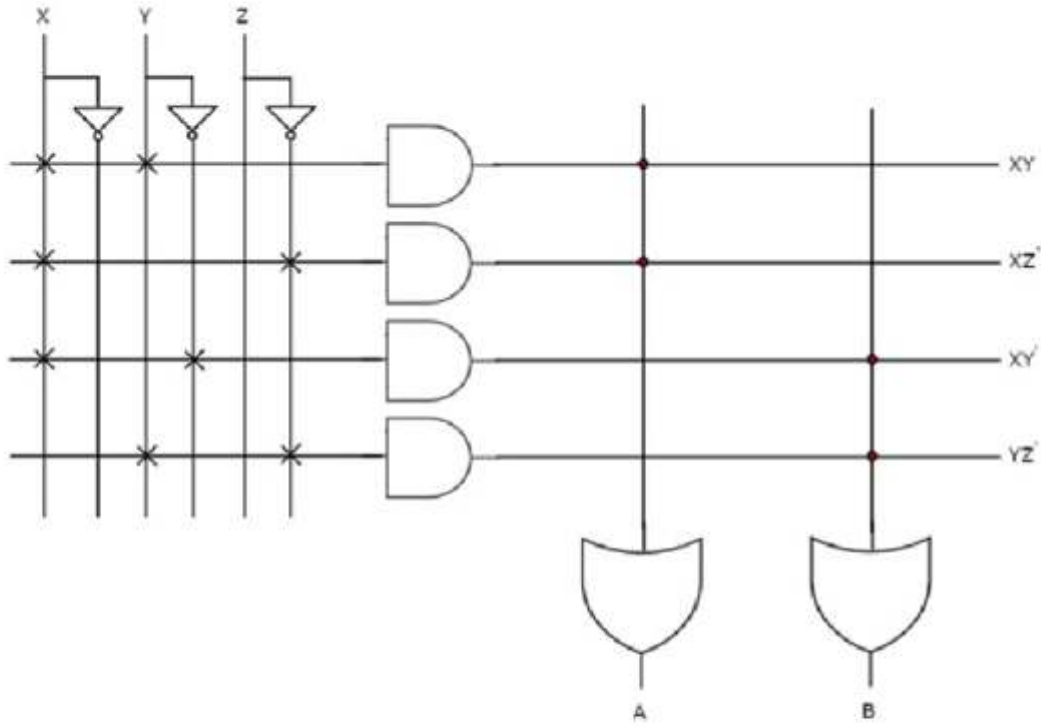
PAL वापरून खालील बुलियन फंक्शन्स अंमलात आणू.

$$A = XY + XZ'$$

$$A = XY' + YZ'$$

दिलेली दोन फंक्शन्स उत्पादनांच्या बेरीजमध्ये आहेत. प्रत्येक बुलियन फंक्शन मध्ये दोन गुणात्माक संज्ञा आहेत. तर, त्या दोन फंक्शन्सच्या निर्मितीसाठी चार प्रोग्रामेबल AND गेट्स आणि दोन फिक्स्ड OR गेट्स आवश्यक आहेत. संबंधित PAL आकृती 5.34 मध्ये दर्शविली आहे.



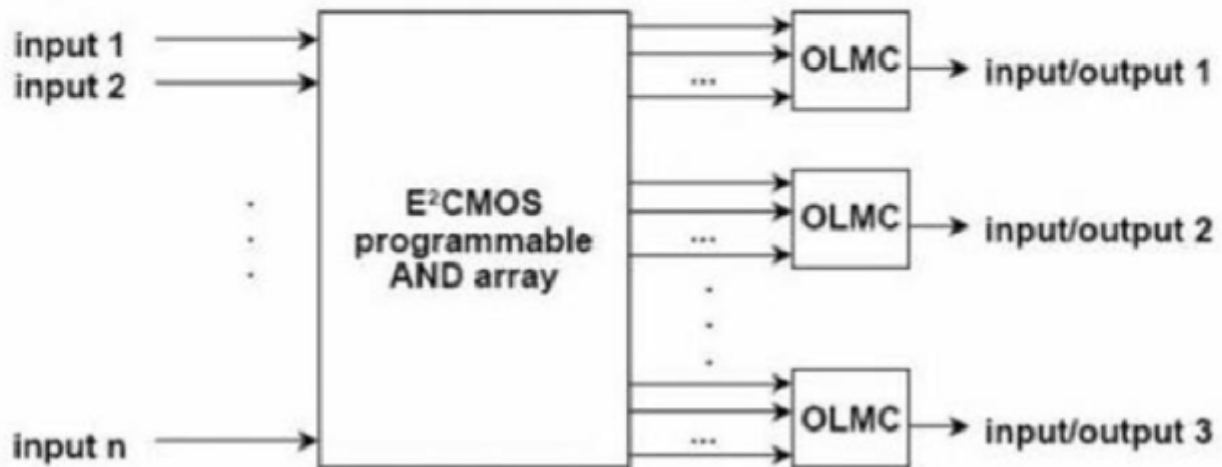


आकृती 5.34 PAL सर्किट डायग्राम

प्रोग्रामेबल AND गेट्स मध्ये व्हेरिएबल्सच्या सामान्य आणि पूरक दोन्ही इनपुटचा ऍक्सेस आहे. आकृती 5.34 मध्ये, X, X', Y, Y', Z आणि Z', इनपुट प्रत्येक AND गेटच्या इनपुटवर उपलब्ध आहेत. प्रत्येक AND गेटद्वारे एक गुणात्माक टर्म उत्पन्न करण्यासाठी फक्त आवश्यक अक्षरे प्रोग्राम करा. 'X' हे चिन्ह प्रोग्रामेबल कनेक्शनसाठी वापरले जाते. OR गेट्सचे इनपुट निश्चित प्रकारचे असतात. आवश्यक गुणात्माक अटी प्रत्येक OR गेटच्या इनपुटशी जोडलेल्या आहेत. जेणेकरून OR गेट्स संबंधित बुलियन फंक्शन्स तयार करतात. '.' हे चिन्ह जोडणीसाठी वापरले जाते.

#### d) GAL (जेनेरिक अ‍ॅरि लॉजिक)

हा PALची ऍडव्हान्स डेव्हलपमेंट आहे. जेनेरिक अ‍ॅरि लॉजिक मध्ये PAL सारखेच आर्किटेक्चर असते. फक्त एवढाच आहे GAL चे प्रोग्राम करण्या योग्य आणि अ‍ॅरि erase करू शकतात आणि पुन्हा प्रोग्राम केले जाऊ शकतात. GAL डिव्हाइसचे आउटपुट लॉजिक देखील री-प्रोग्राम करण्या योग्य आहे. इरेस आणि री-प्रोग्रामिंग PAL प्रोग्रामरनेच केले जाऊ शकते. प्रोग्रामेबल लॉजिक आणि आउटपुट लॉजिक मॅक्रोसेल (OLMC) यांचा समावेश असतो ज्यामध्ये OR गेट्स आणि फ्लिप-फ्लॉप्स (flip-flops) वगळले जातात.



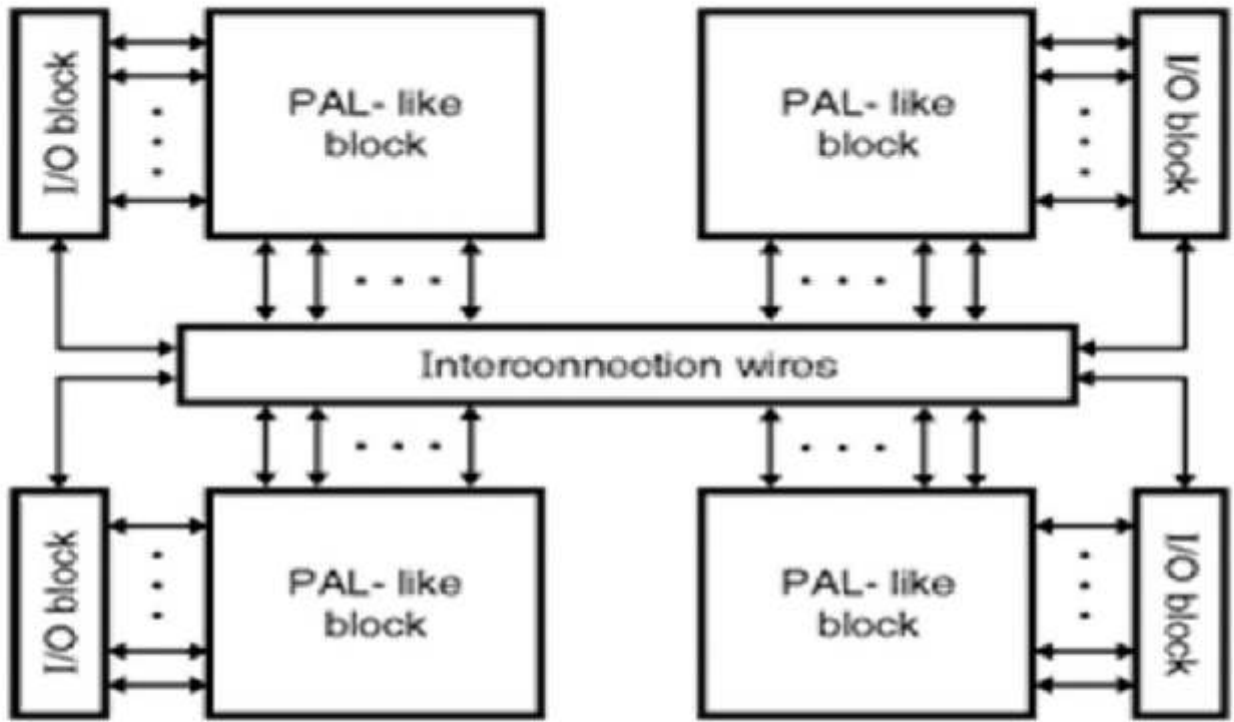
आकृती 5.35 GAL आर्कीटेक्टर

### 5.3.6 PAL आणि PLA मधील फरक

प्रोग्रामेबल अँरे लॉजिक (PAL)	प्रोग्रामेबल लॉजिक अँरे (PLA)
PAL म्हणजे प्रोग्रामेबल अँरे लॉजिक आहे	PLA म्हणजे प्रोग्रामेबल लॉजिक अँरे आहे
PAL चे कन्स्ट्रक्शन AND-OR गेट्सचे कलेक्शन वापरून केले जाऊ शकते.	PLA चे कन्स्ट्रक्शन OR गेट्सचे AND आणि फिक्स्ड कलेक्शन वापरून केले जाऊ शकते.
PAL ची उपलब्धता कमी आहे	PLA ची उपलब्धता अधिक आहे
PAL प्रोग्रामिंगची फ्लेक्सिबिलिटी अधिक आहे.	PLA ची फ्लेक्सिबिलिटी कमी आहे.
PAL ची किंमत महाग आहे.	PLA ची किंमत मध्यम श्रेणीची आहे.
PAL मधील फंक्शन्सची संख्या मोठी आहे	PLA मधील फंक्शन्सची संख्या मर्यादित आहे
PAL चा वेग कमी आहे	PLA चा वेग जास्त आहे

### 2. CPLD (Complex Programmable Logic Devices).

CPLD कडे लॉजिक गेट गुणोत्तर (ratio) जास्त इनपुट आहे. मोठ्या (larger) I/Os समाविष्ट असलेल्या उपयोगांमध्ये CPLD वापरतो, डेटा प्रोसेस तुलनेने कमी आहे. ही उपकरणे SPLDs पेक्षा डेन्सिटी जास्त आहे. त्यांची कार्यक्षमता चांगली आहे. CPLDs EPROM किंवा EEPROM तंत्रज्ञानावर आधारित आहेत. CPLD ही एकाच चिपवर अनेक SPLD सारखी ब्लॉक्सची व्यवस्था (arrangement) आहे. हे सर्किट PAL-सारखे किंवा PLA-सारखे ब्लॉक्स असू शकतात.



आकृती 5.36 सीपीएलडी आर्किटेक्चर

### 3. FPGA (फील्ड प्रोग्रामेबल गेट अरे)

फील्ड प्रोग्रामेबल गेट अरेमध्ये संपूर्ण लॉजिक सिस्टीम एका चिपवर इंटीग्रेट केली जाते. हे सिस्टीम डिझायनर्सना री-प्रोग्रामिंगसाठी उत्कृष्ट फ्लेक्सिबिलिटी देते. एक हजारपेक्षा जास्त गेट्स असलेल्या लॉजिक सर्किटमध्ये FPGAs वापरतात. सामान्य कस्टम सिस्टीम चिपच्या तुलनेत, FPGA मध्ये दहापट चांगली इंटीग्रेटेड डेन्सिटी आहे. FPGA सिस्टीमच्या आर्किटेक्चरमध्ये अनेक लॉजिक ब्लॉक्स, त्यांच्यामधील इंटर कनेक्शन्स आणि इनपुट-आउटपुट ब्लॉक्स (IOBs) असतात. या लॉजिक ब्लॉक्समध्ये एक लुकअप टेबल (LUT) असते ज्यामध्ये सिक्वेन्शियल सर्किटरी वापरली जाते. IOBs, FPGA चा वापर जलद काउंटरमध्ये केला जातो.

**FPGA** साठी प्रोग्रामिंगचे तीन मुख्य प्रकार आहेत:

- a SRAM प्रोग्रामिंग
- b अँटी-फ्यूजप्रोग्रामिंग
- c EPROM / EEPROM प्रोग्रामिंग (किंवा फ्लॅश प्रोग्रामिंग)

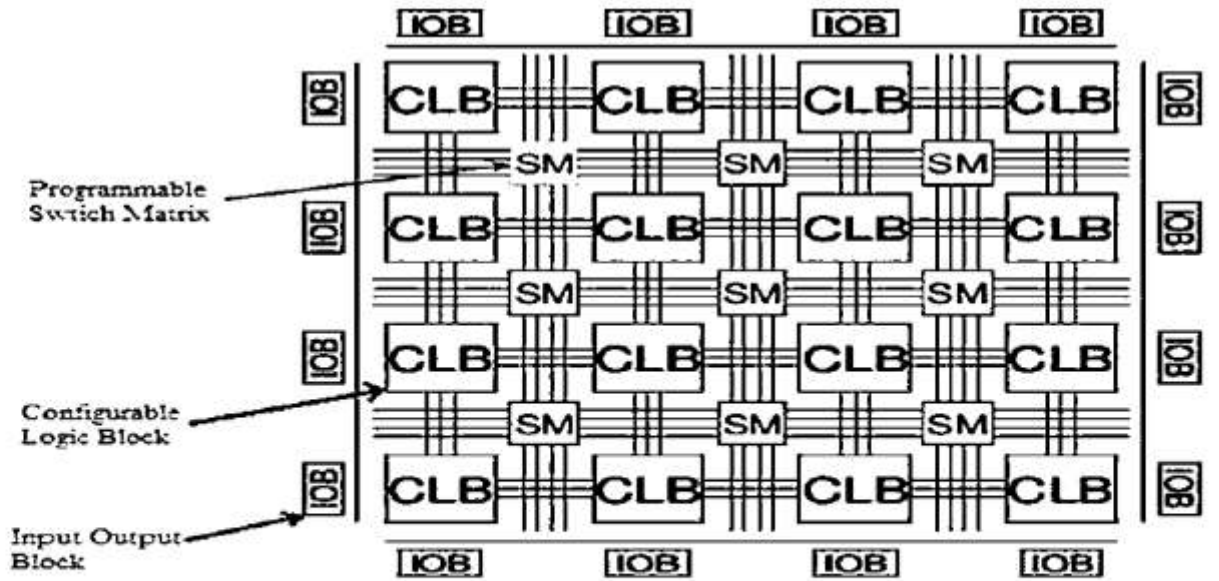
a SRAM प्रोग्रामकेलेले FPGA

SRAM प्रोग्राम केलेले FPGA लॉजिक सेल डेटा कॉन्फिगरेशन स्टॅटिक मेमरीमध्ये साठवते. पास (pass) ट्रान्झिस्टर, ट्रान्समिशन गेट्स आणि मल्टीप्लेक्सर्स वापरून परस्पर जोडलेले मार्ग तयार केले जातात. SRAM प्रोग्रामिंग मार्ग डिफाइन करण्यासाठी एक किंवा दोन-मार्गी स्विच आणि मक्स वापरते. एकेरी (one-way) NMOS ट्रान्झिस्टर वापरून वन-वे स्विचेस बनवले जाऊन या रवि-मार्गी (two-way) सिस्टीममध्ये NMOS वा PMOS वा एक



ट्रान्समिशन गेट तयार करण्यासाठी समांतर (parallel) ठेवले जाते. प्रत्येक दोन लॉजिक ब्लॉक्समध्ये, तीन कनेक्शन केले जातात.

या सिस्टिमचा एक फायदा असा आहे की तुम्हाला री-प्रोग्रामिंगसाठी FPGA सर्किटमधून बाहेर काढण्याची गरज नाही. हे प्रोग्रामिंग फक्त सिस्टिममध्ये पॉवर येईपर्यंत कार्यक्षम असते. एकदा सिस्टिमची पॉवर गमावली की मेमरी नष्ट होते. प्रोग्राम साठवण्यासाठी काही प्रकारची नॉन-व्होलॅटाइल मेमरी आवश्यक आहे. सिस्टिमला अतिरिक्त प्रोग्राम लोडर देखील आवश्यक असते.



आकृती 5.37 SRAM आधारित FPGA चे आर्किटेक्चर

प्रोग्रामिंग दोन मोडमध्ये चालते - मास्टर मोड आणि स्लेव्ह मोड

#### 1. मास्टर मोड:

FPGA बाह्य source कडून कॉन्फिगरेशन माहिती वाचते, ही बाह्य फ्लॅश मेमरी चिप आहे.

#### 2. स्लेव्ह मोड:

FPGA दुसऱ्या बाह्य मुख्य उपकरणाद्वारे कॉन्फिगर केले आहे. हे एक तर dedicated कॉन्फिगरेशन इंटरफेसद्वारे किंवा boundary-scan इंटरफेसद्वारे केले जाऊ शकते.

#### b अँटी फ्यूज प्रोग्रामिंग

अँटीफ्यूज प्रोग्राम केलेले FPGA फक्त एकदाच प्रोग्राम केले जाऊ शकतात. अँटीफ्यूज सुरुवातीला करंट conduct करत नाही परंतु करंट conduct करण्यासाठी 'बर्न' केले जाते. हे behaviour फ्यूजच्या विरुद्ध आहे आणि अशा प्रकारे म्हणतात आणि अँटीफ्यूज बर्न झालेला अँटीफ्यूज पूर्ववत करता येत नसल्यामुळे हे प्रोग्रामिंग पुन्हा केले जाऊ शकत नाही.

### c. फ्लॅश

अंतर्गत फ्लॅशसह SRAM-प्रोग्राम केलेले FPGAs SRAM सेल मधील डेटा लोड करण्यासाठी फक्त स्टार्ट-अपवर फ्लॅश मेमरी वापरतात. फ्लॅश प्रोग्राम केलेले FPGAs फ्लॅशचा वापर प्राथमिक कॉन्फिगरेशन स्टोरेज resource म्हणून करतात. ही पद्धत कमी उर्जा वापरते, ज्यामुळे ती SRAM FPGAs पेक्षा चांगली निवड बनते. ते रेडिएशन प्रभावांना देखील सहन (tolerant) करतात. हे कोणत्याही अनऑथोरिज्ड (unauthorized) बिट-स्ट्रीम कॉपी करण्याला प्रतिबंध करतात.

### स्वाध्याय:

1. DAC चे विविध प्रकार लिहा.
2. प्रोग्रामेबल लॉजिक अरेचा ब्लॉक डायग्राम काढा.
3. सक्सेसीव्ह अप्रॉक्सिमेशन टाइप (Successive Approximation type) ADC च्या कार्य तत्वाचे वर्णन करा.
4. खालील तुलना करा:
  - (i) Volatile with Non Volatile.
  - (ii) EEPROM with EPROM
5. 4 बिट वेटेड रजिस्टर प्रकारासाठी अॅनालॉग आउटपुटची खालील इनपुटसाठी DAC गणना करा
  - (i) 1011
  - (ii) 1001      गृहीत धरा  $V_{fs} = 5V$  आहे
6. C- DAC 0808 ची कोणतीही दोन वैशिष्ट्ये लिहा.
7. बिट DAC चे आउटपुट +10V आणि -10V दरम्यान बदलते. खालील गणना करा:
  - i) ठराव ii) टक्केवारी ठराव.

### लघु प्रकल्प: (Micro Circuit)

कोणत्याही अॅनालॉग इनपुट चे डिजिटल मध्ये रूपांतर करणारे सर्किट बांधा.



## HEAD OFFICE



Secretary,  
Maharashtra State Board of Technical Education  
49, Kherwadi, Bandra (East), Mumbai - 400 051  
Maharashtra (INDIA)  
Tel: (022)26471255 (5 -lines)  
Fax: 022 - 26473980  
Email: -secretary@msbte.com

**Web -[www.msbte.org.in](http://www.msbte.org.in)**

## REGIONAL OFFICES:

### MUMBAI

Deputy Secretary (T),  
Mumbai Sub-region,  
2<sup>nd</sup> Floor, Govt. Polytechnic Building,  
49, Kherwadi, Bandra (East)  
Mumbai - 400 051  
Phone: 022-26473253 / 54  
Email: rbtemumbai@msbte.com

### PUNE

Deputy Secretary (T),  
M.S. Board of Technical Education,  
Regional Office,  
412-E, Bahirat Patil Chowk,  
Shivaji Nagar, Pune  
Phone: 020-25656994 / 25660319  
Fax: 020-25656994  
Email: rbtepn@msbte.com

### NAGPUR

Deputy Secretary (T),  
M.S. Board of Technical Education  
Regional Office,  
Mangalwari Bazar, Sadar, Nagpur - 440 001  
Phone: 0712-2564836 / 2562223  
Fax: 0712-2560350  
Email: rbteng@msbte.com

### AURANGABAD

Deputy Secretary (T),  
M.S. Board of Technical Education,  
Regional Office,  
Osmanpura, Aurangabad -431 001.  
Phone: 0240-2334025 / 2331273  
Fax: 0240-2349669  
Email: rbteau@msbte.com