



महाराष्ट्र राज्य तंत्रशिक्षण मंडळ, मुंबई

(स्वायत्त) (ISO 9001:2015) (ISO/IEC 27001:2013)

अभियांत्रिकी आणि तंत्रज्ञान पदविका

शिक्षण पुस्तिका
(Learning Material)

डिजिटल तंत्रे

DIGITAL TECHNIQUES
(313303)

अणुविद्युत अभियांत्रिकी गट

K-SCHEME

(के-स्कीम)

मराठी - इंग्रजी (द्विभाषिक) माध्यम

(अभियांत्रिकी व तंत्रज्ञानातील तृतीय सत्र पदविका)

(शिक्षण पुस्तिका)

(Learning Material)

डिजिटल तंत्रे

DIGITAL TECHNIQUES

(313303)

अणुविद्युत अभियांत्रिकी गट

K-SCHEME

(के-स्कीम)

मराठी - इंग्रजी (द्विभाषिक) माध्यम
(अभियांत्रिकी व तंत्रज्ञानातील तृतीयसत्र
पदविका)



महाराष्ट्र राज्य तंत्रशिक्षण मंडळ, मुंबई

(स्वायत्त) (ISO 9001:2015) (ISO/IEC 27001:2013)

मार्गदर्शक

अकोले किशोर प्रल्हाद

प्रभारी विभागप्रमुख, अणुविद्युत अभियांत्रिकी

प्रमुख समन्वयक

डॉ. एस एम गणेचारी

प्राचार्य

समन्वयक

निर्मला कांबळे

अधिव्याख्याता, अणुविद्युत आणि दूरसंचार अभियांत्रिकी

लेखक

दिपक कुलकर्णी

विभागप्रमुख, अणुविद्युत आणि दूरसंचार अभियांत्रिकी

अमृता ठोकडे

अधिव्याख्याता, अणुविद्युत आणि दूरसंचार अभियांत्रिकी

माधुरी कुलकर्णी

अधिव्याख्याता, अणुविद्युत आणि दूरसंचार अभियांत्रिकी

मनीष साळवी

अधिव्याख्याता, संगणक आणि दूरसंचार अभियांत्रिकी

संध्या विष्णू करांडे

विभागप्रमुख, अणुविद्युत आणि दूरसंचार अभियांत्रिकी



महाराष्ट्र राज्य तंत्र शिक्षण मंडळ

(स्वायत्त) (ISO: ९००१:२०१५) (ISO/IES: २७००१-२०१३)

शासकीय तंत्रनिकेतन इमारत, चौथा मजला, ४९, खेरवाडी, बांद्रा (पूर्व), मुंबई - ४०० ०५१.

दूरध्वनी क्र.: ०२२-६२५४२१७० / १६१

Email : director@msbte.com

Web : www.msbte.org.in



प्रास्ताविक

महाराष्ट्र राज्यातील पदविका स्तरावरील तंत्रशिक्षणामध्ये विद्यार्थ्यांचे रोजगार कौशल्य विकसित करून विद्यार्थ्यांचा सर्वांगीण विकास घडवून आणण्याकरिता महाराष्ट्र राज्य तंत्रशिक्षण मंडळ कटिबद्ध आहे. उद्योगधंद्यातील बदलत्या तंत्रज्ञानाशी संबंधित गरजा लक्षात घेऊन महाराष्ट्र राज्य तंत्र शिक्षण मंडळाकडून पदविका अभ्यासक्रम वेळोवेळी अद्यावत करण्यात येतो. अभियांत्रिकी पदविका अभ्यासक्रम शिकत असतांना संकल्पनात्मक ज्ञान, सुसंगत संदर्भ, प्रश्न विचारणे, विश्वसनिय पुरावे, कारणमीमांसा आणि सुस्पष्ट निकष यांचा वापर करून अर्थाची उकल करण्याची, विश्लेषण व मूल्यमापन करण्याची तसेच तर्काने अनुमान काढण्याची क्षमता म्हणजेच चिकित्सक विचार विद्यार्थ्यांमध्ये अधिक दृढ होतील असा मला विश्वास आहे. जेव्हा विद्यार्थी ज्ञान मिळवण्याच्या माध्यमाशी पूर्णपणे परिचित आणि सोयीस्कर असतात, तेव्हा त्यांच्यासाठी वर्गातील चर्चेत भाग घेणे सोपे होते, संकल्पनात्मक व सैद्धांतिक बाबींचे आकलन परिपूर्ण होते, संज्ञानात्मक क्षमता सुधारते आणि त्यांचा आत्मविश्वास देखील वाढतो या सर्व गोष्टींचा विचार करून मंडळाकडून शैक्षणिक सामुग्रीची निर्मिती करण्यात आलेली आहे. भारत देश हा खेड्यापाड्यातून विकसित झालेला देश असून ग्रामीण भागातील विद्यार्थ्यांना तांत्रिक शिक्षण घेतांना भाषेचा अडसर न येता तांत्रिक बाबींचा आशय समजून घेणे शक्य होईल या दृष्टिकोनातून महाराष्ट्र राज्य तंत्र शिक्षण मंडळाने पदविका स्तरावरील तांत्रिक शिक्षणाकरिता विद्यार्थ्यांना मराठी-इंग्रजी द्विभाषिक माध्यमाचा पर्याय शैक्षणिक वर्ष २०२१-२२ पासून उपलब्ध करून दिलेला आहे.

राष्ट्रीय शैक्षणिक धोरण-२०२० प्रादेशिक भाषेतील शिक्षणास प्रोत्साहन देते, ज्यामुळे विद्यार्थ्यांना तांत्रिक अभ्यासक्रमांसाठी प्रादेशिक भाषांतुन शिक्षणाचे माध्यम निवडता येते. सदर धोरणामुळे प्रादेशिक भाषांमध्ये तांत्रिक सामग्री आणि अभ्यास सामग्रीचा विकास आणि भाषांतर निर्माण करण्याची आवश्यकता आहे. त्यास अनुसरून मंडळाने मराठी-इंग्रजी द्विभाषिक माध्यमाचा पर्याय द्वितीय व तृतीय वर्षाकरिताही उपलब्ध करून देण्यात आला आहे. तसेच त्याकरिताची शैक्षणिक सामग्रीही संबंधीत भागधारकरांना उपलब्ध करून देण्यात येत आहे.

पदविका स्तरावरील तंत्रशिक्षण अधिक दर्जेदार करण्यासाठी महाराष्ट्रातील अनुभवी व तज्ञ अध्यापकांनी व्यवहारिक मराठी भाषा व इंग्रजी भाषेतील तांत्रिक शब्दावली यांचा वापर करून मराठी - इंग्रजी भाषेचा सुवर्णमध्य साधण्याचा प्रयत्न केलेला आहे. मंडळाच्या स्तरावर गठीत सुकाणू समितीमार्फत सदर शैक्षणिक सामुग्रीचा दर्जा, तसेच इतर बाबींची तपासणी करण्यात आलेली आहे. त्यामुळे सदर शैक्षणिक सामुग्री अधिक सम्पन्न झालेली असून विद्यार्थी त्यांच्या व्यक्तिमत्त्वाचा सुसंवादी आणि सर्वांगीण विकास साधतील. परिणामतः विश्वस्तरीय मनुष्यबळाच्या गरजा पूर्ण करण्यात महाराष्ट्र राज्य अग्रेसर राहील व पर्यायाने राष्ट्रनिर्मिती करिता निश्चितच हातभार लागेल, असा मला विश्वास आहे.

अभियांत्रिकी पदविका अभ्यासक्रमातील प्रमुख विषयांची मराठी-इंग्रजी द्विभाषिक शैक्षणिक सामुग्री बनविण्यासाठी अध्यापक व सुकाणू समितीचे सदस्य यांनी दर्शविलेले समर्पण व वचनबद्धता कौतुकास पात्र आहे, या सर्वांचे मी मनः पूर्वक अभिनंदन करतो !


(प्रमोद नाईक)

संचालक

म. रा. तंत्र शिक्षण मंडळ, मुंबई.

अनुक्रमणिका

अ. क्र.	युनिटचे नाव	पान क्र.
1	नंबर सिस्टम्स Number Systems	1
2	लॉजिक गेट्स आणि बुलियन बीजगणित Logic Gates and Boolean Algebra	33
3	कॉम्बिनेशनल लॉजिक सर्किट्स Combinational Logic Circuits	55
4	सीक्वेंशियल लॉजिक सर्किट्स Sequential Logic Circuits	97
5	डेटा कोणवेरटर्स आणि मेमरीज Data Converters and Memories	145

युनिट 1: नंबर सिस्टम्स (Number Systems)

कोर्स निष्पत्ती COURSE OUTCOME (CO) :

डिजिटल प्रणालीच्या कार्याचा अर्थ लावण्यासाठी नंबर सिस्टम्स आणि कोड संकल्पना लागू (Apply) करणे.

Apply number system and codes concept to interpret working of digital systems.

युनिट निष्पत्ती (Unit Outcomes):

1.a दिलेली संख्या एका नंबर सिस्टम्स मधून दुसऱ्या नंबर सिस्टम्स मध्ये रूपांतरित करणे.

Convert the given number from one number system to another number system.

1.b बायनरी संख्यांवर अंकगणितीय प्रक्रिया (operations) करणे.

Perform arithmetic operations on binary numbers.

1.c दिलेल्या बायनरी संख्यांची वजाबाकी (subtraction) 1's आणि 2's च्या पूरक (1's and 2's complement) पद्धतीचा वापर करून करणे.

Subtract given binary numbers using 1's and 2's complement method.

1.d दिलेला सांकेतिक क्रमांक इतर निर्दिष्ट (specified) कोड मध्ये रूपांतरित करणे.

Convert the given coded number into the other specified code.

1.e दिलेल्या कोडचे अनुप्रयोग (applications) लिहिणे.

Write the application of the given code.

1.f दिलेल्या दशांश (Decimal) संख्येसाठी BCD बेरीज आणि वजाबाकी करणे.

Perform BCD addition and subtraction for the given Decimal numbers.

परिचय :

बहुतकरून सर्व आधुनिक इलेक्ट्रॉनिक उपकरणे जसे की संगणक, माहिती उपकरणे, डिजिटल कॅमेरे, डिजिटल टेलिव्हिजन, फ्लॅश मेमरी, मोबाईल फोन, हार्ड डिस्क आणि संगणक मेमरीची उपकरणे, इत्यादी डिजिटल इलेक्ट्रॉनिक्सवर अवलंबून असतात. डिजिटल इलेक्ट्रॉनिक्स सामान्यतः लॉजिक सर्किट (Logic circuit) वर आधारित असतात. उदाहरणार्थ, विद्युत प्रवाह (electric current) असल्यास हे '1' जर विद्युत प्रवाह उपस्थित नसेल तर हे '0' म्हणून प्रस्तुत केले जाते. डिजिटल इलेक्ट्रॉनिक्स एक (1) आणि शून्याच्या (0) मालिकेवर आधारित आहे.

सिग्नल (Signal):

सिग्नल हा एक विद्युत चुंबकीय लहर (electromagnetic signal) किंवा विद्युत प्रवाह (electric current) आहे, जो एका शोपनसिस्टम (system) मधून दुसऱ्या सिस्टममध्ये डेटा वाहून नेण्यासाठी वापरला जातो.

अॅनालॉग (Analog) सिग्नल :

हा एक सतत स्वरूपाचा तरंग (continuous wave) आहे किंवा ज्यामध्ये निरनिराळ्या मूल्यांची असीम (infinite) संख्या असते त्याला अॅनालॉग सिग्नल म्हणतात.

उदाहरणार्थ : रेडिओ वेव्स (radio waves) टेलिव्हिजन वेव्स (television waves) किंवा साउंड वेव्स (sound waves) , इत्यादी.

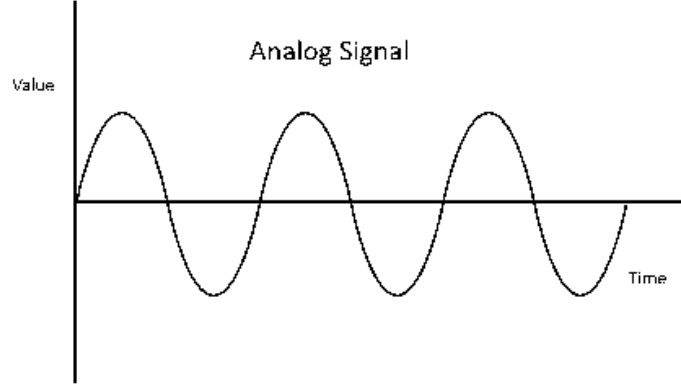


Fig 1.1 Analog signal

अॅनालॉग सिग्नल चे फायदे :

1. ऑडिओ - व्हिडिओ ट्रान्समिशन साठी योग्य असतात.
2. प्रक्रिया करण्यास सोपे असतात.
3. डिजिटल सिग्नल पेक्षा कमी बँडविड्थ (bandwidth) वापरतात.

अॅनालॉग सिग्नल चे तोटे :

1. अॅनालॉग मध्ये डिजिटल पेक्षा कमी गुणवत्तेचे सिग्नल असतात.
2. अॅनालॉग सिग्नलबाह्य प्रभावांबद्दल संवेदनशील (sensitive to external influences) असतात.
3. अॅनालॉग सिग्नल संग्रहित (store) करणे खूप कठीण आहे.

डिजिटल (Digital) सिग्नल :

ज्या सिग्नलमध्ये विशिष्ट मूल्यांची मर्यादित (finite number of distinct values) संख्या असते त्याला डिजिटल सिग्नल म्हणतात. डिजिटल सिग्नल सतत तरंग (continuous wave) नसतात. ते स्वतंत्र वेळ सिग्नल (discrete time) सिग्नल आहेत. उदाहरणार्थ : डिजिटल सिग्नल्स वापरणाऱ्या उपकरणांची विस्तृत श्रेणी आहे. यामध्ये स्मार्टफोन, वैद्यकीय उपकरणे, स्मार्ट घड्याळे, डिजिटल घड्याळे, इत्यादी उपकरणांचा समावेश आहे.

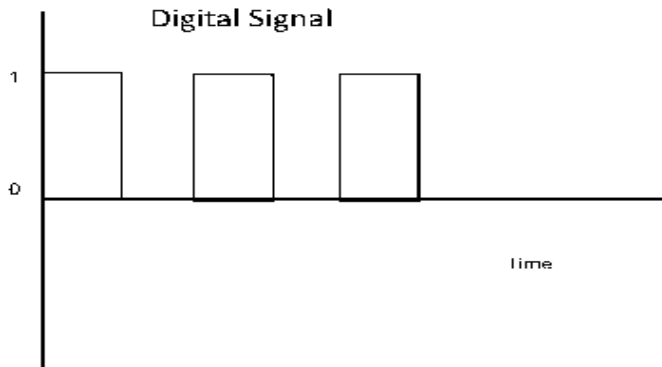


Fig 1.2 Digital Signal

डिजिटल सिग्नल चे फायदे :

1. डिजिटल डेटा (data) सहजपणे कॉम्प्रेस (compress) केला जाऊ शकतो.
2. डिजिटल डेटा संग्रहित करणे शक्य आहे.

- डिजिटल सिग्नल कमी आवाज (less noise), डिस्टोरशन (distortion) आणि इंटरफरन्स (interference) सह माहिती पोहोचवू शकतात.
- डिजिटल सिग्नल वापरणारे उपकरण अधिक सामान्य व कमी खर्चिक आहे.

डिजिटल सिग्नल चे तोटे :

- सॅम्पलिंग (sampling) मुळे माहितीचे नुकसान (loss) होऊ शकते.
- त्यासाठी अधिक बँडविड्थ आवश्यक आहे.

अॅनालॉग सिग्नल आणि डिजिटल सिग्नलमधील तुलना :

Table 1.1 Difference between Analog and Digital Signal

अ.क्र.	अॅनालॉग सिग्नल	डिजिटल सिग्नल
1	अॅनालॉग सिग्नल हा साइन वेव (sine wave), ट्रॅंग्यूलर वेव (triangular wave), रँडम सिग्नल, ध्वनि वेव्स (voice signal) इ. द्वारे दर्शविला जातो.	डिजिटल सिग्नल हा स्क्वेअर वेव (square wave), पल्सेस (pulses) द्वारे दर्शविला जातो.
2	अॅनालॉग सिग्नल हा एक सतत स्वरूपाचा तरंग (continuous wave) आहे जो वेळेनुसार बदलतो.	डिजिटल सिग्नल हे स्वतंत्र वेळ (discrete time) असते जे बायनरी स्वरूपात माहितीचे प्रतिनिधित्व करते.
3	सिग्नल बँडविड्थ कमी असते.	सिग्नल बँडविड्थ जास्त असते.
4	अॅनालॉग सिग्नल वेव्स च्या स्वरूपात डेटा प्रसारित करतो.	डिजिटल सिग्नल बायनरी स्वरूपात (0 आणि 1) डेटा प्रसारित करतो.
5	रेडिओ वेव्स (radio waves) टेलिव्हिजन वेव्स (television waves), ध्वनी वेव्स (sound waves) इ. ही अॅनालॉग सिग्नलची उदाहरणे आहेत.	संगणकात वापरले जाणारे बायनरी सिग्नल, स्मार्ट फोन, वैद्यकीय उपकरणे, स्मार्ट घड्याळे, अशा प्रकारच्या डिजिटल इलेक्ट्रॉनिक्स उपकरणांमध्ये इ. ही डिजिटल सिग्नलची उदाहरणे आहेत.
6	फिल्टर्स, ऍम्प्लिफायर, पावर सप्लायर्स यामध्ये अॅनालॉग सिग्नलचा वापर होतो.	संगणक, काउंटर्स, मायक्रोप्रोसेसर यामध्ये डिजिटल सिग्नलचा वापर होतो.

1.1 नंबर सिस्टम्स (Number Systems) :

नंबर सिस्टम्स ही संख्या दर्शविण्याचा एक मार्ग आहे. डिजिटल सिस्टीम मध्ये काही अंक आणि चिन्हे असतात जे संख्येत असलेल्या स्थानावर अवलंबून भिन्न मूल्य दर्शवतात म्हणून नंबर सिस्टम्स प्रत्येक संख्येचे एक विशेष प्रतिनिधित्व (unique representation) प्रदान करते. संख्येतील कोणत्याही अंकाचे मूल्य पुढील मुद्द्याद्वारे निर्धारित केले जाऊ शकते.

- अंक (Number)
- दशांश चिन्ह (Decimal point)
- नंबर सिस्टमचा आधार (Base of number system)

मूलांक किंवा बेस (Radix or Base) :

अंकामध्ये असू शकणाऱ्या मूल्यांची संख्या सिस्टीमच्या पायाइतकी (Base) असते. स्थानात्मक अंक प्रणालीमध्ये बेस हे संख्या दर्शविण्यासाठी वापरतात. त्याला मूलांक (Radix) असेही म्हणतात. उदाहरणार्थ डेसिमल प्रणालीसाठी बेस 10 आहे कारण ते दहा अंक म्हणजे 0 ते 9 पर्यंतचे वापरतात, ज्यामध्ये प्रत्येक अंकात 10 भिन्न मूल्य असू शकतात. आकृती 1.3 मध्ये डेसिमल प्रणालीमध्ये प्रत्येक अंकाला बेस दहाच्या पॉवर ने गुणाकार केलेला आहे.

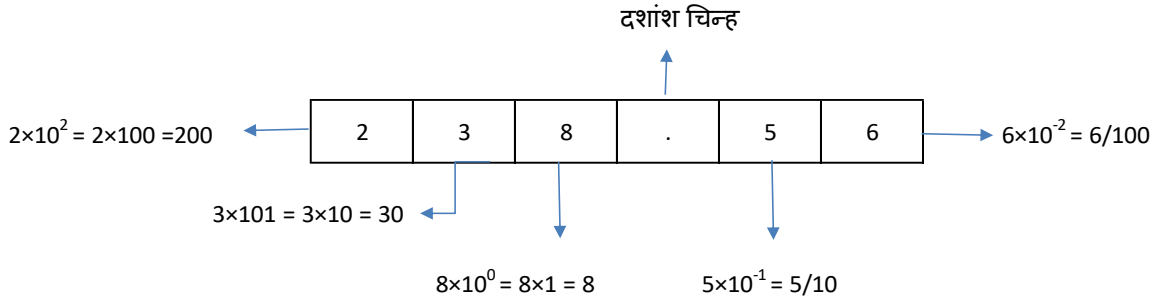


Fig 1.3 Number multiplied by base 10

म्हणूनच, सर्व नंबर सिस्टम्ससाठी एक नियम लागू करतात एका सामान्य संख्येसाठी आकृती 1.4 मध्ये दाखवल्याप्रमाणे प्रत्येक अंकाला बेसच्या काही पॉवर ने गुणाकार करतात.

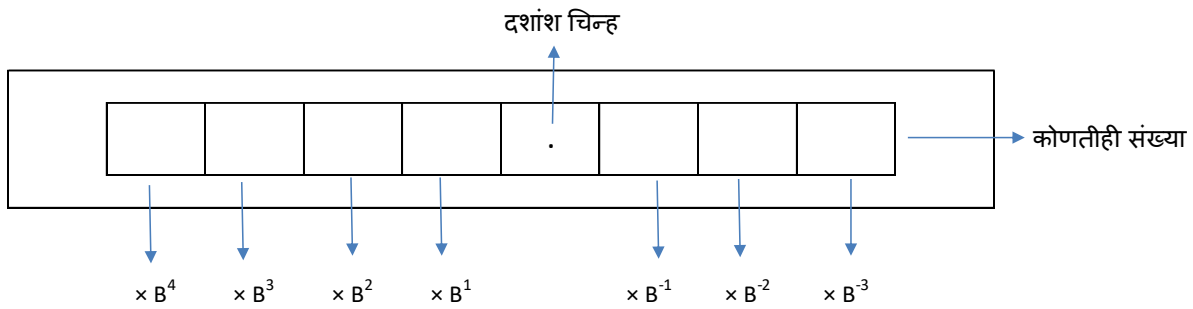


Fig 1.4 Base 8

1.1.1 टाइप्स ऑफ नंबर सिस्टम्स (Types of number systems) :

नंबर सिस्टम्सचे विविध प्रकार आहेत:

1. दशांश नंबर सिस्टम्स बेस -10 (Decimal Number System)
2. बायनरी नंबर सिस्टम्स बेस -2 (Binary Number System)
3. ऑक्टल नंबर सिस्टम्स बेस -8 (Octal Number System)
4. हेक्साडेसिमल नंबर सिस्टम्स बेस -16 (Hexadecimal Number System)

1. डेसीमल नंबर सिस्टम (Decimal Number System) :

डेसिमल ला दशांश नंबर सिस्टम्स म्हणून ओळखले जाते कारण यामध्ये एकूण दहा अंकांचा समावेश असतो. 0 पासून 9 पर्यंत अंक असलेल्या नंबर सिस्टम्सला डेसिमल नंबर सिस्टम्स म्हणतात.

डेसिमल नंबर सिस्टम्सची वैशिष्ट्ये :

1. डेसिमल नंबर सिस्टम्सचा बेस 10 आहे.
2. अंकाचे सर्वात मोठे मूल्य 9 आहे.
3. प्रत्येक ठिकाण 10 च्या भिन्न गुणाकार (multiples) दर्शवते. या गुणाकारांना भारित मूल्ये (weighted values) देखील म्हणतात.

Table 1.2 Decimal Number system

10^3	10^2	10^1	10^0		10^{-1}	10^{-2}	10^{-3}
=1000	=100	=10	=1	.	=0.1	=0.01	=0.001
सर्वात लक्षणीय अंक(Most significant digit/Bit) (MSB)				दशांश चिन्ह			किमान लक्षणीय अंक(Least significant digit/Bit) (LSB)

- सर्वात लक्षणीय अंक (Most significant digit-MSB): सर्वात जास्त भार (weight) असलेल्या डाव्या अंकाला एखाद्या संख्येचा सर्वात महत्त्वाचा अंक असे म्हणतात.
- किमान लक्षणीय अंक (Least significant digit-LSB): सर्वात कमी भार (weight) असलेल्या सर्वात उजव्या अंकाला एखाद्या संख्येचा कमीत कमी लक्षणीय अंक असे म्हणतात.

उदाहरण :

1) 10 च्या पॉवरच्या दृष्टीने डेसिमल संख्या 1248.356 दर्शवा.

उत्तर : या संख्येचा पूर्णांकाचा भाग 1248 व या संख्येचा अपूर्णाकीय भाग 0.356 आहे. 8, 4, 2 आणि 1 या अंकाचे भार अनुक्रमे 10^0 , 10^1 , 10^2 आणि 10^3 असे आहे. त्याचप्रमाणे 3, 5 आणि 6 या अंकाचे भार अनुक्रमे 10^{-1} , 10^{-2} , 10^{-3} असे आहे.

$$1248.356 = (1 \times 10^3) + (2 \times 10^2) + (4 \times 10^1) + (8 \times 10^0) + (3 \times 10^{-1}) + (5 \times 10^{-2}) + (6 \times 10^{-3})$$

2. बायनरी नंबर सिस्टम्स बेस -2 (Binary Number System) :

आधुनिक डिजिटल प्रणाली हे बायनरी नंबर सिस्टीम सह कार्य करतात. याचा बेस दोन आहे व त्याचे घटक किंवा अंक फक्त 0 आणि 1 असतात. या प्रणालीचा बेस दोन असल्याने ही बायनरी नंबर सिस्टीम म्हणून ओळखली जाते. या प्रणालीतील 0 आणि 1 हे अंक म्हणून ओळखले जातात. डिजिटल इलेक्ट्रॉनिक्स मध्ये या नंबर सिस्टीमचा मोठ्या प्रमाणात वापर होतो.

बायनरी नंबर सिस्टम्सची वैशिष्ट्ये :

1. या नंबर सिस्टम्सचा बेस दोन आहे.
2. 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.
3. या नंबर सिस्टम्सत बायनरी पॉईंटच्या डाव्या बाजूस क्रमिक स्थिती असलेल्या अंकाचे भार 2^0 , 2^1 , 2^2 , इत्यादी असतात. त्याचप्रमाणे बायनरी पॉईंटच्या उजवीकडे क्रमिक स्थाने असलेल्या अंकाचे भार 2^{-1} , 2^{-2} , 2^{-3} , इत्यादी असतात. म्हणजे प्रत्येक स्थानाचे विशिष्ट भाग असतात जे बेस 2 ची पॉवर असतात.

Table 1.3 Binary Number system

2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}
=8	=4	=2	=1	.	=0.5	=0.25	=0.125
सर्वात लक्षणीय अंक(Most significant bit) (MSB)				बायनरी पॉइंट			किमान लक्षणीय अंक(Least significant bit) (LSB)

उदाहरण:

1) २ च्या पॉवर ने बायनरी नंबर 1101.010 दर्शवा.

उत्तर : $1101.010 = (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) + (0 \times 2^{-1}) + (1 \times 2^{-2}) + (0 \times 2^{-3})$

3. ऑक्टल नंबर सिस्टम बेस -8 (Octal Number System) :

ऑक्टल (octal) नंबर सिस्टममध्ये बेस 8 असतो आणि तो संख्यांचे प्रतिनिधित्व करण्यासाठी 0 ते 7 पर्यंतच्या अंकांचा वापर करतो. मोठ्या बायनरी संख्या लहान करण्यासाठी ऑक्टल नंबर सिस्टीम वापरतात. डिजिटल सर्किट्स आणि संगणक अंतर्गतपणे बायनरी सिस्टीम प्रमाणे कार्य करतात ते ऑक्टल सिस्टीम वापरत नाहीत. ऑक्टल सिस्टीम चा वापर फक्त वापरकर्ता (user) ची सोय म्हणून केला जातो.

ऑक्टल नंबर सिस्टमची वैशिष्ट्ये :

- या नंबर सिस्टमचा बेस आठ आहे.
- ऑक्टल (octal) प्रणालीतील अंकांचे सर्वात मोठे मूल्य सात आहे.
- या नंबर सिस्टमसत ऑक्टल बिंदूच्या डाव्या बाजूस क्रमिक स्थान असलेल्या अंकांचे भार $8^0, 8^1, 8^2, 8^3, \dots$ इत्यादी असतात. त्याचप्रमाणे ऑक्टल बिंदूच्या उजवीकडे क्रमिक स्थान असलेल्या अंकांचे भार $8^{-1}, 8^{-2}, 8^{-3}, \dots$ इत्यादी असतात. म्हणजे प्रत्येक स्थानाचे विशिष्ट भार असतात जे बेस 8 ची पॉवर असतात.

Table 1.4 Octal number system

8^3	8^2	8^1	8^0		8^{-1}	8^{-2}	8^{-3}
=512	=64	=8	=1	.	=1/8	=1/64	=1/512
सर्वात लक्षणीय अंक(Most significant digit/Bit) (MSB)				ऑक्टल पॉइंट			किमान लक्षणीय अंक(Least significant digit/Bit) (LSB)

उदाहरण :

1) ऑक्टल संख्या 625 ला 8 च्या पॉवरच्या दृष्टीने दर्शवा.

उत्तर : $625 = (6 \times 8^2) + (2 \times 8^1) + (5 \times 8^0)$

4. हेक्साडेसिमल नंबर सिस्टम्स बेस -16 (Hexadecimal Number System) :

हेक्साडेसिमल प्रणालीमध्ये संख्या 16 बेस ने दर्शविल्या जातात . प्रथम संख्यांचे डेसिमल प्रमाणे 0 ते 9 पर्यंतचे अंक वापरतात आणि पुढील 10 ते 15 पर्यंतच्या संख्या A ते F पर्यंतच्या अक्षरांचा वापर करून दर्शविल्या जातात. हेक्साडेसिमल संख्येचा आकार बायनरी नंबर पेक्षा खूपच लहान असल्याने त्यांना लिहिणे व लक्षात ठेवणे सोपे असते. मायक्रोप्रोसेसर / मायक्रोकंट्रोलर मध्ये या संख्या मोठ्या प्रमाणावर वापरतात .

हेक्साडेसिमल नंबर सिस्टम्सची वैशिष्ट्ये :

1. या नंबर सिस्टम्सचा बेस 16 आहे.
2. हेक्साडेसिमल प्रणालीतील अंकाचे सर्वात मोठे मूल्य 15(F) आहे.
3. या नंबर सिस्टम्सत हेक्साडेसिमल बिंदूच्या डाव्या बाजूस क्रमिक स्थान असलेल्या अंकांचे भार $16^0, 16^1, 16^2, 16^3, \dots$ इत्यादी असतात. त्याचप्रमाणे हेक्साडेसिमल बिंदूच्या उजवीकडे क्रमिक स्थान असलेल्या अंकांचे भार $16^{-1}, 16^{-2}, 16^{-3}, \dots$ इत्यादी असतात. म्हणजे प्रत्येक स्थानाचे विशिष्ट भार असतात जे बेस 16 ची पॉवर असतात.

Table 1.5 Hexadecimal number system

16^3	16^2	16^1	16^0		16^{-1}	16^{-2}	16^{-3}
=4096	=256	=16	=1	.	=1/16	=1/256	=1/4096
सर्वात लक्षणीय अंक(Most significant digit/Bit) (MSB)				हेक्साडेसिमल पॉईंट			किमान लक्षणीय अंक(Least significant digit/Bit) (LSB)

उदाहरण :

- 2) हेक्साडेसिमल संख्या AF32 ला 16 च्या पॉवरच्या दृष्टीने दर्शवा.

$$\text{उत्तर : AF32} = (10 \times 16^3) + (15 \times 16^2) + (3 \times 16^1) + (2 \times 16^0)$$

Table 1.6 Relation between Decimal, Binary, Octal and Hexadecimal number system

डेसिमल (बेस - 10)	बायनरी (बेस - 2)	ऑक्टल (बेस - 8)	हेक्साडेसिमल (बेस - 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

1.1.2 नंबर सिस्टम चे रूपांतरण :

बेस (radix) r मधील संख्येचे दशांश मध्ये रूपांतर पाँवर सीरिजमध्ये दिलेल्या संख्येचा विस्तार करून आणि सर्व संज्ञा जोडून केले जाते.

बेस r पासून दशांश नंबर सिस्टममध्ये रूपांतरण (Base r to Decimal number system conversion) :

स्टेप्स :

1. दिलेल्या संख्येची नोंद करा.
2. वेगवेगळ्या स्थानांचे संबंधित भार (weight) लिहा.
3. दिलेल्या संख्येतील प्रत्येक अंकाला गुणात्मक (product) संख्या मिळविण्यासाठी संबंधित भारासह गुणाकार करा.
4. डेसिमल समतुल्य मिळवण्यासाठी सर्वगुणात्मक संख्या जोडा.

खालील उदाहरणे बायनरी (binary), ऑक्टल (octal) आणि हेक्स (Hex) नंबरचे त्याच्या समतुल्य दशांश मध्ये रूपांतरण दर्शवितात.

उदाहरणे :

- 1) बायनरी नंबर $(1010)_2$ चे डेसिमल समतुल्य मध्ये रूपांतरित करा.

उत्तर :

बायनरी नंबर :	1	0	1	0
संबंधित भार :	2^3	2^2	2^1	2^0
गुणाकार :	(1×2^3)	(0×2^2)	(1×2^1)	(0×2^0)
	=8	=0	=2	0
बेरीज :	$8+0+2+0 = 10$			
उत्तर :	$(10)_{10}$			

- 2) बायनरी नंबर $(0.1011)_2$ चे डेसिमल समतुल्य मध्ये रूपांतरित करा.

$$\text{उत्तर : } (0.1011) = (0 \times 2^0) + (1 \times 2^{-1}) + (0 \times 2^{-2}) + (1 \times 2^{-3}) + (1 \times 2^{-4})$$

$$= 0 + 0.5 + 0 + 0.125 + 0.0625$$

$$(0.1011) = (0.6875)_{10}$$

- 3) ऑक्टल संख्या $(172)_8$ ला त्याच्या डेसिमल समतुल्य मध्ये रूपांतरित करा.

$$\text{उत्तर : } (172)_8 = (1 \times 8^2) + (7 \times 8^1) + (2 \times 8^0)$$

$$= 64 + 56 + 2$$

$$(172)_8 = (122)_{10}$$

- 4) ऑक्टल संख्या $(23.25)_8$ ला त्याच्या डेसिमल समतुल्य मध्ये रूपांतरित करा.

$$\text{उत्तर : } (23.25)_8 = (2 \times 8^1) + (3 \times 8^0) + (2 \times 8^{-1}) + (5 \times 8^{-2})$$

$$= 16 + 3 + 0.25 + 0.07812 = (19.32812)_{10}$$

5) हेक्साडेसिमल संख्या $(2A5)_{16}$ ला त्याच्या डेसिमल समतुल्य मध्ये रूपांतरित करा.

$$\begin{aligned}\text{उत्तर : } (2A5)_{16} &= (2 \times 16^2) + (A \times 16^1) + (5 \times 16^0) \\ &= (2 \times 16^2) + (10 \times 16^1) + (5 \times 16^0) \\ &= 512 + 160 + 5\end{aligned}$$

$$(2A5)_{16} = (677)_{10}$$

6) हेक्साडेसिमल संख्या $(54.D2)_{16}$ ला त्याच्या डेसिमल समतुल्य मध्ये रूपांतरित करा.

$$\begin{aligned}\text{उत्तर : } (54.D2)_{16} &= (5 \times 16^1) + (4 \times 16^0) + (D \times 16^{-1}) + (2 \times 16^{-2}) \\ &= (5 \times 16^1) + (4 \times 16^0) + (13 \times 16^{-1}) + (2 \times 16^{-2}) \\ &= 80 + 4 + 0.8125 + 0.0078125\end{aligned}$$

$$(54.D2)_{16} = (84.8203125)_{10}$$

डेसीमल नंबर सिस्टम तो बेस र कन्वर्शन (Decimal number system to base r conversion) : स्टेप्स :

1. दिलेली संख्या नोंद करा .
2. दिलेल्या संख्येला बेस r ने बाकी 0 (zero) येईपर्यंत भागा, समतुल्य (equivalent) शोधण्यासाठी उर्वरित मूल्यांची खालून वर पर्यंत उलट्या क्रमाने यादी करा.
3. जर दिलेल्या डेसिमल संख्ये मध्ये 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.
4. पूर्णांक भागासाठी : डेसिमल संख्येचा पूर्णांक भाग बेस r द्वारे विभाजित करा आणि उर्वरित भाग खाली नोंदवा बाकी शून्य येईपर्यंत भागाचे विभाजन करणे सुरू ठेवा. प्रत्येक स्टेप वरील उर्वरित भागाची नोंद घ्या, समतुल्य शोधण्यासाठी उर्वरित मूल्यांची खालून वरपर्यंत उलट्या क्रमाने यादी करा.
5. अपूर्णांक भागासाठी:
 - i) डेसिमल संख्येच्या अपूर्णांक भागाला बेस r ने गुणाकार करा. गुणाकारत निर्माण झालेली कॅरी एम एस डी (MSD) म्हणून लिहून ठेवा.
 - ii) आता गुणात्मक संख्येच्या अपूर्णांक संख्येला आधीच्या स्टेप पासून बेस r ने गुणाकार करा आणि कॅरी (carry) ला एम एस डी ची पुढची बिट म्हणून नोंदवा.
 - iii) वरील पायऱ्यांची शेवटपर्यंत पुनरावृत्ती करा. शेवटची कॅरी एल एस डी (LSD) चे प्रतिनिधित्व करते, आता रूपांतरित संख्या मिळविण्यासाठी वर पासून खालपर्यंत मूल्यांची यादी करा.

उदाहरणे:

1) डेसिमल $(29)_{10}$ ला बायनरी मध्ये रूपांतरित करा.

उत्तर :

		बाकी	
2	29		
2	14	1	LSB
2	7	0	
2	3	1	
2	1	1	
2	0	1	MSB

$$(29)_{10} = (11101)_2$$

2) डेसिमल $(146.25)_{10}$ ला ऑक्टल मध्ये रूपांतरित करा.

उत्तर : $(146.25)_{10}$

8	146	2	LSB	
8	18	2		
8	2	2	MSB	
	0			



$$(146)_{10} = (222)_8$$

$$(0.25)_{10} = 0.25 \times 8 = 2.0 \quad \begin{matrix} 2 & \text{MSB} \\ 0.0 \times 8 = 0.0 & 0 & \text{LSB} \end{matrix}$$


$$(0.25)_{10} = (0.20)_8$$

दोन्ही परिणाम एकत्र करा

$$(146.25)_{10} = (222.20)_8$$

3) डेसिमल $(3509)_{10}$ ला हेक्साडेसिमल मध्ये रूपांतरित करा.

उत्तर :

16	3509	5	LSD	
16	219	11		
16	13	13	MSD	
	0			



हेक्साडेसिमल मूल्यासाठी $13=D$, $11=B$, $5=5$

$$(3509)_{10} = (DB5)_{16}$$

सुक्सेसिवे मल्टिप्लिकेशन फॉर फ्रॅक्शनल पार्ट कन्व्हर्शन (Successive multiplication for fractional part conversion) :

स्टेप्स :

1. दिलेल्या अपूर्णाक दशांश संख्येचा बेस (रेडिक्स r) ने गुणाकार करा.
2. या गुणाकारात उत्पन्न झालेला हातचा (carry) सर्वात लक्षणीय अंक (MSB) म्हणून नोंदवा.
3. स्टेप 2 मधील गुणाकाराच्या केवळ अंशात्मक संख्येचा बेसने गुणाकार करा आणि कॅरीला MSB वर पुढील बिट (bit) म्हणून नोंदवा.
4. स्टेप्स 2 आणि 3 शेवटपर्यंत पुन्हा करा. शेवटचा कॅरी रूपांतरित क्रमांकाच्या समतुल्य LSB चे प्रतिनिधित्व करेल.

उदाहरणे :

1) डेसिमल संख्या $(0.8)_{10}$ समतुल्य बायनरी संख्येमध्ये रूपांतरित करा.

उत्तर :

अपूर्णांक

दशांश	बेस	गुणाकार	कॅरी
0.8	×	2	= 1.6
0.6	×	2	= 1.2
0.2	×	2	= 0.4
0.4	×	2	= 0.8
0.8	×	2	= 1.6

$$(0.8)_{10} = (0.11001)_2$$

2) डेसिमल संख्या $(0.6234)_{10}$ समतुल्य ऑक्टल संख्येमध्ये रूपांतरित करा.

उत्तर :

अपूर्णांक

दशांश	बेस	गुणाकार	कॅरी
0.6234	×	8	= 4.9872
0.9872	×	8	= 7.8976
0.8976	×	8	= 7.1808
0.1808	×	8	= 1.4464
0.4464	×	8	= 3.5712

$$(0.6234)_{10} = (0.47713)_8$$

3) डेसिमल संख्या $(0.31)_{10}$ समतुल्य हेक्साडेसिमल संख्येमध्ये रूपांतरित करा.

उत्तर :

अपूर्णांक

दशांश	बेस	गुणाकार	कॅरी	हेक्साडेसिमल
0.31	×	16	= 4.96	4
0.96	×	16	= 15.36	15
0.36	×	16	= 5.76	5
0.76	×	16	= 12.16	12
0.16	×	16	= 2.56	2

$$(0.31)_{10} = (0.4F5C2)_{16}$$

बायनरी मधून इतर नंबरसिस्टममध्ये रूपांतरण :**स्टेप्स :**

1. ऑक्टल संख्यांसाठी LSB पासून सुरुवात करून बायनरी नंबरला 3 बिट्सच्या गटांमध्ये विभाजित करा.
2. हेक्साडेसिमल नंबरसाठी बायनरी नंबरला LSB पासून MSB पर्यंत 4 बिट विभागात विभाजित करा.
3. ऑक्टलसाठी 3 बिट बायनरी क्रमांकाचा प्रत्येक गट आणि हेक्साडेसिमलसाठी 4 बिट बायनरी क्रमांक त्याच्या समतुल्य ऑक्टल क्रमांक आणि हेक्स समतुल्य मध्ये रूपांतरित करा.

उदाहरणे :

- 1) बायनरी संख्या $(00010001)_2$ समतुल्य ऑक्टल संख्येमध्ये रूपांतरित करा.

उत्तर : दिलेल्या बायनरी संख्येमध्ये आठ बिट्स असल्याने डाव्या बाजूला एम एस बी (MSB) ला एक शून्य ऍड करा.

बायनरी संख्या : 3 बिट चा गट

000	010	001
↓	↓	↓
0	2	1

ऑक्टल संख्या :

$$(00010001)_2 = (021)_8 = (21)_8$$

- 2) बायनरी संख्या $(1101.0011)_2$ समतुल्य हेक्साडेसिमल संख्येमध्ये रूपांतरित करा.

उत्तर :

बायनरी संख्या : 4 बिट चा गट

1101	.	0011
↓	↓	↓
D	.	3

हेक्साडेसिमल संख्या :

$$(1101.0011)_2 = (D.3)_{16}$$

इतर नंबरसिस्टम मधून बायनरी मध्ये रूपांतरण :

उदाहरणे :

- 1) ऑक्टल संख्या $(62)_8$ समतुल्य बायनरी संख्येमध्ये रूपांतरित करा.

उत्तर :

ऑक्टल संख्या :

6	2
110	010

3 बीटचा बायनरी गट :

$$(62)_8 = (110010)_2$$

- 2) हेक्साडेसिमल संख्या $(5AC)_{16}$ समतुल्य बायनरी संख्येमध्ये रूपांतरित करा.

उत्तर :

हेक्साडेसिमल संख्या :

4 बीटचा बायनरी गट :

5	A	C
0101	1010	1100

$$(5AC)_{16} = (010110101100)_2$$

1.2 बायनरी अरिथमेटिक (Binary Arithmetic) :

बायनरी अंकगणित हा सर्व संगणक आणि इतरांना एक डिजिटल प्रणालीचा आवश्यक भाग आहे. बेरीज, वजाबाकी, गुणाकार आणि भागाकार यासारखे अंकगणितीय ऑपरेशन बायनरी अंक गणितात केली जातात. बायनरी नंबर वर ही ऑपरेशन करताना काही नियम लागू करावे लागतात.

1.2.1 बायनरी अडिटीओ (Binary Addition) :

बायनरी बेरीज ही डेसिमल प्रणालीतील बेरीज प्रमाणेच केली जाते. ही बेरीज बायनरी गुणाकार, वजाबाकी, आणि भागाकाराची गुरुकिल्ली आहे.

Table 1.7 Rules for binary addition

केस	A+B	कॅरी (Carry)	बेरीज (Sum)
1	0+0	0	0
2	0+1	0	1
3	1+0	0	1
4	1+1	1	0

उदाहरणे :

1) $(10)_{10} + (5)_{10}$ बायनरी बेरीज करा.

उत्तर : $(10)_{10} = (1010)_2$ आणि $(5)_{10} = (0101)_2$

$$\begin{array}{r} 1010 \\ + 0101 \\ \hline \end{array}$$

$$1111 \quad (10)_{10} + (5)_{10} = (15)_{10} = (1111)_2$$

2) बायनरी बेरीज करा $(1001.111)_2 + (1011.11)_2$.

उत्तर :

$$\begin{array}{r} 11111 \quad \text{कॅरी (carry)} \\ 1001.111 \\ + 1011.110 \\ \hline 10101.101 \end{array}$$

1.2.2 बायनरी सबट्रॅक्शन (Binary Subtraction) :

वजाबाकी आणि बॉरो हे दोन शब्द बायनरी वजाबाकी साठी वापरले जातात. बायनरी वजाबाकीचे चार मूलभूत नियम खालील दिलेल्या टेबल 1.8 प्रमाणे आहेत.

Table 1.8 Rules for Binary subtraction

केस	A-B	बॉरो (Borrow)	वजाबाकी (Difference)
1	0-0	0	0
2	0-1	1	1
3	1-0	0	1
4	1-1	0	0

बायनरी सबट्रॅक्शन (subtraction) करताना काही वेळा पुढील अंकातून बॉरो (borrow) घेणे आवश्यक असते. तक्ता 1.8 मध्ये केस 2 मध्ये दाखविल्याप्रमाणे जेव्हा आपण 0 मधून 1 वजा करतो तेव्हा फक्त बॉरो घेणे आवश्यक असते.

उदाहरणे :

1) बायनरी वजाबाकी करा $(21)_{10} - (7)_{10}$

उत्तर : $(21)_{10} = (10101)_2$ आणि $(7)_{10} = (0111)_2$

$$\begin{array}{r}
 111 \\
 10101 \\
 - 00111 \\
 \hline
 01110
 \end{array}$$

$$(21)_{10} - (7)_{10} = (14)_{10} = (01110)_2$$

2) बायनरी वजाबाकी करा $(110)_2 - (101)_2$.

उत्तर :

$$\begin{array}{r}
 1 \\
 110 \\
 - 101 \\
 \hline
 001
 \end{array}$$

$$(110)_2 - (101)_2 = (001)_2 = (1)_2$$

1.2.3 बायनरी मल्टिप्लिकेशन (Binary Multiplication) :

बायनरी गुणाकार हा डेसिबल गुणाकार सारखाच असतो. यामध्ये खालील चार मूलभूत नियमांचे पालन करावे लागते.

Table 1.9 Rules for Binary multiplication

केस	$A \times B$	गुणाकार (Multiplication)
1	0×0	0
2	0×1	0
3	1×0	0
4	1×1	1

उदाहरण :

1) $(101)_2 \times (11)_2$

$$\begin{array}{r}
 \text{उत्तर :} \quad 1 \ 0 \ 1 \\
 \times \quad 1 \ 1 \\
 \hline
 \quad 1 \ 0 \ 1 \\
 + \ 1 \ 0 \ 1 \\
 \hline
 1 \ 1 \ 1 \ 1
 \end{array}$$

2) बायनरी गुणाकार करा $(12)_{10} \times (15)_{10}$

$$\begin{array}{r}
 \text{उत्तर :} \quad 1 \ 1 \ 0 \ 0 \\
 \times \quad 1 \ 1 \ 1 \ 1 \\
 \hline
 \quad 1 \ 1 \ 0 \ 0 \\
 + \ 1 \ 1 \ 0 \ 0 \\
 + \ 1 \ 1 \ 0 \ 0 \\
 + \ 1 \ 1 \ 0 \ 0 \\
 \hline
 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0
 \end{array}$$

$$(12)_{10} \times (15)_{10} = (180)_{10} = (10110100)_2$$

1.2.4 बायनरी डिवीजन (Binary Division) :

बायनरी डिविजन हे डेसिमल भागाकारासारखेच असते. याला दीर्घ विभाजन प्रक्रिया असे म्हणतात .तो बिट्सचे एक्सऑरिंग (EX-ORing) करतो.

उदाहरण :

$$\begin{array}{r}
 1) \quad (1100)_2 \div (100)_2 \\
 \quad 1 \ 1 \\
 \hline
 100 \overline{) 1100} \\
 - \quad 1 \ 0 \ 0 \\
 \hline
 \quad 0 \ 1 \ 0 \ 0 \\
 - \quad 1 \ 0 \ 0 \\
 \hline
 \quad 0 \ 0 \ 0
 \end{array}$$

1. सबट्रैक्शन बूकिंग १'स एंड २'स कॉम्प्लीमेंट मेथड (Subtraction using 1's and 2's complement method):

1.3.1 १'स कॉम्प्लीमेंट (1's complement) :

१ चे पूरक (complement) मूळ संख्येची ऋण (negative) संख्या दर्शविते. बायनरी संख्येचे १ चे कॉम्प्लीमेंट दिलेल्या बायनरी संख्येमधील ० ला १ ने आणि १ ला ० ने बदलून मिळतात.

उदाहरण : दिलेल्या बायनरी संख्यांचे १ चे पूरक काढा.

i) 101011

ii) 110010

उत्तर : i) 1 0 1 0 1 1

ii) 1 1 0 0 1 0



0 1 0 1 0 0



0 0 1 1 0 1

बायनरी संख्येचे १ चे पूरक मध्ये रूपांतर करण्यासाठी दिलेली संख्या उलट करा. बायनरी नंबर इनपुटच्या प्रत्येक बिटसाठी फक्त इन्व्हर्टर किंवा नॉट गेट (Inverter Or NOT Gate) वापरून १ चे पूरक तयार करतात. खालील आकृतीमध्ये वन्स कॉम्प्लीमेंट लॉजिक (1's complement logic) ची अंमलबजावणी दिली आहे.

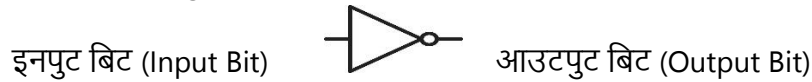


Fig 1.5: Inverter Logic

१ च्या पूरक पद्धतीने $(A)_2 - (B)_2$ वजाबाकी करण्यासाठी खालील स्टेप्स आहेत. जिथे A हा मिन्यूएंड(minuend) आणि बी हा सबट्राहेन्ड(subtrahend) आहे.

स्टेप्स :

- स्टेप १: सबट्राहेन्ड B चे १ चे कॉम्प्लीमेंट काढा.
- स्टेप २: A मध्ये B चे १ चे कॉम्प्लीमेंट ऍड करा.
- स्टेप ३: जर अंतिम कॅरी १ असेल तर स्टेप २ मध्ये मिळालेल्या बेरजेच्या निकालात ऍड करा. जर कॅरी १ असेल तर वजाबाकी पॉझिटिव्ह असेल म्हणून ते मूळ (True) स्वरूपात आहे.
- स्टेप ४: जर स्टेप २ मधील उत्पादित कॅरी ० असेल, तर मिळालेला परिणाम निगेटिव्ह आणि १ च्या कॉम्प्लीमेंट स्वरूपात असेल म्हणून ते मूळ स्वरूपात रूपांतरित करा.

उदाहरण:

- १ च्या पूरक पद्धतीचा वापर करून वजाबाकी करा $(01000011)_2 - (01010100)_2$.

उत्तर : सबट्राहेन्ड $(01010100)_2$ चा १ चा पूरक : 10101011.

मिन्यूएंड : 0 1 0 0 0 0 1 1

सबट्राहेन्ड चा १ चा पूरक : + 1 0 1 0 1 0 1 1

1 1 1 0 1 1 1 0

अंतिम कॅरी 0 दर्शविते की उत्तर नकारात्मक (negative) असल्याने दुरुस्त (correction) करणे आवश्यक आहे. उत्तराला मूळ स्वरूपात रूपांतरित करण्यासाठी सर्व बिट्स उलट (invert) करा.

$$\begin{array}{ccccccccc} 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}$$

$$(01000011)_2 - (01010100)_2 = (00010001)_2$$

1.3.2 2 चे पूरक (2's complement) :

2 चे पूरक हे संगणकीय चिन्हांकित (signed) संख्येच्या प्रतिनिधित्वाची पद्धत म्हणून वापरले जाते. चिन्हांकित संख्यांचे प्रतिनिधित्व करण्यासाठी इतर प्रणालींच्या तुलनेत 2 चे पूरक हे बेरीज, वजाबाकी आणि गुणाकाराची मूलभूत अंकगणितीय क्रिया उच्च अचूकतेने (highly accurately) लागू करते.

बायनरी संख्येचे 2 चे पूरक संख्या 1 चे कॉम्प्लिमेंट संख्येच्या किमान लक्षणीय बिट (LSB) मध्ये 1 जोडून प्राप्त केले जाते किंवा 1 च्या पूरक मध्ये 1 ऍड (add) करून काढली जाते.

2 चे पूरक = 1 चे पूरक + 1

उदाहरण : दिलेल्या बायनरी संख्यांचे 2 चे पूरक काढा.

i) $(11010)_2$

ii) $(101010)_2$

उत्तर: i)

1	1	0	1	0	
↓	↓	↓	↓	↓	
0	0	1	0	1	
			+	1	
					1 चे पूरक (1's complement)

0	0	1	1	0	
					2 चे पूरक (2's complement)

ii)

1	0	1	0	1	0
↓	↓	↓	↓	↓	↓
0	1	0	1	0	1
					1 चे पूरक (1's complement)
					+
					1
					2 चे पूरक (2's complement)

0	1	0	1	1	0
					2 चे पूरक (2's complement)

बायनरी संख्यांच्या 2 चे कॉम्प्लिमेंटचे (2's complement) विविध उपयोग आहेत. मुख्यतः चिन्हांकित (signed) केलेल्या बायनरी संख्येच्या प्रतिनिधित्व मध्ये आणि बायनरी संख्यांसाठी विविध अंकगणितीय ऑपरेशन्स, उदाहरणार्थ, बेरीज, वजाबाकी मध्ये 2 चे कॉम्प्लिमेंट विशेषतः उपयुक्त आहे.

उदाहरण : 5 बिट रजिस्टर वापरलेला आहे तर -5 आणि +5 चे प्रतिनिधित्व खालील आकृती प्रमाणे असेल.



Fig 1.6: -5 and +5 representation

आकृतीमध्ये दाखविल्याप्रमाणे $-5 = 11011$ आणि $+5 = 00101$. या प्रणालीचा फायदा असा आहे की 2 चे पूरक प्रतिनिधित्वामध्ये 0 नेहमी पॉझिटिव्ह आहे असे मानले जाते आणि 1 हे निगेटिव्ह आहे असे मानले जाते.

वजाबाकीच्या 2 चे पूरक पद्धतीच्या मदतीने आपण खालील स्टेप्स वापरून दोन बायनरी संख्या सहजपणे वजा करू शकतो.

स्टेप्स :

1. स्टेप 1: सबट्राहेन्डचे 2 चे पूरक शोधा.
2. स्टेप 2: नंतर ते मिनूएंड मध्ये जोडले जाते.
3. स्टेप 3: बेरीज चे अंतिम कॅरी 1 असेल तर ते काढून टाकले जाते आणि उत्तर पॉझिटिव्ह असते.
4. स्टेप 4: बेरीज चे अंतिम कॅरी 0 असेल तर उत्तर निगेटिव्ह असते व त्यामध्ये सुधारणा केली जाते.

उदाहरणे:

- 1) 2 चे पूरक पद्धती वापरून वजाबाकी करा $(110110)_2 - (100001)_2$.

उत्तर:	1	0	0	0	0	1	सबट्राहेन्ड
	0	1	1	1	1	0	सबट्राहेन्डचे 1 चे पूरक
					+	1	
	0	1	1	1	1	1	सबट्राहेन्डचे 2 चे पूरक
	1	1	1	1			कॅरी (carry)
	1	1	0	1	1	0	मिनूएंड
+	0	1	1	1	1	1	सबट्राहेन्डचे 2 चे पूरक
	1	0	1	0	1	0	1
	↓						अंतिम कॅरी

अंतिम कॅरी एक असल्याने काढून टाका. $(110110)_2 - (100001)_2 = (010101)_2$

- 2) 2 चे पूरक पद्धती वापरून वजाबाकी करा $(1010.11)_2 - (1001.01)_2$.

उत्तर:	1	0	0	1	.	0	1	सबट्राहेन्ड
	0	1	1	0	.	1	0	सबट्राहेन्डचे 1 चे पूरक
						+	1	
	0	1	1	0	.	1	1	सबट्राहेन्डचे 2 चे पूरक
	1	1		1		1		कॅरी

$$\begin{array}{r}
 1 \quad 0 \quad 1 \quad 0 \quad . \quad 1 \quad 1 \quad \text{मिनूएंड} \\
 + \quad 0 \quad 1 \quad 1 \quad 0 \quad . \quad 1 \quad 1 \quad \text{सबट्राहेन्डचे 2 चे पूरक} \\
 \hline
 \end{array}$$

1 0 0 0 1 . 1 0



अंतिम कैरी

अंतिम कैरी एक असल्याने काढून टाका. $(1010.11)_2 - (1001.01)_2 = (0001.10)_2 = (1.10)_2$

1.4 कोड (Codes) :

जेव्हा संख्या अक्षरे किंवा मजकूर, वर्ण चिन्हांच्या विशिष्ट गटाद्वारे दर्शविले जातात तेव्हा असे म्हटले जाते की संख्या, अक्षर किंवा शब्द एन्कोड (encode) केले जात आहेत आणि चिन्हांच्या गटाला कोड म्हणतात. डिजिटल डेटा बायनरी कोड म्हणून प्रस्तुत, संग्रहित आणि प्रसारित केला जातो. बायनरी कोड डिजिटल कम्युनिकेशन आणि डिजिटल कॉम्प्युटरमध्ये वापरले जातात.

1.4.1 बायनरी कोडेड दशांश (decimal) किंवा BCD कोड:

या कोडमध्ये प्रत्येक दशांश अंक चार बिट बायनरी संख्येद्वारे दर्शविला जातो. बीसीडी कोडमधील बायनरी बिट्सशी संबंधित पोजिशनल वेट्स 8-4-2-1 आहेत ज्यात 1 एलएसबीशी (LSB) संबंधित आहे आणि 8 एमएसबीशी (MSB) संबंधित आहे. हे सामान्यतः डिजिटल डिस्प्ले (digital displays) मध्ये वापरले जातात. 0 ते 9 दशांश संख्यांशी संबंधित BCD संख्या संबंधित बायनरी संख्यांप्रमाणेच असतात. 0 ते 15 दशांश पर्यंत बायनरी आणि BCD संख्यांची तुलना टेबल 1.10 मध्ये दर्शविली आहे.

Table 1.10: Comparison between Binary and BCD code

दशांश (Decimal Number)	बायनरी (Binary Number)	बीसीडी (BCD)
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	0111	0111
8	1000	1000
9	1001	1001
10	1010	0001 0000
11	1011	0001 0001
12	1100	0001 0010
13	1101	0001 0011
14	1110	0001 0100
15	1111	0001 0101

बीसीडी कोड चे फायदे :

1. हे डेसिमल प्रणाली सारखेच आहेत.
2. फक्त 0 ते 9 डेसिमल संख्यांच्या समतुल्य बायनरी नंबर लक्षात ठेवण्याची गरज आहे.

बीसीडी कोडचे तोटे :

1. बीसीडी चे बेरीज आणि वजाबाकीचे वेगवेगळे नियम आहेत.
2. बीसीडीचे अंकगणित अधिक गुंतागुंतीचे असते.
3. डेसिमल संख्या दर्शविण्यासाठी बीसीडी ला जास्त बिट्स लागतात.

उदाहरण :

1) खालील दशांश संख्या BCD मध्ये रूपांतरित करा i) $(39)_{10}$ ii) $(327.89)_{10}$

i) $(39)_{10}$: 3 9 दशांश संख्या

↓ ↓

0011 1001 बीसीडी समतुल्य

$(39)_{10} = (00111001)_{BCD}$

ii) $(327.89)_{10}$ 3 2 7 . 8 9 दशांश संख्या

↓ ↓ ↓ ↓ ↓ ↓

0011 0010 0111 . 1000 1001 बीसीडी समतुल्य

$(327.89)_{10} = (001100100111.10001001)_{BCD}$

1.4.2 ग्रे कोड (Gray Code) :

याला चक्रीय कोड (cyclic code) किंवा परावर्तित बायनरी कोड (reflective binary code) असेही म्हणतात. ग्रे कोड मध्ये एकाच स्टेप वरून दुसऱ्या स्टेप वर जाताना कोड ग्रुप मधील फक्त एक बिट बदलते. ग्रे कोड हा एकक अंतर कोड मध्ये सर्वात लोकप्रिय आहे, परंतु तो अंकगणितीय क्रियांसाठी योग्य नाही. रोटरी (rotary) आणि ऑप्टिकल (optical) एनकोडर मध्ये ग्रे कोड वापरले जातात.

ग्रे कोड ते बायनरी कोड रूपांतरण (Gray code to binary code conversion):

स्टेप 1: ग्रे आणि बायनरी चे MSB समान आहेत. त्यामुळे ते थेट लिहा.

स्टेप 2: ग्रे कोडच्या पुढील बिटमध्ये बायनरी MSB ऍड करा. (MOD-2 addition) जोडा. निकाल लक्षात ठेवा आणि कॅरी (carry) कडे दुर्लक्ष (discard) करा.

स्टेप 3: LSB पोहोचेपर्यंत ही प्रक्रिया सुरू ठेवा.

मॉड टू (Mod-2) एडिशनचे नियम :

$$0 \oplus 0 = 0$$

$$0 \oplus 1 = 1$$

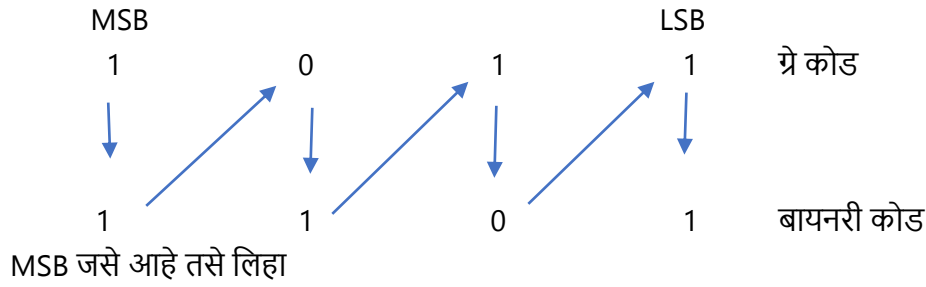
$$1 \oplus 0 = 1$$

$$0 \oplus 1 = 0$$

उदाहरण :

1) 1011 ग्रे कोड ला बायनरी कोडमध्ये रूपांतरित करा.

उत्तर :



$$(1011)_G = (1101)_2$$

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

स्टेप 1: ग्रे आणि बायनरी चे MSB समान आहेत. त्यामुळे ते थेट लिहा.

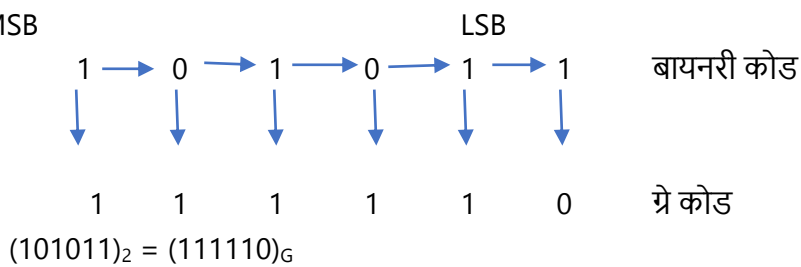
स्टेप 2: हा बिट पुढील बिटमध्ये ऍड करा आणि बेरीज लक्षात ठेवा आणि कॅरी (carry) कडे लक्ष न देता discard करा.

स्टेप 3: स्टेप 2 पुन्हा करा.

उदाहरण:

1) 101011 बायनरी कोड ला ग्रे कोडमध्ये रूपांतरित करा.

उत्तर: MSB



1.4.3 एक्सेस – 3 कोड (Excess-3 or XS-3 code):

हा डेसिमल संख्या व्यक्त करण्यासाठी वापरला जाणारा नॉनवेटेड कोड आहे. एक्सेस-3 मध्ये सांकेतिक शब्द 8421 BCD कोड शब्दापासून 8421 मधील प्रत्येक कोड शब्दांमध्ये $(0011)_2$ किंवा $(3)_{10}$ जोडून तयार केलेले असतात.

डेसिमल संख्या $\xrightarrow{\text{ADD } 0011}$ 8421 BCD $\xrightarrow{\text{ADD } 0011}$ एक्सेस – 3 (XS-3)

Table 1.11: Conversion of Decimal, BCD, Excess -3 code

दशांश (Decimal Number)	बीसीडी (BCD)	एक्सेस थ्री (Excess- 3) BCD + 0011
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111

दशांश (Decimal Number)	बीसीडी (BCD)	एक्सेस थ्री (Excess- 3) BCD + 0011
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

उदाहरण :

1) $(37)_{10}$ साठी एक्सेस-3 कोड मिळवा.

उत्तर : 3 7 डेसिमल संख्या

↓ ↓

0011 0111 बीसीडी

+ 0011 0011 3 ऍड करा

0110 1010 एक्सेस-3 कोड

2) $(246.5)_{10}$ साठी एक्सेस-3 कोड मिळवा.

उत्तर : 2 4 6 . 5 डेसिमल संख्या

↓ ↓ ↓ ↓ ↓

0010 0100 0110 . 0101 बीसीडी

+ 0011 0011 0011 . 0011 3 ऍड करा

0101 0111 1001 . 1000 एक्सेस-3 को

1.4.4 आस्की कोड (ASCII code):

आस्की कोड हा एक 7 बिट कोड आहे आणि तो $2^7 = 128$ संभाव्य वर्णांचे (characters) चे प्रतिनिधित्व करतो. हा कोड सर्व स्टॅंडर्ड कीबोर्ड कॅरेक्टर्स (keyboard characters) तसेच इतर इलेक्ट्रॉनिक इक्विपमेंट इत्यादी सारख्या नियंत्रण कार्यांचे प्रतिनिधित्व करतो. तक्ता 1.12 मध्ये आस्की कोडची यादी दर्शविली आहे. प्रत्येक कॅरेक्टरसाठी कॅरेक्टरच्या स्तंभात पहिले तीन बिट्स (b7,b6,b5) आणि ओळीत उरलेले चार बिट्स (b4,b3,b2,b1) असे सात बायनरी बिट्स दाखविले आहेत. आस्की चा समतुल्य डेसिमल नंबर हा या बायनरी नंबर वरून शोधला जातो. उदाहरणार्थ, K अक्षरासाठी आस्की कोड $(1001011)_2$ आहे.

Table 1.12: ASCII Codes

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

1.4.5 कोडचे अनुप्रयोग (Applications of codes):

1. संगणक आणि इतर डिजिटल सर्किट (digital circuit) बायनरी स्वरूपात डेटा प्रक्रिया करतात. डेटाचे प्रतिनिधित्व करण्यासाठी विविध बायनरी कोड वापरले जातात. डेटाचे स्पष्टीकरण तेव्हाच शक्य आहे जेव्हा कोडमध्ये माहिती उपलब्ध आहे.
2. बीसीडी कोड बायनरी प्रणालींना उत्कृष्ट इंटरफेस (interface) प्रदान करतो, उदाहरणार्थ, कीपॅड इनपुट (keypad input) आणि डिजिटल रीडआउट्स (digital readouts).
3. ग्रे कोड कर्नाफ मॅप (Karnaugh map) मध्ये लिटरल्स प्लॉट (literals plot) करण्यासाठी [संदर्भ: युनिट 3, सबपॉईंट 3.2], सिग्नलमध्ये त्रुटी शोधणे (Error detection in signal), कोन मापन यंत्रे (Angle measuring devices) इत्यादींमध्ये वापरले जाते. तसेच, ग्रे कोड एकाधिक अनुवांशिक अल्गोरिदम (Multiple genetic algorithm) आणि रोटेटरी शाफ्ट (Rotatory shaft) मध्ये वापरले जातात जेथे मूल्ये (values) झपाट्याने बदलत आहेत कारण त्याच्या सलग मूल्यांमध्ये फक्त 1-बिट फरक आहे.
5. एक्सेस-3 कोड दशांश अंकगणित डिजिटल पद्धतीने करण्यासाठी उपयुक्त आहे.
6. आस्की कोड चा वापर हा प्रामुख्याने संगणकामध्ये केला जातो.

1.5 BCD अंकगणित (BCD Arithmetic) :

1.5.1 BCD बेरीज (BCD Addition) :

बीसीडी बेरीज मध्ये तीन भिन्न परिस्थिती आहेत. दोन 4 बिट बीसीडी क्रमांक A आणि B बेरीज केले जात आहेत असे गृहीत धरा. त्यानंतर पुढील केसेस (cases) चा विचार केला जातो.

केस 1 : बेरीज 9 च्या समान किंवा कमी आहे आणि कॅरी शून्य आहे ($\text{Sum} \leq 9, \text{carry} = 0$) मग बेरीज योग्य बीसीडी फॉर्ममध्ये आहे आणि कोणत्याही दुरुस्तीची आवश्यकता नाही.

केस 2 : बेरीज 9 पेक्षा मोठी आहे आणि कॅरी 0 आहे ($\text{Sum} > 9, \text{carry} = 0$) तर आपल्याला दशांश 6 किंवा BCD 0110 ऍड करून बेरीज दुरुस्त करावी लागेल आणि आपल्याला योग्य BCD बेरीज मिळेल.

केस 3 : बेरीज 9 पेक्षा कमी किंवा समान आहे परंतु कॅरी 1 आहे ($\text{Sum} \leq 9, \text{carry} = 1$) जरी संख्या वैध BCD आहे. परंतु अंतिम कॅरी 1 मुळे, उत्तर दुरुस्त करण्यासाठी आपल्याला बेरीजमध्ये दशांश 6 किंवा बीसीडी 0110 ऍड करावे लागेल.

केस 1 : बेरीज 9 च्या समान किंवा कमी आहे आणि कॅरी शून्य आहे ($\text{Sum} \leq 9, \text{carry} = 0$)

उदाहरण :

1) बीसीडी बेरीज करा $(5)_{10} + (3)_{10}$

उत्तर :	डेसिमल	बीसीडी
	5	0 1 0 1
+	3	+ 0 0 1 1
	8	1 0 0 0

बेरीज कॅरी 0 सह वैध BCD संख्या आहे

केस 2: बेरीज 9 पेक्षा मोठी आहे आणि कॅरी 0 आहे ($\text{Sum} > 9, \text{carry} = 0$)

उदाहरण :

1) बीसीडी बेरीज करा $(8)_{10} + (6)_{10}$

उत्तर :	डेसिमल	बीसीडी
	8	1 0 0 0
+	6	+ 0 1 1 0
	14	1 1 1 0

बेरीज कॅरी 0 सह अवैध BCD संख्या आहे म्हणून 6 ची दुरुस्ती आवश्यक आहे

	1 1
	1 1 1 0
	+ 0 1 1 0
	0 1 0 0
अंतिम कॅरी 1	0 1 0 0
0 0 0 1	0 1 0 0
↓	↓
1	4

$$(8)_{10} + (6)_{10} = (14)_{10}$$

केस 3 : बेरीज 9 पेक्षा कमी किंवा समान आहे परंतु कॅरी 1 आहे (Sum <= 9, carry = 1)

उदाहरण :

1) बीसीडी बेरीज करा $(9)_{10} + (8)_{10}$

उत्तर :	डेसिमल	बीसीडी
	9	1 0 0 1
+	8	+ 1 0 0 0
17 अंतिम कॅरी 1 0 0 0 1		

बेरीज वैध BCD संख्या आहे परंतु कॅरी 1 आहे, म्हणून 6 ची दुरुस्ती आवश्यक आहे

0 0 0 1	0 0 0 1
+	0 0 0 0
0 0 0 1 0 1 1 1	
1	7

$$(8)_{10} + (6)_{10} = (17)_{10}$$

1.5.2 9 चा आणि 10 चा पूरक वापरून वजाबाकी (Subtraction using 9's and 10's complement method):

जर संख्या बायनरी असेल तर आपण 1 ची पूरक आणि 2 ची पूरक वापरतो परंतु जर संख्या दशांश संख्या असेल तर आपण 9 ची पूरक आणि 10 ची पूरक वापरू. 10 ची पूरक संख्या 9 च्या पूरक पासून प्राप्त होते आणि आम्ही r आणि $(r-1)$ च्या पूरक सूत्राचा वापर करून 9 ची पूरक आणि 10 ची पूरक देखील शोधू शकतो.

1.5.2.1 9 चा पूरक (9's complement) :

दशांश संख्यांची वजाबाकी शोधण्यासाठी 9 ची पूरकता वापरली जाते. संख्येचा प्रत्येक अंक 9 ने वजा करून संख्येची 9 ची पूरक गणना केली जाते. उदाहरणार्थ, समजा आपल्याकडे 1423 संख्या आहे आणि आपल्याला त्या संख्येची 9 ची पूरकता शोधायची आहे. त्यासाठी आपण 1423 या संख्येचा प्रत्येक अंक 9 ने वजा करू. तर 1423 या संख्येची 9 ची पूरकता $9999 - 1423 = 8576$ आहे.

9 चा पूरक वापरून वजाबाकी (Subtraction using 9's complement) :

स्टेप्स :

स्टेप 1 : आपल्याला प्रथम सबट्राहेंडची 9 ची पूरकता शोधावी लागेल. मग आपण हे पूरक मूल्य minuend मध्ये ऍड करू.

उदाहरण :

1) $(841)_{10} - (329)_{10}$ 9 चा पूरक वापरून वजाबाकी करा.

उत्तर :	3	2	9	डेसिमल
	6	7	0	9 चा पूरक

1

$$\begin{array}{r}
 9 \text{ चा पूरक वापरून वजाबाकी : } 8 \quad 4 \quad 1 \\
 + \quad 6 \quad 7 \quad 0 \quad (329)_{10} \text{ चा } 9 \text{ चा पूरक} \\
 \hline
 1 \quad 5 \quad 1 \quad 1 \\
 + \quad \quad \quad 1 \\
 \hline
 5 \quad 1 \quad 2
 \end{array}$$

$$(841)_{10} - (329)_{10} = (512)_{10}$$

1.5.2.2 10 चा पूरक (10's complement) :

10 च्या पूरक संख्येची गणना त्याच्या 9 च्या पूरकतेमध्ये 1 ऍड करून केली जाते, आपण त्याचे 10 चे पूरक मूल्य मिळवू शकतो. उदाहरणार्थ, समजा आपल्याकडे 1423 क्रमांक आहे आणि आपल्याला त्या संख्येची 10 ची पूरकता शोधायची आहे. यासाठी, $9999 - 1423 = 8576$ या संख्येच्या 1423 चा 9 ची पूरकता सापडली आणि आता आपण निकालात 1 ऍड करू. तर 1423 या संख्येची 10 ची पूरकता $8576 + 1 = 8577$ आहे.

10 चा पूरक वापरून वजाबाकी (Subtraction using 10's complement) :

स्टेप्स :

स्टेप 1 : प्रथम आपल्याला सबट्राहेंडची 10 ची पूरकता शोधावी लागेल. मग आपण हे पूरक मूल्य minuend मध्ये ऍड करू.

उदाहरण :

1) $(821)_{10} - (413)_{10}$ 10 चा पूरक वापरून वजाबाकी करा.

$$\begin{array}{r}
 \text{उत्तर : } \quad 4 \quad 1 \quad 3 \quad \text{डेसिमल} \\
 \quad \quad 5 \quad 8 \quad 6 \quad 9 \text{ चा पूरक} \\
 \quad \quad \quad + \quad 1 \\
 \hline
 \quad \quad 5 \quad 8 \quad 7 \quad 10 \text{ चा पूरक}
 \end{array}$$

1

$$\begin{array}{r}
 10 \text{ चा पूरक वापरून वजाबाकी : } 8 \quad 2 \quad 1 \\
 + \quad 5 \quad 8 \quad 7 \quad (329)_{10} \text{ चा } 9 \text{ चा पूरक} \\
 \hline
 1 \quad 4 \quad 0 \quad 8 \quad \text{कॅरी दुर्लक्ष करा} \\
 \hline
 \quad \quad 4 \quad 0 \quad 8
 \end{array}$$

$$(821)_{10} - (413)_{10} = (408)_{10}$$

स्वयंअध्ययन (Self learning) :

- 1) हेक्साडेसिमल नंबर सिस्टम्स उदाहरणासह स्पष्ट करा.
- 2) बायनरी नंबर $(1011.01)_2$ चे डेसिमल समतुल्य मध्ये रूपांतरित करा.
- 3) रूपांतरित करा $(268.75)_{10} = (?)_2$.
- 4) सोडवा 101101×110 .

- 5) 2 चे पूरक पद्धती वापरून वजाबाकी करा $(48)_{10} - (68)_{10}$.
- 6) खालील BCD अंकगणित ऑपरेशन करा $(245)_{10} + (186)_{10}$.
- 7) खालील संख्या प्रणालींचा बेस लिहा: दशांश, बायनरी, ऑक्टल, हेक्साडेसिमल .
- 8) कोडचे कोणतेही दोन अनुप्रयोग द्या.
- 9) बायनरी संख्येचे 1 आणि 2 च्या पूरक ची उदाहरणासह व्याख्या करा.

स्व-शैक्षणिक क्रियाकलाप - Self Learning Activities (SLA):

लघु प्रकल्प (Micro project):

- 1) बायनरी, बीसीडी, दशांश रूपांतरणांशी संबंधित कार्यरत मॉडेल तयार करणे.
- 2) बायनरी अंकगणित क्रियांच्या विविध नियमांवर तक्ते तयार करणे.
- 3) बीसीडी संख्यांचे एक्झाडेसिमल संख्या मध्ये रूपांतरण करणाऱ्या एकात्मिक सर्किट (IC) चा अभ्यास करा.

संदर्भ (References) :

Sr No	Author	Title	Publisher with ISBN Number
1.	R.P. Jain	Modern Digital Electronics	Mc Graw-Hill Publishing, New Delhi, 2009 ISBN:9780070669116
2.	Donald P Leach , Albert Paul Malvino, Goutam Saha	Digital principles and applications	Tata McGraw Hill Education Private limited. ISBN 978-0-07-014170-4

संकेतस्थळे (websites) :

<https://studytronics.weebly.com/digital-electronics.html> - Basics of Digital Electronics

https://onlinecourses.nptel.ac.in/noc24_ee147/preview - Digital Circuits

<https://www.javatpoint.com/number-system-in-digital-electronics> - Javatpoint-free learning platform

यूनिट 2: लॉजिक गेट्स व बूलीयन बीजगणित

Logic Gates and Boolean Algebra

कोर्स निष्पत्ती (Course Outcome):

जटिलबूलीयन समीकरणे सुलभ करण्यासाठी बूलीयन कायदे लागू करणे.

Apply Boolean laws to minimize complex Boolean functions

यूनिट निष्पत्ती (Unit Outcomes):

2.1 डिजिटल लॉजिक फॅमिलीच्या कार्यक्षमता मापदंड (performance parameters) परिभाषित करणे.

Define the given characteristics parameters of the digital logic families.

2.2 लॉजिक गेट्स चे चिन्ह आणि तृथ टेबल काढणे.

Draw symbol and truth table of given logic gates.

2.3 बफरची संकल्पना आणि ट्रायस्टेट लॉजिक (logic) स्पष्ट करणे.

Explain the concept of Buffer and Tristate logic .

2.4 यूनिवर्सल गेट्स च्या मदतीने बेसिक गेट्स आणि इतर गेट्स अंमल करणे.

Implement basic gates and other gates with the help of universal gate.

2.5 बूलीयन नियम वापरून दिलेली समीकरणे सुलभ करणे आणि लॉजिक सर्किट्स विकसित करणे.

Simplify the given expression using Boolean laws and develop logic circuits.

परिचय

लॉजिक गेट्स आणि फॅमिलीच्या शोधाचे श्रेय डिजिटल अनुविधुतमधील विविध प्रवर्तकांना दिले जाते. क्लॉड शॅननने बूलीयन बीजगणितचा सैद्धांतिक पाया दिला. 1947 मध्ये वॉल्टर हाऊसर ब्रॅटन, जॉन बार्डीन आणि विल्यम शॉकले यांनी शोधलेल्या ट्रान्झिस्टरने इलेक्ट्रॉनिक लॉजिक गेट्सच्या विकासास मदत केली. TTL आणि CMOS सारखी विशिष्ट फॅमिली 1960 च्या दशकात टेक्सास इन्स्ट्रुमेंट्स सारख्या कंपन्यांच्या प्रयत्नातून उदयास आली.

लॉजिक गेट्स हे डिजिटल इलेक्ट्रॉनिक्समधील मूलभूत घटक आहेत जे बूलीयन लॉजिक ऑपरेशन्स करतात. त्यामध्ये AND, OR, NOT, NAND, NOR, Ex-OR आणि Ex-NOR गेट्स समाविष्ट आहेत. लॉजिक फॅमिलीज अशा लॉजिक गेट्सच्या गटांना म्हणतात ज्यामध्ये समान वैशिष्ट्ये असतात जसे की विद्युत दाब लेवल आणि गती. नेहमीच वापरल्या जाणाऱ्या लॉजिक फॅमिलीजमध्ये TTL, CMOS, ECL, आणि I²L यांचा समावेश आहे. TTL सुमारे 5V विद्युत दाबवर चालते आणि सर्वात जुन्या फॅमिली पैकी एक आहे. CMOS गेट्स कमी वीजशक्ति वापरासाठी ओळखले जातात. ECL आणि I²L ही गतिमान फॅमिली आहेत. लॉजिक फॅमिली निवडणे हे विजेचा वापर, वेग आणि अनुपयोगासाठी आवश्यक विद्युत दाब पातळी यासारख्या घटकांवर अवलंबून असते.

2.1 लॉजिक फॅमिली (Logic Family):

अनेक जटिल डिजिटल फंक्शन्स विविध स्वरूपात मध्ये साकारल्या जाऊ शकतात आणि प्रत्येक फॉर्मला लॉजिक फॅमिली म्हणतात. डिजिटल IC बायपोलर लॉजिक फॅमिली (Bipolar Logic family-BJT's) किंवा युनिपोलर लॉजिक फॅमिली (Unipolar Logic family- FET's) वापरून बनवलेले आहेत.

एक सुसंगत IC चा गट जो समान तर्क (लॉजिक) पातळी आणि सप्लाय विद्युत दाब सह लॉजिक फंक्शन्स एका विशिष्ट सर्किट कॉन्फिगरेशनचा वापर करून तयार केले गेले आहेत ज्याला लॉजिक फॅमिली म्हणून संदर्भित केले जाते.

बायपोलर लॉजिक फॅमिली:

बायपोलर आयसीचे मुख्य घटक म्हणजे रेझिस्टर, डायोड्स आणि ट्रान्झिस्टर आहेत. ते एकतर सॅच्युरेशन (saturation) किंवा नॉन-सॅच्युरेशन (non-saturation) विभागामध्ये (कार्य करतात).

सॅच्युरेशन तर्कशास्त्रात, ट्रान्झिस्टर सॅच्युरेशनकडे कार्याणवित केले जातात.

उदा. RTL, DCTL, I²L, DTL, HTL, TTL.

नॉन सॅच्युरेशन तर्कशास्त्रातामध्ये, ते सॅच्युरेशनकडे कार्याणवित केले जातात.

उदा. Schottky TTL आणि ECL.

युनिपोलर तर्कशास्त्रात फॅमिली:

MOS घटक (MOS components) युनिपोलर असतात. MOS तर्कशास्त्रात फॅमिली मध्ये PMOS, NMOS आणि CMOS आहेत.

2.1.1 डिजिटल IC ची वैशिष्ट्ये:

1. प्रोपॅगेशन डीले (propagation delay) : इनपुट लागू केल्यावर आउटपुट प्राप्त करण्यासाठी लागणारा वेळ त्याला प्रोपॅगेशन डीले (Propagation Delay) परिभाषित केला जातो. ते नॅनो सेकंदात दिले जाते. ($1 \text{ ns} = 10^{-9}$ सेकंद).

लॉजिक गेटचे इनपुट आणि आउटपुट वेवफॉर्म आकृति २.१ दाखवल्या प्रमाणे आहेत:

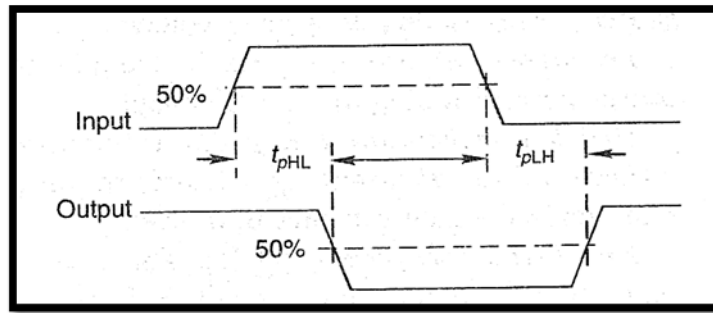


Fig 2.1 - Propagation Delay

डीले वेळ इनपुट आणि आउटपुट वेव फॉर्मच्या 50% विद्युत दाब पातळी दरम्यान मोजला जातो. दोन डीले वेळा असतात:

- जेव्हा आउटपुट उच्च (हाय) वरून निम्नाकडे (लो) जाते [t_{pHL}] आणि
- जेव्हा ते निम्न (लो) वरून उच्च (हाय) [t_{pLH}] वर जाते तेव्हा.

लॉजिक गेटचा प्रोपॅगेशन डीले (propagation delay) वेळ या दोन्ही डीले वेळेची सरासरीची गणना केला जाते.

$$\text{propagation delay} = \frac{t_{pHL} + t_{pLH}}{2}$$

2. शक्ति अपव्यय (Power dissipation-mw): शक्ति अपव्यय ही IC करिता विसर्जित होणारी शक्ती आहे. IC निवडण्यासाठी, फिगर ऑफ मेरिट विचारात घेतला जातो. शक्ति अपव्यय हे प्रसार विलंब आणि पॉवर चे गुणाकार करून मोजले जातात.

3. फॅन-आउट (Fan Out): फॅन-आउट म्हणजे गेटद्वारे चालविल्या जाऊ शकणाऱ्या समान गेट्सची संख्या. जास्त फॅन आउट असणे चांगले आहे कारण ते अधिक गेट्स चालविण्यासाठी अतिरिक्त ड्रायव्हर्सची आवश्यकता कमी करते.

4. फॅन-इन (Fan In): फॅन-इन गेटच्या इनपुटची संख्या म्हणून परिभाषित केले जाते.

उदा: दोन इनपुट गेट चा फॅन-इन दोन असेल.

5. विद्युत प्रवाह आणि विद्युत दाब पैरामीटर्स (Current and Voltage parameters):

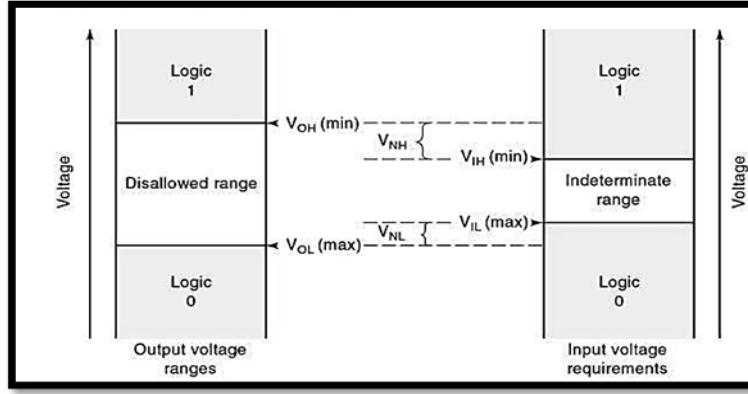


Fig 2.2: Current and Voltage parameters

- हाय लेवल इनपुट विद्युत दाब (V_{IH}):- V_{IH} किमान इनपुट विद्युत दाब जे लॉजिक '1' म्हणून गेटद्वारे ओळखले जाते.
- लो लेवल इनपुट विद्युत दाब (V_{IL}):- V_{IL} कमाल इनपुट विद्युत दाब जे लॉजिक '0' म्हणून गेटद्वारे ओळखले जाते.
- हाय लेवल आउटपुट विद्युत दाब (V_{OH}):- V_{OH} किमान आउटपुट विद्युत दाब जे लॉजिक '1' म्हणून गेटद्वारे ओळखले जाते.
- लो लेवल आउटपुट विद्युत दाब (V_{OL}):- V_{OL} कमाल आउटपुट विद्युत दाब जे लॉजिक '0' म्हणून गेटद्वारे ओळखले जाते.
- उच्च स्तरीय I/P विद्युत प्रवाह (I_{IH}):- I_{IH} हा इनपुटवर उच्च-स्तरीय विद्युत दाब लागू केल्यावर इनपुटमध्ये प्रवाहित होणारा प्रवाह.
- लो लेव्हल I/P विद्युत प्रवाह (I_{IL}):- I_{IL} हा इनपुटवर लो-लेव्हल विद्युत दाब लागू केला जातो तेव्हा इनपुटमध्ये प्रवाहित होणारा प्रवाह.
- उच्च स्तरीय O/P विद्युत प्रवाह (I_{OH}):- I_{OH} हा लोड स्थितीत मध्ये लॉजिक 1 स्थितीत आउटपुटमधून प्रवाहित होणारा प्रवाह
- लो लेव्हल O/P विद्युत प्रवाह (I_{OL}):- I_{OL} हा लोड स्थितीत लॉजिक 0 स्थितीत आउटपुटमधून वाहणारा प्रवाह.

6. नॉइज प्रतिकारशक्ती (Noise Immunity): इनपुट आणि आउटपुट विद्युत दाब लेव्हलसमध्ये काही अवांछित (unwanted) विद्युत दाब, विद्युत आणि चुंबकीय क्षेत्रामुळे प्रेरित होऊ शकतात, ज्याला नॉइज म्हणतात. यामुळे लॉजिक सर्किटच्या इनपुटवरील विद्युत दाब V_{IH} च्या खाली जाऊ शकते किंवा V_{IL} च्या वर जाऊ शकते आणि अवांछित (undesirable) ऑपरेशन होऊ शकते. हे नॉइज संदेश सहन करण्याच्या सर्किटच्या क्षमतेला नॉइज प्रतिकारशक्ती असे म्हणतात.

7. विद्युत प्रवाह सोर्सिंग आणि सिंकिंग (Current Sourcing and Sinking): विद्युत प्रवाह सोर्सिंग आणि सिंकिंग क्रिया आकृति 2.3 दाखवल्या आहे. गेट 1 स्त्रोत म्हणून कार्य करते आणि गेट 2 लोड म्हणून कार्य करते.

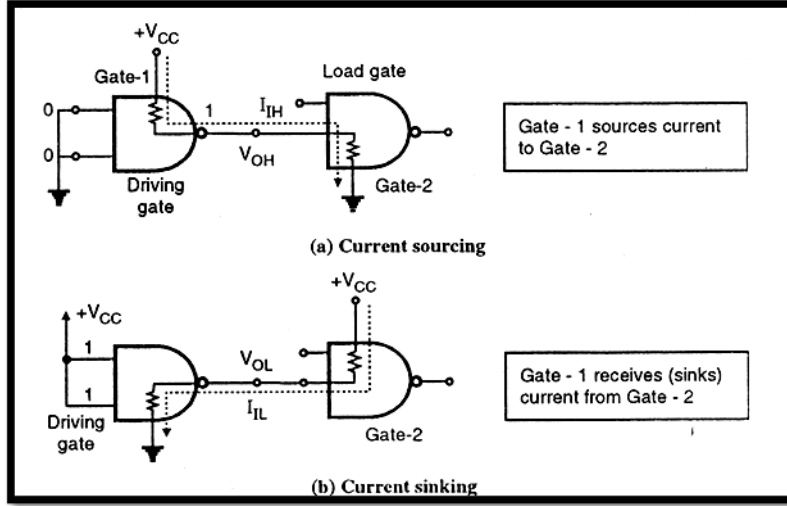


Fig 2.3: Current Sourcing and Sinking

- गेट 1 चे आउटपुट जास्त आहे. हे गेट 2 च्या इनपुटला विद्युत प्रवाह प्रवाह पुरवते. याला विद्युत प्रवाह सोर्सिंग क्रिया म्हणून ओळखले जाते.
- गेट 1 चे आउटपुट कमी झाल्यावर, आकृति 2.3 मध्ये दाखवल्याप्रमाणे गेट 1 च्या आउटपुट टर्मिनलमध्ये विद्युत प्रवाह वाहू लागतो. अशाप्रकारे जेव्हा गेट 1 त्याच्या आउटपुट टर्मिनलद्वारे विद्युत प्रवाह स्वीकारत असतो तेव्हा ते विद्युत प्रवाह सिंकिंग असल्याचे म्हटले जाते.

8. सप्लाय विद्युत दाब : TTL ला 4.75 V आणि 5.25 V च्या दरम्यान सप्लाय विद्युत दाब आवश्यक आहे.

9. क्लॉक फ्रिक्वेन्सी (frequency): डिजिटल सर्किट ज्या कमाल क्लॉकच्या वारंवारतेवर कार्य करू शकते त्याला f_{MAX} म्हणतात.

TTL, ECL आणि CMOS मधील तुलना:

Table 2.1: Comparison between TTL, ECL and CMOS

अनु. क्र.	पॅरामीटर्स	TTL	ECL	CMOS
1	बेसिक गेट्स	NAND	OR- NOR	OR or NAND
2	फॅन-इन	8	5	10
3	फॅन-आउट	10	25	>50
4	शक्ति अपव्यय प्रति गेट	10mW	50mW	0.01mW
5	नॉइज मार्जिन (प्रतिकारशक्ती)	0.4V good	0.25V (poor)	5V (excellent)
6	प्रसारविलंब (propagation delay)	10 ns	2 ns	70 ns
7	स्पीड-पॉवर प्रॉडक्ट	100	100	0.7
8	फ्लिप-फ्लॉप क्लॉकदर	35 MHz	60 MHz	10 MHz
9	पॅकिंग घनता (density)	Lower	Lower	Larger

2.2 पॉजिटिव आणि नेगेटिव लॉजिक प्रणालींचा परिचय, गेट्स: चिन्ह , द्रुथ टेबल: बेसिक गेट्स (AND,OR,NOT), युनिव्हर्सल गेट्स (NAND,NOR) आणि स्पेशल गेट्स (EX-OR, EX-NOR)

पॉजिटिव आणि नेगेटिव लॉजिक प्रणाली (Positive and Negative Logic System):

- डिजिटल इलेक्ट्रॉनिक प्रणालींमध्ये, जर सिग्नलचे हाय वॉल्यू (विद्युत दाब किंवा विद्युत प्रवाह) लॉजिक 1 चे प्रतिनिधित्व करण्यासाठी वापरले जाते आणि सिग्नलचे लो वॉल्यू (विद्युत दाब किंवा विद्युत प्रवाह) लॉजिक 0 चे प्रतिनिधित्व करण्यासाठी वापरले जाते, तर त्याला पॉजिटिव लॉजिक म्हणतात.

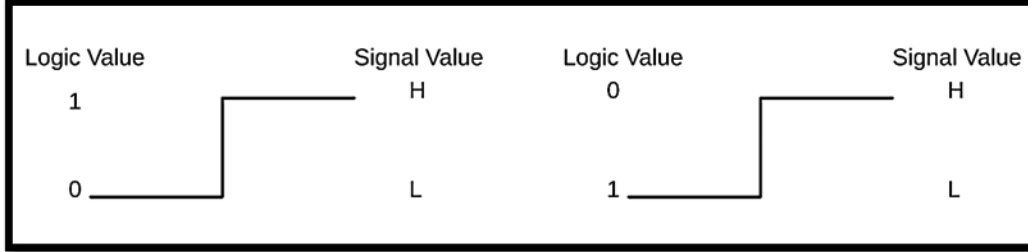


Fig2.4 : Positive and Negative Logic Systems

- डिजिटल इलेक्ट्रॉनिक सिस्टीममध्ये, जर सिग्नलचे हाय वॉल्यू (विद्युत दाब किंवा विद्युत प्रवाह) लॉजिक 0 चे प्रतिनिधित्व करण्यासाठी वापरले जाते आणि लॉजिक 1 चे प्रतिनिधित्व करण्यासाठी सिग्नलचे लो वॉल्यू (विद्युत दाब किंवा विद्युत प्रवाह) वापरले जाते, तर त्याला नेगेटिव लॉजिक प्रणाली म्हणतात.

पॉजिटिव आणि नेगेटिव लॉजिक प्रणाली ह्यांच्यातील निवड नॉइज मार्जिन, डिझाइनची जटिलता आणि उर्जेचा वापर प्रभावित करू शकते. कार्यक्षम सर्किट रचना आणि विविध घटक इंटरफेस करण्यासाठी दोन्ही समजून घेणे आवश्यक आहे.

लॉजिक गेट्स:

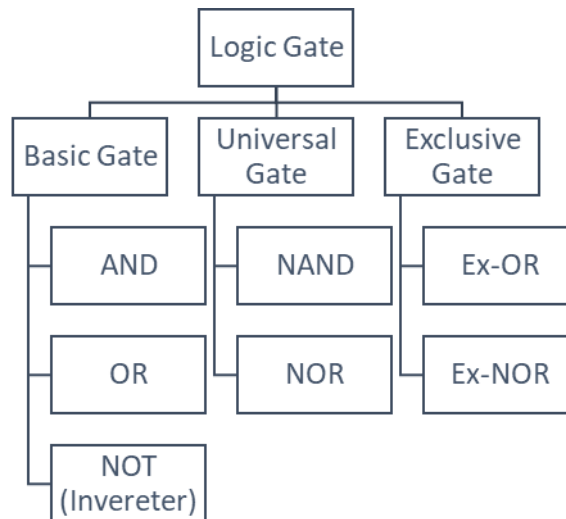


Fig 2.5: Logic Gates Classification

बेसिक गेट्स:

AND गेट : AND ऑपरेशन करणारे सर्किट आकृति 2.6 दाखवल्याप्रमाणे आहे. ज्यात दोन इनपुट आणि फक्त एक आउटपुट आहे. अश्याप्रकारे 'N' इनपुट AND गेट देखील असतात.

चिन्ह (symbol):

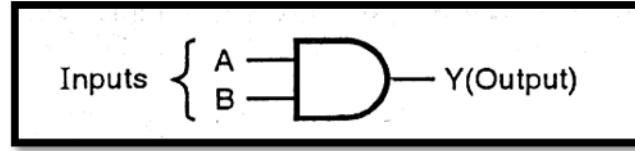


Fig 2.6 - AND Gate Symbol

A, B म्हणून चिन्हांकित केलेल्या इनपुट टर्मिनलवर डिजिटल सिग्नल लागू केले जातात आणि Y म्हणून चिन्हांकित टर्मिनलवर आउटपुट प्राप्त केले जाते.

बुलीयन समीकरण (Boolean expression):

$$Y = A \text{ AND } B = A.B$$

तथ टेबल (Truth Table):

Table 2.2 : AND Gate Truth Table

Input		Output
A	B	$Y=A.B$
0	0	0
0	1	0
1	0	0
1	1	1

AND ऑपरेशनची व्याख्या "जर सगळे इनपुट उच्च (logic 1) असतील AND गेटचे आउटपुट उच्च असेल."

OR गेट :

OR ऑपरेशन करणारे सर्किट आकृति 2.7 मध्ये दाखवल्याप्रमाणे आहे. ज्यात दोन इनपुट आणि फक्त एक आउटपुट आहे. अश्याप्रकारे 'n' इनपुट OR गेट देखील असेल

चिन्ह (symbol):

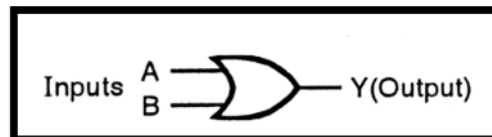


Fig 2.7 -OR Gate Symbol

A, B म्हणून चिन्हांकित केलेल्या इनपुट टर्मिनलवर डिजिटल सिग्नल लागू केले जातात आणि Y म्हणून चिन्हांकित टर्मिनलवर आउटपुट प्राप्त केले जाते.

बुलीयन समीकरण (Boolean expression):

$$Y = A \text{ OR } B = A + B$$

तथ टेबल (Truth Table):

Table 2.3: OR Gate Truth Table

Input		Output
A	B	$Y=A+B$
0	0	0
0	1	1
1	0	1
1	1	1

OR ऑपरेशनची व्याख्या जर एक किंवा अधिक इनपुट उच्च असतील तर OR गेटचे आउटपुट पण उच्च (लॉजिक 1) असेल.

NOT गेट: NOT ऑपरेशन इन्व्हर्टर म्हणून ओळखले जाते.

चिन्ह (symbol):

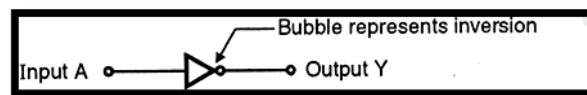


Fig 2.8 - NOT Gate

यात एक इनपुट A आणि एक आउटपुट Y आहे.

बुलीयन समीकरण (Boolean expression):

$$Y = \text{NOT } A = \bar{A}$$

आणि "Y हा A चा पूरक आहे." ["Y equals NOT A" किंवा "Y equals complement of A"] असे वाचले जाते.

तथ टेबल :

Table 2.4 : NOT Gate Truth Table

Input	Output
A	Y
0	1
1	0

युनिव्हर्सल गेट्स:

AND, OR आणि NOT गेट्स वापरून कोणतीही बुलीयन किंवा लॉजिक एक्सप्रेसन्स साकारता येतात या तीन ऑपरेशन्समधून आणखी दोन ऑपरेशन्स जसे की NAND आणि NOR ऑपरेशन्स प्राप्त होतात. हे गेट्स मोठ्या प्रमाणावर वापरले जातात कारण NAND किंवा NOR गेट वापरून कोणतीही लॉजिकल एक्सप्रेसन (logical expression) साकारता येते. त्यामुळे या गेट्स ना युनिव्हर्सल गेट्स असे म्हणतात.

NAND गेट: NOT- AND ऑपरेशनला NAND गेट म्हणून ओळखले जाते. ह्या गेट मध्ये कमीत कमी दोन इनपुट आणि एक आउटपुट असतात. त्यात AND गेट आधी आणि त्यानंतर NOT गेट आहे. ह्या गेट ला 'n' इनपुट देखील असू शकतात.

चिन्ह (symbol):

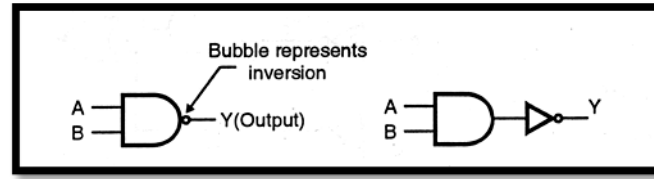


Fig 2.9 - NAND Gate Symbol

A, B इनपुट व्हेरिएबल्स आणि Y हे आउटपुट व्हेरिएबल आहे.

बुलीयन समीकरण (Boolean expression):

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

तृथ टेबल :

Table 2.5: NAND Gate Truth Table

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

NOR गेट: NOT-OR ऑपरेशनला NOR गेट म्हणून ओळखले जाते. ह्या गेट मध्ये कमीत कमी दोन इनपुट आणि एक आउटपुट असतात. त्यात OR गेट आधी आणि त्यानंतर NOT गेट असते. ह्या गेटला 'n' इनपुट असू शकतात.

चिन्ह (symbol):

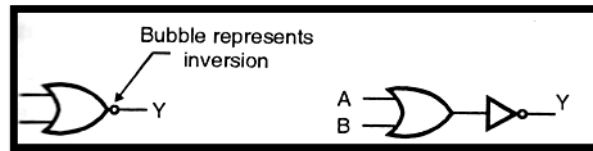


Fig 2.10 - NOR Gate

A, B हे इनपुट व्हेरिएबल्स आणि Y हे आउटपुट व्हेरिएबल आहेत

बुलीयन समीकरण (Boolean expression):

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

तृथ टेबल:

Table 2.6: NOR Gate Truth Table

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

एक्सक्लूझिव (Exclusive) गेट्स:

AND, OR, NOT, NAND आणि NOR गेट व्यतिरिक्त, दोन विशेष गेट्स आहेत, ते म्हणजे, XOR आणि XNOR. त्यांचे बूलीयन आउटपुट फंक्शन लॉजिक गेट म्हणून मानले जाण्यासाठी पुरेसे आहे. XOR आणि XNOR गेट्स हे हायब्रीड गेट्स आहेत.

Ex-OR गेट:

डिजिटल इलेक्ट्रॉनिक्समध्ये, XOR गेट नावाचे लॉजिक सर्किट आहे जे दोन इनपुट घेते आणि आउटपुट तयार करतो. जे इनपुटच्या संयोजनावर केलेल्या एक्सक्लूझिव ऑर (Exclusive-OR) ऑपरेशनचे परिणाम आहे.

चिन्ह (symbol):

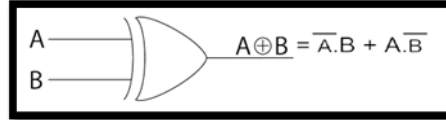


Fig 2.11- XOR Gate

A, B हे इनपुट व्हेरिएबल्स आणि Y हे आउटपुट व्हेरिएबल आहेत.

बुलीयन समीकरण (Boolean expression):

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

तथ टेबल :

Table 2.7: XOR Gate Truth Table

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

Ex-NOR gate :

XNOR गेट हे XOR गेटचे पूरक (complement) आहे. हे एक हायब्रीड गेट आहे. हे XOR गेट आणि NOT गेटचे संयोजन आहे. XNOR गेट (Exclusive NOR गेट), ज्याला समतुल्य गेट म्हणून देखील ओळखले जाते, हे डिजिटल लॉजिक गेट आहे जे हाय किंवा उच्च सिग्नल प्रदान करते जेव्हा उच्च इनपुटची संख्या समान असते.

चिन्ह (symbol):

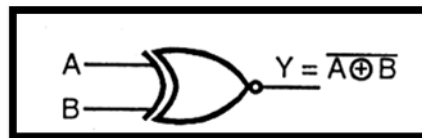


Fig 2.12: XNOR Gate

बुलीयन समीकरण (Boolean expression):

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

तृथ टेबल:

Table 2.8 : XNOR Gate Truth Table

Input A	Input B	Output
0	0	1
0	1	0
1	0	0
1	1	1

NOR आणि NAND गेट्स चा युनिव्हर्सल गेट म्हणून वापर:

युनिव्हर्सल गेट हे असे लॉजिक गेट आहे जे इतर कोणत्याही प्रकारच्या लॉजिक गेट न वापरता कोणतेही बूलीयन फंक्शन लागू करू शकते. NOR गेट आणि NAND गेट हे युनिव्हर्सल गेट आहेत. ह्याचा अर्थ असा की फक्त NOR गेट किंवा NAND गेट्स वापरून कोणतीही लॉजिकल बूलीयन एक्सप्रेसन तयार करू शकतो.

NAND गेटचा युनिव्हर्सल गेट म्हणून वापर :

a) NOT गेट

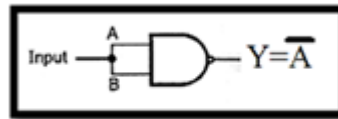


Fig 2.13- NAND Gate as NOT

b) AND गेट

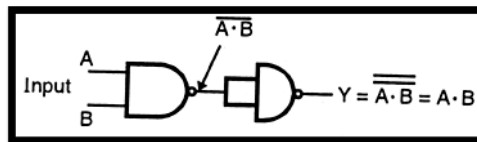


Fig 2.14: NAND Gate as AND

c) OR गेट

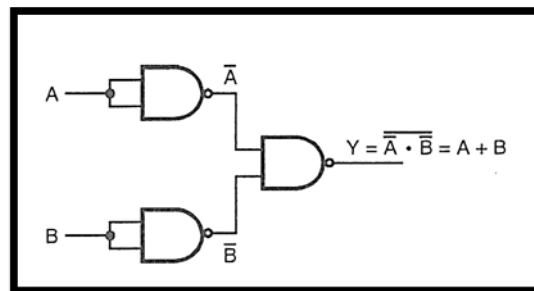


Fig 2.15- NAND Gate as OR

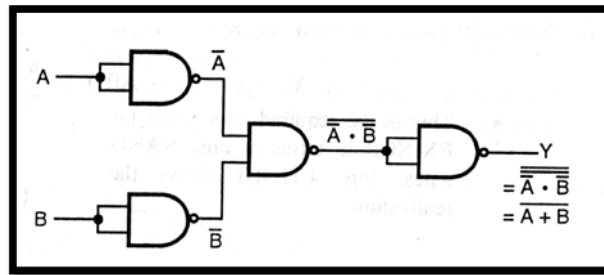
d) NOR गेट

Fig 2.16: NAND Gate as NOR

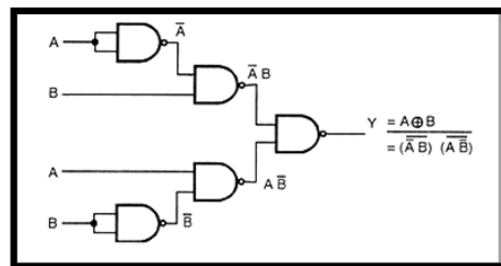
e) XOR गेट

Fig 2.17- NAND Gate as XOR

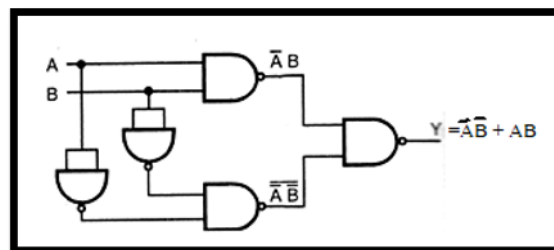
f) XNOR गेट

Fig 2.18 - NAND Gate as XNOR

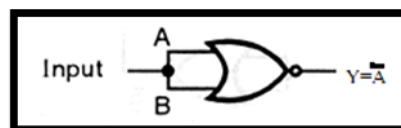
NOR गेट युनिव्हर्सल गेट**a) NOT गेट**

Fig 2.19: NOR Gate as NOT

b) OR गेट

Fig 2.20- NOR Gate as OR

c) AND गेट

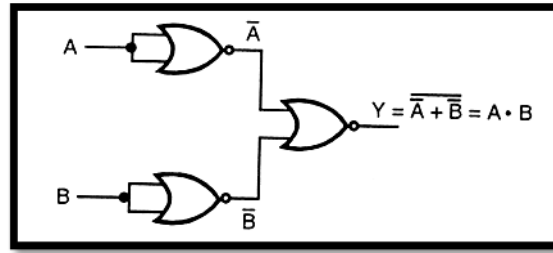


Fig 2.21- NOR Gate as AND

a) NAND गेट

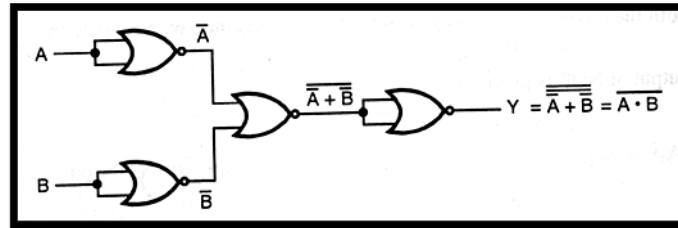


Fig 2.22- NOR Gate as NAND

b) XOR गेट

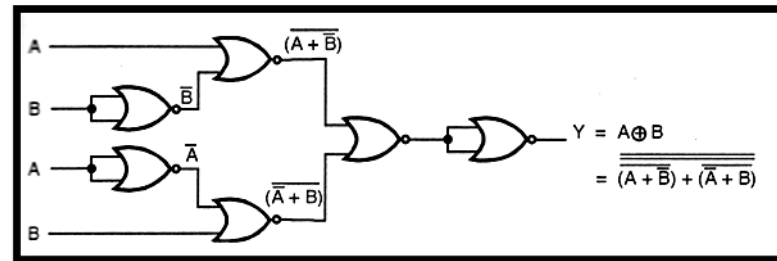


Fig 2.23- NOR Gate as XOR

g) XNOR गेट

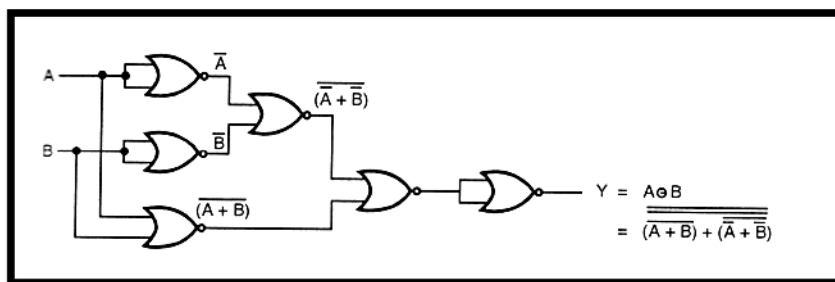


Fig 2.24 - NOR Gate as XNOR

2.3 बफर: ट्रायस्टेट लॉजिक, एकदिशीय आणि द्विदिशात्मक (Unidirectional and Bidirectional)

बफर

बफर हे एक गेट किंवा सर्किट आहे जे तात्पुरते डेटा संग्रहित करते आणि दोन किंवा अधिक घटकांमध्ये ते डेटा हस्तांतरित करण्यात मदत करते.

चिन्ह (symbol):

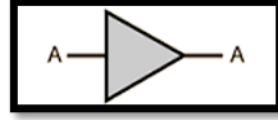


Fig 2.25 – Buffer

तृथ टेबल :

Table 2.9 : Truth Table of Buffer

Input	Output
0	0
1	1

डेटा प्रवाह व्यवस्थापित करण्यासाठी, प्रणालीच्या विविध भागांमध्ये सुरळीत आणि कार्यक्षम संप्रेषण सुनिश्चित करण्यासाठी बफर सामान्यतः डिजिटल इलेक्ट्रॉनिक्समध्ये वापरले जातात. ते सर्किट एलिमेंट्स वेगळे करून आणि डेटा लॉस रोखून संदेशाची अखंडता सुधारण्यात मदत करू शकतात. एकूणच, अणूवीधूत प्रणाली ची कार्यक्षमता आणि विश्वासार्हता अनुकूल (optimise) करण्यात बफर महत्त्वपूर्ण भूमिका बजावतात.

ट्रायस्टेट लॉजिक (Tristate Logic):

ट्रायस्टेट लॉजिक, ज्याला थ्री-स्टेट लॉजिक असेही म्हणतात, लॉजिक सर्किटचा एक प्रकार आहे ज्यामध्ये तीन संभाव्य आउटपुट अवस्था आहेत:

- हाय (लॉजिक 1),
- लो (लॉजिक 0), आणि
- उच्च प्रतिबाधा (High impedance-Z).

उच्च प्रतिबाधा स्थिती सर्किटमधून आउटपुट प्रभावीपणे डिस्कनेक्ट करते, एकाधिक डिव्हाइसेसना एकमेकांमध्ये हस्तक्षेप न करता एक सामान्य बस सामायिक करण्यास अनुमती देते.

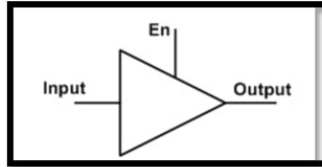


Fig 2.26- Tristate Buffer

तृथ टेबल :

Table 2.10 : Tristate Buffer Truth Table

Enable	A	B
0	0	Z
0	1	Z
1	0	0
1	1	1

ट्रायस्टेट तर्कशस्त्रामध्ये, आउटपुट उच्च, कमी किंवा उच्च प्रतिबाधा स्थितीत आहे की नाही हे नियंत्रण संदेश निर्धारित करते. हे विशेषतः डिजिटल प्रणाली मध्ये उपयुक्त आहे जेथे अनेक उपकरणांना सार्वजनिक बसवर डेटा हस्तांतरण (transfer) करणे

आवश्यक आहे. निष्क्रिय उपकरणे उच्च प्रतिबाधा स्थितीत ठेवून, आपण त्यांना सक्रिय उपकरणांमधील डेटा हस्तांतरण मध्ये हस्तक्षेप करण्यापासून रोखू शकता.

एकदिशीय बफर(Unidirectional buffer):

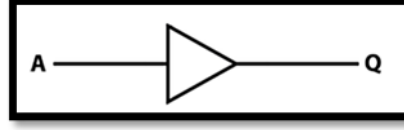


Fig 2.27 - Unidirectional Buffer

एकदिशीय बफर केवळ एका दिशेने संदेश प्रसारित करतो. हा इनपुट संदेश घेतो आणि समान वैशिष्ट्यांसह परंतु वाढीव ड्रायव्हिंग क्षमतेसह (driving capability) आउटपुट संदेश तयार करतो. एकदिशीय बफर सामान्यतः सर्किटचे भाग वेगळे (isolate) करण्यासाठी किंवा मूळ संदेश देऊ शकतील त्यापेक्षा जास्त विद्युत प्रवाह किंवा विद्युत दाब आवश्यक असलेले लोड चालविण्यासाठी वापरले जातात.

द्विदिशात्मक (Bidirectional) बफर:

द्विदिशात्मक बफर दोन्ही दिशांमध्ये सिग्नल वाढवू शकतात, ज्यामुळे त्यांना द्विदिशात्मकरित्या डेटा प्रसारित करता येतो.

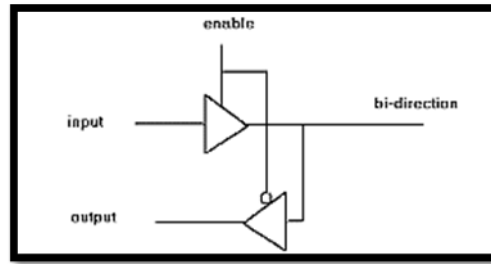


Fig 2.28 : Bidirectional Buffer

हे बफर विशेषतः अशा प्रणालींमध्ये उपयुक्त आहेत जिथे डेटा दोन्ही दिशांना प्रवाहित करणे आवश्यक आहे, जसे की कम्युनिकेशन बसेस किंवा इंटरफेसमध्ये. द्विदिशात्मक बफर्समध्ये सहसा नियंत्रण सिग्नल असतात जे डेटा प्रवाहाची दिशा ठरवतात, हे सुनिश्चित करतात की डेटा केवळ विशिष्ट दिशेने प्रसारित केला जातो.

2.4बूलीयन बीजगणित: बूलीयन बीजगणित (Boolean Algebra) नियम, ड्युअलिटी प्रमेय (Duality Theorem), डी-

मॉर्गनचे प्रमेय (De-Morgans Theorem)

बूलीयन बीजगणित ही बीजगणिताची श्रेणी आहे ज्यामध्ये व्हेरिएबलची मूल्ये, टू (true) आणि फॉलस (false) , सामान्यतः अनुक्रमे 1 आणि 0 दर्शविली जातात. हे डिजिटल सर्किट्स किंवा डिजिटल गेट्सचे विश्लेषण आणि सुलभ करण्यासाठी वापरले जाते. त्याला बायनरी बीजगणित किंवा लॉजिकल बीजगणित असेही म्हणतात. डिजिटल अनूविधुत च्या विकासामध्येबूलीयन बीजगणित मूलभूत आहे आणि सर्व आधुनिक प्रोग्रामिंग भाषांमध्ये प्रदान केले आहे.

a.बूलीयन बीजगणितचे नियम:

1.अँड (AND) नियम

- $A.0 = 0$
- $A.1 = A$
- $A.A = A$

$$A \cdot \bar{A} = 0$$

2. ऑर (OR) नियम

$$A + 0 = A$$

$$A + 1 = 1$$

$$A + A = A$$

$$A + \bar{A} = 1$$

3. कम्युटेटिव्ह (Commutative) नियम

कम्युटेटिव्ह लॉ सांगतो की व्हेरिएबल्सचा क्रम बदलल्याने लॉजिक सर्किटच्या आउटपुटवर कोणताही परिणाम होत नाही.

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

$$A + B = B + A$$

4. असोसिएटिव्ह (Associative) नियम

असोसिएटिव्ह लॉ हेबूलीयन बीजगणिततिल एक मूलभूत तत्त्व आहे जे ऑपरेशन्स कसे ग्रुप केले जाऊ शकतात हे नियंत्रित करते.

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

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

5. डिसट्रीब्यूशन (Distribution) नियम

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

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

6. इनवर्जन (Inversion) नियम

बूलीयन बीजगणित मध्ये, इनवर्जन लॉ सांगतो की व्हेरिएबलच्या दुहेरी इनवर्जन चा परिणाम मूळ व्हेरिएबलमध्येच होतो.

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

b. ड्युअलिटी प्रमेय (Duality Theorem)

हे प्रमेय असे म्हणते की एका बूलीयन फंक्शनचा द्वैत (dual) फंक्शन मिळवण्यासाठी लॉजिकल AND ऑपरेटरला लॉजिकल OR ऑपरेटरने आणि शून्यांना एकांनी बदलून मिळवता येते. प्रत्येक बूलीयन फंक्शनसाठी, एक संबंधित द्वैत फंक्शन असते.

Table 2.11 : Duality Theorem

Group1	Group2
$x + 0 = x$	$x \cdot 1 = x$
$x + 1 = 1$	$x \cdot 0 = 0$
$x + x = x$	$x \cdot x = x$
$x + x' = 1$	$x \cdot x' = 0$
$x + y = y + x$	$x \cdot y = y \cdot x$
$x + y + z = x + y + z$	$x \cdot y \cdot z = x \cdot y \cdot z$
$x \cdot y + z = x \cdot y + x \cdot z$	$x + y \cdot z = x + y \cdot x + z$

प्रत्येक ओळीत दोन बूलीयन समीकरणे असतात आणि ती एकमेकांच्या द्वैत असतात. आपण द्वैत प्रमेय वापरून Group1 आणि Group2 मधील सर्व बूलीयन समीकरणे पडताळू शकतो.

c. डी मॉर्गन्स प्रमेय (De-Morgans Theorem)

1. डी मॉर्गन्स प्रमेय १

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

NAND = Bubbled OR

या थेरमची डावी बाजू (LHS) इनपुट A आणि B सह NAND गेट दर्शवते, तर थेरमची उजवी बाजू (RHS) इनवर्टेड इनपुटसह OR गेट दर्शवते. या OR गेटला Bubbled OR असे म्हणतात.

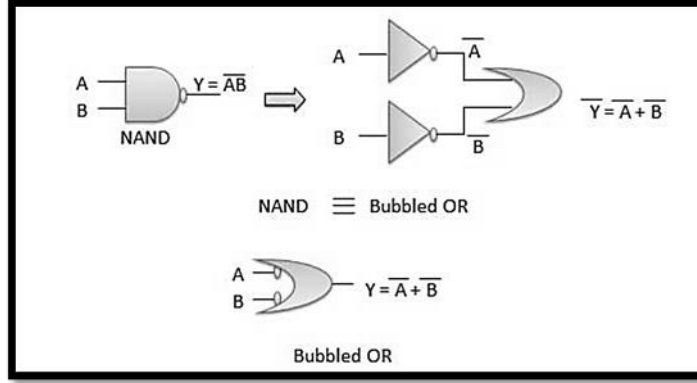


Fig 2.29- De-Morgans Theorem

तृथ टेबल :

Table 2.11: Truth Table De Morgans 1st Theorem

De-Morgan's First Theorem Verification						
A	B	A · B	$\overline{A \cdot B}$	$\overline{A}$	$\overline{B}$	$\overline{A} + \overline{B}$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

2. डी मॉर्गन्स प्रमेय २

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

NOR = Bubbled AND

या थेरमचे LHS इनपुट A आणि B सह NOR गेटचे प्रतिनिधित्व करते, तर RHS इनवर्टेड इनपुटसह AND गेटचे प्रतिनिधित्व करते. ह्या AND गेटला Bubbled AND असे म्हणतात.

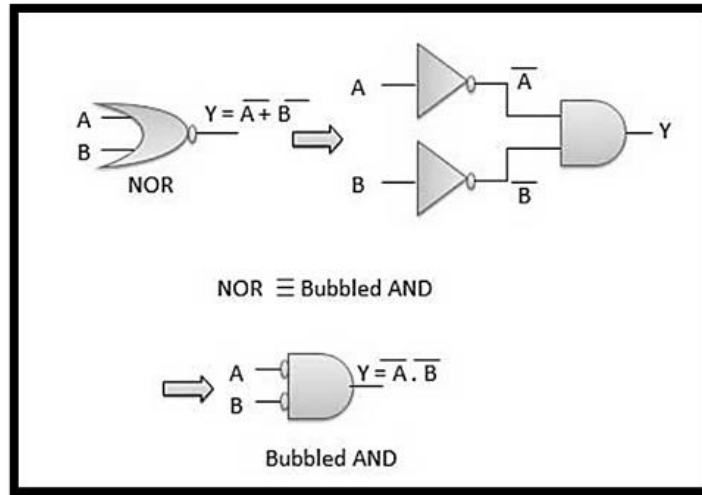


Fig 2.30- De Morgans Second Theorem

तृथ टेबल

Table 2.12 : De Morgans Second Theorem Truth Table

De-Morgan's Second Theorem Verification						
A	B	$A + B$	$\overline{A + B}$	$\overline{A}$	$\overline{B}$	$\overline{A} \cdot \overline{B}$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

स्वाध्याय:

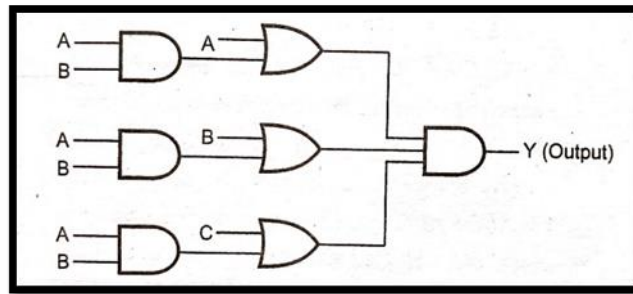
1. पॉजिटिव आणि नेगेटिव लॉजिक प्रणाली वर्णन करा.
2. NOT गेट चे चिन्ह आणि तृथ टेबल द्या.
3. लॉजिक फॅमिली चे खालील वैशिष्ट्ये परिभाषित करा
 - a. प्रोपॅगेशन डीले
 - b. शक्ति अपव्यय
 - c. फॅन आउट
 - d. क्लॉक वारंवारता
3. NOR गेट्स ला यूनिवर्सल गेट म्हणून सिद्ध करा.
4. खालील समीकरणे सरलीकृत करा:
 - a. $Y = A + \overline{A}\overline{B}C + \overline{A}\overline{B}\overline{C} + ABC + \overline{A}\overline{B}$
 - b. $Y = \overline{A}\overline{B}C + AC$
 - c. $Y = AB + \overline{A}\overline{B} + \overline{A}B$
 - d. $Y = (A + \overline{B})(\overline{A} + B)(A + B)$

5. TTL, ECL अँड CMOS ची दिलेल्या पॅरामीटर्स वर तुलना करा:

- फॅन इन
- फॅन आउट
- अनु उपयोग
- नॉइज इम्युनिटी

6. दिलेल्या लॉजिक आकृतीसाठी लॉजिक अभिव्यक्ती (output expression) निश्चित करा.

(Derive the output expression for the given logic circuit)



7. डी मॉर्गन्स प्रमेय 1 आणि 2 ची सिद्धान्त द्या आणि सिद्ध करा.

8. खालील बुलियन समीकरण लागू करण्यासाठी बेसिक लॉजिक गेट्स (AND, OR, NOT) वापरून डिजिटल सर्किटची रचना करा:

(Construct the digital circuit for the given Boolean function using basic Logic gates)

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

लघु प्रकल्प

- डायोड्स वापरून बेसिक गेट्स तयार करणे आणि तृथ टेबल तपासा.
- बेसिक गेट्स वापरून इलेक्ट्रॉनिक लॉक तयार करणे.
- XOR गेट वापरून स्टेअरकेस लाइट कंट्रोल करणे.

संदर्भ: References पुस्तके

Sr No.	Title	Author	Publisher
1.	Modern Digital Electronics	R.P Jain	McGraw Hill Education ISBN 13-978-0070669116
2.	Digital Electronics	Dr. Menka Yadav	AICTE E- Kutumb Khanna Publisher ISBN : 978-81-959863-0-9

संकेतस्थल:

- <https://da-iitb.vlabs.ac.in>
- <https://www.techtarget.com/whatis/definition/logic-gate-AND-OR-XOR-NOT-NAND-NOR-and-XNOR>
- <https://www.geeksforgeeks.org/introduction-to-logic-family/>

विद्यार्थ्यांसाठी संदर्भ सारणी

TTL आणि CMOS कुटुंबातील लॉजिक गेट्स IC क्रमांक:

Sr No	Device	TTL IC Number	CMOS Equivalent
1	Quad 2-Input AND Gate	IC 7408	CD4081
2	Quad 2-Input OR Gate	IC 7432	CD4071
3	Hex Inverter	IC7404	CD4069
4	Quad 2-Input NAND Gate	IC 7400	CD 4011
5	Quad 2-Input NOR Gate	IC 7402	CD4001
6	Quad 2-Input XOR Gate	IC 7486	CD4070
7	Quad 2-Input XNOR Gate	IC 74266	CD4077
8	Octal Buffer/Driver with 3-State Outputs	IC 74244	CD4503

युनिट 3: कॉम्बिनेशनल लॉजिक सर्किट

Combinational Logic Circuits

कोर्स निष्पत्ती (Course outcome) :

दिलेल्या अनुप्रयोगासाठी सोपे संयुक्तिक लॉजिक मंडळ विकसित करणे.

युनिट निष्पत्ती (Unit Outcomes):

- 3a. दिलेले लॉजिक एक्सप्रेसनसाठी मानक SOP/POS स्वरूपातील लॉजिक सर्किट्स विकसित करणे
Develop logic circuits for standard SOP/POS form of the given logic expression.
- 3b. k-map वापरून दिलेले लॉजिक समीकरण कमी करणे. (चार व्हेरिएबल्स पर्यंत)
Minimize the given logic expression using K-map (upto 4 Variables).
- 3c. k-map वापरून बेरीज आणि वजाबाकी करण्यासाठी लॉजिक मंडळ (Logic Circuit) विकसित करणे
Design adder and subtractor using K map.
- 3d. निर्दिष्ट एन्कोडर आणि डीकोडरचे कार्य ब्लॉक (Block) आकृती आणि तृथ टेबल च्या मदतीने वर्णन करणे.
Describe working of specified Encoder and Decoder with help of block diagram and truth table.
- 3e. मल्टीप्लेक्सर (Multiplexer) आणि डी-मल्टीप्लेक्सर (DEMUX)चे कार्य वर्णन करणे.
Describe the working of Multiplexer and Demultiplexer.

परिचय

डिजिटल सर्किटचे दोन मुख्य प्रकार आहेत 1) एकत्रित किंवा कॉम्बिनेशनल (Combinational) मंडळ 2) एकसर(Sequential) मंडळ .

कॉम्बिनेशनल मंडळामध्ये मिळणारे निर्गम (output) हे त्यावेळी दिलेल्या निवेश (Input) वर फक्त अवलंबून असते, अशा सर्किट्सला स्मरणशक्ती (Memory) नसते परंतु एकसर मंडळामध्ये मिळणारे निर्गम (Output) हे त्या वेळच्या निवेश (Input) वर तसेच पूर्वीच्या निवेश (Input) आणि निर्गम (Output) वर देखील अवलंबून असते. अशा सर्किट्सला पूर्वीची माहिती साठवण्यासाठी स्मरणशक्ती (Memory) असते.

कॉम्बिनेशनल लॉजिक मंडळ तयार करण्यासाठी मल्टिप्लेक्सर्स (multiplexer) ,डी मल्टिप्लेक्सर्स (Demultiplexer) वापरतात. तसेच (Encoder) व (Decoder) साठी आय सी (IC) चा वापर केल्यास सिस्टम ची विश्वासहर्ता वाढते व किंमत कमी होते.

3.1 स्टँडर्ड बुलियन प्रतिनिधित्व (Standard Boolean Representation)

लिटरेल्स (literals) : अपूरक (Uncomplimented) किंवा पूरक (complimented) स्वरूपातील व्हेरिएबलला लिटरल्स म्हणतात. एक स्टँडर्ड गुणात्मक पद (Standard Product term) किंवा मीन टर्म (minterm) हे दिलेल्या लॉजिक समीकरणासाठी (Logical Expression) सर्व स्वतंत्र निवेशीत (Input) व्हेरिएबल्सचे गुणात्मक पद आहे जे लॉजिक (Logic) '1' निर्गममासह (Output) सत्यसारणीच्या पंक्तीशी (Row) संबंधित आहे.

एक स्टँडर्ड बेरजीय पद (Standard sum form) किंवा मॅक्स टर्म (max term) हे दिलेल्या लॉजिक समीकरणासाठी (Logical Expression) सर्व स्वतंत्र निवेशीत (Input) व्हेरिएबल्सचे बेरजीय पद आहे जे लॉजिक (Logic) '0' निर्गममासह (Output) सत्यसारणीच्या पंक्तीशी (Row) संबंधित आहे. सामान्यतः फंक्शन (Function) लहान (Minimise) करण्याआधी दोन स्टँडर्ड फॉर्म पैकी एका मध्ये लिहावी लागते.

दोन स्टँडर्ड प्रकार पुढील प्रमाणे आहेत

- 1) गुणात्मक राशींचे बेरजीय रूप (Sum Of Product, SOP form)
- 2) बेरजीय राशींचे गुणात्मक रूप (Product of Sum , POS form)

SOP फॉर्म: गुणात्मक पद फॉर्म ची बेरीज हा बुलियन बीजगणितातील समीकरणाचा(Expression) एक प्रकार आहे ,ज्या मध्ये इनपुटच्या(input) भिन्न गुणात्मक पद संज्ञा एकत्रित केल्या जातात.

मीन टर्म (min term): एक "मीन -टर्म "(min term) एक बुलियन समीकरण(Expression) आहे परिणामी एका सेलच्या आउटपुट साठी '1' आणि तृथ टेबल तील इतर सर्व सेल साठी '0's असतात.

उदाहरण : $F(A,B) = A+B$ साठी

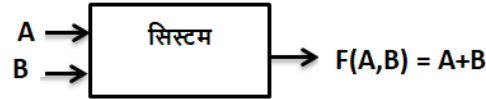


Fig 3.1 - System Diagram

$F(A,B)$ चे सर्व संभाव्य मूल्य पाहण्यासाठी तृथ टेबल तक्ता क्रमांक 3.1 दिली आहे

Table 3.1: Truth Table

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

तृथ टेबल (Truth Table)मध्ये इनपुट आणि आऊटपुट चा संच असतो . जर 'n' पुट व्हेरिएबल्स असतील ,तर शून्य आणि एक सह 2^n संभाव्य संयोजन (Possible combinations) असतील. त्यामुळे प्रत्येक आउटपुट व्हेरिएबल चे मूल्य इनपुट व्हेरिएबल्सच्या संयोजनावर(combination) अवलंबून असते. तर प्रत्येक आउटपुट मध्ये इनपुट व्हेरिएबल्सच्या काही संयोजनासाठी '1' आणि इनपुट व्हेरिएबल्सच्या इतर संयोजनासाठी '0' असेल. म्हणून आपण प्रत्येक आउटपुट खालील दोन प्रकारे व्यक्त करू शकतो .

- 1) गुणात्मक राशींचे बेरजीय रूप (Sum Of Product SOP)
- 2) बेरजीय राशींचे गुणात्मक रूप (Product of Sum POS)

3.1.1 गुणात्मक राशींचे बेरजीय रूप (Sum Of Product -SOP)

कॅनोनिकल(canonical) S.O.P फॉर्म(standard SOP) म्हणजे कॅनोनिकल गुणात्मकची बेरीज फॉर्म -या फॉर्ममध्ये प्रत्येक गुणात्मक पदांमध्ये सर्व व्हेरिएबल्स (variables) समाविष्ट असतात .गुणात्मक पद किमान पद(min term) आहेत म्हणून कॅनोनिकल एस ओ पी फॉर्म ला किमान पद(min term) फॉर्म ची बेरीज असेही म्हणतात.

कॅनोनिकल S.O.P फॉर्म(standard SOP) मिळवण्याची पद्धत:

प्रथम तृथ टेबल मध्ये ज्या साठी आऊटपुट व्हेरिएबल '1' आहे ते किमान पद (minterm) ओळखा आणि नंतर त्या आऊटपुट व्हेरिएबल संबंधित बुलियन एक्स्प्रेसन मिळवण्यासाठी त्या किमान पदाचे (minterm) लॉजिकल OR करा. किमान पद (minterm) लिहिताना इनपुट '1' ला अपूरक (Uncomplimented) व्हेरिएबल्स(variables) म्हणून विचारात घ्यावे व इनपुट '0' ला कॉम्प्लिमेंटरी व्हेरिएबल्स(variables) म्हणून विचारात घ्यावे.

हे बुलियन फंक्शन किमान पदाच्या(min term) बेरीज (Sum) स्वरूपात असेल .एकापेक्षा जास्त आउटपुट व्हेरिएबल असल्यास इतर आउटपुट व्हेरिएबल साठी देखील हीच पद्धत वापरतात.

उदाहरण: तक्ता क्रमांक 3.2 तृथ टेबल (truth Table)विचारात घ्या.

Table 3.2 – Truth Table

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 संयोजनासाठी आउटपुट '1' आहे म्हणून संबंधित किमान पद(min term) आहेत. या चार किमान पदांचे लॉजिकल 'OR' करून आउटपुट 'F' चे बुलियन फंक्शन मिळेल. म्हणून आउटपुट चे बुलियन फंक्शन आहे

$$F = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

हे फंक्शन खालील दोन नोटेशन मध्ये दर्शवू शकतो

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

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

3.1.2 : SOP फॉर्म ला स्टँडर्ड SOP फॉर्म मध्ये रूपांतरित करण्यासाठी स्टेप्स :

- 1) प्रत्येक गुणात्मक पदा मध्ये उपस्थित नसलेले लिटरल शोधा.
- 2) समीकरणा मध्ये असलेल्या प्रत्येक पदाशी (term), त्या पदामध्ये नसलेल्या व्हेरिएबल व त्याचा कॉम्प्लिमेंट यांचे OR केलेले पद, यांचे AND करा.
- 3) विस्तृत नियम लागू करून पदाचा विस्तार करा आणि शब्दशः पुनर्कमीत करा.
- 4) पुनरावृत्ती केलेली पदे(term) एकदाच लिहा ,कारण $A+A=A$ (बुलियन कायदा)

खालील एक्सप्रेसन स्टँडर्ड SOP मध्ये रूपांतरित करा.

$$F = AB + A\overline{C} + BC$$

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

Min term:

$$\begin{aligned} F(A,B,C) &= ABC + AB\overline{C} + A\overline{B}\overline{C} + \overline{A}BC \\ &= m_3 + m_4 + m_6 + m_7 \\ &= \sum m(3,5,6,7) \end{aligned}$$

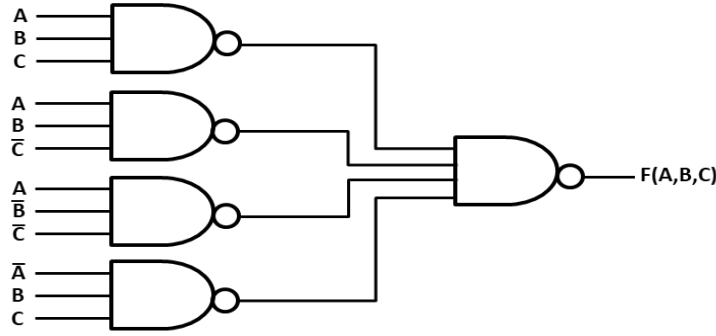


Fig 3.2 – Representation using NAND gate

3.1.3: बेरजीय राशींचे गुणात्मक रूप (Product of Sum POS):

POS फॉर्म: बेरजीय राशींचे गुणात्मक रूप (POS) हा बुलियन बीजगणितातील समीकरणा चा एक प्रकार आहे, ज्यामध्ये इनपुटच्या भिन्न बेरीज संज्ञा एकत्रितपणे गुणात्मक रूपात तयार केल्या जातात.

मॅक्स टर्म(max term) : मॅक्स टर्म हे एक बुलियन समीकरण आहे परिणामी एका सेलच्या आउटपुट साठी 0 's आणि तृथ टेबल तील इतर सर्व सेल साठी 1's असतात.

3.1.4: POS फॉर्म ला स्टँडर्ड POS फॉर्म मध्ये रूपांतरित करण्यासाठी स्टेप्स:

- 1) प्रत्येक बेरजीय पदा मध्ये उपस्थित नसलेले लिटरल शोधा.
- 2) समीकरणा मध्ये असलेल्या प्रत्येक पदाशी (term), त्या पदामध्ये नसलेल्या व्हेरिएबल व त्याचा कॉम्प्लिमेंट यांचे AND केलेले पद, यांचे OR करा.
- 3) विस्तृत नियम लागू करून पदाचा विस्तार करा आणि शब्दशः पुनर्क्रमीत करा.
- 4) पुनरावृत्ती केलेली पदे(term) एकदाच लिहा .कारण $A+A=A$ (बुलियन कायदा)

Example:

- 1) खालील समीकरण(Expression) स्टँडर्ड POS मध्ये रूपांतरित करा.

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

- 1) प्रत्येक बेरजीय पदामध्ये उपस्थित नसलेले लिटरल शोधा

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



गहाळ असलेले लिटरल B

गहाळ असलेले लिटरल A

गहाळ असलेले लिटरल C

- 2) समीकरणा मध्ये असलेल्या प्रत्येक पदाशी (term), त्या पदामध्ये नसलेल्या व्हेरिएबल व त्याचा कॉम्प्लिमेंट यांचे AND केलेले पद, यांचे OR करा.

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

- 3) विस्तृत नियम लागू करून पदाचा विस्तार करा आणि शब्दशः पुनर्क्रमीत करा.

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

- 4) पुनरावृत्ती केलेली पदे(term) एकदाच लिहा .

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

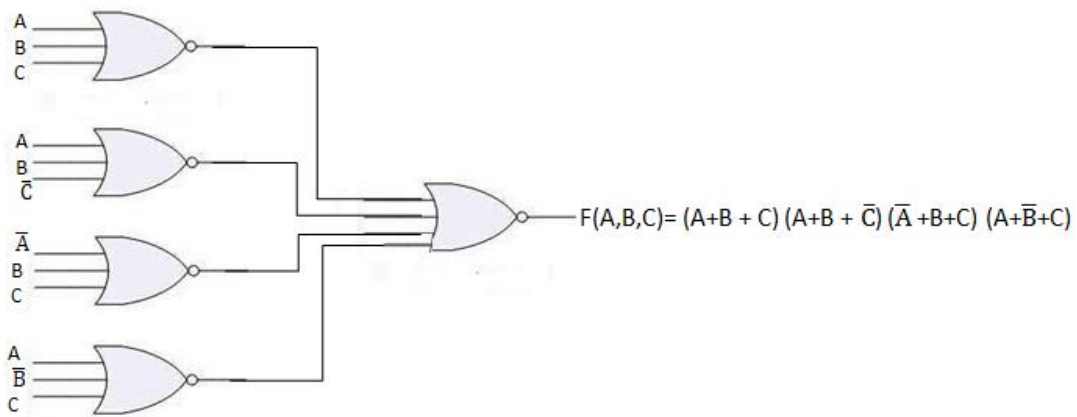


Fig 3.3 – Representation using NOR Gate

Max Terms :

$$\begin{aligned}
 1. \quad F(A,B,C) &= (A+B+C) \cdot (A+\bar{B}+C) \cdot (A+\bar{B}+\bar{C}) \cdot (\bar{A}+\bar{B}+\bar{C}) \\
 &= M_0 + M_2 + M_3 + M_7 \\
 &= \prod M(0, 2, 3, 7)
 \end{aligned}$$

3.1.5: स्टॅंडर्ड SOP मधून स्टॅंडर्ड POS फॉर्म मध्ये रूपांतर करणे.

दिलेल्या स्टॅंडर्ड SOP समीकरणा (Expression) मधील गुणात्मक पद संज्ञांचे बायनरी मूल्य समतुल्य(standard) POS समीकरणा मध्ये उपस्थित नाहीत.

बायनरी मूल्य जी SOP एक्सप्रेशन मध्ये दर्शविली जात नाहीत ती समतुल्य मूल्य POS समीकरणा मध्ये उपस्थित आहेत.

Step1: SOP समीकरणा मधील प्रत्येक गुणात्मक पदाचे मूल्यमापन करा. म्हणजेच गुणात्मक पद दर्शविणाऱ्या बायनरी संख्या निश्चित करा .

Step2: स्टेप 1 मधील मूल्यमापनात समाविष्ट नसलेल्या सर्व बायनरी संख्या निश्चित करा .

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

उदाहरण- दिलेल्या फंक्शनचे स्टॅंडर्ड SOP मधून स्टॅंडर्ड POS फॉर्म मध्ये रूपांतर करा.

$$F(A,B,C) = \sum m(0, 2, 3, 5, 7)$$

Steps:

SOP form-

$$F(A,B,C) = \sum m(0, 2, 3, 5, 7)$$

$$0=000 \text{-----} \bar{A} \cdot \bar{B} \cdot \bar{C} \quad \text{'0' s as normal, '1' s as complimentary variable}$$

$$2=010 \text{-----} \bar{A} \cdot B \cdot \bar{C}$$

$$3=011 \text{-----} \bar{A} \cdot B \cdot C$$

$$5=101 \text{-----} A \cdot \bar{B} \cdot C$$

$$7=111 \text{-----} A \cdot B \cdot C$$

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

SOP मधील मूल्यमापनात समाविष्ट नसलेल्या सर्व बायनरी संख्या POS फॉर्ममध्ये असतात.

POS form:

$$F(A,B,C) = \Pi M(1,4,6)$$

$$1 = 001 \text{-----} (A+B+\bar{C}) \quad 1' \text{ s as normal, } 0' \text{ s as complimentary variable.}$$

$$4 = 100 \text{-----} (\bar{A}+B+C)$$

$$6 = 110 \text{-----} (\bar{A}+\bar{B}+C)$$

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

3.2 : K-map वापरून दिलेली लॉजिक समीकरण लहान करणे (चार व्हेरिएबल्स पर्यंत)

अनेक डिजिटल सर्किट्स मध्ये आणि व्यावहारिक समस्यांमध्ये कमीत कमी व्हेरिएबल सह लहान एक्सप्रेशन शोधणे आवश्यक आहे. कोणत्याही बुलियन बीजगणित प्रमेयांचा वापर न करता के मॅप वापरून 3, 4 व्हेरिएबल्सचे बुलियन एक्सप्रेशन अगदी सहजपणे लहान करू शकतो. के मॅप मध्ये समस्येच्या गरजेनुसार SOP आणि POS असे दोन प्रकार असू शकतात. के मॅप हा टेबलासारखा आहे. पण तो तृथ टेबल पेक्षा जास्त माहिती देतो. k-map n व्हेरिएबल्ससाठी 2^n चौरस(square) सूचित करतात.

3.2.1: SOP आणि POS साठी के-मॅप:

SOP आणि POS साठी आपण k-map नकाशा दोन वेगवेगळ्या स्वरूपात तयार करू शकतो. कारण SOP मध्ये पूरक मूल्य (Normal variable) म्हणून शून्य (0) आणि अ-पूरक मूल्य (Complimentary variable) एक (1) आहे, त्याचप्रमाणे POS मध्ये पूरक मूल्य (Normal variable) म्हणून एक (1) आणि अ-पूरक मूल्य (Complimentary variable) शून्य (0) आहे.

2-व्हेरिएबल के-मॅप: 2 व्हेरिएबल के-मॅपमध्ये, चार स्केअर (चौकोन) तयार केले जातात. प्रत्येक स्केअरमध्ये दोन व्हेरिएबल्सची एक संज्ञा असते.

SOP आणि POS फॉर्मसाठी 2 व्हेरिएबल k-map आकृती 3.4 प्रमाणे आहे:

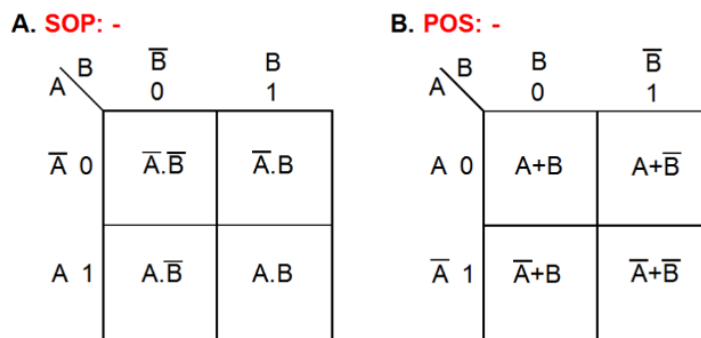


Fig 3.4: 2-Variable K-map

- **3 व्हेरिएबल्स के-मॅप:** तीन व्हेरिएबल k-map मध्ये, 8 स्केअर (चौकोन) आवश्यक आहे. SOP आणि POS फॉर्मसाठी 3 - व्हेरिएबल्स के-मॅपचे आकृती 3.5 प्रमाणे आहे:

SOP: -

	BC	$\overline{B}\overline{C}$	$\overline{B}C$	BC	$B\overline{C}$
A	00	01	11	10	
$\overline{A}0$	$\overline{A}\overline{B}\overline{C}$ 0	$\overline{A}\overline{B}C$ 1	$\overline{A}BC$ 3	$\overline{A}B\overline{C}$ 2	
A1	$A\overline{B}\overline{C}$ 4	$A\overline{B}C$ 5	ABC 7	$AB\overline{C}$ 6	

POS: -

	B+C	$B+\overline{C}$	$\overline{B}+C$	$\overline{B}+\overline{C}$
A	00	01	11	10
A0	$A+B+C$ 0	$A+B+\overline{C}$ 1	$A+\overline{B}+C$ 3	$A+\overline{B}+\overline{C}$ 2
$\overline{A}1$	$\overline{A}+B+C$ 4	$\overline{A}+B+\overline{C}$ 5	$\overline{A}+\overline{B}+C$ 7	$\overline{A}+\overline{B}+\overline{C}$ 6

	C	$\overline{C}$
AB	0	1
$\overline{A}\overline{B}00$	$\overline{A}\overline{B}\overline{C}$ 0	$\overline{A}\overline{B}C$ 1
$\overline{A}B01$	$\overline{A}B\overline{C}$ 2	$\overline{A}BC$ 3
AB11	$AB\overline{C}$ 6	ABC 7
$A\overline{B}10$	$A\overline{B}\overline{C}$ 4	$A\overline{B}C$ 5

Fig 3.5: 4-Variable K-map

4 व्हेरिएबल्स के-मॅप: चार व्हेरिएबल k-map मध्ये, 16 स्केअर(चोकोन) आवश्यक आहे. SOP आणि POS फॉर्मसाठी 3 - व्हेरिएबल्स के-मॅपचे आकृती 3.6 प्रमाणे आहे:

A. SOP: -

	CD	$\overline{C}\overline{D}$	$\overline{C}D$	$C\overline{D}$
AB	00	01	11	10
$\overline{A}\overline{B}00$	$\overline{A}\overline{B}\overline{C}\overline{D}$ 0	$\overline{A}\overline{B}C\overline{D}$ 1	$\overline{A}\overline{B}C D$ 3	$\overline{A}\overline{B}\overline{C} D$ 2
$\overline{A}B01$	$\overline{A}B\overline{C}\overline{D}$ 4	$\overline{A}B\overline{C} D$ 5	$\overline{A}B C \overline{D}$ 7	$\overline{A}B C D$ 6
AB11	$AB\overline{C}\overline{D}$ 12	$AB\overline{C} D$ 13	$AB C \overline{D}$ 15	$AB C D$ 14
$A\overline{B}10$	$A\overline{B}\overline{C}\overline{D}$ 8	$A\overline{B}\overline{C} D$ 9	$A\overline{B} C \overline{D}$ 11	$A\overline{B} C D$ 10

B. POS: -

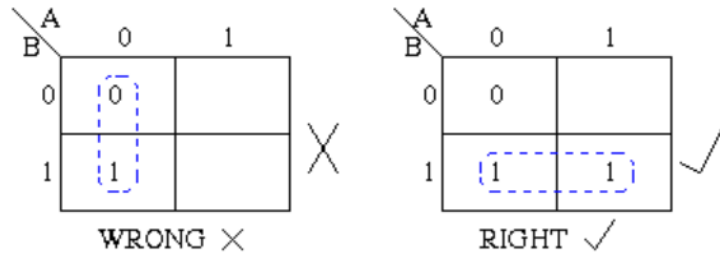
	C+D	$C+\overline{D}$	$\overline{C}+D$	$\overline{C}+\overline{D}$
A+B	00	01	11	10
A+B00	$A+B+C+D$ 0	$A+B+C+\overline{D}$ 1	$A+B+\overline{C}+D$ 3	$A+B+\overline{C}+\overline{D}$ 2
A+B01	$A+B+C+D$ 4	$A+B+C+\overline{D}$ 5	$A+B+\overline{C}+D$ 7	$A+B+\overline{C}+\overline{D}$ 6
A+B11	$A+B+C+D$ 12	$A+B+C+\overline{D}$ 13	$A+B+\overline{C}+D$ 15	$A+B+\overline{C}+\overline{D}$ 14
A+B10	$A+B+C+D$ 8	$A+B+C+\overline{D}$ 9	$A+B+\overline{C}+D$ 11	$A+B+\overline{C}+\overline{D}$ 10

Fig 3.6: 4-Variable K-map

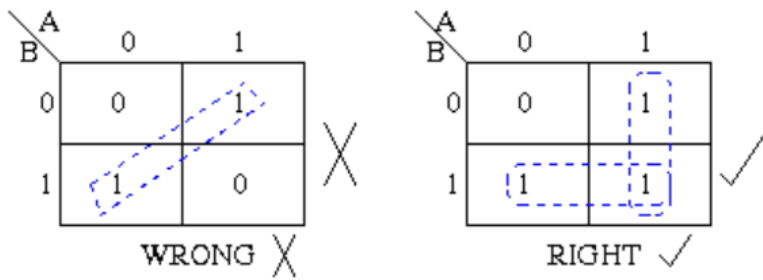
3.2.2 : के मॅप (K-map) सरलीकरणाचे नियम :

के मॅप समीप असलेल्या पेशींचा एकत्रित समूह करून एक्सप्रेसशनच्या सरली करण्यासाठी खालील नियम वापरतो-

1) गटांमध्ये शून्य असलेल्या कोणत्याही सेलचा समावेश असू शकत नाही.



2) गट क्षैतिज (horizontal) किंवा अनुलंब (vertical) असू शकतात परंतु कर्ण रेषा (diagonal) नसतात



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

म्हणजेच $n=1$ असल्यास, गटात $2^1=2$ दोन 1 असतील. जर $n=2$ असल्यास, गटात $2^2=4$ चार 1 असतील.

Right

Wrong

A \ B	0	1
0	1	1
1	0	0

A \ B	0	1
0	1	1
1	1	1

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

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

4) प्रत्येक गट शक्य तितका मोठा असावा

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

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

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

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

6) गट ओव्हरलॅप होऊ शकतात.

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

Right

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

Wrong

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

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

8) शक्य तितके 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

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

Wrong

9) X(don't care) चे 1 किंवा 0 बरोबर गट करता येतात.

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

AB \ CD	00	01	11	10
00				
01			0	0
11	X	X	X	X
10	0	0		

3.2.3: के मॅप वापरून एक्सप्रेशन सोडवण्याच्या स्टेप्स:

1. व्हेरिएबलच्या संख्येनुसार के मॅप निवडा.
2. समस्येमध्ये दिल्याप्रमाणे किमान पद (min term) किंवा कमाल पद (max term) ओळखा.
3. एसओपी(SOP) साठी 1's के मॅप च्या सेलमध्ये किमान पद (min term) मध्ये ठेवा.
4. पीओएस(POS) साठी 0's के मॅप च्या सेलमध्ये कमाल पद (max term) मध्ये ठेवा.

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 गटांमध्ये गुणात्मक पद शोधा आणि SOP फॉर्मसाठी त्यांची बेरीज करा.

3.2.4 : SOP फॉर्म के -मॅप : K-map वापरून खालील समीकरण(Expression) लहान(Minimise) करा.

1) **तीन व्हेरिएबल्सचा के मॅप (K-map) :**

$$F(A,B,C) = \sum m(0,1,4,6,7)$$

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

Labels: $\bar{A}\bar{B}$ (top-left), $\bar{A}B$ (top-right), AB (bottom-right), $B\bar{C}$ (bottom-left)

पहिल्या गटात, व्हेरिएबल A बदलत आहे आणि B आणि C अपरिवर्तित(Constant) आहे. त्यामुळे आउटपुटची पहिली संज्ञा $\bar{B}\bar{C}$ असेल (कारण या गटातील B आणि C = 0).

दुसऱ्या गटात, व्हेरिएबल C बदलत आहे आणि व्हेरिएबल A आणि B अपरिवर्तित(Constant) आहे. तर आउटपुटची दुसरी संज्ञा $\bar{A}\bar{B}$ असेल. (कारण या गटातील A आणि B = 0)

अंतिम एक्सप्रेशन $F = \bar{B}\bar{C} + \bar{A}\bar{B} + AB$

2) $F(A,B,C) = \sum m(0,1,4,6,7)$

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

Labels: $\bar{A}\bar{B}$ (top-left), AB (top-right), $A\bar{C}$ (bottom-left), AB (bottom-right)

अंतिम एक्सप्रेशन $F = AB + \bar{A}\bar{B} + A\bar{C}$

3) **चार व्हेरिएबल्सचा के मॅप (K-map) :**

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

के मैप फोल्ड करून

		$\overline{B} \overline{D}$			
CD	AB	00	01	11	10
		00	01	11	10
	00	1	0	0	1
	01	0	1	1	0
	11	0	1	1	0
	10	1	0	0	1
		BD			
		0	1	3	2
		4	5	7	6
		12	13	15	14
		8	9	11	10

चार 1's चा एक गट तयार करणे

अंतिम एक्सप्रेशन

$$F = BD + \overline{B} \overline{D}$$

4) X (don't care) चे के मैप (K-map)

$$F(A,B,C,D) = \sum m(1, 5, 6, 11, 12, 13, 15) + d(4)$$

		CD			
AB		00	01	11	10
		00	01	11	10
	00		1		
	01	X	1		1
	11	1	1		1
	10			1	

अंतिम एक्सप्रेशन

$$F = B \overline{C} + \overline{A} \overline{C} D + B \overline{D} + A \overline{B} C D$$

3.2.5: POS फॉर्म के- मैप (K-map): K-map वापरून खालील एक्सप्रेशन लहान (Minimise) करा.

1) तीन व्हेरिएबल्सचा के मैप (K-map)

$$F(A, B, C) = \prod M(0, 3, 6, 7)$$

		BC			
A		00	01	11	10
		00	01	11	10
	0	0	0	0	0
	1	0	0	0	0
		4	5	7	6

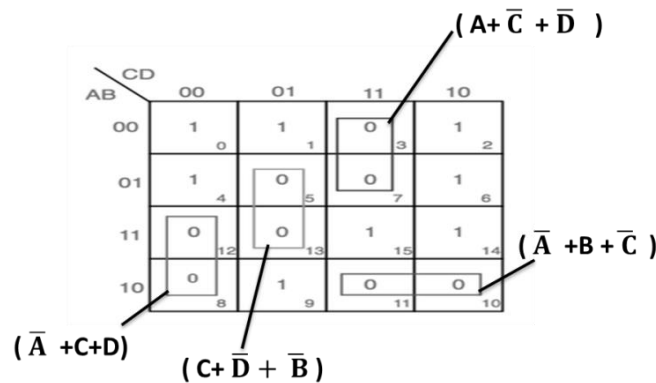
$Y = (\overline{A} + \overline{B})_1 (A + \overline{C})$

अंतिम एक्सप्रेशन

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

2) चार व्हेरिएबल्सचा के मॅप (K-map) :

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



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

3.3: अॅडर(Adder) आणि सबट्रॅक्टर(Subtractor)

गणितीय मंडळ हे कोणत्याही प्रोसेसरच्या ALU मध्ये वापरल्या जाणाऱ्या युनिट पैकी एक आहे. हे अंकगणितीय सर्किट एकापेक्षा जास्त बीट संख्यांच्या बेरीज आणि वजाबाकी साठी वापरतात.

3.3.1 हाफ अॅडर(Half Adder) : हाफ अॅडर हे एक लॉजिक सर्किट असून त्यामध्ये एका वेळी 2 बायनरी बीट (bit) ची बेरीज केली जाते. अॅडर साठी एक एक्स और (XOR) एक अँड (AND) गेट यांची आवश्यकता असते. एक्स और XOR गेटच्या आउटपुटला बेरीज (SUM) व अँड गेटच्या आउटपुटला हातचा (CARRY) म्हणतात

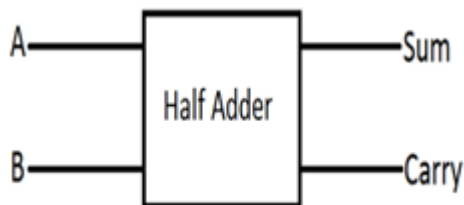


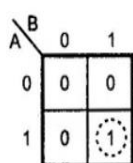
Fig.3.7 Half Adder

Input		Output	
A	B	M	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Table 3.3.- Half Adder Truth Table

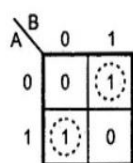
K-map वापरून हाफ अॅडर(Half Adder) विकसित /डिझाईन करणे.

For Carry

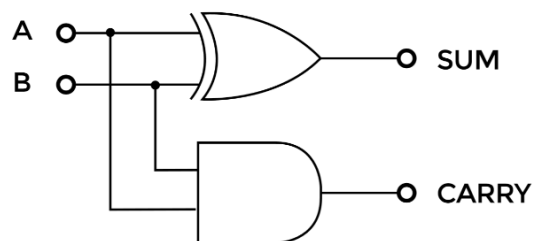


$$\text{Carry} = AB$$

For Sum



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

Fig 3.8
Half Adder

K-map

Fig 3.9 Half Adder Logic Circuit

3.3.2 : फुल अॅडर (FULL Adder) : फुल अॅडर हे एक लॉजिक सर्किट असून त्यामध्ये एका वेळी 3 बायनरी बीट (bit) ची बेरीज केली जाते. फुल अॅडर साठी एक एक्स और (XOR), तीन अॅंड (AND) गेट यांची आवश्यकता असते. एक्स और XOR गेटच्या आउटपुटला बेरीज (SUM) व अॅंड गेटच्या आउटपुटला हातचा (CARRY) म्हणतात.

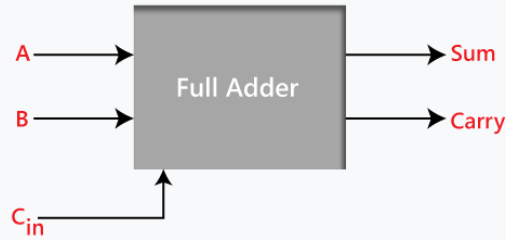
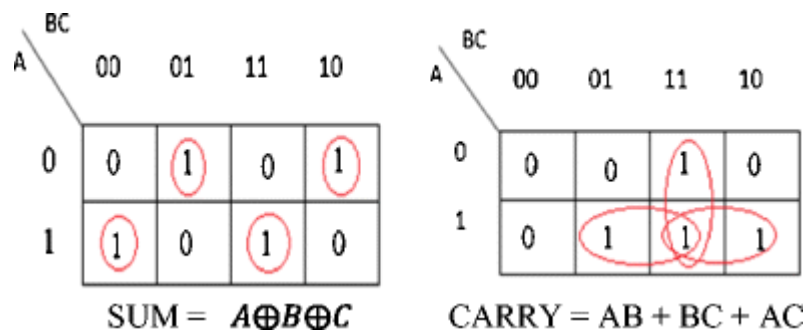


Fig 3.10: Full Adder

Table 3.4 Full Adder Truth table

Inputs			Outputs	
A	B	C _{in}	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



आकृती 3.11 फुल अॅडर के मॅप

$$S = \overline{A} \overline{B} C_{in} + \overline{A} B \overline{C}_{in} + A \overline{B} \overline{C}_{in} + A B C_{in}$$

$$S = C_{in} (\overline{A} \overline{B} + \overline{A} B + A \overline{B} + A B) + \overline{C}_{in} (\overline{A} \overline{B} + \overline{A} B + A \overline{B} + A B)$$

EX-NOR EX-OR

$$\therefore S = C_{in} (\overline{A} \overline{B} + \overline{A} B + A \overline{B} + A B) + \overline{C}_{in} (\overline{A} \overline{B} + \overline{A} B + A \overline{B} + A B)$$

$$\text{Let } X = \overline{A} \overline{B} + \overline{A} B + A \overline{B} + A B$$

$$\therefore S = C_{in} X + \overline{C}_{in} X = C_{in} \oplus X$$

$$\therefore S = C_{in} \oplus (\overline{A} \overline{B} + \overline{A} B + A \overline{B} + A B)$$

$$\text{But } \overline{A} \overline{B} + \overline{A} B + A \overline{B} + A B = A \oplus B$$

$$\therefore S = C_{in} \oplus A \oplus B$$

$$C_o = AB + AC_{in} + BC_{in}$$

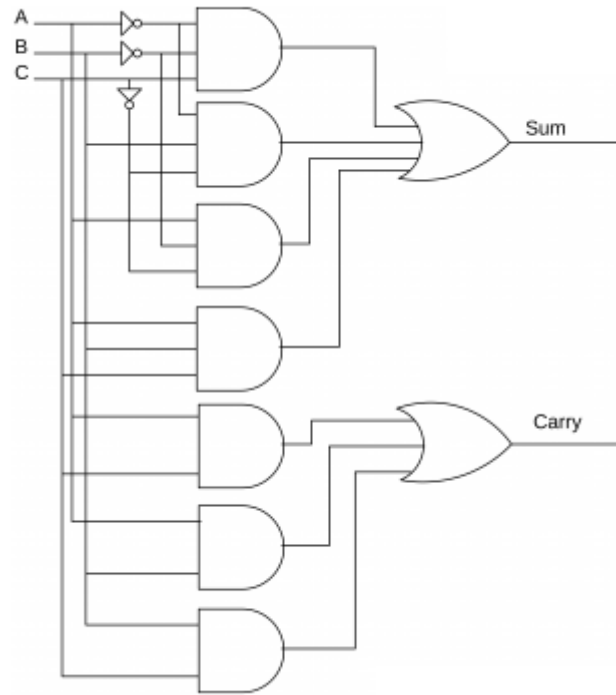


Fig. 3.12 Full Adder Logic Circuit

3.3.3: दोन 2 हाफ अॅडर वापरून फुल अॅडर: दोन हाफ अॅडर आणि एक OR गेटसह फुल सबट्रॅक्टर डिझाइन केले जाऊ शकते. दुसऱ्या हाफ अॅडर मधील बेरीज आउटपुट हे C_{in} आणि पहिल्या हाफ अॅडर चे XOR आहे. जे पूर्ण फुल अॅडर च्या बेरीज आउटपुटसारखेच आहे.

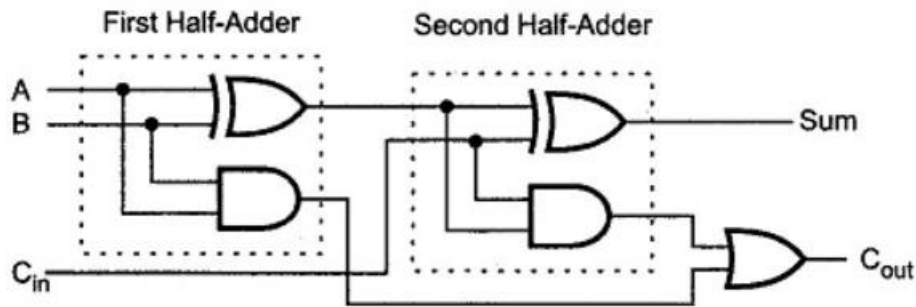


Fig 3.13 Full Adder using two Half adders

3.3.4 : हाफ सबट्रॅक्टर (Half Subtractor) : हाफ सबट्रॅक्टर हे एक लॉजिक सर्किट असून त्यामध्ये एका वेळी 2 बायनरी बीट (bit) ची वजाबाकी केली जाते. सबट्रॅक्टर साठी एक एक्स ओर (XOR), एक अॅंड (AND) गेट, एक NOT गेट यांची आवश्यकता असते.

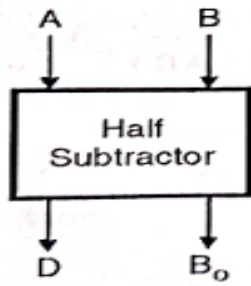


Fig 3.14 Half Subtractor

Input		Output	
A	B	D	B ₀
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Truth Table 3.5 – Half Subtractor Truth Table

For Difference

A \ B	0	1
0	0	1
1	1	0

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

For Borrow

A \ B	0	1
0	0	1
1	0	0

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

Fig 3.15 Half Subtractor K-Map

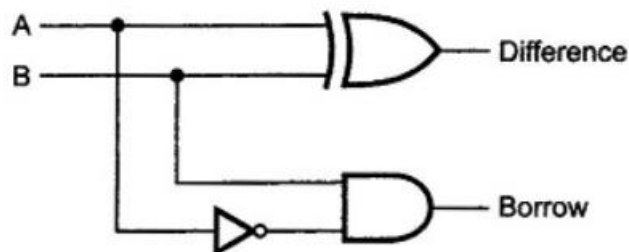
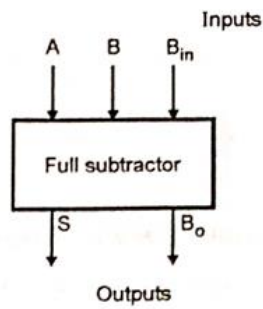


Fig 3.16 : Half Subtractor Logic Circuit

3.3.5: फुल सबट्रैक्टर (FULL Subtractor):

फुल सबट्रैक्टर: हे कॉम्बिनेशनल लॉजिक सर्किट आहे.

हे दोन सिंगल बिट संख्या वजा करण्याच्या उद्देशाने वापरले जाते. अशा प्रकारे, पूर्ण वजा करणाऱ्यामध्ये तीन बिट्सची वजाबाकी करण्याची क्षमता असते. दाखवल्याप्रमाणे पूर्ण वजाबाकीमध्ये 3 इनपुट आणि 2 आउट पूट असतात.



Inputs			Outputs	
A	B	Borrow _{in}	Diff	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Fig 3.17 : Full Subtractor

Table 3.6- Full Subtractor Truth Table

फरक(difference) आणि हातचा(borrow) साठी के मॅप,

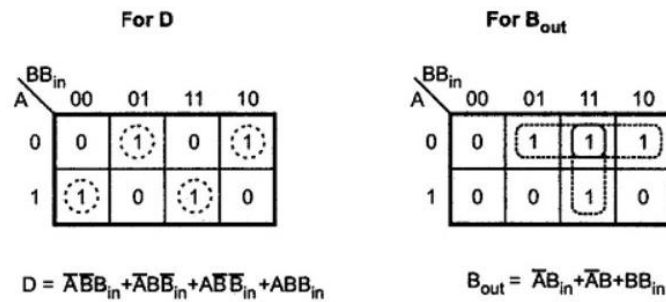


Fig 3.18 Full Subtractor K-Map

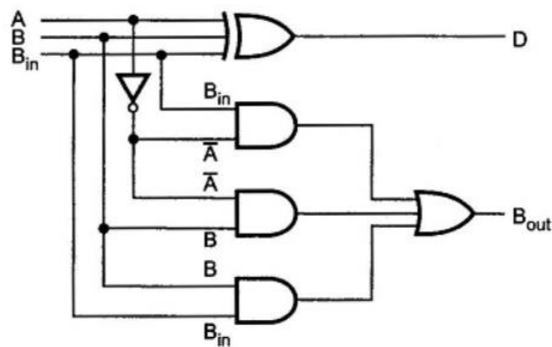


Fig 3.19 : Full Subtractor Logic Circuit

3.3.6: दोन हाफ सबट्रॅक्टर वापरून फुल सबट्रॅक्टर: दोन हाफ सबट्रॅक्टर आणि एक OR गेटसह फुल सबट्रॅक्टर डिझाइन केले जाऊ शकते. दुसऱ्या हाफ सबट्रॅक्टर मधील फरक आउटपुट हे B_{in} आणि पहिल्या हाफ सबट्रॅक्टरचे XOR आहे. जे पूर्ण फुल सबट्रॅक्टर च्या फरक आउटपुटसारखेच आहे.

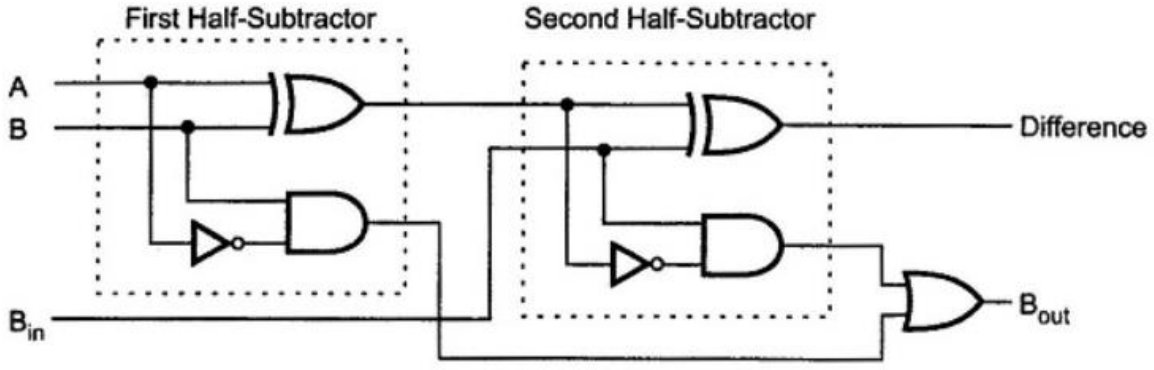


Fig 3.20 : Full Subtractor using two half subtractors

3.3.7: एन-बिट समांतर अॅडर (n-bit parallel adder):

पॅरलल अॅडर हे एक बायनरी अॅडर सर्किट आहे जे एन-बिट असलेल्या दोन बायनरी नंबर जोडण्यासाठी लागू केले जाते (उदाहरणार्थ, 4-बिट बायनरी नंबर जोडण्यासाठी, आम्ही 4-बिट पॅरलल अॅडर वापरतो आणि असेच). त्याच्या नावाप्रमाणेच, पॅरलल अॅडर हे डिजिटल कॉम्बिनेशनल सर्किट आहे जे समांतर स्वरूपात दोन बायनरी संख्या जोडते आणि समांतर स्वरूपात त्या बायनरी संख्यांची अंकगणित बेरीज तयार करते.

वर नमूद केल्याप्रमाणे एक पूर्ण अॅडर फक्त दोन, एक-बिट (1 bit) बायनरी संख्या जोडू शकतो. ज्यामध्ये दोन इनपुट बिट आणि एक इनपुट कॅरी बिट असतात, म्हणजे तीन बिट जोडणे. पण प्रत्यक्ष व्यवहारात आपल्याला अशा बायनरी संख्या जोडायच्या आहेत ज्यांची लांबी एक बिटापेक्षा जास्त आहे. अशा बायनरी संख्या जोडण्यासाठी, आपण समांतर बायनरी अॅडर वापरतो. जो 4-बिट, 5-बिट इत्यादी कोणत्याही बिट लांबीच्या दोन बायनरी संख्या जोडण्यास सक्षम आहे.

साखळी पद्धतीने जोडलेल्या फुल-अॅडर्सच्या मदतीने एन-बिट समांतर अॅडर आकृती 3.21 मध्ये दर्शविला आहे

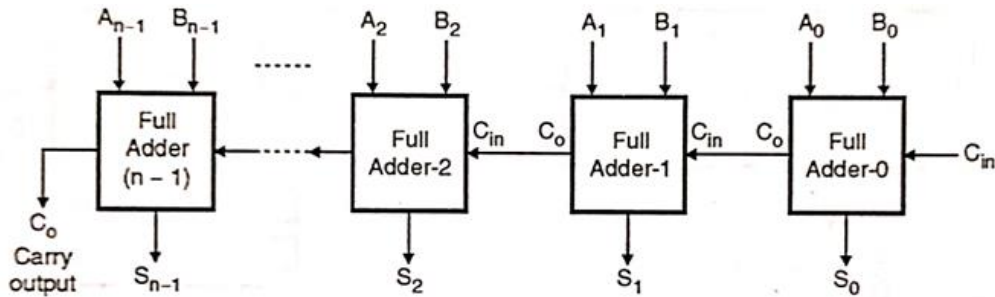


Figure 3.21: n-bit parallel adder:

- 1) प्रत्येक फुल अॅडर, कॅरी-इन इनपुट म्हणून घेतो आणि आउटपुट म्हणून कॅरी-आउट आणि सम बिट तयार करतो.
- 2) फुल अॅडर द्वारे उत्पादित कॅरी-आउट त्याच्या सर्वात जवळच्या लक्षणीय फुल अॅडर साठी कॅरी-इन म्हणून काम करते.
- 3) जेव्हा कॅरी-इन फुल अॅडर साठी उपलब्ध होते, तेव्हा ते फुल अॅडर सक्रिय करते फुल अॅडर सक्रिय झाल्यानंतर, ते कार्यान्वित होते.

3.3.7: एन (n) बिट समांतर सबट्रॅक्टर (n-bit parallel subtractor):

एक फुल सबट्रॅक्टर (full Subtractor) फक्त दोन, एक-बिट (1 bit) बायनरी संख्या वजा करू शकतो. ज्यामध्ये दोन इनपुट बिट आणि एक इनपुट बॉरो बिट असतात, म्हणजे तीन बिट असतात. प्रत्यक्ष व्यवहारात आपल्याला अशा बायनरी संख्या वजा करावयाच्या आहेत ज्यांची लांबी एक बिटापेक्षा जास्त आहे. अशा बायनरी संख्या वजा करण्यासाठी समांतरवजाकर्ता (parallel subtractor) वापरतो. जो 4-बिट, 5-बिट इत्यादी कोणत्याही बिट लांबीच्या दोन बायनरी संख्या वजा करण्यास सक्षम आहे.

साखळी पद्धतीने जोडलेल्या फुल सबट्रॅक्टर मदतीने एन-बिट समांतर फुल सबट्रॅक्टर (n bit parallel subtractor) आकृती

3.22 मध्ये दर्शविला आहे.

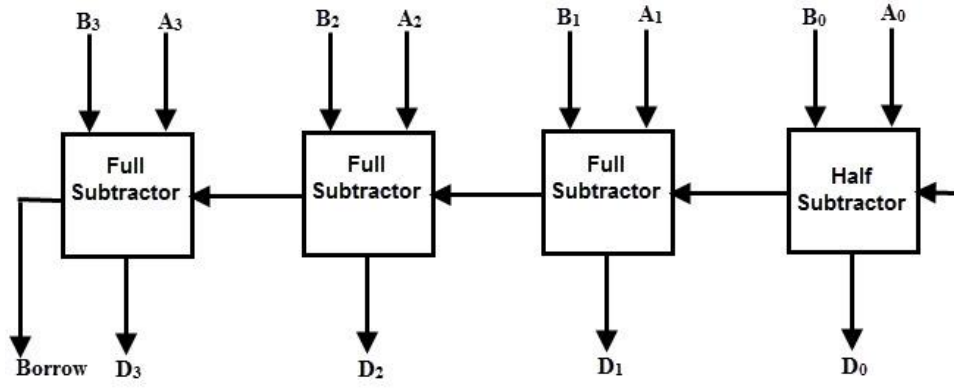


Figure 3.22 n-bit parallel subtractor

3.4 एन्कोडर आणि डीकोडर

3.4.1: एन्कोडर्स (Encoders)

बायनरी माहिती N-bit आउटपुट लाईन्समध्ये बदलणारे कॉम्बिनेशनल सर्किट्स एन्कोडर (encoder) म्हणून ओळखले जातात. बायनरी माहिती 2^N इनपुट लाईन्सच्या स्वरूपात दिली जाते. आउटपुट लाईन बायनरी माहितीसाठी एन-बिट कोड परिभाषित करतात. सोप्या शब्दात, एन्कोडर डीकोडरचे उलट ऑपरेशन करते. एका वेळी फक्त एक इनपुट लाईन सक्रिय केली जाते. उत्पादित एन-बिट आउटपुट कोड बायनरी माहितीच्या समतुल्य असतो.

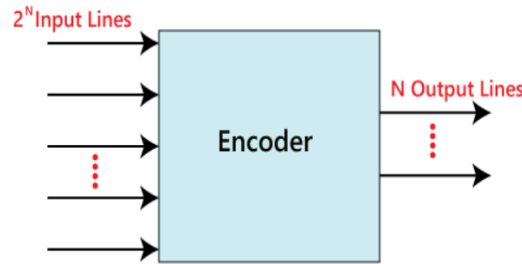


Figure 3.23 Encoder Block Diagram

एन्कोडर्सचे कार्य (Functions of Encoder):

- 1) ओळींच्या (Lines) संख्येत घट: एन्कोडर्स एकाधिक इनपुटमधून एकाच आउटपुटमध्ये माहिती प्रसारित करण्यासाठी आवश्यक असलेल्या ओळींची संख्या कमी करतात, ज्यामुळे सिस्टमचे डिझाइन सुलभ होते आणि घटकांची किंमत कमी होते.
- 2) सुधारित विश्वासार्हता: एकाधिक इनपुट्स एकाच सीरियल कोडमध्ये रूपांतरित करून, एन्कोडर माहितीच्या प्रसारणातील त्रुटीची शक्यता कमी करू शकतात.
- 3) सुधारित कार्यप्रदर्शन: एन्कोडर्स अनेक इनपुटमधून एकाच आउटपुटमध्ये माहिती प्रसारित करण्यासाठी लागणारा वेळ कमी करून डिजिटल प्रणालीचे कार्यप्रदर्शन वाढवू शकतात.

3.4.2 डिकोडर्स (Decoder):

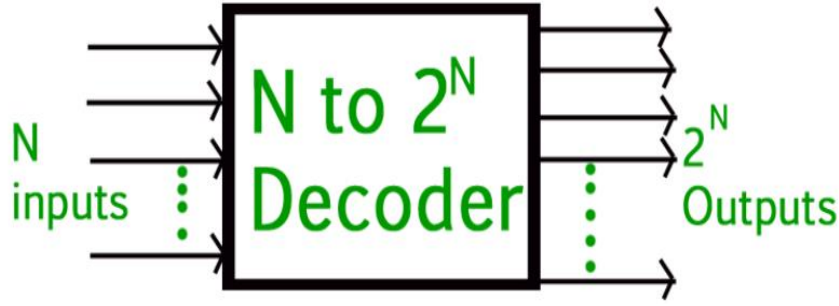


Figure 3.24 Decoder Block Diagram

बायनरी डिकोडर हे एक डिजिटल सर्किट आहे जे बायनरी कोडला आउटपुटच्या सेटमध्ये रूपांतरित करते. बायनरी कोड इच्छित आउटपुटची स्थिती दर्शवतो आणि सक्रिय असलेले विशिष्ट आउटपुट निवडण्यासाठी वापरला जातो. बायनरी डीकोडर हे एन्कोडर्सचे व्यस्त असतात आणि सामान्यतः डिजिटल सिस्टममध्ये सिरीयल कोडला आउटपुटच्या समांतर संचामध्ये रूपांतरित करण्यासाठी वापरले जातात.

बायनरी डीकोडरचे मूलभूत तत्त्व म्हणजे प्रत्येक संभाव्य बायनरी कोडसाठी एक अद्वितीय आउटपुट नियुक्त करणे. उदाहरणार्थ, 4 इनपुट आणि $2^4 = 16$ आउटपुट असलेला बायनरी डीकोडर 16 संभाव्य 4-बिट बायनरी कोडपैकी प्रत्येकाला एक अद्वितीय आउटपुट देऊ शकतो.

बायनरी डीकोडरचे इनपुट सामान्यतः कमी सक्रिय असतात, याचा अर्थ कोणत्याही वेळी फक्त एक इनपुट सक्रिय (कमी) असते आणि उर्वरित इनपुट निष्क्रिय (उच्च) असतात. सक्रिय असलेले विशिष्ट आउटपुट निवडण्यासाठी सक्रिय (कमी) इनपुटचा वापर केला जातो.

बायनरी डिकोडरचे विविध प्रकार आहेत, ज्यामध्ये प्रायोरिटी डीकोडर आहेत, जे प्रत्येक आउटपुटला प्राधान्य देतात आणि एरर-डिटेक्टिंग डीकोडर आहेत, जे बायनरी कोडमधील त्रुटी शोधू शकतात आणि एरर सिग्नल तयार करू शकतात.

सारांश, बायनरी डिकोडर हे एक डिजिटल सर्किट आहे जे बायनरी कोडला आउटपुटच्या सेटमध्ये रूपांतरित करते. बायनरी डीकोडर हे एन्कोडर्सचे व्यस्त असतात आणि सिरीयल कोडला समांतर आउटपुटमध्ये रूपांतरित करण्यासाठी डिजिटल सिस्टममध्ये मोठ्या प्रमाणावर वापरले जातात.

3.4.3 डिकोडरचे कार्य (Functions of decoder):

- 1) वाढीव लवचिकता (flexibility): बायनरी डिकोडर बायनरी कोडवर आधारित एकाधिक आउटपुटपैकी एक निवडण्याचा एक लवचिक मार्ग प्रदान करतात, ज्यामुळे अनुप्रयोगांच्या विस्तृत श्रेणीसाठी परवानगी मिळते.
- 2) सुधारित कार्यप्रदर्शन: सिरीयल कोडला आउटपुटच्या समांतर संचामध्ये रूपांतरित करून, बायनरी डीकोडर एका इनपुटमधून एकाधिक आउटपुटमध्ये माहिती प्रसारित करण्यासाठी लागणारा वेळ कमी करून डिजिटल सिस्टमची कार्यक्षमता सुधारू शकतात.
- 3) सुधारित विश्वासार्हता: एका इनपुटमधून एकाधिक आउटपुटमध्ये माहिती प्रसारित करण्यासाठी आवश्यक असलेल्या ओळींची संख्या कमी करून, बायनरी डीकोडर माहितीच्या प्रसारणात त्रुटीची शक्यता कमी करू शकतात.

3.4.4 प्राधान्य एन्कोडर (Priority Encoder):

प्राधान्य एन्कोडर:

सामान्य एन्कोडरच्या बाबतीत, कोणत्याही वेळी एक आणि फक्त एक दशांश इनपुट सक्रिय केले जाऊ शकते. परंतु काही व्यावहारिक डिजिटल प्रणालींच्या बाबतीत, दोन किंवा अधिक दशांश इनपुट एकाच वेळी अनावधानाने सक्रिय होऊ शकतात ज्यामुळे गोंधळ होऊ शकतो. उदाहरणार्थ, कीबोर्डवर, वापरकर्ता दुसरी की 2 सोडण्यापूर्वी 4 की दाबतो. अशा स्थितीत, आउटपुट की (key) चार किंवा दोन किंवा ऐवजी की (key) सहा शी संबंधित असेल. प्राधान्य एन्कोडरच्या मदतीने अशा प्रकारच्या समस्या सोडवल्या जाऊ शकतात.

डिजिटल इलेक्ट्रॉनिक्समध्ये, एकाच वेळी सक्रिय होऊ शकणाऱ्या सर्वांपैकी केवळ एका इनपुटला प्रतिसाद म्हणून आउटपुट तयार करणाऱ्या कॉम्बिनेशनल लॉजिक सर्किटला प्राधान्य एन्कोडर म्हणतात. यासाठी, ते प्राधान्य प्रणाली वापरते, आणि म्हणूनच त्याला असे नाव देण्यात आले आहे.

वापरलेली एक सर्वात लोकप्रिय प्राधान्य प्रणाली इनपुटच्या सापेक्ष परिमाणांवर आधारित आहे. प्राधान्य प्रणालीनुसार, सर्व एकाचवेळी इनपुटमध्ये सर्वात मोठे परिमाण असलेले दशांश इनपुट एन्कोड केलेले आहे. त्यामुळे, या प्रायोरिटी एन्कोडिंग सिस्टीमनुसार, जर 4 आणि 2 एकाच वेळी सक्रिय असतील तर प्राधान्य एन्कोडर 4 एन्कोड करेल.

काही व्यावहारिक प्रणालींमध्ये, प्राधान्य एन्कोडरमध्ये अनेक इनपुट असतात जे एकाच वेळी नियमितपणे सक्रिय असतात. अशा परिस्थितीत, एन्कोडरचे प्राथमिक कार्य सर्वोच्च प्राधान्याने इनपुट निवडणे आहे. प्राधान्य एन्कोडरचे हे कार्य निर्णय कर्ता (ARBITRAL) म्हणून ओळखले जाते. उदाहरणार्थ, संगणक प्रणालीमध्ये, एकाधिक इनपुट उपकरणे जोडलेली असतात आणि त्यापैकी अनेक एकाच वेळी सिस्टमला डेटा पुरवण्याचा प्रयत्न करू शकतात. या प्रकरणात, सर्व इनपुट उपकरणांमध्ये सर्वोच्च प्राधान्य असलेले इनपुट डिव्हाइस सक्षम करण्यासाठी प्राधान्य एन्कोडर जबाबदार आहे.

3.4.5 : प्राधान्य एन्कोडरचे (Priority encoder) प्रकार

अनेक प्रकारचे प्राधान्य एन्कोडर आहेत. काही सर्वात महत्वाचे प्रकारचे प्राधान्य एन्कोडर खाली सूचीबद्ध आणि स्पष्ट केले आहेत.

- 1) 4:2 इनपुट प्राधान्य एन्कोडर
- 2) ऑक्टल ते बायनरी (8:3) प्राधान्य एन्कोडर
- 3) दशांश ते BCD (Decimal to BCD) प्राधान्य एन्कोडर

1) 4-इनपुट 2 आउटपुट प्राधान्य एन्कोडर(4:2priority encoder)

4-इनपुट प्राधान्य एन्कोडरचे Q_0 , Q_1 आणि V द्वारे नियुक्त केलेले तीन आउटपुट आहेत. जेथे, Q_0 आणि Q_1 हे सामान्य आउटपुट आहेत आणि V हे आउटपुट आहे जे वैध बिट इंडिकेटर म्हणून कार्य करते. जेव्हा एन्कोडरचे एक किंवा अधिक इनपुट लॉजिक 1 च्या समान असतात तेव्हा हे तिसरे आउटपुट (V) लॉजिक 1 वर सेट केले जाते. जेव्हा एन्कोडरचे इनपुट 0 च्या समान असतात, तेव्हा कोणतेही वैध (VALID) आउटपुट नसते तेव्हा Q_0 आणि Q_1 हे आउटपुट " काळजी करू नका (don't care) म्हणून निर्दिष्ट केले जातात. आणि आउटपुट (V) लॉजिक 0 वर सेट केले जाते.

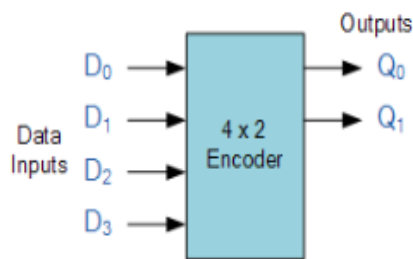


Figure 3.25 : 4:2 Priority Encoder

INPUT				OUTPUT		
D_3	D_2	D_1	D_0	Q_1	Q_0	V
0	0	0	0	x	x	0
0	0	0	1	0	0	1
0	0	1	x	0	1	1
0	1	x	x	1	0	1
1	x	x	x	1	1	1

Table 3.7- Truth Table 4:2 Priority Encoders

ऑक्टल ते बायनरी प्राधान्य एन्कोडर(8:3) encoder (Octal to Binary Priority Encoder)

या प्रकारच्या प्रायोरिटी एन्कोडरचा वापर ऑक्टल कोडचे बायनरी कोडमध्ये एन्कोडिंग करण्यासाठी केला जातो. म्हणून, या प्रकारच्या प्रायोरिटी एन्कोडरमध्ये आठ इनपुट आणि तीन आउटपुट आहेत जे खालील तृथ टेबल मध्ये दिल्याप्रमाणे संबंधित बायनरी कोड तयार करतात.

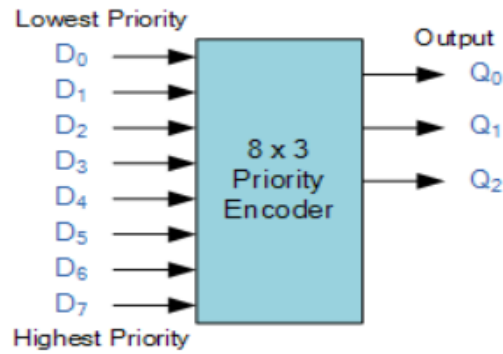


Figure 3.26: 8:3 Priority

Encoder

Table 3.8 Third Table 8:3 Preference Encoders

INPUTS								OUTPUTS			
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	Q ₂	Q ₁	Q ₀	V
0	0	0	0	0	0	0	0	x	x	x	0
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	1	x	0	0	1	1
0	0	0	0	0	1	x	x	0	1	0	1
0	0	0	0	1	x	x	x	0	1	1	1
0	0	0	1	x	x	x	x	1	0	0	1
0	0	1	x	x	x	x	x	1	0	1	1
0	1	x	x	x	x	x	x	1	1	0	1
1	x	x	x	x	x	x	x	1	1	1	1

3.4.6 : BCD ते 7 सेगमेंट डिकोडर:

बायनरी कोडेड दशांश (BCD) एन्कोडिंग स्कीम प्रत्येक दशांश संख्या (0-9) त्याच्या समतुल्य बायनरी पॅटर्नद्वारे दर्शविली जाते (जे साधारणपणे 4-बिट्सचे असते).

तर, सेव्हन सेगमेंट डिस्प्ले हे एक इलेक्ट्रॉनिक उपकरण आहे ज्यामध्ये काही निश्चित पॅटर्नमध्ये (सामान्य कॅथोड किंवा सामान्य एनोड प्रकार) व्यवस्था केलेले सात प्रकाश उत्सर्जक डायोड्स (एलईडी) असतात, ज्याचा वापर हेक्साडेसिमल अंक प्रदर्शित करण्यासाठी केला जातो (या प्रकरणात दशांश संख्या, इनपुट म्हणून बीसीडी आहे म्हणजे (०-९)).

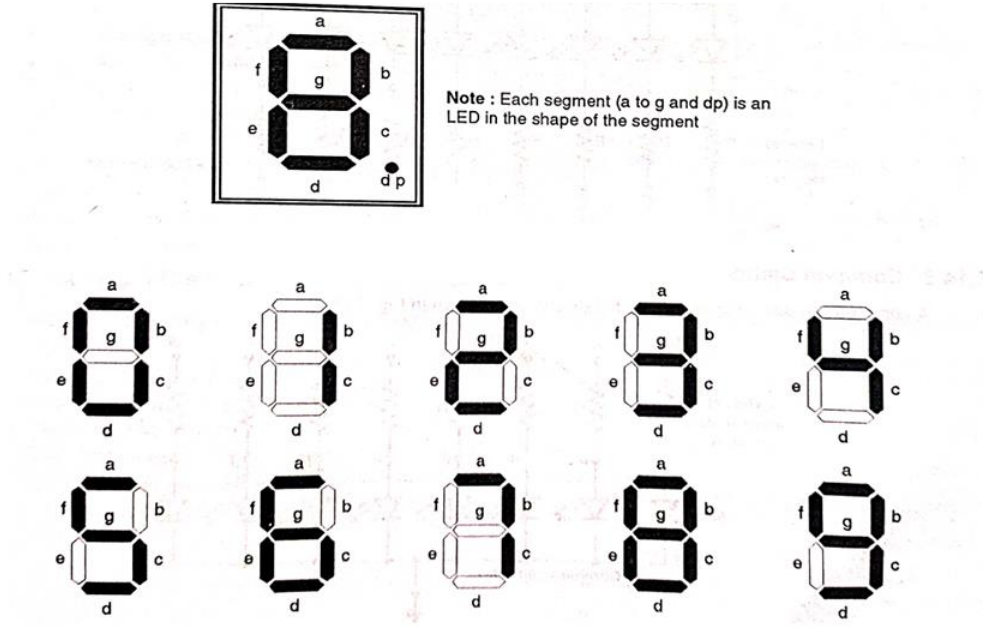


Fig 3.28 Two types of seven-segment LED displays:

1. कॉमन कॅथोड (Cathode) प्रकार: या प्रकारच्या डिस्प्लेमध्ये सात LED चे सर्व कॅथोड जमिनीवर एकत्र जोडलेले असतात किंवा $-V_{cc}$ (म्हणून, कॉमन कॅथोड) आणि LED डिस्प्ले अंक दाखवतात जेव्हा वैयक्तिक एनोड्सना काही 'हाय' सिग्नल पुरवला जातो.

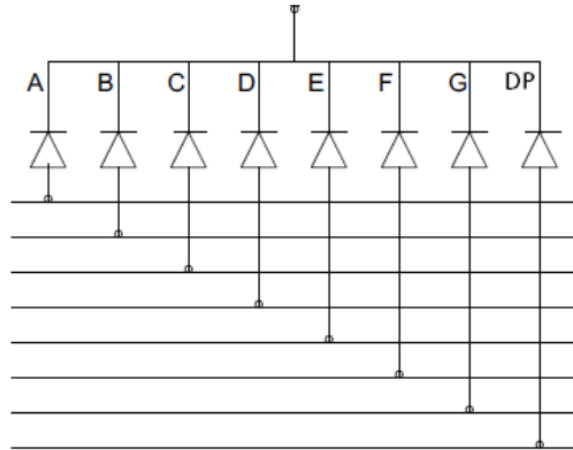


Figure 3.28 Common Cathode Type

2) कॉमन एनोड (Anode) प्रकार: या प्रकारच्या डिस्प्लेमध्ये सात एलईडीचे सर्व एनोड बॅटरीशी जोडलेले असतात किंवा $+V_{cc}$ आणि LED डिस्प्ले अंक दाखवतात जेव्हा वैयक्तिक कॅथोड्सना काही 'लो' सिग्नल पुरवले जातात.

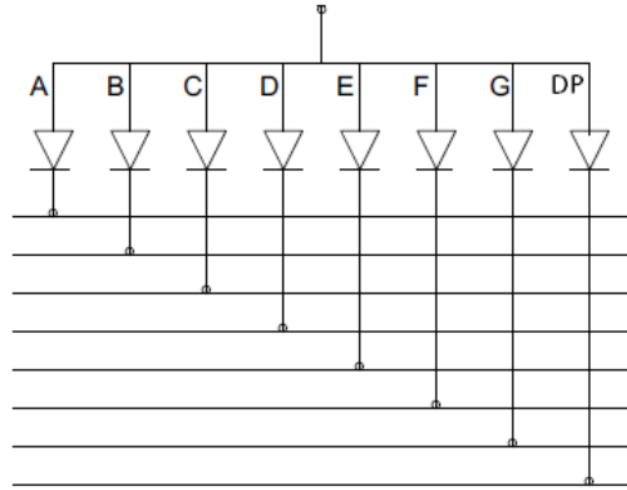


Figure 3.29 Common Anode Types

BCD ते 7 सेगमेंट डिस्प्ले डिकोडर: (कॉमन एनोड(Anode) प्रकार)

बीसीडी ते सात सेगमेंट डिकोडरमध्ये चार इनपुट लाइन (ए, बी, सी आणि डी) आणि 7 आउटपुट लाइन (ए, बी, सी, डी, ई, एफ आणि जी) आहेत, हे आउटपुट सात सेगमेंट एलईडी डिस्प्लेला दिले जाते जे प्रदर्शित करते. इनपुटवर अवलंबून दशांश संख्या. सात विभागातील डिस्प्ले एलईडीच्या विविध विभागांना थेट व्होल्टेज पुरवून कार्य करत नाही. प्रथम, आपली दशांश संख्या त्याच्या बीसीडी समतुल्य सिग्नलमध्ये बदलली जाते, त्यानंतर बीसीडी ते सात सेगमेंट डिकोडर त्या सिग्नलला त्या फॉर्ममध्ये रूपांतरित करते जे सात खंड डिस्प्लेला दिले जाते.

अथवा (OR)

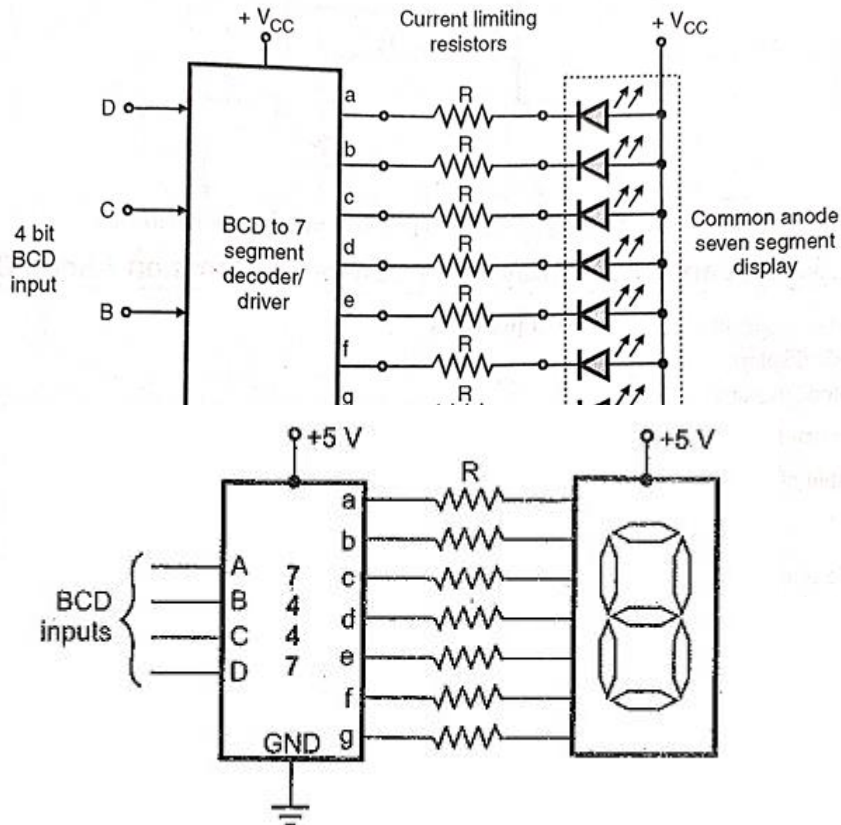


Figure 3.30 : BCD to 7 segment display decoder common anode

Table No. 3.9 - Truth Table BCD to 7 Segment Decoder Common Anode

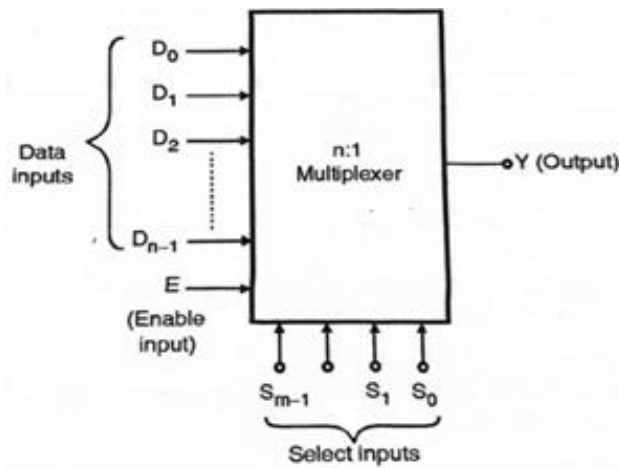
DECIMAL	INPUTS				OUTPUTS						
	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	1	0	0	1	1	1	1
2	0	0	1	0	0	0	1	0	0	1	0
3	0	0	1	1	0	0	0	0	1	1	0
4	0	1	0	0	1	0	0	1	1	0	0
5	0	1	0	1	0	1	0	0	1	0	0
6	0	1	1	0	0	1	0	0	0	0	0
7	0	1	1	1	0	0	0	1	1	1	1
8	1	0	0	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	0	1	0	0

3.5 मल्टीप्लेक्सर (Multiplexer), डी-मल्टीप्लेक्सर (De-multiplexer)

3.5.1 मल्टीप्लेक्सर (Multiplexer):

डिजिटल लॉजिक सर्किट जे अनेक डेटा इनपुट स्वीकारते आणि आउटपुटमधून फक्त एकालाच प्रवाहित करण्याची परवानगी देते त्याला मल्टीप्लेक्सर किंवा MUX म्हणतात.

आधीच नमूद केल्याप्रमाणे, मल्टीप्लेक्सर, ज्याला MUX असेही संबोधले जाते, हे एक कॉम्बिनेशन लॉजिक सर्किट आहे जे एकाधिक इनपुट सिग्नल स्वीकारण्यासाठी आणि त्यापैकी फक्त एक आउटपुट लाइनद्वारे हस्तांतरित करण्यासाठी डिझाइन केलेले आहे. सोप्या शब्दात, मल्टीप्लेक्सर हे एक डिजिटल लॉजिक डिवाइस आहे जे अनेक इनपुट पैकी एक (1-आउट-ऑफ-N) इनपुट डेटा स्रोत निवडते आणि निवडलेला डेटा एका आउटपुट लाइनवर प्रसारित करते.



मल्टीप्लेक्सरला डेटा सिलेक्टर असेही म्हणतात कारण तो अनेकांमधून एक निवडतो. मल्टीप्लेक्सरच्या बाबतीत, आउटपुट लाइनमधून प्रवाहित होण्यासाठी इच्छित डेटा इनपुटची निवड SELECT लाईन्सच्या मदतीने नियंत्रित केली जाते. आकृती 3.31 मधील मल्टीप्लेक्सरच्या ब्लॉक डायग्राममध्ये D0, D1, ... Dn-1, म्हणजे (2^N) इनपुट रेषा आहेत आणि "m" या निवडक रेषा $(2^m=N)$ आहेत. या निवडक रेषा आउटपुटवर कोणते इनपुट राउट करायचे हे ठरवतील. म्हणून, मल्टीप्लेक्सर मल्टी-

पोझिशन स्विच म्हणून कार्य करते ज्याचे ऑपरेशन डिजिटल सिग्नलद्वारे नियंत्रित केले जाते. कोणते डेटा इनपुट, आउटपुट लाइनवर स्विच केले जाईल हे निर्धारित करण्यासाठी हे डिजिटल नियंत्रण सिग्नल निवडलेल्या ओळीवर लागू केले जातात.

3.5.2: मल्टीप्लेक्सचे प्रकार

इनपुट डेटा लाइन्स आणि सिलेक्ट लाइन्सवर आधारित, मल्टीप्लेक्सर अनेक प्रकारचे असू शकतात.

1. 2:1 मल्टीप्लेक्सर
2. 4:1 मल्टीप्लेक्सर
3. 8:1 मल्टीप्लेक्सर
4. 16:1 मल्टीप्लेक्सर

1) 2:1 मल्टीप्लेक्सर(2:1 MUX)

2:1 मल्टीप्लेक्सर मूलभूत दोन इनपुट मल्टीप्लेक्सर आहे. ज्यामध्ये D_0 आणि D_1 म्हणून नियुक्त केलेल्या दोन डेटा इनपुट लाइन आहेत, एक निवडक डेटा सिलेक्ट लाइन S ने दर्शविली आहे आणि एक आउटपुट लाइन Y द्वारे दर्शविली आहे.

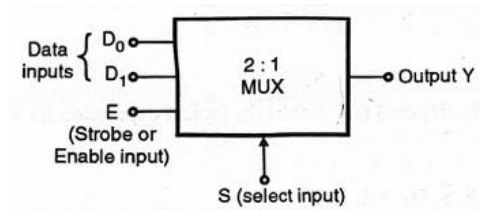


Figure 3.32 (a) 2:1 Multiplexers

Table No. 3.11 Truth Table 2:1 Multiplexer

Input		Output
E	S	D
0	x	0
1	0	D_0
1	1	D_1

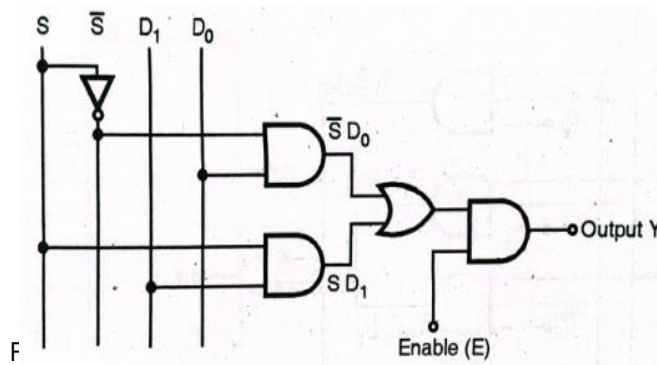


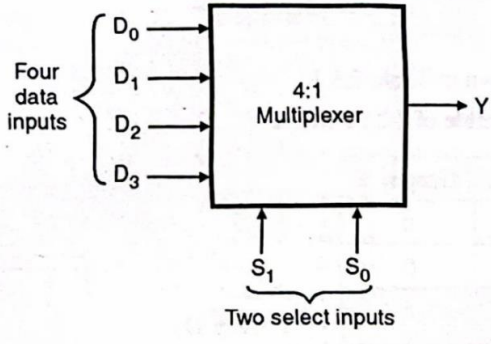
Figure 3.32 (b) 2:1 Multiplexers Circuit diagram

2) 4:1 मल्टीप्लेक्सर(4:1 MUX)

4:1 मल्टीप्लेक्सर मध्ये 4 डेटा इनपुट लाइन, 2 निवडक (data Select) ओळी आणि 1 आउटपुट लाइन आहे.

येथे, D₀, D₁, D₂, आणि D₃ या चार डेटा इनपुट लाइन आहेत, S₀ आणि S₁ या दोन निवडक रेषा आहेत आणि Y ही आउटपुट लाइन आहे. अशा प्रकारे, S₀ आणि S₁ सिलेक्ट लाईन्सवर लागू केलेल्या डिजिटल सिग्नलची लॉजिक लेव्हल कोणते इनपुट आउटपुट लाइनवर जाईल हे निर्धारित करतात.

Table No. 3.12 - Truth Table 4:1 Multiplexer



Input			Output
E	S ₁	S ₀	D
0	x	x	0
1	0	0	D ₀
1	0	1	D ₁
1	1	0	D ₂
1	1	1	D ₃

Figure 3.34 4:1 Multiplexers

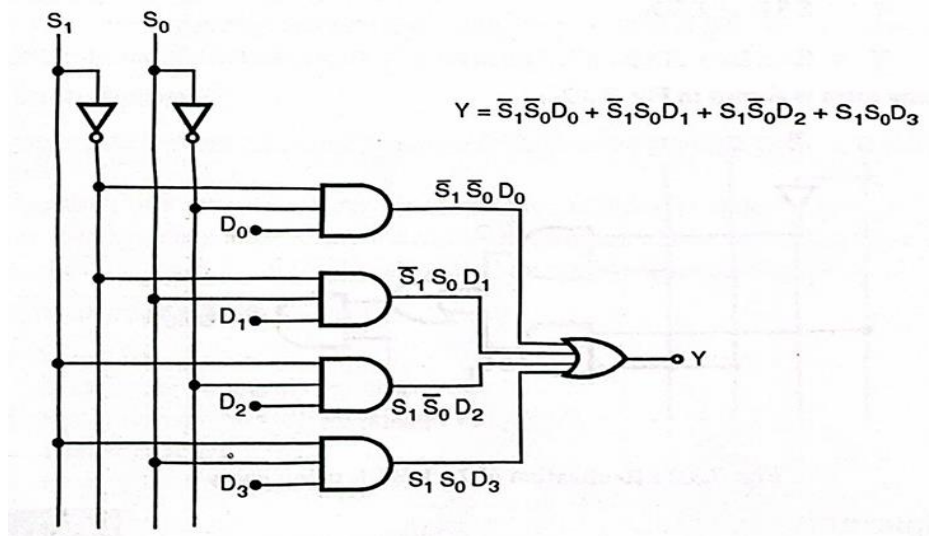
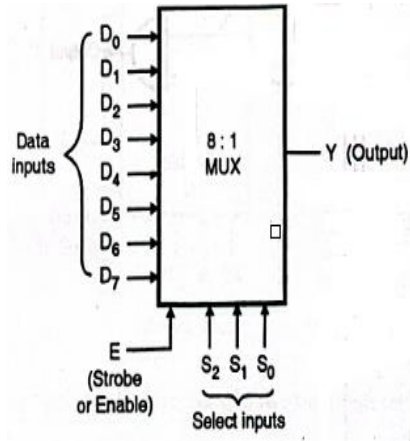


Figure 3.37 : 4:1 Multiplexer Logic Circuit

3) 8:1 मल्टीप्लेक्सर(8:1 MUX):

8:1 मल्टीप्लेक्सर एक मल्टीप्लेक्सर आहे ज्यामध्ये 8 डेटा इनपुट लाइन, 3 निवडक(data Select) ओळी आणि 1 आउटपुट लाइन आहे. येथे, D₀, D₁, D₂, D₃, D₄, D₅, D₆, D₇ या 8 डेटा इनपुट लाइन आहेत, S₀, S₁, S₂ या 3 निवडक रेषा(select lines) आहेत आणि Y ही आउटपुट लाइन आहे. अशा प्रकारे, S₀, S₁, S₂ सिलेक्ट लाईन्सवर लागू केलेल्या डिजिटल सिग्नलची लॉजिक लेव्हल कोणते इनपुट आउटपुट लाइनवर जाईल हे निर्धारित करतात.



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

Figure 3.38 : 8:1 Multiplexers

Table No. 3.13 - Truth Table 8:1 Multiplexer

$$Y = \bar{S}_2 \bar{S}_1 \bar{S}_0 D_0 + \bar{S}_2 \bar{S}_1 S_0 D_1 + \bar{S}_2 S_1 \bar{S}_0 D_2 + \bar{S}_2 S_1 S_0 D_3 + S_2 \bar{S}_1 \bar{S}_0 D_4 + S_2 \bar{S}_1 S_0 D_5 + S_2 S_1 \bar{S}_0 D_6 + S_2 S_1 S_0 D_7$$

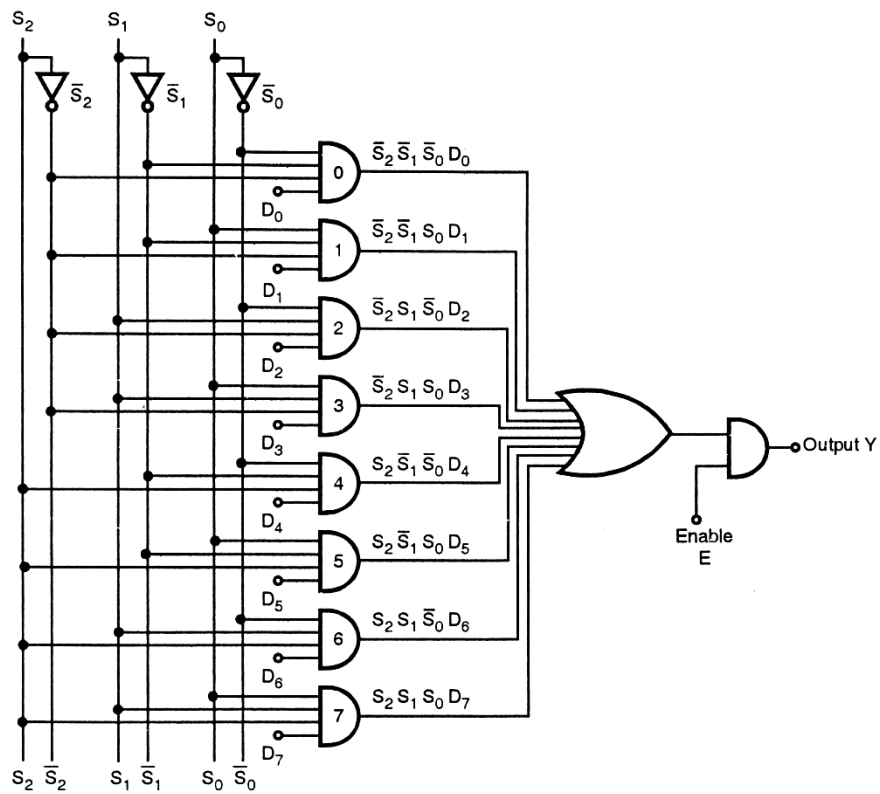


Fig 3.39 8:1 Multiplexer Logic Circuit

4) 16:1 मल्टीप्लेक्सर(16:1 MUX):

16:1 मल्टीप्लेक्सर एक मल्टीप्लेक्सर आहे ज्यामध्ये 16 डेटा इनपुट लाइन, 4 निवडक(data Select) ओळी आणि 1 आउटपुट लाइन आहे .येथे, D0- D15 या 16 डेटा इनपुट लाइन आहेत, S0,S1,S2,S3 या 4 निवडक रेषा(select lines) आहेत

आणि Y ही आउटपुट लाइन आहे. अशा प्रकारे, S₀, S₁, S₂, S₃ सिलेक्ट लाईन्सवर लागू केलेल्या डिजिटल सिग्नलची लॉजिक लेव्हल कोणते इनपुट आउटपुट लाइनवर जाईल हे निर्धारित करतात.

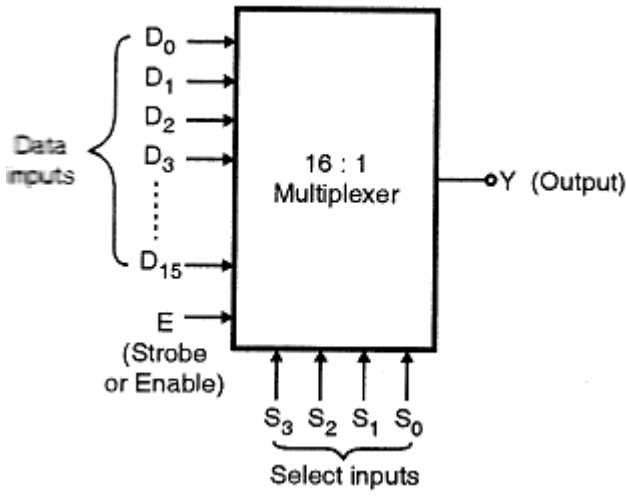


Fig 3.40 16:1 Multiplexer

Enable	Select inputs				Output
E	S ₃	S ₂	S ₁	S ₀	Y
0	0	0	0	0	D ₀
1	0	0	0	1	D ₁
1	0	0	1	0	D ₂
1	0	0	1	1	D ₃
1	0	1	0	0	D ₄
1	0	1	0	1	D ₅
1	0	1	1	0	D ₆
1	0	1	1	1	D ₇
1	1	0	0	0	D ₈
1	1	0	0	1	D ₉
1	1	0	1	0	D ₁₀
1	1	0	1	1	D ₁₁
1	1	1	0	0	D ₁₂
1	1	1	0	1	D ₁₃
1	1	1	1	0	D ₁₄
1	1	1	1	1	D ₁₅

Table 3.12 16:1 Multiplexer Truth Table

3.5.3 मल्टिप्लेक्सर ट्री (MULTIPLEXER TREE)

1) 4:1 मल्टिप्लेक्सर आणि 2:1 वापरून 8:1 मल्टिप्लेक्सर डिझाइन करणे:

लोअर ऑर्डर मल्टिप्लेक्सर वापरून 8:1 मल्टिप्लेक्सर करू शकतो. 8:1 मल्टिप्लेक्स तयार करण्यासाठी दोन 4:1 आणि एक 2:1 मल्टिप्लेक्सची आवश्यकता आहे. 4:1 मल्टिप्लेक्सरला दोन निवडक रेषा, चार इनपुट आणि एक आउटपुट आहे. 2:1 मल्टिप्लेक्सरला फक्त एक निवडक रेषा आहे. 8 डेटा इनपुट मिळवण्यासाठी दोन 4:1 मल्टिप्लेक्सची आवश्यकता आहे.

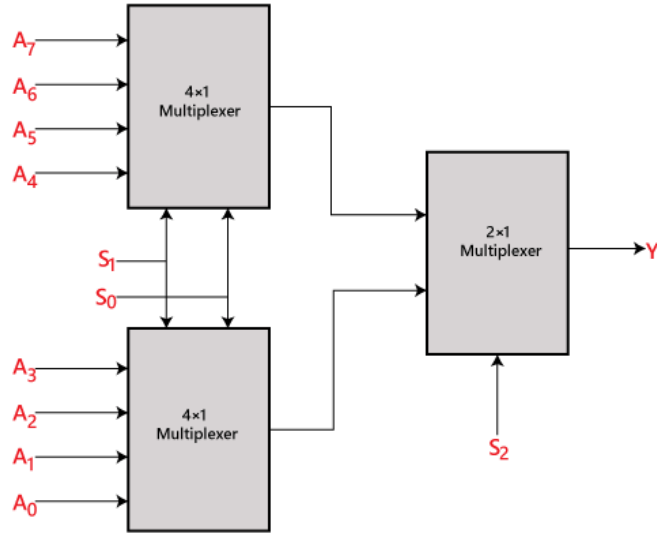


Figure 3.41 8:1 Multiplexer Tree

Table No. 3.15 - Truth Table 8:1 Multiplexer

SELECT INPUTS			OUT PUT	
S_2	S_1	S_0	Y	
0	0	0	D_0	MUX-1 ENABLE
0	0	1	D_1	
0	1	0	D_2	
0	1	1	D_3	
1	0	0	D_4	MUX-2 ENABLE
1	0	1	D_5	
1	1	0	D_6	
1	1	1	D_7	

2) 16:1 मल्टिप्लेक्सर 4:1 मल्टिप्लेक्सर आणि 2:1 वापरून डिझाइन करणे:

लोअर ऑर्डर मल्टिप्लेक्सर वापरून 8:1 मल्टिप्लेक्सर करू शकतो. 8:1 मल्टिप्लेक्स तयार करण्यासाठी दोन 4:1 आणि एक 2:1 मल्टिप्लेक्सची आवश्यकता आहे. 4:1 मल्टिप्लेक्सरला दोन निवडक रेषा, चार इनपुट आणि एक आउटपुट आहे. 2:1 मल्टिप्लेक्सरला फक्त एक निवडक रेषा आहे. 8 डेटा इनपुट मिळवण्यासाठी दोन 4:1 मल्टिप्लेक्सची आवश्यकता आहे.

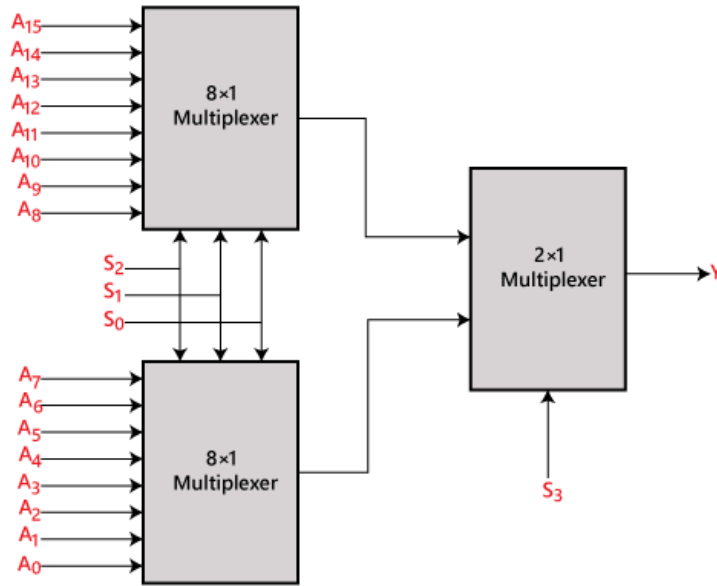


Figure 3.42 16:1 Multiplexer Tree

Table 3.16 Truth Table

SELECT INPUTS				MUX OUTPUT				FINAL OUTPUT Y	
S ₃	S ₂	S ₁	S ₀	Y ₁	Y ₂	Y ₃	Y ₄		
0	0	0	0	D ₀	D ₄	D ₈	D ₁₂	D ₀	S S=00 MUX-5 SELECTS Y
0	0	0	1	D ₁	D ₅	D ₉	D ₁₃	D ₁	
0	0	1	0	D ₂	D ₆	D ₁₀	D ₁₄	D ₂	
0	0	1	1	D ₃	D ₇	D ₁₁	D ₁₅	D ₃	
0	1	0	0	D ₀	D ₄	D ₈	D ₁₂	D ₄	S S=01 MUX-5 SELECTS Y
0	1	0	1	D ₁	D ₅	D ₉	D ₁₃	D ₅	
0	1	1	0	D ₂	D ₆	D ₁₀	D ₁₄	D ₆	
0	1	1	1	D ₃	D ₇	D ₁₁	D ₁₅	D ₇	
1	0	0	0	D ₀	D ₄	D ₈	D ₁₂	D ₈	S S=10 MUX-5 SELECTS Y
1	0	0	1	D ₁	D ₅	D ₉	D ₁₃	D ₉	
1	0	1	0	D ₂	D ₆	D ₁₀	D ₁₄	D ₁₀	
1	0	1	1	D ₃	D ₇	D ₁₁	D ₁₅	D ₁₁	
1	1	0	0	D ₀	D ₄	D ₈	D ₁₂	D ₁₂	S S=11 MUX-5 SELECTS Y
1	1	0	1	D ₁	D ₅	D ₉	D ₁₃	D ₁₃	
1	1	1	0	D ₂	D ₆	D ₁₀	D ₁₄	D ₁₄	
1	1	1	1	D ₃	D ₇	D ₁₁	D ₁₅	D ₁₅	

3.5.4 : मल्टीप्लेक्सर वापरून लॉजिक समीकरणाची (Logical Expression) अंमलबजावणी करणे.

स्टेप्स 1: दिलेल्या बुलियन एक्सप्रेशन मधील नंबर ऑफ व्हेरिएबल्स ओळखा (m). त्यानंतर $2^m = N$ (m = निवड रेषा (Select Lines), N = इनपुट लाइन) फॉर्मूला वापरून मल्टीप्लेक्सर ची निवड करा.

स्टेप्स 2 : किमान-अटीशी संबंधित इनपुट (min terms) लॉजिक 1 शी आणि उर्वरित इनपुट लॉजिक 0 (ग्राउंडेड) शी जोडा.

1) दिलेल्या लॉजिक एक्सप्रेसन ची मल्टीप्लेक्सर वापरून अंमलबजावणी करा.

$$Y(A,B,C,D) = \sum m(2,3,4,5,8,9,12)$$

स्टेप्स 1: दिलेल्या लॉजिक एक्सप्रेसन 4 व्हेरिएबल्स आहेत म्हणून 4 निवड रेषा (Select Lines) आहेत. $2n = 16$, म्हणून 16 इनपुट लाइन, 4 निवड रेषा आणि 1 आउटपुट असलेला म्हणजेच 16:1 मल्टीप्लेक्सर वापरा.

स्टेप्स 2: किमान-अटींशी (min term) संबंधित इनपुट (2,3,4,5,8,9,12) लॉजिक 1 शी आणि उर्वरित इनपुट लॉजिक 0 (ग्राउंडेड) शी जोडलेले आहेत.

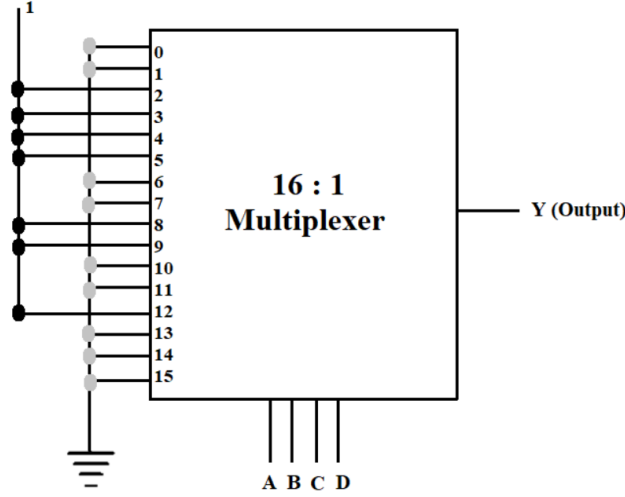


Figure 3.43 Logical expression using multiplexers

3.5.5: डी-मल्टीप्लेक्सर (D-Multiplexer)

डी-मल्टीप्लेक्सर हे एक कॉम्बिनेशनल सर्किट आहे ज्यामध्ये फक्त 1 इनपुट लाइन आणि आउटपुट लाइन N ($N=2^m$) आहेत. फक्त, मल्टीप्लेक्सर एक सिंगल-इनपुट आणि मल्टी-आउटपुट कॉम्बिनेशनल सर्किट आहे. सिंगल इनपुट लाइन्समधून माहिती प्राप्त होते आणि आउटपुट लाइनवर निर्देशित केली जाते. निवड ओळींच्या (select lines m) मूल्यांच्या आधारावर, इनपुट यापैकी एका आउटपुटशी कनेक्ट केले जाईल. डी-मल्टीप्लेक्सर मल्टीप्लेक्सरच्या विरुद्ध आहे. डी-मल्टीप्लेक्सरला डी-मक्स (D EMUX) देखील मानले जाते.

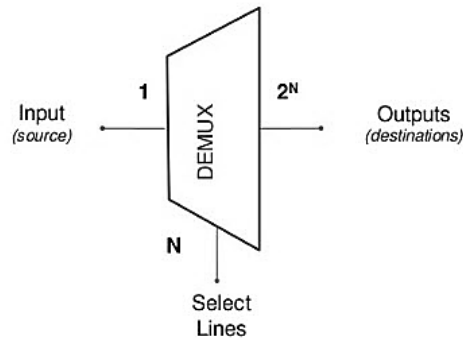


Figure 3.44 Demultiplexer block diagram

डी-मल्टीप्लेक्सरचे विविध प्रकार आहेत जे खालीलप्रमाणे आहेत

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

1) 1:2 डी-मल्टीप्लेक्सर(1:2 MUX):

1:2 डी-मल्टीप्लेक्सर , फक्त दोन आउटपुट आहेत म्हणजे, Y_0 , आणि Y_1 , 1 निवड रेषा, म्हणजे, S_0 , आणि सिंगल इनपुट, म्हणजे, D . निवड मूल्याच्या आधारावर, इनपुट, आउटपुटपैकी एकाशी कनेक्ट केलेले असेल .

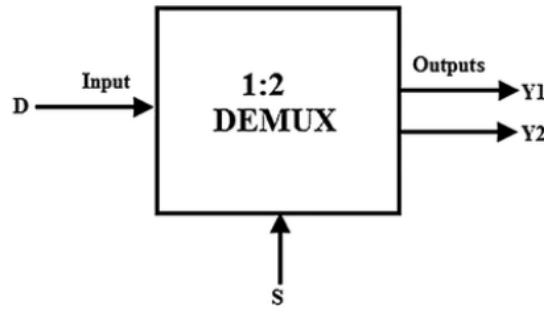


Figure 3.45 1:2 Demultiplexer block diagram

Table No. 3.17 – Truth Table 1:2D-Multiplexer

Select	Input	Outputs	
S	D	Y_2	Y_1
0	0	0	0
0	1	0	1
1	0	0	0
1	1	1	0

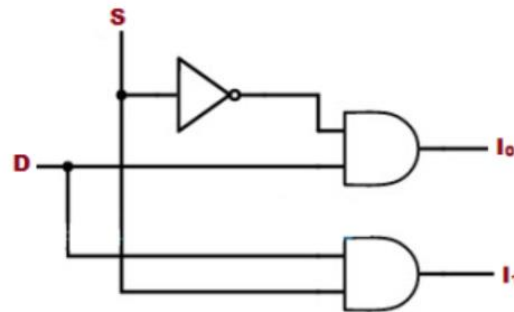


Figure 3.46 1:2 DeMultiplexer Logic Circuit

2) 1:4 डी-मल्टीप्लेक्सर (1:4 DMUX):

1:4 डी-मल्टीप्लेक्सरला, फक्त चार आउटपुट आहेत म्हणजे, Y_0 , आणि Y_1 , Y_2 , Y_3 , निवड रेषा, दोन म्हणजे, S_0 , S_1 , आणि सिंगल इनपुट, म्हणजे, D . निवड मूल्याच्या आधारावर, इनपुट, आउटपुटपैकी एकाशी कनेक्ट केलेले असेल

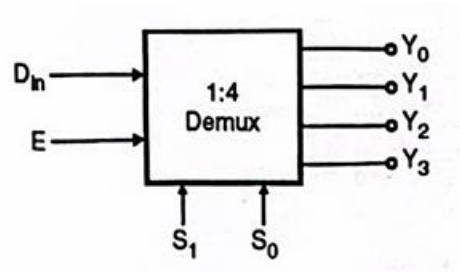


Figure 3.47 1:4 Demultiplexer block diagram

Table No. 3.18 - Truth Table 1:4 DeMultiplexer

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

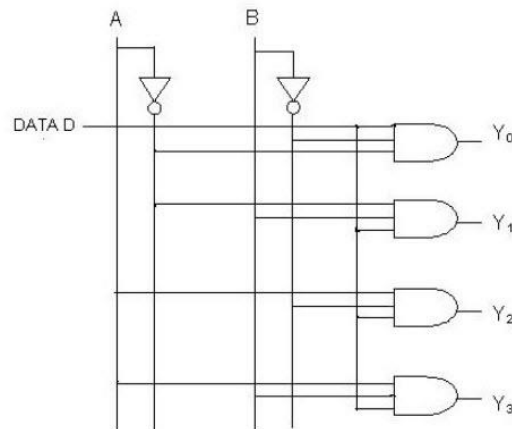


Figure 3.48 1:4 DeMultiplexer Logic Circuit

3) 1:8 डी-मल्टीप्लेक्सर(1:8 DMUX)

1:8 डी-मल्टीप्लेक्सरला , आठ आउटपुट आहेत म्हणजे, $Y_0, Y_1, Y_2, \dots, Y_7$, निवड रेषा, तीन म्हणजे, S_0, S_1, S_2 आणि सिंगल इनपुट, म्हणजे, D. निवड मूल्याच्या आधारावर, इनपुट, आउटपुटपैकी एकाशी कनेक्ट केलेले असेल .

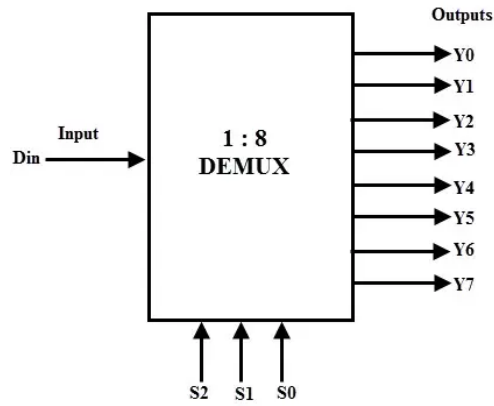


Figure 3.50 1:8 Demultiplexer block diagram

Table No. 3.19 - Truth Table 1:8 D-Multiplexer

Data Input	Select Inputs			Outputs							
D	S ₂	S ₁	S ₀	Y ₇	Y ₆	Y ₅	Y ₄	Y ₃	Y ₂	Y ₁	Y ₀
D	0	0	0	0	0	0	0	0	0	0	D
D	0	0	1	0	0	0	0	0	0	D	0
D	0	1	0	0	0	0	0	0	D	0	0
D	0	1	1	0	0	0	0	D	0	0	0
D	1	0	0	0	0	0	D	0	0	0	0
D	1	0	1	0	0	D	0	0	0	0	0
D	1	1	0	0	D	0	0	0	0	0	0
D	1	1	1	D	0	0	0	0	0	0	0

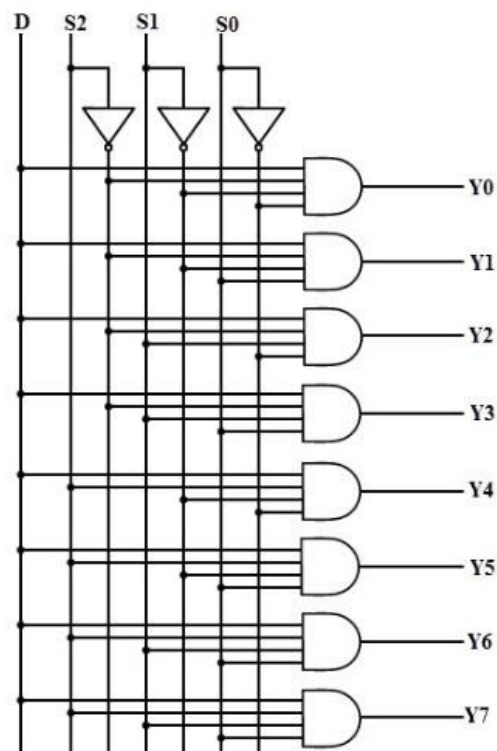


Figure 3.51 1:8 D-Multiplexer Logic Circuit

4) 1:16 डी-मल्टीप्लेक्सर(1:16DMUX)

1:16 डी-मल्टीप्लेक्सरला , सोळा आउटपुट आहेत म्हणजे, $Y_0, Y_1, Y_2, \dots, Y_{15}$, निवड रेषा, चारम्हणजे, S_0, S_1, S_2, S_3 आणि सिंगल इनपुट, म्हणजे, D. निवड मूल्याच्या आधारावर, इनपुट, आउटपुटपैकी एकाशी कनेक्ट लेले .

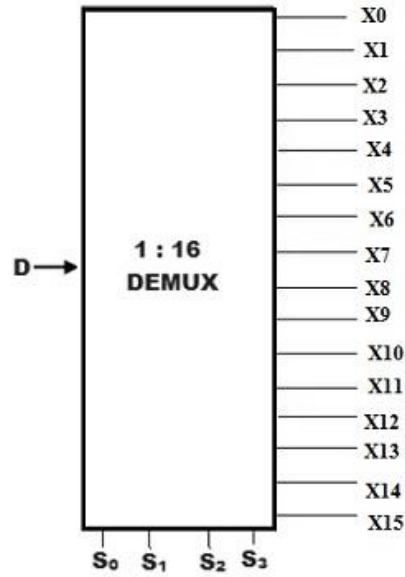


Figure 3.52 1:16 Demultiplexer block diagram

Table No. 3.20: Truth Table of 1:16 D-Multiplexer

SELECT I/P				OUTPUT															
S_3	S_2	S_1	S_0	Y_{15}	Y_{14}	Y_{13}	Y_{12}	Y_{11}	Y_{10}	Y_9	Y_8	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DIN
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DIN	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DIN	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	DIN	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	DIN	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0	0	0	DIN	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	DIN	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	DIN	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	DIN	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	DIN	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	DIN	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	DIN	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	DIN	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	DIN	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	DIN	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	DIN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

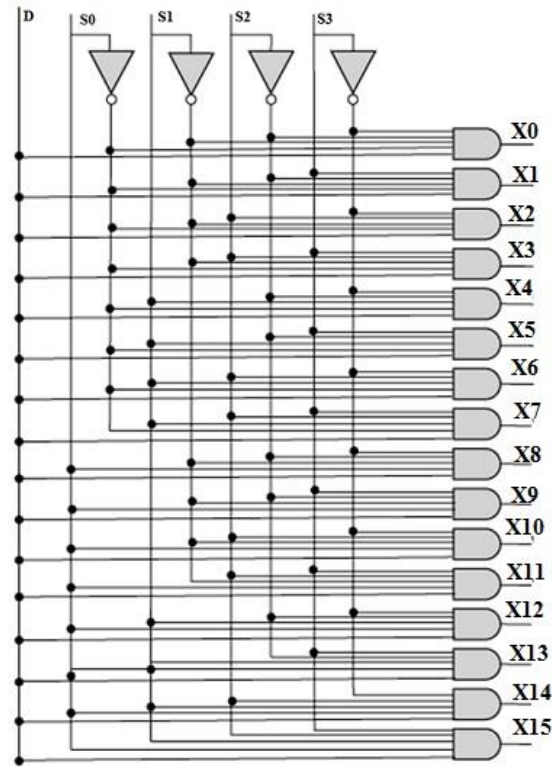


Figure 3.55 1:16 DeMultiplexer Logic Circuit

3.5.6: डी मल्टीप्लेक्सर ट्री: (D Multiplexer TREE)

1) 1:2 डी-मल्टीप्लेक्सर वापरून 1:4 डी-मल्टीप्लेक्सर डिझाइन करणे:

लोअर ऑर्डर 1:2 डी-मल्टीप्लेक्सर वापरून आम्ही 1:4 डी-मल्टीप्लेक्सर लागू करू शकतो. 1:4 डी-मल्टीप्लेक्सर लागू करण्यासाठी, आम्हाला दोन 1:2 डी-मल्टीप्लेक्सर गरज आहे. 1:4 डी-मल्टीप्लेक्सरला 2 निवड रेषा, 1 इनपुट आणि 4 आउटपुट आहेत.

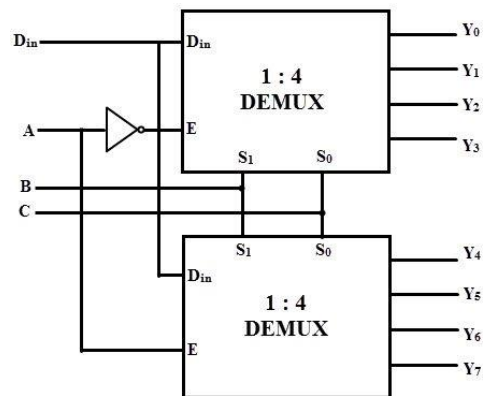


Fig 3.56 1:8 Demultiplexer Tree

2) डी-मल्टीप्लेक्सर वापरून 1:8 डी-मल्टीप्लेक्सर डिझाइन करणे .

लोअर ऑर्डर 1:4 डी-मल्टीप्लेक्सर वापरून आम्ही 1:8 डी-मल्टीप्लेक्सर लागू करू शकतो. 1:8 डी-मल्टीप्लेक्सर लागू करण्यासाठी, आम्हाला दोन 1:4 डी-मल्टीप्लेक्सर गरज आहे. 1:8 डी-मल्टीप्लेक्सरला 3 निवड रेषा, 1 इनपुट आणि 8 आउटपुट आहेत.

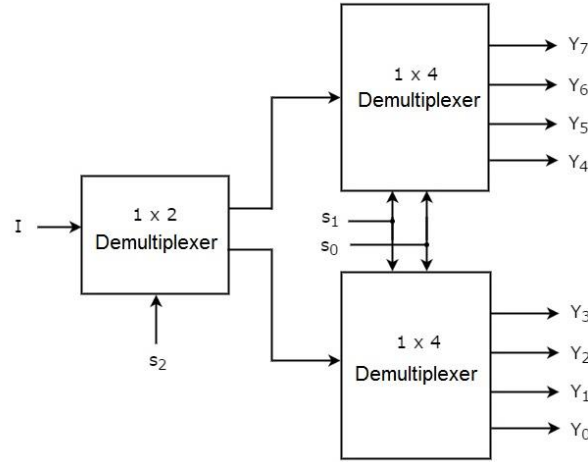


Figure 3.57 1:8 DeMultiplexer Tree

5.7: डीमल्टीप्लेक्सरचा डीकोडर म्हणून वापर करणे.

डीकोडर म्हणून डीमल्टीप्लेक्सर कसे वापरले जाऊ शकते याचे वर्णन करा.

डीकोडर म्हणून डीमल्टीप्लेक्सरचा वापर करणे शक्य आहे.

1:8 डीमल्टीप्लेक्सरचा 3:8 डीकोडर म्हणून कसा वापर करायचा ते पाहू.

- 1) 1:8 डीमल्टीप्लेक्सर मध्ये D_{in} डेटा इनपुट आहे, S_0, S_1, S_2 निवडक रेषा (select lines) आहेत आणि $Y_0 - Y_7$ आउटपुट आहेत
- 2) 3:8 डीकोडर म्हणून वापर करण्यासाठी आपल्याला S_0, S_1, S_2 निवडक रेषा (select lines)
- 3) इनपुट म्हणून वापरावे लागतील.
- 4) D_{in} लॉजिक 1 ला जोडावा लागेल आणि $Y_0 - Y_7$ चा आऊटपुट म्हणून वापर करावा लागेल.

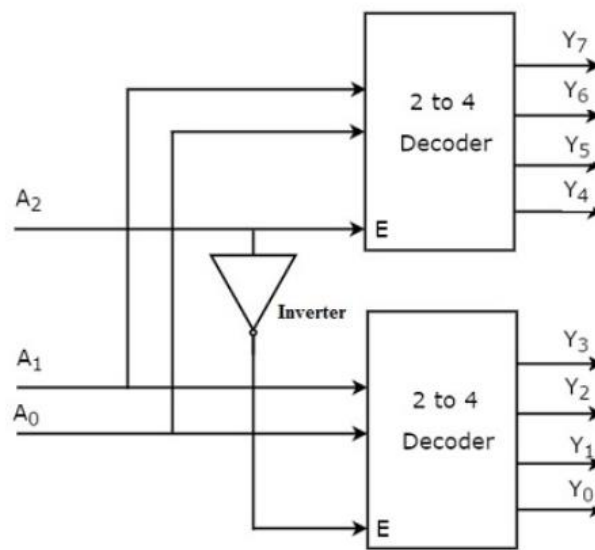


Figure 3.58 1:8 Demultiplexer as 3:8 decoder

अध्याय

- 1) K-map च्या संदर्भात किमान-टर्म आणि कमाल-टर्म ची व्याख्या लिहा.
- 2) मल्टीप्लेक्सर ची व्याख्या लिहा आणि तर्क चिन्ह (SYMBOLIC REPRESENTATION) काढा. डिमल्टीप्लेक्सर ची व्याख्या लिहा आणि तर्क चिन्ह (SYMBOLIC REPRESENTATION) काढा.
- 3) मल्टीप्लेक्सरचे आणि डी-मल्टीप्लेक्सरचे विविध प्रकार लिहा.
- 4) एन्कोडर डिकोडर ची व्याख्या लिहा.
- 5) SOP आणि POS फॉर्म ची व्याख्या लिहा.
- 6) K-map सरलीकरण वापरून हाफ अँडरची लॉजिक आकृती काढा. आणि तृथ टेबल लिहा
- 7) खालील एक्सप्रेशन स्टँडर्ड SOP मध्ये रूपांतरित करा.

$$Y = \bar{A} + B\bar{C}D \quad 2) Y = A + B$$
- 8) खालील एक्सप्रेशन स्टँडर्ड POS मध्ये रूपांतरित करा.

$$Y = (\bar{A} + B)(B + \bar{C})(\bar{A} + \bar{C}) \quad 2) Y = (A + B)(A + C)$$
- 9) 4:1 mux वापरून 16:1 MUX त्रू काढा.
- 10) 1:8 डिमल्टीप्लेक्सर वापरून 1:16 डिझाईन डिमल्टीप्लेक्सर काढा. सत्य सारणी तक्ता देखील लिहा.
- 11) K-map वापरून खालील समीकरण(Expression) लहान(Minimise) करा.

$$F(A, B, C, D) = \sum m(0, 1, 2, 4, 5, 7, 8, 9, 10)$$
 ii) मूलभूत गेट्स(BASIC) वापरून लहान केलेले समीकरण (minimize expression) विकसित (Design) करा.
- 12) K-map वापरून खालील समीकरण(Expression) लहान(Minimise) करा.

$$Y = \sum m(0, 2, 5, 7, 8, 10, 13, 15)$$
 ii) मूलभूत गेट्स(BASIC) वापरून लहान केलेले समीकरण (minimize expression) विकसित (Design) करा.
- 13) K-map वापरून खालील समीकरण(Expression) लहान(Minimise) करा.

$$Y = \pi M(0, 1, 2, 3, 4, 5, 7, 8, 9, 12, 13)$$
 ii) मूलभूत गेट्स(BASIC) वापरून लहान केलेले समीकरण (minimize expression) विकसित (Design) करा.
- 14) दोन हाफ अँडर वापरून फुल अँडर आकृती काढा.
- 15) 8 : 1 मल्टीप्लेक्सर वापरून 32 : 1 मल्टीप्लेक्सर डिझाईन करा.
- 16) k-map सरलीकरण वापरून फुल अँडरची(Full Adder) लॉजिक आकृती(Logic diagram) काढा आणि तृथ टेबल लिहा.

लघु प्रकल्प (Micro Project):

- 1) BCD ते 7 - सेगमेंट डिकोडर सर्किट आय सी (IC) वापरून ब्रेड बोर्डवर तयार करणे व ते वेगवेगळ्या इनपुट साठी तपासून घेणे.
- 2) 4 बीट अँडर सर्किट आय-सी वापरून ब्रेड बोर्डवर तयार करणे व ते वेगवेगळ्या इनपुट साठी तपासून घेणे.
- 3) 1:4 व 1:2 DE-MUX वापरून 1:8 DEMUX सर्किट तयार करणे.

संदर्भ: References

Sr.No	Author Title	Title	Publisher with ISBN Number
1	Jain R.P	Modern Digital Electronics	McGraw-Hill Publishing, New Delhi, 2009 ISBN:9780070669116
2	Anand Kumar	Fundamentals of Digital Circuits	PHI learning Private limited, ISBN:978-81 203-5268-1

Sr.No	Author Title	Title	Publisher with ISBN Number
3	3 Salivahanan S, Arivazhagan S	Digital Circuits and Design	Vikas Publishing House, New Delhi, 2013 ISBN: 9789325960411 alivahanan S, Arivazhagan
4	Puri.V.K	Digital Electronics	McGraw-Hill Publishing, New Delhi, 2016 ISBN:97800746331751
5	Malvino A.P Donald .P. Leach	Digital Principles	McGraw-Hill Education, New Delhi ISBN:9789339203405
6	Anil.K.Maini	Digital Electronics: Principles, Devices and Applications	Wiley India, Delhi, 2007

संकेतस्थळ (Websites):

https://onlinecourses.nptel.ac.in/noc19_ee51/preview

https://www.electronics-tutorials.ws/combinational/comb_2.html

<https://www.javatpoint.com/combinational-logic-circuits-in-digital-electronics>

युनिट-4: सिक्वेन्शियल लॉजिक सर्किट्स (Sequential Logic Circuits)

विषय निष्पत्ती (Course Outcome):

फ्लिप फ्लॉपचा वापर करून सीक्वेन्शियल (Sequential) सर्किट तयार करणे

Develop sequential logic circuits using Flip-flops

युनिट निष्पत्ती (Unit Outcomes) :

4.a लॅच आणि फ्लिप फ्लॉपमध्ये फरक स्पष्ट करणे.

Differentiate between Latch and Flip Flop.

4.b मूलभूत मेमरी सेल वर्णन करा आणि दिलेल्या डिजिटल सर्किटसाठी संबंधित ट्रिगरिंग तंत्र वापरणे.

Explain basic memory cell and use relevant triggering technique for the given digital circuit.

4.c दिलेल्या फ्लिप फ्लॉप्स, फ्लिप फ्लॉप्सच्या अनुपयोगासाठी सत्य सारण्यांचे वर्णन करणे.

Describe the truth tables for the given Flip flops, applications of Flip flops.

4.d विशिष्ट प्रकारच्या (काउंटर) गणकाची रचना करण्यासाठी दिलेल्या प्रकारचे फ्लिप फ्लॉप आणि त्याचे उत्तेजना सारणीचा (excitation table) वापर करणे.

Use the given type of flip flop and its excitation table to design specific type of counter.

4.5 टाईमिंग डायग्रामच्या मदतीने विनिर्दिष्ट (विचारलेल्या) शिफ्ट रजिस्टरच्या कार्याचे वर्णन करा.

Describe the working of specified shift register with the help of timing diagram.

4.6 फ्लिप फ्लॉप्स वापरून विचारलेले मोड्युलो-एन काउंटर/ गणक (Modulo-N counter) डिझाइन तयार करणे.

Design specified modulo-N counter using Flip flops.

4.7 दिलेल्या फ्लिप-फ्लॉपचा वापर करून रिंग/ट्विस्टेड रिंग (काउंटर) गणक डिझाइन तयार करणे.

Design Ring /Twisted ring counter using given Flip-Flop.

परिचय :

सीक्वेन्शियल सर्किट (sequential circuit) एक लॉजिकल सर्किट आहे, जेथे आउटपुट इनपुट सिग्नलच्या वर्तमान मूल्यावर तसेच मागील इनपुटच्या अनुक्रमांवर अवलंबून असते. कॉम्बिनेशनल सर्किट हे फक्त वर्तमान इनपुटचे कार्य आहे. सीक्वेन्शियल सर्किट हे कॉम्बिनेशनल सर्किट आणि स्टोरेज घटकांचे संयोजन आहे. सीक्वेन्शियल सर्किट्स वर्तमान इनपुट व्हेरिएबल्स आणि मागील इनपुट व्हेरिएबल्स वापरतात जे संग्रहित केले जातात आणि पुढील क्लॉक पल्स सायकलवर सर्किटला डेटा प्रदान करतात.

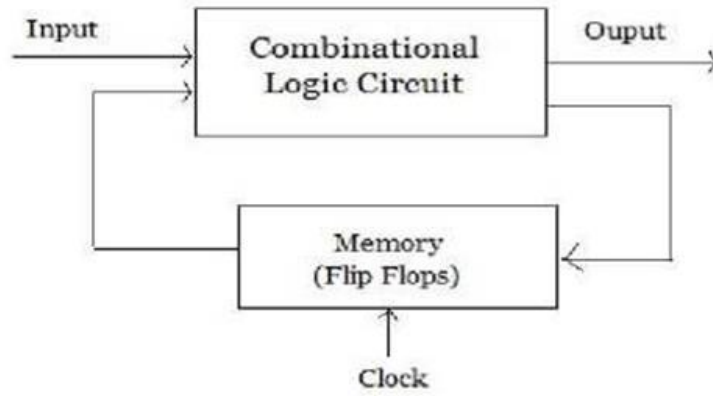


Figure 4.1: Sequential circuit

सीक्वेंशियल लॉजिक सर्किट्सची रचना प्रक्रिया

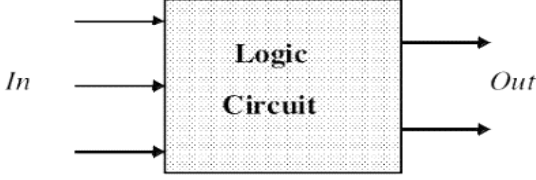
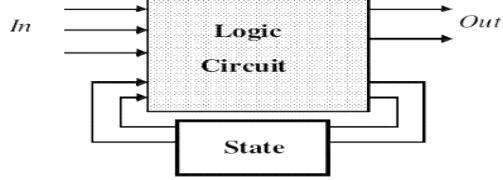
या प्रक्रियेत खालील चरणांचा समावेश असतो.

1. प्रथम, विविध स्टेट्स ची आकृती काढणे.
2. विविध स्टेट्स सारणी किंवा समतुल्य प्रतिनिधित्व (Equivalent representation) स्टेट् आकृती काढणे.
3. स्टेट्स कमी करण्याच्या तंत्राने स्टेट्सची संख्या कमी करणे.
4. आवश्यक फ्लिप-फ्लॉपची संख्या सत्यापित करणे.
5. तसेच वापरण्यासाठी फ्लिप-फ्लॉपचा प्रकार निवडणे.
6. उत्तेजनाची समीकरणे (excitation equations) काढणे.
7. K-Map नकाशा किंवा इतर काही सरलीकरण पद्धती (simplification method,) वापरून, आउटपुट फंक्शन आणि फ्लिप-फ्लॉप इनपुट फंक्शन्स मिळवणे.
8. लॉजिक डायग्राम किंवा बुलियन फंक्शन्सची सूची काढणे, ज्यामधून लॉजिक डायग्राम तयार करणे.

4.1 कॉम्बिनेशनल आणि सिक्वेन्शियल लॉजिक सर्किट्स यांच्यातील फरक

Table 4.1: Differences Between Combinational and Sequential Logic Circuits

कॉम्बिनेशनल सर्किट्स (Combinational)	सीक्वेंशियल सर्किट्स (Sequential)
ज्या सर्किटचे आउटपुट, कोणत्याही वेळी, फक्त त्या क्षणी उपस्थित असलेल्या इनपुटवर अवलंबून असते, त्याला कॉम्बिनेशनल सर्किट असे म्हणतात	ज्या सर्किटचे आउटपुट केवळ वर्तमान इनपुटवरच नाही तर मागील आउटपुटवर देखील अवलंबून असते, त्याला सीक्वेंशियल सर्किट असे म्हणतात
या प्रकारच्या सर्किट्समध्ये मेमरी युनिट नसते.	या प्रकारच्या सर्किट्समध्ये मागील आउटपुट संचयित करण्यासाठी मेमरी युनिट असते.
ते अधिक वेगवान आहे	ते कमी गतीचे आहे.
हे डिझाइन करणे सोपे आहे.	हे डिझाइन करणे कठीण आहे.
कॉम्बिनेशनल सर्किट्सची उदाहरणे हाफ अँडर, फुल अँडर, मॅग्निच्युड कंपॅरेटर, मल्टीप्लेक्सर,	सीक्वेंशियल सर्किट्सची उदाहरणे म्हणजे फ्लिप-फ्लॉप, रजिस्टर्स, काउंटर इ.

कॉम्बिनेशनल सर्किट्स (Combinational)	सीक्वेंशियल सर्किट्स (Sequential)
डिमल्टीप्लेक्सर इ.	
ब्लॉक आकृती Block diagram	ब्लॉक आकृती Block diagram
	

4.1.1 सीक्वेंशियल लॉजिक सर्किट्सच्या श्रेणी

सीक्वेंशियल लॉजिक सर्किट खालीलप्रमाणे तीन श्रेणींमध्ये विभागले गेले आहेत.

- **इव्हेंट-चालित (Event-driven):**

असिंक्रोनस सर्किट्स (unsecronus circuit) जे सक्षम केल्यावर लगेच स्थिती बदलू शकतात. असिंक्रोनस (मूलभूत मोड) सीक्वेंशियल सर्किट: वर्तन हे इनपुट सिग्नलच्या व्यवस्थेवर अवलंबून असते जे वेळेनुसार सतत बदलत असते आणि आउटपुट कधीही बदलता येते.

- **क्लॉक पल्स वापरून चालवलेले (Clock-driven/time dependent):** समकालिक सर्किट जे विशिष्ट क्लॉक सिग्नलवर सिंक्रोनाइझ केले जातात. सिंक्रोनस (लॅच मोड) सीक्वेंशियल सर्किट: क्लॉक सिग्नलचा वापर करून सिंक्रोनाइझेशन साध्य करणाऱ्या सर्किट्सच्या स्थितीवरून वर्तन परिभाषित केले जाऊ शकते.

- **पल्स ड्रिव्हन (Pulse driven):** ट्रिगर करणाऱ्या pulse ना प्रतिसाद देणारे हे दोघांचे मिश्रण आहे.
- **टाइम इंडिपेंडंट (un-clocked):** येथे क्लॉक चा वापर होत नाही

4.1.2 सीक्वेंशियल सर्किट्सचे प्रकार

सीक्वेंशियल सर्किट्सचे दोन प्रकारांमध्ये वर्गीकरण केले जाते

a. सिंक्रोनस सर्किट- सिंक्रोनस सीक्वेंशियल सर्किट्समध्ये, क्लॉक पल्सला प्रतिसाद म्हणून डिव्हाइसची स्थिती वेळोवेळी बदलते.

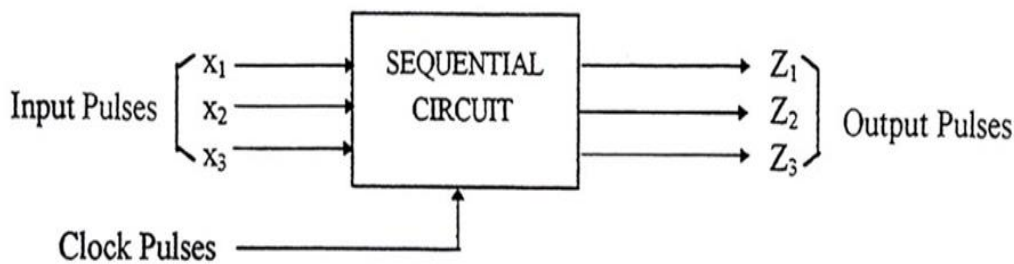


Figure 4.2: Synchronous circuit (powered using clock pulse)

सिंक्रोनस सर्किट्समध्ये, इनपुट हे पल्स रुंदी (width) आणि प्रसार विलंबावर (Delay) विशिष्ट निर्बंधांसह पल्स असतात. अशा प्रकारे सिंक्रोनस सर्किट्स हे क्लॉकड आणि अन-क्लॉकड सीक्वेंशियल सर्किट्समध्ये विभागले जाऊ शकतात.

सिंक्रोनस सर्किट: क्लॉक केलेले सीक्वेंशियल (Clocked) सर्किट्स

क्लॉक पल्स वापरलेल्या सीक्वेंशियल सर्किट्समध्ये त्याच्या मेमरी घटकांसाठी फ्लिप-फ्लॉप किंवा गेट लॅचेस वापरतात. अवस्थेतील सर्व अंतर्गत बदल समक्रमित करण्यासाठी सर्किटच्या सर्व मेमरी घटकांच्या क्लॉक इनपुटशी जोडलेले

नियतकालिक क्लॉक असते. म्हणून सर्किटचे ऑपरेशन क्लॉक च्या नियतकालिक पल्सद्वारे नियंत्रित आणि समक्रमित केले जाते.

b. असिंक्रोनस सर्किट - एसिंक्रोनस सर्किट्समध्ये, बदलत्या इनपुटच्या प्रतिसादात डिझाईसची स्थिती बदलते.

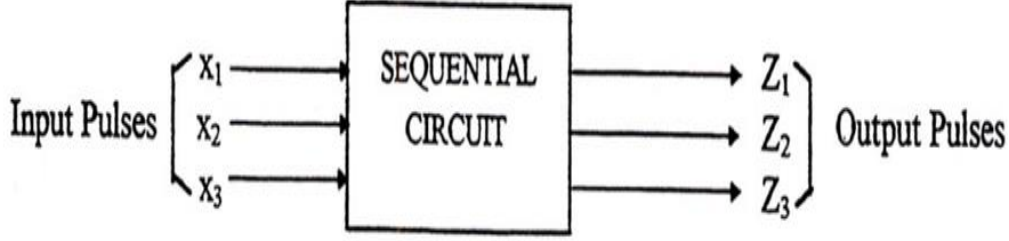


Figure 4.3: Asynchronous circuit(event-driven)

c. अन क्लॉक केलेले (un-clocked) सीक्वेंशियल सर्किट

अनक्लॉक केलेल्या सीक्वेंशियल सर्किटला सर्किटची स्थिती बदलण्यासाठी Zero '0' आणि '1' दरम्यान सलग दोन संक्रमणे आवश्यक असतात. अनक्लॉक केलेले मोड सर्किट विशिष्ट कालावधीच्या pulse ना प्रतिसाद देण्यासाठी डिझाइन केलेले आहे जे सर्किटच्या वर्तनावर परिणाम करत नाही.

अनक्लॉक केलेले सीक्वेंशियल सर्किट्स सिंक्रोनस लॉजिक सर्किट खूप सोपे आहे. लॉजिक गेट्स जे डेटावर ऑपरेशन्स करतात, त्यांना इनपुटमधील बदलांना प्रतिसाद देण्यासाठी मर्यादित कालावधी आवश्यक असतो.

4.1.3 असिंक्रोनस (Asynchronous) सर्किट्स एसिंक्रोनस सर्किटमध्ये त्याच्या विविध स्टेट्स अंतर्गत बदलांना समक्रमित करण्यासाठी क्लॉक सिग्नल नसतो. त्यामुळे प्राथमिक इनपुट लाइन्समध्ये होणाऱ्या बदलांच्या थेट प्रतिसादात विविध स्टेट्स बदल होतो. असिंक्रोनस सर्किटला फ्लिप-फ्लॉप्सकडून अचूक वेळेचे नियंत्रण आवश्यक नसते.

असिंक्रोनस लॉजिक सर्किट डिझाइन करणे अधिक कठीण आहे आणि सिंक्रोनस (**Synchronous**) लॉजिकच्या तुलनेत त्यात काही समस्या आहेत. मुख्य अडचण अशी आहे की डिजिटल मेमरी त्यांचे इनपुट सिग्नल त्यांच्यापर्यंत पोहोचण्याच्या क्रमाने संवेदनशील असते, जसे की, दोन सिग्नल एकाच वेळी फ्लिप-फ्लॉपवर आले तर, सर्किट कोणत्या स्थितीत जाते यावर अवलंबून असू शकते.

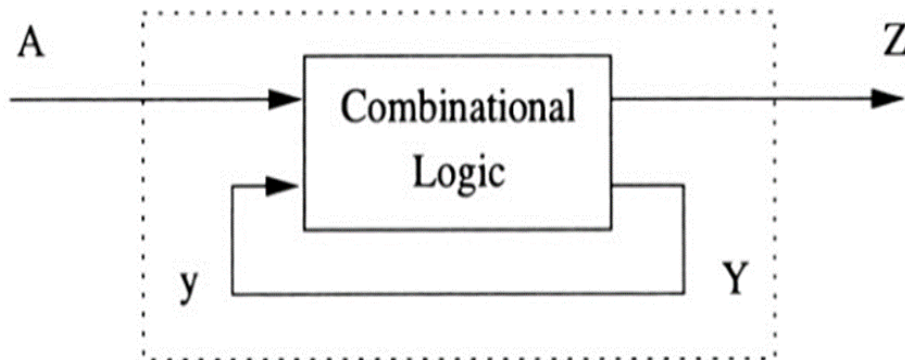


Figure 4.4: Asynchronous Circuit

सीक्वेंशियल सर्किट्सचे उपयोग :

मायक्रोप्रोसेसर आणि डिजिटल सिग्नल प्रोसेसिंग सर्किट्समध्ये सिंक्रोनस सर्किट्सचा वापर सिंक्रोनस सिस्टमच्या महत्वपूर्ण भागांमध्ये केला जातो जेथे सिस्टमची गती प्राधान्य असते.

लॅच:Latch

लॅच हे एक इलेक्ट्रॉनिक उपकरण आहे, जे लागू केलेल्या इनपुटच्या आधारे त्याचे आउटपुट त्वरित बदलते. हे कोणत्याही निर्दिष्ट वेळी 1 किंवा शून्य (0) संचयित करण्यासाठी वापरले जाते. यात "SET" आणि RESET असे दोन इनपुट आणि दोन आउटपुट असतात, जे एकमेकांना पूरक असतात.

Table 4.2: Difference between Latch and Flip Flop

फ्लिप-फ्लॉप / Flip-Flop	लॅच / Latch
फ्लिप-फ्लॉप हे बाय स्टेबल (bi-stable) उपकरण आहे, म्हणजे, त्याच्या दोन स्थिर अवस्था आहेत ज्या शून्य (0) आणि 1 म्हणून दर्शविल्या जातात.	लॅच हे एक बिस्टेबल (bi-stable) उपकरण आहे ज्याची अवस्था शून्य (0) आणि 1 म्हणून देखील दर्शविली जाते.
हे इनपुट तपासते परंतु क्लॉक पल्स सिग्नल किंवा clock pulse इतर कोणत्याही नियंत्रण सिग्नलद्वारे परिभाषित केलेल्या वेळीच आउटपुट बदलते.	हे इनपुट सतत तपासते आणि इनपुटमधील बदलांना त्वरित प्रतिसाद देते.
हे एज ट्रिगर केलेले उपकरण आहे.	लेव्हल ट्रिगर केलेले उपकरण आहे.
NOR, NOT, AND, NAND सारखे गेट्स फ्लिप फ्लॉपचे बिल्डिंग ब्लॉक्स आहेत.	हे देखील NOR, NAND गेट्स वापरून करता येतात.
ते एसिंक्रोनस किंवा सिंक्रोनस फ्लिपफ्लॉपमध्ये वर्गीकृत आहेत.	लॅचेमध्ये असे कोणतेही वर्गीकरण नाही.
हे काउंटर सारख्या अनेक सीक्वेंशियल सर्किट्सचे बिल्डिंग ब्लॉक्स बनवते.	हे सीक्वेंशियल सर्किट्सच्या डिझाइनसाठी वापरले जाऊ शकतात परंतु सामान्यतः प्राधान्य दिले जात नाही.
फ्लिप-फ्लॉपमध्ये नेहमी क्लॉक पल्स सिग्नल असतो लॅचमध्ये क्लॉक पल्स सिग्नल नसतो.	लॅचमध्ये क्लॉक पल्स सिग्नल नसतो.
फ्लिप-फ्लॉप लॅचमधून बनवता येतात.	लॅचेस गेट्समधून बनवता येतात.
उदा: D फ्लिप-फ्लॉप, जेके फ्लिप-फ्लॉप.	उदा: एसआर लॅच, डी लॅच.

4.1.4 बेसिक मेमरी सेल (ज्याला **वन बिट मेमरी** इलिमेंट असेही म्हणतात) हे एक डिजिटल सर्किट आहे जे एक बिट माहिती साठवू शकते. हा एक प्रकारचा सीक्वेंशियल सर्किट आहे जो नवीन इनपुट सिग्नल प्राप्त होईपर्यंत त्याची स्थिती धारण करू

शकतो, ज्यामुळे स्थिती बदलते. एक बिट मेमरी सेल डिजिटल सिस्टममध्ये तात्पुरते स्टोरेज घटक म्हणून वापरले जातात आणि फ्लिप-फ्लॉप, लॅचेस आणि स्टेट मशीन्स सारख्या अधिक जटिल सीकेंडरियल सर्किट्सचे बिल्डिंग ब्लॉक्स आहेत.

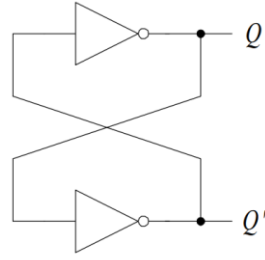


Figure 4.5: Basic Memory Cell (One Bit) Memory Cell

SR फ्लिपफ्लॉप तयार करण्याच्या पद्धती:

1. NOR Gate वापरून SR फ्लिप फ्लॉपचे लॉजिक सर्किट
2. NAND लॅच वापरून SR फ्लिप फ्लॉपचे लॉजिक सर्किट

1. NAND लॅच वापरून SR फ्लिप फ्लॉपचे लॉजिक सर्किट:

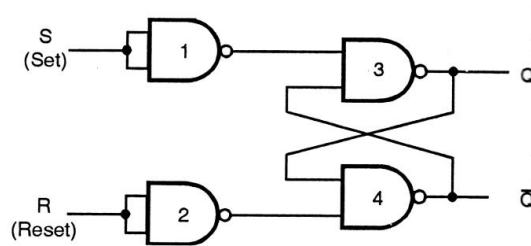


Figure 4.6: Logic circuit for SR flip flops manufactured using NAND latch

Table .4.3: SR Flip Flops Table using NAND Latch

Inputs		Outputs		States
S	R	Q	Q'	
0	0	1	1	Invalid
0	1	0	1	Q' is set to 1 (Reset)
1	0	1	0	Q is set to 1 (Set)
1	1	1	0	No Change

2) NOR Gate वापरून SR फ्लिप फ्लॉपचे लॉजिक सर्किट:

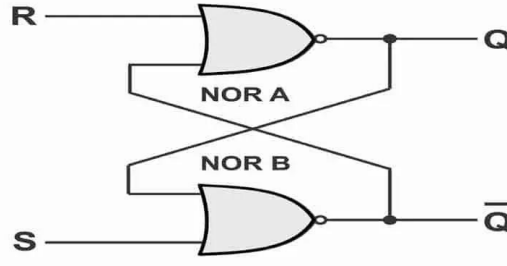


Figure 4.7: Logic circuit for SR flip flops manufactured using NOR latch

4.1.5 फ्लिप फ्लॉप ट्रिगरिंग पद्धती: उच्च पातळी ट्रिगरिंग, लो लेव्हल ट्रिगरिंग, पॉझिटिव्ह एज ट्रिगरिंग आणि निगेटिव्ह एज ट्रिगरिंग यासारख्या मूलभूत पल्स ट्रिगरिंग पद्धती clear करतो. Clock signal संक्रमणाचे मूलभूत तत्त्व देखील clear केले आहे. इनपुट सिग्नलमध्ये थोडासा बदल करून फ्लिप फ्लॉपचे आउटपुट बदलले जाऊ शकते. हा छोटासा बदल क्लॉक पल्सच्या मदतीने किंवा सामान्यतः ट्रिगर पल्स म्हणून ओळखला जाऊ शकतो.

जेव्हा अशी ट्रिगर पल्स इनपुटवर लागू केली जाते तेव्हा आउटपुट बदलतो आणि अशा प्रकारे फ्लिप फ्लॉप ट्रिगर झाला असे म्हणतात. फ्लिप फ्लॉप हे काउंटर किंवा रजिस्टर्सच्या डिझाईनिंगमध्ये लागू होतात जे मल्टी-बिट नंबर्सच्या स्वरूपात डेटा संग्रहित करतात. परंतु अशा रजिस्टर्सना सीक्वेंशियल सर्किट्स म्हणून एकमेकांशी जोडलेल्या फ्लिप फ्लॉप्सच्या गटाची आवश्यकता असते. आणि या सीक्वेंशियल सर्किट्सला ट्रिगर pulse ची आवश्यकता असते.

सर्किटला इनपुट ट्रिगर केलेल्या pulses काउंटरमधील संख्या निर्धारित करते. मल्टी-बिट डेटा संचयित करणाऱ्या रजिस्टरवर लागू केल्यावर एक clock pulse मुळे फक्त एका स्थानावर बदल करते.

SR फ्लिप फ्लॉप्सच्या बाबतीत, सिग्नल पातळीतील बदल इनपुटला कोणत्या प्रकारचे ट्रिगर द्यायचे हे ठरवते. परंतु सर्किटला दुसरा पल्स देण्यापूर्वी मूळ पातळी परत मिळवणे आवश्यक आहे.

फ्लिप फ्लॉपचे आउटपुट बदलत असताना त्याच वेळी फ्लिप फ्लॉपच्या इनपुटला clock pulse दिली असल्यास, यामुळे सर्किटमध्ये अस्थिरता येऊ शकते. या अस्थिरतेचे कारण आउटपुट कॉम्बिनेशनल सर्किटमधून मेमरी घटकांना दिलेला फीडबॅक आहे. पल्स कालावधीपेक्षा पल्स संक्रमणास फ्लिप फ्लॉप अधिक संवेदनशील बनवून ही समस्या एका विशिष्ट स्तरावर सोडविली जाऊ शकते.

States चा बदल क्लॉक पल्स नियंत्रित केले जातात. "क्लॉक पल्स (clock)" हे एक विशेष सर्किट आहे जे अचूक पल्स रुंदीसह आणि सलग pulse मधील अचूक अंतरासह pulse पाठवते. सलग pulse मधील अंतराला क्लॉक पल्स चक्र वेळ म्हणतात. क्लॉक पल्स वेग साधारणपणे हर्ट्झ, मेगाहर्ट्झ किंवा गिगाहर्ट्झमध्ये मोजला जातो.

क्लॉक पल्स संक्रमण (Pulse Transition)

ट्रिगर पल्सची हालचाल नेहमी शुन्य (0) ते 1 आणि नंतर 1 ते शुन्य (0) पर्यंत असते. अशा प्रकारे एकाच सिग्नलमध्ये दोन संक्रमणे होतात. जेव्हा ते 0 ते 1 वरून हलते तेव्हा त्याला पॉझिटिव्ह (Positive) संक्रमण म्हणतात आणि जेव्हा ते 1 ते शुन्य (0) वर जाते तेव्हा त्याला नकारात्मक (Negative) संक्रमण म्हणतात. अधिक समजून घेण्यासाठी खालील प्रतिमा पहा.

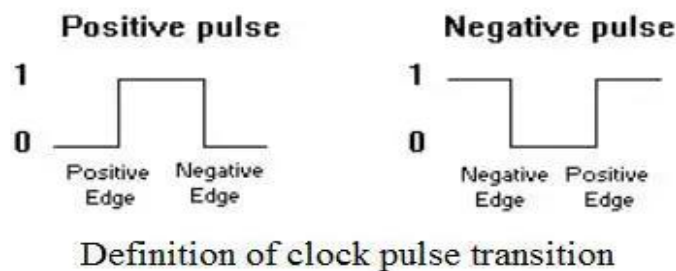


Figure 4.8: Clock Pulse Transition

आधीच सादर केलेले क्लॉक केलेले फ्लिप-फ्लॉप signal च्या शुन्य (0) ते 1 संक्रमणादरम्यान ट्रिगर केले जातात आणि क्लॉक पल्स उच्च स्तरावर पोहोचताच स्टेट संक्रमण (States transition) सुरू होते. क्लॉक सिग्नल 1 असताना इतर इनपुट बदलल्यास, नवीन आउटपुट स्थिती येऊ शकते. जर फ्लिप-फ्लॉप केले असेल तर बहु-संक्रमण समस्या दूर केली जाऊ शकते. बहु-संक्रमण समस्या थांबविली जाऊ शकते फ्लिप फ्लॉप संपूर्ण wave कालावधीलाप्रतिसाद देण्याव्यतिरिक्त केवळ पॉजिटिव किंवा नकारात्मक किनारी संक्रमणास (positive or negative edge triggered) प्रतिसाद देण्यासाठी केले जाते.

सीक्वेंशियल लॉजिक सर्किट्सची उदाहरणे:

सीक्वेंशियल लॉजिक सर्किट्सची उदाहरणे खाली चर्चा केली आहेत.

- डिजिटल घड्याळे (Digital Clocks)
- बहुतेक सीक्वेंशियल सर्किट्सच्या स्थितीतील बदल फ्री-रनिंग क्लॉक सिग्नलद्वारे निर्दिष्ट केलेल्या वेळी होतात. नावाप्रमाणेच, सीक्वेंशियल लॉजिक सर्किट्सला एक साधन आवश्यक आहे ज्याद्वारे घटनांचा क्रम लावला जाऊ शकतो. पल्स ट्रिगर करण्याच्या पद्धती प्रामुख्याने चार प्रकारच्या आहेत. इलेक्ट्रॉनिक सर्किट्स पल्स ट्रिगर ला ज्या पद्धतीने प्रतिसाद देतात त्यामध्ये ते भिन्न आहेत. ते आहेत

1. लेव्हल ट्रिगरिंग

a. उच्च स्तरीय ट्रिगरिंग

जेव्हा फ्लिप फ्लॉपला त्याच्या उच्च स्थितीत प्रतिसाद देणे आवश्यक असते, तेव्हा उच्च स्तर ट्रिगरिंग (high level triggering) पद्धत वापरली जाते. हे प्रामुख्याने खाली दर्शविलेल्या प्रतीकात्मक प्रतिनिधित्वावरून ओळखले जाते.

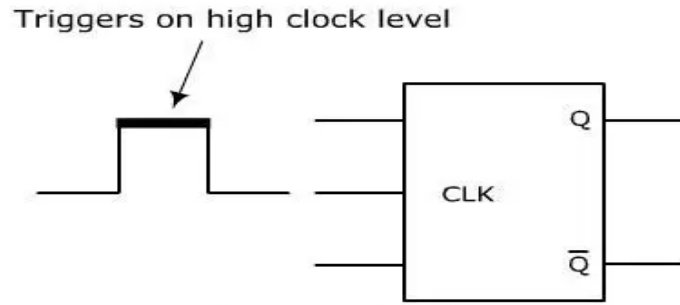


Figure 4.9 : High Level Triggering

b. निम्न पातळी ट्रिगरिंग (Low level triggered)

जेव्हा फ्लिप फ्लॉपला त्याच्या निम्न स्थितीत प्रतिसाद देणे आवश्यक असते, तेव्हा एक निम्न स्तर ट्रिगरिंग पद्धत वापरली जाते. हे मुख्यतः कमी स्थिती निर्देशक बबलसह क्लॉक पल्स वरून इनपुट (clock input) ओळखले जाते. खाली दर्शविलेल्या प्रतीकात्मक प्रतिनिधित्वावर एक नजर टाका.

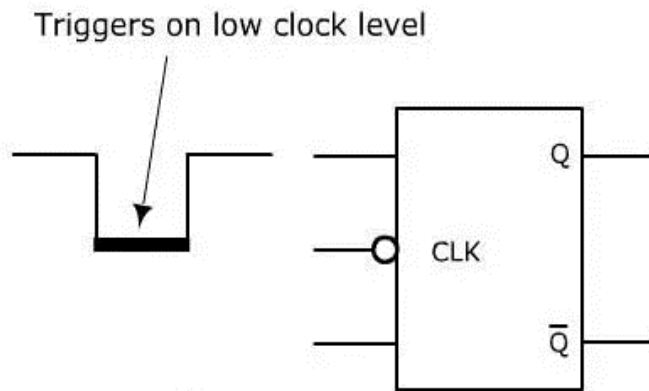


Figure 4.10: Low Level Triggering

3. एज (Edge)ट्रिगरिंग

a. पॉजिटिव एज (Positive Edge) ट्रिगरिंग

जेव्हा फ्लिप फ्लॉपला कमी ते उच्च संक्रमण स्थितीत प्रतिसाद देणे आवश्यक असते, तेव्हा पॉजिटिव एज ट्रिगरिंग पद्धत वापरली जाते. हे प्रामुख्याने त्रिकोणासह क्लॉक पल्स इनपुट लीडवरून ओळखले जाते. खाली दर्शविलेल्या प्रतीकात्मक प्रतिनिधित्वावर एक नजर टाका.

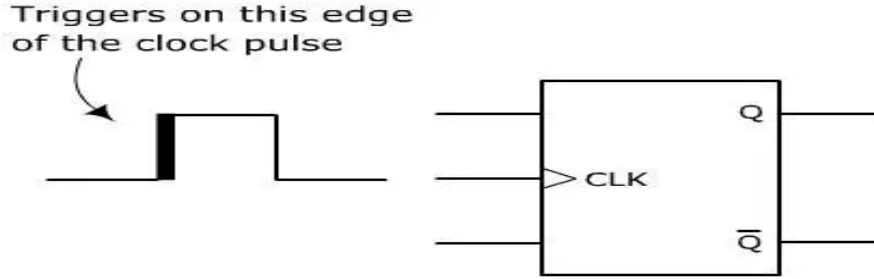


Figure 4.11: Positive Edge Triggering

b. नकारात्मक एज (Negative Edge) ट्रिगरिंग

जेव्हा उच्च ते निम्न संक्रमण अवस्थेत फ्लिप फ्लॉपला प्रतिसाद देणे आवश्यक असते, तेव्हा नकारात्मक एज ट्रिगरिंग पद्धत वापरली जाते. हे मुख्यतः कमी-स्थिती निर्देशक आणि त्रिकोणासह क्लॉक पल्स इनपुट सिग्नल वरून ओळखले जाते.

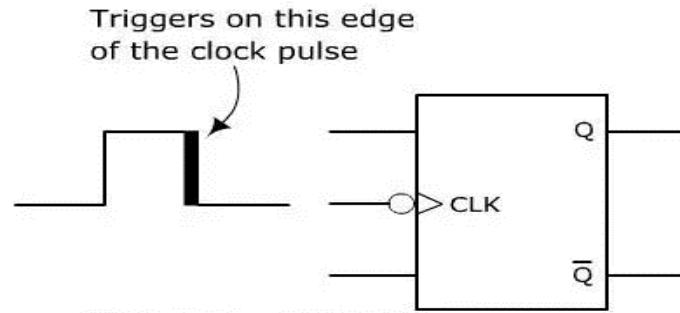


Figure 4.12: Negative Edge Triggering

4.2: फ्लिप फ्लॉपस

फ्लिप-फ्लॉप हे सीक्वेंशियल सर्किट आहे जे इनपुटचे नमुने घेते आणि विशिष्ट वेळी आउटपुट बदलते. यात दोन स्थिर अवस्था आहेत आणि राज्य माहिती संग्रहित करण्यासाठी वापरली जाऊ शकतात. सर्किटची स्थिती बदलण्यासाठी एक किंवा अधिक कंट्रोल इनपुटवर सिग्नल लागू केले जातात आणि एक किंवा दोन आउटपुट असतील.

हे सीक्वेंशियल (Analysis) लॉजिक आणि डिजिटल इलेक्ट्रॉनिक सिस्टमच्या मूलभूत बिल्डिंग ब्लॉक्समधील मूलभूत स्टोरेज घटक आहे. ते व्हेरिएबलच्या मूल्याची नोंद ठेवण्यासाठी वापरले जाऊ शकतात. सर्किटची कार्यक्षमता नियंत्रित करण्यासाठी फ्लिप-फ्लॉप देखील वापरला जातो.

4.2.1 फ्लिप फ्लॉप चे प्रकार

1. आर एस फ्लिप फ्लॉप (RS- reset-set flipflop)
2. जे के फ्लिप फ्लॉप (J-K flipflop)
3. डी फ्लिप फ्लॉप (Delay flipflop)
4. टी फ्लिप फ्लॉप (Toggle flipflop)

1. आर-एस फ्लिप फ्लॉप

R-S फ्लिप-फ्लॉप सर्वात सोपा फ्लिप-फ्लॉप आहे. यात दोन इनपुट आणि दोन आउटपुट आहेत, एक आउटपुट दुसऱ्याच्या उलट आहे आहेत. दोन इनपुट सेट आणि रीसेट आहेत.

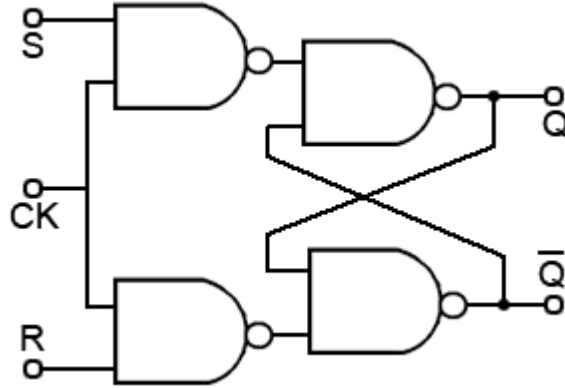


Fig 4.13 S-R Flip Flop

Table 4.4: Clocked SR flip flop Truth Table

S	R	CLK	(t+1)	Comments
0	0	X	Q(t)	No Change
0	1	↑	0	Reset
1	0	↑	1	Set
1	1	↑	?	Forbidden

2. JK फ्लिप फ्लॉप

जेके फ्लिप-फ्लॉप हे महत्वाचे फ्लिप-फ्लॉप आहे. J आणि K इनपुट एक असल्यास आणि जेव्हा क्लॉक पल्स लागू केले जाते, तेव्हा भूतकाळातील परिस्थिती लक्षात न घेता आउटपुट बदलतो. J आणि K इनपुट शुन्य (0) असल्यास आणि क्लॉक पल्स लागू केल्यावर, आउटपुटमध्ये कोणताही बदल होणार नाही. JK फ्लिप-फ्लॉपमध्ये कोणतीही अनिश्चित स्थिती नाही.

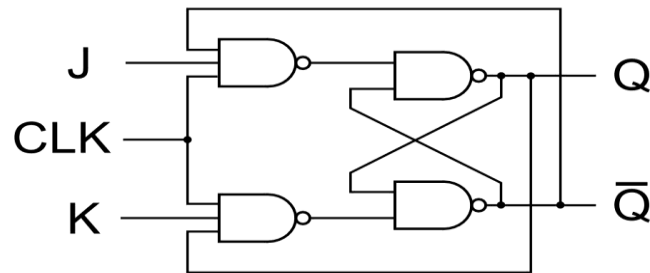


Fig 4.14: J-K Flip Flop circuit

Table 4.5 J-K Flip Flop Truth table

Clock	Inputs		Output	States
	J	K	Q_{n+1}	
0	X	X	Q_n	
1	0	0	Q_n	Hold
1	0	1	0	Reset
1	1	0	1	Set
1	1	1	Q'_n	Toggle

3. डी फ्लिप फ्लॉप (Delay)

D फ्लिप-फ्लॉपमध्ये एक डेटा लाइन आणि एक क्लॉक पल्स इनपुट आहे. डी फ्लिप-फ्लॉप हे SR फ्लिप-फ्लॉपचे सरलीकरण आहे. D फ्लिप-फ्लॉपचे इनपुट थेट इनपुट S वर जाते आणि कॉम्प्लिमेंटेड इनपुट R ला जाते.

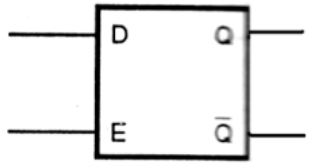


Fig 4.15: D Flip Flop symbol

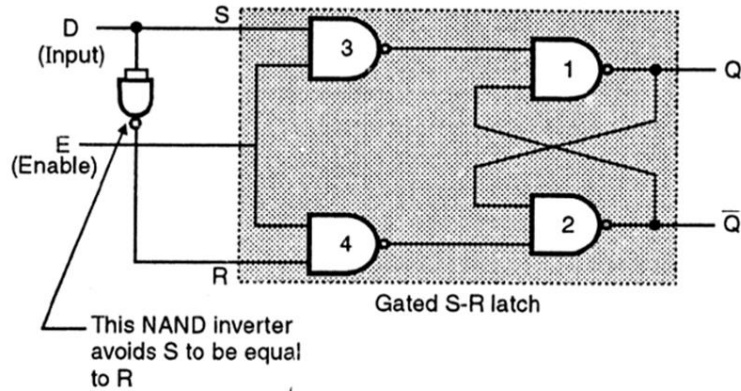


Fig 4.16: D Flip Flop using NAND Gate

Table 4.6: D (Delay) Flip Flop Truth Table

Clock	D	Q	$\bar{Q}$
0	0	Q	$\bar{Q}$
0	1	Q	$\bar{Q}$
1	0	0	1
1	1	1	0

4. टी (Toggle) फ्लिप फ्लॉप

ही RS फ्लिप-फ्लॉप प्रक्रियेत आढळणारी अनिश्चित स्थिती टाळण्याची एक पद्धत आहे. हे फक्त एक इनपुट प्रदान करण्यासाठी आहे, म्हणजे टी इनपुट. हा फ्लिप-फ्लॉप टॉगल स्विच म्हणून काम करतो. टॉगल म्हणजे State बदलणे. टी फ्लिप-फ्लॉप क्लॉक केलेल्या आरएस फ्लिप-फ्लॉपपासून डिझाइन केले आहे.

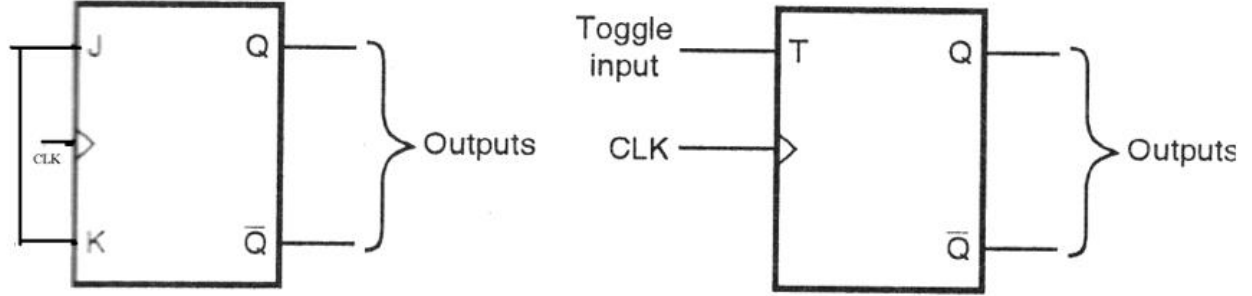


Fig 4.17: T(Toggle) Flip Flop

Table 4.7: T(Toggle) Flip Flop Truth Table

Clock	T	Q	$\bar{Q}$
0	0	0	0
0	1	1	1
0	0	0	1
0	1	1	0

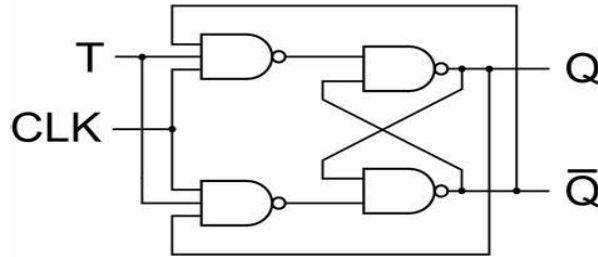


Fig 4.18: T Flip Flop circuit

4.2.2 उत्तेजना सारणी (Excitation table) मध्ये किमान इनपुट आहेत, जे फ्लिप फ्लॉपला त्याच्या सध्याच्या स्थितीतून पुढील स्थितीत जाण्यासाठी उत्तेजित करेल किंवा ट्रिगर करेल. हे तृथ टेबल तून घेतले आहे.

सामान्यतः, प्रत्येक फ्लिप-फ्लॉपचे ऑपरेशन तृथ टेबल च्या मदतीने clear केले जाते. तृथ टेबल मध्ये सर्व इनपुट कॉम्बिनेशन आहेत, ज्यासाठी फ्लिप फ्लॉप पुढील स्टेट आउटपुट तयार करण्यासाठी प्रतिक्रिया देतो. उत्तेजना सारणीमध्ये सध्याच्या स्थितीसाठी (Q_n) आणि पुढील स्थितीसाठी (Q_{n+1}) असे दोन स्तंभ आणि प्रत्येक इनपुटसाठी एक किंवा दोन स्तंभ आउटपुट असतात. इनपुट कॉलम फ्लिप-फ्लॉपच्या प्रकारावर अवलंबून असतात. आता, प्रत्येक फ्लिप-फ्लॉपसाठी उत्तेजना सारणी पाहू.

फ्लिप फ्लॉपसाठी उत्तेजना सारणी मिळवणे:

1. States साठी Gray कोड असाइनमेंट वापरतात.
2. एस-आर, जे-के, डी आणि टी फ्लिप फ्लॉपसाठी उत्तेजना सारणी excitation table मिळवा

Table 4.8: S R Flip flop Excitation table

SR फ्लिप फ्लॉप उत्तेजना सारणी			
Present State	Next State	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Table 4.10: J K Flip Flop Excitation table

J K फ्लिप फ्लॉप उत्तेजना सारणी			
Present State	Next State	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Table 4.11: D Flip Flop Excitation table

Present State	Next State	D
0	0	0
0	1	1
1	0	0
1	1	1

Table 4.12: T Flip Flop Excitation table

Present State	Next State	T
0	0	0
0	1	1
1	0	1
1	1	0

4.2.3 फ्लिप फ्लॉप सीक्वेंशियल सर्किट अनुप्रयोगामध्ये वापरले जातात- फ्लिप-फ्लॉपचे इनपुट निर्धारित करण्यासाठी जेव्हा सध्याची स्थिती आणि clock input च्या स्थिती नंतर फ्लिप-फ्लॉप ज्या स्थितीत जातो .

फ्लिप फ्लॉप चे प्रमुख अनुप्रयोग/ ऍप्लिकेशन्स आहेत,

1. काउंटर म्हणून,
2. शिफ्ट रजिस्टर,
3. फ्लिप-फ्लॉप.
4. मेमरी युनिट तयार करण्यासाठी वापरले जाते.
5. प्रोग्राम करण्यायोग्य उपकरणे म्हणून (PLDs, FPGA, CPLDs)

4.3 Race round condition

रेस अराउंड कंडिशन :

जेके फ्लिप फ्लॉपमध्ये रेस अराउंड कंडिशन उद्भवते जेव्हा त्याचे इनपुट $J=K=1$ आणि क्लॉक पल्स =1 दीर्घ कालावधीसाठी. हे अनियंत्रित टॉगलिंग मास्टर-स्लेव्ह (Master-Slave) व्यवस्थेचा वापर करून दुरुस्ती केले जाऊ शकते. जेथे आउटपुटमध्ये J मूल्याचे प्रसारण अर्ध्या क्लॉक पल्सच्या चक्राने विलंबित होते आणि लगेच इनपुट बाजूला परत दिले जात नाही.

4.3.1 जेके फ्लिप-फ्लॉपमध्ये रेस-अराउंड कंडिशन

JK फ्लिप-फ्लॉप्सच्या वरील तृथ टेबल वरून, हे लक्षात येते की जेव्हा $J = 1$ आणि $K = 1$, नंतर आउटपुट $Q_{n+1} = Q_n$, म्हणजे इनपुट $J = 1$ आणि $K = 1$ साठी, JK फ्लिप -फ्लॉप टॉगल स्विच म्हणून कार्य करते.

आपण इनपुट्स $J = 1$ आणि $K = 1$, आणि आउटपुट $Q =$ शुन्य (0) विचारात घेऊ या. फ्लिप-फ्लॉपच्या प्रसार विलंबानंतर, JK फ्लिप-फ्लॉपचे आउटपुट zero (0) ते 1 पर्यंत बदलते. जसे आपण जाणून घ्या, JK फ्लिप-फ्लॉपचे आउटपुट त्याच्या इनपुटशी जोडलेले आहे. म्हणून, आउटपुट देखील इनपुट म्हणून कार्य करते आणि अशा प्रकारे पुढील विलंबानंतर, आउटपुट 1 ते zero (0) पर्यंत बदलेल. लागू केलेल्या क्लॉक पल्स सिग्नलच्या समाप्तीपर्यंत ही प्रक्रिया सुरू राहील. अशा प्रकारे, JK फ्लिप-फ्लॉपचे आउटपुट अनिश्चित आहे. JK फ्लिप-फ्लॉपच्या या स्थितीला रेस-अराउंड स्थिती म्हणतात.

रेस-अराउंड स्थिती कशी टाळता येईल?

रेस-अराउंड स्थितीची समस्या फ्लिप-फ्लॉपमध्ये अस्तित्वात नाही जेथे clock pulseच्या उपस्थितीत इनपुट बदलत नाहीत. परंतु, JK फ्लिप-फ्लॉपच्या बाबतीत, इनपुट आणि आउटपुट दरम्यान उपस्थित फीडबॅक मार्गामुळे clock pulseच्या दरम्यान इनपुट बदलतात. म्हणून, JK फ्लिप-फ्लॉपमध्ये, स्थितीभोवतीची शर्यत ही एक मोठी समस्या आहे.

रेस-अराउंड स्थितीची समस्या आणि आउटपुटची अनिश्चितता फ्लिप-फ्लॉपचा विलंब वाढवून टाळता येऊ शकते. त्यासाठी, फ्लिप-फ्लॉपचा विलंब clock signal च्या कालावधीपेक्षा जास्त असणे आवश्यक आहे. . दुसऱ्या मार्गाने, लागू केलेल्या क्लॉक पल्स सिग्नलचा कालावधी (Time) विलंबापेक्षा कमी करण्यासाठी कमी करणे आवश्यक आहे. फ्लिप-फ्लॉप चे तथापि, फ्लिप-फ्लॉप्सच्या विलंबात वाढ ही चांगली सराव नाही कारण यामुळे सिस्टमची गती कमी होते. दुसरीकडे, clock pulse चा कालावधी (T-फ्लिप-फ्लॉप्स) च्या विलंबापेक्षा कमी करणे देखील खूप कठीण आहे. कारण, JK फ्लिप-फ्लॉप चा विलंब नॅनोसेकंदांच्या क्रमाचा आहे.

म्हणून, JK फ्लिप-फ्लॉपमधील रेस-अराउंड स्थितीची समस्या सोडवण्याचा सर्वात व्यावहारिक मार्ग म्हणजे मास्टर आणि स्लेव्ह मोडमध्ये JK फ्लिप-फ्लॉप वापरणे. JK फ्लिप-फ्लॉपच्या मास्टर-स्लेव्ह मोडमध्ये, दोन JK फ्लिप-फ्लॉप, एक मागे एक (cascade) केले जातात.

हे सर्व जेके फ्लिप-फ्लॉपमधील रेस-अराउंड स्थिती आणि त्याच्या उपायांबद्दल आहे.

4.3.2 मास्टर-स्लेव्ह जेके फ्लिप फ्लॉप मध्ये रेस-अराउंड कंडिशन

JK फ्लिप-फ्लॉपमध्ये - J-K फ्लिप-फ्लॉपसाठी, जर $J=K=1$, आणि जर clock =1 दीर्घ कालावधीसाठी, तर Q आउटपुट जोपर्यंत CLK जास्त असेल तोपर्यंत टॉगल होईल, ज्यामुळे फ्लिपचे आउटपुट होते. फ्लॉप अस्थिर किंवा अनिश्चित. J-K फ्लिप-फ्लॉपमध्ये या समस्येला रेस अराउंड कंडिशन म्हणतात. ही समस्या (रेस अराउंड कंडिशन) हे सुनिश्चित करून टाळता येऊ शकते की क्लॉक पल्स इनपुट वापरून लॉजिक "1" वर अगदी थोड्या काळासाठी वापरून मास्टर स्लेव्ह जेके फ्लिप फ्लॉपची संकल्पना सुरू झाली. मास्टर स्लेव्ह जेके फ्लिप फ्लॉप - मास्टर-स्लेव्ह फ्लिप-फ्लॉप हे मुळात serial configuration मालिका कॉन्फिगरेशनमध्ये एकत्र जोडलेल्या दोन जेके फ्लिप-फ्लॉपचे संयोजन आहे. यापैकी, एक "मालक(master)" आणि दुसरा "गुलाम(Slave)" म्हणून कार्य करतो. मास्टर फ्लिप फ्लॉपचे आउटपुट स्लेव्ह फ्लिप फ्लॉपच्या दोन इनपुटशी जोडलेले असते ज्यांचे आउटपुट मास्टर फ्लिप फ्लॉपच्या इनपुटला परत दिले जाते. या दोन फ्लिप-फ्लॉप व्यतिरिक्त, सर्किटमध्ये इन्व्हर्टर देखील समाविष्ट आहे. इन्व्हर्टर clock pulse शी अशा प्रकारे जोडलेले आहे की

clock pulse स्लेव्ह फ्लिप-फ्लॉपला दिली जाते. दुसऱ्या शब्दांत जर मास्टर फ्लिप-फ्लॉपसाठी CP(क्लॉक पल्स) = शुन्य (0) असेल, तर स्लेव्ह फ्लिप-फ्लॉपसाठी CP(क्लॉक पल्स) = 1 आणि मास्टर फ्लिप फ्लॉपसाठी CP(क्लॉक पल्स) = 1 असेल तर स्लेव्ह फ्लिपसाठी 0 होईल.

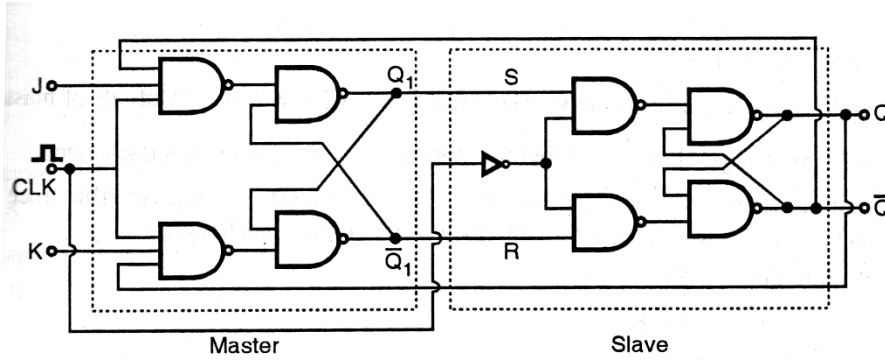


Fig 4.19: Master Slave Flip Flop Race around condition

मास्टर स्लेव्ह फ्लिप फ्लॉप चे कार्य करणे.

- जेव्हा clock pulse 1 वर जाते, तेव्हा गुलाम वेगळा होतो; J आणि K इनपुट सिस्टमच्या स्थितीवर परिणाम करू शकतात. CP zero (0) वर जाईपर्यंत स्लेव्ह फ्लिप-फ्लॉप वेगळे केले जाते. जेव्हा CP परत **शुन्य (0)** वर जाते, तेव्हा मास्टर फ्लिप-फ्लॉपकडून स्लेव्हकडे माहिती दिली जाते आणि आउटपुट प्राप्त होते.
- प्रथमतः मास्टर फ्लिप फ्लॉप पॉजिटिव स्तर ट्रिगर (Positive edge trigger) आहे आणि स्लेव्ह फ्लिप फ्लॉप नकारात्मक स्तर ट्रिगर (Negative edge trigger) आहे, म्हणून मास्टर स्लेव्हच्या आधी प्रतिसाद देतो.
- J=0(zero) आणि K=1 असल्यास, मास्टरचे उच्च Q' आउटपुट स्लेव्हच्या K इनपुटवर जाते आणि क्लॉक पल्स स्लेव्हला रीसेट करण्यास भाग पाडते, अशा प्रकारे स्लेव्ह मास्टरची कॉपी करतो.
- J=1 आणि K= **0(zero)** असल्यास, मास्टरचे उच्च Q आउटपुट स्लेव्हच्या J इनपुटवर जाते आणि घड्याळाचे नकारात्मक संक्रमण स्लेव्ह सेट करते, मास्टरची कॉपी करते.
- J=1 आणि K=1 असल्यास, ते clock pulse च्या पॉजिटिव संक्रमणावर टॉगल करते आणि अशा प्रकारे स्लेव्ह negative edge triggered clock pulse नकारात्मक संक्रमणावर टॉगल करते.
- J= **शुन्य (0)** आणि K= **शुन्य (0)** असल्यास, फ्लिप फ्लॉप अक्षम केला जातो आणि Q अपरिवर्तित राहतो.

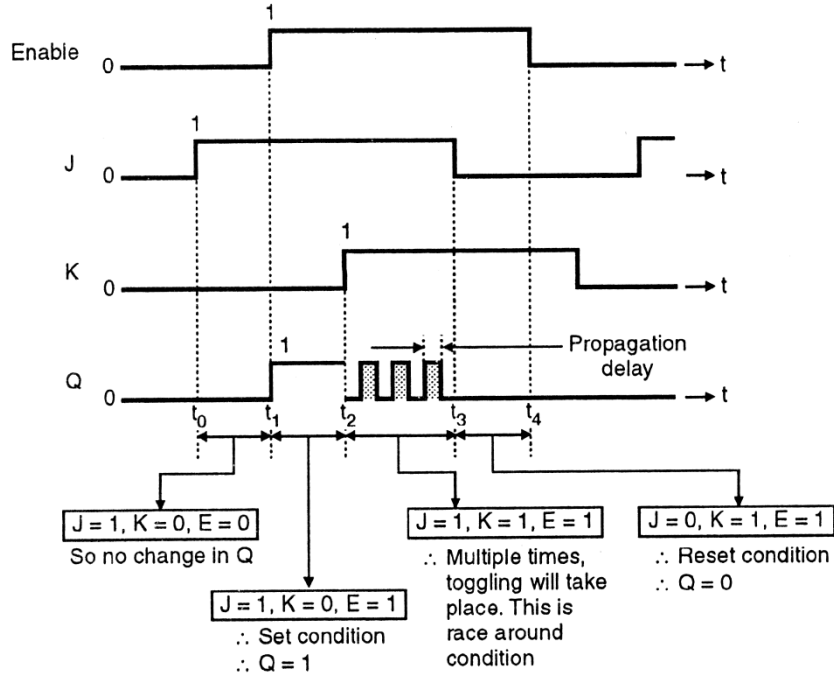


Fig 4.20: Master Slave JK Flip Flop Timing Diagram

1. क्लॉक पल्स जेव्हा high असते तेव्हा मास्टरचे आउटपुट high असते आणि clock pulse low होईपर्यंत ते high राहते कारण स्थिती साठवली जाते.
2. आता जेव्हा clock pulse पुन्हा high होते तेव्हा मास्टरचे आउटपुट low होते आणि clock pulse पुन्हा high होईपर्यंत कमी राहते.
3. अशा प्रकारे clock cycle साठी टॉगलिंग होते.
4. जेव्हा clock pulse जास्त असते, तेव्हा मास्टर (master) कार्यरत असतो परंतु गुलाम (slave) नाही अशा प्रकारे clock pulse क्लॉक पल्स high राहेपर्यंत slave चे उत्पादन low राहते.
5. जेव्हा clock pulse कमी होते, तेव्हा **गुलाम** (slave) कार्यरत होते आणि clock pulse पुन्हा कमी होईपर्यंत उच्च राहते.
6. संपूर्ण प्रक्रियेदरम्यान टॉगलिंग होते कारण आऊटपुट सायकलमध्ये एकदा बदलत असतो.

4.4 शिफ्ट रजिस्टरस (Shift Registers)

शिफ्ट रजिस्टरस एक प्रकारचे circuit आहे जो डेटा शिफ्ट करण्यास सक्षम आहे.

फ्लिप फ्लॉपचा एक गट जो डेटाचे एकाधिक बिट (multiple bits) संचयित करण्यासाठी वापरला जातो आणि डेटा एका फ्लिप फ्लॉपमधून दुसऱ्यामध्ये हलविला जातो त्याला शिफ्ट रजिस्टर म्हणून ओळखले जाते. जेव्हा clock pulse रजिस्टरच्या आत आणि आत किंवा बाहेर पुरवठा करण्यासाठी लावली जाते तेव्हा रजिस्टरमध्ये साठवलेले बिट्स हलवले जातात. एन-बिट शिफ्ट रजिस्टर तयार करण्यासाठी, आपल्याला फ्लिप फ्लॉपची संख्या n (N) जोडावी लागेल. तर, बायनरी क्रमांकाच्या बिट्सची संख्या फ्लिप फ्लॉपच्या संख्येशी थेट प्रमाणात असते. फ्लिप फ्लॉप अशा प्रकारे जोडलेले आहेत की पहिल्या फ्लिप फ्लॉपचे आउटपुट दुसऱ्या फ्लिप फ्लॉपचे इनपुट बनते.

शिफ्ट रजिस्टर खालील प्रकारांमध्ये वर्गीकृत केले आहे:

1. सिरीयल इन सिरीयल आऊट शिफ्ट रजिस्टर (Serial In Serial Out Shift Register)
2. सिरीयल इन पॅरालल आऊट शिफ्ट रजिस्टर (Serial In Parallel Out Shift Register)

3. पॅरालल इन सिरीयल आऊट शिफ्ट रजिस्टर (Parallel In Serial Out Shift Register)
4. पॅरालल इन पॅरालल आऊट शिफ्ट रजिस्टर (Parallel In Parallel Out Shift Register)
5. द्वि-दिशात्मक शिफ्ट रजिस्टर (Bidirectional Shift Register)
6. युनिव्हर्सल शिफ्ट रजिस्टर (Universal Shift Register)

1. सिरीयल इन सिरीयल आऊट शिफ्ट रजिस्टर (Serial in Serial out Shift Register):

"सिरीयल इनपुट सिरीयल आउटपुट" मध्ये, डेटा अनुक्रमे "IN" किंवा "आउट" हलविला जातो. "सिरीयल इनपुट सिरीयल आउटपुट" (SISO) मध्ये, क्लॉक पल्सच्या नियंत्रणाखाली एका वेळी एक बिट उजवीकडे किंवा डावीकडे शिफ्ट केला जातो. सुरुवातीला, सर्व फ्लिप-फ्लॉप "रीसेट" स्थितीत सेट केले जातात म्हणजे $Y_3=Y_2=Y_1=Y_0=0$ (शून्य (0)). जर आपण बायनरी क्रमांक 1111 पास केला, तर नंबरचा LSB बिट प्रथम D_{in} बिटवर लागू केला जातो. तिसऱ्या फ्लिप फ्लॉपचे D_3 इनपुट, म्हणजे, FF-3, सीरीयल डेटा इनपुट D_3 शी थेट जोडलेले आहे. आउटपुट Y_3 पुढील फ्लिप फ्लॉपच्या डेटा इनपुट D_2 ला पास केले जाते. ही प्रक्रिया उर्वरित फ्लिप फ्लॉपसाठी समान राहते. "Serial IN Serial OUT" चा ब्लॉक डायग्राम खाली दिला आहे.

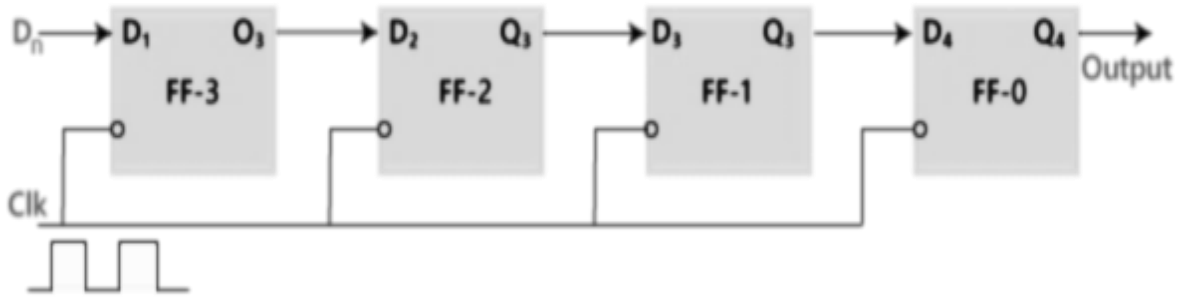


Figure 4.21: Serial in Serial Out Shift Register

कार्य पद्धत / संकलन

जेव्हा clock signal ऍप्लिकेशन अक्षम (disabled) केले जाते, तेव्हा $Y_3 Y_2 Y_1 Y_0 = 0000$ आउटपुट होते. नंबरचा LSB बिट डेटा इनपुट D_{in} ला पास केला जातो, म्हणजे, D_3 . आम्ही clock signal लागू करू, आणि यावेळी D_3 चे मूल्य 1 आहे. पहिला फ्लिप फ्लॉप, म्हणजे, FF-3, सेट केला आहे, आणि शब्द घड्याळाच्या पहिल्या घसरणीच्या काठावर रजिस्टरमध्ये संग्रहित केला जातो. आता, संग्रहित शब्द **1000** आहे.

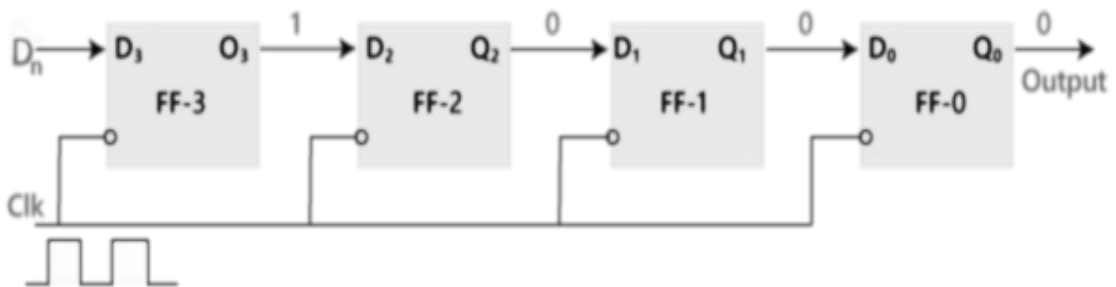


Figure 4.22: Serial in Serial Out Shift Register Phase 1

बायनरी क्रमांकाचा पुढील बिट, म्हणजे, 1, डेटा इनपुट D_2 ला पास केला जातो. दुसरा फ्लिप फ्लॉप, म्हणजे, FF-2, सेट केला जातो आणि जेव्हा clock pulse ची पुढील नकारात्मक एज (negative edge trigger) दिली जाते तेव्हा data संग्रहित केला जातो. संग्रहित data 1100 मध्ये बदलला आहे.

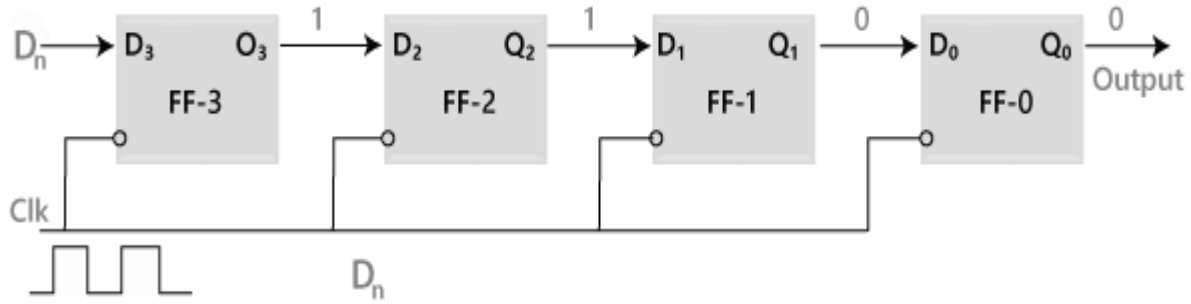


Figure 4.23: Serial in Serial Out Shift Register Phase II

बायनरी नंबरचा पुढील बिट, म्हणजे 1, डेटा इनपुट D1 ला पास केला जातो आणि clock pulse लागू केले जाते. तिसरा फ्लिप फ्लॉप, म्हणजे, FF-1, सेट केला जातो आणि जेव्हा (negative edge trigger) दिली जाते तेव्हा शब्द संग्रहित केला जातो. संग्रहित शब्द 1110 मध्ये बदलला आहे.

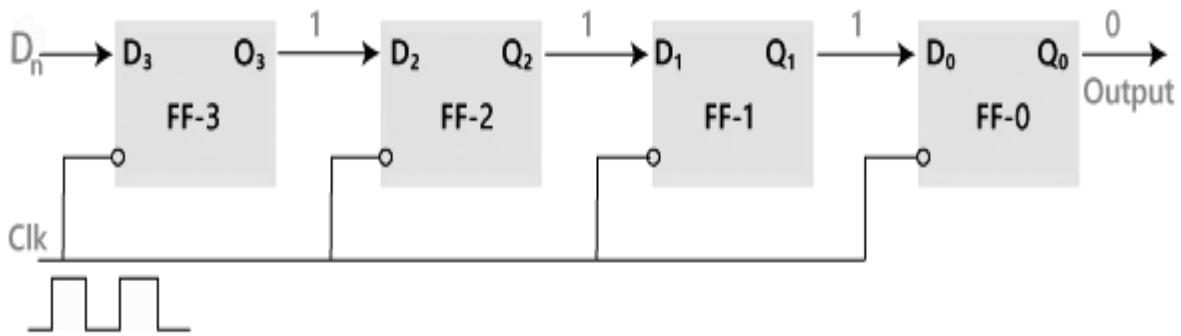


Figure 4.24: Serial in serial out shift register third phase

त्याचप्रमाणे, बायनरी क्रमांकाचा शेवटचा बिट, म्हणजे, 1, डेटा इनपुट D0 ला पास केला जातो आणि clock pulse लागू केले जाते. शेवटचा फ्लिप फ्लॉप, म्हणजे, FF-0, सेट केला जातो आणि जेव्हा (negative edge trigger) दिली जाते तेव्हा शब्द संग्रहित केला जातो. संग्रहित शब्द 1111 मध्ये बदलला आहे.

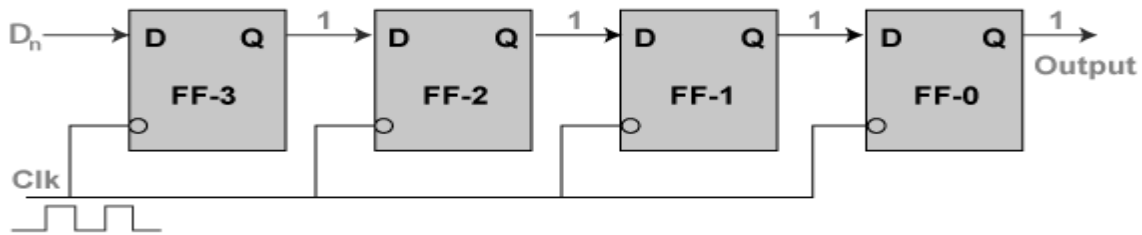


Figure 4.25: Serial in serial out shift register final phase

Table 4.13: Data entry at different stages

	Clk	$D_n = Q_3$	$Q_3 = D_2$	$Q_2 = D_1$	$Q_1 = D_0$	Q_0
Initially			0	0	0	0
(1)	↓	1 →	1	0	0	0
(2)	↓	1 →	1	1	0	0
(3)	↓	1 →	1	1	1	0
(4)	↓	1 →	1	1	1	1

→ Direction of data travel

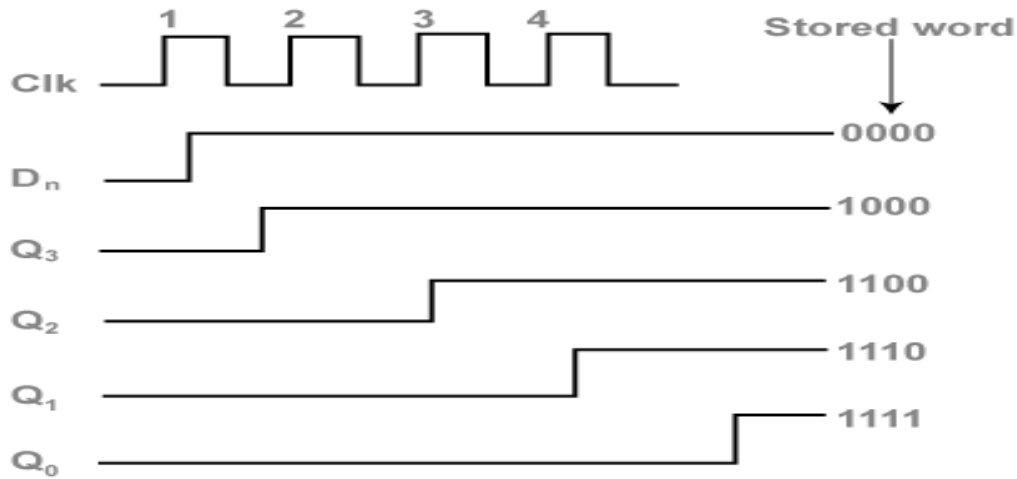


Fig 4.26: Waveforms at different stages of the shift register

2. सिरीयल इन पॅरालल आऊट शिफ्ट रजिस्टर (Serial in Parallel out Shift Register)

"Serial IN Parallel OUT" शिफ्ट रजिस्टरमध्ये, डेटा फ्लिप फ्लॉपवर क्रमाने पास केला जातो आणि आउटपुट समांतर पद्धतीने आणले जातात. रजिस्टरमध्ये डेटा थोडा-थोडा पास केला जातो आणि जोपर्यंत डेटा डेटा इनपुटवर जात नाही तोपर्यंत आउटपुट अक्षम राहते. जेव्हा डेटा रजिस्टरला पास केला जातो तेव्हा आउटपुट सक्षम केले जातात आणि फ्लिप फ्लॉपमध्ये त्यांचे रिटर्न व्हॅल्यू असते.

खाली समांतर-आउट शिफ्ट रजिस्टरमधील 4-बिट सिरीयलचा ब्लॉक आकृती आहे. चार डी फ्लिप-फ्लॉप असलेल्या सर्किटमध्ये हे चार फ्लिप फ्लॉप रीसेट करण्यासाठी clear आणि क्लॉक पल्स सिग्नल असतो. SIPO मध्ये, दुसऱ्या फ्लिप फ्लॉपचे इनपुट हे पहिल्या फ्लिप फ्लॉपचे आउटपुट असते आणि असेच. प्रत्येक फ्लिप फ्लॉपवर समान क्लॉक पल्स सिग्नल लागू केला जातो कारण फ्लिप फ्लॉप एकमेकांना सिंक्रोनाइझ करतात. (Refer FF-1 as Least Significant Bit (LSB) and FF-4 as Most Significant Bit (MSB)).

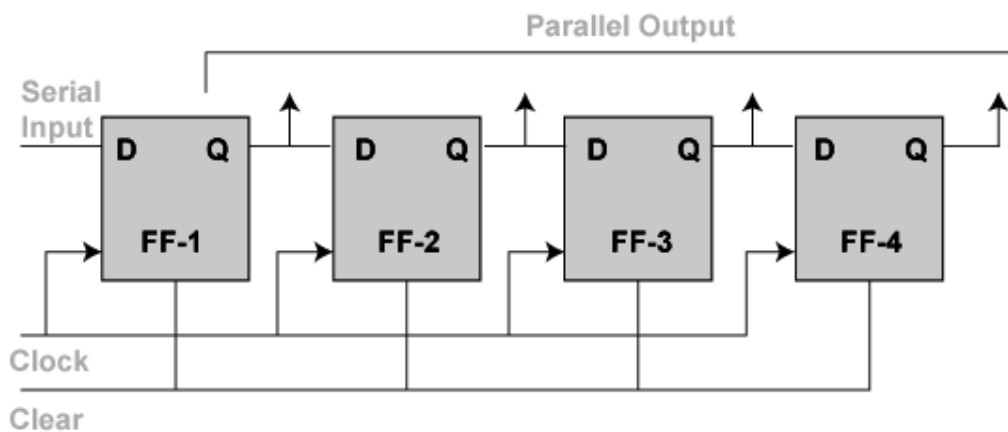


Fig 4.27: Serial in Parallel Out Shift Register

3. पॅरालल इन सिरीयल आऊट शिफ्ट रजिस्टर (Parallel In Serial Out Shift Register)

पॅरालल इन सिरीयल आऊट शिफ्ट रजिस्टर (Parallel IN Serial OUT) शिफ्ट रजिस्टर मध्ये, डेटा समांतर पद्धतीने एंटर केला जातो आणि परिणाम क्रमाने येतो. चार-बिट "समांतर इन सिरीयल आउट" रजिस्टर खाली डिझाइन केले आहे. फ्लिप फ्लॉपचे इनपुट हे मागील फ्लिप फ्लॉपचे आउटपुट आहे. इनपुट आणि आउटपुट कॉम्बिनेशनल सर्किटद्वारे जोडलेले आहेत. या कॉम्बिनेशनल सर्किटद्वारे बायनरी इनपुट B0, B1, B2, B3 पास केले जातात. शिफ्ट मोड आणि लोड मोड हे दोन मोड आहेत.

ज्यामध्ये "PISO" सर्किट कार्य करते.

लोड मोड

दुसरा, चौथा आणि सहावा "AND" गेट सक्रिय असताना B₀, B₁, B₂ आणि B₃ बिट्स संबंधित फ्लिप फ्लॉपवर पास केले जातात. जेव्हा शिफ्ट किंवा लोड बार लाईन 0 वर सेट केली जाते तेव्हा हे गेट्स सक्रिय असतात. घड्याळाची एज कमी असताना बायनरी इनपुट B₀, B₁, B₂ आणि B₃ संबंधित फ्लिप-फ्लॉपमध्ये लोड केले जातात. अशा प्रकारे, समांतर लोडिंग होते.

शिफ्ट मोड

लोड आणि शिफ्ट लाईन 0 वर सेट केल्यावर दुसरे, चौथे आणि सहावे गेट्स निष्क्रिय असतात. त्यामुळे, आम्ही समांतर पद्धतीने डेटा लोड करू शकत नाही. यावेळी, पहिले, तिसरे आणि पाचवे गेट्स सक्रिय केले जातात आणि डेटाचे स्थलांतर उजवीकडे सोडले जाईल. अशा प्रकारे, "पॅरालल इन सिरीयल आऊट" ऑपरेशन होते

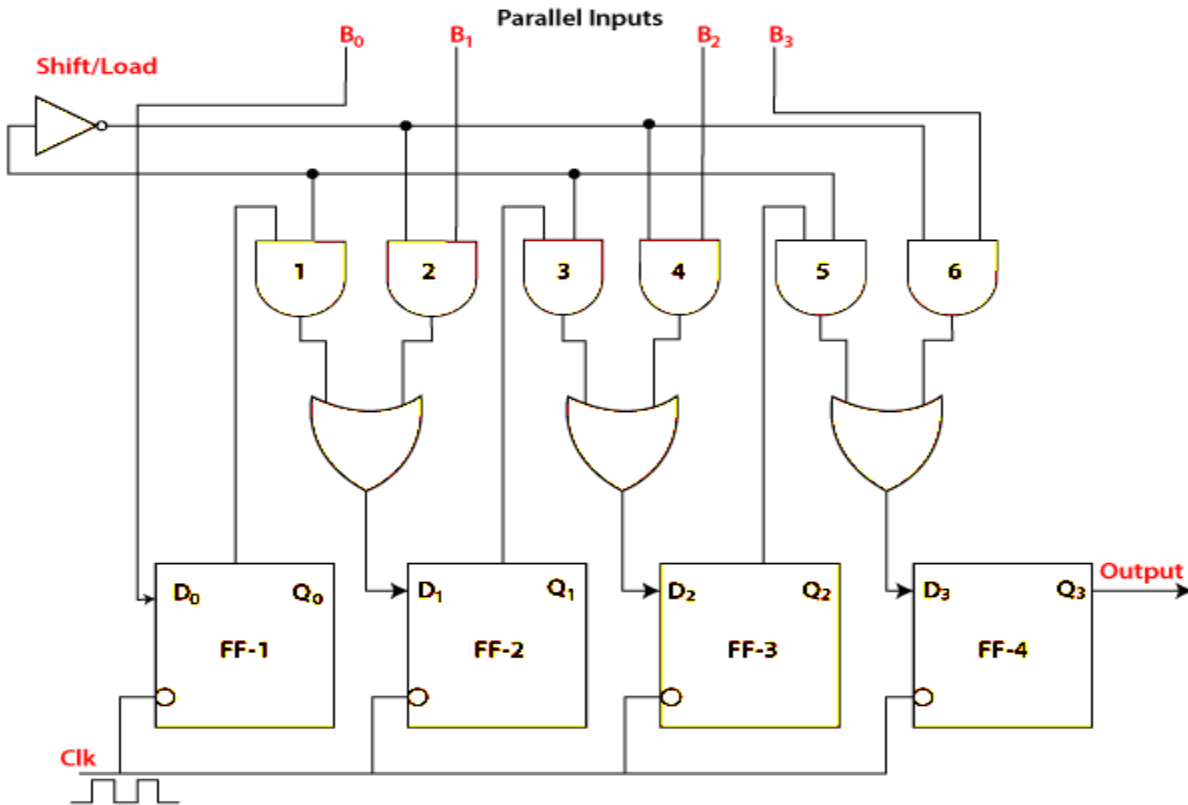


Fig 4.28: Parallel in Serial Out Shift Register

4. पॅरालल इन पॅरालल आऊट शिफ्ट रजिस्टर (Parallel in Parallel out Shift Register)

पॅरालल इन पॅरालल आऊट शिफ्ट रजिस्टर (Parallel In Parallel Out Shift Register) मध्ये, इनपुट आणि आउटपुट रजिस्टरमध्ये समांतर पद्धतीने दिले जातात. फ्लिप फ्लॉपच्या डेटा इनपुट D₀, D₁, D₂ आणि D₃ कडे पाठवले जातात. नकारात्मक क्लॉक पल्सची Edge input दिल्यावर बायनरी इनपुटचे बिट फ्लिप फ्लॉपवर लोड केले जातात. सर्व बिट्स लोड करण्यासाठी क्लॉक पल्स आवश्यक आहे. आउटपुट बाजूला, लोड केलेले बिट्स दिसतात.

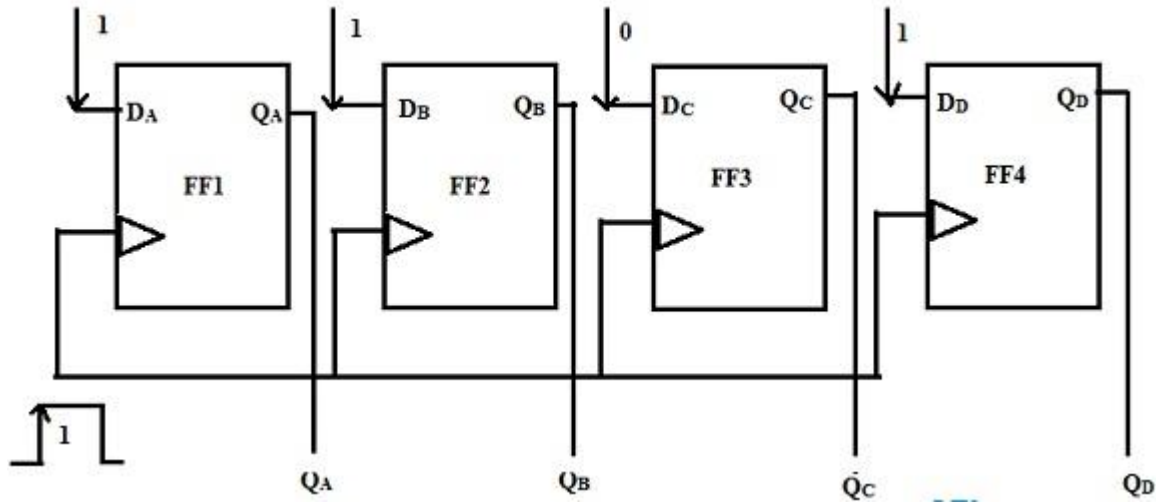


Fig 4.29: Parallel in Parallel Out Shift Register

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

Data च्या प्रत्येक बिटला एका स्थानाने डावीकडे हलवल्यानंतर बायनरी संख्या ही मूळ संख्येचा 2 ने गुणाकार करून तयार केलेल्या संख्येशी समतुल्य असेल. त्याच प्रकारे, संख्येचा प्रत्येक बिट उजवीकडे हलवल्यानंतर बायनरी संख्या मूळ संख्येला 2 ने भागून तयार केलेल्या संख्येच्या समतुल्य एक स्थान असेल.

शिफ्ट रजिस्टर वापरून गुणाकार आणि भागाकार ऑपरेशन करण्यासाठी, डेटा रजिस्टरमध्ये डावीकडे किंवा उजवीकडे दोन्ही दिशेने हलविला जाणे आवश्यक आहे. अशा नोंदणींना बायडायरेक्शनल शिफ्ट रजिस्टर (Bidirectional shift register) म्हणतात.

आकृती 4.30 : 4-बिट बायडायरेक्शनल शिफ्ट रजिस्टर (Bidirectional shift register) ची आकृती आहे. जेथे DR "सिरियल राईट शिफ्ट डेटा इनपुट", DL "डावी शिफ्ट डेटा इनपुट" आहे, आणि M हा "मोड सिलेक्ट इनपुट" आहे.

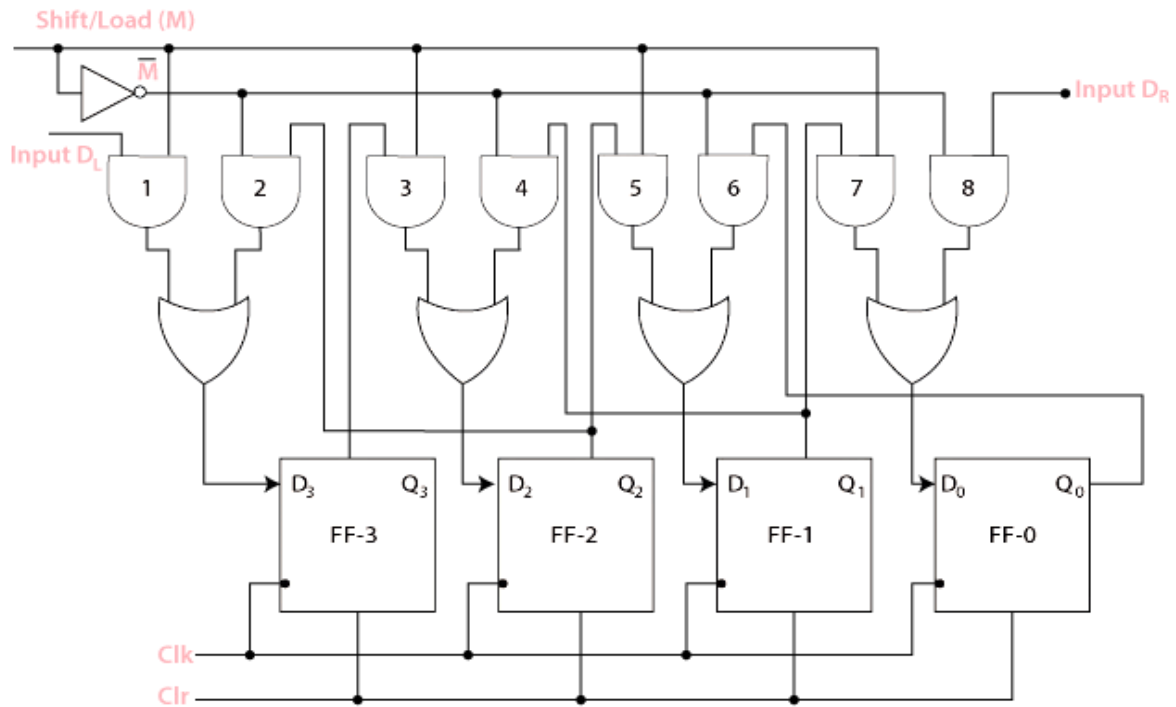


Fig 4.30: 4-bit Shift Register Block Diagram

ऑपरेशन्स**1) उजवीकडे शिफ्ट ऑपरेशन (M=1)**

- पहिला, तिसरा, पाचवा आणि सातवा आणि गेट्स सक्षम केले जातील, परंतु दुसरे, चौथे, सहावे आणि आठवे आणि गेट्स अक्षम केले जातील.
- जेव्हा क्लॉक पल्स लागू केली जाते तेव्हा डेटा इनपुट DR वर उपस्थित असलेला डेटा चौथ्या फ्लिप फ्लॉपपासून पहिल्या फ्लिप फ्लॉपवर हलविला जातो. अशा प्रकारे, शिफ्ट राईट ऑपरेशन होते.

२) डावीकडे शिफ्ट ऑपरेशन (M=शून्य (0))

- दुसरा, चौथा, सहावा आणि आठवा आणि गेट्स सक्षम(Enable) केले जातील, परंतु AND गेट्स पहिला, तिसरा, पाचवा आणि सातवा अक्षम(Disable) केला जाईल.
- जेव्हा घड्याळाची पल्स लागू केली जाते तेव्हा डेटा इनपुट DR वर उपस्थित असलेला डेटा पहिल्या फ्लिप फ्लॉपपासून चौथ्या फ्लिप फ्लॉपवर हलविला जातो. अशा प्रकारे, शिफ्ट राईट ऑपरेशन होते.

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

युनिव्हर्सल शिफ्ट रजिस्टर

एक रजिस्टर जिथे डेटा एका दिशेने हलवला जातो त्याला "एक-दिशात्मक" शिफ्ट रजिस्टर म्हणून ओळखले जाते. एक रजिस्टर ज्यामध्ये डेटा दोन्ही दिशेने शिफ्ट केला जातो तो "द्वि-दिशात्मक" शिफ्ट रजिस्टर म्हणून ओळखला जातो. "युनिव्हर्सल" शिफ्ट रजिस्टर हे एक विशेष प्रकारचे रजिस्टर आहे जे समांतर पद्धतीने डेटा लोड करू शकते आणि डेटा दोन्ही दिशांना, म्हणजे उजवीकडे आणि डावीकडे शिफ्ट करू शकते.

इनपुट M, म्हणजे, मोड कंट्रोल इनपुट, समांतर लोडिंग ऑपरेशन करण्यासाठी 1 वर सेट केले आहे. जर हे इनपुट शून्य (0) वर सेट केले असेल, तर सीरियल शिफ्टिंग ऑपरेशन केले जाईल. जर आपण मोड कंट्रोल इनपुट जमिनीशी कनेक्ट केले, तर सर्किट "द्वि-दिशात्मक" रजिस्टर म्हणून काम करेल.

जेव्हा इनपुट सीरियल इनपुटवर दिले जाते, तेव्हा रजिस्टर "सीरियल लेफ्ट" ऑपरेशन करते. जेव्हा 'डी' ला इनपुट पास केले जाते, तेव्हा रजिस्टर सीरियल राइट ऑपरेशन करते.

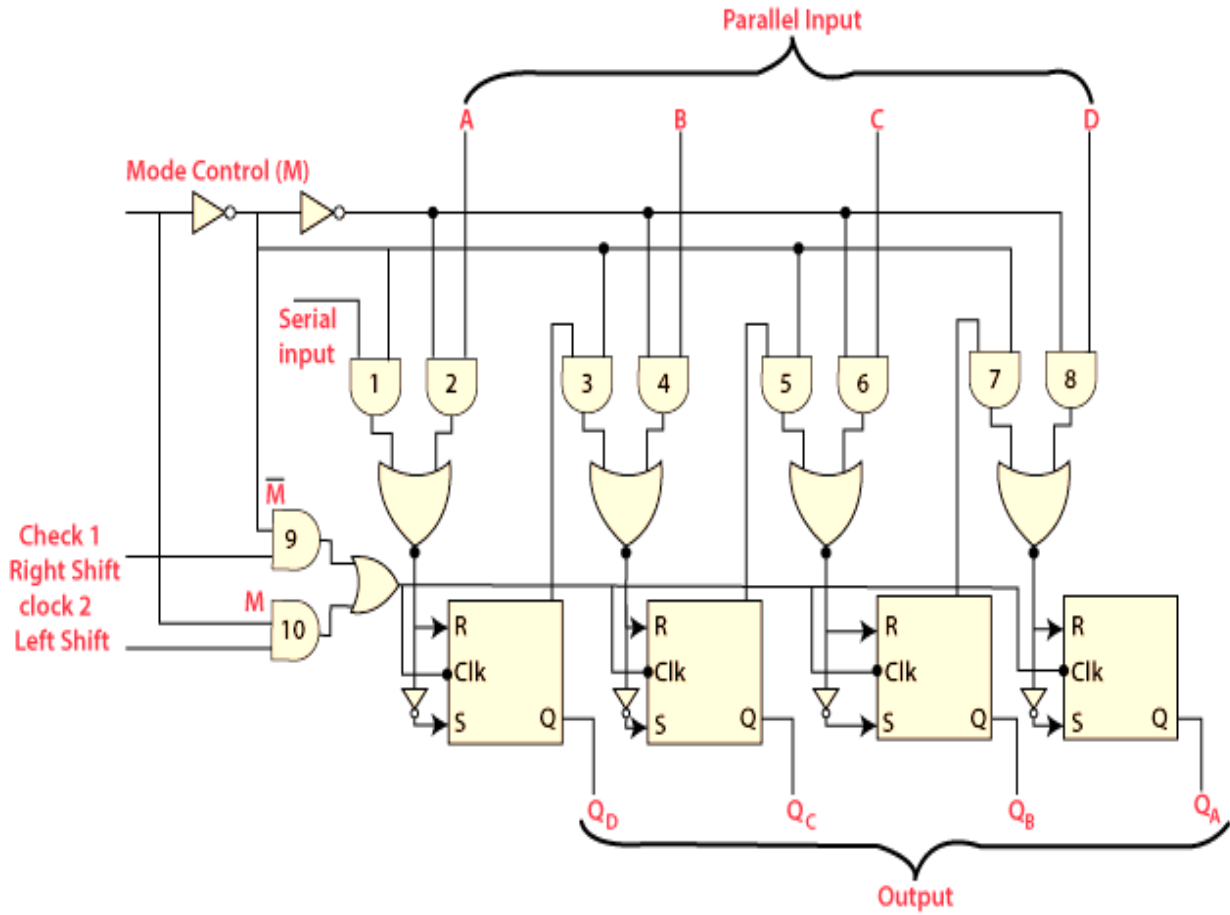


Fig 4.31 : 4 Bit Universal Shift Register

शिफ्ट रजिस्टर्सचे अनुप्रयोग

- शिफ्ट रजिस्टर्सचा वापर तात्पुरत्या डेटा स्टोरेजसाठी केला जातो.
- शिफ्ट रजिस्टर्सचा वापर डेटा ट्रान्सफर आणि डेटा मॅनिप्युलेशनसाठी देखील केला जातो.
- सिरियल-इन सिरियल-आउट आणि पॅरलल-इन पॅरलल-आउट शिफ्ट रजिस्टर्सचा उपयोग डिजिटल सर्किट्समध्ये वेळ विलंब करण्यासाठी केला जातो.
- सिरियल-इन पॅरलल-आउट शिफ्ट रजिस्टर्सचा वापर सिरियल डेटाला समांतर डेटामध्ये रूपांतरित करण्यासाठी केला जातो अशा प्रकारे ते संप्रेषण लाइनमध्ये (suppression) वापरले जातात जेथे डेटा लाइनचे (data line) अनेक समांतर (data line) मध्ये (Parallel) डीमल्टीप्लेक्सिंग आवश्यक असते.
- समांतर डेटाला सिरियल डेटामध्ये रूपांतरित करण्यासाठी सिरियल आउट शिफ्ट रजिस्टरमधील समांतर वापरले जाते.

रजिस्टर्सचे विविध प्रकार

1. MAR किंवा मेमरी ॲड्रेस रजिस्टर
2. Program काउंटर
3. संचयक नोंदणी
4. MDR किंवा मेमरी डेटा रजिस्टर
5. इंडेक्स रजिस्टर

6. मेमरी बफर रजिस्टर
7. डेटा रजिस्टर

4.5 काउंटर (Counter)

डिजिटल बायनरी काउंटर

डिजिटल बायनरी काउंटर हे binary संख्या मोजण्यासाठी वापरले जाते. डिजिटल काउंटरमध्ये प्रामुख्याने फ्लिप-फ्लॉप आणि काही कॉम्बिनेशनल सर्किटचा वापर विशेष वैशिष्ट्यांसाठी केला जातो.

फ्लिप-फ्लॉप clock pulse वर चालते. त्यामुळे या pulse चा वापर करून काउंटर वर-खाली up-down मोजले जाईल. Clock pulse च्या वेळेचा परिणाम काउंटरमधील वेळेच्या विलंबावर होईल. त्यामुळे clock pulse च्या सिग्नलचा कालावधी १ सेकंदाचा असल्याने या काउंटरमध्येही बदल करता येऊ शकतो.

Pulse मोजण्यासाठी वापरल्या जाणाऱ्या विशिष्ट प्रकारच्या सीक्वेंशियल सर्किटला काउंटर म्हणून ओळखले जाते किंवा फ्लिप फ्लॉपचा संग्रह जिथे घड्याळाचा सिग्नल लावला जातो त्याला काउंटर म्हणून ओळखले जाते.

काउंटर फ्लिप फ्लॉपच्या विस्तृत अनुप्रयोगांपैकी एक आहे. clock pulse वर आधारित, काउंटरच्या आउटपुटमध्ये पूर्वनिर्धारित स्थिती असते. काउंटरच्या आउटपुटचा वापर करून clock states ची संख्या मोजता येते.

Table 4.26: Counter Truth Table

Clock	Counter output		State number	Decimal counter output
	Q _B	Q _A		
Initially	0	0	-	0
1 st	0	1	1	1
2 nd	1	0	2	2
3 rd	1	1	3	3
4 th	0	0	4	0

डिजिटल लॉजिक सर्किटमध्ये दोन प्रकारचे काउंटर आहेत जे बिट्सची संख्या मोजण्यासाठी वापरले जातात आणि हे प्रकार फ्लिप फ्लॉपवर लावलेल्या clock signal वर अवलंबून असतात.

दोन प्रकारचे काउंटर आहेत:

1. सिंक्रोनस काउंटर (Synchronous counter) क्लॉक दिलेली असतात या काउंटरमध्ये सर्व फ्लिप फ्लॉपना एकाच दिलेली असतात.

हे असे काउंटर आहेत ज्यामध्ये सर्व फ्लिप फ्लॉपसाठी सार्वत्रिक क्लॉक पल्स वापरतात. सिंक्रोनस काउंटरचा सर्किट डायग्राम खाली दिलेली आहे:

सिंक्रोनस काउंटरमध्ये, सर्व फ्लिप फ्लॉपच्या क्लॉक पल्स इनपुटवर समान एकाचवेळी क्लॉक पल्स दिली जाते. सर्व फ्लिप फ्लॉपद्वारे तयार केलेले क्लॉक पल्स सिग्नल एकमेकांसारखेच असतात. खाली आकृती 4.32 : 2-बिट सिंक्रोनस काउंटरचा आकृती आहे ज्यामध्ये पहिल्या फ्लिप फ्लॉपचे इनपुट, म्हणजे, **FF-A, 1** वर सेट केले आहेत. त्यामुळे, पहिला फ्लिप फ्लॉप टॉगल फ्लिप-फ्लॉप म्हणून कार्य करेल. पहिल्या फ्लिप फ्लॉपचे आउटपुट पुढील **JK** फ्लिप फ्लॉपच्या दोन्ही इनपुटला दिले जाते.

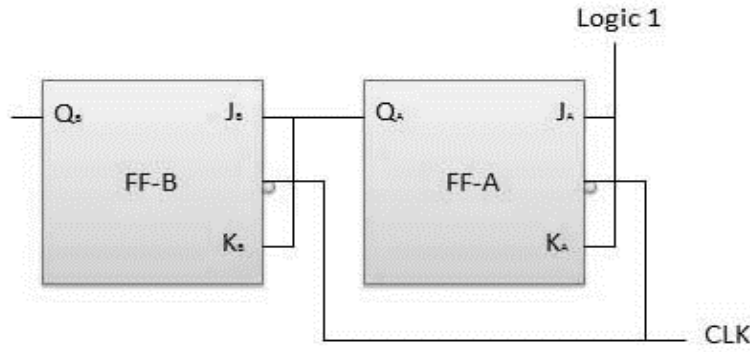


Fig 4.32: 2- bit synchronous counter

2. असिंक्रोनस (Asynchronous counter) काउंटर किंवा रिपल (ripple Counter) काउंटर व काउंटरमध्ये सर्व फ्लिप फ्लॉपना एकाच वेळी क्लॉक दिलेली नसते.

असिंक्रोनस काउंटरमध्ये, वर्तमान काउंटरचे आउटपुट पुढील काउंटरच्या इनपुटवर जाते. तर, काउंटर साखळीप्रमाणे जोडलेले आहेत. या प्रणालीचा दोष असा आहे की ते मोजणी विलंब निर्माण करते, आणि प्रसार विलंब देखील मोजणीच्या टप्प्यात होतो. सिंक्रोनस काउंटर ही कमतरता दूर करण्यासाठी डिझाइन केले आहे.

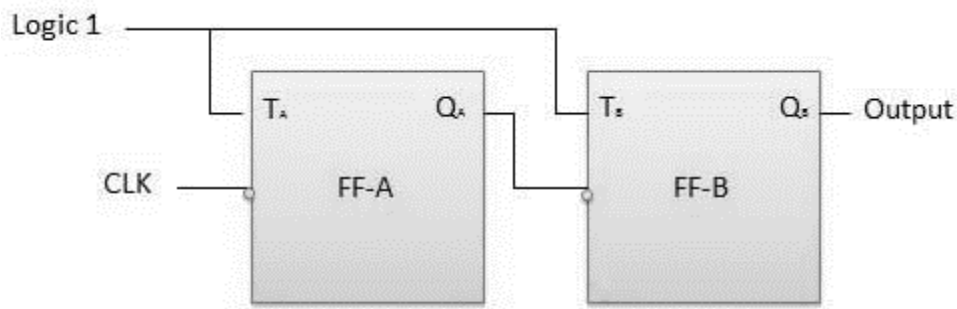


Fig 4.33: 2 Bit Asynchronous Counter

Table 4.27: Differences Between Synchronous and Asynchronous Counts

S.NO	सिंक्रोनस काउंटर (Synchronous Counter)	असिंक्रोनस काउंटर (Asynchronous Counter)
1	सिंक्रोनस काउंटरमध्ये आम्ही एक सार्वत्रिक क्लॉक पल्स वापरतो जे सर्किटमधून सर्व फ्लिप फ्लॉपसाठी सामान्य आहे.	असिंक्रोनस काउंटरमध्ये मुख्य क्लॉक पल्स फक्त पहिल्या फ्लिप फ्लॉपवर लागू केले जाते आणि नंतर उर्वरित फ्लिप फ्लॉपसाठी मागील फ्लिप फ्लॉपचे आउटपुट क्लॉक पल्स म्हणून घेतले जाते.
2	सिंक्रोनस काउंटर असिंक्रोनस काउंटरच्या तुलनेत अधिक वेगवान आहे.	असिंक्रोनस काउंटर ऑपरेशन सिंक्रोनस काउंटरच्या तुलनेत हळू आहे.

3	सिंक्रोनस काउंटर कोणत्याही डीकोडिंग त्रुटी निर्माण करत नाही.	असिंक्रोनस काउंटर डीकोडिंग त्रुटी निर्माण करते.
4	राज्यांची संख्या वाढल्यामुळे सिंक्रोनस काउंटर डिझाईनिंग तसेच अंमलबजावणी कठीण आहे.	असिंक्रोनस काउंटर डिझाइन करणे तसेच अंमलबजावणी करणे खूप सोपे आहे.
5	सिंक्रोनस काउंटर कोणत्याही इच्छित गणना क्रमाने कार्य करेल.	असिंक्रोनस काउंटर केवळ निश्चित गणना अनुक्रमात (UP/DOWN) कार्य करेल.
6	सिंक्रोनस काउंटर उदाहरणे आहेत: रिंग काउंटर, जॉन्सन काउंटर.	असिंक्रोनस काउंटर उदाहरणे आहेत: रिपल अप काउंटर, रिपल डाउन काउंटर.
7	सिंक्रोनस काउंटरमध्ये, प्रसार विलंब कमी आहे.	असिंक्रोनस काउंटरमध्ये, उच्च प्रसार विलंब होतो.
8	प्रसार विलंब (propagation delay) खूप कमी आहे	सिंक्रोनस काउंटरच्या तुलनेत प्रसार विलंब जास्त आहे
9	हे असिंक्रोनस काउंटरपेक्षा वेगवान आहेत	यांचे कार्य सिंक्रोनस काउंटरपेक्षा हळू चलते.
10	डिझाइन करण्यासाठी मोठ्या प्रमाणात लाजिक गेट्स आवश्यक आहेत	लॉजिक गेट्सची कमी संख्या आवश्यक आहे.

4.5.1 मोड्युलस ऑफ काउंटर (Modulus of counter)

स्टेट्स ची संख्या किंवा मोजणी क्रम ज्याद्वारे विशिष्ट काउंटर त्याच्या मूळ पहिल्या स्थितीत परत येण्याआधी पुढे सरकते त्याला मॉड्युलस (MOD) म्हणतात. दुसऱ्या शब्दांत, मापांक (किंवा फक्त मॉड्युलो) ही काउंटर मोजत असलेल्या राज्यांची संख्या आहे आणि काउंटरची भागाकार संख्या आहे.

कोणत्याही इच्छित संख्येपर्यंत मोजण्यासाठी अनेक काउंटर मालिकेत जोडलेले आहेत. काउंटर सर्किट जी संख्या मोजू शकते त्याला "मॉड" किंवा "मॉड्युलस" म्हणतात. n बिट्स मोजल्यानंतर काउंटर स्वतः रीसेट झाल्यास त्याला "मॉड-एन काउंटर" "मॉड्युलोन काउंटर" म्हणतात, जिथे n पूर्णांक आहे.

Mod n काउंटर 0 ते $2^n - 1$ पर्यंत मोजू शकतो. मॉड 4 काउंटर, मॉड 8 काउंटर, मॉड 16 काउंटर आणि मॉड 5 काउंटर इत्यादी अनेक प्रकारचे काउंटर उपलब्ध आहेत.

डिझाइन मोड - एन सिंक्रोनस काउंटर / Design Mod – N synchronous Counter

N चे मूल्य 2 च्या पॉवरपेक्षा वेगळे असू शकते. तसेच, मोजणी क्रम यादृच्छिक असू शकतो उदाहरणार्थ काही चक्रीय कोड (8421, 2423 इ.). मॉड N (एन) आणि कोणत्याही मोजणी क्रमासाठी डिझाइन करण्यासाठी खालील पद्धत लागू केली आहे.

मॉड-एन काउंटरसाठी डिझाइन: डिझाइनसाठी खालील स्टेप्स आहेत.

1. फ्लिप-फ्लॉपच्या संख्येसाठी निर्णय -

उदाहरण : जर आपण मॉड एन (N) काउंटर डिझाइन करत आहोत आणि 'n' फ्लिप-फ्लॉपची संख्या आवश्यक असेल तर 'n' या समीकरणाद्वारे शोधले जाऊ शकते.

$$N \leq 2^n$$

येथे आपण Mod-10 काउंटर डिझाइन करत आहोत, म्हणून $N = 10$ आणि फ्लिप फ्लॉपची संख्या आवश्यक आहे.

$n = 3$, $10 \leq 8$ साठी, जे असत्य आहे.

$n = 4$, $10 \leq 16$ साठी, जे खरे आहे.

म्हणून मॉड-10 काउंटरसाठी आवश्यक (FF) फ्लिपफ्लॉपची संख्या 4 आहे.

4.5.3 असिंक्रोनस किंवा रिपल काउंटर

हे काउंटर आहेत ज्यामध्ये आपण सार्वत्रिक क्लॉक पल्स वापरत नाही, मुख्य क्लॉक पल्स फक्त पहिल्या फ्लिप फ्लॉपवर लागू केले जाते आणि नंतर उर्वरित फ्लिप फ्लॉपसाठी मागील फ्लिप फ्लॉपचे आउटपुट क्लॉक पल्स म्हणून घेतले जाते.

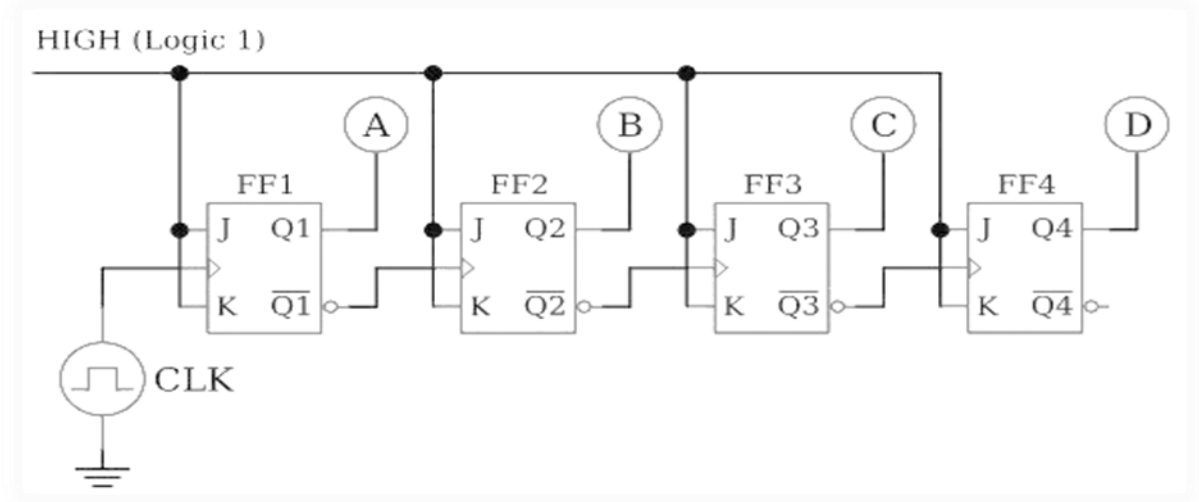


Fig4.34: Asynchronous counter using JK flip flops

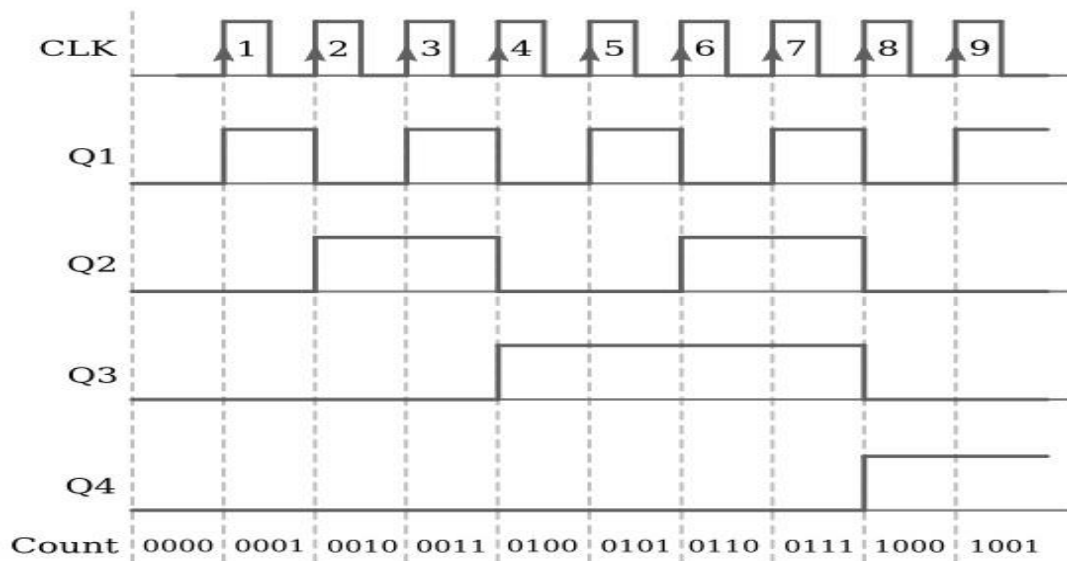


Fig 4.35: Waveforms of asynchronous counters using J K flip flops

4.5.4 रिंग काउंटर (Ring counter)

रिंग काउंटर हा शिफ्ट रजिस्टरचा एक सामान्य अनुप्रयोग आहे. रिंग काउंटर जवळजवळ शिफ्ट काउंटर सारखेच आहे. फक्त बदल असा आहे की शेवटच्या फ्लिप-फ्लॉपचे आउटपुट रिंग काउंटरच्या बाबतीत पहिल्या फ्लिप-फ्लॉपच्या इनपुटशी जोडलेले असते परंतु शिफ्ट रजिस्टरच्या बाबतीत ते आउटपुट म्हणून घेतले जाते. हे सोडले तर बाकी सर्व गोष्टी सारख्याच आहेत.

रिंग काउंटरमधील राज्यांची संख्या = वापरलेल्या फ्लिप-फ्लॉपची संख्या

तर, 4-बिट रिंग काउंटर डिझाइन करण्यासाठी आम्हाला 4 फ्लिप-फ्लॉपची आवश्यकता आहे.

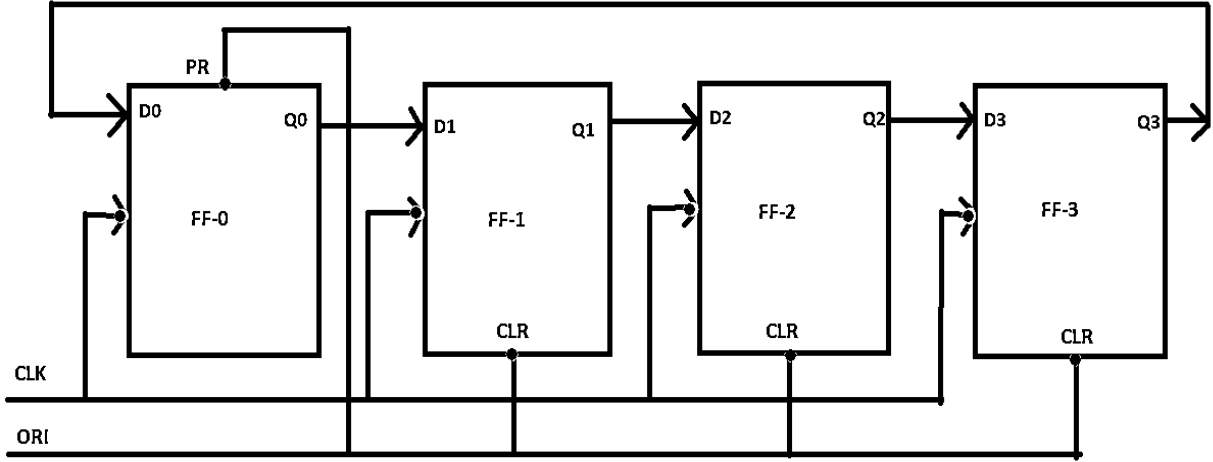


Fig 4.36: 4-bit ring counter

या आकृतीत, **clock pulse** सर्व फ्लिप-फ्लॉपवर एकाच वेळी लागू केली जाते. म्हणून, हे सिंक्रोनस काउंटर आहे. तसेच, येथे आपण प्रत्येक फ्लिप-फ्लॉपसाठी ओव्हरराइडिंग इनपुट (**ORI**) वापरतो. प्रीसेट (**PR**) आणि **Clear (CLR)** **ORI** म्हणून वापरले जातात. जेव्हा **PR 0** असेल, तेव्हा आउटपुट **1** असेल. आणि जेव्हा **CLR 0** असेल तेव्हा आउटपुट **0** असेल. **PR** आणि **CLR** दोन्ही सक्रिय लो सिग्नल आहेत जे नेहमी **0** मूल्यामध्ये कार्य करतात.

PR = 0, Q = 1

CLR = 0, Q = 0

ही दोन मूल्ये नेहमीच स्थिर असतात. ते इनपुट डी आणि क्लॉक पल्स (**CLK**) च्या मूल्यापासून स्वतंत्र आहेत. कार्यरत - येथे, **ORI FF-0** मध्ये प्रीसेट (**PR**) शी कनेक्ट केलेले आहे आणि ते **FF-1, FF-2** आणि **FF-3** मध्ये क्लियर (**CLR**) शी कनेक्ट केलेले आहे. अशा प्रकारे, आउटपुट **Q = 1** **FF-0** वर व्युत्पन्न होते, आणि उर्वरित फ्लिप-फ्लॉप आउटपुट **Q = 0** व्युत्पन्न करते. **FF-0** वर हे आउटपुट **Q = 1** प्री-सेट **1** म्हणून ओळखले जाते जे रिंग तयार करण्यासाठी वापरले जाते.

1. रिंग काउंटर हे प्रीसेट केलेले **1 ORI** कमी करून व्युत्पन्न केले जाते आणि त्या वेळी क्लॉक पल्स (**CLK**) बनते. त्यानंतर **ORI** उच्च वर केले जाते आणि कमी क्लॉक पल्स पल्स सिग्नल लागू केले जाते कारण क्लॉक पल्स (**CLK**) नकारात्मक एज ट्रिगर होते. त्यानंतर, प्रत्येक क्लॉक पल्स वर, प्रीसेट केलेले **1** पुढील फ्लिप-फ्लॉपवर हलवले जाते आणि अशा प्रकारे एक रिंग बनते. वरील टेबलवरून आपण असे म्हणू शकतो की 4-बिट रिंग काउंटरमध्ये **4 states** आहेत. **4 states** आहेत: **1 000, 0100, 0010, 0001** अशा प्रकारे चार डी फ्लिप-फ्लॉप वापरून 4-बिट रिंग काउंटर डिझाइन करू शकता. द्विस्टेड रिंग काउंटर रिंग काउंटरचे प्रकार: रिंग काउंटरचे दोन प्रकार आहेत: **1. स्ट्रेट रिंग काउंटर:** याला वन हॉट काउंटर असेही म्हणतात. या काउंटरमध्ये, शेवटच्या फ्लिप-फ्लॉपचे आउटपुट पहिल्या फ्लिप-फ्लॉपच्या इनपुटशी जोडलेले आहे. या काउंटरचा मुख्य मुद्दा असा आहे की ते अंगठीभोवती एकच (किंवा शून्य) बिट फिरवते.

येथे, पहिल्या फ्लिप-फ्लॉपमध्ये प्रीसेट (**PR**) आणि शेवटच्या तीन फ्लिप-फ्लॉपसाठी क्लॉक (**CLK**) वापरतो. येथे, आम्ही पहिल्या फ्लिप-फ्लॉपमध्ये प्रीसेट (**PR**) आणि शेवटच्या तीनसाठी क्लॉक (**CLK**) वापरतो. फ्लिप-फ्लॉप.

Table 4.28: Truth Table of ring counter

ORI	CLK	Q0	Q1	Q2	Q3
low	X	1	0	0	0
high	low	0	1	0	0
high	low	0	0	1	0
high	low	0	0	0	1
high	low	1	0	0	0

4.5.5 ट्विस्टेड रिंग काउंटर - हे स्विच-टेल रिंग काउंटर, वॉकिंग रिंग काउंटर किंवा जॉन्सन काउंटर म्हणून देखील ओळखले जाते. हे शेवटच्या शिफ्ट रजिस्टरच्या आउटपुटच्या पूरकतेला पहिल्या रजिस्टरच्या इनपुटशी जोडते आणि रिंगभोवती शून्यांनंतरचा प्रवाह फिरवते.

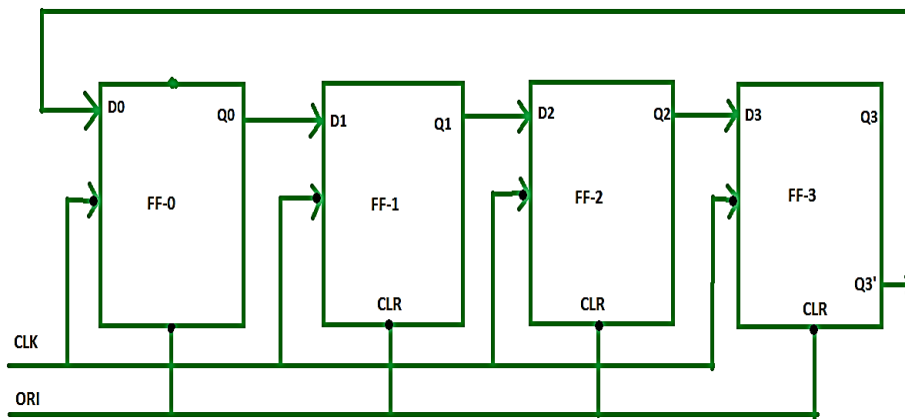


Fig 4.37: Twisted Ring Counter

येथे, सर्व फ्लिप-फ्लॉप ना क्लॉक पल्स (CLK) अफ्लाय केला जातो. ट्विस्टेड रिंग काउंटरमध्ये, States ची संख्या = 2 X फ्लिप-फ्लॉपची संख्या.

Table 4.29 : Truth Table of Twisted Ring Counter

Clock Input	Q ₃	Q ₂	Q ₁	Q ₀
1	0	0	0	0
2	0	0	0	1
3	0	0	1	1
4	0	1	1	1
5	1	1	1	1
6	1	1	1	0
7	1	1	0	0
8	1	0	0	0

4.5.5 रिंग काउंटर आणि जॉन्सन काउंटर मधील फरक

Table 4.30: Difference Between Ring Counter and Twisted Ring/Johnson Counter

तुलना गुणांचा आधार	रिंग काउंटर (Ring Counter)	ट्विस्टेड रिंग / जॉन्सन काउंटर (Twisted /Johnson Cunter)
अनुपयोग Applications	रिंग काउंटर बहुतेक क्रमिक अंदाजे प्रकार ADC आणि स्टेपर मोटर नियंत्रण मध्ये वापरले जाते.	जॉन्सन काउंटरला वॉकिंग काउंटर किंवा स्विचिंग टेल काउंटर असेही संबोधले जाते आणि ते मुख्यतः फेज शिफ्ट किंवा फंक्शन जनरेटरमध्ये वापरले जाते.
आउटपुट Output	रिंग काउंटरमध्ये, शेवटच्या फ्लिप फ्लॉपचे आउटपुट पहिल्या फ्लिप फ्लॉपच्या इनपुटशी जोडलेले असते.	जॉन्सन काउंटरमध्ये, शेवटच्या फ्लिप फ्लॉपचा आउटपुट बार किंवा क्यू-बार पहिल्या फ्लिप फ्लॉपच्या इनपुटशी जोडलेला असतो.
डीकोडिंग Decoding	रिंग काउंटरमध्ये डीकोडिंग करणे सोपे आहे कारण राज्यांची संख्या फ्लिप फ्लॉपच्या संख्येइतकी आहे.	रिंग टू काउंटरच्या तुलनेत जॉन्सन काउंटर डीकोड करणे जटिल आहे.
फ्लिप-फ्लॉपची संख्या वि वापरलेल्या संभाव्य स्टेट्स ची संख्या Number Of Flip-Flops Vs Number of Possible States Used	फ्लिप-फ्लॉपची संख्या ही वापरलेल्या संभाव्य स्टेट्स ची संख्या जर 'n' ही रिंग काउंटरमध्ये वापरल्या जाणाऱ्या फ्लिप फ्लॉपची संख्या असेल तर संभाव्य अवस्थांची संख्या 'n' असेल. म्हणजे स्टेट्स ची संख्या वापरलेल्या फ्लिप फ्लॉपच्या संख्येइतकी आहे.	जॉन्सन काउंटरमध्ये वापरल्या जाणाऱ्या फ्लिप फ्लॉपची संख्या 'n' असल्यास, वापरलेल्या स्टेट्स ची एकूण संख्या '2n' आहे.

मुद्द्यांवर आधारित फरक

ऑपरेशन (Operation):

- जॉन्सन काउंटर हा शिफ्ट रजिस्टरचा एक प्रकार आहे जिथे शेवटच्या टप्प्याचे आउटपुट पहिल्या टप्प्याच्या इनपुटवर परत दिले जाते. हे 0s आणि 1s चा क्रम प्रसारित करते, प्रत्येक राज्यासाठी एक अद्वितीय नमुना तयार करते.
- रिंग काउंटर हा देखील शिफ्ट रजिस्टरचा एक प्रकार आहे, परंतु तो टप्प्यांमध्ये एकच 1 बिट प्रसारित करतो, परिणामी प्रत्येक राज्यात फक्त एक टप्पा उच्च (1) असतो तर इतर कमी (0) असतो.

आउटपुट स्थिती (Output States):

- 'n' टप्प्यांसह जॉन्सन काउंटरमध्ये, पुनरावृत्ती करण्यापूर्वी काउंटर 2n अवस्थांमधून जातो. यात 2n संभाव्य आउटपुट अवस्था आहेत, ज्यामुळे अनुक्रमांची विस्तृत श्रेणी मिळते.
- 'n' टप्प्यांसह रिंग काउंटरमध्ये फक्त n संभाव्य आउटपुट अवस्था असतात, कारण एका वेळी फक्त एकच टप्पा जास्त असतो (1). याचा परिणाम आउटपुट अनुक्रमांचा अधिक मर्यादित संच होतो.

फ्लिप-फ्लॉपची संख्या (Number of Flip-Flops):

- जॉन्सन काउंटरला 'n' टप्प्यांसाठी $2n$ फ्लिप-फ्लॉपची आवश्यकता असते कारण त्याला वर्तमान स्थिती आणि त्याचे पूरक दोन्ही संग्रहित करणे आवश्यक असते.
- रिंग काउंटरला फक्त 'n' टप्प्यांसाठी 'n' फ्लिप-फ्लॉपची आवश्यकता असते, कारण ते त्यांच्यामध्ये एकच 1 बिट प्रसारित करते.

अनुपयोग (Applications):

- जॉन्सन काउंटर बऱ्याचदा अशा ऍप्लिकेशन्ससाठी वापरले जातात ज्यांना अनन्य स्थितींचा क्रम आवश्यक असतो, जसे की वारंवारता विभागणी, वेळ विलंब निर्मिती आणि Pseudo random संख्या निर्मिती.
- डिकोडिंग ऑपरेशन्स, विशिष्ट फंक्शन्स निवडणे आणि वेळ विलंब तयार करणे यासारख्या अनेक टप्प्यांमध्ये एकच उच्च आउटपुट तयार करणे आवश्यक असलेल्या अनुप्रयोगांमध्ये रिंग काउंटरचा वापर वारंवार केला जातो.

अप काउंटर UP counter

हे समकालिक काउंटर 0 ते 15 (4-बिट काउंटर) पर्यंत मोजले जाते. टी-फ्लिप फ्लॉप (सामान्य इनपुटसह जेके-फ्लिप फ्लॉप) आणि डी-फ्लिप फ्लॉप वापरून अप काउंटर डिझाइन केले जाऊ शकते. या दोन्ही फ्लिप-फ्लॉपचे कॉन्फिगरेशन वेगळे आहे.

Flip-flops FF0, FF1, FF2 चे आउटपुट अनुक्रमे Q0, Q1, Q2 द्वारे दर्शविल्या जाणाऱ्या प्रत्येक बिट गणनेसह 3-बिट काउंटरचा विचार करा. मग राज्य सारणी असेल:

Table 4.31: 3-bit Up Counter Truth Table

Clock Input	Q ₂	Q ₁	Q ₀
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

4.5.6 डाउन काउंटर (Down Counter)

डाउन काउंटर 15 ते 0 (4-बिट काउंटर) पर्यंत उतरत्या क्रमाने मोजले जाते. डाउन काउंटर टी-फ्लिप फ्लॉप आणि डी-फ्लिप फ्लॉप वापरून देखील डिझाइन केले जाऊ शकते.

अनुक्रमे फ्लिप-फ्लॉप FF0, FF1, FF2 चे आउटपुट Q0, Q1, Q2 द्वारे दर्शविलेल्या प्रत्येक बिटसह 3-बिट काउंटरचा विचार करा. डाउन काउंटरसाठी राज्य सारणी खाली दिली आहे:

State सारणीपेक्षा Q_0, Q_1, Q_2, Q_3 as काउंटरच्या 4 बिट्सचा विचार करा.

Table 4.33 Truth Table fo Decade Counter

Clock Input	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

मोड-१० काउंटर / दशक काउंटर (बीसीडी काउंटर)

दशक काउंटर (Decade counter/MOD 10 Counter)

एक दशक काउंटर दहा वेगवेगळ्या (states) राज्यांची गणना करतो आणि नंतर त्याच्या सुरुवातीच्या स्थितीवर रीसेट करतो. एक साधा दशक काउंटर शुन्य (0) ते 9 पर्यंत मोजला जाईल परंतु आम्ही दशक काउंटर देखील बनवू शकतो जे शुन्य (0) ते 15 (4 बिट काउंटरसाठी) दरम्यान कोणत्याही दहा राज्यांमधून जाऊ शकतात.

बीसीडी (बायनरी कोडेड डेसिमल) किंवा डिकेड काउंटर सर्किट हा डिजिटल इलेक्ट्रॉनिक्समधील एक आवश्यक घटक आहे, जो प्रत्येक दशांश अंकाचे प्रतिनिधित्व करून बायनरी स्वरूपात 0 ते 9 पर्यंत क्रमवार मोजण्यासाठी डिझाइन केलेले आहे. त्याबद्दल अधिक समजून घेऊ.

बायनरी कोडेड दशांश (BCD) हा एक सीरियल डिजिटल काउंटर आहे ज्यामध्ये दहा अंक मोजले जातात. आणि प्रत्येक नवीन क्लॉक पल्स इनपुटसाठी ते रीसेट केले जाते. हे आउटपुटच्या 10 अद्वितीय संयोजनांमधून जाऊ शकते, याला "डेकेड काउंटर" असेही म्हणतात. बीसीडी काउंटर 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000 आणि 1001. आणि असेच मोजू शकते.

4 बिट बायनरी काउंटर 16 (24) आउटपुटपैकी कोणतेही सहा आउटपुट वगळून दशक काउंटर म्हणून काम करेल. दशक काउंटरसाठी काही उपलब्ध ICs आहेत जे आपण आपल्या सर्किटमध्ये सहजपणे वापरू शकतो, जसे की 74LS90. हे असिंक्रोनस दशक काउंटर आहे.

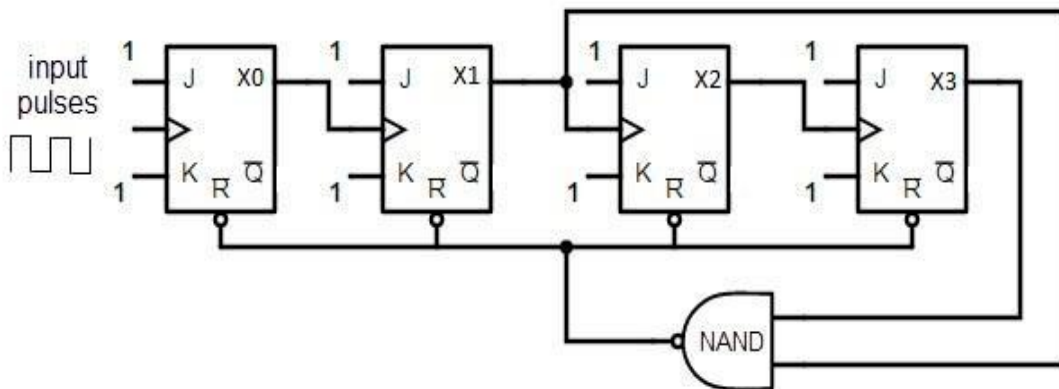


Fig 4.39: Mod-N counter

वरील आकृती जेके फ्लिप फ्लॉपसह तयार केलेले दशक काउंटर दर्शवते. **J** आउटपुट आणि **K** आउटपुट लॉजिक 1 शी

जोडलेले आहेत. प्रत्येक फ्लिप फ्लॉपचे क्लॉक पल्स इनपुट शेवटच्या फ्लिप फ्लॉपच्या आउटपुटशी जोडलेले आहे. **NAND** गेटचे आउटपुट सर्व फ्लिप फ्लॉप्सना **clear** इनपुट '**CLR**' च्या समांतर जोडलेले आहे. हा रिपल काउंटर **16** म्हणजे **24** पर्यंत मोजू शकतो. दशक काउंटर ऑपरेशन जेव्हा दशक काउंटर प्रारंभिक स्थितीमध्ये असतो, तेव्हा संख्या **0000** च्या बरोबरीची असते. काउंटर सायकलचा हा पहिला टप्पा आहे. जेव्हा आपण काउंटर सर्किटला क्लॉक पल्स सिग्नल इनपुट कनेक्ट करतो, तेव्हा सर्किट बायनरी अनुक्रम मोजेल. पहिली क्लॉक पल्स **9 (1001)** पर्यंत मोजण्यासाठी सर्किट बनवू शकते. पुढील क्लॉक पल्स **10 (1010)** मोजण्यासाठी पुढे जाते. नंतर **X1** आणि **X3** पोर्ट जास्त असतील. आपल्याला माहित आहे की उच्च इनपुटसाठी, **NAND** गेट आउटपुट कमी असेल. **NAND** गेट आउटपुट क्लिअर इनपुटशी जोडलेले आहे, त्यामुळे ते दशक काउंटरमध्ये सर्व फ्लिप फ्लॉप स्टेज रीसेट करते. याचा अर्थ गणना **9** नंतरची क्लॉक पल्स पुन्हा गणना **0** पासून गणना सुरू करेल.

Table 4.34: Truth Table of Decade Counter

Clock Input	D	C	B	A
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
0	0	0	0	0 (Reset)

वरील सारणी दशक काउंटरच्या मोजणी ऑपरेशनचे वर्णन करते. हे इनपुट पल्सच्या दशांश मोजणीसाठी सर्किटची संख्या दर्शवते. जेव्हा संख्या $10 (1010)_2$ पर्यंत पोहोचते तेव्हा **NAND** गेट आउटपुट शून्य होते.

NAND गेट **X1** आणि **X3** च्या इनपुटद्वारे गणना डीकोड केली जाते. गणना **10** नंतर, लॉजिक गेट **NAND** त्याचे आउटपुट **1** ते **0** पर्यंत ट्रिगर करेल आणि ते सर्व फ्लिप फ्लॉप रीसेट करेल.

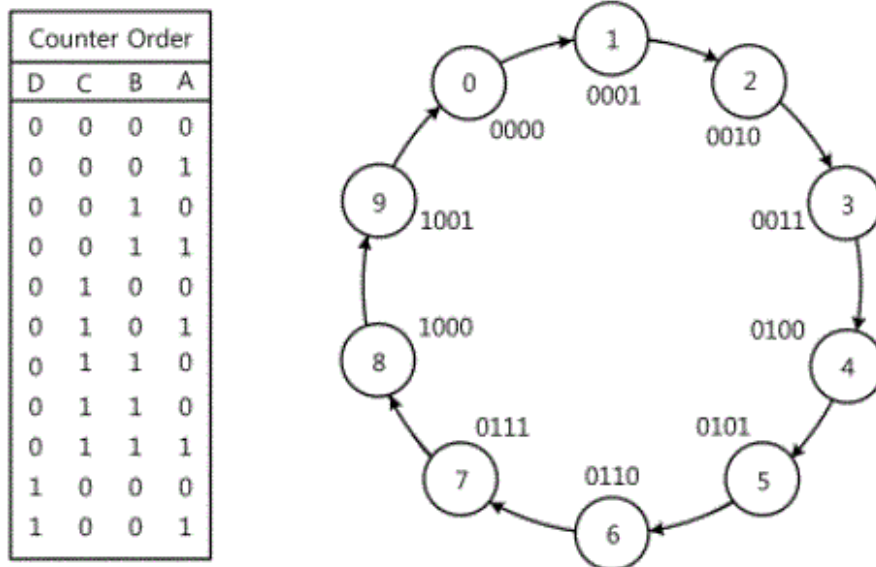


Figure 4.40: State diagram of decade counters

जर आपण दशक काउंटर सर्किट आकृतीचे निरीक्षण केले, तर त्यात चार टप्पे आहेत, ज्यामध्ये प्रत्येक टप्प्यात एकच फ्लिप फ्लॉप आहे. म्हणून ते 16 बिट्स किंवा 16 संभाव्य अवस्था मोजण्यास सक्षम आहे, ज्यामध्ये फक्त 10 वापरले जातात. गणना 0000 (शून्य) पासून 1001 (9) पर्यंत सुरू होते आणि नंतर NAND गेट सर्किट रीसेट करेल. कोणत्याही इच्छित संख्येपर्यंत मोजण्यासाठी अनेक काउंटर मालिकेत जोडलेले आहेत. काउंटर सर्किट जी संख्या मोजू शकते त्याला "मॉड" किंवा "मॉड्युलस" म्हणतात. n बिट्स मोजल्यानंतर काउंटर स्वतः रीसेट झाल्यास त्याला "मॉड-एन काउंटर" "मॉड्युलोन काउंटर" म्हणतात, जिथे n पूर्णांक आहे.

Mod N काउंटर 0 ते $2n-1$ पर्यंत मोजू शकतो.

मॉड 4 काउंटर, मॉड 8 काउंटर, मॉड 16 काउंटर आणि मॉड 5 काउंटर इत्यादी अनेक प्रकारचे काउंटर उपलब्ध आहेत.

डिझाइन मोड - एन सिंक्रोनस काउंटर / Design Mod - N synchronous Counter

N चे मूल्य 2 च्या पॉवरपेक्षा वेगळे असू शकते. तसेच, मोजणी क्रम यादृच्छिक असू शकतो उदाहरणार्थ काही चक्रीय कोड (8421, 2423 इ.). मॉड एन आणि कोणत्याही मोजणी क्रमासाठी डिझाइन करण्यासाठी खालील पद्धत लागू केली आहे.

मॉड-एन काउंटरसाठी डिझाइन:

डिझाइनसाठी पायऱ्या आहेत -

पायरी 1 : फ्लिप-फ्लॉपच्या संख्येसाठी निर्णय -

उदाहरण : जर आपण मॉड एन काउंटर डिझाइन करत आहोत आणि n फ्लिप-फ्लॉपची संख्या आवश्यक असेल तर n या समीकरणाद्वारे शोधले जाऊ शकते.

$$N \leq 2^n$$

येथे आपण Mod-10 काउंटर डिझाइन करत आहोत, म्हणून $N = 10$ आणि फ्लिप फ्लॉपची संख्या आवश्यक आहे.

$n = 3$, $10 \leq 8$ साठी, जे असत्य आहे.

$n = 4$, $10 \leq 16$ साठी, जे खरे आहे.

म्हणून मॉड-10 काउंटरसाठी आवश्यक FF संख्या 4 आहे.

स्टेप 2 : फ्लिप फ्लॉपचे उत्तेजना सारणी लिहा - येथे T FF वापरला आहे

Table 4.35: Excitation table of T Flip Flop

Previous state(Q_n)	Next state(Q_{n+1})	T
0	0	0
0	1	1
1	0	1
1	1	0

सर्किट उत्तेजना सारणी -

येथे Q_3 Q_2 Q_1 Q_0 चार फ्लिप-फ्लॉपच्या वर्तमान स्थिती आहेत आणि Q_3^* Q_2^* Q_1^* Q_0^* पुढील 4 फ्लिप फ्लॉपच्या मोजणीच्या स्थिती आहेत. जर सध्याच्या स्थितीत संक्रमण असेल, म्हणजे Q_3 चे मूल्य 0 ते 1 किंवा 1 ते 0 पर्यंत बदलले तर संबंधित T(टॉगल) बिट 1 अन्यथा 0 असे लिहिलेले आहे.

स्टेप 3 :- स्टेट डायग्राम ड्रॉ करणे

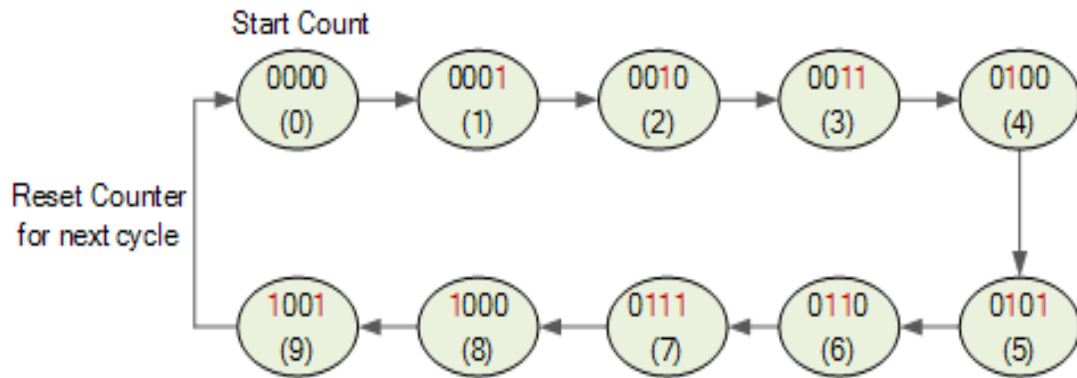


Figure 4.41: Decade Counter State Diagram

दशकाच्या काउंटरला मोड -10 किंवा 10 काउंटरने विभाजित करा असे म्हणतात. हे 0 ते 9 पर्यंत मोजले जाते आणि पुन्हा 0 वर रीसेट केले जाते. ते नैसर्गिक बायनरी अनुक्रमात मोजले जाते. येथे 4 T फ्लिप फ्लॉप वापरले जातात. ते $Q_3 Q_2 Q_1 Q_0 = 1001$ नंतर रीसेट होते.

Table 4.36: Truth table using T flipflop excitation table

Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0	T_3	T_2	T_1	T_0
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	1	0	0	0	1	1
0	0	1	0	0	0	1	1	0	0	0	1
0	0	1	1	0	1	0	0	0	1	1	1
0	1	0	0	0	1	0	1	0	0	0	1
0	1	0	1	0	1	1	0	0	0	1	1
0	1	1	0	0	1	1	1	0	0	0	1
0	1	1	1	1	0	0	0	1	1	1	1
1	0	0	0	1	0	0	1	0	0	0	1
1	0	0	1	0	0	0	0	1	0	0	1

स्टेप 4 : इनपुट व्हेरिएबल म्हणून फ्लिप-फ्लॉप आउटपुटच्या दृष्टीने प्रत्येक FF इनपुटसाठी कर्नाफ नकाशा तयार करा K Map सुलभ करा

$Q_3 Q_2$ \ $Q_1 Q_0$		00	01	11	10
		00	01	11	10
00	00	0	0	0	0
01	01	0	0	1	0
11	11	X	X	X	X
10	10	0	1	X	X

$$T_3 = Q_3 Q_0 + Q_2 Q_1 Q_0$$

$Q_3 Q_2$ \ $Q_1 Q_0$		00	01	11	10
		00	01	11	10
00	00	0	0	1	0
01	01	0	0	1	0
11	11	X	X	X	X
10	10	0	0	X	X

$$T_2 = Q_1 Q_0$$

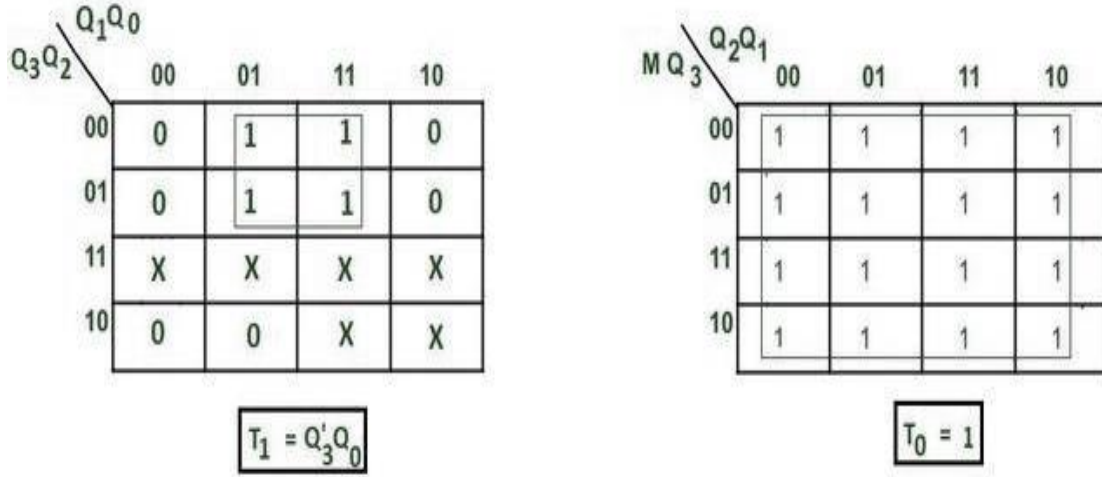


Fig 4.42: K map to find the minimum expression.

स्टेप 5 : सर्किट डायग्राम तयार करा येथे नकारात्मक एज ट्रिगर केलेले क्लॉक पल्स टॉगल करण्याच्या उद्देशाने वापरले जाते.

- क्लॉक पल्स प्रत्येक फ्लिप फ्लॉपला वेळेच्या त्याच क्षणी प्रदान केले जाते.
- **K** नकाशाच्या सरलीकृत समीकरणानुसार प्रत्येक फ्लिप फ्लॉपला टॉगल(**T**) इनपुट प्रदान केले जाते.

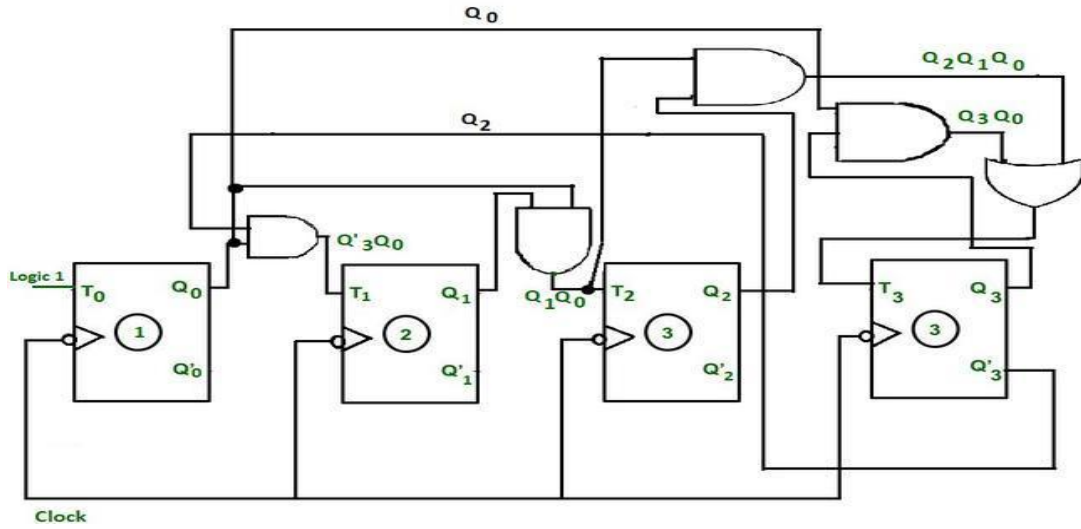


Fig 4.43: MOD-10 Counter Circuit Diagram

टाइमिंग डायग्राम: येथे टॉगलिंग वापरले जाते. जेव्हा **FF** चे टॉगल इनपुट(**T**) 1 असेल तेव्हाच **FF** ची स्थिती बदलेल.

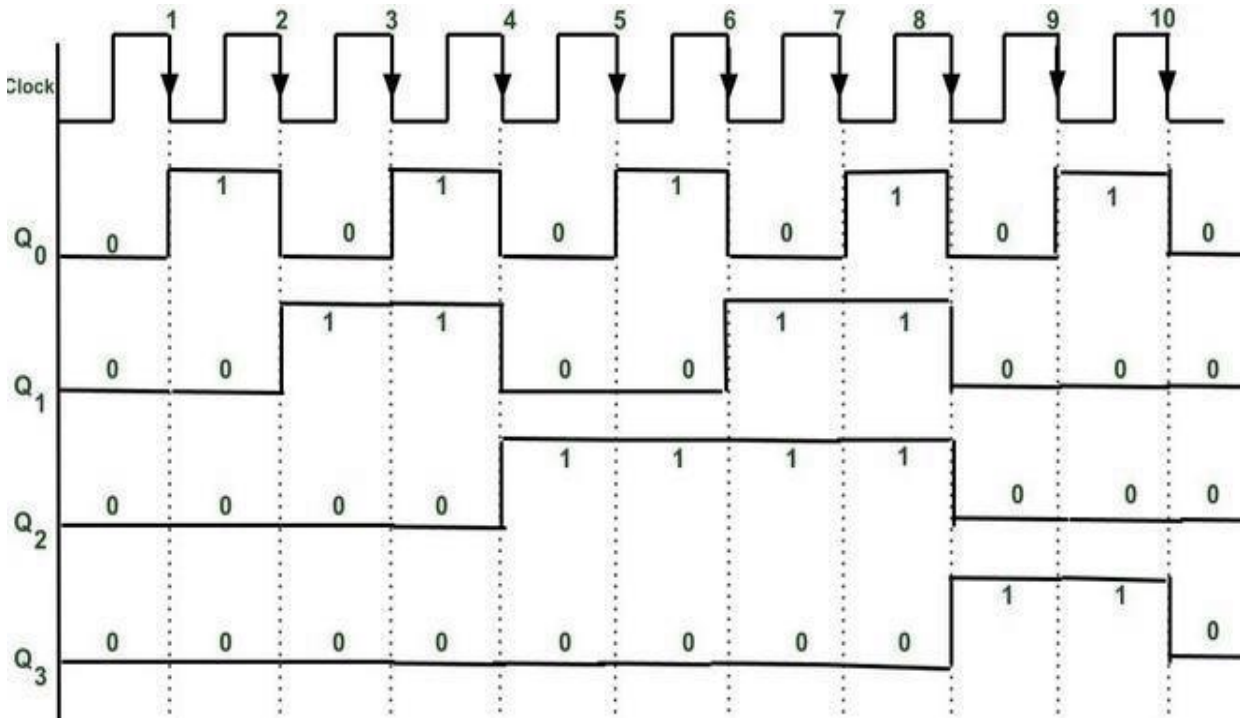


Fig 4.44: Synchronous Decade Counter Timing Diagram

स्पष्टीकरण:

- सुरुवातीला Q3 Q2 Q1 Q0 0 0 0 0 आहेत.
- काउंटरचा क्रम वेळेच्या आकृतीवरून सत्यापित केला जाऊ शकतो. घड्याळाच्या आउटपुटच्या प्रत्येक घसरत्या काठावर Q0 टॉगल होतो कारण T0 लॉजिक 1 शी जोडलेला आहे.
- T1 तेव्हाच 1 होतो जेव्हा अभिव्यक्ती $T1 = Q_3 Q_0$ 1 होते तसेच घड्याळाची धार पडल्यास (कारण नकारात्मक एज ट्रिगर होत असेल) तर T1 ची आउटपुट स्थिती म्हणजेच Q1 बदलेल.
- T2 फक्त 1 होते जेव्हा अभिव्यक्ती $T2 = Q_1 Q_0$ 1 होते तसेच घड्याळाची धार पडल्यास आउटपुट स्थिती Q2 बदलेल.
- T3 फक्त 1 होते जेव्हा अभिव्यक्ती $T3 = Q_3 Q_0 + Q_2 Q_1 Q_0$ परिणामी 1 होते तसेच घड्याळाची घसरण एज आली (कारण नकारात्मक एज ट्रिगर होत आहे) तर Q3 ची स्थिती बदलेल.
- आम्हाला Q3(MSB) Q2 Q1 Q0(LSB) असे आउटपुट मिळते.
- 10 व्या घसरणीनंतर सर्व FF ची आउटपुट स्थिती पुन्हा 0 0 0 0 होते.

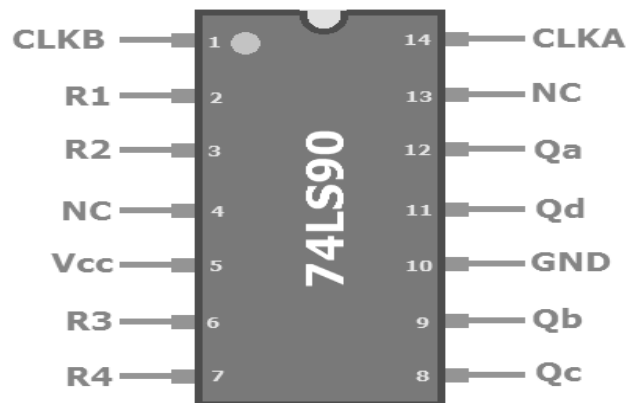
दशक काउंटर (Decade counter) IC 7490:

Fig 4.45 : Pin Diagram of Counter IC 74LS90

74LS90 बायनरी काउंटर वैशिष्ट्ये

1. IC साठी पॉवर इनपुट श्रेणी 4.75 watt ते 5.25 watt आहे.
2. IC साठी ऑपरेटिंग तापमान श्रेणी 0 ते 70°C आहे.
3. उच्च स्थितीत IC साठी इनपुट व्होल्टेज श्रेणी किमान 2.0V आहे आणि निम्न स्थिती 0.7V कमाल आहे.
4. IC उच्च स्थितीत आउटपुट करंट काढतो -0.4mA आणि निम्न स्थितीसाठी 8.0mA आहे
5. अंतर्गत क्लॅम्प डायोड संरक्षण श्रेणी -1.5V आहे.
6. हे 0 - 9 पासून एक साथे काउंटर म्हणून वापरले जाते.
7. IC मध्ये 0 पासून सुरू होण्याची आणि 9 ला आपोआप समाप्त होण्याची क्षमता आहे.
8. टीटीएल आउटपुटमुळे IC कोणत्याही TTL आधारित उपकरण आणि मायक्रोकंट्रोलरचा वापर केला जाऊ शकतो.
9. IC चा वीज वापर कमी आहे.
10. IC एकाधिक पॅकेजेसमध्ये येतो, PDSO, PDIP आणि GDIP
11. IC ला क्लॅम्प व्होल्टेजपासून अंतर्गत संरक्षण असते.

7490 काउंटर चे कार्य

या विभागात 7490 चे कार्य clear केले जाईल. IC मध्ये 4 फ्लिप फ्लॉपची अंतर्गत रचना आहे आणि पहिला फ्लिप फ्लॉप MOD-2 म्हणून वापरला जातो आणि इतर तीन MOD 5 म्हणून वापरला जातो. दोन क्लॉक पिन आहेत आणि ते आउटपुट स्थिती बदलण्यासाठी वापरले जातात. रीसेट पिन आणि गेटद्वारे नियंत्रण आहेत.

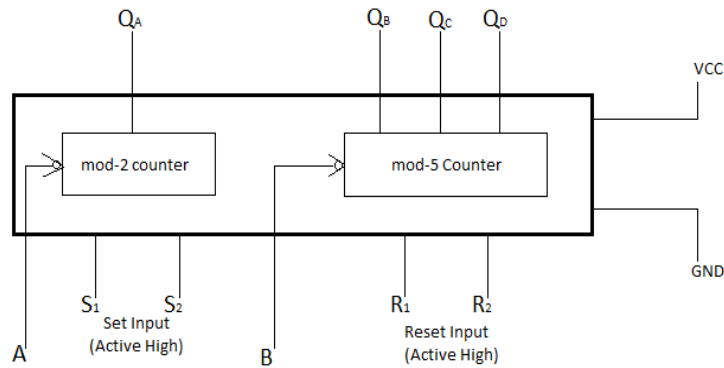


Fig 4.46: Decade Counter Block Diagram

7490 दशक काउंटरचे उपयोग

1. IC चा वापर 0 - 9 पासून साध्या बायनरी मोजणीसाठी केला जाऊ शकतो.
2. IC 7-सेगमेंटसह डिजिटल उपकरणांमध्ये येतो.
3. IC 74LS90 सर्किट, नेटवर्किंग आणि डिजिटल प्रणालींमध्ये देखील वापरतात.

सिंक्रोनस काउंटरचे फायदे/तोटे

1. हे सर्किट डिझाइन करणे खूप सोपे आहे कारण आम्ही सर्व गेट्ससाठी समान क्लॉक प्लस सेट करू शकतो.
2. चुकीच्या स्थितीत संपण्याची शक्यता कमी असते.
3. ते जलद आहेत कारण एसिंक्रोनस काउंटरच्या तुलनेत प्रसार विलंब लहान आहे.
4. असिंक्रोनस काउंटरच्या तुलनेत मोजणी त्रुटी नसतात.
5. कार्यप्रदर्शन अधिक चांगले असते. आणि सर्किट पोर्टेबल असते.

4.5.9 काउंटरचे अनुप्रयोग

1. अलार्म क्लॉक पल्स , एसी टायमर सेट करा, छायाचित्र काढण्यासाठी कॅमेरामध्ये वेळ सेट करा, ऑटोमोबाईलमधील फ्लॅशिंग लाइट इंडिकेटर, कार पार्किंग नियंत्रण इ.
2. शेड्यूलरद्वारे विशेष प्रक्रिया किंवा कार्यक्रमासाठी दिलेला वेळ मोजणे.
3. UP/DOWN काउंटर स्वयं-रिव्हर्सिंग काउंटर म्हणून वापरले जाऊ शकते.
4. हे क्लॉक डिवायडर सर्किट म्हणून देखील वापरले जाते.
5. समांतर लोड वैशिष्ट्य काही प्रारंभिक मोजणीसाठी काउंटर प्रीसेट करण्यासाठी वापरले जाऊ शकते.
6. वॉशिंग मशीन, मायक्रोवेव्ह स्वतःचे, टाइम शेड्यूल एलईडी इंडिकेटर, की बोर्ड कंट्रोलर इत्यादी सारख्या घरगुती उपकरणांमध्ये वापरलेले कॉमन.
7. छायाचित्र घेण्यासाठी कॅमेरामध्ये वेळ सेट करणे. ते मशीन मूव्हिंग कंट्रोलमध्ये देखील वापरले जातात.
8. मुख्यतः डिजिटल घड्याळे आणि मल्टिप्लेक्सिंग सर्किटमध्ये वापरले जाते.
9. ते सॉ-टूथ वेव्हफॉर्म (स्टेअर केस व्होल्टेज) तयार करण्यासाठी वापरले जातात
10. हे डिजिटल ते ॲनालॉग कन्व्हर्टरमध्ये देखील वापरले जाते.
11. ऑटोमोबाईलमधील फ्लॅशिंग लाइट इंडिकेटर.
12. डिजिटल क्लॉक व मॉड- काउंटर n बिट काउंटरसाठी, मॉड-6 आणि मॉड-10 काउंटरसाठी वापरले जातात.
13. वॉशिंग मशीनमध्ये, सेट केलेला टाइम मोजतो.
14. स्टेडियम किंवा सभागृहात प्रवेश करणाऱ्या आणि सोडणाऱ्या लोकांच्या संख्येची गणना
15. पार्किंगमध्ये वाहने
16. डिजिटल इलेक्ट्रॉनिक सर्किट्समध्ये, काउंटर म्हणून वापरले जातात.
17. फ्रिकेन्सी काउंटर विभाजक.
18. कॅस्केडिंग काउंटर (Cascading counters)

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 हप्प्यातील आउटपुट वारंवारता, मॉड्यूलसने विभाजित केलेल्या इनपुट वारंवारते इतकी असते.
- 74LS90 IC काउंटरचे दशक काउंटर मिळविण्यासाठी कॅस्केडिंग आवश्यक आहे. मॉड-5 काउंटरसह एज मॉड-2 काउंटर कॅस्केड करून दशक काउंटर तयार होतो. मॉड-2 x मॉड-5 मॉड-10 जातो

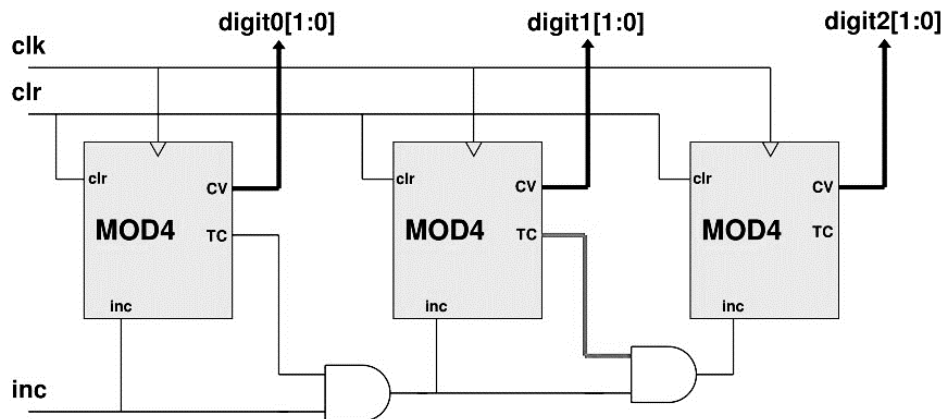


Fig 4.47: Cascading counter

स्वाध्याय

1. डिजिटल सर्किट्सचे प्रकार सूचीबद्ध करा.
2. डिजिटल सर्किट्समध्ये क्लॉकचे कार्य सांगा. .(कोणतेही ४)
3. डिजिटल सर्किट्समध्ये ट्रिगरिंगचे प्रकार सूचीबद्ध करा
4. फ्लिप फ्लॉपची व्याख्या लिहा नॉर गेट वापरलेल्या **R-S** फ्लिप फ्लॉपचे कार्य सांगा .
5. फ्लिप फ्लॉपची व्याख्या लिहा NAND गेट वापरून R-S फ्लिप फ्लॉपचे कार्य सांगा.
6. फ्लिप फ्लॉपची व्याख्या लिहा फ्लिप फ्लॉपचे प्रकार लिहा
7. क्लॉकड R-S फ्लिप फ्लॉपचे आकृती काढून कार्य सांगा.
8. R-S फ्लिप फ्लॉपचे तोटे लिहा आणि ते फ्लिप फ्लॉपमध्ये कसे नाहीसे केले जातात ते लिहा.
9. **J-K** फ्लिप फ्लॉपची लॉजिक आकृती काढून त्याची तृथ टेबल लिहा व कार्य सांगा.
10. **D** फ्लिप फ्लॉपची आकृती काढून त्याची तृथ टेबल लिहा.
11. रेस- अराऊंड कंडिशन कार्य सांगा.
12. एज टिगर्ड **D** फ्लिप फ्लॉपचे सविस्तर वर्णन करा व कार्य सांगा.
13. एज टिगर्ड **J-K** फ्लिप फ्लॉपचे कार्य सांगा.
14. मास्टर स्लेव्ह **J-K** फ्लिप फ्लॉपचे सविस्तर वर्णन करा.
15. लॉजिक आकृतीच्या साहाय्याने **T** फ्लिप फ्लॉपचे वर्णन करा
16. रजिस्टरची संकल्पना कार्य सांगा. व त्यांचे मूलभूत प्रकार सांगून त्यांच्या ब्लॉक डायग्राम काढा
17. लेफ्ट शिफ्ट रजिस्टरची लॉजिक आकृती व टायमिंग आकृती काढून कार्य लिहा.
18. राईट शिफ्ट रजिस्टरचे कार्य वर्णन करा
19. लॉजिक आकृतीच्या साहाय्याने **3** बीट रिपल काउंटर वर्णन करून टायमिंग आकृती काढा.
20. लॉजिक आकृतीच्या साहाय्याने डिकेड काउंटरचे कार्य सांगा.. **v** वेव्हफॉर्म काढा.
21. IC 7490 वापरून डिकेड काउंटरची आकृती काढून त्याची कार्य सांगा
22. सिंक्रोनस आणि एसिंक्रोनस काउंटरमधील फरक लिहा.(कोणतेही 4)
23. कॉम्बिनेशनल आणि सिंक्रोनस सर्किट्सचे व्याख्या आकृतीसह स्पष्टी करा.
24. कॉम्बिनेशनल आणि सिंक्रोनस सर्किट्समधील फरक करा.
25. ट्रॉफिक सिग्नल च्या वेगवेगळ्या स्टेट्स ची नोंद करा आणि त्यात वापरलेले सीक्वेंसीअल सर्किट ची यादी करा.
26. . रेल्वे स्टेशन वर वापरलेले विविध डिजिटल सर्किटचे अनुप्रयोग लिहा.

लघु प्रकल्प करणे (Micro project suggestions)

1. 1:4 /1:2 DE-MUX वापरून 1:8 DEMUX लागू करा.
2. 4 बिट अँडर लागू करण्यासाठी एक सर्किट तयार करा.
3. 4बिट पॅरिटी जनरेटर आणि पॅरिटी टेस्टर तयार करा.
4. 8:1/4:1 MUX वापरून 16:1 MUX लागू करा.
5. 7 बिट सेगमेंट डिस्प्लेची चाचणी घेण्यासाठी एक सर्किट तयार करा.
6. एक LED डिस्प्ले बार तयार करा. बर्गलर अलार्मवर प्रकल्प विकसित करा.
7. NAND गेट वापरून लाईट डिटेक्टर सर्किट.
8. IC 7490 वापरून दशक काउंटर तयार करा.
9. J-K फ्लिप फ्लॉपचे वापरून रिपल काउंटर तयार करा.

10. आमच्या घरात विविध उपकरणे असलेले डिजिटल सर्किट ओळखा. जसे वॉशिंग मशीन, रेफ्रिजरेटर, मायक्रोवेव्ह ओव्हन, टेलिव्हिजन, व्हिडिओ डोअरबेल युनिट.
11. रोड सिग्नलला भेट द्या आणि वेगवेगळ्या **States** च्या कामकाजाचे निरीक्षण करा आणि रिपोर्ट बनवा.

संदर्भ (References):

Sr. No	Author	Title	Publisher with ISBN Number
1	Jain R.P	Modern Digital Electronics	McGraw-Hill Publishing, New Delhi, 2006 ISBN: 9780070669116
2	Salivahanan S, Arivazhagan S.	Digital Circuits and Design	Vikas Publishing House, New Delhi, 2011 ISBN: 9789325960411
3	G. K. Kharate	Digital Electronics	Oxford ISBN: 9780198061830

संकेतस्थल (Websites):

- <https://studytronics.weebly.com/digital-electronics.html>
- <https://www.udemy.com/course/basics-of-digital-techniques>
- <https://www.geeksforgeeks.org/synchronous-sequential-circuits-in-digital-logic>
- https://onlinecourses.nptel.ac.in/noc19_ee51/preview
- <https://de-iitr.vlabs.ac.in/>

युनिट 5: डेटा कन्वर्टर आणि मेमरीज (Data Converters and Memories)

विषय निष्पत्ती (Course Outcome):

डिजिटल इलेक्ट्रॉनिक सिस्टीममधील डेटा कन्वर्टर आणि मेमरीच्या कार्याचा अर्थ लावणे.

Interpret the functions of data converters and memories in digital electronic systems

युनिट निष्पत्ती (Unit Outcome) :

5.1 दिलेल्या DAC च्या कार्याचे वर्णन करणे.

Describe the working of the given type of DAC.

5.2 निर्दिष्ट DAC साठी दिलेल्या डिजिटल इनपुटसाठी आउटपुट विद्युत दाब (voltage) ची गणना करणे.

Calculate the output voltage for the given digital input for specified DAC.

5.3 दिलेल्या ADC च्या कार्याचे वर्णन करणे.

Describe the working of the given type of ADC.

5.4 ROM, EPROM, EEPROM आणि Flash Memory च्या कामाची तुलना करणे.

Compare working of ROM, EPROM, EEPROM and Flash Memory.

परिचय:

वास्तविकता जगातील सर्व संदेश हे मुळतः अॅनालॉग स्वरूपात आहेत. आपण या प्रमाणांचे (quantities) विद्युतीय (electrically) रीतीने अॅनालॉग सिग्नल म्हणून प्रतिनिधित्व (represent) करू शकतो. अॅनालॉग सिग्नल हा वेळ बदलणारा सिग्नल आहे ज्यामध्ये दिलेल्या टाइम स्लॉटसाठी कितीही (any number) मूल्ये (भिन्नता) (values (variations)) असतात. याउलट, डिजिटल सिग्नल एका स्तरावरून दुसऱ्या स्तरावर अचानक बदलतो आणि दिलेल्या वेळेच्या स्लॉटसाठी केवळ मर्यादित संख्येची मूल्ये (भिन्नता) (values (variations)) असतील.

5.1 डेटा कन्वर्टरची आवश्यकता:

काहीवेळा आपल्याला अॅनालॉग सिग्नलसना त्यांच्या डिजिटल समतुल्य स्वरूपात आणि डिजिटल सिग्नलसना अॅनालॉग स्वरूपात रूपांतरित करावे लागते. अशा रूपांतरणासाठी वापरल्या जाणाऱ्या इलेक्ट्रॉनिक्स सर्किट्सना डेटा कन्वर्टर म्हणतात. डिजिटल प्रणालींद्वारे मिळणाऱ्या फायद्यांमुळे, ते इन्स्ट्रुमेंटेशन, संगणक, संप्रेषण आणि नियंत्रण (instrumentation, computers, communication and control) यासारख्या अनेक क्षेत्रांमध्ये मोठ्या प्रमाणावर वापरले जातात. अशा अनेक प्रयोगांमध्ये सिग्नल डिजिटल स्वरूपात उपलब्ध नाहीत. बहुतेक भौतिक प्रमाण (physical quantities) जसे की तापमान, दाब, विस्थापन, कंपन (temperature, pressure, displacement, vibrations) .इ. अॅनालॉग स्वरूपात उपलब्ध आहेत हे प्रमाण अॅनालॉग स्वरूपात अचूकपणे प्रस्तुत (represented) केले जातात परंतु अॅनालॉग सिग्नलवर प्रक्रिया करणे, संचयित करणे, प्रसारित (transmit) करणे कठीण आहे कारण त्रुटी (error) सहजपणे ओळखल्या (introduced) जातात. त्यामुळे या चुका कमी करण्यासाठी हे भौतिक प्रमाण (physical quantities) डिजिटल स्वरूपात व्यक्त करणे चांगले.

डिजिटल सिग्नलचे स्टोरेज आणि प्रक्रिया हे जलद, कार्यक्षम आणि सोपे आहे. सर्व रियल वर्ल्ड ॲप्लिकेशन डिजिटल पद्धतीने स्टोअर केले जातात. त्यामुळे अॅनालॉग ते डिजिटल आणि डिजिटल ते अॅनालॉग मध्ये रूपांतरित करावे लागते.

डेटा कन्वर्टर हे एक इलेक्ट्रॉनिक सर्किट आहे जे एका स्वरूपातून दुसऱ्या स्वरूपामध्ये रूपांतरित करते.

डेटा कन्वर्टरचे उपयोग :

- ट्रान्सड्यूसर (Transducer) सह एकत्र वापरले जातात.
- संगणकात (Computer)मध्ये अॅनालॉग सिग्नलला डिजिटल सिग्नलमध्ये रूपांतरित करण्यासाठी
- मायक्रोकंट्रोलरसह वापरले जाते .
- डिजिटल सिग्नल प्रक्रियेत
- डिजिटल स्टोरेज ओसिलोस्कोपमध्ये वापरले जाते .
- सेल फोन मध्ये वापरले जाते .
- सायंटिफीक इंस्ट्रुमेंट्स मध्ये वापरले जाते

5.1.1 डेटा कन्वर्टरचे प्रकार (Types of Data converters)

- डिजिटल ते अॅनालॉग कनव्हर्टर(DAC)
- अॅनालॉग ते डिजिटल कनव्हर्टर(ADC)

डिजिटल ते अॅनालॉग कन्व्हर्टर(DAC):

- डिजिटल ते अॅनालॉग कन्व्हर्टर, डिजिटल किंवा बायनरी डेटाला त्याच्या समतुल्य अॅनालॉग मूल्यामध्ये रूपांतरित करते.
- डिजिटल ते अॅनालॉग कन्व्हर्टरसाठी इनपुट डिजिटल डेटा n -bit बायनरी शब्द D आहे.
- बिट b_1 ला सर्वात लक्षणीय बिट (MSB) आणि बिट b_n सर्वात कमी लक्षणीय बिट (LSB) म्हणतात. मग प्रमाण D ने दर्शविले जाऊ शकते.
- $D = b_1 2^{-1} + b_2 2^{-2} + b_3 2^{-3} + \dots + b_n 2^{-n}$

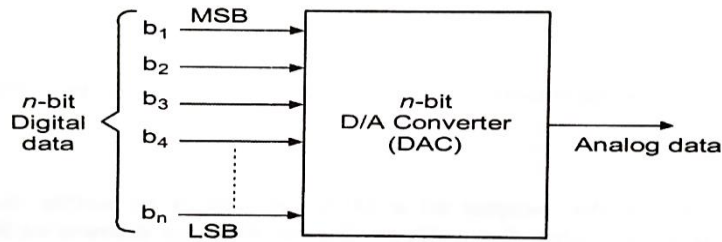


Fig 5.1: N-Bit Digital to

Converter

Analog

- डिजिटल टू अॅनालॉग कन्व्हर्टरसाठी, विद्युत दाब आउटपुट ,एक्स्प्रेसन मध्ये दिल्या प्रमाणे आहे .
- $V_o = K V_Fs = K V_Fs (b_1 2^{-1} + b_2 2^{-2} + b_3 2^{-3} + \dots + b_n 2^{-n})$

 V_o - आउटपुट व्होल्टेज **V_Fs** - व्होल्टेजची पूर्ण स्केल श्रेणी **K** - स्केलिंग घटक, सहसा एकता (Unity) **$b_1 \dots b_n$** = n -बिट बायनरी प्रॅक्शनल शब्द बायनरी पॉइंटसह डावीकडे स्थित आहे. **b_1** - सर्वात लक्षणीय बिट (MSB) म्हणतात. **b_n** - किमान लक्षणीय बिट (LSB)**5.1.2 डिजिटल ते अॅनालॉग कनव्हर्टर(DAC) चे प्रकार:**

- वेटेड रेजिस्टर DAC (Weighted Resistor D /A Converter)
- $R-2R$ लॅडर DAC ($R-2R$ Ladder D /A Converter)

a. वेटेड रेजिस्टर DAC (Weighted Resistor D/A Converter):

- रेजिस्टन्स व्हॅल्यूज बायनरी वेट्सनुसार वेटेड केले जातात, म्हणून सर्किटला वेटेड रेजिस्टर डी/ए नेटवर्क म्हणतात. हे बायनरी वेटेड रेजिस्टर आणि ऑपरेशनल अॅम्प्लिफायर (operational amplifier)/Op-Amp चे नेटवर्क वापरते. Op-Amp हा, बेरीज करणारा (अॅडर), विद्युत प्रवाह ते विद्युत दाब रूपांतर (current to voltage converter) करण्यासाठी कार्यान्वित (operate) केला जातो.

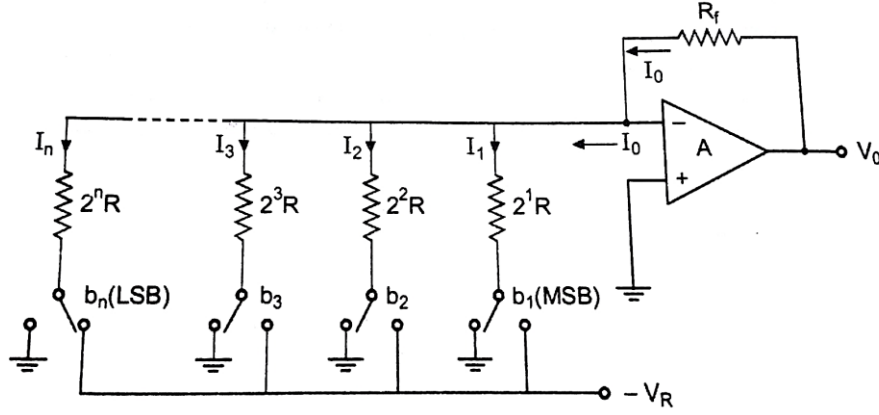


Fig 5.2: N-bit weighted resistor DAC

- प्रतिरोधक $2^1 R = 2R$, $2^2 R = 4R$, $2^3 R = 8R$, ..., $2^n R$ च्या नेटवर्कमधील बायनरी वेटेड प्रतिरोधक आहेत. संदर्भ व्होल्टेज ($-V_R$) डिजिटली नियंत्रित स्विचेस $S_1, S_2, \dots, S_n$.
- विविध स्विचेसच्या स्थानांवर अवलंबून, करंट $I_1, I_2, \dots, I_n$ अनुक्रमे वेटेड प्रतिरोधक $2R, 4R, \dots, 2^n R$ मधून वाहतो.
- n बिट डिजिटल शब्दात b_1 MSB आणि b_n ला LSB आहे. R_f हा फीडबॅक रेजिस्टर आहे. आणि I_0 हा आउटपुट करंट आहे. I_0 मधून वाहणाऱ्या वैयक्तिक प्रवाहांची बेरीज म्हणून व्यक्त केले जाऊ शकते वेटेड प्रतिरोधक.
- $I_0 = I_1 + I_2 + I_3 + \dots + I_n$
- $I_0 = V_R / 2R b_1 + V_R / 4R b_2 + \dots + V_R / 2^n R b_n$
लक्षात घ्या की $b_1, b_2, \dots, b_n$ हे बायनरी अंक आहेत आणि त्यांचे मूल्य '0' किंवा '1' असू शकतो.
- $I_0 = V_R / R [b_1 2^{-1} + b_2 2^{-2} + \dots + b_n 2^{-n}]$
आउटपुट व्होल्टेज V_0 द्वारे दिले जाते
- $V_0 = I_0 R_f$
 $= V_R \cdot R_f / R [b_1 2^{-1} + b_2 2^{-2} + \dots + b_n 2^{-n}]$
जर $R_f = R$,
- $V_0 = V_R [b_1 2^{-1} + b_2 2^{-2} + \dots + b_n 2^{-n}]$
- नकारात्मक संदर्भ व्होल्टेज ($-V_R$) चा वापर पॉजिटिव्ह अॅनालॉग आउटपुट देते.

वेटेड रेजिस्टर DAC चे फायदे:

- साधी परिक्रमा (simple Circuitry)
- सोपी गणना (Easy Calculations)

वेटेड रेजिस्टर DAC तोटे:

- डी/ए कन्व्हर्टरची अचूकता आणि स्थिरता प्रतिरोधकांच्या अचूकतेवर आणि वापरलेल्या तापमान अवलंबनावर आधारित आहे.
- प्रतिरोधकांना बिट मूल्यांवर आधारित भिन्न प्रवाह हाताळावे लागतात.

- c. त्यात आवश्यक अचूकतेसह प्रतिरोधक मूल्यांची विस्तृत श्रेणी असण्याची आणि तापमानाच्या विस्तृत श्रेणीवर कोणते ट्रॅक ठेवण्याची समस्या आहे.
- d. मोनोलिथिक IC मध्ये रेझिस्टर व्हॅल्यूजची एवढी विस्तृत श्रेणी तयार करणे कठीण आहे.
- e. ही अडचण R-2R लॅडर नेटवर्क डी/ए कन्व्हर्टरद्वारे दूर केली जाते.

उदाहरणे :

1. 4 बिट DAC च्या **1101** डिजिटल इनपुटसाठी एनालॉग आउटपुटची गणना करणे. $V_{FS}=5V$ गृहीत धरा.

$$V_o = KV_{FS} = KV_{FS} (b_1 2^{-1} + b_2 2^{-2} + b_3 2^{-3} + \dots + b_n 2^{-n})$$

$$\begin{aligned} V_o &= 5[(1 \times 2^{-1}) + (1 \times 2^{-2}) + (0 \times 2^{-3}) + (1 \times 2^{-4})] \\ &= 5 [0.5 + 0.25 + 0 + 0.0625] \end{aligned}$$

$$V_o = 4.0625 V$$

२. 4 बिट DAC च्या **1011** डिजिटल इनपुटसाठी एनालॉग आउटपुटची गणना करणे. $V_{FS}=5V$ गृहीत धरा.

$$V_o = KV_{FS} = KV_{FS} (b_1 2^{-1} + b_2 2^{-2} + b_3 2^{-3} + \dots + b_n 2^{-n})$$

$$\begin{aligned} V_o &= 5[(1 \times 2^{-1}) + (0 \times 2^{-2}) + (1 \times 2^{-3}) + (1 \times 2^{-4})] \\ &= 5 [0.5 + 0 + 0.125 + 0.0625] \end{aligned}$$

$$V_o = 3.4375 V$$

b. R-2R लॅडर DAC (R-2R Ladder D /A Converter):

- R-2R लॅडर D /A कन्व्हर्टरमध्ये, फक्त दोन मूल्यांचे प्रतिरोधक म्हणजे R आणि 2R वापरले जातात. म्हणून ते IC फॅब्रिकेशनसाठी योग्य आहे. वापरलेली R ची विशिष्ट मूल्ये 2.5 K Ω ते 10 K Ω पर्यंत बदलतात. 4 बिट D/A रूपांतरणासाठी लॅडर प्रकारच्या नेटवर्कच्या ऑपरेशनचे सिद्धांत असे आहे.
- आकृती 5.3 मध्ये दाखवल्याप्रमाणे ,4 बिट बायनरी इनपुट $b_1b_2b_3b_4$, एनालॉग आउटपुट V_o , टर्मिनेटिंग रेझिस्टर, 2R.

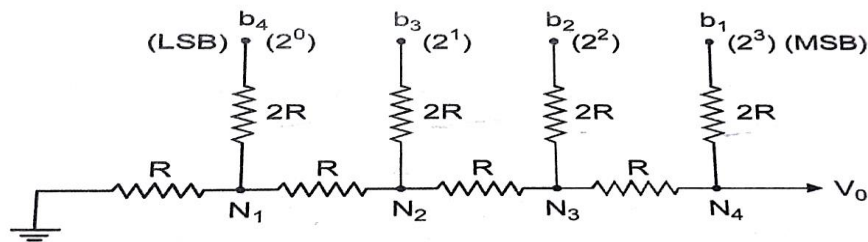
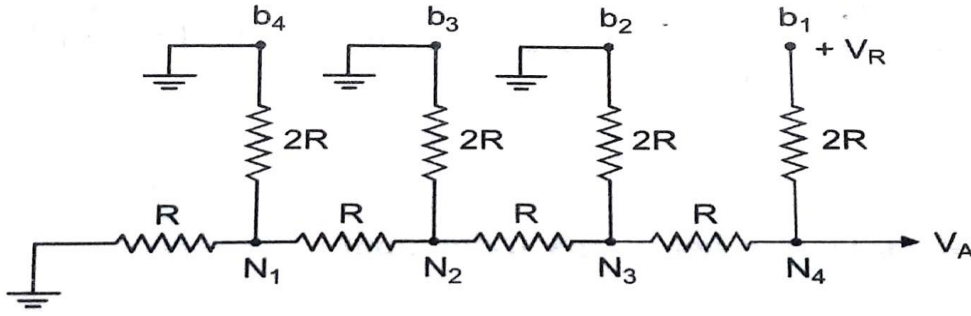


Fig 5.3: 4 Bit R-2R Ladder Type D/A Converter

- या लॅडर सर्किटमध्ये, आउटपुट व्होल्टेज ही डिजिटल इनपुटची वेटेड बेरीज आहे.
- उदाहरणार्थ, जर 4 बिट बायनरी इनपुट $b_1b_2b_3b_4 = 1000$ असेल म्हणजे MSB 1 असेल तर इतर 3 इनपुट 0 आहेत.

Fig 5.4: Equivalent circuit for binary input $b_1b_2b_3b_4 = 1000$

- येथे टर्मिनेटिंग रेझिस्टर ($2R$) आणि b_4 इनपुट ($2R$) शी जोडलेले रेझिस्टर नोड N_1 वर एकत्र केले जातात ज्यामुळे समतुल्य रेझिस्टर (R) बनतो.

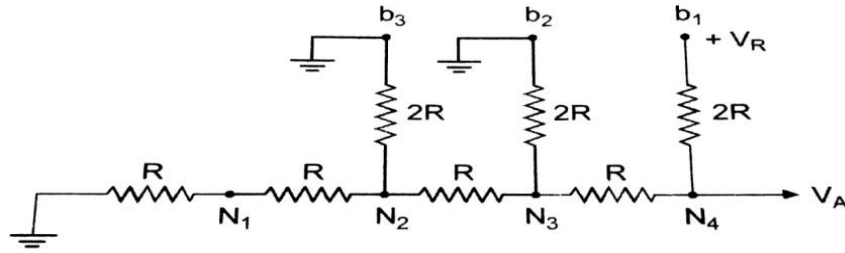


Fig 5.5: Equivalent circuits of the first stage

- नंतर नोड N_2 वर, b_3 इनपुट ($2R$) सह जोडलेले रेझिस्टर असू शकते.
- रेझिस्टर ($R+R=2R$) सह एकत्रित करून समतुल्यचा दुसरा टप्पा तयार होतो.

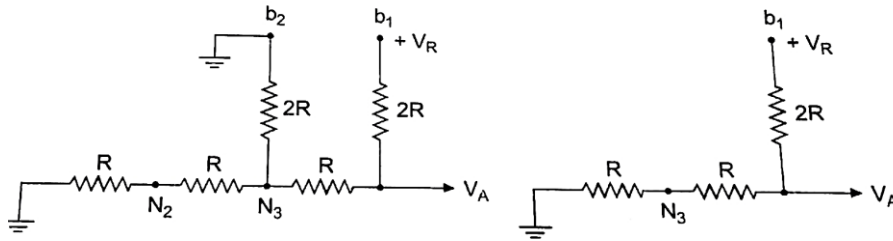


Fig 5.6: Second stage, third stage equivalent circuit

- त्याचप्रमाणे नोड N_3 वर, समतुल्य रोधक हा समतुल्य मध्ये दाखवल्याप्रमाणे R आहे
- नंतर V_0 द्वारे ऑनलॉग आउटपुट व्होल्टेज दिले जाते द्वारे दिले जाते

$$V_o = \frac{V_R \times 2R}{R + R + 2R} = \frac{V_R}{2}$$

- अशा प्रकारे डिजिटल इनपुट 1000 साठी, आउटपुट $V_R/2$ आहे, 0100 इनपुटसाठी, आउटपुट $V_R/4$, 0010 आहे, आउटपुट $V_R/8$ आहे आणि 0001 हे $V_R/16$ आहे. रेझिस्टिव्ह लॅडर हे रेखीय नेटवर्क असल्याने, n बिटसाठी आउटपुट व्होल्टेज D/A कन्व्हर्टर खालीलप्रमाणे आहे.

$$V_o = \frac{V_R}{2^1} + \frac{V_R}{2^2} + \frac{V_R}{2^3} + \dots + \frac{V_R}{2^n}$$

4 बिट R-2R लॅडर D/A नेटवर्क (4 bit R-2R Ladder D/A Network):

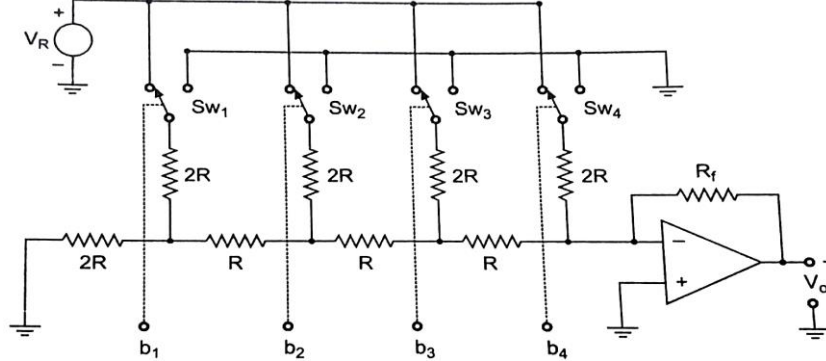


Fig 5.7: 4 Bit R-2R Ladder D/A Network

$$V_o = V_R \frac{R_f}{R} \left(\frac{b_1}{2^1} + \frac{b_2}{2^2} + \frac{b_3}{2^3} + \frac{b_4}{2^4} \right)$$

$$= V_R \frac{R_f}{R \times 2^4} (b_1 2^3 + b_2 2^2 + b_3 2^1 + b_4 2^0)$$

'n' बिट इनपुट सिग्नलसाठी, $R_f = R$ गृहीत धरून; R_f - op-amp चे फीडबॅक रेजिस्टर.

$$V_o = \frac{V_R}{2^n} (b_1 2^{n-1} + b_2 2^{n-2} + \dots + b_n 2^0)$$

5.1.3 DAC 0808 चे पिन कॉन्फिगरेशन: (Pin Configuration)

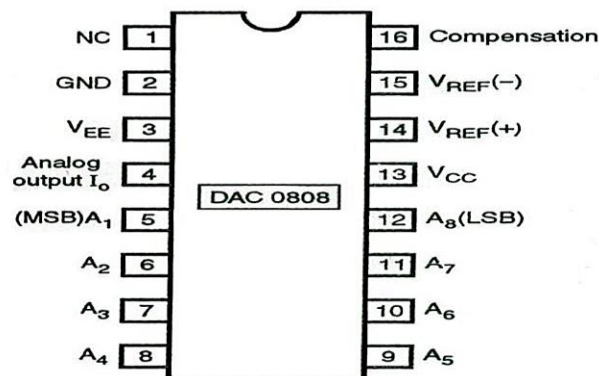


Fig 5.8: DAC 0808 pin configuration

5.1.4 DAC 0808 चे तपशील (Specifications/selection of DAC 0808)

- जलद सेटलिंग वेळ: 150 (nsec) नॅनो सेकंद.
- वीज पुरवठा व्होल्टेज श्रेणी: $\pm 5v$
- कमी उर्जा वापर: 33 mw.

- d. उच्च गती गुणाकार इनपुट स्ल्यूरेट : $8\text{mA} / \mu\text{sec}$ (मायक्रो सेकंद.)
 e. TTL, आणि CMOS लॉजिक लेव्हलससह थेट इंटरफेस

5.2 . अॅनालॉग ते डिजिटल कनव्हर्टर (Analog to Digital Data Converter)/(ADC):

ADC कनव्हर्टर अॅनालॉग सिग्नलला त्याच्या समतुल्य n बिट बायनरी कोडेड डिजिटल आउटपुट सिग्नलमध्ये रूपांतरित करतो. अॅनालॉग इनपुटचे नमुने इनपुट सिग्नलच्या कमाल वारंवारता घटकापेक्षा जास्त वारंवारतेवर केले जातात. ADC मधील डिजिटल आउटपुट सिरियल किंवा प्यारलल स्वरूपात असू शकते.

ADC अॅनालॉग इनपुट V_i स्वीकारतो आणि आउटपुट $b_1, b_2, \dots, b_n$ फ्रॅक्शनल व्हॅल्यू D तयार करतो.

- $D = b_1 2^{-1} + b_2 2^{-2} + b_3 2^{-3} + \dots + b_n 2^{-n}$,
 b_1 हा LSB आहे आणि b_n हा MSB आहे.

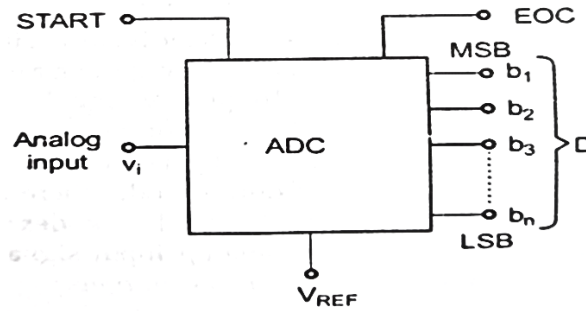


Fig 5.9: ADC 0808 Symbolic Presentation

ADC सह दोन अतिरिक्त कंट्रोल पिन स्टार्ट (START) इनपुट आणि एंड ऑफ रूपांतरण (EOC- End of Conversion) आउटपुट प्रदान केले आहेत. START इनपुट रूपांतरण (conversion) सुरू करतो आणि रूपांतरण पूर्ण झाल्यावर EOC घोषणा (Announces) करतो.

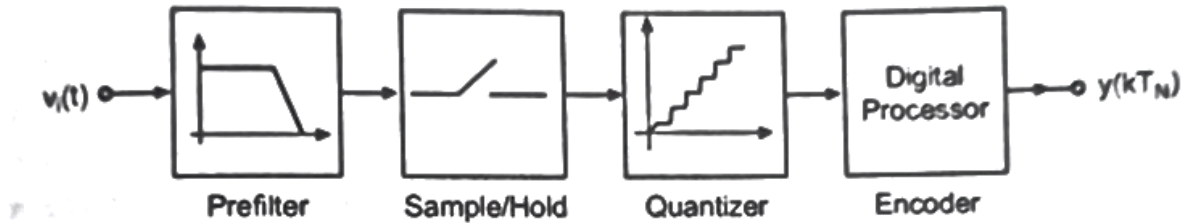


Fig 5.10: Block diagram of ADC

ADC चे ब्लॉक आकृती (block diagram) आकृती 5.10 मध्ये दर्शविले आहे. यात प्रीफिल्टर, सॅम्पल आणि होल्ड अॅम्प्लिफायर, क्वांटायझर आणि एन्कोडर असतात.

प्रीफिल्टर उच्च फ्रिक्वेंसी सिग्नलचे अलईयसिंग (aliasing) टाळते. सॅम्पल आणि होल्ड सर्किट ADC मध्ये इनपुट अॅनालॉग सिग्नल रूपांतरणाच्या वेळी स्थिर मूल्यावर ठेवते. क्वांटायझर संदर्भ व्होल्टेज सिग्नलला सबरेंजमध्ये विभागतो.

सामान्यतः, n बिट डिजिटल आउटपुट कोडसाठी, 2^n सबरेंज असतात. डिजिटल प्रोसेसर एन्कोडर सर्किट बनवतो जो सबरेंजला संबंधित डिजिटल बिट्समध्ये एन्कोड (Encode) करतो .

5.2.1 ADC चे प्रकार:

- ड्युअल स्लोप अॅनालॉग ते डिजिटल कनव्हर्टर (Dual slope A/D Converter)
- सकसेसीव एप्रोक्सीमेशन अॅनालॉग ते डिजिटल कनव्हर्टर (Successive Approximation type A/ D Converter)

c. फ्लॅश टाइप अॅनालॉग ते डिजिटल कनव्हर्टर (Flash type A/D Converter)

a. ड्युअल स्लोप A/D Converter (Dual slope A/D Converter)

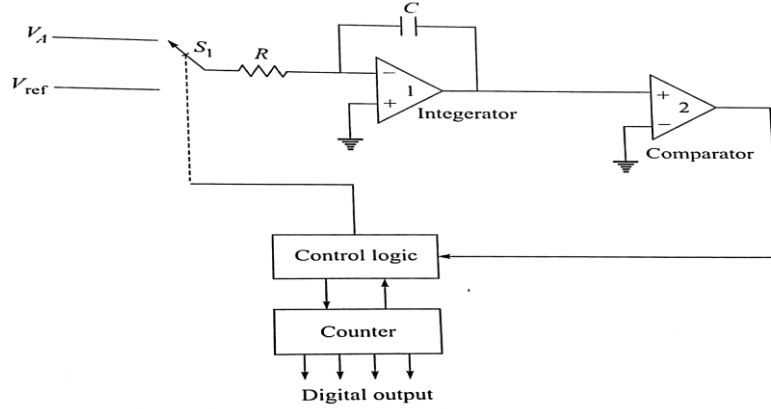


Fig 5.11: ड्युअल स्लोप A/D Converter

1. ड्युअल स्लोप A/D कनव्हर्टरमध्ये, इंटीग्रेटर दोन भिन्न रॅम्प तयार करतो- इनपुट म्हणून अज्ञात (Unknown Analog Input Voltage) अॅनालॉग इनपुट विद्युत दाब, दुसरे इनपुट म्हणून ज्ञात संदर्भ (Reference voltage V_{ref}) विद्युत दाब. म्हणून, त्याला ड्युअल स्लोप A/D कनव्हर्टर म्हणून ओळखले जाते.
2. ड्युअल स्लोप ए/डी कनव्हर्टरचे मूलभूत तत्त्व असे आहे की, अॅनालॉग सिग्नल आनुपातिक कालावधीत रूपांतरित केला जातो, जे नंतर डिजिटल काउंटर वापरून मोजले जाते.
3. यात समाविष्ट आहे इंटीग्रेटर, कंपॅरेटर, काउंटर, कंट्रोल लॉजिक आणि संदर्भ व्होल्टेज.
4. रेझिस्टर R सह ऑपरेशनल ऑप्लिफायर (Op-amp 1) आणि कॅपेसिटर लीनियरली (linearly) चार्ज केला जातो.
5. कॅपेसिटरवरील व्होल्टेज अॅनालॉग इनपुट व्होल्टेजचे मोठेपणा (Amplitude) आणि कालावधी T वर अवलंबून असतो.
6. ओपन लूपमध्ये (open loop) ऑपरेशनल ऑप्लिफायर (Op-amp 2) वापरला जातो. हे तुलनाकर्ता म्हणून कार्य करते आणि संदर्भ इनपुटसह इंटीग्रेटरच्या आउटपुटची तुलना करते. संदर्भ व्होल्टेज (reference voltage) शून्य आहे आणि तुलनाकर्ता (comparator) शून्य क्रॉसिंग डिटेक्टर म्हणून कार्य करतो.
7. कॅपेसिटरमधील व्होल्टेजच्या आधारावर (comparator) तुलनाकर्त्याचे आउटपुट उच्च किंवा कमी आहे.
8. जेव्हा कॅपेसिटरमधील व्होल्टेज शून्यावर पोहोचतात, तेव्हा तुलनाकर्त्याचे आउटपुट असते - V_{sat} (सॅचुरेशन व्होल्टेज), जे कंट्रोल लॉजिकला दिले जाते आणि कंट्रोल लॉजिक काउंटरची मोजणी थांबवते.

फायदे:

- a. हे सोपे आणि तुलनेने स्वस्त आहे.
- b. यात उच्च रूपांतरण अचूकता आहे. (High conversion accuracy)
- c. वेळ, तापमान आणि इनपुट व्होल्टेजचा परिणाम होत नाही.

तोटे:

- a. इतर कोणत्याही ADC च्या तुलनेत याच्या रूपांतरणाची वेळ जास्त आहे.
- b. याचा रूपांतरणाचा (conversion) वेग खूपच कमी आहे.

b. सकसेसीव एप्रोक्सीमेशन A/D Convertor

(Successive Approximation type A /D Converter)

- सकसेसीव एप्रोक्सीमेशन A/D कनवर्टरमध्ये रूपांतरण वेळ स्थिर ठेवली जाते, आणि ते डिजिटल आउटपुटमधील बिट्सच्या संख्येच्या प्रमाणात आहे.
- या A/D कनवर्टरचे मूळ तत्त्व असे आहे की अज्ञात (unknown) अॅनालॉग इनपुट व्होल्टेज हे MSB पासून सुरुवात करून एका वेळी थोडा प्रयत्न करून एन-बिट डिजिटल मूल्याच्या विरुद्ध अंदाजे केले जाते.
- हे सलगपणे विद्युत दाबाला अर्ध्याने विभाजित करून कार्य करते.
- 8 बिट सकसेसीव एप्रोक्सीमेशन A/D कनवर्टरसाठी, अॅनालॉग इनपुट व्होल्टेजच्या मोठेपणाकडे दुर्लक्ष करून रूपांतरणासाठी फक्त 8 चक्रांची आवश्यकता आहे.

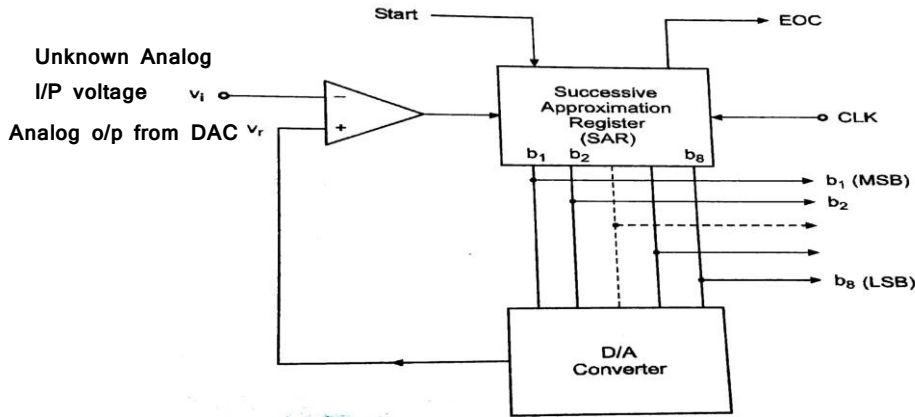


Fig 5.12: Successive Appropriation A/D Converter.

1. सर्किट सकसेसीव एप्रोक्सीमेशन रजिस्टर (SAR) वापरते जे चाचणी (Trial) आणि त्रुटी (Error) पद्धतीने प्रत्येक क्रमिक बिटचे आवश्यक मूल्य शोधते.
2. सकसेसीव एप्रोक्सीमेशन A /D Convertor चे आउटपुट एन-बिट डी/ए कनवर्टरला दिले जाते.
3. डी/ए कनवर्टरचे अॅनालॉग आउटपुट तुलनेच्या नॉन इनव्हर्टिंग इनपुटवर लागू केले जाते.
4. कम्पॅरेटरचे दुसरे इनपुट अज्ञात अॅनालॉग इनपुट व्होल्टेज V_i सह रूपांतरण अंतर्गत जोडलेले आहे.
5. कम्पॅरेटरचे आउटपुट SAR चे सीक्वेंशियल अंदाजे तर्क सक्रिय करण्यासाठी वापरले जाते.
6. जेव्हा स्टार्ट कमांड लागू केली जाते, तेव्हा SAR डिजिटल सिग्नलचे MSB सेट करते, तर इतर बिट्स शून्य केले जातात, जेणेकरून चाचणी कोड 1 नंतर शून्य होतो.
7. सकसेसीव एप्रोक्सीमेशन A /D Convertor चे आउटपुट अॅनालॉग समतुल्य V_r मध्ये रूपांतरित केले जाते आणि इनपुट सिग्नल V_i शी तुलना केली जाते.
8. जर $V_i > V_r$ असेल तर MSB 1 म्हणून ठेवला जातो आणि LSB 1 केला जातो आणि चाचणीची पुनरावृत्ती होते.
9. जर $V_i < V_r$ असेल तर मूल्य अचूक डिजिटल समतुल्यपेक्षा मोठे आहे.
10. म्हणून तुलनाकर्ता दुसरा MSB 0 वर रीसेट करतो आणि पुढील MSB वर जातो.
11. सर्व बिट पोझिशन्सची चाचणी होईपर्यंत ही प्रक्रिया पुनरावृत्ती केली जाते.
12. जेव्हा सर्व बिट्स स्कॅन केले जातात आणि D /A कनवर्टर आउटपुटचे मूल्य फक्त V_i ओलांडते तेव्हा EOC सिग्नल पाठविला जातो.

सकसेसीव एप्रोक्सीमेशन A/D कनवर्टरचे फायदे:

- a. याची कन्व्हर्जन गती अधिक (high) आहे.
- b. यात कन्व्हर्जन टाइम स्थिर (constant) असतो.
- c. हे फक्त 1 क्लॉकच्या पल्समध्ये 1 बिट A टू D रूपांतरण करू शकते.

तोटे:

- हा ADC महाग आहे.
- सर्किट गुंतागुंतीचे (complex) आहे.
- यासाठी DAC आवश्यक आहे.

c. फ्लॅश टाइप A/D Converter :

- फ्लॅश प्रकार A /D कन्व्हर्टर संदर्भ व्होल्टेजच्या संचासह अज्ञात अॅनालॉग इनपुट व्होल्टेजच्या समावेशावर आधारित आहे.
- एनालॉग सिग्नलला n आउटपुट बिट्स $(2^n - 1)$ च्या डिजिटल सिग्नलमध्ये रूपांतरित करण्यासाठी तुलनाकर्त्याची संख्या आवश्यक आहे.
- उदाहरणार्थ, 2 बिट A /D कनव्हर्टरला 3 किंवा $(2^2 - 1)$ कंपॅरेटरची आवश्यकता असते, तर 3 बिट कन्व्हर्टरला 7or $(2^3 - 1)$ तुलनाकारांची आवश्यकता असते.

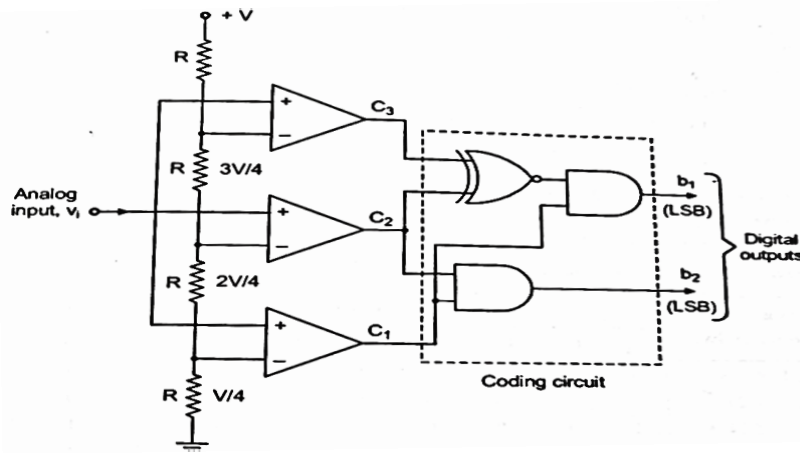


Fig 5.13 Flash Type A/D Converter

- 2 बिट फ्लॅश प्रकार A/D कनव्हर्टरचा ब्लॉक डायग्राम आकृती 5.13 मध्ये दर्शविला आहे.
- तीन ऑप अॅम्प्स एक तुलनात्मक म्हणून वापरले जातात. तिन्ही तुलनाकर्त्यांचे नॉन-इनव्हर्टिंग इनपुट अॅनालॉग इनपुट व्होल्टेजशी जोडलेले आहेत.
- OP amps चे इनव्हर्टिंग इनपुट अनुक्रमे $V/4$, $2V/4$ आणि $3V/4$ सेट संदर्भ व्होल्टेजशी जोडलेले असतात, जे रेझिस्टिव्ह डिव्हायडर नेटवर्क आणि पॉवर सप्लाय $+V$ वापरून मिळवले जातात.
- जेव्हा नॉन-इनव्हर्टिंग इनपुट टर्मिनलवरील व्होल्टेज इनव्हर्टिंग टर्मिनलवरील व्होल्टेजपेक्षा जास्त असते आणि अन्यथा ते नकारात्मक संपृक्ततेच्या स्थितीत असते तेव्हा तुलनाकर्त्याचे आउटपुट पॉजिटिव्ह संपृक्ततेच्या स्थितीत असते.
- जेव्हा अॅनालॉग इनपुट व्होल्टेज $V/4$ पेक्षा कमी असतो, तेव्हा तीन तुलनाकर्त्यांच्या नॉन-इनव्हर्टिंग टर्मिनलवरील व्होल्टेज त्यांच्या संबंधित इनव्हर्टिंग इनपुट व्होल्टेजपेक्षा कमी असतो आणि म्हणून तुलनाकर्त्याचे आउटपुट $C1C2C3=000$ असतात.
- जेव्हा अॅनालॉग इनपुट व्होल्टेज $V/4$ आणि $V/2$ पेक्षा कमी असते, तेव्हा तुलना करणारे आउटपुट $C1C2C3=100$ असतात. Table 5.1 अॅनालॉग व्होल्टेजच्या विविध श्रेणी आणि त्यांच्या संबंधित डिजिटल आउटपुटसाठी तुलनात्मक आउटपुट दाखवते.

Table 5.1: Comparator and Digital Output for 2 Bit Flash Type A/D Converter

Analog Input Conditions	Comparator Outputs			Digital output	
	C ₁	C ₂	C ₃	B ₁	B ₀
$0 \leq V_{in} \leq \frac{V}{4}$	0	0	0	0	0
$\frac{V}{4} \leq V_{in} \leq \frac{2V}{4}$	1	0	0	0	1
$\frac{2V}{4} \leq V_{in} \leq \frac{3V}{4}$	1	1	0	1	0
$\frac{3V}{4} \leq V_{in} \leq V$	1	1	1	1	1

फ्लॅश टाइप A/D Converter फायदे:

- Flash प्रकार ADC हा सर्वात वेगवान आहे कारण AD रूपांतरण एकाच वेळी तुलनाकर्त्याच्या संचाद्वारे केले जाते.
- रूपांतरण वेळ 100 nsec किंवा कमी आहे
- रूपांतरण वेळ साधे आणि डिझाइन करणे सोपे आहे.

तोटे:

हे 3 किंवा 4 पेक्षा जास्त डिजिटल आउटपुट बिटसह A/D रूपांतरणासाठी योग्य नाही. कारण n बिट ADC साठी $(2^n - 1)$ तुलनाकार (कॉम्प्यारटोर्स) आवश्यक आहेत, आणि प्रत्येक जोडलेल्या बिटसाठी आवश्यक असलेल्या तुलनाकर्त्याची (कॉम्प्यारटोर्स) संख्या दुप्पट आहे.

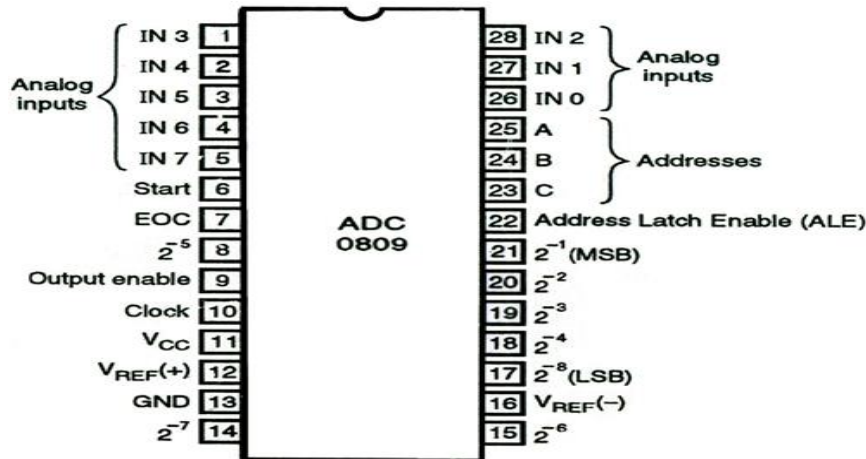
ADC 0808 चे पिन कॉन्फिगरेशन: (Pin Configuration)

Fig 5.14 : Pin configuration of ADC 0808

ADC 0808 चे तपशील (Specifications of ADC 0808 :)

- मल्टिप्लेक्सरसह अंगभूत 8 अॅनालॉग चॅनेल.
- 8 बिट रिझोल्यूशन
- कमी रूपांतरण वेळ: 100 μ सेकंद.
- कमी उर्जा वापर: 15 मेगावॉट

- e. उच्च अचूकता.
- f. आउटपुट TTL सुसंगत आहे.
- g. सर्व मायक्रोप्रोसेसरसह इंटरफेस करणे सोपे.

DAC 0808, ADC 0808 चे तपशील (Specifications of DAC 0808, ADC 0808.):

DAC 0808 चे तपशील:

a. रिझोल्यूशन (स्टेप साइज): (Resolution)

- डिजिटल इनपुटमधील बदलामुळे अॅनालॉग आउटपुटमध्ये होणारा सर्वात लहान बदल म्हणून त्याची व्याख्या केली जाते.
- रिझोल्यूशन नेहमी LSB च्या वजनाच्या बरोबरीचे असते आणि त्याला स्टेप साइज म्हणून देखील ओळखले जाते, कारण जेव्हा डिजिटल इनपुट डेटा एका पायरीवरून दुसऱ्या टप्प्यावर जातो तेव्हा V_o चे प्रमाण बदलते.

$$\% \text{ रिझोल्यूशन} = 1 / (2^n - 1) \times 100$$

b. अचूकता: (Accuracy)

- डी/ए कन्व्हर्टरची अचूकता हे वास्तविक आउटपुट व्होल्टेज आणि अपेक्षित आउटपुट व्होल्टेजमधील फरकाचे मोजमाप आहे.
- हे पूर्ण स्केल किंवा कमाल आउटपुट विद्युत दाबाची (voltage) टक्केवारी म्हणून निर्दिष्ट केले आहे.
- परिपूर्ण अचूकता (Absolute accuracy) आदर्श मूल्यापासून आउटपुटचे कमाल विचलन (deviation) परिभाषित करते आणि ते 1 LSB च्या अपूर्णाकांमध्ये व्यक्त केले जाते.

c. सेटलिंग वेळ: (Settling time)

- डी/ए कन्व्हर्टरच्या आउटपुटसाठी दिलेल्या डिजिटल इनपुटसाठी अंतिम मूल्याच्या $\pm (1/2)$ LSB च्या आत सेटल होण्यासाठी लागणारा वेळ सेटलिंग टाइम म्हणून ओळखला जातो.
- सेटलिंग वेळ (Settling time) साधारणपणे 150 nsec ते 10 μ sec पर्यंत असते.

d. रेखीयता: (Linearity)

- डी/ए कन्व्हर्टरमध्ये डिजिटल इनपुटच्या संख्यात्मक महत्त्वाच्या समान वाढीमुळे अॅनालॉग आउटपुट व्होल्टेजमध्ये समान वाढ झाली पाहिजे.
- वास्तविक सर्किटमध्ये, इनपुट आउटपुट संबंध रेखीय (Linear) नसतो. हे रेझिस्टर व्हॅल्यूज आणि स्विचेसमधील व्होल्टेजमधील त्रुटीमुळे आहे.
- कन्व्हर्टरची रेखीयता हे अचूकतेचे (precision) मोजमाप आहे ज्यासह रेखीय इनपुट आउटपुट संबंध समाधानी आहेत.

ADC 0808 चे तपशील:

a. अचूकता (Accuracy):

- हे सैद्धांतिक आउटपुट व्होल्टेजमध्ये वास्तविक आउटपुट विद्युत दाब (voltage) किती जवळ आहे हे मोजते. अचूकता परिमाणीकरण (quantization) स्तरावर (ADC चा आकार) अवलंबून असते. खालच्या बिट्सच्या तुलनेत उच्च बिट्सची अचूकता अधिक आहे.

b. रूपांतरण वेळ (Conversion time):

- ए/डी कन्व्हर्टरला अॅनालॉग इनपुट मूल्य त्याच्या समतुल्य डिजिटल डेटामध्ये रूपांतरित करण्यासाठी लागणाऱ्या वेळेला रूपांतरण वेळ म्हणतात. ते शक्य तितके किमान असावे. (100 μ sec.)

c. रिझोल्यूशन :

- हे आउटपुटमधील 1-बिट बदलासाठी आवश्यक व्होल्टेज इनपुट बदल म्हणून परिभाषित केले आहे. व्होल्टेजच्या संदर्भात रिझोल्यूशन म्हणजे पूर्ण-स्केल इनपुट व्होल्टेज भागिले स्तरांच्या एकूण संख्येने. एन-बिट एडीसीची क्रांती अशी दिली आहे,

$$\text{रिझोल्यूशन} = V_{FS}/(2^n - 1)$$

V_{FS} = पूर्ण-स्केल इनपुट व्होल्टेज, n = स्तरांच्या एकूण संख्येने

5.3 सेमीकंडक्टर मेमरी :

मेमरी हा संगणक आणि मायक्रोप्रोसेसर आधारित प्रणालींचा अपरिहार्य भाग आहे. आणि प्रोग्राम आणि डेटा संग्रहित करण्यासाठी वापरला जातो. मुख्य मेमरी घटक काही नसून सेमीकंडक्टर उपकरणे (semiconductor devices) आहेत जे कोड आणि माहिती कायमस्वरूपी साठवतात.

सेमीकंडक्टर मेमरी हा डिजिटल मेमरी तंत्रज्ञानाचा एक प्रकार आहे जो डिजिटल डेटा संचयित करण्यासाठी आणि पुनर्प्राप्त करण्यासाठी सिलिकॉन सारख्या अर्धसंवाहकांचा वापर करतो. हे सामान्यतः इलेक्ट्रॉनिक उपकरणांमध्ये वापरले जाते, जसे की संगणक, स्मार्टफोन आणि इतर डिजिटल उपकरणे, प्राथमिक किंवा दुय्यम स्टोरेज माध्यम म्हणून. सेमीकंडक्टर चिप्स वापरून अंमलात आणलेली मेमरी म्हणजे सेमीकंडक्टर मेमरी.

5.3.1 सेमीकंडक्टर मेमरीचा ब्लॉक डायग्राम: Block Diagram of Semiconductor Memory

मूलभूतपणे, सेमीकंडक्टर मेमरीच्या IC मध्ये n नंबर अॅड्रेस लाईन्स आणि m डेटा लाईन्स असतात. त्याद्वारे $2^n \times m$ बिट्सची एकूण मेमरी क्षमता प्रदान करते. याचा अर्थ असा होतो की त्यात 2^n मेमरी स्थाने आहेत आणि प्रत्येक स्थान m -bit पर्यंत डेटा संचयित करू शकते.

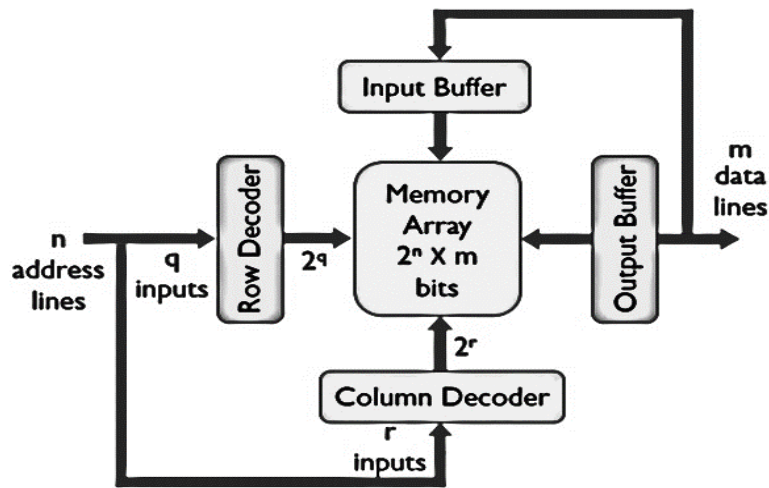


Fig 5.15 : Block diagram of semiconductor memory

- ब्लॉक डायग्राममध्ये मेमरी अॅरे आणि I/O बफरसह एक रो आणि कॉलम (row and a column) अॅड्रेस डीकोडरचा समावेश आहे. हे बफर विशिष्ट कालावधीसाठी डेटा ठेवतात.
- अॅड्रेस लाईन्सची एकूण संख्या म्हणजे, n , q आणि r मध्ये विभागली गेली आहे आणि स्वतंत्रपणे पंक्ती आणि स्तंभ डीकोडरमध्ये इनपुट म्हणून प्रदान केली आहे. या दोन डीकोडरचे आउटपुट $2n$ क्रॉसिंग पॉइंट्स असलेले $2^q \times 2^r$ आकाराचे मॅट्रिक्स अॅरे बनवते.

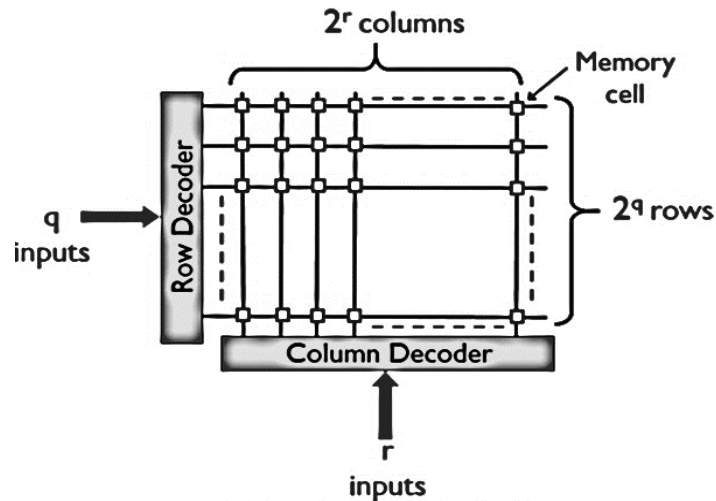


Fig 5.16 : Memory Array

- या क्रॉसिंग पॉइंट्सना मेमरी सेल म्हणून संबोधले जाते. आणि प्रत्येक मेमरी सेलमध्ये बायनरी डेटाचा एक-बिट संचयित करण्याची क्षमता असते. म्हणून, जेव्हा जेव्हा प्रोसेसर मेमरी IC ला पत्ता पाठवतो तेव्हा पंक्ती आणि स्तंभ डीकोडर त्यानुसार एक ओळ निवडतो, जे मॅट्रिक्समधून मेमरी सेल निवडते.
- अशा प्रकारे, प्रोसेसरने पाठवलेल्या पत्त्याद्वारे मेमरी सेल निवडले जातात. पुढे, व्युत्पन्न नियंत्रण सिग्नलनुसार (generated control signal) डेटा विशिष्ट निवडलेल्या मेमरी सेलमध्ये वाचला किंवा लिहिला जाऊ शकतो.
- सुरुवातीला, सेमीकंडक्टर मेमरीच्या मेमरी पेशी रेझिस्टर आणि कॅपेसिटर सारख्या निष्क्रिय घटकांपासून बनवल्या गेल्या होत्या. त्यानंतर डायोड्सचाही वापर करण्यात आला. परंतु नवीन तंत्रज्ञानाच्या आगमनाने, द्विध्रुवीय आणि एमओएस ट्रान्झिस्टरने (bipolar and MOS transistors) डायोड, प्रतिरोधक आणि कॅपेसिटरची जागा घेतली.
- आणि आजकाल मेमरी सेल CMOS आणि HMOS तंत्रज्ञान वापरून बनवले जातात ज्यात कमी उर्जा वापरासह उच्च ऑपरेशनल गती असते.

सर्व सेमीकंडक्टर मेमरिमध्ये साम्य असलेल्या काही महत्त्वाच्या बाबी म्हणजे capacity, organization आणि speed.

1. मेमरी क्षमता (Memory capacity):

सेमीकंडक्टर मेमरी चिप (chip) साठवू शकणाऱ्या बिट्सच्या संख्येला चिप कॅपॅसिटी म्हणतात. हे Kbits (kilobits), M bits (megabits) इत्यादी युनिट्समध्ये असतात. मेमरी चिपची मेमरी क्षमता नेहमी बिट्समध्ये दिली जाते, संगणक प्रणालीची मेमरी क्षमता बाइट्स (Bytes) मध्ये असते

2. मेमरी ऑर्गनायझेशन (Memory Organization):

ब्लॉक डायग्राम आकृती 5.16 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 अड्रेस 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, ही अड्रेस पिनची संख्या आहे आणि N ही चिपवरील डेटा पिनची संख्या आहे.

3. गती (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.

मेमरी बँक (Memory Bank) : हे इलेक्ट्रॉनिक्समधील स्टोरेजचे लॉजिकल युनिट आहे, जे हार्डवेअरवर अवलंबून आहे. संगणकामध्ये, हार्डवेअर मेमरी स्लॉटच्या भौतिक संस्थेसह मेमरी कंट्रोलरद्वारे मेमरी बँक निर्धारित केली जाऊ शकते. ठराविक सिंक्रोनस डायनॅमिक रँडम-एॅक्सेस मेमरी (SDRAM) किंवा दुहेरी डेटा रेट SDRAM (DDR SDRAM) मध्ये, बँकेमध्ये स्टोरेज युनिट्सच्या अनेक पंक्ती आणि स्तंभ असतात आणि सहसा अनेक चिप्समध्ये पसरलेले असते. एकाच रीड किंवा राइट ऑपरेशनमध्ये, फक्त एका बँकेत प्रवेश केला जातो, म्हणून स्तंभ किंवा एका पंक्तीमधील बिट्सची संख्या, प्रति बँक आणि प्रति चिप, बिट्समधील मेमरी बसच्या रुंदीच्या (सिंगल चॅनेल) समान असते. बँकेचा आकार पुढे एका स्तंभातील बिट्स आणि एका पंक्तीमध्ये, प्रति चिप, बँकेतील चिप्सच्या संख्येने गुणाकार करून निर्धारित केला जातो.

काही संगणकांमध्ये RAM च्या अनेक एकसारख्या मेमरी बँक असतात आणि त्या दरम्यान स्विक करण्यासाठी बँक स्विचिंग वापरतात. हार्वर्ड आर्किटेक्चर कॉम्प्युटरमध्ये (किमान) मेमरीच्या दोन अगदी भिन्न बँक असतात, एक प्रोग्राम स्टोरेजसाठी आणि दुसरा डेटा स्टोरेजसाठी.

5.3.2 सेमीकंडक्टर मेमरीचे वर्गीकरण:

सेमीकंडक्टर मेमरी दोन मुख्य श्रेणींमध्ये विभागली जाऊ शकते: अस्थिर आणि नॉन-अस्थिर मेमरी. अस्थिर मेमरी, जसे की डायनॅमिक रँडम-एॅक्सेस मेमरी (डीआरएएम) आणि स्टॅटिक रँडम-एॅक्सेस मेमरी (एसआरएएम), संचयित डेटा टिकवून ठेवण्यासाठी शक्ती आवश्यक आहे. नॉन-व्होलॅटाइल मेमरी, जसे की केवळ-वाचनीय मेमरी (ROM), फ्लॅश मेमरी आणि इलेक्ट्रिकली इरेजेबल प्रोग्रामेबल रीड-ओन्ली मेमरी (EEPROM), वीज बंद असतानाही डेटा राखून ठेवू शकते.

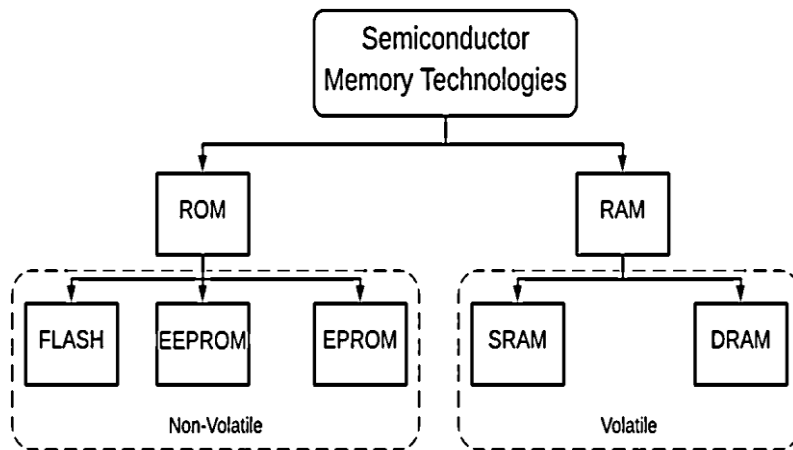


Fig 5.16: Classification of Semiconductor Memory

रीड-ओन्ली मेमरी (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.

रीड-ओन्ली मेमरी (ROM) चे विविध प्रकार आणि त्यांची वैशिष्ट्ये.

1. PROM (Programmable ROM):

PROM ही केवळ वाचनीय मेमरी आहे. युजर फक्त एकदाच मॉडिफाय करू शकतो. जर रिक्त (blank) PROM खरेदी करतो आणि PROM प्रोग्राम वापरून इच्छित (desired) कन्टेन्ट स्टोर (store) करतो PROM चिपच्या आत लहान प्यूज असतात जे प्रोग्रामिंग दरम्यान बर्न होतात हे फक्त एकदाच प्रोग्राम केले जाऊ शकते आणि इरेज येत नाही.

2. EPROM (Erasable and Programmable Read Only Memory)

EPROM काही मिनिटांपर्यंत अल्ट्राव्हायोलेट 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, अल्ट्राव्हायोलेट लाईट कार्डज क्रिस्टल विंडोमधून पाठवली जाते. संपर्कात येण्यामुळे चार्ज नष्टहोतो त्यामुळे वापरताना कार्डजचे लिड स्टिकरने सील केले जाते.

3. 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. EEPROM मध्ये, इरेज करणे आणि प्रोग्रामिंग दोन्ही साठी 4 ते 10 मिलिसेकंड वेळ घेतो. 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.

4. फ्लॅश मेमरी (Flash Memory)

फ्लॅश मेमरी ही एक अस्थिर (nonvolatile) स्टोरेज मेमरी चिप आहे जी ब्लॉक किंवा सेक्टर म्हणून ओळखल्या जाणाऱ्या छोट्या युनिट्समध्ये प्रोग्राम केलेली किंवा लिहिली जाऊ शकते.

फ्लॅश मेमरीची सामग्री किंवा डेटा, जी EEPROM प्रकारची संगणक मेमरी आहे, वीज पुरवठा बंद असताना गमावला जाऊ शकत नाही.

संगणकावरून डिजिटल उपकरणावर डेटा पाठविण्यासाठी देखील याचा वापर केला जातो.

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-

रैंडम अॅक्सेस मेमरी (Random Access Memory) (RAM):

RAM- रैंडम अॅक्सेस मेमरी (ही डेटा, प्रोग्राम आणि प्रोग्राम परिणाम (result) साठविण्यासाठी CPU अंतर्गत मेमरी आहे ही Read / Write मेमरी आहे जी मशीन काम करेपर्यंत डेटा संग्रहित करते. मशीन बंद होताच डेटा delete केला जातो.

रैंडम 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 रॅम खूप महाग आहे.

रॅम अस्थिर असते, म्हणजे जेव्हा आपण संगणक बंद करतो किंवा विद्युत प्रवाहामध्ये बिघाड होतो तेव्हा त्यात साठवलेला डेटा गमावला जातो म्हणून बॅकअप अनइंटरप्टिबल पॉवर सिस्टीम (UPS) बहुतेक वेळा संगणकांसह वापरली जाते रॅम त्याच्या भौतिक (physical) आकाराच्या दृष्टीने आणि तो ठेवू शकणाऱ्या डेटाच्या प्रमाणात लहान आहे.

RAM दोन प्रकारची असतात

a) स्थिर रॅम (Static RAM)

b) डायनॅमिक रॅम (Dynamic RAM)

a) स्टॅटिक रॅम (SRAM):

स्टॅटिक रॅममध्ये जोपर्यंत पॉवर पुरवली जात आहे तोपर्यंत मेमरी त्यातील कन्टेन्ट राखून ठेवते. अस्थिर नेचरमुळे पॉवर कमी झाल्यावर डेटा गमावला जातो. SRAM चिप्स मध्ये कॅपॅसिटर 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 अॅक्सेस आहे.

स्टॅटिक RAM ची वैशिष्ट्ये

1. लॉग लाइफ
2. रीफ्रेश करण्याची गरज नाही
3. जलद
4. कॅशे मेमरी म्हणून वापरली जाते.
5. मोठा आकार
6. महाग
7. जास्त ऊर्जेचा वापर होतो (high power consumption)

डायनॅमिक रॅम :Dynamic (DRAM):

DRAM ला डेटा राखण्यासाठी (hold) सतत रीफ्रेश करणे आवश्यक आहे हे रीफ्रेश सर्किटवर .

मेमरी ठेवून केले जाते जे प्रति सेकंद अनेक वेळा डेटा पुन्हा लिहिते DRAM बहु 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 रीफ्रेश करणे आवश्यक आहे
3. SRAM च्या तुलनेत स्लो आहे.
5. आकाराने लहान
6. कमी खर्चिक
7. वीज वापर कमी होतो.

5.3.3 RAM आणि ROM, EPROM आणि फ्लॅश मेमरीची तुलना रॅम आणि रोम मधील तुलना (Difference between RAM AND ROM)

Table 5.2 : Comparison between RAM and ROM

रॅम (RAM)	रॉम (ROM)
ही एक रँडम-एँक्सेस मेमरी आहे.	ही एक रँडम-एँक्सेस मेमरी आहे. ती केवळ वाचनीय मेमरी आहे.
चन आणि लेखन ऑपरेशन केले जाऊ शकते.	चन आणि लेखन ऑपरेशन केले जाऊ शकते. फक्त ड ऑपरेशन केले जाऊ शकते.
व्हा वीज पुरवठा बंद असतो तेव्हा अस्थिर मेमरीमध्ये डेटा गमावला जाऊ शकतो.	व्हा वीज पुरवठा बंद असतो तेव्हा अस्थिर मेमरीमध्ये डेटा गमावला जाऊ शकतो. वीज पुरवठा बंद असताना नॅन-व्होलॅटाइल मेमरीमध्ये डेटा गमावला जाऊ शकत नाही.
RAM मध्ये स्टोरेज डेटा रिक्रेश करणे आवश्यक आहे.	ROM मध्ये स्टोरेज डेटा रिक्रेश करण्याची आवश्यकता नाही.
टा साठवण्यासाठी चिपचा आकार रॉम चिपपेक्षा मोठा असतो.	मान प्रमाणात डेटा साठवण्यासाठी चिपचा आकार रॅम चिपपेक्षा लहान असतो.

Table 5.3: Comparison between Static RAM and Dynamic RAM

स्टॅटिक रॅम (Static RAM)	डायनॅमिक रॅम (Dynamic RAM)
ही एक स्थिर रँडम-एँक्सेस मेमरी आहे.	ही एक डायनॅमिक रँडम-एँक्सेस मेमरी आहे.
SRAM चा प्रवेश वेळ मंद आहे.	DRAM चा प्रवेश वेळ जास्त आहे.
प्रत्येक बिट माहिती साठवण्यासाठी ते फ्लिप-फ्लॉप वापरते.	प्रत्येक बिट माहिती साठवण्यासाठी ते कॅपेसिटर वापरते.
माहिती जतन करण्यासाठी वेळोवेळी रीफ्रेश करण्याची आवश्यकता नाही.	माहिती जतन करण्यासाठी वेळोवेळी रीफ्रेश करणे आवश्यक आहे.
हे कॅशे मेमरीमध्ये वापरते.	हे मुख्य मेमरीमध्ये वापरले जाते.
SRAM ची किंमत महाग आहे.	DRAM ची किंमत कमी आहे.

Table 5.4: Comparison between EPROM and Flash Memory

ईपीरॉम (EPROM)	फ्लॅश (Flash)
इलेक्ट्रिकल पल्स देऊन फक्त बाइट बाय बाइट मिटवता येते	इलेक्ट्रिकल पल्स देऊन ब्लॉक करून ब्लॉक मिटवता येतो.
बाइट प्रोग्राम करण्यायोग्य	ब्लॉक प्रोग्राम करण्यायोग्य
खर्च जास्त आहे.	खर्च कमी आहे
प्रोग्रामिंग जलद आहे	प्रोग्रामिंग जलद आहे

5.3.4 सिंगल इन-लाइन मेमरी मॉड्यूल आणि सॉलिड स्टेट ड्राइव्हे: (SIMM आणि SSD)

सिंगल इन-लाइन मेमरी मॉड्यूल:

- हा मेमरी मॉड्यूलचा एक प्रकार आहे जो 1980 च्या सुरुवातीपासून ते 2000 च्या दशकाच्या सुरुवातीच्या काळात संगणकांमध्ये वापरला जातो. हा एक मुद्रित सर्किट बोर्ड आहे ज्यावर रँडम-ऍक्सेस मेमरी एका किंवा दोन्ही बाजूंना जोडलेली असते.
- SIMM संगणकावर उपलब्ध RAM (रँडम ऍक्सेस मेमरी) वाढवते. RAM संगणकांना मल्टीटास्क आणि वेगाने कार्य करण्याची क्षमता प्रदान करते आणि सामान्यतः RAM जितकी जास्त असेल तितकी कार्यक्षमता चांगली असते. SIMM लहान सर्किट बोर्डवर मेमरी मॉड्यूल एकत्र करते. हे एका स्लॉटमध्ये प्लग इन करते जे संगणकाच्या मदरबोर्डला जोडते. काही संगणकांमध्ये अनेक स्लॉट असतात. हे RAM ला अनेक वेळा वाढवण्याची परवानगी देते. हे लक्षात घेणे महत्वाचे आहे की काही जुन्या प्रणाली अजूनही SIMM वापरत असताना, बहुतेक प्रकरणांमध्ये ते ड्युअल इनलाइन मेमरी मॉड्यूलस (DIMM) ने बदलले आहेत, जे खूप जलद आणि अधिक कार्यक्षम आहेत.
- सिंगल इन-लाइन मेमरी मॉड्यूलस (SIMM) म्हणजे लहान सर्किट बोर्ड ज्या ठिकाणी रॅम चिप्स फिक्स केल्या जातात त्या ठिकाणी नॉच असतात. SIMM कनेक्टर आणि म्हणून मदरबोर्डवर असलेले स्लॉट धातूचे (सोने किंवा टिन) बनलेले आहेत. SIMM मध्ये, दोन्ही बाजूंच्या पिन जोडलेल्या असतात. दोन प्रकारचे SIMM प्रेझेंट्स आहेत, एक 30 पिनसह आणि दुसरे 72 पिनसह.

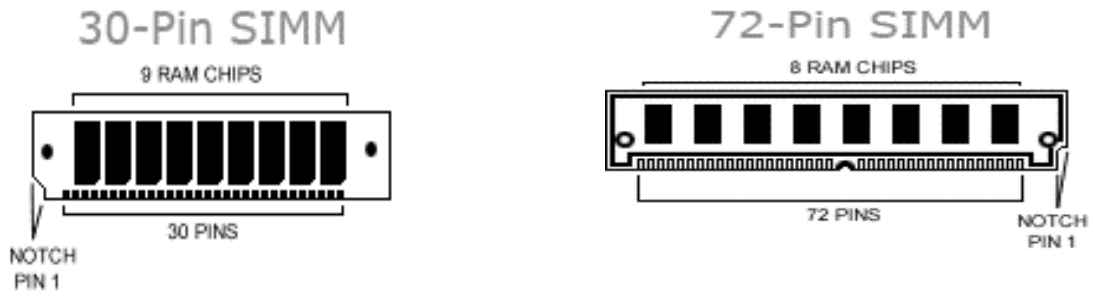


Fig.5.18 : 30,72 Pin Single In-Line Memory Modules (SIMMs)

सिंगल इन-लाइन मेमरी मॉड्यूल वैशिष्ट्ये :

1. सिंगल इन-लाइन मेमरी मॉड्यूल (SIMM) मध्ये, दोन्ही बाजूंच्या पिन जोडलेल्या असतात.
2. डेटा ट्रान्सफर करण्यासाठी SIMM 32 बिट चॅनेलला सपोर्ट करते.
3. SIMM 5 व्होल्ट पॉवर वापरते.
4. SIMM 4 MB ते 64 MB स्टोरेज प्रदान करते.
5. SIMM मॉड्यूलचे क्लासिक किंवा सर्वात सामान्य पिन कॉन्फिगरेशन 72 पिन आहे.
6. सिम एका वेळी जोड्यांमध्ये स्थापित केले जातात.
7. SIMM चा वापर 486 CPU तसेच सुरुवातीच्या Pentium संगणकांद्वारे केला जातो.
8. SIMM ची लांबी आणि रुंदी अनुक्रमे 4.25 इंच आणि 1 इंच आहे
9. SIMM मध्ये सिंगल नॉचेस असतात

सॉलिड स्टेट ड्राइव्ह (Solid State Drive) SSD :

- SSD या शब्दाचा अर्थ सॉलिड स्टेट ड्राइव्ह आहे. एसएसडी, किंवा सॉलिड-स्टेट ड्राइव्ह, संगणकांमध्ये वापरल्या जाणाऱ्या स्टोरेज डिव्हाइसचा एक प्रकार आहे. हे नॉन-व्हॉलॅटाइल स्टोरेज मीडिया सॉलिड-स्टेट फ्लॅश मेमरीवर सतत डेटा संग्रहित करते. एसएसडी संगणकामध्ये पारंपारिक हार्ड डिस्क ड्राइव्ह (एचडीडी) बदलतात आणि हार्ड ड्राइव्ह

प्रमाणेच मूलभूत कार्ये करतात. परंतु SSDs तुलनेत लक्षणीय वेगवान आहेत. SSD सह, डिव्हाइसची ऑपरेटिंग सिस्टम अधिक वेगाने बूट होईल, प्रोग्राम जलद लोड होतील आणि फाइल्स जलद जतन केल्या जाऊ शकतात.

- पारंपारिक हार्ड ड्राइव्हमध्ये ॲक्ट्युएटर नावाच्या यांत्रिक हातावर रीड/राईट हेड असलेली स्पिनिंग डिस्क असते. HDD चुंबकीय पद्धतीने डेटा वाचतो आणि लिहितो. तथापि, चुंबकीय गुणधर्मांमुळे यांत्रिक बिघाड होऊ शकतो.
- तुलनेने, एसएसडीमध्ये तुटण्यासाठी किंवा वर किंवा खाली फिरण्यासाठी कोणतेही हलणारे भाग नाहीत. SSD मधील दोन प्रमुख घटक म्हणजे फ्लॅश कंट्रोलर आणि NAND फ्लॅश मेमरी चिप्स. हे कॉन्फिगरेशन सीक्वेंशियल आणि रँडम ऍक्सेस मेमरी डेटा विनंत्यांसाठी उच्च वाचन/लेखन कार्यप्रदर्शन वितरीत करण्यासाठी ऑप्टिमाइझ (optimize) केले आहे.
- एसएसडी कुठेही वापरले जातात जेथे हार्ड ड्राइव्ह तैनात केले जाऊ शकतात. ग्राहक उत्पादनांमध्ये, उदाहरणार्थ, ते वैयक्तिक संगणक (पीसी), लॅपटॉप, संगणक गेम, डिजिटल कॅमेरे, डिजिटल संगीत प्लेअर, स्मार्टफोन, टॅब्लेट आणि थंब ड्राइव्हमध्ये वापरले जातात. ते ग्राफिक्स कार्डसह देखील समाविष्ट केले आहेत.



Fig 5.19 : Solid State Drive

सॉलिड स्टेट ड्राइव्ह (SSD) ची प्रमुख वैशिष्ट्ये :

- एसएसडीच्या डिझाइनची अनेक वैशिष्ट्ये आहेत. त्याचे कोणतेही हलणारे भाग नसल्यामुळे, एसएसडी हे त्याच यांत्रिक बिघाडांच्या अधीन नाही जे HDD मध्ये येऊ शकतात.
- SSDs देखील शांत असतात आणि कमी उर्जा वापरतात. आणि SSD चे वजन हार्ड ड्राइव्हपेक्षा कमी असल्यामुळे, ते लॅपटॉप आणि मोबाईल कॉम्प्युटिंग उपकरणांसाठी योग्य आहेत.
- याव्यतिरिक्त, SSD कंट्रोलर सॉफ्टवेअरमध्ये भविष्यसूचक विश्लेषणे समाविष्ट आहेत जी वापरकर्त्यास संभाव्य ड्राइव्ह अयशस्वी होण्याआधी अलर्ट करू शकतात.
- फ्लॅश मेमरी निंदनीय असल्यामुळे, ऑल-फ्लॅश ॲर्रे विक्रेते डेटा रिडक्शन तंत्र वापरून वापरण्यायोग्य स्टोरेज क्षमता हाताळू शकतात.

स्वाध्याय:

1. डिजिटल ते अॅनालॉग कन्व्हर्टर IC-0808 ची कोणतीही दोन वैशिष्ट्ये सूचीबद्ध करा.
2. डिजिटल ते अॅनालॉग कन्व्हर्टर च्या प्रकारांची यादी करा.
3. अॅनालॉग ते डिजिटल कन्व्हर्टर IC-0808/0809 ची कोणतीही दोन वैशिष्ट्ये सूचीबद्ध करा.
4. अॅनालॉग ते डिजिटल कन्व्हर्टर च्या प्रकारांची यादी करा.
5. अॅनालॉग ते डिजिटल कन्व्हर्टर शी संबंधित रिझोल्यूशन आणि रूपांतरण वेळ परिभाषित (Define) करा.
6. R-2R लॅडर डिजिटल ते अॅनालॉग कन्व्हर्टर चे सर्किट काढा आणि कार्याचे वर्णन करा.
7. सकसेसीव एप्रोक्सीमेशन अॅनालॉग ते डिजिटल कन्व्हर्टर चे सर्किट काढा आणि कार्याचे वर्णन करा.
8. ड्युअल स्लोप अॅनालॉग ते डिजिटल कन्व्हर्टर चे सर्किट काढा आणि कार्याचे वर्णन करा.
9. 5 बिट डिजिटल ते अॅनालॉग कन्व्हर्टर (DAC) च्या डिजिटल इनपुटसाठी 10011, अॅनालॉग आउटपुटची गणना करा; $V_F = 5V$ गृहीत धरा.
10. मेमरीचे वर्गीकरण मेमरीचे वर्गीकरण लिहा
11. ईपीरॉम आणि फ्लॅश मेमरी च्या कामाची तुलना करा.
12. सॉलिड स्टेट ड्राइव्ह ची कोणतीही दोन वैशिष्ट्ये सूचीबद्ध करा.

असाइनमेंट/ लघु प्रकल्प

1. अॅनालॉग ते डिजिटल कन्व्हर्टर (ADC) IC 0808 डेटाशीट डाउनलोड करा
2. डेटा शीट वापरून त्याची वैशिष्ट्ये आणि पिन फंक्शन्स प्रदर्शित करण्यासाठी विविध प्रकारच्या ADC आणि DAC चा चार्ट तयार करा.
3. ADC 0808 वापरून तापमान नियंत्रकावर प्रकल्प तयार करा

संदर्भ (References):

Sr. No	Author	Title	Publisher with ISBN Number
1	Jain R.P	Modern Digital Electronics	McGraw-Hill Publishing, New Delhi, 2009 ISBN: 9780070669116
2	Salivahanan S, Arivazhagan S.	Digital Circuits and Design	Vikas Publishing House, New Delhi, 2013 ISBN: 9789325960411
3	G. K. Kharate	Digital Electronics	OXFORD University press, ISBN: 9780198061830

संकेतस्थल

- https://www.google.com/search?q=npTEL+adc%2Cdac+types&oeq=npTEL+adc%2Cdac+types&gs_lcrp=EgZjaHJvbWUyBggAEEUYOTIHCAEQIRigAdIBCTEWNDgyajBqN6gCCLACAQ&sourceid=chrome&ie=UTF-8#fpstate=ive&vld=cid:af6864c2,vid:wa7plviT-do,st:0
- <https://www.techtarget.com/searchstorage/definition/SSD-solid-state-drive>
- <https://www.geeksforgeeks.org/difference-between-simm-and-dimm/>

