महाराष्ट्र राज्य तंत्रशिक्षण मंडळ, मुंबई

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

अभियांत्रिकी आणि तंत्रज्ञान पदविका

शिक्षण पुस्तिका

(Learning Material)

डिजिटल इलेक्ट्रॉनिक्स आणि मायक्रोकंट्रोलर ॲप्लिकेशन्स

३१४३२४

Digital Electronics & Microcontroller Applications

(314324)

विद्युत अभियांत्रिकी गट

(के -स्कीम)

मराठी – इंग्रजी (द्विभाषिक) माध्यम

(अभियांत्रिकी व तंत्रज्ञानातील चतुर्थ सत्र पदविका)

शिक्षण पुस्तिका

(Learning Material)

डिजिटल इलेक्ट्रॉनिक्स आणि मायक्रोकंट्रोलर ॲप्लिकेशन्स

३१४३२४

Digital Electronics & Microcontroller Applications

विद्युत अभियांत्रिकी गट

मराठी - इंग्रजी (द्विभाषिक) माध्यम

(अभियांत्रिकी व तंत्रज्ञानातील चतुर्थ सत्र पदविका)

महाराष्ट्र राज्य तंत्रशिक्षण मंडळ, मुंबई

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

डिजिटल इलेक्ट्रॉनिक्स आणि मायक्रोकंट्रोलर ॲप्लिकेशन्स

Digital Electronics & Microcontroller Applications

(314324)

मार्गदर्शक

डॉ.संजय वसंत भंगाळे
विभाग प्रमुख, विद्युत अभियांत्रिकी

विषय समन्वयक/समिक्षक

श्री.तुषार मोहिते पाटील
अधिव्याख्याता(निवड श्रेणी),
औद्योगिक अणुविद्युत अभियांत्रिकी

संकलक

सौ. योगिनी बागडे
व्याख्याता, विद्युत अभियांत्रिकी

श्री. दिपक कुलकर्णी
अधिव्याख्याता(निवड श्रेणी),
अणुविद्युत आणि दूरसंचार अभियांत्रिकी

कु. प्रज्ञा बोडके
व्याख्याता, औद्योगिक अणुविद्युत अभियांत्रिकी

सौ. गौतमी पुजारे
अधिव्याख्याता, संगणक अभियांत्रिकी

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

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

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

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

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

(प्रमोद नाईक)
संचालक
म. रा. तंत्र शिक्षण मंडळ, मुंबई.

अनुक्रमणिका

अ.क्र.	युनिटचे नाव	पृष्ठ क्रमांक
1	नंबर सिस्टीम आणि लॉजिक गेट्स **(Number System and Logic Gates)**	1
2	डिजिटल लॉजिक सर्किट्स **(Digital Logic Circuits)**	23
3	8051 मायक्रोकंट्रोलर आर्किटेक्चर **(8051 Microcontroller Architecture)**	48
4	8051 इंस्ट्रक्शन सेट आणि प्रोग्रामिंग **(8051 Instruction Set and Programming)**	63
5	8051 इंटरफेसिंग आणि ॲप्लिकेशन्स **(8051 Interfacing and Application)**	70

युनिट - I
नंबर सिस्टीम आणि लॉजिक गेट्स
(Number System and Logic Gates)

विषय निष्पत्ती (Course Outcomes COs)

डिजिटल सिस्टीमच्या कामात नंबर सिस्टीम आणि लॉजिक सर्किट्सचे ज्ञान वापरणे.

(To apply knowledge of number system and logic circuits in working of digital system.)

विषय निष्पत्ती बरोबर जुळणारी थिअरी लर्निंग निष्पत्ती (Theory Learning Outcomes (TLOs) aligned with Course Outcomes (CO's))

1.1 दिलेल्या संख्येला ओळखा आणि निर्दिष्ट नंबर सिस्टीम मध्ये रूपांतरित करा. (Recognize and convert the given number into the specified number system.)

1.2 दिलेल्या संख्यांवर बायनरी आणि BCD अंकगणितीय क्रिया करा.(Perform the binary and BCD arithmetic operation on the given numbers.)

1.3 दिलेल्या NAND/NOR गेटला युनिव्हर्सल गेट म्हणून वापरून बेसिक गेट्स विकसित करा. (Develop the basic gates using the given NAND/NOR gate as universal gate.)

1.4 लॉजिक गेट्स वापरून ॲडर आणि सबट्रॅक्टर सर्किट तयार करा. (Construct adder and subtractor circuit using logic gates)

1.1 नंबर सिस्टीम: डेसिमल, बायनरी, ऑक्टल, हेक्साडेसिमल, बीसीडी, एका प्रणालीचे दुसऱ्या प्रणालीमध्ये रूपांतर. (Number System: Decimal, Binary, octal, hexadecimal, BCD. Conversion of one system into other.)

नंबर सिस्टीम- नंबर सिस्टीम ही चिन्हांचा क्रम संच आहे. हा एक संख्या दर्शविण्याचा किंवा व्यक्त करण्याचा मार्ग आहे. संगणकाच्या संदर्भात काहि विशिष्ट प्रकारच्या नंबर सिस्टीम परिभाषित आहेत:

a. डेसिमल नंबर सिस्टीम(Decimal Number system)
b. बायनरी नंबर सिस्टीम(Binary Number system)
c. ऑक्टल नंबर सिस्टीम(Octal Number system)
d. हेक्साडेसिमल नंबर सिस्टीम(Hexadecimal Number system)

नंबर सिस्टीमचा आधार किंवा मूलांक (Base)_- नंबर सिस्टीमचा आधार म्हणजे संख्या प्रणालीतील अंक किंवा बिट्सची संख्या आहे. उदाहरणार्थ डेसिमल नंबर सिस्टीमचा आधार/ बेस 10 आहे कारण त्यात एकूण 10 संख्या आहेत (म्हणजे 0 ते 9).

1.1.1 डेसिमल नंबर सिस्टीम (बेस 10) (Decimal Number system base 10)-

- या नंबर सिस्टीममध्ये, संख्या दर्शवण्यासाठी 0 ते 9 अंक वापरले जातात.
- एखाद्या संख्येचे प्रतिनिधित्व करण्यासाठी 10 अंकांचा वापर केल्यामुळे, त्याला बेस 10 नंबर सिस्टीम देखील म्हणतात.
- प्रत्येक अंकाला त्याच्या स्थानावर आधारित मूल्य असते ज्याला स्थान मूल्य म्हणतात.
- जेव्हा आपण संख्येमध्ये उजवीकडून डावीकडे जातो तेव्हा स्थितीचे मूल्य 10 पट वाढते.

उदाहरणार्थ, $(786)_{10}$ चे मूल्य आहे

$= 7 \times 10^2 + 8 \times 10^1 + 6 \times 10^0$

$= 700 + 80 + 6 = 786$

1.1.2 बायनरी नंबर सिस्टीम (बेस 2) (Binary Number system (base 2) -

- संगणक फक्त स्विचची "चालू" आणि "बंद" स्थिती समजू शकतो.
- ह्या दोन अवस्था 1 आणि 0 ने दर्शविलेल्या आहेत.
- 1 आणि 0 चे संयोजन बायनरी संख्या बनवते.
- हे आकडे विविध डेटाचे प्रतिनिधित्व करतात.
- संख्या दर्शवण्यासाठी दोन अंकांचा वापर केला जातो म्हणून त्याला बायनरी किंवा बेस 2 नंबर सिस्टीम म्हणतात.

- बायनरी नंबर सिस्टीम पोझिशनल नोटेशन (Positional notation) वापरते. परंतु यात, प्रत्येक अंकचा त्याच्या स्थानावर आधारित 2 च्या योग्य घाताने गुणाकार केला जातो.
- उदाहरणार्थ, $(101101)_2$ ही बायनरी (binary) संख्या आहे ज्यामध्ये फक्त दोन बिट 0s आणि 1s चे संयोजन असते.
- डेसिमल मध्ये वरील संख्येचे मूल्य आहे-

$= 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

$= 1 \times 32 + 0 \times 16 + 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1$

$= 32 + 8 + 4 + 1$

$= (45)_{10}$

डेटाच्या मोजमापाची एकके (Units of measurement of data): मशीन भाषा(Machine language)ही बायनरी आहे आणि त्यामुळे संगणकात साठवलेल्या डेटाचे मोजमाप कसे करावे याबद्दल समजून घेणे आवश्यक आहे. डेटा मोजण्यासाठी बिट (bit) आणि बाइट (byte) ही एकके आहेत.

1. बिट(bit)

'बिट' हा शब्द 'बायनरी'(binary) आणि 'डिजिट'(digit) या शब्दांचे अब्रेव्हेशन(abbreviation) आहे. हे मेमरी किंवा सूचनांचे सर्वात लहान एकक आहे जे संगणकावर दिले जाऊ शकते किंवा संग्रहित केले जाऊ शकते. बिट एकतर 0 किंवा 1 आहे. वरील उदाहरणातील संख्या ही 6-बिट संख्या आहे कारण त्यात 6 बायनरी अंक आहेत (0s आणि 1s).

2. निबल(Nibble)

निबल हा शब्द ४ बायनरी बिट्सच्या गटासाठी वापरला जातो. उदा. 1101

0101 1101

अप्पर निबल ← → लोअर निबल

3. बाइट (Byte)

01100001 सारखा 8 बिटचा समूह म्हणजे बाइट. बाइट्सचे संयोजन किलोबाइट सारख्या विविध नावांसह येते. एक किलोबाइट(kilobyte) म्हणजे 1000 बाइट्सचा संग्रह. 1KB = 1000 बाइट्स

1.1.3 ऑक्टल नंबर सिस्टीम (बेस 8) (Octal Number System (Base 8)-

- या नंबर सिस्टीम मध्ये संख्या दर्शवण्यासाठी 0 ते 7 अंक (म्हणजे 8 अंक) वापरतात.
- या नंबर सिस्टीम मध्ये प्रणालीचा आधार 8 आहे.

उदाहरणार्थ, डेसिमल मध्ये $(24)_8$ आहे

$= 2 \times 8^1 + 4 \times 8^0$

$= 16 + 4$

$= (20)_{10}$

1.1.4 हेक्साडेसिमल नंबर सिस्टीम (बेस 16)- (Hexadecimal Number System (Base 16)-

- या नंबर सिस्टीम मध्ये, दिलेल्या संख्येचे प्रतिनिधित्व करण्यासाठी 16 अंक वापरले जातात.
- याला बेस 16 (base 16) नंबर सिस्टीम म्हणूनही ओळखले जाते.
- प्रत्येक अंकाचे स्थान मूल्य हे 16 च्या पटीत वाढते.
- आधार 10 पेक्षा जास्त असल्याने, या नंबर सिस्टीम मध्ये अक्षरांचाही समावेश आहे.
- खालील हेक्साडेसिमल चिन्हे आहेत: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

उदाहरणार्थ, $(2C4)_{16}$ डेसिमल मध्ये आहे

$= 2 \times 16^2 + C \times 16^1 + 4 \times 16^0$

$= 2 \times 256 + 12 \times 16 + 4 \times 1$

$= 512 + 192 + 4$

$= (708)_{10}$

Name	Base/ Radix	Symbols	Example
Decimal	10	0,1,2,3,4,5,6,7,8,9	$(2795)_{10}$
Binary	2	0,1	$(111000010)_2$
Octal	8	0,1,2,3,4,5,6,7	$(1576)_8$
Hexadecimal	16	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F	$(3DB)_{16}$

Table no **1.1 (**Number System Summary Chart 1)

Decimal	Hexadecimal	Binary	Octal	Binary representation of octal
0	0	0000	0	000
1	1	0001	1	001
2	2	0010	2	010
3	3	0011	3	011
4	4	0100	4	100
5	5	0101	5	101
6	6	0110	6	110
7	7	0111	7	111
8	8	1000	10	001 000
9	9	1001	11	001 001
10	A	1010	12	001 010
11	B	1011	13	001 011
12	C	1100	14	001 100
13	D	1101	15	001 101
14	E	1110	16	001 110
15	F	1111	17	001 111

Table no 1.2 (Number System Chart 2)

1.1.5 एका नंबर सिस्टीममधून दुसऱ्या नंबर सिस्टीममध्ये नंबरचे रुपांतर- (Conversion of Number from one number system to another number system-)

रूपांतरणे खालीलप्रमाणे वर्गीकृत केली जाऊ शकतात -

1. इतर नंबर सिस्टीमपासून डेसिमल नंबर सिस्टीममध्ये संख्येचे रूपांतर

i बायनरी ते डेसिमल (Binary to Decimal)

ii ऑक्टल ते डेसिमल (Octal to Decimal)

iii हेक्साडेसिमल ते डेसिमल (Hexadecimal to Decimal)

2. डेसिमल संख्येचे इतर नंबर सिस्टीममध्ये रूपांतर. (Conversion of Decimal Number to Other Number System)

i डेसिमल ते बायनरी (Decimal to Binary)

ii डेसिमल ते ऑक्टल (Decimal to Octal)

iii डेसिमल ते हेक्साडेसिमल (Decimal to Hexadecimal)

3. कोणत्याही नंबर सिस्टीमचे डेसिमल व्यतिरिक्त इतर कोणत्याही नंबर सिस्टीममध्ये रूपांतर (Conversion of any number system to any number system other than Decimal)

i बायनरी ते ऑक्टल (Binary to octal)

ii. बायनरी ते हेक्साडेसिमल Binary to Hexadecimal

iii ऑक्टल ते बायनरी(Octal to binary)

iv. ऑक्टल ते हेक्साडेसिमल(Octal to hexadecimal)

v. हेक्साडेसिमल ते बायनरी(Hexadecimal to binary)

vi. हेक्साडेसिमल ते ऑक्टल(Hexadecimal to Octal)

1.1.5 a इतर नंबर सिस्टीमवरून डेसिमल नंबर सिस्टीममध्ये संख्येचे रूपांतर (Conversion of number from Other number system to Decimal Number)

इतर नंबर सिस्टीमवरून संख्या डेसिमल संख्येमध्ये रूपांतरित करण्यासाठी, बेसच्या पॉवर्ससह वाढत्या क्रमाने प्रत्येक अंकाचा गुणाकार करा.

i) बायनरी ते डेसिमल (Binary to Decimal)

Ex- $(110110)_2$ बायनरी डेसिमल संख्येत रूपांतरित करा.

दिलेल्या बायनरी संख्येचे डेसिमल संख्येत रूपांतर करण्यासाठी, खाली दर्शविल्याप्रमाणे बायनरी संख्येच्या प्रत्येक अंकाचा, बेस (2) च्या पॉवर्ससह गुणाकार करा आणि बेरीज करा-

डेसिमल संख्या = $1X2^5 + 1X2^4 + 0x2^3 + 1X2^2 + 1X2^1 + 0X2^0$

= 1x32 + 1x16 + 0x8 + 1x4 + 1x2 + 0x1

= 32+16+0+4+2+0

= 54

म्हणून बायनरी संख्या $(110110)_2$ चा डेसिमल इक्विव्हॅलेंट (Equivalent)$(54)_{10}$ आहे.

उत्तर: $(110110)_2 = (54)_{10}$

फ्रॅक्शनल बायनरी संख्येचे डेसिमल संख्येमध्ये रूपांतर- (Conversion of Fractional Binary Number to Decimal Number-)

(Ex- $(1010.01)_2$

डेसिमल संख्या = $1\times2^3+0x2^2+1\times2^1+0x2^0+0x2^{-1}+1\times2^{-2}$

= 1x8+0x4+1x2+0x1+0x0.5+1x0.25

= 8+0+2+0+0+0.25

उत्तर: $(1010.01)_2 = (10.25)_{10}$

ii) ऑक्टल ते डेसिमल (Octal to Binary)

Ex- $(536)_8$ डेसिमल संख्येमध्ये रूपांतरित करा.

दिलेल्या ऑक्टल संख्येचे डेसिमल संख्येत रूपांतर करण्यासाठी, खाली दर्शविल्याप्रमाणे बेस (8) च्या वाढत्या पॉवर्ससह ऑक्टल संख्येच्या प्रत्येक अंकाचा गुणाकार करा आणि बेरीज करा-

डेसिमल संख्या = $5x8^2 + 3x8^1 + 6x8^0$

= 5x64 + 3x8 + 6x1

= 320 + 9 + 6

= 335

त्यामुळे ऑक्टल$(536)_8$ चा डेसिमल इक्विव्हॅलेंट(Equivalent) $(335)_{10}$ आहे.

उत्तर: $(536)_8 = (335)_{10}$

फ्रॅक्शनल ऑक्टल नंबरचे डेसिमल नंबर मध्ये रूपांतर-(Conversion of Fractional Octal Number to Decimal Number-)

$(12.2)_8 = 1 \text{ x } 8^1 + \underline{2 \text{ x } 8^0} + 2 \text{ x } 8^{-1}$

= 1x8+2x1+2/8

= 8+2+0.25

= 10.25

उत्तर: $(12.2)_8 = (10.25)_{10}$

iii) हेक्साडेसिमल ते डेसिमल (Hexadecimal to Decimal)

Ex- $(2C4)_{16}$ डेसिमल संख्येमध्ये रूपांतरित करा.

दिलेल्या हेक्साडेसिमल संख्येचे डेसिमल संख्येत रूपांतर करण्यासाठी, खाली दर्शविल्याप्रमाणे हेक्साडेसिमल संख्येच्या प्रत्येक अंकाचा, बेस (16) च्या वाढत्या पॉवर्ससह गुणाकार करा आणि बेरीज करा-

डेसिमल संख्या

= 2×162 + Cx161 + 4×160

= 2x256 + 12x16 + 4x1

$= (708)_{10}$

त्यामुळे हेक्साडेसिमल क्रमांक $(2C4)_{16}$ चा डेसिमल इक्विव्हॅलेंट(Equivalent) $(708)_{10}$ आहे.

उत्तर: $(2C4)_{16} = (708)_{10}$.

फ्रॅक्शनल हेक्साडेसिमल नंबरचे डेसिमल संख्येमध्ये रूपांतर- (Conversion of Fractional Hexadecimal Number to Decimal Number-)

$(2CA.52)_{16} = 2 \times 16^2 + 12 \times 16^1 + 10 \times 16^0 + 5 \times 16^{-1} + 2 \times 16^{-2}$

$= 2 \times 256 + 12 \times 16 + 10 \times 1 + 5/16 + 2/256$

$= 512 + 192 + 10 + 0.313 + 0.008$

उत्तर: $(2CA.52)_{16} = (714.321)_{10}$

1.1.5 b डेसिमल संख्येचे इतर संख्या प्रणाली मध्ये रूपांतर (Conversion of Decimal Number to Other Number System)

डेसिमल संख्या इतर संख्या प्रणालीमध्ये रूपांतरित करण्यासाठी रूपांतरण नियम

- पहिली पायरी(First step) म्हणजे ज्या संख्या प्रणालीमध्ये आपल्याला दिलेल्या डेसिमल संख्येचे रूपांतर करायचे आहे त्या प्रणालीचा आधार/ बेस ओळखणे.
- दुसरी पायरी (Second step) म्हणजे दिलेल्या डेसिमल संख्येला पहिल्या पायरीत मिळालेल्या बेससह भागाकार करणे आणि प्रत्येक विभागातील भागांक आणि बाकी नोंदवणे.
- तिसरी पायरी(Third step) म्हणजे आपल्याला मिळालेले अंक उलट क्रमाने लिहिणे म्हणजे पहिले मूल्य LSB आणि शेवटचे मूल्य MSB असे लिहिणे.

i) डेसिमल संख्येचे बायनरी संख्या प्रणाली मध्ये रूपांतर (Conversion of Decimal number to Binary Number System)

उदाहरण- डेसिमल संख्या $(29)_{10}$ ला बायनरी संख्येमध्ये रूपांतरित करा

दिलेल्या डेसिमल संख्येला बेस 2 ने भागून वरील पायऱ्या वापरा.

Base	Decimal no.	Reminder	
2	29	1	LSB
2	14	0	↑ उलट क्रम(Reverse order)
2	7	1	
2	3	1	
2	1	1	MSB
	0		

उत्तर: **$(29)_{10} = (11101)_2$**

फ्रॅक्शनल डेसिमल नंबरचे बायनरी नंबर सिस्टममध्ये रूपांतर **(Conversion of Fractional Decimal Number to Binary Number system)**

उदा. $(29.273)_{10} = (?)_2$

पूर्णांक भागाचे रूपांतरण **(Conversion of integer part)**

Base	Decimal no.	Reminder	
2	29	1	LSB
2	14	0	↑ उलट क्रम (Reverse order)
2	7	1	
2	3	1	
2	1	1	MSB
	0		

फ्रॅक्शनल भागाचे रूपांतरण – **(Conversion of fractional part)**

$(0.273)_{10}$ =	0.273 x 2=0.546	0	MSB
	0.546 x 2=1.092	1	↓
	0.092 x 2=0.184	0	
	0.184 x 2=0.368	0	LSB

उत्तर: **$(29.273)_{10} = (11101.0100)_2$**

ii) डेसिमल नंबरचे ऑक्टल नंबर सिस्टममध्ये रूपांतर (Conversion of Decimal number to Octal Number System)

उदाहरण- डेसिमल संख्या $(29)_{10}$ चे ऑक्टल संख्येत रूपांतर करा

वरील चरणांचा(Steps) वापर करून दिलेल्या डेसिमल संख्येला बेस 8 ने भागणे.

8	29	5	LSB ↑
8	3	3	उलट क्रम(Reverse order)
	0		MSB

उत्तर: : $(29)_{10} = (35)_8$

फ्रॅक्शनल डेसिमल नंबरचे ऑक्टल नंबर सिस्टममध्ये रूपांतर **(Conversion of Fractional Decimal Number to Octal Number system)**

उदा. $(29.652)_{10} = (?)_8$

पूर्णांक भागाचे रूपांतरण (**Conversion of integer part)**

8	29	5	LSB ↑
8	3	3	उलट क्रम
9			MSB

फ्रॅक्शनल भागाचे रूपांतरण – **(Conversion of fractional part)**

0.652 x 8=5.216	5	MSB ↓
0.216 x 8= 1.728	1	
0.728 x 8= 5.824	5	LSB

उत्तर: **$(29.652)_{10}= (35.515)_8$**

iii) डेसिमल संख्येचे हेक्साडेसिमल नंबर सिस्टममध्ये रूपांतर (Conversion of Decimal number to Hexadecimal Number System)

उदाहरण- दिलेल्या डेसिमल संख्येचे $(461)_{10}$ हेक्साडेसिमल संख्येत रूपांतर करा

$(461)_{10} = (?)_{16}$

वरील पायऱ्या(steps) वापरून दिलेल्या दशांश(Decimal) संख्येला बेस 16 ने भागणे.

Base	Decimal no.	Reminder	
16	461	13=D	LSB ↓
16	28	12=C	उलट क्रम
16	1	1	MSB
	0		

उत्तर: **$(461)_{10} = (1CD)_{16}$**

फ्रॅक्शनल डेसिमल नंबरचे हेक्साडेसिमल नंबर सिस्टममध्ये रूपांतर **(Conversion of Fractional Decimal Number to Hexadecimal Number system)**

$(461.122)_{10} = (?)_{16}$

पूर्णांक भागाचे रूपांतरण (**Conversion of integer part)**

Base	Decimal no.	Reminder	
16	461	13=D	LSB ↑
16	28	12=C	उलट क्रम
16	1	1	MSB
	0		

फ्रॅक्शनल भागाचे रूपांतरण – **(Conversion of fractional part)**

0.122 x 16=1.952	1	MSB ↓
0.952 x 16=15.232	15=F	
0.232 x 16=3.712	3	LSB

उत्तर: **$(461.122)_{10} = (1CD.1F3)_{16}$**

1.1.5 c कोणत्याही नंबर सिस्टम चे डेसिमल व्यतिरिक्त इतर कोणत्याही नंबर सिस्टममध्ये रूपांतरण (Conversion of any number system to any number system other than Decimal)

i) ऑक्टल नंबरचे बायनरी नंबर सिस्टममध्ये रूपांतर – (Conversion of Octal number to binary number) उदा. $(345)_8 = (?)_2$

खाली दाखवल्याप्रमाणे ऑक्टल नंबरच्या(Octal number) प्रत्येक अंकाच्या जागी संबंधित ३ बिट बायनरी संख्या (3 bit binary number) लिहा.

Octal =	**3**	**4**	**5**	
Binary =	011	100	101	= 011100101

उत्तर: $\mathbf{(345)_8 = (011100101)_2}$

फ्रॅक्शनल ऑक्टल नंबरचे बायनरी नंबर सिस्टममध्ये रूपांतर – (Conversion of Fractional Octal to Binary)

$(627.01)_8 = (?)_2$

Octal =	**6**	**2**	**7**	**.**	**0**	**1**
Binary =	110	010	111	.	000	001

उत्तर: $\mathbf{(627.01)_8 = (110010111.000001)_2}$

ii) **हेक्साडेसिमल नंबरचे बायनरी नंबर सिस्टममध्ये रूपांतर - (Conversion of Hexadecimal number to Binary number)**

उदा. $(3B29)_{16} = (?)_2$

खाली दाखवल्याप्रमाणे हेक्साडेसिमल संख्येच्या प्रत्येक अंकाच्या जागी इक्विव्हॅलेंट(Equivalent) 4 बिट बायनरी संख्या लिहा.

3 = 0011
B = 1011
2 = 0010
9 = 1001

हेक्साडेसिमल संख्या $(3B29)_{16}$ चे बायनरी इक्विव्हॅलेंट(equivalent) $(0011101100101001)_2$ आहे.

उत्तर: $(3B29)_{16} = (0011101100101001)_2$

फ्रॅक्शनल हेक्साडेसिमल नंबरचे बायनरी नंबर सिस्टममध्ये रूपांतर – (Fractional Hexadecimal to Binary)

उदा. $(D54.CA)_{16} = (?)_2$

Hexadecimal	**D**	**5**	**4**	**.**	**C**	**A**
Binary	1101	0101	0100	.	1100	1010

उत्तर: $\mathbf{(D54.CA)_{16} = (110101010100.11001010)_2}$

iii) **बायनरी नंबरचे ऑक्टल नंबर सिस्टममध्ये रूपांतर** –(**Conversion of Binary number to Octal number)**

उदा. $(111001101010\ 011)_2 = (?)_8$

- वरील बायनरी संख्येमध्ये उजवीकडून डावीकडे प्रत्येकी ३ बिट्सचा गट बनवा.
- जर बिट्सची संख्या कमी असेल तर डाव्या बाजूला 0 जोडा.
- प्रत्येक गटासाठी इक्विव्हॅलेंट(Equivalent) ऑक्टल नंबर लिहा.

111 001 101 010 011
7 1 5 2 3

उत्तर: $\mathbf{(111001101010011)_2 = (71523)_8}$

फ्रॅक्शनल बायनरी नंबरचे ऑक्टल नंबर सिस्टममध्ये रूपांतर –(Conversion of Fractional Binary number to Octal number)

उदा. $(10100.1101)_2 = (?)_8$

010 100 . 110 100
2 4 . 6 4

उत्तर: **$(10100.1101)_2 = (24.64)_8$**

iv) **बायनरी नंबरचे हेक्साडेसिमल नंबर सिस्टममध्ये रूपांतर – (Conversion of Binary number to Hexadecimal number)**

उदा. $(1011001101011010)_2 = (?)_{16}$

- वरील बायनरी संख्येमध्ये उजवीकडून डावीकडे प्रत्येकी 4 बिट्सचा गट बनवा.
- जर बिट्सची संख्या कमी असेल तर डाव्या बाजूला 0 जोडा.
- प्रत्येक गटासाठी इक्विव्हॅलेंट(Equivalent) हेक्साडेसिमल नंबर लिहा.

(1011 0011 0101 1010)
B 3 5 A

उत्तर: **$(1011001101011010)_2 = (B35A)_{16}$**

फ्रॅक्शनल बायनरी नंबरचे हेक्साडेसिमल नंबर सिस्टममध्ये रूपांतर –(**Conversion of fractional Binary number to Hexadecimal number)**

उदा. $(10100.1101)_2 = (?)_{16}$

0001 0100 . 1101
1 4 . D

उत्तर: **$(10100.1101)_2 = (14.D)_{16}$**

v) **ऑक्टल नंबरचे हेक्साडेसिमल नंबर सिस्टममध्ये रूपांतर –(Conversion of Octal number to Hexadecimal number)**

हे रूपांतरण करण्याचे 2 मार्ग आहेत-

1. ऑक्टल - बायनरी - हेक्साडेसिमल (Octal –Binary- Hexadecimal)
2. ऑक्टल – डेसिमल - हेक्साडेसिमल (Octal –Decimal- Hexadecimal)

Ex. $(345)_8 = (?)_{16}$

प्रथम दिलेल्या ऑक्टल संख्येचे बायनरी संख्येत रूपांतर करा.

345 = 011 100 101

आता बायनरी संख्या हेक्साडेसिमल संख्येत रूपांतरित करा-

0000 1110 0101
0 E 5

उत्तर**: $(345)_8 = (0\,E5)_{16}$**

फ्रॅक्शनल ऑक्टल नंबरचे हेक्साडेसिमल नंबर सिस्टममध्ये रूपांतर –(**Conversion of Fractional Octal number to Hexadecimal number)**

Ex. $(752.41)_8 = (?)_{16}$

प्रथम दिलेल्या ऑक्टल संख्येचे बायनरी संख्येत रूपांतर करा.

752.41=111 101 010 .100 001

आता बायनरी संख्या हेक्साडेसिमल संख्येत रूपांतरित करा-

0001 1110 1010 . 1000 0100
1 E A . 8 4

उत्तर**: $(752.41)_8 = (1EA.84)_{16}$**

vi) **हेक्साडेसिमल नंबरचे ऑक्टल नंबर सिस्टममध्ये रूपांतर – (Conversion of Hexadecimal number to Octal number)**

हे रूपांतरण करण्याचे 2 मार्ग आहेत-

1. हेक्साडेसिमल - बायनरी - ऑक्टल (Hexadecimal -Binary – Octal)
2. हेक्साडेसिमल – डेसिमल - ऑक्टल (Hexadecimal -Decimal – Octal)

Ex. $(9D3)_{16} = (?)_8$

प्रथम हेक्साडेसिमलला बायनरीमध्ये रूपांतरित करा-

9D3= 1001 1101 0011

आता बायनरीला ऑक्टलमध्ये रूपांतरित करा.

100 111 010 011
4 7 2 3

उत्तर**: $(9D3)_{16} = (4723)_8$**

फ्रॅक्शनल हेक्साडेसिमल नंबरचे ऑक्टल नंबर सिस्टममध्ये रूपांतर – (**Conversion of fractional Hexadecimal number to Octal number)**

Ex. $(5B.4A)_{16} = (?)_8$

प्रथम दिलेल्या हेक्साडेसिमल संख्येचे बायनरी संख्येत रूपांतर करा-

5B.4A= 0101 1011 . 0100 1010

आता बायनरीला ऑक्टलमध्ये रूपांतरित करा.

001 011 011 . 010 010 100
1 3 3 . 2 2 4

उत्तर**: $(5B.4A)_{16} = (133.224)_8$**

1.2 बायनरी अरीथमॅटिक : - बेरीज, वजाबाकी (1's आणि 2's कॉम्पलिमेन्ट्स) गुणाकार, भागाकार. बीसीडी बेरीज (Binary Arithmetic: - Addition, Subtraction (1's and 2's complement) Multiplication, Division. BCD addition)

1.2.1 बायनरी अडिशनचे नियम –(**Binary Addition Rules)**

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0 कॅरी = 1
1 + 1 + 1 = 1 कॅरी = 1

सोडवलेली उदाहरणे-

1. $(11011)_2 + (10011)_2$

```
               11
Answer      11011     27
          + 10011     19
           101110     46
```

2. $(101011)_2 + (110101)_2$

```
             11111
Answer      101011     43
          + 110101     53
           1100000     96
```

3. $(111.01)_2 + (100.11)_2 = (1100.00)_2$

```
Answer   111.01
       + 100.11
        1100.00
```

1.2.2 बायनरी वजाबाकीचे नियम- (Binary Subtraction Rules)

0 - 0 = 0
1 - 0 = 1
1 - 1 = 0
0 – 1 = 1 बॉरो 1

Solved Examples-

1. $(11101)_2 - (10010)_2$

```
        01
     11101      29
-    10010    - 18
     01011      11
```

2. $(11011)_2 - (11000)_2$

```
    11011      27
 -  11000    - 24
    00011      03
```

1.2.2 a 1's आणि 2's कॉम्पलिमेन्ट्स चा वापर करून बायनरी वजाबाकी – (Subtraction using 1's and 2's complement method-)

1. 1's कॉम्पलिमेन्ट्(Complement) म्हणजे बायनरी संख्येमध्ये सर्व 0 ला 1 ने आणि 1 ला 0 ने बदलून मिळवलेली संख्या होय.

उदाहरणार्थ $(11010011)_2$ या बायनरी नंबरचा 1's कॉम्पलिमेन्ट्(Complement) 00101100 आहे.

2. 2's कॉम्पलिमेन्ट् म्हणजे बायनरी संख्येच्या 1's कॉम्पलिमेन्ट्(Complement) मध्ये 1 (add) ऍड करून मिळवलेली संख्या होय.

उदाहरणार्थ $(11010011)_2$

```
   00101100     1's कॉम्पलिमेन्ट्
 +        1
   00101101     2's कॉम्पलिमेन्ट्
```

खालील उदाहरणांसाठी 1's and 2's कॉम्पलिमेन्ट्(Complement) शोधा.

1. 1000 — Ans. 0111 1000

2. 1010 — Ans. 0101 0110

1's कॉम्पलिमेन्ट्(Complement) पद्धतीने वजाबाकी करणे-

1's कॉम्पलिमेन्ट्(Complement) पद्धतीने वजाबाकी करण्यासाठी खालील पायऱ्या वापराव्या लागतात: (A-B)

i) B चा 1's कॉम्पलिमेन्ट्(Complement) शोधा.

ii) बी च्या 1's कॉम्प्लिमेंट (Complement) ला A मध्ये ऍड करा

iii) जर बेरजेच्या रिझल्ट मध्ये कॅरी 1 असेल तर तो रिझल्ट मध्ये ऍड करा. शेवटचे उत्तर +ve आहे.

iv) जर कॅरी नसेल किंवा तो 0 असेल तर रिझल्ट- ve आहे म्हणून बेरीजच्या रिझल्ट चा 1's कॉम्प्लिमेंट (Complement) काढा ते फायनल उत्तर असेल.

सोडवलेली उदाहरणे

1. $(1101)_2 - (1000)_2$

1000 चा 1's कॉम्प्लिमेंट(Complement) 0111 आहे.

पहिला नंबर आणि दुसऱ्या नंबर चा 1's कॉम्प्लिमेंट यांची बेरीज करा

```
                    पहिला नंबर -     1 1 0 1
 दुसऱ्या नंबर चा 1's कॉम्प्लिमेंट -   + 0 1 1 1
                  कॅरी ओव्हर -   1  0 1 0 0
                                  +      1
                                   0 1 0 1
```

आवश्यक फरक 0101 आहे, तो +ve आहे

वरील उदाहरणात संख्या आहेत 13 - 8 = 5

बायनरीमध्ये देखील रिझल्ट 0101 म्हणजेच 5 आहे, तो +ve आहे.

2. 110101 – 100101 (53 – 37)

100101 चा 1's कॉम्प्लिमेंट(Complement) 011010 आहे.

पहिला नंबर आणि दुसऱ्या नंबर चा 1's कॉम्प्लिमेंट यांची बेरीज करा

```
                    पहिला नंबर -     1 1 0 1 0 1
 दुसऱ्या नंबर चा 1's कॉम्प्लिमेंट     + 0 1 1 0 1 0
                  कॅरी ओव्हर -   1  0 0 1 1 1 1
                                  +          1
                                   0 1 0 0 0 0
```

आवश्यक फरक 10000 आहे, तो +ve आहे

वरील उदाहरणात संख्या आहेत 53 - 37 = 16

बायनरीमध्ये देखील रिझल्ट 10000 म्हणजेच 16 आहे, तो +ve आहे.

3. 1000 - 1101 (8 – 13)

1101 चा 1's कॉम्प्लिमेंट(Complement) 0010 आहे.

पहिला नंबर - 1 0 0 0
दुसऱ्या नंबर चा 1's कॉम्प्लिमेंट - + 0 0 1 0
कॅरी ओव्हर - 0 1 0 1 0

कोणताही कॅरी जनरेटेड नसल्यामुळे रिजल्ट –ve आहे आणि कॉम्प्लिमेंटेड स्वरूपात आहे, खरे उत्तर मिळविण्यासाठी रिजल्टचा 1's कॉम्प्लिमेंट घ्या.

आवश्यक फरक 0101 आहे, तो -ve आहे

वरील उदाहरणात संख्या आहेत 8 - 13 = -5

बायनरीमध्ये देखील रिझल्ट 0101 म्हणजेच 5 आहे, तो -ve आहे.

4. 10000 – 0100 i.e. (16-04)

1's कॉम्प्लिमेंट वजाबाकीमध्ये दोन्ही संख्यांसाठी बिट्सची संख्या समान असणे आवश्यक आहे.

00100 1's कॉम्प्लिमेंट 11011 आहे

Minuend 10000 16
+ 11011 1's कॉम्प्लिमेंट
1 01011
+ → 1
01100 12

आवश्यक फरक 01100 आहे, तो +ve आहे

वरील उदाहरणात संख्या आहेत 16 - 4 = 12

बायनरीमध्ये देखील रिझल्ट 01100 म्हणजेच 12 आहे, तो +ve आहे.

2's कॉम्पलिमेन्ट् (Complement) पद्धतीने वजाबाकी करणे- (Subtraction using 2's complement method)

2's कॉम्पलिमेन्ट् पद्धतीने वजाबाकी करण्यासाठी खालील पायऱ्या(Steps) वापराव्या लागतात:

(A-B)

i) B चा 2's कॉम्पलिमेन्ट् काढा .

ii) B च्या 2's कॉम्प्लिमेंट ला A मध्ये ॲड करा

iii) जर बेरजेच्या रिझल्ट मध्ये कॅरी 1 असेल तर तो डिसकार्ड करा. शेवटचे उत्तर +ve आहे.

iv) जर कॅरी नसेल किंवा तो ० असेल तर रिझल्ट - ve आहे आणि कॉम्प्लिमेंटेड स्वरूपात आहे म्हणून बेरीजच्या रिझल्टचा 2's कॉम्प्लिमेंट काढा ते फायनल उत्तर असेल. सोडवलेली उदाहरणे

1. $(1101)_2 – (1000)_2$

1000 चा 2's कॉम्प्लिमेंट 1000 आहे.

पहिला नंबर आणि दुसऱ्या नंबर चा 2's कॉम्प्लिमेंट यांची बेरीज करा

पहिला नंबर - 1 1 0 1
दुसऱ्या नंबर चा 2's कॉम्प्लिमेंट - + 1 0 0 0
कॅरी ओव्हर - ~~1~~ 0 1 0 1
कॅरी डिसकार्ड

आवश्यक फरक 0101 आहे, तो +ve आहे

वरील उदाहरणात संख्या आहेत 13 - 8 = 5

बायनरीमध्ये देखील निकाल 0101 म्हणजेच 5 आहे, तो +ve आहे.

2. $(10110)_2 – (11010)_2$

11010 चा 2's कॉम्प्लिमेंट 00110 आहे.

पहिला नंबर आणि दुसऱ्या नंबर चा 2's कॉम्प्लिमेंट यांची बेरीज करा

पहिला नंबर -	1 0 1 1 0
दुसऱ्या नंबर चा 2's कॉम्प्लिमेंट -	+ 0 0 1 1 0
Result of addition -	0 1 1 1 0 0

कोणताही कॅरी जनरेटेड नसल्यामुळे रिजल्ट –ve आहे आणि कॉम्प्लिमेंटेड स्वरूपात आहे, खरे उत्तर मिळविण्यासाठी रिजल्टचा 2's कॉम्प्लिमेंट घ्या.

11100 2's कॉम्प्लिमेंट i.e. (00011 + 1) or 00100.

आवश्यक फरक 100 आहे, तो -ve आहे

वरील उदाहरणात संख्या आहेत 22 - 26 = - 4

बायनरीमध्ये(Binary) देखील निकाल 100 म्हणजेच 4 आहे, तो -ve आहे.

1.2.3 बायनरी गुणाकार चे नियम- (Binary Multiplication Rules)

0 x 0 = 0
1 x 0 = 0
0 x 1 = 0
1 x 1 = 1

```
1.   101      5
   x 010    x 2
     000
    101x
   000xx
   01010     10
```

Ans: (1010)

1.2.4 बायनरी भागाकार-(Binary Division)

Ex. 1100/11

```
      100
11 | 1100
     11↓
     000↓
     0000
     0000
```

1.2.5 BCD कोड आणि BCD बेरीज- (BCD Code & BCD Addition)

- BCD ही बायनरी कोडेड डेसिमल संख्या आहे.
- BCD कोडमध्ये डेसिमल संख्येचा प्रत्येक अंक त्याच्या इक्विव्हॅलेंट(equivalent) 4 बिट बायनरी संख्येने दर्शविला जातो.
- BCD कोड फक्त 0 ते 9 पर्यंतच्या डेसिमल (Decimal)अंकांसाठीच व्हॅलीड(valid) आहे.

उदाहरणार्थ, 10 चा BCD कोड हा आहे-

```
 1    0
0001 0000
```

उदाहरणार्थ, 24 चा BCD कोड हा आहे-

```
 2    4
0010 0100
```

उदाहरणार्थ, 973 चा BCD कोड हा आहे-

```
 9    7    3
1001 0111 0011
```

Binary Code A B C D	Decimal Number	BCD Code B_5	B_4	B_3	B_2	B_1
0 0 0 0	0	0	0	0	0	0
0 0 0 1	1	0	0	0	0	1
0 0 1 0	2	0	0	0	1	0
0 0 1 1	3	0	0	0	1	1
0 1 0 0	4	0	0	1	0	0
0 1 0 1	5	0	0	1	0	1
0 1 1 0	6	0	0	1	1	0
0 1 1 1	7	0	0	1	1	1
1 0 0 0	8	0	1	0	0	0
1 0 0 1	9	0	1	0	0	1
1 0 1 0	10	1	0	0	0	0
1 0 1 1	11	1	0	0	0	1
1 1 0 0	12	1	0	0	1	0
1 1 0 1	13	1	0	0	1	1
1 1 1 0	14	1	0	1	0	0
1 1 1 1	15	1	0	1	0	1

Table no 1.3 Chart representing BCD and Binary codes of Decimal numbers

BCD बेरीज करण्याचे नियम **(Rules for BCD Addition)**

1. दिलेल्या डेसिमल संख्यांना त्यांच्या BCD समतुल्य(Equivalent) मध्ये रूपांतरित करा
2. बायनरी बेरीज नियमांनुसार दोन BCD नंबरची बेरीज करा.
3. प्रकार 1 - जर 4 बिट बेरीज <= 9 असेल तर तो व्हॅलीड(Valid) BCD आहे
4. प्रकार 2 - जर 4 बिट बेरीज > ९ असेल तर तो इनव्हॅलीड(Invalid) BCD आहे.
5. योग्य उत्तर मिळविण्यासाठी 4 बिट बेरजमध्ये सुधारणा घटक $6=(0110)_2$ ऍड करा.
6. प्रकार ३ - 4 बिट बेरीज <= 9 आहे परंतु कॅरी जनरेट झाला आहे, तर तो इनव्हॅलीड(Invalid) BCD आहे.
7. योग्य उत्तर मिळविण्यासाठी 4 बिट बेरजमध्ये सुधारणा घटक $6=(0110)_2$ ऍड करा.

BCD बेरीजची उदाहरणे: - **(Examples of BCD addition)**

1. प्रकार 1 - जर 4 बिट बेरीज <= 9 आहे

(3) + (4) = (?)

संख्यांचे त्यांच्या संबंधित BCD कोडमध्ये रूपांतर करा

3 = 0011 आणि 4 = 0100

बायनरी बेरजे सारखी दोन BCD नंबरची बेरीज करा-

```
  0011
+ 0100
  0111   व्हॅलीड( Valid )  BCD
```

वरील उदाहरणातील 4 बिट बायनरी बेरीज < 9 असल्याने निकाल व्हॅलीड(Valid) BCD आहे.

तर बरोबर उत्तर 3+4 = 7 आहे.

2. प्रकार 2 - जर 4 बिट बेरीज > 9 आहे

(5) + (6) = (?)

संख्यांचे त्यांच्या संबंधित BCD कोडमध्ये रूपांतर करा

5 = 0101 आणि 6 = 0110

बायनरी बेरजे सारखी दोन BCD नंबरची बेरीज करा-

```
  0101
+ 0110
  1011   इनव्हॅलीड( Invalid )   BCD
```

या उदाहरणातील 4 बिट बेरीज >9 असल्याने निकाल इनव्हॅलीड(Invalid) BCD आहे, म्हणून सुधारणा घटक $6=(0110)_2$ ऍड करावा लागेल.

```
  11        Carry
  1011
+ 0110
1 0001  ←  व्हॅलीड( Valid )  BCD Result (11)
```

तर बरोबर उत्तर आहे 5 + 6 = 11

3. प्रकार 3 - 4 बिट बेरीज < = 9 आहे परंतु कॅरी जनरेट झाला आहे

(8) + (9) = (?)

संख्यांचे त्यांच्या संबंधित BCD कोडमध्ये रूपांतर करा

8 = 1000 आणि 9 = 1001

बायनरी बेरजे सारखी दोन BCD नंबरची बेरीज करा-

```
        1001
      + 1000
Carry  1 0001  इनव्हॅलीड  BCD
```

या उदाहरणातील 4 बिट बेरीज <= 9 आहे परंतु कॅरी जनरेट झाला असल्याने निकाल इनव्हॅलीड(Invalid) BCD आहे, म्हणून सुधारणा घटक $6=(0110)_2$ ऍड करावा लागेल.

```
  0001 0001
+      0110
  0001 0111
    1    7   ←  इनव्हॅलीड( Invalid ) BCD Result (17)
```

तर बरोबर उत्तर आहे 8 + 9 = 17

उदाहरण (184) + (576)

संख्यांचे त्यांच्या संबंधित BCD कोडमध्ये रूपांतर करा

बायनरी बेरजे सारखी दोन BCD नंबरची बेरीज करा-

```
  0001 1000  0100
+ 0101 0111  0110
  0110 1111  1010     इनव्हॅलीड( Invalid )  BCD
     1 1111  11       Carry
  0110 1111  1010
+      0110  0110     सुधारणा घटक 6=(0110)2 ऍड करा
  0111 0110  0000
    7    6     0
```

1.3 लॉजिक गेट्स: सिम्बॉल, स्विच सर्किट, लॉजिकल एक्सप्रेशन, बेसिक लॉजिक गेट्स (AND, OR, NOT), युनिव्हर्सल गेट्स (NAND आणि NOR) आणि स्पेशल पर्पज गेट्स (EX-OR, EX-NOR) यांचे ट्रुथ टेबल. (Logic Gates: Symbol, switch circuit, logical expression, truth table of basic logic gates (AND, OR, NOT), Universal gates (NAND and NOR) and Special purpose gates (EX-OR, EX-NOR)).

1.3.1 तीन प्रकारचे मूलभूत लॉजिक गेट्स आहेत –

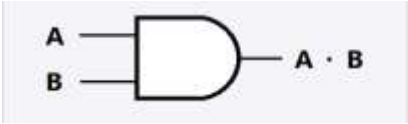
1. अँड गेट (AND Gate)
2. ऑर गेट (OR Gate)
3. नॉट गेट (NOT Gate)

1. अँड गेट (AND GATE)-

AND गेटचा वापर बायनरी इनपुटचा लॉजिकल गुणाकार करण्यासाठी केला जातो. जर दोन्ही इनपुट हाय (1) असतील तर AND गेटची आउटपुट स्थिती हाय (1) असेल, अन्यथा जर कोणताही इनपुट लो (0) असेल तर आउटपुट स्थिती लो (0) असेल.

AND गेटसाठी बुलियन एक्सप्रेशन किंवा लॉजिक म्हणजे इनपुटचा लॉजिकल गुणाकार जो खाली दिल्याप्रमाणे पूर्णविराम किंवा सिंगल डॉटने (.)दर्शविला जातो.

Figure 1.1 Symbol of AND Gate



बुलियन एक्स्प्रेशन -

Y = A . B

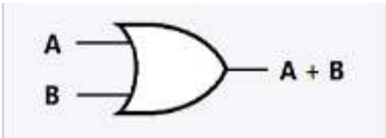
Table no 1.4 Truth Table

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

2. ऑर गेट (OR Gate)

ऑर गेट हा सर्वात जास्त वापरला जाणारा डिजिटल लॉजिक सर्किट आहे आणि डिजिटल लॉजिकमध्ये तो प्राथमिक डिजिटल इलेक्ट्रॉनिक्स बिल्डिंग ब्लॉक म्हणून ओळखला जातो. ऑर गेटचे दोन्ही किंवा एक जरी इनपुट हाय (1)असेल तर ऑर गेटचे आउटपुट हाय म्हणजेच 1 असते. सगळे इनपुट 0 असतील तरच आउटपुट 0 असते. OR गेटसाठी बुलियन एक्सप्रेशन म्हणजे इनपुटची बेरीज जी अधिक चिन्हाने (+) दर्शविली जाते.

Figure 1.2 Symbol of OR Gate



बुलियन एक्स्प्रेशन –

Y = A + B

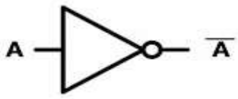
Table no 1.5 Truth Table

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

3. नॉट गेट (NOT Gate)

नॉट गेट, ज्याला इन्व्हर्टर देखील म्हणतात, त्यात फक्त एक इनपुट आणि एक आउटपुट असतो. हा एक लॉजिक गेट आहे ज्याचे आउटपुट नेहमीच त्याच्या इनपुटचे कॉम्प्लिमेंट(Complement) असते.

Figure 1.3 Symbol of NOT Gate



बुलियन एक्स्प्रेशन –

$\mathbf{Y = \overline{A}}$

Table no 1.6 Truth Table

A	Y
0	1
1	0

1.3.2 लॉजिक गेट्सचे स्विच सर्किट-(Switch Circuit of Logic gates)

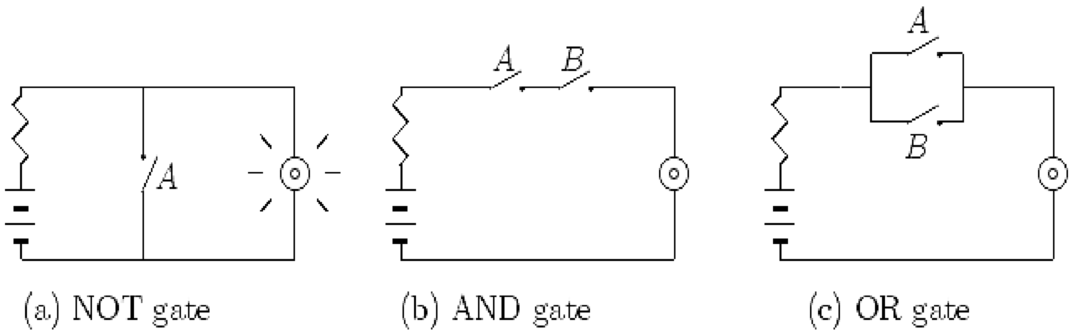


Figure 1.4 Switch Circuit of Logic gates

1.3.3 युनिव्हर्सल गेट्स - नँड आणि नॉर गेट्स (Universal Gates - NAND and NOR Gates)

युनिव्हर्सल गेट हे एक लॉजिक गेट आहे जे इतर कोणत्याही प्रकारच्या गेटची आवश्यकता न पडता कोणतेही बुलियन फंक्शन करू शकतो. नँड (NAND) आणि नॉर (NOR) गेट्स हे युनिव्हर्सल गेट्स आहेत. अँड, ऑर, नॉट, एक्सऑर आणि एक्सनॉर (AND, OR, NOT, XOR and XNOR) हे बेसिक बुलियन फंक्शन्स मिळवण्यासाठी, युनिव्हर्सल गेट्सचे इनपुट व आउटपुट हे वेगवेगळ्या प्रकारांनी एकमेकांशी जोडले जाऊ शकतात.

i) नँड गेट (NAND Gate) -

नँड (NAND) गेट हे अँड आणि नॉट (AND आणि NOT) गेटचे कॉम्बिनेशन आहे. अँड (AND) आणि त्यानंतर नॉट (NOT) गेट येते. जेव्हा त्याचे सर्व इनपुट हाय(1) (High) असतात फक्त तेव्हा त्याचे आउटपुट लो(0) (Low) असते; अन्यथा, ते 1 आउटपुट देते.

Figure 1.5 Symbol of NAND Gate

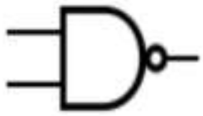


Table no 1.7 Truth Table

Input A	Input B	$Y = \overline{AB}$
0	0	1
0	1	1
1	0	1
1	1	0

बुलियन एक्स्प्रेशन

$Y = \overline{A.B}$

ii) नॉर गेट (NOR Gate)-

नॉर (NOR) गेट हे ऑर आणि नॉट (OR आणि NOT) गेटचे कॉम्बिनेशन आहे. ऑर (OR) आणि त्यानंतर नॉट (NOT) गेट येते. जेव्हा त्याचे सर्व इनपुट लो (Low) असतात फक्त तेव्हा त्याचे आउटपुट हाय (High) असते; अन्यथा, ते लो (Low) आउटपुट देते.

Figure 1.6 Symbol of NOR Gate

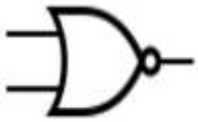


Table no 1.8 Truth Table

Input A	Input B	0 = (A + B)'
0	0	1
0	1	0
1	0	0
1	1	0

बुलियन एक्स्प्रेशन –

Y = (A +B)'

1.3.4 स्पेशल परपज गेट्स (Special purpose Gates-)

i) एक्सऑर (EX-OR) गेट-

याला "एक्सक्लुझिव्ह ऑर (Exclusive-OR) गेट " असे म्हणतात. एक्सऑर (EX-OR) गेट मध्ये जर इनपुट वेगवेगळे असतील तरच एक्सऑर गेटचे आउटपुट हाय किंवा ट्रु (True) असते. जर इनपुट सारखे असतील तर आउटपुट लो (Low) किंवा फॉल्स (False) असते.

उदाहरणार्थ, जर इनपुट 1 आणि 0 असतील तर आउटपुट हाय किंवा ट्रु (True) असेल (कारण अनुक्रमे ट्रु (True) आणि फॉल्स (False) इनपुटची संख्या विषम आहे). जर इनपुट 1 आणि 1 किंवा 0 आणि 0 असेल तर आउटपुट लो (Low) किंवा फॉल्स False असेल (कारण अनुक्रमे ट्रु (True) आणि फॉल्स (False) इनपुटची संख्या सम आहे).

Figure 1.7 Symbol of X-OR Gate

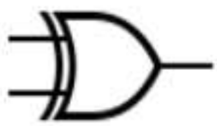


Table no 1.9 Truth Table

A (Input 1)	B (Input 2)	X = A'B + AB'
0	0	0
0	1	1
1	0	1
1	1	0

बुलियन एक्स्प्रेशन –

$A \oplus B$ **= AB' + A'B**

ii) एक्सनॉर (EX-NOR) गेट-

याला "एक्सक्लुझिव्हली नॉर (NOR) गेट" असे म्हणतात. एक्सनॉर (XNOR) गेट हा मुळात NOT गेटसह जोडलेला XOR गेट आहे. तो XOR गेटच्या अगदी विरुद्ध किंवा उलट आहे. या गेटमध्ये दोन्ही इनपुट व्हॅल्यूज समान (म्हणजे दोन्ही 0 किंवा दोन्ही 1) असल्यास आउटपुट हाय (HIGH) किंवा ट्रु (True) असते. अन्यथा आउटपुट लो (Low) किंवा फॉल्स (False) असते.

Figure 1.8 Symbol of X-NOR Gate

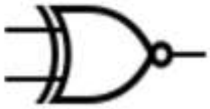


Table no 1.10 Truth Table

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

बुलियन एक्स्प्रेशन –

Y = A.B + A'.B'

1.3.4 a युनिव्हर्सल गेट्स वापरून बेसिक गेट्सची रचना –(Design of basic gates using Universal gates)

A) नँड (NAND) गेट वापरून बेसिक गेट्सची रचना

1. नँड (NAND) गेट वापरून नॉट (NOT) गेटची रचना **(Realization of NOT gate using NAND gate)**
   खालील आकृतीमध्ये दाखवल्याप्रमाणे एकच नँड (NAND) गेट वापरून नॉट (NOT) गेटची रचना केली जाते

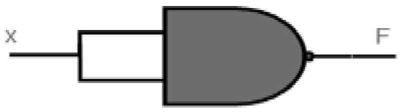


Figure 1.9 Realization of NOT gate using NAND gate

2. नँड (NAND) गेट वापरून अँड (AND) गेटची रचना **(Realization of AND gate using NAND gate)**
   खालील आकृतीमध्ये दाखवल्या प्रमाणे दोन नँड (NAND) गेट वापरून अँड (AND) गेटची रचना केली जाते

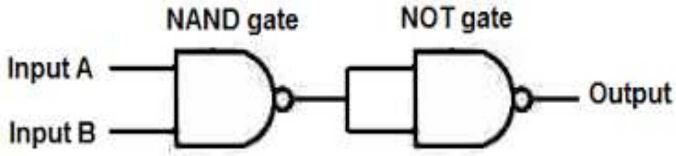


Figure 1.10 Realization of AND gate using 2 NAND Gates

3. नँड (NAND) गेट वापरून ऑर (OR) गेटची रचना **(Realization of OR gate using NAND gate)**
   खालील आकृतीमध्ये दाखवल्याप्रमाणे तीन नँड (NAND) गेट वापरून ऑर (OR) गेटची रचना केली जाते

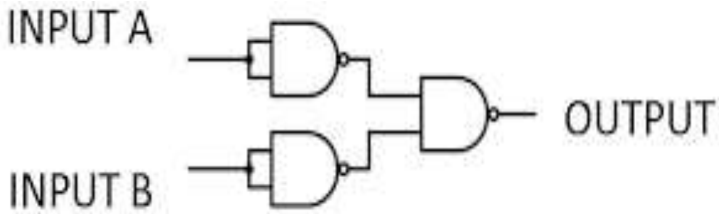


Figure 1.11 Realization of OR gate using 3 NAND Gates

B) नॉर (NOR) गेट वापरून बेसिक गेट्सची रचना (Realization of Basic gates using NOR gate)

1. नॉर (NOR) गेट वापरून नॉट (NOT) गेटची रचना **(Realization of NOT gate using NOR gate)**

खालील आकृतीमध्ये दाखवल्याप्रमाणे एकच नॉर (NOR) गेट वापरून नॉट (NOT) गेटची रचना केली जाते

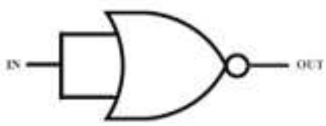


Figure 1.12 Realization of NOT gate using NOR gate

2. नॉर (NOR) गेट्स वापरून ऑर (OR) गेटची रचना **(Realization of OR gate using NOR Gate)**

खालील आकृतीमध्ये दाखवल्याप्रमाणे दोन नॉर (NOR) गेट्स वापरून ऑर (OR) गेटची रचना केली जाते

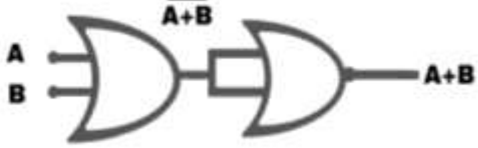


Figure 1.13 Realization of OR gate using NOR Gate

3. नॉर NOR गेट्स वापरून अँड (AND) गेटची रचना (**Realization of AND gate using NOR Gate)**
खालील आकृतीमध्ये दाखवल्याप्रमाणे तीन नॉर NOR गेट्सवापरून अँड (AND) गेटची रचना केली जाते

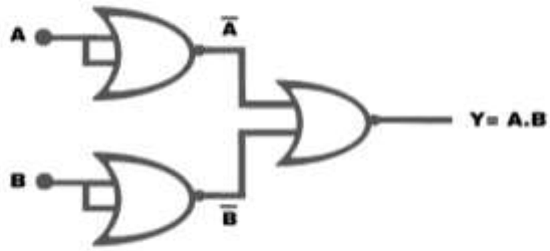


Figure 1.14 Realization of AND gate using NOR Gate

नँड (NAND) युनिव्हर्सल गेट्स वापरून बेसिक गेट्सची रचना – **(Implementation of all Basic gates using NAND gate)**

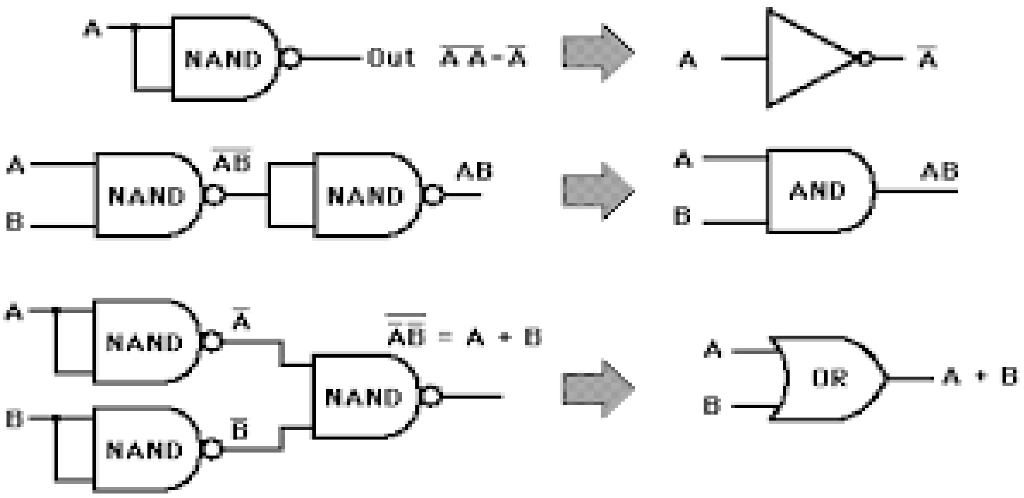


Figure 1.15 Implementation of all Basic gates using NAND gate

नॉर (NOR) युनिव्हर्सल गेट्स वापरून बेसिक गेट्सची रचना –**(Implementation of all Basic gates using NOR gate)**

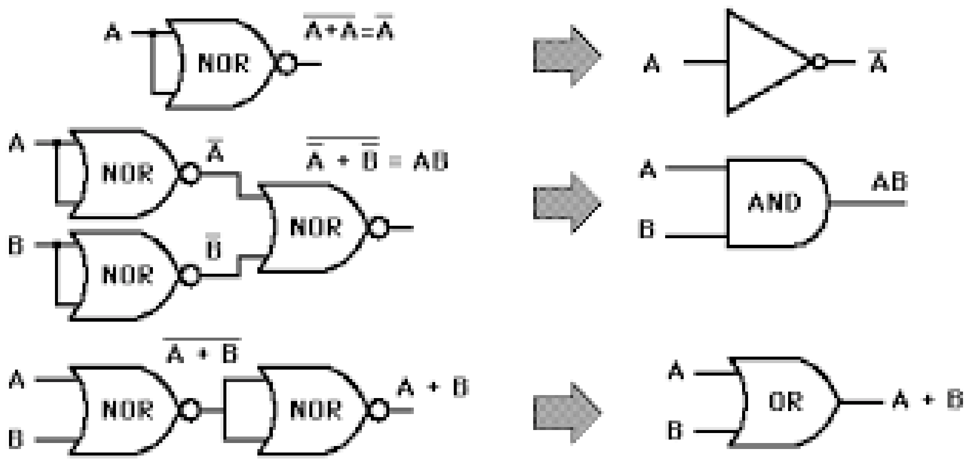


Figure 1.16 Implementation of all Basic gates using NOR gate

1.4 सर्किट्स: हाफ आणि फुल ॲडर, हाफ आणि फुल सबट्रक्टर ट्रुथ-टेबल बुलियन एक्सप्रेशन आणि लॉजिक गेट्स वापरून सर्किट्स.

(Arithmetic Circuits: Half and full Adder, Half and full subtractor with its truth table, Boolean expression and circuits using logic gates)

1.4.1 हाफ ॲडर (HA) (Half Adder(HA)- ज्या सर्किटमध्ये २ बायनरी बिट्सची बेरीज केली जाते त्याला हाफ ॲडर (Half adder) म्हणतात. A आणि B बिट्स हे इनपुट व्हेरिअबल्स (variables) आहेत आणि सम अँड कॅरी बिट्स (sum and carry bits) हे आउटपुट व्हेरिअबल्स (variables) आहेत.

Figure 1.17 Symbol of Half Adder

हाफ ॲडर सर्किटमध्ये एक महत्त्वाची कमतरता आहे:
इनपुट A आणि B व्यतिरिक्त, बिट्सची जोडी आउटपुट कॅरी निर्माण करू शकते. हाफ ॲडरच्या डिझाइनमध्ये अशा कॅरी ओव्हर इनपुटसाठी कोणतीही तरतूद नाही.

A	B	S =SUM	C=CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Table no 1.11 Truth Table of Half Adder

हाफ ॲडरच्या ट्रुथ टेबल वरून, सम आणि कॅरीसाठी(Sum and Carry) एक्स्प्रेशन खालीलप्रमाणे आहे-
सम =S= A XOR B
कॅरी= C= A AND B
म्हणून हाफ ॲडरचे सर्किट खाली दर्शविल्याप्रमाणे आहे-

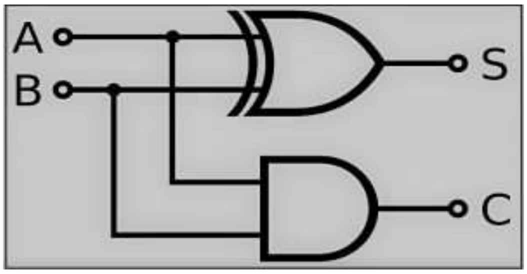


Figure 1.18 Half Adder Circuit

1.4.2 फुल ॲडर सर्किट (Full adder circuit)

फुल ॲडर सर्किट तीन इनपुटची बेरीज करते आणि सम आणि कॅरी दोन आउटपुट देते. हाफ ॲडर सर्किटची कमतरता दूर करण्यासाठी हे डिझाइन केले आहे. पहिले दोन इनपुट A आणि B आहेत आणि तिसरे इनपुट इनपुट कॅरी आहे ज्याला Cin म्हणतात. आउटपुट कॅरीला Cout म्हणतात आणि सम(sum) आउटपुटला S म्हणतात.

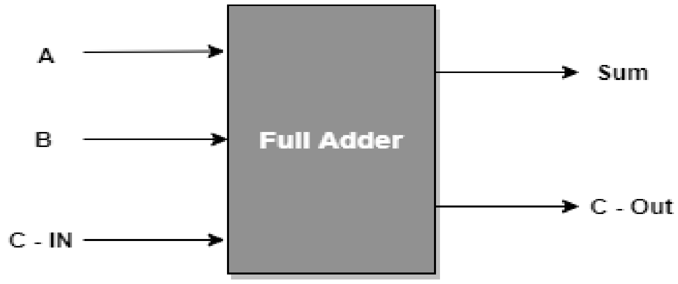


Figure 1.19 Symbol Full Adder Circuit

Inputs			Outputs	
A	B	C – IN	Sum	C - Out
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

Table no 1.12 Truth Table Full Adder

फुल ॲडर ट्रुथ टेबल वरून, सम आणि कॅरी साठी एक्स्प्रेशन खालीलप्रमाणे आहे-

सम = S = A XOR B XOR Bin

कॅरी = C = ABin + AB + BBin

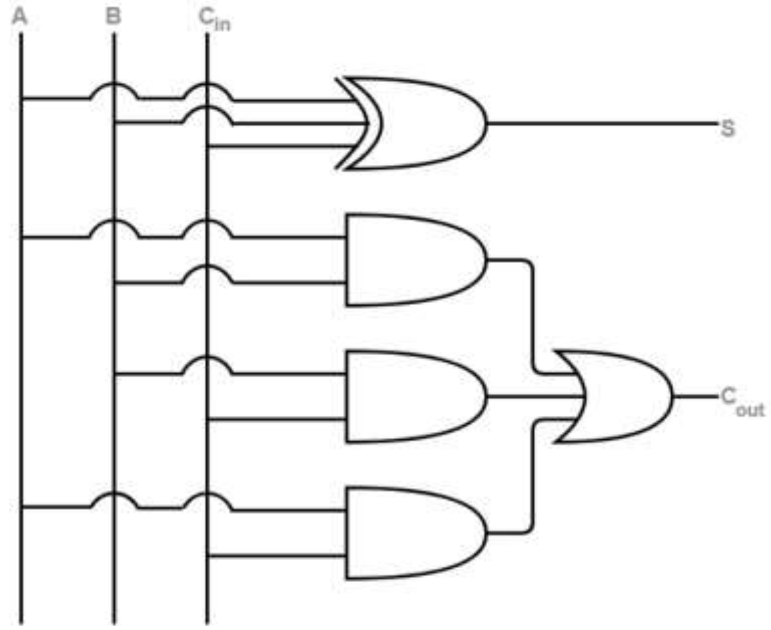


Figure 1.20 Full Adder Circuit

1.4.3 हाफ सबट्रक्टर (HS) (Half Subtractor)

ज्या सर्किटमध्ये २ बायनरी बिट्सची वजाबाकी केली जाते त्याला हाफ सबट्रक्टर(Half subtractor) म्हणतात. त्यात दोन इनपुट मिन्युएंड आणि सबट्राहेंड (minuend and subtrahend) आणि दोन आउटपुट आहेत. डिफरेन्स (Difference) आणि बॉरो (Borrow).वजाबाकी बायनरी वजाबाकीच्या नियमानुसार(Rules of binary subtraction) केली जाते.

A	B	D =Difference	B= Borrow
0	0	0	0
0	1	1	0
1	0	1	1
1	1	0	0

Table no 1.13 Truth Table of Half Subtractor

हाफ सबट्रक्टरच्या ट्रुथ टेबल वरून, डिफरेन्सआणि बॉरोसाठी(Difference and borrow)एक्स्प्रेशन खालीलप्रमाणे आहे-

डिफरेन्स = D = A XOR B

बॉरो = B = A’ AND B

म्हणून हाफ सबट्रक्टर चे सर्किट खाली दर्शविल्याप्रमाणे आहे-

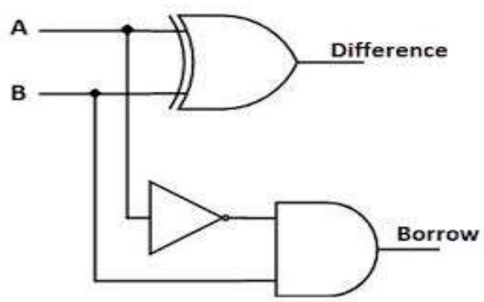


Figure 1.21 Half Subtractor Circuit

1.4.4 फुल सबट्रक्टर(Full Subtractor)

फुल सबट्रक्टर (Full subtractor) सर्किट तीन इनपुटची वजाबाकी करते. डिफरेन्स आणि बॉरो असे दोन आउटपुट देते. या सर्किटमध्ये तीन इनपुट आणि दोन आउटपुट आहेत. तीन इनपुट A, B आणि Bin, अनुक्रमे मिन्युएंड,

सबट्राहेंड(minuend and subtrahend) आणि बॉरो दर्शवितात. दोन आउटपुट, D आणि Bout अनुक्रमे डिफरेन्स आणि बॉरो दर्शवितात.

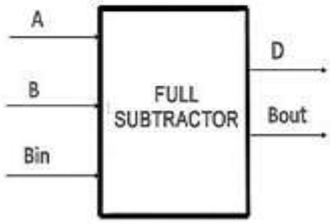


Figure 1.22 Symbol Full Subtractor

Minuend (A)	Subtrahend (B)	Borrow In (B_{in})	Difference (D)	Borrow Out (B_O)
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

Table no 1.14 Truth Table of Full Subtractor

फुल सबट्रक्टरच्या ट्रुथ टेबल वरून, डिफरेन्स आणि बॉरोसाठी(Difference and borrow)एक्स्प्रेशन खालीलप्रमाणे आहे-

डिफरेन्स = D=A XOR B XOR Bin

बॉरो =B = A’Bin+A’B+BBin

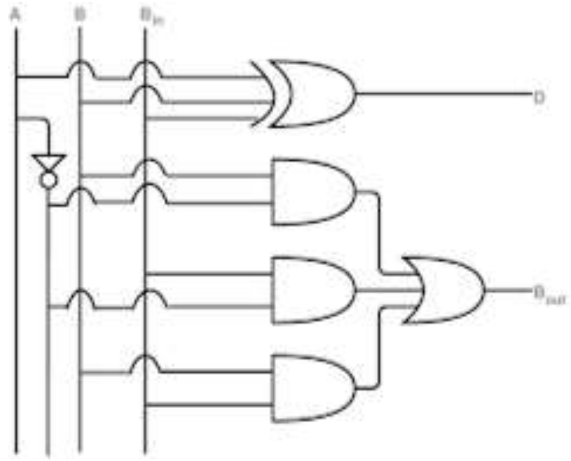


Figure 1.23 Full Adder Circuit

प्रश्न (Questions):

1. खालील संख्या रूपांतरित करा, सर्व पायऱ्या(**steps**) दाखवा.(Convert the following)

a) $(36.15)10 = (?)_{16} = (?)_{8} = (?)_{2}$

b) $(74.9)10 =(?)_{16} = (?)_{8} = (?)_{2}$

c) $(11000)2=(?)_{16} = (?)_{8} = (?)_{10}$

d) $(AD92 . BC A)_{16}= (?)_{10}= (?)_{8}= (?)_{2}$

e) $(1101011)2 = (?)_{16}$

f) $(1111011)_{2} = (?)_{8}$

2. खालील बायनरी ऑपरेशन्स (Binary operations) करा

a) (10111001) + (11010)

b) (11001) - (1101)

c) (111011) – (11100)

d) (1010) – (101)

e) (1101) x (110)

f) 1011100 / 100

3. 1's कॉम्पलिमेन्ट्स पद्धतीचा (1's complement) वापर करून वजाबाकी करा

a) (1110) – (0101)

b) (10001) – (11000)

4. 2's कॉम्पलिमेन्ट्स पद्धतीचा (2's complement method) वापर करून वजाबाकी करा

a) (50)10 – (2A)16 Hint – Convert each to Binary

b) (10011) – (1101)

5. बीसीडी(BCD) मध्ये रूपांतरित करा आणि बेरिज करा

a) $(17)_{10}$+ $(43)_{10}$

b) $(174)_{10}$+ $(743)_{10}$

6. बेसिक गेट्स आणि युनिव्हर्सल गेट्स चे सिम्बॉल, ट्रुथ टेबलं(Truth table) काढा आणि बुलियन एक्स्प्रेशन (Boolean expression) लिहा.

7. EX-OR आणि EX-NOR गेटचे सिम्बॉल काढा.

8. केवळ NAND गेट वापरून AND, OR आणि NOT गेट्स काढा.

9. केवळ NOR गेट वापरून AND, OR आणि NOT गेट्स काढा.

10.ट्रुथ टेबलं (Truth table)वापरून हाफ ॲडर(HA) डिझाइन करा, सर्किट काढा.

11. ट्रुथ टेबलं(Truth table) वापरून हाफ सबट्रक्टर(HS) डिझाइन करा, सर्किट काढा.

12. ट्रुथ टेबलं(Truth table), लॉजिक आकृती वापरून फुल ॲडर(FA) सर्किटचे कार्य वर्णन करा.

13. ट्रुथ टेबलं(Truth table) ,लॉजिक आकृती वापरून फुल सबट्रक्टर(FS) सर्किटचे कार्य वर्णन करा.

संदर्भ पाठ्यपुस्तके (Reference Books):

Sr. No	Author	Title	Publisher with ISBN Number
1	R. P. Jain	Modern Digital Electronics	McGraw Hill Publishing, New Delhi, 2009; ISBN: 9780070669116
2	V. K. Puri	Digital Electronics	McGraw Hill Education (1 July 2017); ISBN-13 : 978-0074633175
3	Salivahanan S.; Arivazhagan S.	Digital Circuits and Design	Oxford University Press India; 5th edition ; ISBN13- 978-0199488681
4	Malvino, A.P.; Leach, D.P.; Saha G.	Digital Principles and Applications	McGraw Hill Education, New Delhi, 2014, ISBN : 9789339203405

माहती संकेतस्थळ (Reference Links)

1. https://www.youtube.com/watch?v=Ri0VhKhw7dY
2. https://www.youtube.com/watch?v=31N9tjaWZbs
3. https://www.youtube.com/watch?v=kx8DFNf6pQU
4. https://www.youtube.com/watch?v=L_m7jBvtzpQ
5. **Diagram Courtesy** https://www.researchgate.net/figure/Schematic-diagram-of-implementation-of-basic-gates-using-NOR-gate_fig3_324926558
6. **Diagram Courtesy** https://www.researchgate.net/figure/Schematic-diagram-of-implementation-of-basic-gates-using-NAND-gate_fig2_324926558

युनिट - II
डिजिटल लॉजिक सर्किट्स
(Digital Logic Circuits)

विषय निष्पत्ती (Course Level Learning Outcome):

CO2 - कॉम्बिनेशनल आणि सीक्वेन्सियल सर्किट तयार करा.
(Build simple combinational and sequential circuits.)

विषय निष्पत्ती बरोबर जुळणारी थिअरी लर्निंग निष्पत्ती (Theory Learning Outcomes (TLOs) aligned to Course Outcomes (COs)) :

2.1 दिलेल्या इनपुट आणि आउटपुट लाईन्ससाठी मक्स/डिमक्स(MUX/DEMUX) ट्री काढा.
(Draw MUX/DEMUX tree for the given number of input and output lines.)
2.2 फ्लिप-फ्लॉपच्या नमूद केलेल्या रचनेच्या प्रकियेचे वर्णन करा
(Describe the building process of the specified type of flip-flop.)
2.3 असिंक्रोनस काउंटर डिझाइन करण्यासाठी दिलेल्या फ्लिप-फ्लॉपच्या एक्सायटेशन टेबलचा वापर करा.
(Use excitation table of the given flip-flop to design asynchronous counter.)

2.1 मल्टीप्लेक्सर आणि डीमल्टीप्लेक्सरचे कार्य, ट्रुथ टेबल (Truth Table) आणि मल्टीप्लेक्सर आणि डीमल्टीप्लेक्सरचे अनुप्रयोग.

2.1.1 मल्टीप्लेक्सर:

- मल्टीप्लेक्सर हे डिजिटल सर्किट आहे जे n डेटा इनपुटपैकी एक निवडते आणि ते आउटपुटकडे पाठवते . n इनपुटपैकी एकाची निवड सिलेक्शन लाईन्स इनपुटद्वारे केली जाते.
- मल्टीप्लेक्सर हे एक विशेष प्रकारचे कॉम्बिनेशनल सर्किट आहे. n:1 मल्टीप्लेक्सरचा ब्लॉक डायग्राम आकृती 2.1 मध्ये दाखवल्याप्रमाणे आहे. n म्हणजे इनपुट लाईन्सची संख्या.

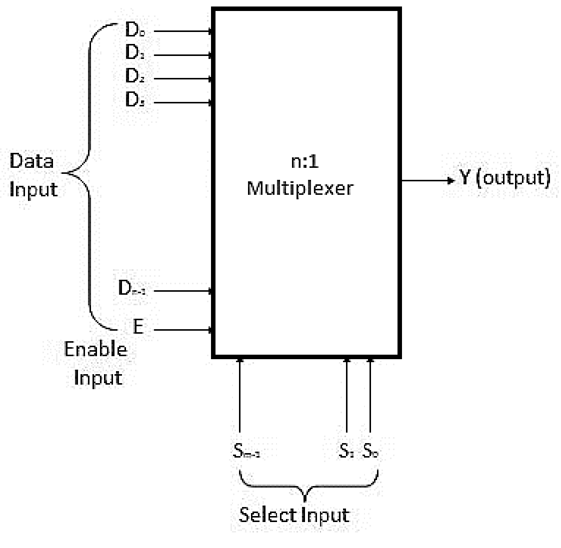


Figure 2.1: Block diagram of multiplexer.

- मल्टीप्लेक्सरच्या बाबतीत, आउटपुट लाईनमधून प्रवाहित होण्यासाठी इच्छित डेटा इनपुटची निवड SELECT लाईन्सच्या मदतीने नियंत्रित केली जाते. आकृती 2.1 मधील मक्स च्या ब्लॉक डायग्राममध्ये, D0, D1,... Dn-1, म्हणजेच, (2^n) इनपुट लाईन्स आहेत आणि "n" इनपुट पैकी एकाची निवड सिलेक्ट लाइनी केली जाते
- E ला स्ट्रोब किंवा एनेबल(ENABLE) इनपुट म्हणतात जे कॅस्केडिंगसाठी उपयुक्त आहे.

2.1.2 मल्टीप्लेक्सरचे प्रकार

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

2.1.3 1: 2 मल्टीप्लेक्सर

2:1 मल्टीप्लेक्सरचा ब्लॉक डायग्रामआकृती 2.2 मध्ये दाखवला आहे. 2:1 मल्टीप्लेक्सर हा मूलभूत दोन इनपुट मल्टीप्लेक्सर आहे ज्यामध्ये I_0 आणि I_1 म्हणून नियुक्त केलेल्या दोन डेटा इनपुट लाइन आहेत, एक डेटा सिलेक्ट लाइन S ने दर्शविली आहे आणि एक आउटपुट लाइन Y ने दर्शविली आहे.

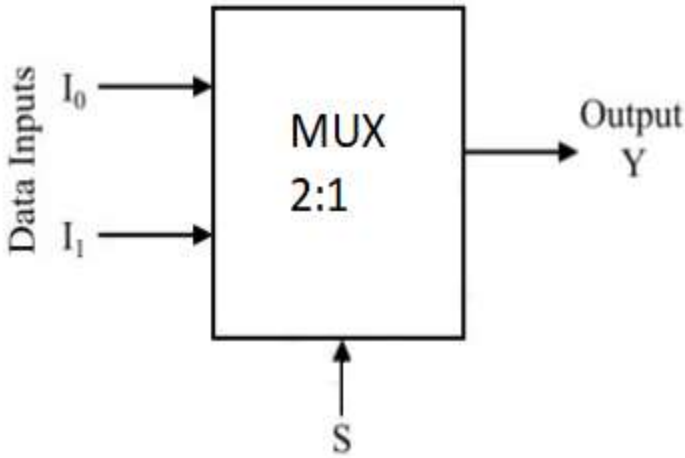


Figure 2.2: Block diagram of 2:1multiplexer

2:1 मल्टीप्लेक्सरमध्ये, सिलेक्ट लाइन वरून डिजिटल सिग्नल कोणता इनपुट डेटा लाइन आटपूटला जोडला जाईल हे ठरवते.) 2:1 मल्टीप्लेक्सरचे कार्यखालील ट्रुथ टेबलवरून (Truth Table) समजू शकते.

सिलेक्ट लाइन (S)	आउटपुट (Y)
0	I_0
1	I_1

Table No 2.1: Truth table of 2:1 multiplexer

Y या एक्स्प्रेशन लॉजिकल एक्स्प्रेशन खालीलप्रमाणे आहे::

$Y=S_0'.I_0+S_0.I_1$

वरील एक्स्प्रेशनचे लॉजिकल सर्किट खाली दिले आहे:

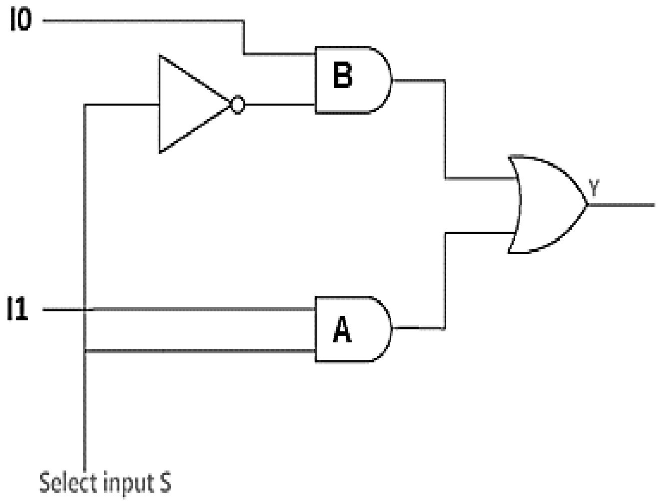


Figure 2.3: Logic circuit Diagram of 2:1 Mux

ऑपरेशन: आकृती-2.3मध्ये दाखवलेल्या 2:1 मक्स (MUX) चे हे लॉजिक सर्किट खालीलप्रमाणे काम करेल -

• जेव्हा S = 0 असते, तेव्हा ॲन्ड(AND) गेट A एनेबल होते आणि B, ॲन्ड(AND) गेट डिसेबल(Disable) होतात म्हणून आऊटपूट Y= I_0

• जेव्हा S = 1 असते, तेव्हा ॲन्ड(AND) गेट A डिसेबल (Disable) होते आणि ॲन्ड(AND) गेट B एनेबल (ENABLE) होतात म्हणून आऊटपूट Y= I_1

2.1.4 4:1 मल्टीप्लेक्सर

4:1 मल्टीप्लेक्सरमध्ये चार डेटा इनपुट I_3, I_2, I_1 आणि I_0, दोन सिलेक्शन लाईन्स S_1आणि S_2 आणि एक आउटपुट Y आहेत. 4:1 मल्टीप्लेक्सरचा ब्लॉक डायग्राम खालील आकृतीत दाखवला आहे.

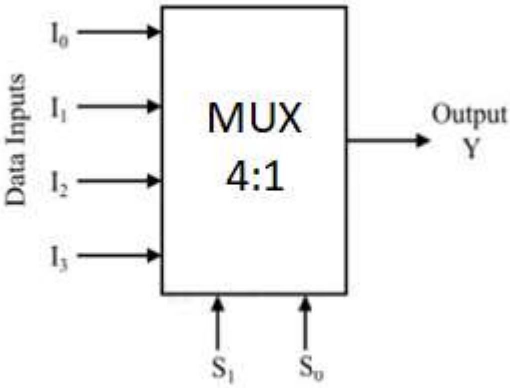


Figure 2.4: Block diagram of 4:1 multiplexer

4:1 मल्टीप्लेक्सरमध्ये, या दोन्ही सिलेक्ट लाइन वरून डिजिटल सिग्नल कोणता इनपुट डेटा लाइन आटपूटला जोडला जाईल हे ठरवते.

4:1 मल्टीप्लेक्सरचा ट्रुथ टेबलखाली दाखवला आहे.

सिलेक्ट लाइन (Selection Lines)		Output
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Table no 2.2: Truth table of 4:1 Multiplexer is shown below.

वरील Y या एक्स्प्रेशन चे लॉजिकल (logical)सर्किट खाली दिले आहे:

$Y = S_1'S_0'I_0+S_1'S_0I_1+S_1S_0'I_2+S_1S_0I_3$

आपण इन्व्हर्टर, ॲड(AND) गेट्स आणि OR गेट वापरून हे बुलियन फंक्शन अंमलात आणू शकतो. 4:1 मल्टीप्लेक्सरचा सर्किट लॉजिक आकृती खालील आकृतीत दाखवला आहे.

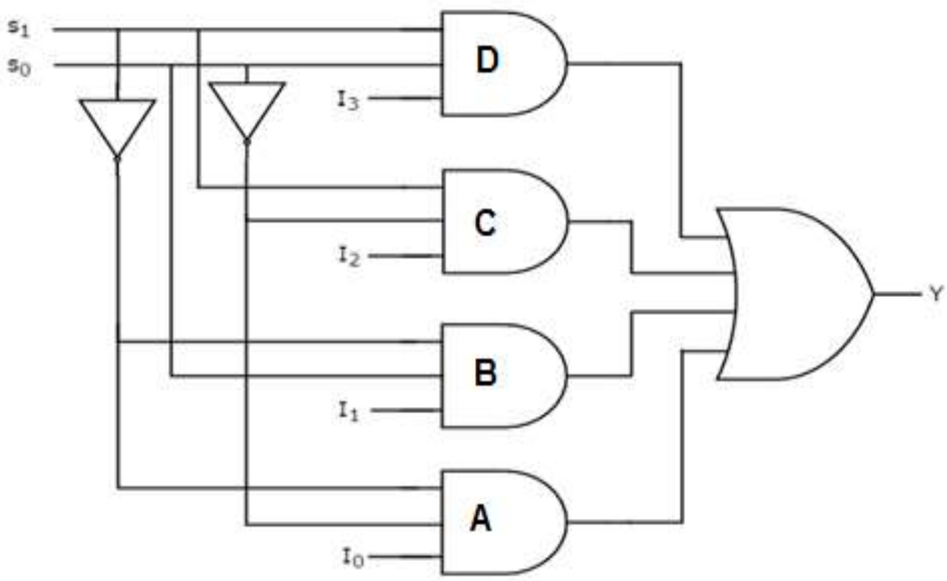


Figure 2.5: Logic circuit Diagram of 4:1 Mux

• **कार्य**:

आकृती-2.5 मध्ये दाखवलेल्या 4:1 मक्सचे हे लॉजिक सर्किट खालीलप्रमाणे काम करेल • जेव्हा $S_1 = 0$ आणि $S_0 = 0$, तेव्हा ॲन्ड(AND) गेट A एनेबल होते आणि B, C, D,ॲन्ड(AND) गेट डिसेबल(Disable) होतात म्हणून आऊटपूट $Y= I_0$

• जेव्हा $S_1 = 0$ आणि $S_0 = 1$ तेव्हा ॲन्ड(AND) गेट B एनेबल होते आणि A, C, D,ॲन्ड(AND) गेट डिसेबल(Disable) होतात म्हणून आऊटपूट $Y= I_1$.

• जेव्हा $S_1 = 1$ आणि $S_0 = 0$ तेव्हा ॲन्ड(AND) गेट C एनेबल होते आणि A, B, D,ॲन्ड(AND) गेट डिसेबल(Disable) होतात म्हणून आऊटपूट $Y= I_2$

• जेव्हा $S_1 = 1$ आणि $S_0 = 1$ तेव्हा ॲन्ड(AND) गेट D एनेबल होते आणि A, B, C,ॲन्ड(AND) गेट डिसेबल(Disable) होतात म्हणून आऊटपूट $Y= I_3$

2.1.5 8:1 मल्टीप्लेक्सर

8:1 मल्टीप्लेक्सरमध्ये आठ डेटा इनपुट I_7 ते I_0, तीन सिलेक्शन लाईन्स S_2, S_1 आणि S_0 आणि एक आउटपुट Y आहेत. सिलेक्ट लाइन्सच्या कॉम्बिनेशन वर आठ इनपुट पैकी एक इनपुट आऊटपुटशी जोडले जाते.

8:1 मल्टीप्लेक्सरचा ब्लॉक डायग्राम खालील आकृतीमध्ये दाखवला आहे.

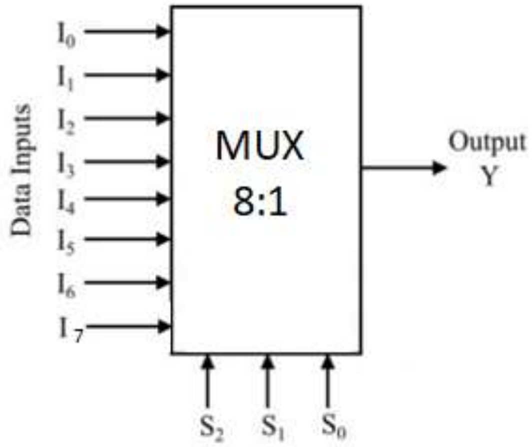


Figure 2.6: Block diagram of 8:1 multiplexer

8:1 मल्टीप्लेक्सरचा ट्रयुथ टेबलखाली दर्शविला आहे.

सिलेक्ट लाइन (Selection Inputs)			Output
S_2	S_1	S_0	Y
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

Table no 2.3.The Truth table of 8:1 Multiplexer

Y या एक्स्प्रेशन लॉजिकल एक्स्प्रेशन खालीलप्रमाणे आहे:

$Y=S_0'.S_1'.S_2'.I_0+S_0.S_1'.S_2'.I_1+S_0'.S_1.S_2'.I_2+S_0.S_1.S_2'.I_3+S_0'.S_1'.S_2\ I_4+S_0.S_1'.S_2\ I_5+S_0'.S_1.S_2\ .I_6+S_0.S_1.S_3.I_7$

वरील Y या एक्स्प्रेशन चे लॉजिकल (logical)सर्किट खाली दिले आहे:

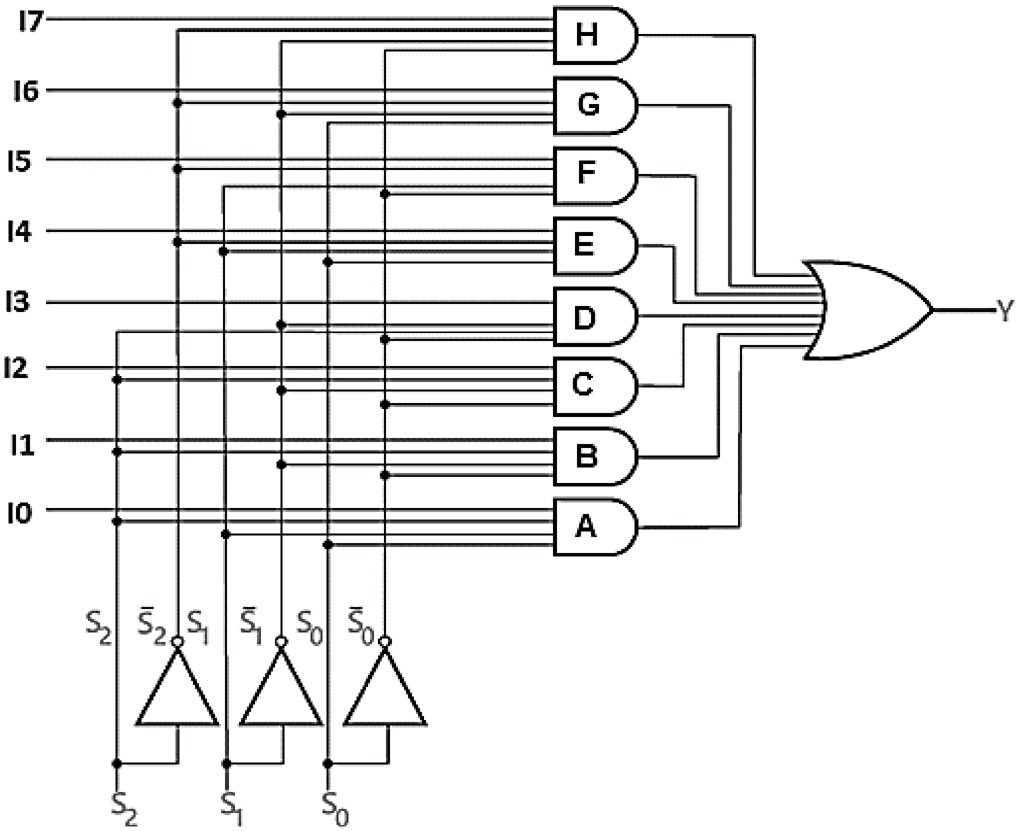


Figure 2.7: Logic circuit Diagram of 8:1 Mux

आकृती 2.7 मध्ये दाखवलेले 8:1 मक्स चे हे लॉजिक सर्किट खालीलप्रमाणे कार्य करते -

• जेव्हा $S_2 = 0$, $S_1 = 0$, आणि $S_0= 0$, तेव्हा ॲन्ड (AND) गेट A एनेबल(ENABLE) होते आणि लॉजिक सर्किटमधील इतर सर्व ॲन्ड (AND) गेट्स डिसेबल होतात . म्हणून, आउटपुट $Y = I_0$.

• जेव्हा $S_2 = 0$, $S_1 = 0$, आणि $S_0= 1$, तेव्हा ॲन्ड (AND) गेट B एनेबल(ENABLE) होतेआणि लॉजिक सर्किटमधील इतर सर्व ॲन्ड (AND) गेट्स डिसेबल होतात . म्हणून, आउटपुट $Y = I_1$.

• जेव्हा $S_2 = 0$, $S_1 = 1$, आणि $S_0= 0$, तेव्हा ॲन्ड (AND) गेट C एनेबल(ENABLE) होतेआणि लॉजिक सर्किटमधील इतर सर्व ॲन्ड (AND) गेट्स डिसेबलहोतात . म्हणून, आउटपुट $Y = I_2$.

जेव्हा $S_2 = 0$, $S_1 = 1$, आणि $S_0= 1$, तेव्हा ॲन्ड (AND) गेट D एनेबल(ENABLE) होतेआणि लॉजिक सर्किटमधील इतर सर्व ॲन्ड (AND) गेट्स डिसेबल होतात . म्हणून, आउटपुट $Y = I_3$.

• जेव्हा $S_2 = 1$, $S_1 = 0$, आणि $S_0 = 0$, तेव्हा ॲन्ड (AND) गेट E एनेबल(ENABLE) होते आणि लॉजिक सर्किटमधील इतर सर्व ॲन्ड (AND) गेट्स डिसेबल होतात . म्हणून, आउटपुट $Y = I_4$.

• जेव्हा $S_2= 1$, $S_1 = 0$, आणि $S_0= 1$, तेव्हा ॲन्ड (AND) गेट F एनेबल(ENABLE) होते आणि लॉजिक सर्किटमधील इतर सर्व ॲन्ड (AND) गेट्स डिसेबल होतात . म्हणून, आउटपुट $Y = I_5$.

• जेव्हा $S_2 = 1$, $S_1 = 1$, आणि $S_0 = 0$, तेव्हा ॲन्ड (AND) गेट G एनेबल(ENABLE) होते आणि लॉजिक सर्किटमधील इतर सर्व ॲन्ड (AND) गेट्स डिसेबल होतात . म्हणून, आउटपुट $Y = I_6$.

• जेव्हा $S_2 = 1$, $S_1 = 1$, आणि $S_0 = 1$, तेव्हा ॲन्ड (AND) गेट H एनेबल(ENABLE) होते आणि लॉजिक सर्किटमधील इतर सर्व ॲन्ड (AND) गेट्स डिसेबल होतात . म्हणून, आउटपुट $Y = I_7$.

2.1.6 16:1 मल्टीप्लेक्सर

16:1 मल्टीप्लेक्सरमध्ये एकूण 16 इनपुट $I_0, I_1, \ldots, I_{16}$, 4 सिलेक्शन लाईन्स, , S_0, S_1, S_2, आणि S_3 आणि एकच आउटपुट Y आहे. सिलेक्शन लाईन्स S_0, S_1, S_2, आणि S_3 वर असलेल्या इनपुटच्या कॉम्बिनेशन आधारावर, या 16 इनपुटपैकी एक आउटपुटशी जोडला जाईल. 16:1 मल्टीप्लेक्सरचा ब्लॉक डायग्राम आणि ट्रुथ टेबल पुढील प्रमाणे आहे.

ब्लॉक डायग्राम:

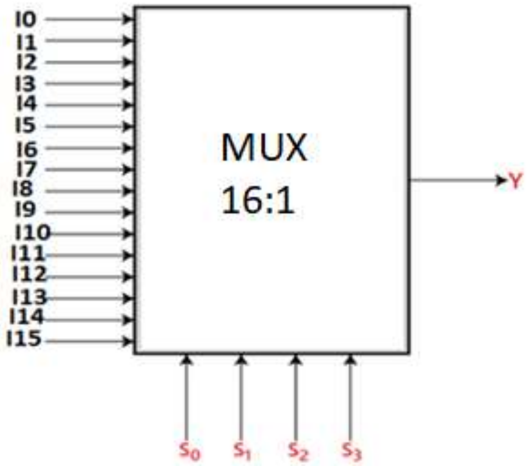


Figure 2.8: Block diagram of 16:1 multiplexer

सिलेक्ट लाइन (Selection Inputs)				Output
S_3	S_2	S_1	S_0	Y
0	0	0	0	I_0
0	0	0	1	I_1
0	0	1	0	I_2
0	0	1	1	I_3
0	1	0	0	I_4
0	1	0	1	I_5
0	1	1	0	I_6
0	1	1	1	I_7
1	0	0	0	I_8
1	0	0	1	I_9
1	0	1	0	I_{10}
1	0	1	1	I_{11}
1	1	0	0	I_{12}
1	1	0	1	I_{13}
1	1	1	0	I_{14}
1	1	1	1	I_{15}

Table no 2.4.The Truth table of 16:1 Multiplexer

Y या एक्स्प्रेशन लॉजिकल एक्स्प्रेशन पुढील प्रमाणे आहे:

$Y=I_0.S_0'.S_1'.S_2'.S_3'+I_1.S_0'.S_1'.S_2'.S_3+I_2.S_0'.S_1'.S_2.S_3'+I_3.S_0'.S_1'.S_2.S_3+I_4.S_0'.S_1.S_2'.S_3'+I_5.S_0'.S_1.S_2'.S_3+I_6.S_1.S_2.S_3'+I_7.S_0'.S_1.S_2.S_3+I_8.S_0.S_1'.S_2'.S_3'+I_9.S_0.S_1'.S_2'.S_3+I_{10}.S_0.S_1'.S_2.S_3'+I_{11}.S_0.S_1'.S_2.S_3+I_{12}S_0.S_1.S_2'.S_3'+I_{13}.S_0.S_1.S_2'.S_3+I_{14}.S_0.S_1.S_2.S_3'+I_{15}.S_0.S_1.S_2'.S_3$

वरील Y या एक्स्प्रेशनचे लॉजिकल (logical)सर्किट खाली दिले आहे:

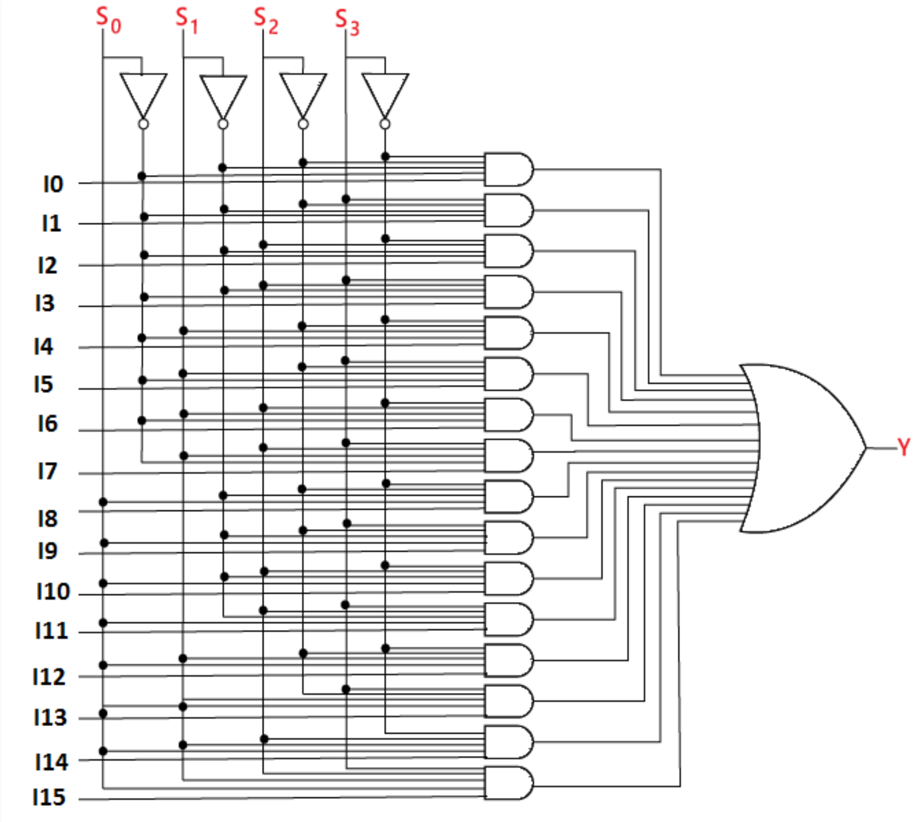


Figure 2.9: Logic circuit Diagram of 16:1 Mux

वरील ट्रुथ टेबल चा विचार करून आपण लोअर ऑर्डर मल्टीप्लेक्सर्स वापरून 16:1 मल्टीप्लेक्सर सहजपणे कंस्ट्रक करू शकतो. 16:1 मल्टीप्लेक्सरचा ब्लॉक आकृती खालील आकृतीमध्ये दाखवला आहे. 8:1 मल्टीप्लेक्सर्सना समान सिलेक्ट लाइन (select lines), S_2, S_1 आणि S_0 लागू केल्या आहेत. वरच्या 8:1 मल्टीप्लेक्सरचे डेटा इनपुट I_{15} ते I_8 आहेत आणि खालच्या 8:1 मल्टीप्लेक्सरचे डेटा इनपुट I_7 ते I_0 आहेत. म्हणून, प्रत्येक 8:1 मल्टीप्लेक्सर सिलेक्ट लाइन च्या(select lines) मूल्यांवर आधारित आउटपुट तयार करतो, S_2, S_1 आणि S_0.

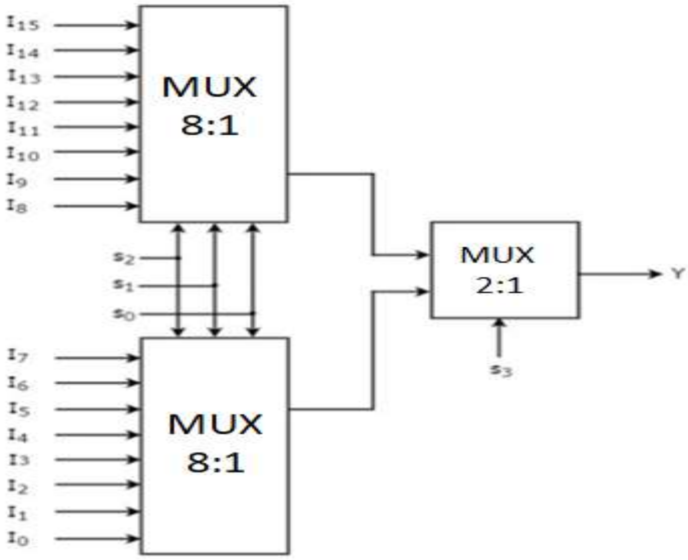


Figure 2.10: 16:1 multiplexer using two 8:1 Mux

पहिल्या टप्प्यातील 8:1 मल्टीप्लेक्सरचे आउटपुट दुसऱ्या टप्प्यात 2:1 मल्टीप्लेक्सरच्या इनपुट म्हणून लागू केले जातात. दुसरी सिलेक्ट लाइन (select line), S_3, 2:1 मल्टीप्लेक्सरवर लागू केली जाते.

जर S_3 शून्य असेल, तर 2:1 मल्टीप्लेक्सरचे आउटपुट सिलेक्ट लाइन S_2, S_1 आणि S_0 च्या मूल्यांवर आधारित 8 इनपुट I_7 ते I_0 पैकी एक असेल.

जर S_3 एक असेल, तर 2:1 मल्टीप्लेक्सरचे आउटपुट सिलेक्ट लाइन S_2, S_1 आणि S_0 च्या मूल्यांवर आधारित 8 इनपुट I_{15} ते I_8 पैकी एक असेल.

म्हणून, दोन 8:1 मल्टीप्लेक्सर आणि एक 2:1 मल्टीप्लेक्सरचे एकूण संयोजन एका 16:1 मल्टीप्लेक्सर म्हणून कार्य करते.

2.1.7 मल्टीप्लेक्सरचे अनुप्रयोग

- डिजिटल सर्किट्सची अंमलबजावणी
- सीपीयूचे नियंत्रण युनिट
- संगणक मेमरी
- डेटा रूटिंग
- कम्युनिकेशन सिस्टम्स

2.1.8 डिमल्टीप्लेक्सर:

डी-मल्टीप्लेक्सर हे एक कॉम्बिनेशनल सर्किट आहे ज्यामध्ये फक्त 1 इनपुट लाइन आणि 2^N आउटपुट लाइन असतात. सोप्या भाषेत सांगायचे तर, मल्टीप्लेक्सर हे एक-इनपुट आणि मल्टी-आउटपुट कॉम्बिनेशनल सर्किट आहे. माहिती सिंगल इनपुट लाइनमधून प्राप्त केली जाते आणि आउटपुट लाइनकडे पाठविले जाते . सिलेक्ट लाइन च्या(select lines) मूल्यांच्या आधारे, इनपुट यापैकी एका आउटपुटशी जोडले जाईल. डी-मल्टीप्लेक्सर हे मल्टीप्लेक्सरच्या विरुद्ध कार्य आहे.

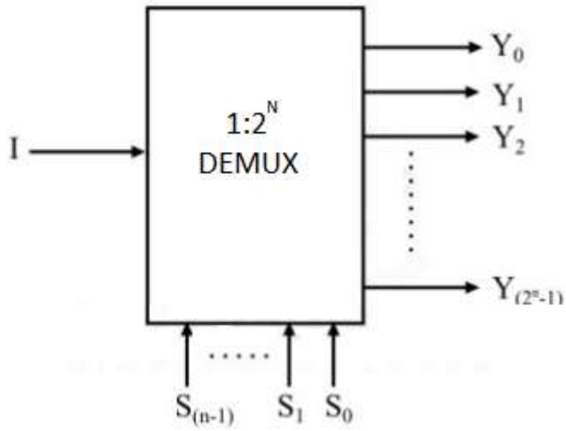


Figure 2.11: Block diagram of $1:2^n$ Demultiplexer

2.1.9 डिमल्टीप्लेक्सरचे प्रकार

आउटपुट लाईन्सच्या संख्येवर आधारित (2^N), डिमल्टीप्लेक्सरचे अनेक प्रकारांमध्ये वर्गीकरण केले जाऊ शकते. डिमल्टीप्लेक्सरचे काही सामान्यतः वापरले जाणारे प्रकार आहेत -

- 1:2 डिमल्टीप्लेक्सर
- 1:4 डिमल्टीप्लेक्सर
- 1:8 डिमल्टीप्लेक्सर
- 1:16 डिमल्टीप्लेक्सर

2.1.10 1:2 डीमल्टीप्लेक्सर:

1:2 डी-मल्टीप्लेक्सरमध्ये, फक्त दोन आउटपुट Y_0, आणि Y_1, 1 सिलेक्शन लाईन्स, S_0, आणि सिंगल इनपुट, A आहे. सिलेक्शन व्हॅल्यूच्या आधारावर, इनपुट एका आउटपुटशी जोडले जाईल. 1:2 मल्टीप्लेक्सरचा ब्लॉक डायग्राम आणि ट्रुथ टेबलखाली दिला आहे

1:2 डीमल्टीप्लेक्सरमध्ये 1 इनपुट लाईन I, 1 सिलेक्ट लाईन S_0 आणि 2 आउटपुट लाईन्स (Y_0 आणि Y_1) असतात. सिलेक्ट लाईनवर लागू केलेला लॉजिक लेव्हल इनपुट डेटा कोणत्या आउटपुट लाईनवर ट्रान्समिट केला जाईल हे ठरवते.

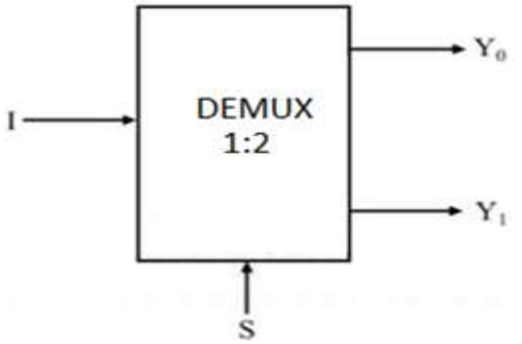


Figure 2.12: Block diagram of 1:2 Demultiplexer

खाली दिलेल्या फंक्शन टेबलच्या मदतीने 1:2 डिमल्टीप्लेक्सरच्या कार्य चे विश्लेषण करता येते.

सिलेक्ट लाइन **(Select Line)**	**Outputs**	
S	Y_1	Y_0
0	0	I
1	I	0

Table no 2.5.The Truth table of 1:2 Demultiplexer

1:2 डिमल्टीप्लेक्सरच्या या फंक्शन टेबलवरून, आपण खालीलप्रमाणे प्रत्येक आउटपुटसाठी थेट बुलियन एक्सप्रेशन मिळवू शकतो.

Y0=S'I and Y1=SI

वरील Y0,Y1 या संज्ञेचे लॉजिकल (logical)सर्किट खाली दिले आहे:

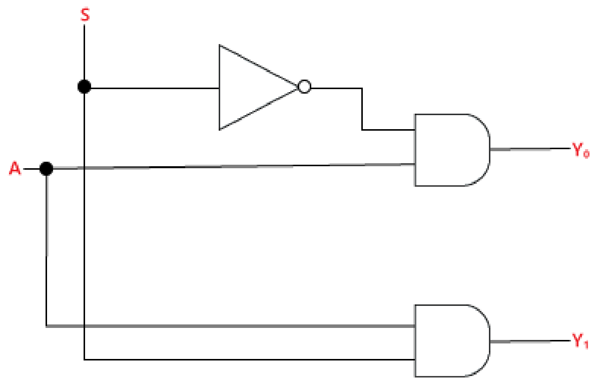


Figure 2.13: Logic circuit Diagram of 1:2 Demux

2.1.11 1:4 डीमल्टीप्लेक्सर

1:4 डीमल्टीप्लेक्सर चा फंक्शनल ब्लॉक आकृती आकृती 2.14 मध्ये दाखवला आहे

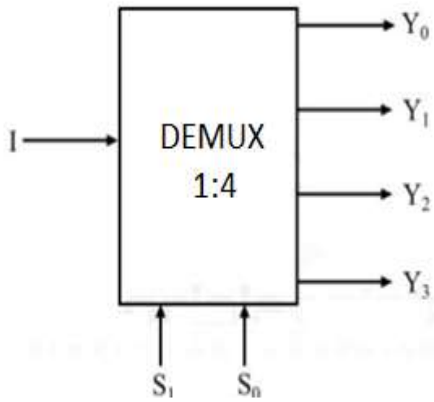


Figure 2.14: Block diagram of 1:4 Demultiplexer

1:4 डिमल्टीप्लेक्सरमध्ये 1 इनपुट लाइन (I), 2 सिलेक्ट लाइन (S_0 आणि S_1) आणि 4 आउटपुट लाइन (Y_0, Y_1, Y_2 आणि Y_3) आहेत. सिलेक्ट लाइनवर लागू केलेला लॉजिक लेव्हल इनपुट डेटा (I) कोणत्या आउटपुट लाइनवर प्रसारित केला जाईल हे ठरवतो.

1:4 डिमल्टीप्लेक्सरचे कार्यखाली दिलेल्या त्याच्या फंक्शन टेबलच्या मदतीने समजू शकते.
ट्रुथ टेबल:

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

Table no 2.6.The Truth table of 1:4 Demultiplexer

Y या एक्स्प्रेशन लॉजिकल एक्स्प्रेशन खालीलप्रमाणे आहे:

$Y_0=S_1'S_0'A$

$Y_1=S_1'S_0A$

$Y_2=S_1S_0'A$

$Y_3=S_1\ S_0\ A$

वरील Y_0, Y_1, Y_2, Y_3या संज्ञेचे लॉजिकल (logical)सर्किट खाली दिले आहे:

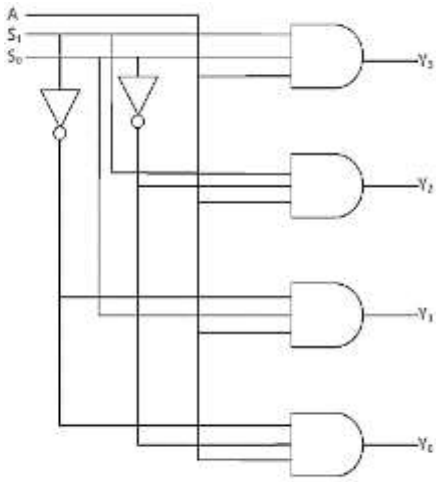


Figure 2.15: Logic circuit Diagram of 1:4 Demux

2.1.12 1:8 डीमल्टीप्लेक्सर

1 : 8 डी-मल्टीप्लेक्सरमध्ये एकूण आठ आउटपुट Y_0, Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, आणि Y_7, 3 सिलेक्ट लाइन (Select Lines) S_0, S_1आणि S_2 आणि एक इनपुट, , A आहे. सिलेक्ट लाइन S_0, S_1आणि S_2 वर असलेल्या इनपुटच्या संयोजनाच्या आधारावर, इनपुट यापैकी एका आउटपुटशी जोडला जाईल. 1:8 डीमल्टीप्लेक्सरचा ब्लॉक आकृती आणि ट्रुथ टेबलखाली दिली आहे.

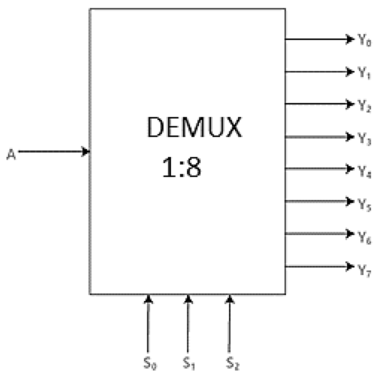


Figure 2.16: Block diagram of 1:8 Demultiplexer

INPUTS			Output							
S_2	S_1	S_0	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	A
0	0	1	0	0	0	0	0	0	A	0
0	1	0	0	0	0	0	0	A	0	0
0	1	1	0	0	0	0	A	0	0	0
1	0	0	0	0	0	A	0	0	0	0
1	0	1	0	0	A	0	0	0	0	0
1	1	0	0	A	0	0	0	0	0	0
1	1	1	A	0	0	0	0	0	0	0

Table no 2.7.The Truth table of 1:8 Demultiplexer

Y या एक्स्प्रेशन लॉजिकल एक्स्प्रेशन खालीलप्रमाणे आहे:

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

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

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

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

$Y_4=S_0'.S_1'.S_2A$

$Y_5=S_0.S_1'.S_2A$

$Y_6=S_0'.S_1.S_2A$

$Y_7=S_0.S_1.S_3.A$

वरील Y_0, Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, Y_7या संज्ञेचे लॉजिकल (logical) सर्किट खाली दिले आहे:

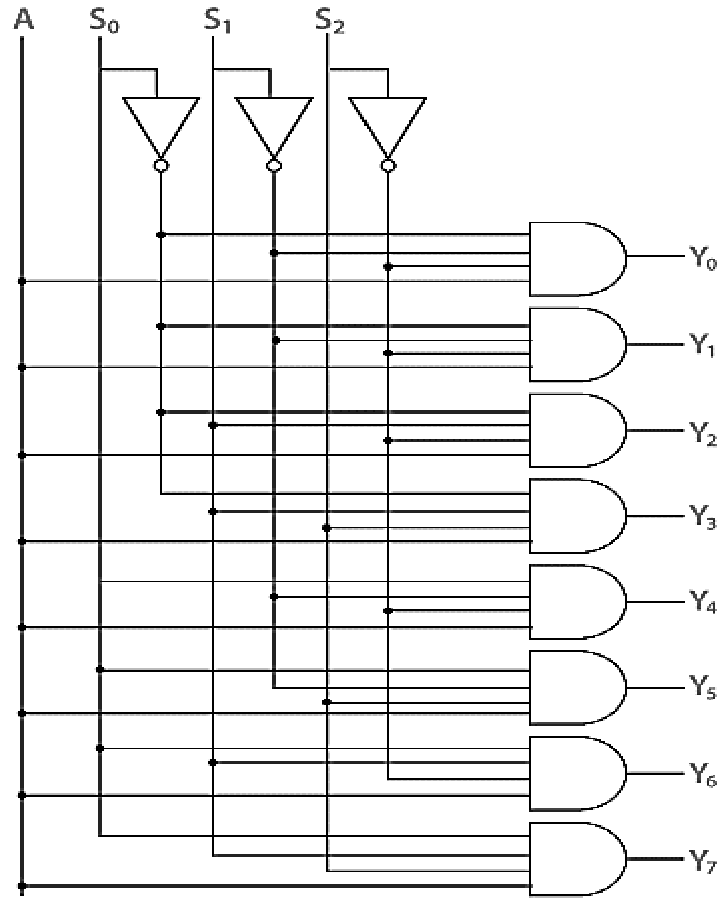


Figure 2.17: Logic circuit Diagram of 1:8 Demux

2.1.13 1:8 डीमल्टीप्लेक्सर 1:4आणि 1:2डीमल्टीप्लेक्सर वापरून

आपण लोअर ऑर्डर डी-मल्टीप्लेक्सर वापरून 1:8 डीमल्टीप्लेक्सर अंमलात आणू शकतो. 1:8 डीमल्टीप्लेक्सर अंमलात आणण्यासाठी, आपल्याला दोन 1:4 डीमल्टीप्लेक्सर आणि एक 1:2 डीमल्टीप्लेक्सर आवश्यक आहे. 1:4 मल्टीप्लेक्सरमध्ये 2 सिलेक्ट लाइन, 4 आउटपुट आणि 1 इनपुट आहे. 1:2 डीमल्टीप्लेक्सरमध्ये फक्त 1 सिलेक्ट लाइन आहे.

8 डेटा आउटपुट मिळविण्यासाठी, आपल्याला दोन 1:4 डीमल्टीप्लेक्सरची आवश्यकता आहे. 1:2 डीमल्टीप्लेक्सर दोन आउटपुट तयार करतो. म्हणून, अंतिम आउटपुट मिळविण्यासाठी, आपल्याला 1:2 डीमल्टीप्लेक्सरचे आउटपुट दोन्ही 1:4 डीमल्टीप्लेक्सरच्या इनपुट म्हणून पास करावे लागतील. 1:4 आणि 1:2 डीमल्टीप्लेक्सर वापरून 1:8 डीमल्टीप्लेक्सरचा ब्लॉक आकृती खाली दिली आहे.

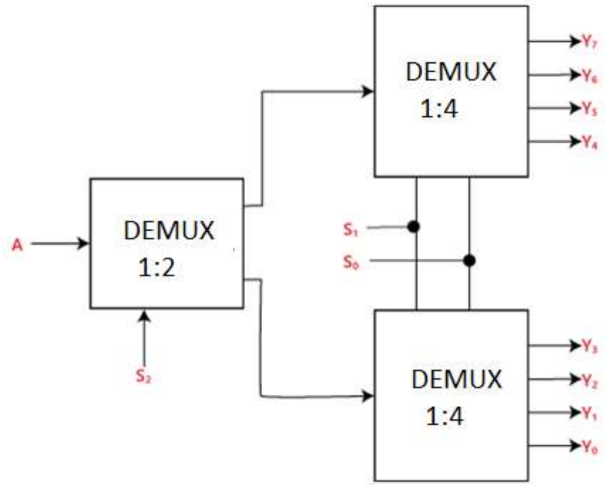


Figure 2.18 :1:8 De-multiplexer using 1:4 and 1:2 de-multiplexer

2.1.14 1:16 डीमल्टीप्लेक्सर

1:16 डी-मल्टीप्लेक्सरमध्ये, एकूण 16 आउटपुट आहेत, $Y_0, Y_1, \ldots, Y_{16}$, 4 सिलेक्ट लाइन, S_0, S_1, S_2, आणि S_3 आणि एक इनपुट, A आहे. सिलेक्ट लाइन S_0, S_1 आणि S_2 वर असलेल्या इनपुटच्या संयोजनाच्या आधारावर, इनपुट यापैकी एका आउटपुटशी जोडला जाईल. 1:16 डीमल्टीप्लेक्सरचा ब्लॉक आकृती आणि ट्रयुथ टेबलखाली दिली आहे.

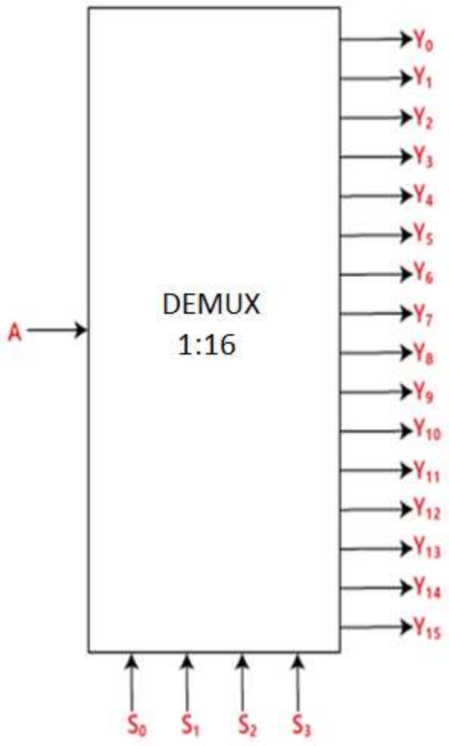


Figure 2.19: Block diagram of 1:16 Demultiplexer

INPUTS				OUTPUTS															
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	A
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	A	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	A	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0	0	0	A	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	A	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	A	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	A	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	A	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	A	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	A	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	A	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	A	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	A	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table no 2.8.The Truth table of 1:16 Demultiplexer

Y या एक्स्प्रेशन लॉजिकल एक्स्प्रेशन खालीलप्रमाणे आहे:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

वरील Y_0, Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, Y_7, Y_8, Y_9, Y_{10}, Y_{11}, Y_{12}, Y_{13}, Y_{14}, Y_{15}या संज्ञेचे लॉजिकल (logical)सर्किट दिले आहे:

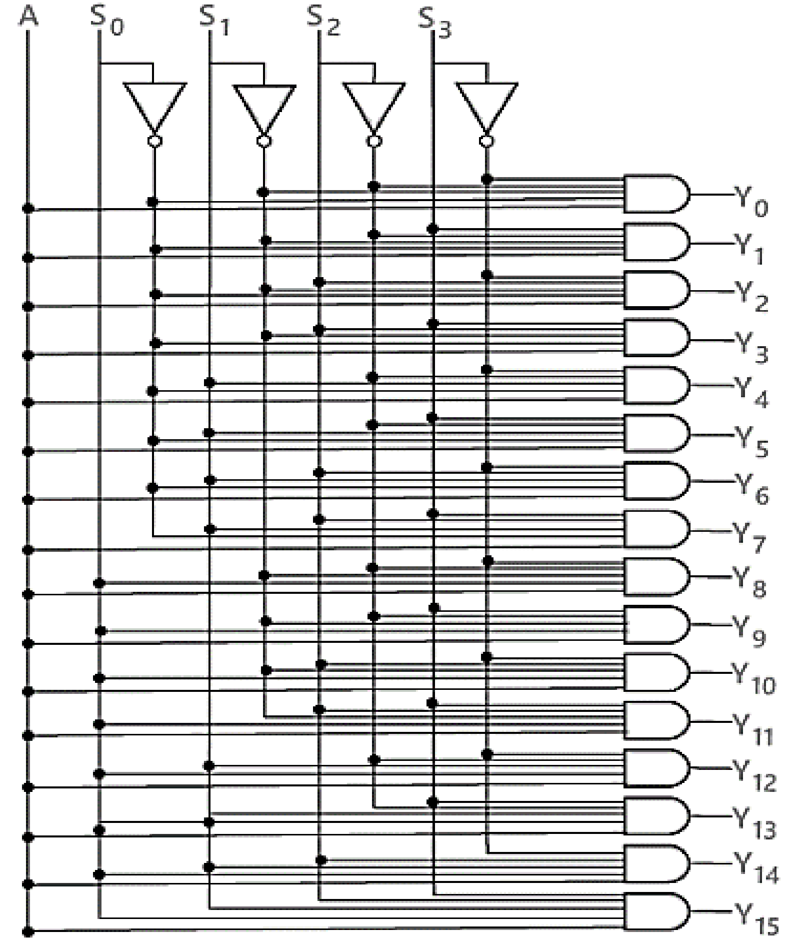


Figure 2.20: Logic circuit Diagram of 1:2 Demux

2.1.15 1:8 आणि 1:2 डीमल्टीप्लेक्सर वापरून 1:16डीमल्टीप्लेक्सर

आपण लोअर ऑर्डर डीमल्टीप्लेक्सर वापरून 1:16 डीमल्टीप्लेक्सर अंमलात(construct) आणू शकतो. 1:16 डीमल्टीप्लेक्सर अंमलात (construct) आणण्यासाठी, आपल्याला दोन 1:8 डीमल्टीप्लेक्सर आणि एक 1:2 डीमल्टीप्लेक्सर आवश्यक आहे. 1:8 मल्टीप्लेक्सरमध्ये 3 सिलेक्शन लाइन, 1 इनपुट आणि 8 आउटपुट आहेत. 1:2 डीमल्टीप्लेक्सरमध्ये फक्त 1 सिलेक्शन लाइन आहे.

16 डेटा आउटपुट मिळविण्यासाठी, आपल्याला दोन 1:8 डीमल्टीप्लेक्सरची आवश्यकता आहे. 1:8 डी-मल्टीप्लेक्सर आठ आउटपुट तयार करतो. म्हणून, अंतिम आउटपुट मिळविण्यासाठी, आपल्याला एकाच इनपुटमधून दोन आउटपुट तयार करण्यासाठी 1:2 डीमल्टीप्लेक्सरची आवश्यकता आहे. नंतर आपण हे आउटपुट इनपुट म्हणून दोन्ही डी-मल्टीप्लेक्सरमध्ये पास करतो. 1:8आणि 1:2 डीमल्टीप्लेक्सर वापरून 1:16 डीमल्टीप्लेक्सरचा ब्लॉक आकृती खाली दिली आहे.

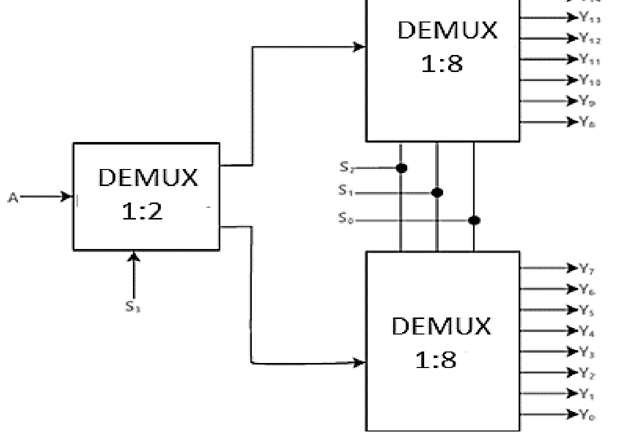


Figure 2.21 :1:16 de-multiplexer using 1:8 and 1:2 de-multiplexer

2. 1:16 डीमल्टीप्लेक्सरचे अनुप्रयोग

डीमल्टीप्लेक्सर हा एक महत्त्वाचा कॉम्बिनेशनल लॉजिक सर्किट आहे जो अनेक अनुप्रयोगांमध्ये वापरला जातो. डीमल्टीप्लेक्सरचे काही महत्त्वाचे उपयोग खाली सूचीबद्ध आहेत –

• डेटा राउटिंगसाठी अनेक इनपुट आणि आउटपुट डिव्हाइसेसमध्ये डीमल्टीप्लेक्सर वापरला जातो.
• सिंक्रोनस सिस्टममध्ये डेटा ट्रान्समिशनसाठी डीमल्टीप्लेक्सर देखील वापरला जातो.
• डेटा ऍक्वीजीशन सिस्टिम (data acquisition systems) डीमल्टीप्लेक्सर देखील वापरला जातो.
• बुलियन फंक्शन्स जनरेट करण्यासाठी डीमल्टीप्लेक्सर वापरला जाऊ शकतो.
• डीमल्टीप्लेक्सरचा वापर सिरीयल ते पॅरलल कन्व्हर्टरमध्ये केला जाऊ शकतो.
• एटीएम पॅकेटच्या प्रसारणासाठी डीमल्टीप्लेक्सरचा वापर केला जातो.
• डीमल्टीप्लेक्सरचा वापर स्वयंचलित चाचणी उपकरणे इत्यादी डिझाइन करण्यासाठी देखील केला जाऊ शकतो.

2.2 फ्लिपफ्लॉप (Flip Flop):

दोन स्थिर अवस्था असलेल्या सर्किटला फ्लिप फ्लॉप मानले जाते. या स्थिर अवस्था बायनरी डेटा साठवण्यासाठी वापरल्या जातात. या अवस्था वेगवेगळ्या इनपुट देऊन बदलता येतात. फ्लिप फ्लॉप हे डिजिटल सिस्टमचे मूलभूत बिल्डिंग ब्लॉक आहेत. फ्लिप फ्लॉप आणि लॅचेस हे डेटा स्टोरेज घटकांची उदाहरणे आहेत. सिक्वेन्सियल (Sequential)लॉजिकल सर्किटमध्ये, फ्लिप फ्लॉप हा मूलभूत स्टोरेज घटक आहे.

फ्लिपफ्लॉपचे प्रकार

• S R फ्लिप-फ्लॉप
• J K फ्लिप-फ्लॉप
• D फ्लिप-फ्लॉप
• T फ्लिप-फ्लॉप

2.2.1 S R फ्लिप-फ्लॉप

आकृतीमध्ये दाखवल्याप्रमाणे आपण नॅन्ड गेट्ससह S R लॅच बांधू शकतो.

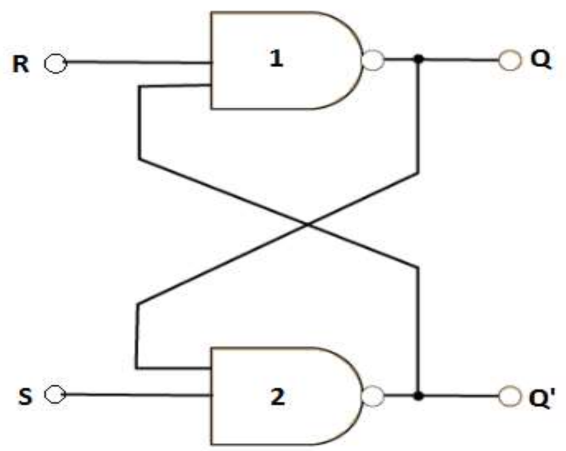


Figure 2.22 S-R Flip-Flop

INPUTS		OUPUTS		
S	R	Q	Q'	Comments
0	0	Qn	Qn'	No change
0	1	0	1	RESET condition
1	0	1	0	SET condition
1	1	Qn	Qn'	No change

Table no 2.8.The Truth table of SR Flip-Flop

S = 0, R = 0 साठी, दोन्ही आउटपुट स्थितीत कोणताही बदल नाही.
S = 0, R = 1 साठी आउटपुट Q = 0, Q' = 1 आहेत आणि ती रीसेट कंन्डिशन आहे.
S = 1, R = 0 साठी, आउटपुट Q = 1, Q' =0 आहेत आणि तिला सेट कंन्डिशन म्हणतात.

S = R = 1 आउटपुट स्थितीत कोणताही बदल नाही.

2.2.2 क्लॉक्ड S R फ्लिप-फ्लॉप

ज्या फ्लिप-फ्लॉप प्रकारात एस (सेट) आणि आर (रीसेट) असे दोन इनपुट असतात त्याला एसआर फ्लिपफ्लॉप म्हणतात. S R फ्लिप-फ्लॉप चे इनपुट आऊटपूट कंट्रोल करतात जेव्हा क्लॉक सिग्नल (लो टु हाय किंवा हाय टु लो) दिलेला असतो कारण, क्लॉक सिग्नल S R फ्लिप-फ्लॉपच्या ऑपरेशनला सिंक्रोनाइझ करतो, म्हणून क्लॉक्ड S R फ्लिप-फ्लॉपला सिंक्रोनस S R फ्लिप-फ्लॉप असेही म्हणतात. क्लॉक्ड किंवा सिंक्रोनस S R फ्लिप-फ्लॉपचा ब्लॉक डायग्राम आकृती 2.23 मध्ये दाखवला आहे.

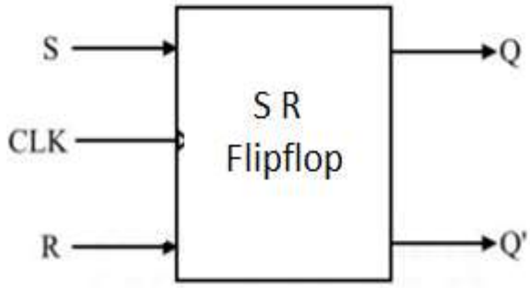


Figure 2.23 Logic Symbol of Clocked S-R Flip-Flop

क्लॉक्ड किंवा सिंक्रोनस एसआर फ्लिप-फ्लॉपचे लॉजिक सर्किट डायग्राम खालील आकृती 2.24 मध्ये दाखवले आहे.

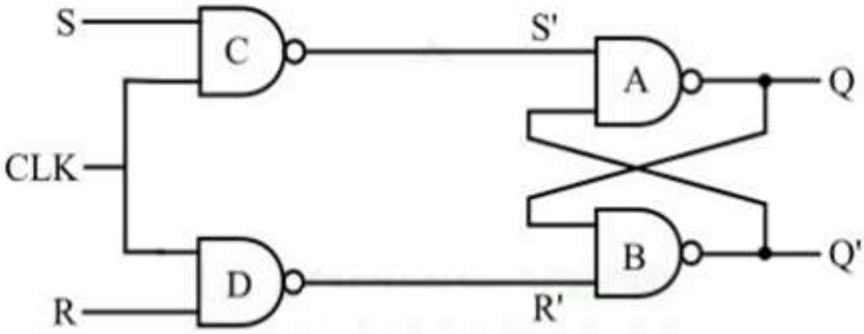


Figure 2.24 Clocked S-R Flip-Flop

सर्किटमध्ये चार नँन्ड NAND गेट्स आहेत हे दिसून येतात. क्लॉक सिग्नल नँन्ड NAND गेट्स C आणि D शी जोडलेला आहे आणि इनपुट S आणि R हे नँन्ड NAND गेट्स C आणि D शी देखील लागू केले आहेत. फ्लिप-फ्लॉपचे स्टोरेज सर्किट तयार करण्यासाठी नँन्ड NAND गेट्स A आणि B क्रॉस-कपल्ड केले आहेत.

- कार्य

क्लॉक्ड S R फ्लिप-फ्लॉपच्या या सर्किटचे कार्य खालीलप्रमाणे वर्णन केले आहे –

○ जेव्हा क्लॉक सिग्नल लागू केला जात नाही, तेव्हा SR फ्लिप-फ्लॉप सर्किट निष्क्रिय राहतो आणि फ्लिप-फ्लॉपच्या आउटपुटमध्ये कोणताही बदल होत नाही.

○ जेव्हा क्लॉक सिग्नल लागू केला जातो, तेव्हा फ्लिप-फ्लॉप सर्किट सक्रिय होते आणि खाली स्पष्ट केल्याप्रमाणे कार्य करते –

a) जेव्हा S = 0 आणि R = 0 अशी अवस्था असते तेव्हा, NAND गेट्स C आणि D चे आउटपुट S' = 1 आणि R' = 1 असते. म्हणून, NAND गेट्स A आणि B चे आउटपुट अपरिवर्तित राहतात. याला SR फ्लिप-फ्लॉपची होल्ड स्टेट म्हणतात.

b) जेव्हा S = 0 आणि R = 1 अशी अवस्था असते तेव्हा, NAND गेट्स C आणि D चे आउटपुट S' = 1 आणि R' = 0 असते, तेव्हा NAND गेट A चे आउटपुट 0 असते आणि NAND गेट B चे आउटपुट 1 असते. याला SR फ्लिप-फ्लॉपची रीसेट स्टेट म्हणतात.

c) जेव्हा S = 1 आणि R = 0, अशी अवस्था असते तेव्हा NAND गेट्स C आणि D चे आउटपुट S' = 0 आणि R' = 1 असते, तेव्हा NAND गेट्स A चे आउटपुट 1 असते आणि NAND गेट B चे आउटपुट 0 असते. याला SR फ्लिप-फ्लॉपची सेट स्टेट म्हणतात.

d) जेव्हा S = 1 आणि R = 1 अशी अवस्था असते तेव्हा, NAND गेट्स C आणि D चे आउटपुट S' = 0 आणि R' = 0 असते, तेव्हा NAND गेट्स A आणि B दोन्हीचे आउटपुट 1 होण्याचा प्रयत्न करतात, जे शक्य नाही. याला SR फ्लिप-फ्लॉपची फोर्बिडन स्टेट म्हणतात.

क्लॉक्ड S R फ्लिप-फ्लॉपचे ट्रुथ टेबल

खाली दिलेल्या ट्रुथ टेबलच्या स्वरूपात आपण क्लॉक्डS Rफ्लिप-फ्लॉपचे कार्यदेखील व्यक्त करू शकतो. येथे, S आणि R इनपुट निर्दिष्ट करतात, Qn आउटपुटची सध्याची स्थिती निर्दिष्ट करतात आणि Qn+1 इनपुटमध्ये बदल आणि क्लॉक पल्सच्या वापरानंतर आउटपुटची स्थिती निर्दिष्ट करतात.

CLOCK	S	R	Qn+1	Qn+1'	Comments
0	x	x	Qn	Qn'	Hold State
1	0	0	Qn	Qn'	Hold State
1	0	1	0	1	RESET condition
1	1	0	1	0	SET condition
1	1	1	Indeterminate		Avoid this condition

Table no 2.9.The Truth table of Clocked SR Flip-Flop

2.2.3 प्रीसेट आणि क्लिअरसह क्लॉक्ड S R फ्लिप फ्लॉप:-

फ्लिप फ्लॉपमध्ये जेव्हा पॉवर चालू होतेतेव्हा सर्किटची स्थिती अनिश्चित असते तेव्हा ती सेट (Q=1) किंवा रीसेट (Q=0) स्थितीत येऊ शकते.

बऱ्याच ॲप्लिकेशन मध्ये सुरुवातीला फ्लिप फ्लॉप सेट किंवा रीसेट करण्याची इच्छा असते. म्हणजेच फ्लिप फ्लॉपची प्रारंभिक स्थिती नियुक्त केली जाते. हे प्रीसेट आणि क्लिअर इनपुट वापरून साध्य केले जाते.

प्रीसेट आणि क्लिअरसह S R फ्लिप फ्लॉप आकृतीमध्ये दर्शविला आहे.

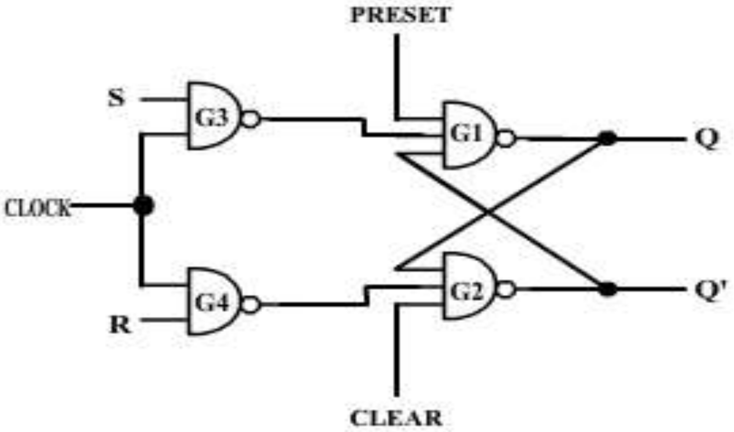
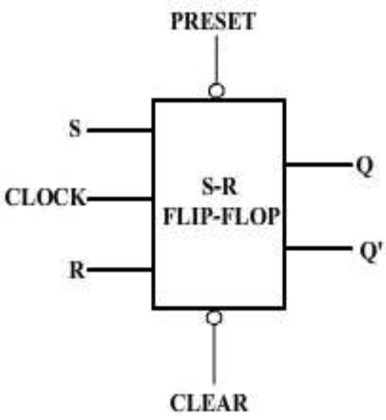


Figure 2.25 Clocked S-R Flip-Flop with Preset and Clear

जर Pr=Cr=1 असेल तर सर्किट S R फ्लिप फ्लॉपच्या ट्रुथ टेबलनुसार कार्य करते

जर Pr=0 आणि Cr=1 असेल तर G1 (Q) चे आउटपुट निश्चितपणे 1 असेल.

Pr=0 बनवणे फ्लिप फ्लॉप सेट करते,कारण G2 चे तिन्ही इनपुट 1 असतील जे $\bar{Q}$=0 करेल आणि Q=1



$\overline{PRE}$	$\overline{CLR}$	CLK	S	R	Q	$\bar{Q}$	Mode
0	1	X	X	X	1	0	preset
1	0	X	X	X	0	1	cleared
0	0	X	X	X	1	1	not used (race)
1	1	⎍	0	0	0	0	hold
1	1	⎍	0	1	0	1	Reset
1	1	⎍	1	0	1	0	Set
1	1	⎍	1	1	1	1	not used (race)

Figure 2.26 Logic symbol and Truth Table of Clocked S-R Flip-Flop With Preset and Clear.

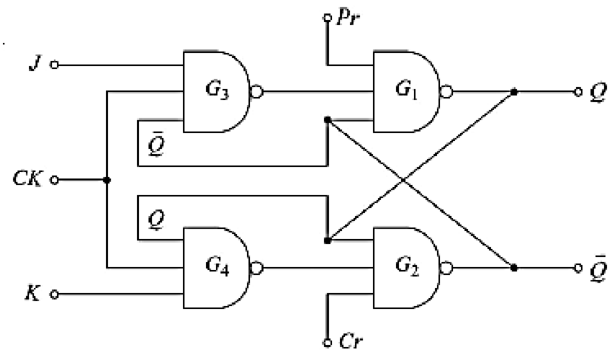
त्याचप्रमाणे जर Pr=1 आणि Cr=0 असतील तर फ्लिप फ्लॉप रीसेट केला जातो.
Pr=Cr=0 ही कंन्डिशन वापरली जाऊ नये, कारण यामुळे अनिश्चित स्थिती निर्माण होते.

2.2.4 S R फ्लिप फ्लॉपचे तोटे (Disadvantages):

• रेस कंडिशन : S Rफ्लिप-फ्लॉप रेस कंडिशनसाठी संवेदनशील असतो, जे इनपुट सिग्नलच्या वेळेतील फरकांमुळे आउटपुट स्टेट अप्रत्याशितपणे बदलते तेव्हा उद्भवते.
• अवैध स्टेट्स: जर सेट आणि रीसेट इनपुट दोन्ही एकाच वेळी सक्रिय केले गेले तरS Rफ्लिप-फ्लॉप अशा अवैध स्टेटमध्ये प्रवेश करू शकते जिथे दोन्ही आउटपुट जास्त असतात किंवा दोन्ही कमी असतात. यामुळे डिजिटल सिस्टममध्ये अनिश्चित वर्तन होऊ शकते.
• मर्यादित स्केलेबिलिटी: S R फ्लिप-फ्लॉपला अधिक कॉम्प्लेक्स डिजिटल सिस्टमपर्यंत स्केल करणे कठीण असू शकते, कारण यामुळे कॉम्प्लेक्ससिटी वाढू शकते आणि त्रुटींची शक्यता वाढू शकते.

2.2.5 J K फ्लिप-फ्लॉप

J K फ्लिप-फ्लॉपचे नाव आय सीचा शोध लावणारे इलेक्ट्रिकल इंजिनिअर जॅक किल्बी यांच्या नावावरून ठेवण्यात आले आहे. J K फ्लिप-फ्लॉपला युनिव्हर्सल प्रोग्रामेबल फ्लिप-फ्लॉप म्हणतात कारण, त्याचे इनपुट जे, के प्रीसेट आणि क्लियर वापरून, इतर कोणत्याही फ्लिप-फ्लॉपचे फंक्शन अनुकरण केले जाऊ शकते. यामध्ये J इनपुट SR फ्लिप-फ्लॉपच्या SET इनपुटसारखे असते आणि K इनपुट SR फ्लिप-फ्लॉपच्या RESET इनपुटसारखे असते. JK फ्लिप-फ्लॉपचे सिंबॉल खाली दाखवले आहे.



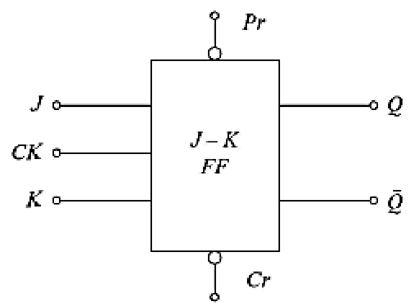


Figure 2.27 Logic symbol of Clocked S-R Flip-Flop with Preset and Clear.

आकृतीमध्ये J K फ्लिप-फ्लॉप लॉजिक आकृती दाखवली आहे. आधी सांगितल्याप्रमाणे, J K फ्लिप-फ्लॉप हे S Rफ्लिप-फ्लॉपचे सुधारित रूप आहे. लॉजिक आकृतीमध्ये तीन इनपुट नँड गेट्स आहेत जे S R फ्लिप-फ्लॉपमधील दोन इनपुट नँड गेट्सची जागा घेतात आणि इनपुट एस आणि आर मधील जे आणि के ने बदलले जातात.
J K फ्लिप-फ्लॉपची रचना अशी आहे की एका नँड गेटचे (G_3) तीन इनपुट जे आहेत, क्लॉक सिग्नल आणि क्यू Q मधील फीडबॅक सिग्नल आणि दुसऱ्या नँडचे (G_4) तीन इनपुट के आहेत, क्लॉक सिग्नल आणि क्यू बार Q' मधील फीडबॅक सिग्नल आहे .ही व्यवस्था S R फ्लिप-फ्लॉपमधील अनिश्चित स्थिती काढून टाकते.

Sr. No.	$\overline{PR}$	$\overline{CR}$	Jn	Kn	Clk	Output	
						Qn	$\overline{Q_n}$
1.	0	1	X	X	X	1	0
2.	1	0	X	X	X	0	1
3.	1	1	0	0	X	Previous Output	
4.	1	1	X	X	0	Previous Output	
5.	1	1	1	0	↓	1	0
6.	1	1	0	1	↓	0	1
7.	1	1	1	1	↓	Toggle	

Table no 2.9.The Truth table of JK Flip-Flop

- **कार्य**

कंन्डिशन 1: जर Pr=0 आणि Cr=1 असतील, तर G1 (Q) चे आउटपुट निश्चितपणे 1 असेल.
परिणामी G2 चे तिन्ही इनपुट 1 असतील जे Q‾=0 करेल. म्हणून Pr=0 बनवणे फ्लिप फ्लॉप सेट करते.
कंन्डिशन 2: त्याचप्रमाणे जर Pr=1 आणि Cr=0 असतील तर फ्लिप फ्लॉप रीसेट केला जातो.
कंन्डिशन 3: जर Pr=Cr=1, Clk शून्य असेल, तर सर्किट J-K फ्लिप फ्लॉपच्या ट्रुथ टेबलनुसार कार्य करते
कंन्डिशन 4: जेव्हा दोन्ही इनपुट J = K= 0 आणि Clk दिले जातात तेव्हा Q त्याचे मागील स्टेट व्हॅल्यू परत करते म्हणजेच ते मागील डेटा धारण करते.
जेव्हा आपण J K फ्लिप फ्लॉपवर क्लॉक पल्स लागू करतो आणि J इनपुट लो असतो तेव्हा इतर नँड NAND गेट्स काहीही असोत, नँड NAND G3 आउटपुट हाय होतो. त्याचप्रमाणे, जर K इनपुट लो असेल तर नँड NAND G4 चे आउटपुट देखील हाय असेल. त्यामुळे आउटपुट त्याच स्थितीत राहतो म्हणजेच फ्लिप फ्लॉपच्या स्थितीत कोणताही बदल होत नाही.
कंन्डिशन 5: जेव्हा J =0 आणि K =1, आणि Clk दिले जाते तेव्हा फ्लिप फ्लॉप रीसेट स्थितीत असेल म्हणजेच Q = 0, Q‾ = 1.
जेव्हा आपण J K फ्लिप फ्लॉपवर क्लॉक पल्स लावतो आणि इनपुट J लो असतात आणि K हाय असतात तेव्हा J इनपुटशी जोडलेल्या नँड NAND G3 गेटचे आउटपुट 1 होते. नंतर Q= 0 होते. हे फ्लिप फ्लॉपला पुन्हा त्याच्या मागील स्थितीत रीसेट करेल. त्यामुळे फ्लिप फ्लॉप RESET स्थितीत असेल.
कंन्डिशन 6: जेव्हा J =1 आणि K =0 असेल, तेव्हा फ्लिप-फ्लॉप सेट स्थितीत असेल म्हणजेच Q = 1, Q‾ = 0
जेव्हा आपण J K फ्लिप फ्लॉपवर क्लॉक पल्स लावतो आणि इनपुट J हाय आणि K लो असतात तेव्हा K इनपुटशी जोडलेल्या नँड NAND गेटचे आउटपुट 1 होते. नंतर Q‾= 0 होते. हे हाय क्लॉक इनपुटसह फ्लिप फ्लॉप सेट करेल. म्हणजे फ्लिप फ्लॉप SET स्थितीत असेल.
कंन्डिशन 7: जेव्हा दोन्ही इनपुट J = K =1 असतील तेव्हा फ्लिप-फ्लॉप टॉगल स्थितीत असेल. याचा अर्थ असा की आउटपुट मागील स्थितीला पूरक असेल.
J K फ्लिप फ्लॉपची रेस अराउंड कंडिशन:-
J K फ्लिप फ्लॉपमध्ये रेस अराउंड कंडिशन येते जेव्हा इनपुट:J = 1 आणि K = 1 असतात.
आपल्याला माहित आहे की, टॉगलिंग नावाची एक संकल्पना आहे. जर तुम्हाला माहित नसेल तर ते जाणून घ्या. Q आणि Q‾चे आउटपुट सिग्नल बदलत राहतात जसे की:
Q: 0 1 0 1 0 … आणि
Q‾: 1 0 1 0 ….
आता, टॉगलिंग म्हणजे वरील प्रमाणेच. परंतु, टॉगलिंग तेव्हा होते जेव्हा डिले टाईम क्लॉक पल्सच्या बरोबरीचा असतो. म्हणून, सिग्नलमध्ये फक्त एकच बदल होतो. जर सुरुवातीची स्थिती 0 असेल, तर ते टॉगल केल्याने ते 1 होईल आणि उलट. म्हणून, एका टॉगलमध्ये फक्त एकच बदल.
रेसिंगच्या बाबतीत, डिले टाईम क्लॉक पल्सपेक्षा जास्त असतो. तर, इनपुटमध्ये कोणताही बदल न होता एकाच क्लॉक पल्समध्ये आउटपुट सिग्नल खूप जलद बदलतो.
रेस अराउंड कंडिशन नियंत्रणात नाही. म्हणून, ते नियंत्रणात खरेदी करणे आवश्यक आहे. असे करण्याचे 3 मार्ग आहेत:

1. क्लॉकची पल्स वाढवा.
2. एज ट्रिगरिंग
3. मास्टर स्लेव्ह

2.2.6 डी फ्लिप फ्लॉप:-

D फ्लिप-फ्लॉपला "डिले फ्लिप-फ्लॉप" किंवा "डेटा फ्लिप-फ्लॉप" असेही म्हणतात. ते 1-बिट बायनरी डेटा साठवण्यासाठी वापरले जातात. ते डिजिटल इलेक्ट्रॉनिक्समध्ये मोठ्या प्रमाणात वापरल्या जाणाऱ्या फ्लिप-फ्लॉपपैकी एक आहेत. डिजिटल सिस्टीममध्ये मूलभूत मेमरी घटक असण्याव्यतिरिक्त, D फ्लिप-फ्लॉपला डिले लाइन घटक आणि झिरो-ऑर्डर होल्ड घटक म्हणून देखील मानले जाते.
D फ्लिप-फ्लॉपमध्ये दोन इनपुट असतात, एक क्लॉक (CLK) इनपुट आणि एक डेटा (D) इनपुट आणि दोन आउटपुट; एक मुख्य आउटपुट Q द्वारे दर्शविला जातो आणि दुसरा Q‾द्वारे दर्शविला जातो. सिंबॉल खाली दर्शविले आहे.

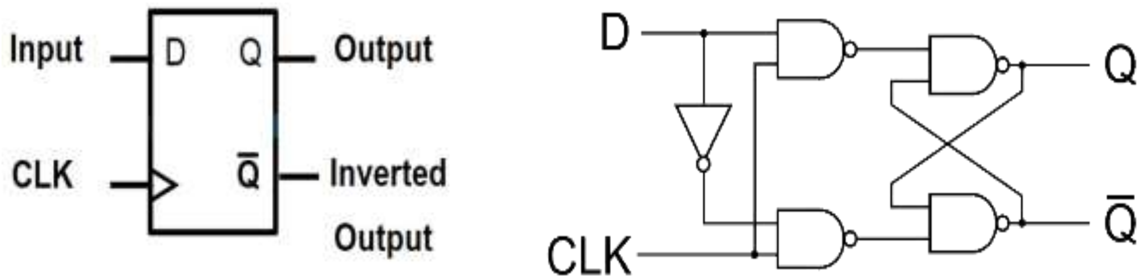


Figure 2.28 Logic symbol and Logic circuit of D Flip-Flop

- **रचना**:-

D फ्लिप-फ्लॉप हा SR फ्लिप-फ्लॉपमध्ये बदल करून तयार केला जातो. S इनपुट D इनपुटसह दिला जातो S आणि R इनपुट इंन्वरडेड D इनपुटसह दिला जातो. म्हणून D फ्लिप-फ्लॉप हा SR फ्लिप-फ्लॉपसारखाच असतो ज्यामध्ये दोन्ही इनपुट एकमेकांना पूरक असतात, त्यामुळे कोणतीही इंटर्मीडियत स्थिती उद्भवण्याची शक्यता नसते. SR फ्लिप-फ्लॉपचा मुख्य तोटा म्हणजे रेस अराउंड कंडिशन जी D फ्लिप-फ्लॉपमध्ये (इंन्वरडेड इनपुटमुळे) काढून टाकली जाते. D फ्लिप-फ्लॉपचा सर्किट खालील आकृतीमध्ये दर्शविला आहे.

- **कार्य** :-

जेव्हा आपण D फ्लिप फ्लॉपवर किंवा क्लॉक सिग्नलच्या घसरत्या फॉल्लिंग ऐज (falling edge) कोणतेही क्लॉक इनपुट लागू करत नाही, तेव्हा आउटपुटमध्ये कोणताही बदल होणार नाही. ते आउटपुट Q वर त्याचे मागील मूल्य(VALUE) कायम ठेवेल. जर क्लॉक सिग्नल रायजिंग ऐज असेल (rising edge) आणि जर D इनपुट हाय असेल, तर आउटपुट देखील हाय असेल आणि जर D इनपुट लो असेल, तर आउटपुट लो होईल. म्हणून क्लॉक सिग्नलच्या उपस्थितीत आउटपुट Q इनपुट D चे अनुसरण करतो.

CLK	D	Q	$\overline{Q}$
↓	X	Q_{PREV}	$\overline{Q}_{PREV}$
↑	1	1	0
↑	0	0	1

Table no 2.10.The Truth table of D Flip-Flop

- **क्लॉक पॉजिटिव ट्रांजिशन (Clock positive transition),**

जर D = 0 असेल तर Q = 0 फ्लिप फ्लॉप रीसेट केला जातो.
जर D = 1 असेल तर Q = 1 फ्लिप फ्लॉप सेट केला जातो.
टीप: ↑ क्लॉक पॉजिटिव ट्रांजिशन दर्शविते आणि ↓ क्लॉक निगेटीव ट्रांजिशन दर्शविते.

- **उपयोग (Applications)-**

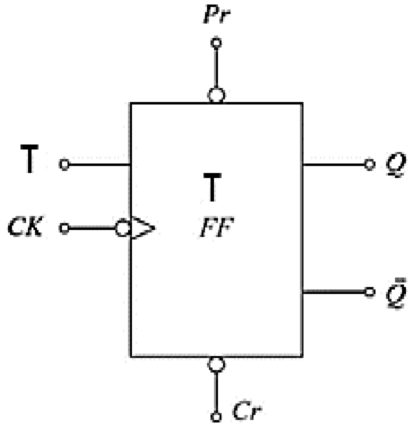
D फ्लिप-फ्लॉप हे सर्वात जास्त वापरल्या जाणाऱ्या फ्लिप-फ्लॉपपैकी एक आहे. D फ्लिप-फ्लॉपच्या अनेक उपयोगांपैकी काही आहेत

a) डेटा स्टोरेज रजिस्टर.
b) शिफ्ट रजिस्टर म्हणून डेटा ट्रान्सफर करणे.
c) फ्रिक्वेन्सी डिव्हिजन सर्किट्स.

2.2.7 T फ्लिप-फ्लॉप(T Flipflop):-

T फ्लिप-फ्लॉपला "टॉगल फ्लिप-फ्लॉप" असेही म्हणतात. SR फ्लिप-फ्लॉपमध्ये इंटरमीडिएट स्टेट टाळण्यासाठी, आपण फ्लिप-फ्लॉपला ट्रिगर इनपुट किंवा टॉगल इनपुट (T) नावाचा फक्त एक इनपुट प्रदान केला पाहिजे. नंतर फ्लिप-फ्लॉप टॉगल स्विच म्हणून काम करतो. टॉगल करणे म्हणजे 'सध्याच्या स्थितीच्या आउटपुटला कॉम्प्लिमेंट करण्यासाठी पुढील स्थितीचे आउटपुट बदलणे'. आपण JK फ्लिप-फ्लॉपमध्ये साधे बदल करून T फ्लिप-फ्लॉप डिझाइन करू शकतो. T फ्लिप-फ्लॉप हे एकल इनपुट डिव्हाइस आहे आणि म्हणूनच J आणि K इनपुट एकत्र जोडून आणि त्यांना T नावाच्या सिंगल इनपुटसह देऊन आपण JK फ्लिप-फ्लॉपला T फ्लिप-फ्लॉपमध्ये रूपांतरित करू शकतो. म्हणून T फ्लिप-फ्लॉपला कधीकधी सिंगल इनपुट JK फ्लिप-फ्लॉप असे म्हणतात.

T फ्लिप-फ्लॉपचे लॉजिक सिंबॉल, ट्रुथ टेबलसह खाली दाखवले आहे.



INPUT	OUTPUT
DN	QN+1
0	Qn
1	Qn'

Figure 2.29 Logic symbol and Truth Table of T Flip-Flop

त्यात एक टॉगल इनपुट (T) आणि एक क्लॉक सिग्नल इनपुट (CLK) आहे

D फ्लिप-फ्लॉपची सर्वात सोपी रचना JK फ्लिप-फ्लॉपसह आहे. JK फ्लिप-फ्लॉपचे J इनपुट आणि K इनपुट एकत्र जोडलेले आहेत आणि T इनपुटसह प्रदान केले आहेत. JK फ्लिप-फ्लॉपपासून बनवलेल्या T फ्लिप-फ्लॉपचे लॉजिक सर्किट खाली दर्शविले आहे.

T फ्लिप-फ्लॉप हे एक एज ट्रिगर केलेले डिव्हाइस आहे

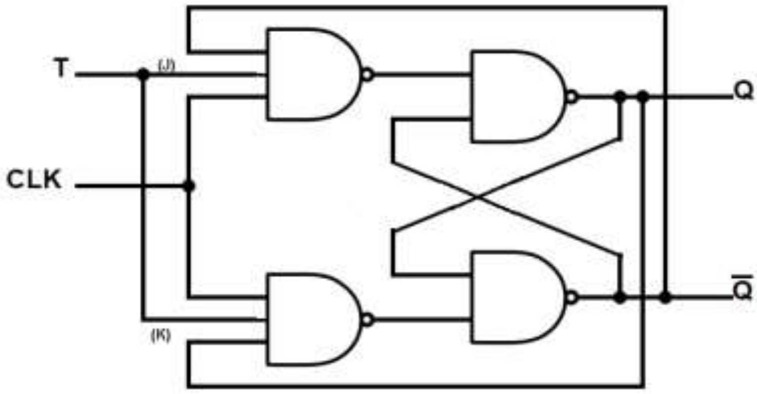


Figure 2.30 Logic circuit of T Flip-Flop

कार्य

जेव्हा T इनपुट लो असतो, तेव्हा T फ्लिप फ्लॉपचा पुढील स्टेट सद्याच्या स्थितीसारखाच असतो.

- T = 0 आणि सद्याची स्टेट = 0 नंतर पुढील स्टेट = 0
- T = 1 आणि सद्याची स्टेट = 1 नंतर पुढील स्टेट = 1

जेव्हा T इनपुट हाय असतो आणि क्लॉकच्या पॉजिटिव ट्रांजिशन मुळे T फ्लिप-फ्लॉपची पुढील स्टेट सद्याच्या स्थितीच्या उलट असते.

- T = 1 आणि सद्याची स्टेट = 0 नंतर पुढील स्टेट = 1
- T = 1 आणि सद्याची स्टेट = 1 नंतर पुढील स्टेट = 1

प्रत्येक येणारा ट्रिगर आळीपाळीने सेट आणि रीसेट इनपुट बदलत असताना, फ्लिप-फ्लॉप टॉगल होतो. म्हणून आउटपुट वेव्ह फॉर्मचे एक पूर्ण चक्र पूर्ण करण्यासाठी त्याला दोन ट्रिगरची आवश्यकता असते. याचा अर्थ असा की T फ्लिप फ्लॉप इनपुट फ्रिक्वेन्सीच्या अगदी अर्ध्या फ्रिक्वेन्सीवर आउटपुट तयार करतो. म्हणून T फ्लिप-फ्लॉप "फ्रिक्वेन्सी डिव्हायडर सर्किट" म्हणून काम करेल.

T फ्लिप-फ्लॉपचा मुख्य तोटा असा आहे की लागू केलेल्या ट्रिगर पल्सवरील फ्लिप-फ्लॉपची स्थिती मागील स्थिती माहित असतानाच कळते.

2.3 काउंटर (Counter):

- पल्स मोजण्यासाठी वापरल्या जाणाऱ्या डिजिटल सर्किटला काउंटर म्हणतात. हे एक सिक्वेनशियल सर्किट आहे.
- काउंटर हा क्लॉक सिग्नल लागू केलेल्या फ्लिप-फ्लॉपचा समूह आहे.

• काउंटर क्लॉक च्या पल्सची संख्या मोजतात. म्हणून फ्रिक्वेनसि (frequency) किंवा कालावधी (time period) मोजण्यासाठी वापरले जाऊ शकते.

काउंटरचे प्रकार:

काउंटर मुळात दोन प्रकारचे असतात:

1. असिंक्रोनस किंवा रिपल काउंटर.
2. सिंक्रोनस काउंटर.

1. असिंक्रोनस किंवा रिपल काउंटर: या काउंटरसाठी बाह्य क्लॉक सिग्नल(external cliock signal) एका फ्लिप-फ्लॉपवर लावला जातो आणि नंतर मागील फ्लिप-फ्लॉपचा आउटपुट पुढील फ्लिप-फ्लॉपच्या क्लॉक शी जोडला जातो.

2. सिंक्रोनस काउंटर: सिंक्रोनस काउंटरमध्ये सर्व फ्लिप-फ्लॉप एकाच वेळी बाह्य क्लॉक (external cliock signal) चे पल्स प्राप्त करतात. रिंग काउंटर आणि जॉन्सन काउंटर ही सिंक्रोनस काउंटरची उदाहरणे आहेत.

2.3.1 असिंक्रोनस किंवा रिपल काउंटर:

आकृती 2.31 मध्ये T फ्लिप-फ्लॉप वापरून 4 बिट असिंक्रोनस काउंटरचा सर्किट आकृती दाखवली आहे.

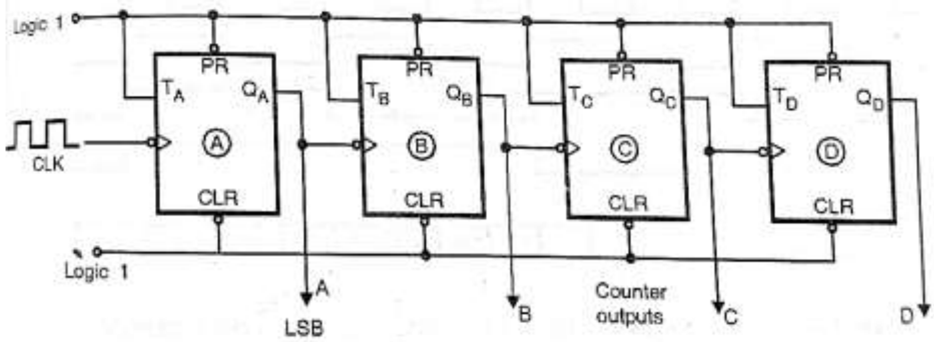


Fig. 2.31: 4 bit asynchronous up counter

हा 4 बिट (4 Bit) रिपल अप काउंटर असल्याने, आपल्याला चार फ्लिप फ्लॉप वापरावे लागतील. सुरुवातीला सर्व फ्लिपफ्लॉपचे आउटपुट शून्य असते. $Q_DQ_CQ_BQ_A = 0000$

सर्व फ्लिपफ्लॉप निगेटिव्ह एज ट्रिगर केलेले असतात. CLK FF-A च्या क्लॉक इनपुटवर लागू होतेजिथे प्रत्येक F/F चे Q आउटपुट पुढील F/F च्या क्लॉक इनपुटवर लागू केले जातात.

तक्ता 2.11 4बिट असिंक्रोनस अप काउंटरसाठी ट्रयुथ टेबलदर्शविते. त्याचे आउटपुट 16 अवस्थांमधून जाते म्हणजेच (0000)-$(1)_{10}$ ते (1111)- $(15)_{10}$.

1111 नंतर, आउटपुट पुन्हा 0000 होतात आणि कार्यस्वतःची पुनरावृत्ती होते.

State	Q_D	Q_C	Q_B	Q_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
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
0	0	0	0	0

Table no 2.11.The Truth table of 4 bit asynchronous up counter

आकृती 2.32 मध्ये 4-बिट असिंक्रोनस अप काउंटरसाठी टाइमिंग आकृती दाखवली आहे.
आउटपुटच्या MSB म्हणून Q_D तर LSB म्हणून Q_A काम करते.

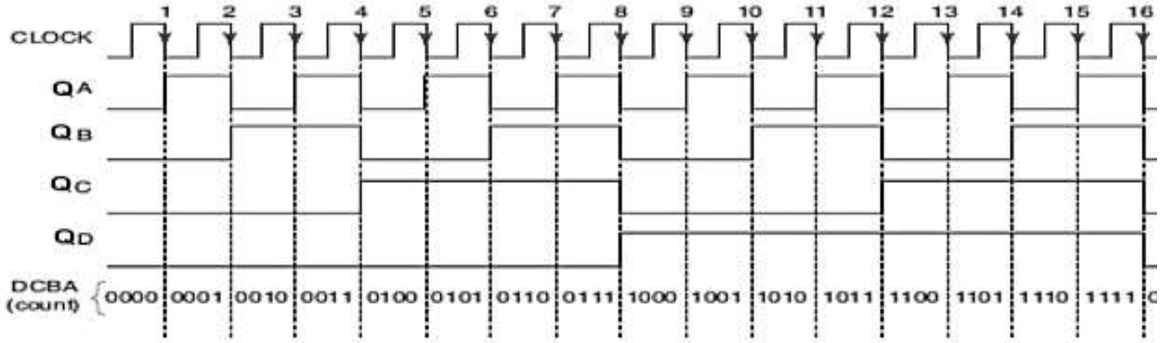


Fig. 2.32 Timing diagram for the 4-bit asynchronous up counter.

- असिंक्रोनस काउंटरचा वापर

1. फ्रिक्वेन्सी डिव्हिजन,
2. डिजिटल क्लॉक,
3. इव्हेंट काउंटिंग.

2.3.2 सिंक्रोनस काउंटर

जर "क्लॉक च्या पल्स एकाच वेळी काउंटरमधील सर्व फ्लिप-फ्लॉपवर लागू केल्या जातात", तर अशा काउंटरला सिंक्रोनस काउंटर म्हणतात.

बायनरी 4-बिट सिंक्रोनस अप काउंटर

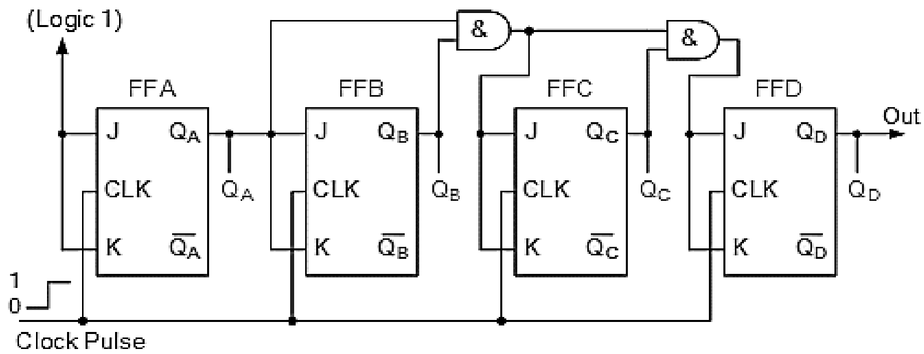


Figure 2.32 ; Binary 4-bit Synchronous Up Counter

एक्सटर्नल क्लॉक चे पल्स (मोजायचे पल्स) काउंटर चेनमधील प्रत्येक J-K FF ला थेट दिले जातात आणि J आणि K दोन्ही इनपुट टॉगल मोडमध्ये एकत्र जोडलेले असतात, परंतु फक्त पहिल्या फ्लिप-फ्लॉप, फ्लिप-फ्लॉप FFA (LSB) मध्ये ते HIGH, लॉजिक "1" ने जोडलेले असतात ज्यामुळे फ्लिप-फ्लॉप प्रत्येक क्लॉक च्या पल्सवर टॉगल करू शकतो. नंतर सिंक्रोनस काउंटर सामान्य क्लॉक च्या सिग्नलला प्रतिसाद म्हणून पूर्वनिर्धारित स्थितींचा क्रम अनुसरण करतो, प्रत्येक पल्ससाठी एक स्थिती पुढे नेतो.

फ्लिप-फ्लॉप FFB चे J आणि K इनपुट फ्लिप-फ्लॉप FFA च्या आउटपुट Q_A शी थेट जोडलेले असतात, परंतु फ्लिप-फ्लॉप FFC आणि FFD चे J आणि K इनपुट वेगळ्या AND गेट्सवरून चालवले जातात ज्यांना मागील टप्प्यातील इनपुट आणि आउटपुटमधून सिग्नल देखील पुरवले जातात. हे अतिरिक्त AND गेट्स पुढील टप्प्यातील JK इनपुटसाठी आवश्यक लॉजिक तयार करतात.

जर आपण प्रत्येक JK फ्लिप-फ्लॉपला मागील सर्व फ्लिप-फ्लॉप आउटपुट (Q) "उच्च(High)" आहेत की नाही यावर आधारित टॉगल करण्यास एनेबल(ENABLE) केले तर आपल्याला असिंक्रोनस सर्किट प्रमाणेच मोजणी क्रम मिळू शकतो परंतु रिपल इफेक्टशिवाय, कारण या सर्किटमधील प्रत्येक फ्लिप-फ्लॉप अगदी त्याच वेळी क्लॉक केला जाईल. मग सिंक्रोनस काउंटरमध्ये कोणताही इन्हेरेंट प्रोपोगेशन डिले (no inherent propagation delay)नसल्यामुळे, सर्व काउंटर स्टेज एकाच वेळी समांतरपणे ट्रिगर केले जातात, या प्रकारच्या फ्रिक्वेन्सी काउंटरची कमाल ऑपरेटिंग वारंवारता समान असिंक्रोनस काउंटर सर्किटपेक्षा खूपच जास्त असते.

4-बिट सिंक्रोनस काउंटर वेव्हफॉर्म टाइमिंग डायग्राम

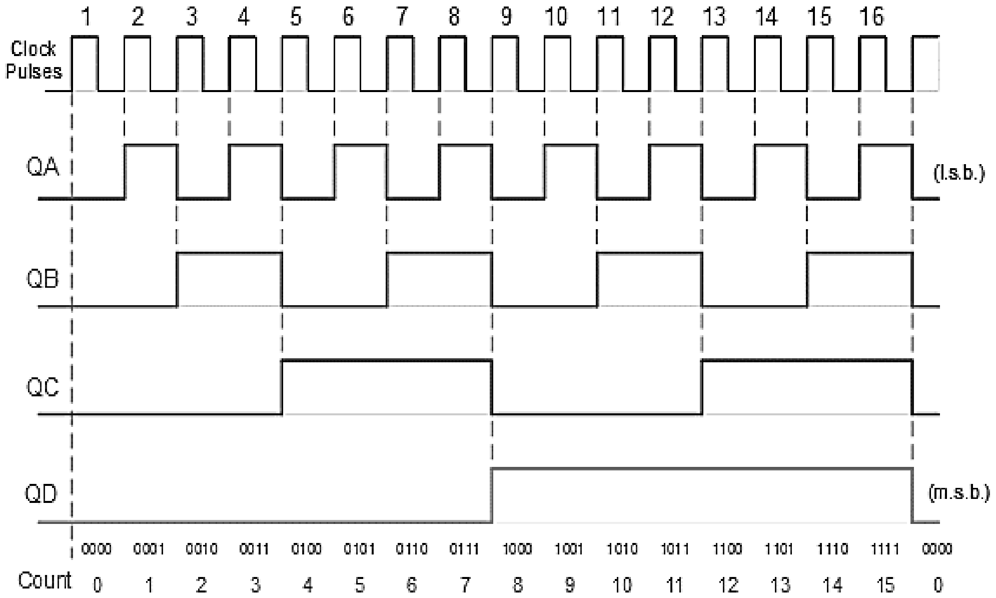


Figure 2.33 Timing diagram of 4-bit Synchronous Counter

कारण हा 4-बिट सिंक्रोनस काउंटर प्रत्येक क्लॉक च्या पल्सवर क्रमाने मोजला जातो, परिणामी आउटपुट 0 (0000) ते 15 (1111) पर्यंत मोजले जातात. म्हणून, या प्रकारच्या काउंटरला 4-बिट सिंक्रोनस अप काउंटर असेही म्हणतात.

Clock	QD	QC	QB	QA
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Table no 2.12.The Truth table of 4 bit synchronous up counter

प्रश्न (Questions):

1. 4:1 मल्टीप्लेक्सरचा लॉजिक आकृती काढा आणि त्याचा ट्रयुथ टेबल द्या.
2. J K फ्लिप-फ्लॉप काढा आणि त्याचा ट्रयुथ टेबल लिहा.
3. प्रीसेटसह क्लॉक केलेला J K फ्लिप-फ्लॉप काढा आणि त्याचा ट्रयुथ टेबल लिहा.
4. फ्लिप-फ्लॉप वापरून 4-बिट सिंक्रोनस काउंटर तयार करा. त्याचा टाइमिंग आकृती देखील काढा.
5. काउंटरचे अनुप्रयोग लिहा
6. फ्लिप-फ्लॉप वापरून असिंक्रोनस काउंटर किंवा रिपल काउंटर तयार करा. त्याचा टाइमिंग आकृती देखील काढा.
7. 1:8 डीमल्टीप्लेक्सरचा लॉजिक आकृती काढा आणि त्याचा ट्रयुथ टेबलद्या.
8. मल्टीप्लेक्सरचे अनुप्रयोग लिहा.

संदर्भ पाठ्यपुस्तके (Reference Books):

Sr No.	Author	Title	Publisher with ISBN Number
1	R.P. Jain	Modern Digital Electronics	McGraw-Hill Publishing, New Delhi, 2009; ISBN: 9780070669116
2	V.K.Puri	Digital Electronics	McGraw Hill Education (1 July 2017); ISBN-13 : 978-0074633175
3	Salivahanan S.; Arivazhagan S.	Digital Circuits and Design	Oxford University Press India; 5th edition ; ISBN13- 978-0199488681

माहती संकेतस्थळ (Reference Links)

1. https://www.tutorialspoint.com/digital-electronics/digital-electronics-multiplexers.htm
2.https://www.javatpoint.com
3.https://electricalfundablog.com
4. **Diagram Courtesy** https://www.tpointtech.com/multiplexer-digital-electronics
5. **Diagram Courtesy** Fig. 2.31: 4 bit asynchronous up counter from Digital electronics by R.P. Jain Book.

यूनिट – III
8051 मायक्रोकंट्रोलर आर्किटेक्चर
(8051 Microcontroller Architecture)

विषय निष्पत्ती (Course outcome)

CO3 - 8051 मायक्रोकंट्रोलरमधील विविध रजिस्टर्स ॲक्सेस करणे.
(Access various registers in 8051 microcontroller.)

विषय निष्पत्ती बरोबर जुळणारी थिअरी लर्निंग निष्पत्ती (Theory Learning Outcomes TLO) .

TLO 3.1 दिलेल्या पॅरामीटर्ससाठी मायक्रोकंट्रोलर आणि मायक्रोकॉम्प्युटरच्या प्रमुख वैशिष्ट्यांची तुलना करा.
(Compare the salient features of microcontroller and microcomputer for the given parameters.)
TLO 3.2 दिलेल्या पॅरामीटर्सवर विविध्य प्रकारच्या आर्किटेक्चरची तुलना करणे.
(Compare the given types of architecture on the given parameters.)
TLO 3.3 8051च्या रजिस्टर्सचे वर्णन करणे.
(Describe the given types of registers of 8051.)
TLO 3.4 8051 मध्ये मेमरीचा वापर कसा करावा याचे समायोजित (justify) करणे.
(Justify the use of the given type of memory in 8051.)

3.1 मायक्रोकॉम्प्युटर आणि मायक्रोकंट्रोलर (मूलभूत परिचय आणि तुलना) Microcomputer and microcontroller (basic introduction and comparison)

मायक्रोप्रोसेसर:

मायक्रोप्रोसेसर ही एक प्रोग्रामेबल चिप आहे जी वापरकर्त्याद्वारे नियंत्रित केलेली असते ज्यामध्ये संगणकाच्या सेंट्रल प्रोसेसिंग यूनिट प्रमाणेच संगणन (computing) आणि निर्णय घेण्याची क्षमता असते.

मायक्रोप्रोसेसर जवळजवळ सर्व प्रकारच्या इलेक्ट्रॉनिक उपकरणांमध्ये जसे की भ्रमणध्वनि (मोबाइल फोन),प्रिंटर, वॉशिंग मशीन इत्यादींमध्ये वापरले जाते. मायक्रोप्रोसेसरचा वापर रडार, कृत्रिम उपग्रह (सॅटलाइट) यासारख्या प्रगत अनुप्रयोगांमध्ये (applications) देखील केला जातो.

इलेक्ट्रॉनिक उद्योगातील जलद प्रगती आणि उपकरणांच्या मोठ्या प्रमाणात एकत्रीकरणामुळे खर्चात लक्षणीय घट होते आणि मायक्रोप्रोसेसर आणि त्यांच्या डेरिव्हेटिव्ह्जचा(derivatives) वापर वाढतो.

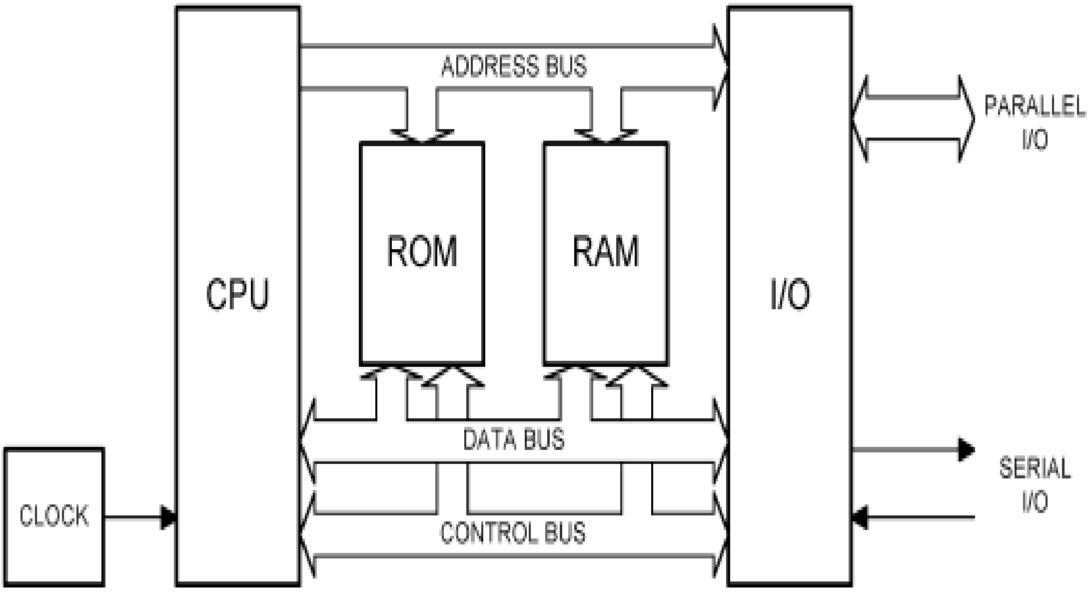


Figure3.1- Basic block diagram of Microprocessor

मायक्रोप्रोसेसरमध्ये इनस्टरकशन (instructions) एक्सझीक्यूट करण्यासाठी कंट्रोल युनिटसह अरेथम्याथिक आणि लॉजिक युनिट (Arithmetic and Logic Unit) असते. सर्व मायक्रोप्रोसेसर स्टोअर-प्रोग्राम संकल्पना वापरतात. स्टोअर-प्रोग्राम संकल्पनामध्ये, प्रोग्राम किंवा इनस्टरकशन क्रमाने एक्सझीक्यूट करण्यासाठी मेमरी मध्ये स्टोर केल्या जातात. मायक्रोप्रोसेसर वापरून कोणतेही कार्य करण्यासाठी, ते वापरकर्त्याने प्रोग्राम केले पाहिजे.

म्हणून, प्रोग्रामरला त्याच्या अंतर्गत संसाधनांचा (resources) , वैशिष्ट्यांचा आणि समर्थित इनस्टरकशन(supported instructions) बद्दल कल्पना असणे आवश्यक आहे.
प्रत्येक मायक्रोप्रोसेसरमध्ये इनस्टरकशनचा एक सेट असतो, जी उत्पादकाद्वारे प्रदान केली जाते. मायक्रोप्रोसेसरचा इनस्टरकशन सेट दोन स्वरूपात प्रदान केला जातो: बायनरी मशीन कोड(Binary macine code) आणि नेमॉनिकस (mnemonics).
मायक्रोप्रोसेसर बायनरी नमबर्स 0 आणि 1 मध्ये कम्युनिकेट आणि ऑपरेट करतो. बायनरी पॅटर्नच्या स्वरूपात इनस्टरकशन सेटला मशीन लँगवेज(Machine Language) म्हणतात आणि ती यूजरला समजणे कठीण आहे. म्हणून, बायनरी पॅटर्नना संक्षिप्त नावे दिली जातात, ज्याला नेमॉनिकस (mnemonics) म्हणतात, जी असेंब्ली लँगवेज बनवते. असेंब्ली-लेव्हल लँगवेज(Assembly Level Language) चे बायनरी मशीन-लेव्हल लँगवेज मध्ये रूपांतर असेंबलर नावाच्या अनुप्रयोगाचा (application) वापर करून केले जाते.

मायक्रोकंट्रोलर

मायक्रोकंट्रोलर (MCU) हा एका इंटीग्रेटेड सर्किटवर असलेला एक संगणक आहे जो इलेक्ट्रॉनिक सिस्टीममधील विशिष्ट कार्ये नियंत्रित करण्यासाठी डिझाइन केलेला आहे. मायक्रोकंट्रोलर, सेंट्रल प्रोसेसिंग युनिट (CPU), मेमरी आणि इनपुट/आउटपुट इंटरफेसची कार्ये एकाच चिपवर एकत्रित करतो.
मायक्रोकंट्रोलरचा वापर घरगुती उपकरणे, ऑटोमोटिव्ह सिस्टीम, वैद्यकीय उपकरणे आणि औद्योगिक नियंत्रण सिस्टम यासारख्या एम्बेडेड(Embedded) सिस्टीममध्ये मोठ्या प्रमाणात केला जातो. ते गेमिंग सिस्टीम, डिजिटल कॅमेरे आणि ऑडिओ प्लेअर्स सारख्या ग्राहक इलेक्ट्रॉनिक्स (consumer electronics)उत्पादनांमध्ये देखील वापरले जातात.

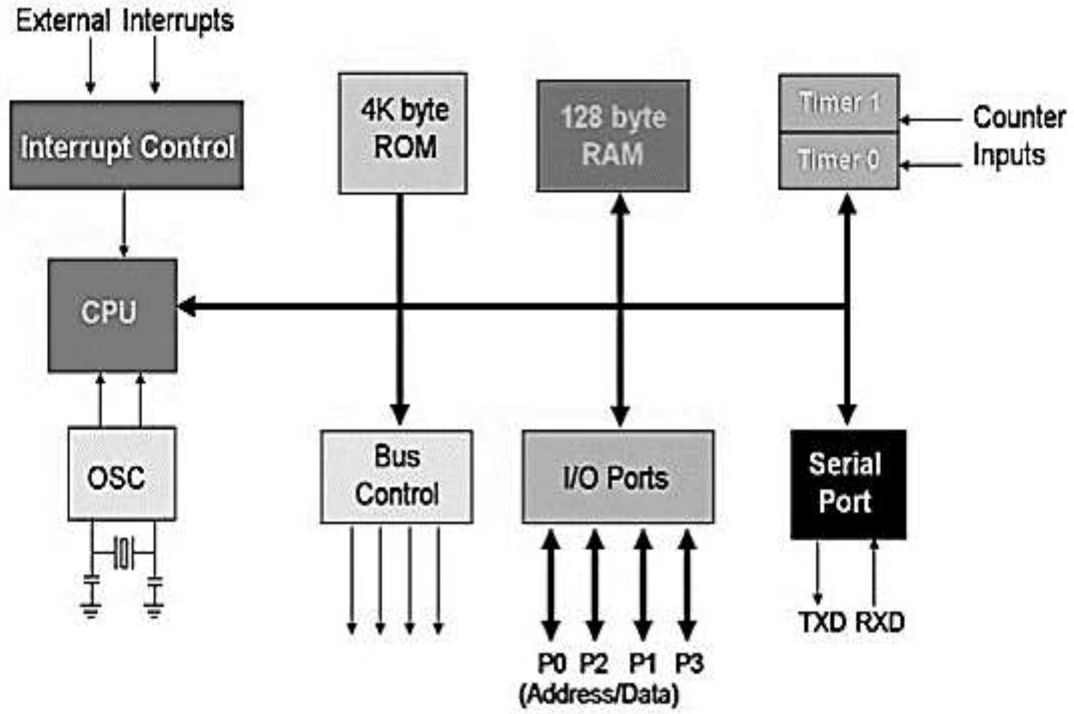


Figure 3.2- Block diagram for Microcontroller

मायक्रोकंट्रोलरमध्ये प्रोसेसर कोर, वोलाटाइल आणि नॉन-वोलाटाइल (volatile and non -volatile) मेमरी, इनपुट/आउटपुट पेरिफेरल्स(Peripherals) आणि विविध कम्युनिकेशन इंटरफेस(Interface) असतात. प्रोसेसर कोर, इनस्टरकशन एक्सझीक्यूशन(Instruction Execution) आणि मायक्रोकंट्रोलरच्या इतर घटकांवर नियंत्रण ठेवण्यासाठी जबाबदार असतो. मेमरीचा वापर डेटा आणि प्रोग्राम कोड स्टोर करण्यासाठी केला जातो, तर इनपुट/आउटपुट पेरिफेरल्सचा वापर एक्सटर्नल(External) वातावरणाशी संवाद साधण्यासाठी केला जातो.
मायक्रोकंट्रोलर प्रोग्रामेबल करण्यायोग्य असतात, याचा अर्थ असा की ते विशिष्ट कार्ये करण्यासाठी कस्टमाइज (Customised) केले जाऊ शकतात. मायक्रोकंट्रोलर कोड लिहिण्यासाठी वापरल्या जाणाऱ्या प्रोग्रामिंग लँगवेज (Programming Language), उत्पादक आणि मायक्रोकंट्रोलरच्या प्रकारानुसार बदलतात. सामान्यतः वापरल्या जाणाऱ्या काही प्रोग्रामिंग लँगवेज मध्ये C, C++ आणि असेंब्ली लँगवेज समाविष्ट आहेत.
मायक्रोप्रोसेसर आणि मायक्रोकंट्रोलरची तुलना:

पॅरामिटर्स	मायक्रोकंट्रोलर	मायक्रोप्रोसेसर
उद्देश	विशिष्ट एम्बेडेड सिस्टम अनुप्रयोगांसाठी डिझाइन करण्यासाठी वापर	सामान्य उद्देश (general-purpose) संगणकीय अनुप्रयोगांसाठी डिझाइन करण्यासाठी वापर
आर्किटेक्चर	ऑन-बोर्ड (on-board) मेमरी, पेरिफेरल्स आणि I/O इंटरफेस असलेली सिंगल-चिप संगणक सिस्टम	लो ऑन-बोर्ड (on board) मेमरी, पेरिफेरल्स आणि I/O इंटरफेससह CPU
इंटीग्रेटेड लेवल	हायली इंटीग्रेटेड	लो इंटीग्रेटेड
सिस्टम आर्किटेक्चर	सिंगल-चिप सिस्टम	CPU + सपोर्ट चिप्स
प्रोसेसिंग पॉवर	कमी प्रोसेसिंग शक्तीची आवश्यकता आहे	हाय प्रोसेसिंग शक्तीची आवश्यकता आहे
ऑन-बोर्ड मेमरी	ऑन-चिप मेमरी	कोणतीही ऑन-बोर्ड मेमरी नाही
इनपुट/आउटपुट (I/O)	अधिक I/O पोर्ट्स	कमी I/O पोर्ट्स
पेरिफेरल उपकरणे	ऑन-बोर्ड पेरिफेरल्स (Peripherals)	बाह्य (external) पेरिफेरल्स (Peripherals)
खर्च	कमी खर्च	जास्त खर्च
वीज वापर	कमी वीज वापर	जास्त वीज वापर
अनुप्रयोग (Application)	एम्बेडेड सिस्टम	सामान्य अनुपयोग (General Application)
डेवलपमेंट	उत्पादकांनी प्रदान केलेले इंटीग्रेटेडडेवलपमेंट एनविरोनमेंट (IDE), विशेष प्रोग्रॅमिंग लँगवेज आणि रिसोर्सेस	स्टँडर्ड डेवलपमेंट टूल्स आणि C, C++ आणि असेंब्ली लँगवेज.
क्लॉक स्पीड (Clock Speed)	कमी क्लॉक स्पीड,100 MHz पेक्षा कमी	उच्च क्लॉक स्पीड, 1 GHz पेक्षा जास्त

Table 3.1- Comparison of Microcontroller and Microprocessor

3.2 बसेसचे प्रकार: ॲड्रेस बस, डेटा बस आणि कंट्रोल बस. हार्वर्ड आणि वॉन-न्यूमन (Harvard and Von-Neumann) आर्किटेक्चर. (Types of buses, address bus, data bus and control bus. Harvard and Von-Neumann architecture.)

बसेसचे प्रकार:

- ॲड्रेस बस(Address bus)
- डेटा बस (Data Bus)
- कंट्रोल बस (Control Bus)

मायक्रोप्रोसेसर सिस्टीममध्ये, प्रोसेसर, मेमरी आणि पेरिफेरल डिव्हाइसेसमध्ये डेटा ट्रान्सफर सक्षम (efficient) करण्यासाठी बसेस(Buses) हे कम्युनिकेशन चॅनेल म्हणून वापरले जातात.

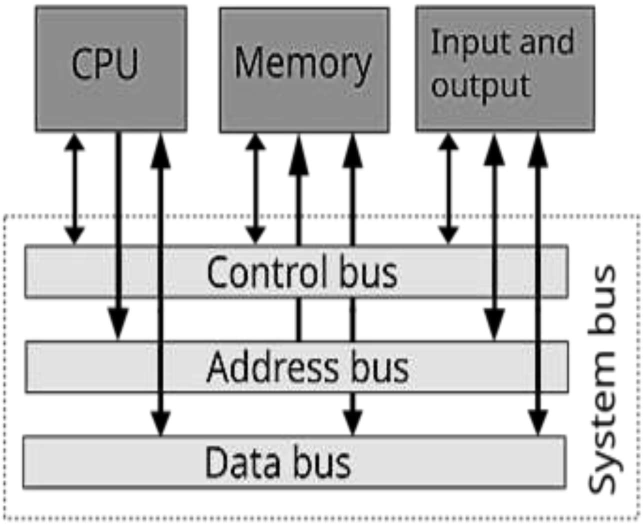


Figure 3.3- Types of Buses

बसेसचे तीन मुख्य प्रकार आहेत, प्रत्येकाचा स्वतःचे विशिष्ट उद्देश आहे:

3.2.1ॲड्रेस बस (Address Bus):

a. फंक्शन: प्रोसेसरमधून मेमरी ॲड्रेसेस इतर घटकांना पाठवते, जे डेटाचे मूळ किंवा गंतव्यस्थान(Destination) दर्शवते.

b. दिशात्मकता (Directionality) : एकदिशात्मक (Unidirectional) ; ॲड्रेसेस CPU मधून मेमरी किंवा इनपुट/आउटपुट डिव्हाइसेसमध्ये जातात.

c. विड्थ(Width) : ॲड्रेस लाइन्स ची संख्या जास्तीत जास्त ॲड्रेस करण्यायोग्य मेमरी दर्शवते. उदाहरणार्थ, 16 -बिट ॲड्रेस बस 64 KB पर्यंत मेमरी ॲक्सेस(Access) करू शकते.

3.2.2 डेटा बस (Data Bus):

a. फंक्शन: प्रोसेसर, मेमरी आणि पेरिफेरल दरम्यान डेटा ट्रान्सफर करतो.

b. दिशात्मकता (Directionality): द्विदिशात्मक (Bidirectional); डेटा, CPU कडे आणि CPU कडून दोन्हीकडे जाऊ शकतो.

c. विड्थ(Width): डेटा लाइन्स ची संख्या (जसे की 8,16, 32, किंवा 64) एकाच वेळी ट्रान्सफर केलेल्या डेटाचे प्रमाण ठरवते. विस्तृत (broader) डेटा बस एकाच वेळी अधिक डेटा ट्रान्सफर करतात, ज्यामुळे सिस्टम ची कार्यक्षमता सुधारते.

3.2.3 कंट्रोल बस (Control Bus):

a. फंकशन: रीड/राइट कमांड आणि इंटरप्ट रिक्वेस्टसह (interrupt request) सिस्टम ऑपरेशन्सचे निरीक्षण आणि सिंक्रोनाइझ (synchronize) करणारे नियंत्रण सिग्नल प्रसारित करणे.

b. दिशात्मकता (Directionality): सामान्यतः द्विदिशात्मक (bidirectional); नियंत्रण सिग्नल CPU मधून इतर घटकांकडे आणि परत CPU कडे प्रवास करू शकतात.

c. सिग्नल(Signal): रीड/राइट(Read/Write) क्रिया, इंटरप्ट व्यवस्थापन,(Interrupt Management) आणि क्लॉक सिग्नल, इत्यादींसाठी लाइन्स समाविष्ट आहेत.

3.2.4 वॉन-न्यूमन (Von-Neumann) आणि हार्वर्ड (Harvard) आर्किटेक्चर:

वॉन-न्यूमन आर्किटेक्चर:

वॉन-न्यूमन आर्किटेक्चर एक संगणक डिझाइन मॉडेल आहे ज्यात एक प्रोसेसिंग युनिट आणि इनस्टरकशन आणि डेटासाठी एक स्वतंत्र स्टोरेज स्ट्रक्चर (storage structure) वापरली जाते. 'स्टोरेज' किंवा 'स्टोर्ड प्रोग्राम संगणक' हा शब्द या डिझाइनचा संगणक दर्शविण्यासाठी वापरला जातो, कारण आधुनिक संगणके सहसा या प्रकारच्या असतात. प्रोसेसर आणि मेमरी दरम्यान इनस्टरकशन साठी आणि डेटासाठी एकच कॉमन मेमरी, एक डेटा बस आणि एक ॲड्रेस बस असते. इनस्टरकशन आणि डेटा अनुक्रमाने आणावे लागतात, ज्यामुळे कार्यात्मक (operating) बँडविड्थ मर्यादित होते.

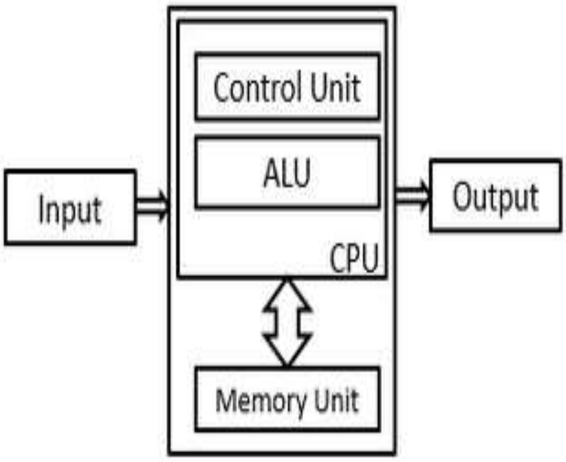


Figure3.4- Von-Neumann Architecture

- **हार्वर्ड आर्किटेक्चर(Harvard Architecture)**:

हार्वर्ड आर्किटेक्चर ही एक संगणक आर्किटेक्चर मॉडेल आहे जी इनस्टरकशन साठी आणि डेटासाठी स्वतंत्र मेमरी आणि बस वापरते. व्हॉन न्यूमन आर्किटेक्चरच्या विपरीत(Unlike), ज्यामध्ये इनस्टरकशन आणि डेटा यासाठी एकच मेमरी आणि बस असते, हार्वर्ड आर्किटेक्चरमध्ये इनस्टरकशन मध्ये आणि डेटामध्ये एकाच वेळी प्रवेश करण्याची सुविधा असते, ज्यामुळे कार्यक्षमता लक्षणीयरीत्या वाढते. या आर्किटेक्चरमध्ये इनस्टरकशन साठी आणि डेटासाठी स्वतंत्र मेमरी युनिट्स आणि स्वतंत्र बस असतात, ज्यामुळे समांतर प्रक्रिया (parallel processing) आणि जलद कार्यक्षमता मिळते. कारण इनस्टरकशन आणि डेटा एकाच बससाठी स्पर्धा करत नाहीत, CPU इनस्टरकशन मिळवताना एकाच वेळी डेटा रीड किंवा राइट करू शकतो. याशिवाय, इनस्टरकशन साठी आणि डेटासाठी वेगवेगळ्या मेमरी आकार आणि तंत्रज्ञान वापरण्याची परवानगी देते, ज्यामुळे मेमरीचे अधिक चांगले ऑप्टिमायझेशन (Optimization)होते. हार्वर्ड आर्किटेक्चर डिजिटल सिग्नल प्रोसेसिंग (DSP) प्रणाली आणि मायक्रोकंट्रोलर्समध्ये मोठ्या प्रमाणावर वापरली जाते, जिथे गती (speed) आणि कार्यक्षमता (efficiency) अत्यंत महत्त्वाची असते.

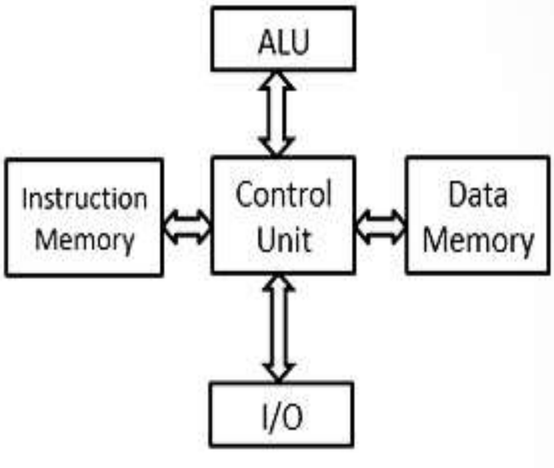
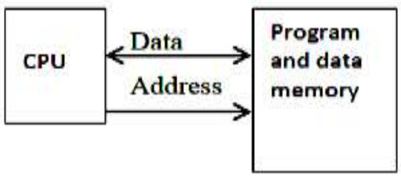
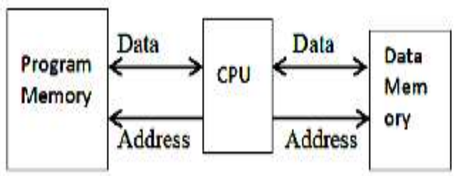


Figure3.5- Harvard Architecture

हार्वर्ड आर्किटेक्चर आणि वॉन न्यूमन आर्किटेक्चर मध्ये तुलना (Comparison Harvard and Von-Neumann Architecture)

Table 2.2- Von-Neumann vs Harvard Architecture

वैशिष्ट्ये	**वॉन न्यूमन आर्किटेक्चर**	**हार्वर्ड आर्किटेक्चर**
आर्किटेक्चर		

वैशिष्ट्ये	**वॉन न्यूमन आर्किटेक्चर**	**हार्वर्ड आर्किटेक्चर**
मेमरी रचना	इनस्टरकशन आणि डेटासाठी एकच मेमरी	इनस्टरकशन आणि डेटासाठी स्वतंत्र मेमरी
डेटा आणि इनस्टरकशन पाथ (Path)	डेटा आणि इनस्टरकशन साठी एकत्रित बस	डेटा आणि इनस्टरकशन साठी स्वतंत्र बस
एक्सेक्यूशन स्पीड (Execution Speed)	बस कनटेनशन (contention) मुळे एक्सेक्यूशन स्पीड स्लो (Bottle neck)	समांतर डेटा आणि इनस्टरकशन प्रवेशामुळे एक्सेक्यूशन स्पीड फास्ट. (Parallel Data and Instruction Access)
कॉम्प्लेक्ससिटि (Complexity)	सोपे आणि कॉस्ट इफेक्टिव	अधिक गुंतागुंतीची आणि महाग
फलेक्सिबिलिटी (Flexibility)	बदल करणे सोपे	बदल करणे कठीण
उदाहरणे	सामान्य-उद्देशीय संगणक (उदा. पारंपारिक PC, सर्व्हर)	एम्बेडेड सिस्टम्स , DSPs, आणि मायक्रोकंट्रोलर्स

3.3 8051 मायक्रोकंट्रोलर आर्किटेक्चर (8051 Microcontroller Architecture)

8051 ची वैशिष्ट्ये:

1. 8-बिट मायक्रोकंट्रोलर
2. 4KB ऑन-चिप प्रोग्राम मेमरी (ROM)
3. 128 बाइट्स ऑन-चिप डेटा मेमरी (RAM)
4. चार रजिस्टर बँक
5. 8-बिट द्वि-दिशात्मक (bidirectional) डेटा बस
6. 16-बिट एकदिशात्मक (unidirectional) ॲड्रेस बस
7. 32 सामान्य उद्देश रजिस्टर(General Purpose Registers), प्रत्येकी 8-बिट
8. 16-बिट टायमर्स
9. तीन अंतर्गत आणि दोन बाह्य इंटरप्ट्स(three internal and two external Interrupts)
10. चार 8-बिट पोर्ट
11. 16-बिट प्रोग्राम काउंटर (PC) आणि डेटा पॉइंटर (DPTR)
12. 8051 मध्ये विशेष कार्यकारी रजिस्टर (SFR) असतात, जसे की टायमर कंट्रोल [TCON], टायमर मोड [TMOD], सिरीयल कंट्रोल [SCON], इंटरप्ट एनेबल [IE], इंटरप्ट प्रायोरिटी [IP] इत्यादी.

- **8051 आर्किटेक्चर:**

8051 मायक्रोकंट्रोलर हा 1980 मध्ये Intel द्वारे विकसित करण्यात आला आणि तो एम्बेडेड सिस्टममध्ये मोठ्या प्रमाणावर वापरला जातो. हे आर्किटेक्चर Harvard आर्किटेक्चर वर आधारित आहे , ज्याचा अर्थ प्रोग्राम आणि डेटा स्टोर करण्यासाठी वेगवेगळ्या मेमरी असतात.

8051 आर्किटेक्चरमध्ये मुख्य घटक समाविष्ट आहेत:

a. सेंट्रल प्रोसेसिंग यूनिट (CPU)
b. मेमरी (ROM आणि RAM)
c. इनपुट/आउटपुट (I/O) पोर्ट
d. टायमर्स
e. सिरीयल कम्युनिकेशन इंटरफेस(Serial Communication Interface)
f. इंटरप्ट नियंत्रण प्रणाली (Interrupt Control)

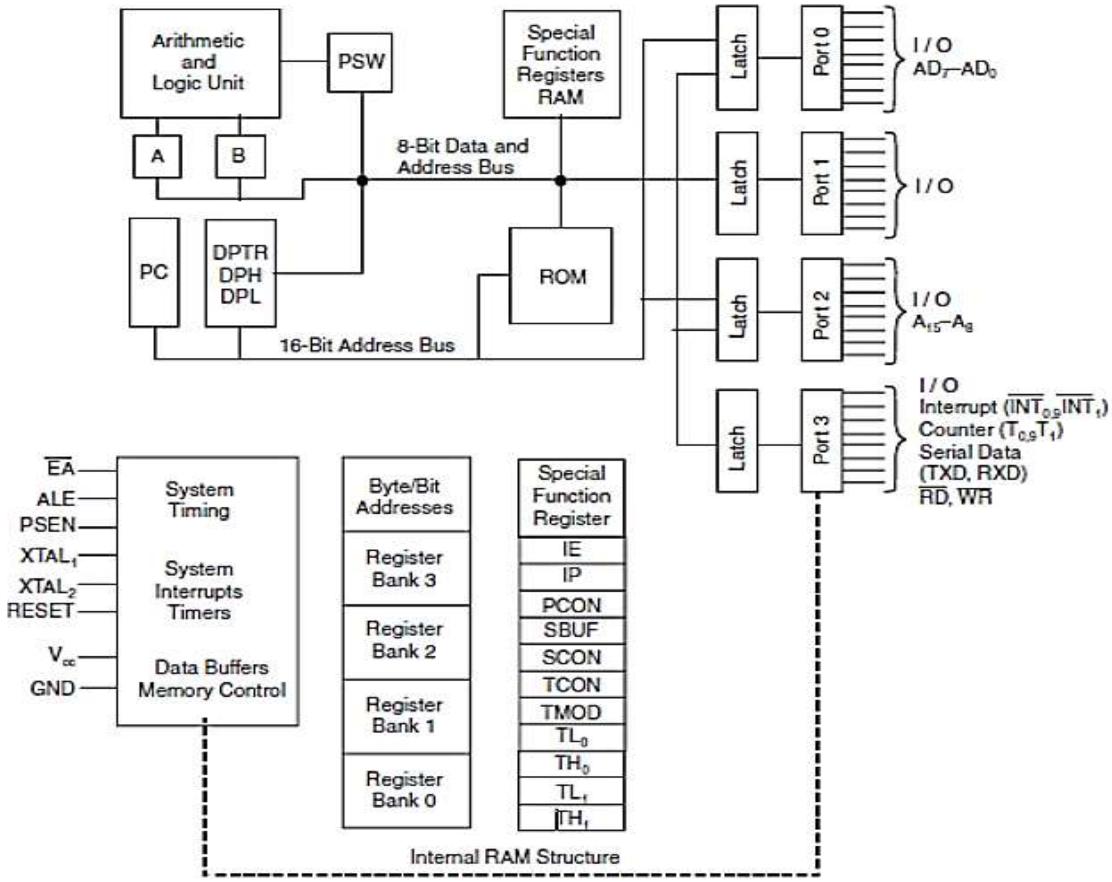


Figure 3.6- 8051 Block Diagram

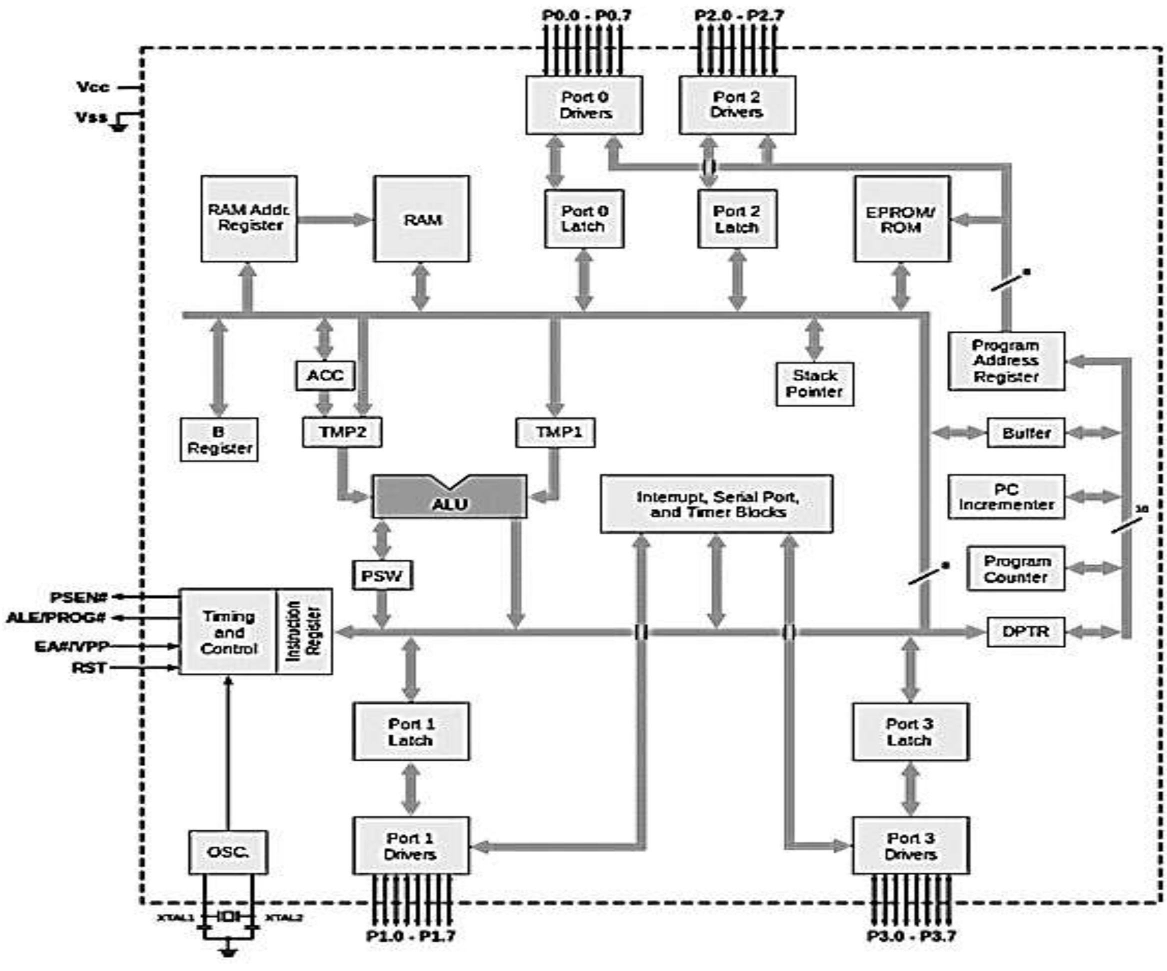


Figure 3.7- 8051 Architecture

3.3.1 सेंट्रल प्रोसेसिंग यूनिट (CPU): CPU 8-बिट ALU ऑपरेशन्स एक्सेकुट(Execute) करतो आणि प्रोग्राम मेमरीमधील इनस्टरकशन ची अंमलबजावणी करतो.

3.3.2 मेमरी ऑर्गनायझेशन (Memory Organization)

ROM (प्रोग्राम मेमरी): 64 KB ऍड्रेसेबल मेमरी जी नॉन-वोलाटाइल मेमरी म्हणून वापरली जाते आणि फर्मवेअर (Firmware) संचयित(Store) करते.

3.3.3 RAM (डेटा मेमरी): 128 बाइट्सची मेमरी जी जनरल-परपज रजिस्टर आणि स्टॅक म्हणून वापरली जाते. यात आठ रजिस्टर असलेले चार बँक, 16 बाइट्स बिट ऍड्रेसेबल क्षेत्र(Bit Addressable Area), आणि 80 बाइट्स जनरल-परपज मेमरी स्थानांचा समावेश आहे.

3.3.4 रजिस्टर (Registers): हे रजिस्टर मायक्रोकंट्रोलर मधील विशिष्ट ऑपरेशन्स नियंत्रित करतात.

a. अक्युम्युलेटर (A): अरीतम्याटिक आणि तार्किक (arithmetic and logical) ऑपरेशन्सच्या निकालांसाठी वापरले जाते. बहुतेक इनस्टरकशन साठी डिफॉल्ट गंतव्य रजिस्टर (destination register) म्हणून कार्य करते.

b. B रजिस्टर: फक्त गुणाकार आणि भागाकार सूचनांसाठी वापरले जाते. याचा सामान्य-उद्देशीय रजिस्टर म्हणूनही उपयोग होऊ शकतो.

c. प्रोग्राम स्टेटस वर्ड (PSW): या रजिस्टरचा वापर निकालाची स्थिती दर्शविण्यासाठी केला जातो आणि यात कॅरी (Carry), ऑक्सिलरी कॅरी (Auxillary Carry), पॅरिटी (Parity) आणि ओव्हरफ्लो(Overflow) यांसारखे फ्लॅग्स(Flags) असतात. तसेच, यात रजिस्टर बँक निवडीसाठी बिट्स असतात.

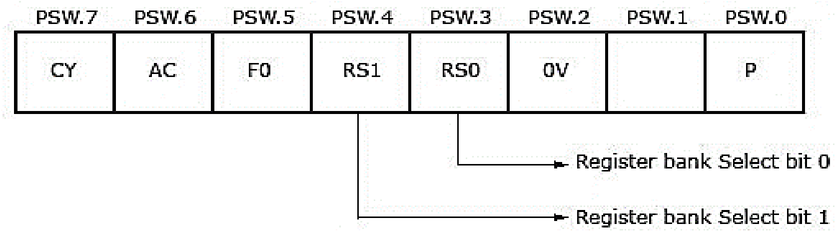


Fig 3.8 PSW register

Bit	Symbol	Name	Function
7	CY	कॅरि फ्लॅग (Carry flag)	अरितम्याटिक ऑपरेशन (arithmetic) मध्ये कॅरि किंवा बोर्रो जनरेट झाला तर हा फ्लॅग सेट होतो
6	AC	औक्सझीलरी कॅरि फ्लॅग (Auxiliary flag)	BCD ऑपरेशन मध्ये ज्यात D3 पासून D4 ला कॅरि ट्रान्सफर झाला तर हा फ्लॅग सेट होतो.
5	F0	जनरल पर्पस फ्लॅग (General-purpose Flag)	हा फ्लॅग यूजर डीफानड(User defined) आहे जो जनरल पर्पस साठी वापरला जातो.
4	RS1	रजिस्टर बँक सिलेक्ट बीट 1 (Register Bank Select Bit 1)	चार रजिस्टर बँकांपैकी एकाची निवड करण्यासाठी RS 0 बीट वापरतात.
3	RS0	रजिस्टर बँक सिलेक्ट बीट 0 (Register Bank Select Bit 0)	रजिस्टर बँक निवडण्यासाठी RS1 सह वापरला जातो.
2	OV	ऑवेरफलो फ्लॅग (Overflow Flag)	साईन्ड अरितम्याटिक ऑपरेशन(Signed Arithmatic Operation) मध्ये जर रिजल्ट ऑवेरफलो (रेंज च्या बाहेर) गेला तर फ्लॅग सेट होतो
1	-	रिजर्व्ड (Reserved)	वापरला जात नाही (बीट वॅल्यू(Bit value) 0 असते)
0	P	पॅरीटी फ्लॅग (Parity Flag)	रिजल्ट मध्ये जर बीट १ ची संख्या सम (even) असेल तर हा फ्लॅग सेट होतो.

Table 3.3 PSW register

d. स्टॅक पॉइंटर (SP): इंटरनल RAM मधील स्टॅकच्या शीर्षस्थानाकडे निर्देश करते. PUSH आणि POP ऑपरेशन्स दरम्यान स्वयंचलितपणे वाढवले आणि कमी केले जाते, तसेच उपकार्यक्रम (subroutine) संबंधित ऑपरेशन्ससाठी परतण्याचा पत्ता साठवते.

e. प्रोग्राम काउंटर (PC), डेटा पॉइंटर (DPTR): बाह्य मेमरीतील त्या स्थानाकडे निर्देश करण्यासाठी वापरले जाते जिथून डेटा वाचला किंवा लिहिला जाऊ शकतो. यात दोन रजिस्टर असतात – DPH (High Byte) आणि DPL (Low Byte). बाह्य मेमरी ॲक्सेस करताना अप्रत्यक्ष ॲड्रेसिंगसाठी वापरले जाते.

8051 मायक्रोकंट्रोलरमधील प्रोग्राम काउंटर (PC) एक 16-बिट रजिस्टर आहे जे पुढील अंमलात येणाऱ्या सूचनांचा पत्ता साठवते (Stores Address). हे सूचनांची फेचिंग(Fetching) पूर्ण झाल्यानंतर स्वयंचलितपणे वाढते आणि ब्रँचिंग दरम्यान अपडेट होते. रीसेट झाल्यावर PC 0000H स्थानी निर्देशित करते, ज्यामुळे योग्य प्रोग्राम अंमलबजावणी होते..

f. इनपुट/आउटपुट रजिस्टर (Input/Output Registers):

पोर्ट रजिस्टर (P0–P3) – हे चार 8-बिट रजिस्टर (P0, P1, P2, P3) इनपुट/आउटपुट पोर्ट नियंत्रित करतात. प्रत्येक बिट विशिष्ट इनपुट/आउटपुट पिनशी संबंधित असतो.

3.3.5 टाइमर रजिस्टर (Timer Registers):

टाइमर 0 आणि टाइमर 1 रजिस्टर (TH0, TL0, TH1, TL1) – हे गणना (counting) आणि टाइमिंग ऑपरेशन्ससाठी वापरले जातात. हे Mode 0 ते Mode 3 पर्यंत विविध मोड्समध्ये कार्य करतात.

a. टाइमर नियंत्रण रजिस्टर (TCON - Timer Control Register):

टाइमर आणि बाह्य व्यत्यय (external interrupts) नियंत्रित करण्यासाठी वापरले जाते.

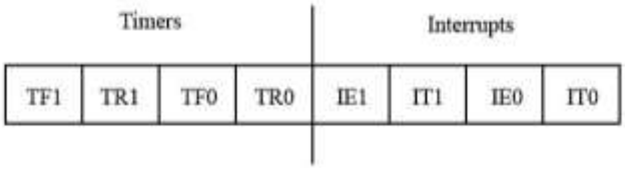


Fig 3.9 TCON register

Bit	Name	Function
7	TF1	टाइमर 1 ओव्हरफ्लो फ्लॅग: टाइमर 1 ओव्हरफ्लो झाल्यावर सेट केला जातो.
6	TR1	1 = स्टार्ट टाइमर 1, 0 = स्टॉप टाइमर 1.
5	TF0	टाइमर 0 ओव्हरफ्लो फ्लॅग: टाइमर 0 ओव्हरफ्लो झाल्यावर सेट.
4	TR0	1 = स्टार्ट टाइमर 0, 0 = स्टॉप टाइमर 0.
3	IE1	एक्सटर्नल इनटरपट (External interrupt) 1 फ्लॅग.
2	IT1	1 = एज-ट्रिगर (edge trigger) 0 = लेव्हल-ट्रिगर (Level trigger).
1	IE0	एक्सटर्नल इनटरपट (External interrupt) 0 फ्लॅग.
0	IT0	1 = एज-ट्रिगर (edge trigger) 0 = लेव्हल-ट्रिगर (Level trigger).

Table 3.4 TCON Register

b. टाइमर मोड रजिस्टर (TMOD): टाइमर 0 आणि टाइमर 1 च्या विविध ऑपरेटिंग मोड्सची संरचना (configuration) करते.

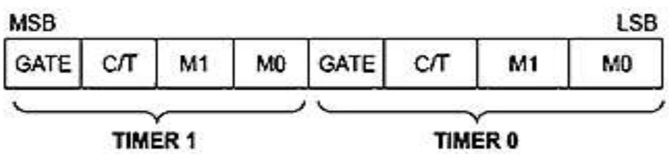


Fig 3.10 TMOD register

Bit	Name	Function
7	GATE1	1 = टाइमर 1 तेव्हाच सुरू होतो जेव्हा INT1 (P3.3) हाय असतो.
6	C/T1	0 = टाइमर मोड, 1 = काउंटर मोड.
5,4	M1, M0 (Timer 1 Mode)	00 = मोड 0 (13-बिट), 01 = मोड 1 (16-बिट), 10 = मोड 2 (8-बिट ऑटो-रिलोड), 11 = मोड 3 (स्प्लिट मोड)।
3	GATE0	1 = टाइमर 0 तेव्हाच सुरू होतो जेव्हा INT0 (पी 3.2) हाय असते .
2	C/T0	0 = टाइमर मोड, 1 = काउंटर मोड.
1-0	M1, M0 (Timer 0 Mode)	टाइमर 1 प्रमाणेच.

Table 3.5 TMOD register

3.3.6 सिरीयल कम्युनिकेशन रजिस्टर (Serial Communication Registers)

➢सिरीयल बफर रजिस्टर (SBUF): ट्रान्समिटेड (transmitted) आणि रीसीव्ड (received) केलेला सिरीयल डेटा साठवते.

➢सिरीयल कंट्रोल रजिस्टर (SCON): सिरीयल कम्युनिकेशन सेटिंग्ज संरचित(Configure) करते, ज्यामध्ये मोड निवड (mode selection) आणि बॉड रेट नियंत्रण (baud rate control) समाविष्ट आहे.

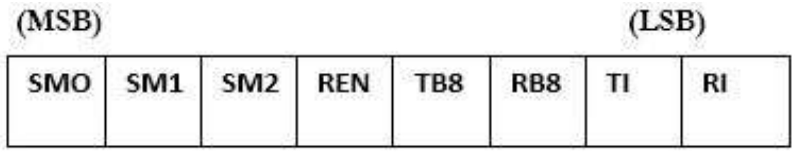


Fig 3.11 SCON register

Bit	Name	Function
7	SM0	सीरियल मोड सिलेक्शन.
6	SM1	1 = मोड 1 8-बिट (UART), 0 = मोड 0 शिफ्ट रजिस्टर (shift register)
5	SM2	मल्टीप्रोसेसर कम्युनिकेशन एनेबल (enable)
3	TB8	ट्रान्समिटेड 9^{th} डेटा बिट मोड 2/3 मध्ये
2	RB8	रीसीव्ड 9^{th} डेटा बिट मोड 2/3 मध्ये
1	TI	ट्रान्समिट इनटरपट फ्लॅग: डेटा सेंड झाल्या वर सेट होतो.
0	RI	रीसीव इनटरपट फ्लॅग: डेटा रीसीव झाल्या वर सेट होतो.

Table 3.6 SCON register

3.3.7 इंटरप्ट रजिस्टर (Interrupt Registers)

➢ इंटरप्ट एनाबल रजिस्टर (IE): विशिष्ट इंटरप्ट्स एनेबल (enable) किंवा डिसेबल (disable) करण्यासाठी वापरले जाते

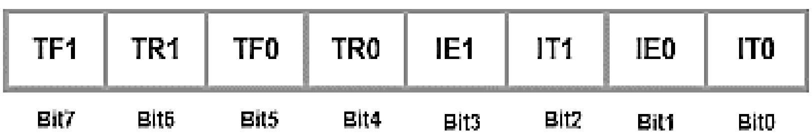


Fig 3.12 Interrupt register

Bit	Name	Function
7	EA	1 = सर्व इनटरपट एनेबल करा, 0 = सर्व इनटरपट डिसेबल करा
6	-	आरक्षित (Reserved).

5	ET2	टाइमर 2 इनटरपट एनेबल (8052 only)
4	ES	सिरियल इनटरपट एनेबल .
3	ET1	टाइमर 1 इनटरपट एनेबल .
2	EX1	एक्सटर्नल इनटरपट 1 एनेबल .
1	ET0	टाइमर 0 इनटरपट एनेबल .
0	EX0	एक्सटर्नल इनटरपट 0 एनेबल .

Table 3.7 Interrupt register

➢ इंटरप्ट प्रायोरिटी रजिस्टर (IP): इंटरप्ट्ससाठी प्राधान्य स्तर (priority levels) सेट करण्यासाठी वापरले जाते.

3.13 Interrupt priority register

Bit	Name	Function
7	-	आरक्षित (Reserved)
6	-	आरक्षित (Reserved)
5	PT1	टाइमर 1 इंटरप्ट प्राधान्य (Priority) बीट
4	PX1	एक्सटर्नल इंटरप्ट 1 प्राधान्य (Priority) बीट
3	PT0	टाइमर 0 इंटरप्ट प्राधान्य (Priority) बीट
2	PX0	एक्सटर्नल इंटरप्ट 0 प्राधान्य (Priority) बीट
1	PS	सिरियल इंटरप्ट प्राधान्य (Priority) बीट
0	-	आरक्षित (Reserved)

Table 3.8 Interrupt priority register

3.3.8 8051 ची पिन संरचना (Pin Configuration of 8051):

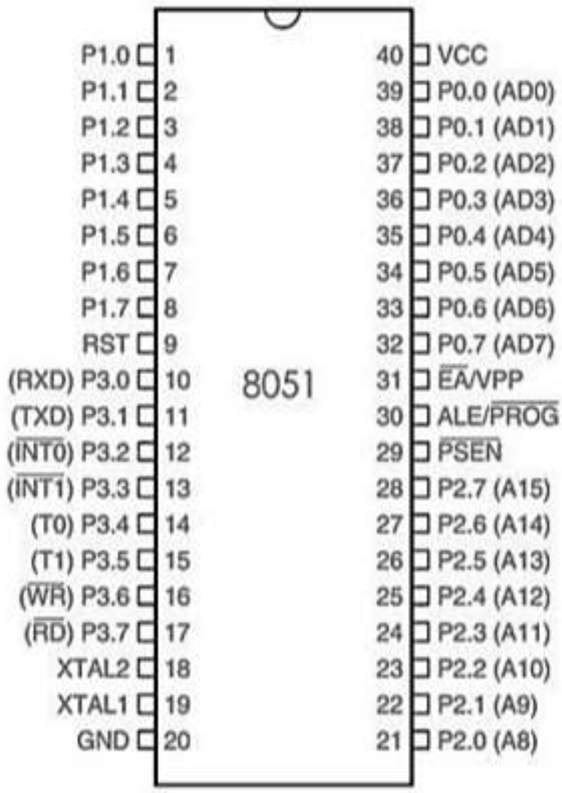


Figure 3.14- Pin out diagram for 8051.

8051 च्या पिनचे वर्णन (Pin Description of 8051)

Pin Numbers	Pin Name	Functions
1 to 8	पोर्ट 1 पिन (P 1.0 ते 1.7)	पोर्ट 1 पिन जे अंतर्गत पुल-अप रेझिस्टरसह द्वि-दिशात्मक (bidirectional) I/O पोर्ट म्हणून कॉन्फिगर केले जाऊ शकते.
9	रीसेट (Reset)	प्रोसेसर रीसेट करण्यासाठी वापरला जातो
10 to 17	पोर्ट 3 पिन (P3.0 ते P 3.7)	पोर्ट 3 पिन I/O पिन म्हणून वागू शकतात आणि इंटरप्ट (interrupts), टाइमर ऑपरेशन्स आणि सीरियल I/O ट्रान्सफरसाठी सिग्नल देखील प्रदान करू शकतात **Port Pin — Alternate Function** P3.0 — RXD (serial input port) P3.1 — TXD (serial output port) P3.2 — $\overline{\text{INT0}}$ (external interrupt 0) P3.3 — $\overline{\text{INT1}}$ (external interrupt 1) P3.4 — T0 (Timer 0 external input) P3.5 — T1 (Timer 1 external input) P3.6 — $\overline{\text{WR}}$ (external data memory write strobe) P3.7 — $\overline{\text{RD}}$ (external data memory read strobe)
18,19	XTAL1 and XTAL2	मायक्रोकंट्रोलरला ऑपरेटिंग फ्रिक्वेन्सी प्रदान करण्यासाठी क्रिस्टल ऑसिलेटर (Crystal Oscillator)जोडण्यासाठी या पिन्सचा वापर केला जातो.
20	GND	ग्राऊंड पिन सिस्टमला 0 V प्रदान करण्यासाठी वापरला जात असे.
21 to 28	पोर्ट 2 पिन (P2.0 to P2.7)	पोर्ट 2 पिन चा वापर द्विदिशा (bidirectional) I/O पिन म्हणून आणि A15 ते A8 पर्यंत उच्च ऑर्डर ॲड्रेस लाइन्स (High order address lines) म्हणून देखील केला जाऊ शकतो.
29	$\overline{\text{PSEN}}$	प्रोग्राम स्ट्रोब इनेबल एक्सटर्नल मेमरीमधून डेटा आणण्यासाठी वापरला जातो.
30	ALE/$\overline{\text{PROG}}$	ॲड्रेस लॅच इनेबल (ALE) चा वापर लोवर ऑर्डर ॲड्रेस / डेटा लाइन्स डीमल्टिप्लेक्स करण्यासाठी केला जातो किंवा EPROM च्या प्रोग्रामिंगसाठी ,प्रोग्रामिंग सिग्नल देण्यासाठी वापरला जातो.
31	$\overline{\text{EA}}$/Vpp	External Access ($\overline{\text{EA}}$) एक्सटर्नल डिव्हाइस ॲक्सेस एनेबल अँड डिसेबल करण्यासाठी वापरला जातो. EPROM च्या प्रोग्रामिंगसाठी 25 V supply (किंवा त्यापेक्षा जास्त) देखील प्रदान करू शकतो.
32 to 39	Port 0 pins (P0.0 to P0.7)	पोर्ट 0 पिन एकतर इनपुट किंवा आउटपुट पिन आणि लोअर ऑर्डर मल्टिप्लेक्स ॲड्रेस / डेटा लाइन्स म्हणून वापर केले जातात.
40	Vcc	पॉवर सप्लाय पिन +५ volt ला जोडली जाते

Table 3.9- Pin Description of 8051 Microcontroller

3.4 स्टॅक आणि स्टॅक पॉइंटर, मेमरी ऑर्गनायझेशन (RAM, ROM) (Stack and stack pointer, Memory organization (RAM , ROM). मेमरी ऑर्गनायझेशन (Memory Organization):

8051 मायक्रोकंट्रोलरमध्ये 128-बाइट्स इंटरनल RAM आणि 4KB ऑन-चिप ROM असते. RAM ला डेटा मेमरी म्हणतात, तर ROM ला प्रोग्राम मेमरी किंवा कोड मेमरी म्हणतात.

एका संगणकाला प्रोग्राम कोड बाइट्ससाठी मेमरी (ROM) आणि कार्यरत असताना डेटा बदलण्यासाठी RAM मेमरी आवश्यक असते. 8051 मध्ये या कार्यांसाठी अंतर्गत RAM आणि ROM असते.

3.4.1 अंतर्गत रॅम (Internal RAM):

128-बाइट्सची अंतर्गत RAM तीन विशिष्ट भागांमध्ये विभागलेली आहे:

1. वर्किंग रजिस्टर बँक (Working Register Banks)– 00H ते 1FH:

a. 32 बाइट्स हे चार बँक्समध्ये विभागलेले असतात (Bank 0 ते Bank 3).

b. प्रत्येक बँकेत 8 रजिस्टर (R0 ते R7) असतात.
c. एखादा रजिस्टर त्याच्या नावाने किंवा RAM पत्त्याने(Address) ऍक्सेस केला जाऊ शकतो.
d. PSW (Program Status Word) मधील RS0 आणि RS1 बिट्स बँक निवडीसाठी वापरले जातात.
e. न निवडलेले रजिस्टर बँक्स सामान्य उद्देशीय-RAM म्हणून वापरले जाऊ शकतात.
f. रीसेट झाल्यावर Bank 0 निवडलेली असते.

2. बिट-ऍड्रेस करण्यायोग्य क्षेत्र (Bit Addressable Area) – 20H ते 2FH

a. 16-बाइट्सचा 2-बिट ऍड्रेस करण्यायोग्य भाग असतो.
b. एकूण 128 ऍड्रेस करण्यायोग्य बिट्स तयार होतात.
c. बिट 00H ते 7FH पर्यंतच्या पत्त्याने(Address) निर्दिष्ट केला जाऊ शकतो.
d. 8 बिट्स एकत्रित करून 20H ते 2FH पर्यंत कोणतेही बाइट ऍड्रेस तयार करता येते.
e. उदाहरणार्थ, बिट अ‍ॅड्रेस 4FH म्हणजे बाइट अ‍ॅड्रेस 29H चा बिट 7.
f. हे बिट्स फ्लॅग संबंधित ऑपरेशन्ससाठी वापरले जातात.

3. सामान्य-उद्देशीय रॅम क्षेत्र (General Purpose RAM) – 30H ते 7FH:

a. 80-बाइट्सचे सामान्य-उद्देशीय RAM क्षेत्र आहे.
b. बिट-ऍड्रेस करण्यायोग्य भागाच्या वरच्या भागात हे बाइट्स स्थित असतात.
c. मायक्रोकंट्रोलर स्टोरेज क्षेत्र म्हणून आणि ऑपरेटिंग स्टॅकसाठी देखील या भागाचा उपयोग केला जातो.

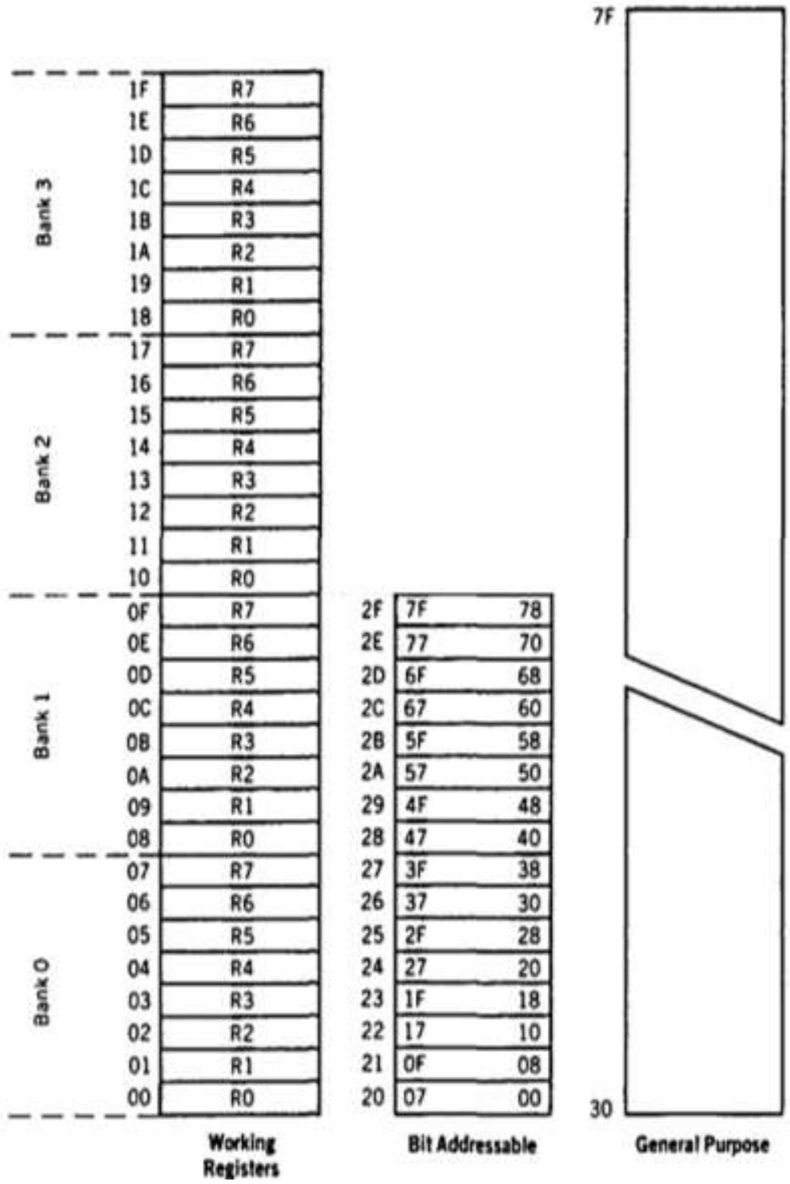


Figure 3.15- Internal RAM organization of 8051

3.4.2 अंतर्गत रॉम (Internal ROM):

8051 डेटा मेमरी आणि प्रोग्राम कोड मेमरी दोन वेगवेगळ्या भौतिक घटकांमध्ये (Physical Entities) असू शकतात. प्रत्येकाची ॲड्रेस रेंज समान आहे.

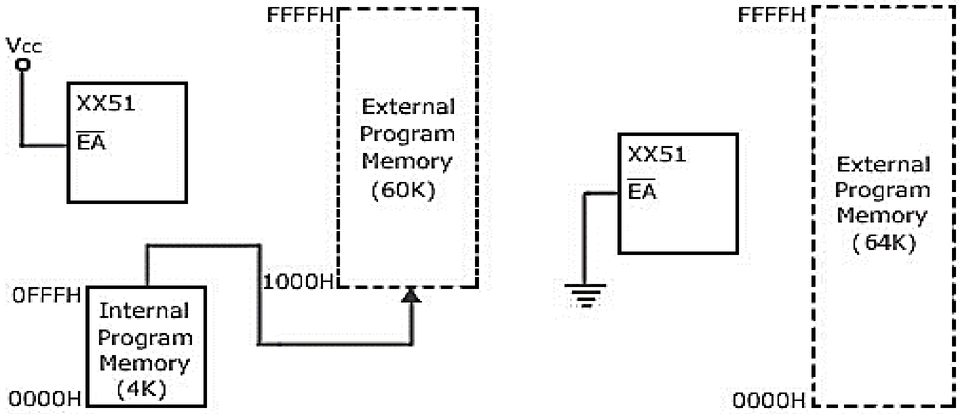


Figure 3.16- ROM organization in 8051.

a. अंतर्गत प्रोग्राम कोडचा संबंधित ब्लॉक कोड ॲड्रेस स्पेस 0000H ते 0FFF H आहे .

b. प्रोग्राम काऊंटर (PC) 0000H ते FFFFH ॲड्रेस पर्यंत प्रोग्राम कोड बाइट्सला संबोधित करण्यासाठी वापरला जातो.

c. 0FFF H पेक्षा जास्त प्रोग्राम ॲड्रेस, जे अंतर्गत रोम ॲड्रेस च्या बाहेर आहेत, 8051 ला बाह्य (external) प्रोग्राम मेमरीमधून कोड बाइट्स आपोआप मिळतात.

d. कोड बाइट्स बाह्य मेमरी 0000 H ते FFFF H मधून देखील आणले जाऊ शकतात, त्यासाठी एक्सटर्नल ॲक्सेस ($\overline{EA}$) पिन नंबर 31 ग्राउंड ला कनेक्ट करायचे.

3.4.3 स्टॅक आणि स्टॅक पॉइंटर (Stack and the Stack Pointer):

स्टॅक ही इंटरनल RAM मधील एक जागा आहे जी काही ऑपकोड्ससह वेगाने डेटा साठवण्यासाठी आणि पुनर्प्राप्त करण्यासाठी वापरली जाते. 8051 मायक्रोकंट्रोलरमध्ये 8-bit स्टॅक पॉइंटर (SP) रजिस्टर इंटरनल RAM च्या त्या ॲड्रेसचा ट्रॅक ठेवतो, जो टॉप ऑफ स्टॅक (TOS) म्हणून ओळखला जातो. SP रजिस्टरमध्ये असलेला ॲड्रेस हा स्टॅक ऑपरेशनद्वारे साठवलेला शेवटचा बाइट असतो.

1. स्टॅक हा "लास्ट इन फर्स्ट आऊट" (LIFO) प्रकारचा मेमरी स्ट्रक्चर आहे जो 8051 मध्ये तात्पुरत्या डेटासाठी वापरला जातो.
2. स्टॅक पॉइंटर (SP) हा 8-bit रजिस्टर आहे जो स्टॅकच्या अप्पर स्थानी (TOS) ट्रॅक ठेवतो.
3. SP चे डिफॉल्ट मूल्य(Default Value) 07H असते, याचा अर्थ पहिले स्टॅक एन्ट्री 08H ॲड्रेस वर स्टोअर केली जाते.

i.

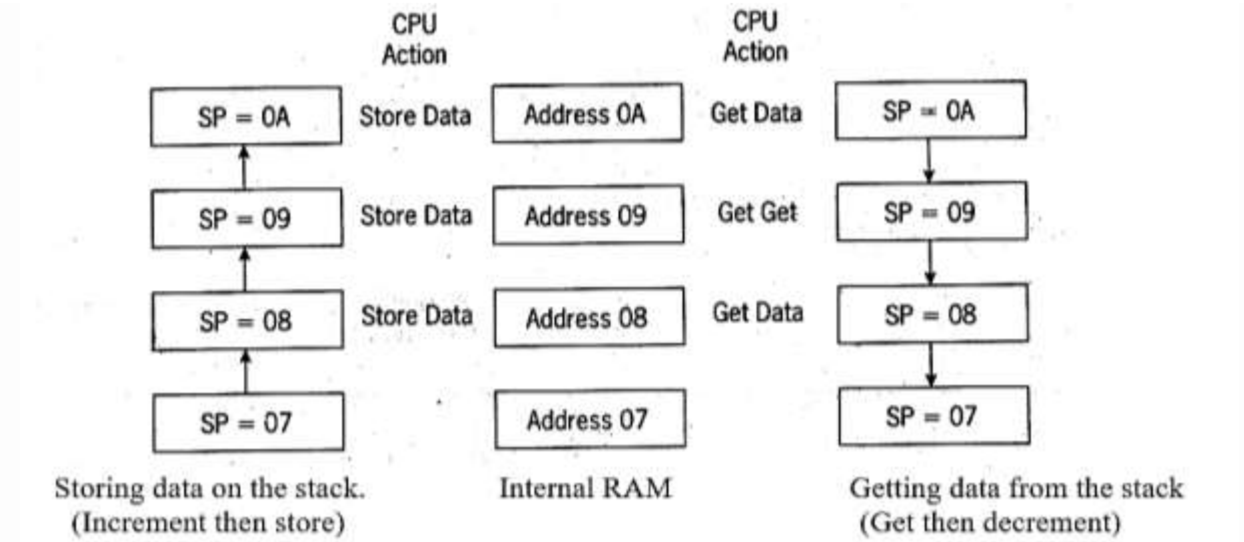


Figure 3.17- Stack operation in 8051.

स्वाध्याय:

1.अॅड्रेस बस आणि डेटा बस ह्यांचे कार्य सांगा.
2.मायक्रोप्रोसेसर आणि मायक्रोकंट्रोलरची तुलना करा. (कोणतेही चार परामेटर्स)
3.मायक्रोकंट्रोलर 8051ची कोणतीही आठ वैशिष्ट्ये लिहा
4.8051 च्या खालील पिनच्या कार्याचे वर्णन करा:
(१) $\overline{PSEN}$. (२) RESET (३) ALE (४) $\overline{EA}$.
5.पोर्ट -3 पिनचे ऑल्टरनेटिव कार्य लिहा.
6.8051 मायक्रो-कंट्रोलरचे आर्किटेक्चर ड्रॉ करा.
7.8051 च्या इंटर्नल आणि एक्सटर्नल मेमरी ऑर्गनायझेशनचे स्पष्टीकरण द्या.
8.8051 चे प्रोग्राम स्टेटस वर्ड (PSW) रजिस्टर ड्रॉ करा.
9.हार्वर्ड आणि व्हॉन-न्यूमन आर्किटेक्चरची तुलना करा (कोणतेही 4 परामेटर्स)
10. TMOD रजिस्टर काढा आणि प्रत्येक बिटचे 8051 चे कार्य समजावून सांगा.
11. $\overline{EA}$= 0 आणि , $\overline{EA}$= 1 साठी मेमरी ऑर्गनायझेशन ड्रॉ करा.

संदर्भ पाठयपुस्तक (Reference Books)

Sr No	Author	Title	Publisher with ISBN Number
1	Mazidi Muhammad Ali, Mazidi Janice Gillispe, Mckinlay Rolin D	The 8051 Microcontroller and Embedded Systems: Using Assembly and C	Pearson Publication, 2017 ISBN: 9788131710265
2	Ayala Kenneth J	The 8051 Microcontroller	Thomson Delmar Learning, 2004 ISBN: 9781401861582
3	Deshmukh Ajay V	Microcontroller: Theory and Application	McGraw Hill,2011 ISBN: 9780070585959
4	Pal Ajit	Microcontrollers: Principle and Application	PHI Learning, 2014 ISBN: 978812034394
5	Chattopadhyay Santanu	Microcontroller and Applications	All India Council for Technical Education, 2023 ISBN: 9788196057602

माहिती संकेतस्थळ (Information Websites):

1. https://en.wikipedia.org/wiki/Intel_MCS-51
2. https://ekumbh.aicte-india.org/allbook.php#
3. https://play.google.com/store/apps/details?id=com.coderbro.tutorial.a8051microcontroller&hl=en_IE&pli=1
4. https://nptel.ac.in/courses/117104072
5. **Diagram Courtesy** Fig 3.6, Fig 3.7, Fig 3.14 from the book ‘The 8051 Microcontroller (Author - Ayala Kenneth J)’

युनिट – IV
8051 इंस्ट्रक्शन सेट आणि प्रोग्रामिंग
(8051 Instruction Set and Programming)

विषय निष्पत्ती (Course Outcomes CO's)

मायक्रोकंट्रोलरसाठी असेंब्ली लँग्वेज प्रोग्रॅम विकसित करा आणि कार्यान्वित करा.

(Develop and execute programs in assembly language for microcontroller.)

विषय निष्पत्ती बरोबर जुळणारी थिअरी लर्निंग निष्पत्ती (Theory Learning Outcomes (TLO's) aligned with Course Outcomes (CO's))

4.1 उदाहरणांसह दिलेल्या इंस्ट्रक्शनचे ॲड्रेसिंग मोड ओळखा.

(Identify addressing mode of the given instruction with examples).

4.2 दिलेल्या इंस्ट्रक्शनचे कार्य योग्य उदाहरणांसह वर्णन करा

(Describe function of the given instruction with suitable examples.)

4.3 दिलेल्या असेंबलर निर्देशांच्या वापराचे उदाहरणांसह समर्थन करा.

(Justify the use of the given assembler directives with examples.)

4.1 ॲड्रेसिंग मोड्स: इमेडिएट ॲड्रेसिंग, रजिस्टर ॲड्रेसिंग, डायरेक्ट ॲड्रेसिंग, इनडायरेक्ट ॲड्रेसिंग, रिलेटिव्ह ॲड्रेसिंग, अब्सोलूट ॲड्रेसिंग, बिट इन्हरेंट ॲड्रेसिंग, बिट डायरेक्ट ॲड्रेसिंग (Addressing Modes: Immediate, register, direct, indirect, indexed, relative, absolute, bit inherent, bit direct).

4.1.1 ॲड्रेसिंग मोड्स (Addressing Modes)

डेटामध्ये प्रवेश करण्याच्या विविध पद्धतींना ॲड्रेसिंग मोड म्हणतात. 8051 ॲड्रेसिंग मोड खालीलप्रमाणे वर्गीकृत केले आहेत

1. इमेडिएट ॲड्रेसिंग (Immediate addressing)
2. रजिस्टर ॲड्रेसिंग (Register addressing)
3. डायरेक्ट ॲड्रेसिंग (Direct addressing)
4. इनडायरेक्टॲड्रेसिंग (Indirect addressing)
5. इनडेक्स ॲड्रेसिंग (Indexed addressing)
6. रिलेटिव्ह ॲड्रेसिंग (Relative addressing)
7. अब्सोलूट ॲड्रेसिंग (Absolute addressing)
8. बिट इन्हरेंट ॲड्रेसिंग (Bit inherent addressing)
9. बिट डायरेक्ट ॲड्रेसिंग (Bit direct addressing)

1. इमेडिएट ॲड्रेसिंग (Immediate addressing)

या ॲड्रेसिंग मोडमध्ये सूचनांचाच एक भाग म्हणून डेटा प्रदान केला जातो. दुसऱ्या शब्दांत, डेटा त्वरित इंस्ट्रक्शनचे (Instruction) अनुसरण करतो.

उदाहरण:

MOV A, #30H

2. रजिस्टर ॲड्रेसिंग (Register addressing)

या ॲड्रेसिंग मोडमध्ये रजिस्टर डेटा धारण करतो. आठ सामान्य रजिस्टरपैकी (R0 ते R7) कोणतेही एक रजिस्टर वापरले जाऊ शकते आणि ते ऑपरेन्ड(Operand) म्हणून निर्दिष्ट केले जाते.

उदाहरण:

MOV A,R0

ADD A,R6

R0 – R7 रजिस्टर बँकेच्या करंट सिलेक्शनमधून(Current Selection) निवडले जाईल. डीफॉल्ट रजिस्टर बँक (Default register bank) म्हणजे बँक 0 असेल.

3. डायरेक्ट ऍड्रेसिंग (Direct addressing)

अंतर्गत मेमरीमध्ये प्रवेश करण्यासाठी दोन पद्धती आहेत: डायरेक्ट ॲड्रेसिंग आणि इनडायरेक्ट ॲड्रेसिंग.डायरेक्ट ॲड्रेसिंग मोड वापरून आपण फक्त अंतर्गत मेमरीच नव्हे तर SFRs (स्पेशल फंक्शन रजिस्टर्स) देखील ॲड्रेस करू शकतो. डायरेक्ट ऍड्रेसिंगमध्ये, 8 बिट इंटर्नल डेटा मेमरी ऍड्रेस इन्स्ट्रक्शन भाग म्हणून निर्दिष्ट केला जातो आणि म्हणूनच, तो फक्त 00H ते FFH च्या श्रेणीमध्ये ॲड्रेस निर्दिष्ट करू शकतो. या ऍड्रेसिंग मोडमध्ये, डेटा थेट मेमरीमधून प्राप्त केला जातो

उदाहरण:

MOV A,60h

ADD A,30h

4. इनडायरेक्ट ऍड्रेसिंग (Indirect addressing)

इनडायरेक्ट ॲड्रेसिंग मोड डेटा हालचालीमध्ये वापरला जाणारा वास्तविक ॲड्रेस(actual address) ठेवण्यासाठी रजिस्टर वापरतो. R0,R1 आणि DPTR हे एकमेव आहेत जे डेटा पॉइंटर म्हणून वापरली जाऊ शकतात. SFR रजिस्टर्सचा संदर्भ देण्यासाठी अप्रत्यक्ष ॲड्रेस (Indirect address) वापरला जाऊ शकत नाही. R0 आणि R1 दोन्ही 8 बिट ऍड्रेस धारण करू शकतात आणि DPTR 16 बिट ऍड्रेस धारण करू शकतो.

उदाहरण:

MOV A,@R0

ADD A,@R1

MOVX A,@DPTR

6. इनडेक्स ऍड्रेसिंग (Indexed addressing)

इंडेक्सड ॲड्रेसिंग मध्ये प्रोग्राम काउंटर (PC) किंवा डेटा पॉइंटर (DPTR) बेस ॲड्रेस साठवण्यासाठी वापरला जातो, आणि अक्युम्युलेटर (A) ऑफसेट ॲड्रेस साठवण्यासाठी वापरला जातो. बेस ॲड्रेस आणि ऑफसेट ॲड्रेस एकत्र करून इफेक्टीव ॲड्रेस तयार केला जातो. JMP आणि MOVC इंस्ट्रक्शन्ससह इंडेक्सड ॲड्रेसिंग वापरले जाते. लुकअप टेबल्स (Look up) सहजपणे इंडेक्सड ॲड्रेसिंगच्या मदतीने अंमलात आणता येतात

उदाहरण:

MOVC A, @A+DPTR // ॲक्युम्युलेटर A आणि DPTR च्या बेरीज द्वारा दर्शवलेल्या मेमरी स्थानामधील सम (sum) ॲक्युम्युलेटर A मध्ये कॉपी केली जाते.

MOVC A, @A+PC // ॲक्युम्युलेटर A आणि प्रोग्रॅम काऊंटरच्या बेरीज द्वारा दर्शवलेल्या मेमरी स्थानामधील सम (sum) ॲक्युम्युलेटर A मध्ये कॉपी केली जाते.

6. रिलेटिव्ह ऍड्रेसिंग (Relative addressing)

रिलेटिव्ह ॲड्रेसिंग फक्त कंडिशनल जंप इंस्ट्रक्शनसाठी वापरली जाते. रिलेटिव्ह ॲड्रेस (ऑफसेट) हा 8-बिट साईन नंबर असतो, जो स्वतःहून PC (प्रोग्राम काउंटर) मध्ये जोडला जातो, त्यामुळे पुढील इन्स्ट्रक्शन ॲड्रेस मिळतो. 8-बिट साइन केलेल्या ऑफसेटमुळे जंप रेंज +127 ते -128 लोकेशन्सपर्यंतअसते. जंप डेस्टिनेशन लेबलने निर्दिष्ट केले जाते, आणि असेंबलर त्यानुसार जंप ऑफसेट कॅल्क्युलेट करतो. या ॲड्रेसिंग मोडचा मुख्य फायदा म्हणजे प्रोग्राम कोड सहज री-लोकेट करता येतो.

उदाहरण:

SJMP LOOP1

JC BACK

7.अब्सोलूट ऍड्रेसिंग (Absolute addressing)

अब्सोलूट ॲड्रेसिंग फक्त AJMP (अब्सोलूट जंप) आणि ACALL (अब्सोलूट कॉल) इंस्ट्रक्शनसाठी वापरली जाते. # चिन्ह सूचित करते की डेटा त्वरित सूचना आहे. या 2 बाइट सूचना आहेत .अब्सोलूट ऍड्रेसिंग मोड इन्स्ट्रक्शन भाग म्हणून मेमरी अड्रेसचा सर्वात कमी 11 बिट निर्दिष्ट करतो .डेस्टिनेशन ॲड्रेसचे अप्पर 5 बिट हे प्रोग्राम काउंटरचे अप्पर 5 बिट आहेत. म्हणूनच, अब्सोलूट ऍड्रेसिंग प्रोग्राम मेमरीच्या 2 केबाइट रेंजमधील ब्रँचिंग मान्य करते.

उदाहरण:

AJMP LOOP1

ACALL LOOP2

8. बिट इन्हरेंट ऍड्रेसिंग (Bit inherent addressing)

या ॲड्रेसिंगमध्ये, ऑपरेन्ड समाविष्ट असलेल्या फ्लॅगचा ॲड्रेस, इन्स्ट्रक्शन ऑपकोडमध्ये सूचित केलेला असतो.

उदाहरण:

CLR C ; कॅरी फ्लॅग 0 होतो

9. बिट डायरेक्ट ऍड्रेसिंग (Bit direct addressing)

या ॲड्रेसिंग मोडमध्ये, बिटचा डायरेक्ट ॲड्रेस इन्स्ट्रक्शन मध्ये निर्दिष्ट केला जातो. रॅम स्पेस 20H ते 2FH आणि बहुतांशी विशेष फंक्शन रजिस्टर(SFR) बिट अड्रेसबल आहेत . बिट ॲड्रेस व्हॅल्यू 00H ते 7FH दरम्यान आहेत.

उदाहरण:

CLR 07h ; 20H रॅम स्पेसपैकी बिट 7 क्लिअर करते

SETB 07H ; 20H रॅम जागेतील बिट 7 सेट करते.

4.2 इन्स्ट्रक्शन सेट(योग्य उदारणासह): (डाटा ट्रान्स्फर, लॉजिकल, अरीथमेटिक, ब्रँचिंग, मशीन कंट्रोल, स्टॉक ऑपरेशन, बुलियन.(Instruction Set (with appropriate example) : Data transfer, Logical, Arithmetic, Branching, Machine control, Stack operation, Boolean.)

4.2.1. डाटा ट्रान्स्फर इन्स्ट्रक्शन (Data Transfer Instructions)

ह्या इन्स्ट्रक्शन, डेटा एका ठिकाणाहून दुसऱ्या ठिकाणी ट्रान्सफर करण्यासाठी वापरले जातात, आणि डेटामध्ये कोणतेही बदल होत नाहीत.

निर्देश	ऑपरेशन	उदाहरण	स्पष्टीकरण
MOV A, R1	डेटा हलविणे	MOV A, R2	R2 मधील डेटा बदलत नाही; तो फक्त A मध्ये कॉपी होतो.
MOV R0, #25H	लोड इमिजिएट व्हॅल्यू	MOV R0, #25H	R0 मध्ये 25H लोड करा.
MOV DPTR, #3000H	लोड डेटा पॉइंटर (DPTR)	MOV DPTR, #3000H	डीपीटीआर रजिस्टरमध्ये ॲड्रेस 3000H लोड करतो.
MOVX A, @DPTR	बाह्य मेमरीमधून (External Memory) डेटा आणण्यासाठी	MOVX A, @DPTR	हा इन्स्ट्रक्शन DPTR मध्ये असलेल्या ॲड्रेसवरून बाह्य मेमरीतील डेटा अक्युम्युलेटर (A) मध्ये हलवतो.
PUSH 07H	स्टॅकवर डेटा ढकलण्यासाठी केला जातो.	PUSH 07H	हे इन्स्ट्रक्शन रजिस्टर 07H मधील डेटा स्टॅकवर पुश करते.
POP 05H	स्टॅकवरील डेटा रजिस्टरमध्ये आणला जातो.	POP 05H	हे इन्स्ट्रक्शन स्टॅकवरील टॉप डेटा रजिस्टर 05H मध्ये पॉप करतो.

Table No.4.1 Example

4.2.2. लॉजिकल इन्स्ट्रक्शन (Logical Instructions)

ह्या इन्स्ट्रक्शन बिटवाइझ लॉजिकल ऑपरेशन्स (AND, OR, XOR) आणि कम्पेअर ऑपरेशन्स (Compare) करण्यासाठी वापरले जातात.

निर्देश	ऑपरेशन	उदाहरण	स्पष्टीकरण
ANL A, R3	AND ऑपरेशन	ANL A, R3	A आणि R3 यांच्यात बिटवाइज AND ऑपरेशन करून परिणाम A मध्ये संग्रहित(store) करते

ORL A, #0F0H	OR ऑपरेशन	ORL A, #0F0H	A आणि 0F0H यांच्यात बिटवाइज OR ऑपरेशन करून, परिणाम A मध्ये संग्रहित (store) करते
XRL A, B	XOR ऑपरेशन	XRL A, B	A आणि B यांच्यात XOR ऑपरेशन करून, परिणाम A मध्ये संग्रहित (store) करते
CPL A	Complement ऑपरेशन	CPL A	अक्युम्युलेटरमधील सर्व बिट्स उलट करते (Invert करते)
CLR C	CY फ्लॅग क्लिअर करणे	CLR C	कॅरी फ्लॅग क्लिअर करते (C = 0)

Table No.4.2 Example

4.2.3. अरीथमेटिक इन्स्ट्रक्शन (Arithmetic Instructions)

ह्या इन्स्ट्रक्शन बेरीज, वजाबाकी, गुणाकार आणि भागाकार यासारख्या गणितीय क्रिया करतात.

निर्देश	ऑपरेशन	उदाहरण	स्पष्टीकरण
ADD A, R2	बेरीज (Addition)	ADD A, R2	रजिस्टर R2 चे डेटा आणि अक्कमुलेटर चे डेटा ची बेरीज करणे
SUBB A, #10H	बोरो घेऊन वजाबाकी	SUBB A, #10H	बोरो घेऊन A मधून 10H वजा करा
MUL AB	गुणाकार	MUL AB	A आणि B ची गुणाकार करणे , रिझल्ट A (कमी बाइट) आणि B (उच्च बाइट) मध्ये स्टोअर करतो
DIV AB	भागाकार	DIV AB	A चा B वर भागाकार करणे. कोशंट A मध्ये स्टोअर करतो आणि उर्वरित B मध्ये स्टोअर करतो
INC R5	वाढ	INC R5	रजिस्टर R5 मध्ये 1 ने वाढ करणे
DEC A	कमी करणे	DEC A	अक्युम्युलेटर A मध्ये 1 ने कमी करणे

Table No.4.3 Example

4.2.4. ब्रँचिंग इन्स्ट्रक्शन (Branching Instructions)

या सूचना प्रोग्रामच्या दुसऱ्या भागात नियंत्रण हस्तांतरित करून अंमलबजावणीचा सामान्य क्रम बदलतात.

निर्देश	ऑपरेशन	उदाहरण	स्पष्टीकरण
SJMP 40H	शॉर्ट जंप	SJMP 40H	128-बाइट रेंजमध्ये 40H अड्रेसवर जंप करणे
LJMP 2000H	लाँग जंप	LJMP 2000H	2000H अड्रेसवर जंप करणे (संपूर्ण पत्त्याच्या श्रेणीमध्ये)
AJMP 300H	एब्सोल्यूट जंप	AJMP 300H	2KB रेंजमध्ये 300H अड्रेसवर जंप करणे
JZ 50H	झिरो असल्यास जंप करा	JZ 50H	झिरो असेल तर 50Hअड्रेसवर जंप करणे
JNC 100H	नो कॅरी असल्यास जंप करा	JNC 100H	कॅरी फ्लॅग = 0 असल्यास 100H अड्रेसवर जंप करा
CALL 3000H	कॉल सबरूटीन	CALL 3000H	3000H ॲड्रेसला सबरूटीन कॉल करा
RET	रिटर्न	RET	सबरूटीन मधून रिटर्न चला

Table No.4.4 Example

4.2.5. मशीन कंट्रोल इन्स्ट्रक्शन (Machine Control Instructions)

प्रोसेसरचे कार्य नियंत्रित करणारे इन्स्ट्रक्शन

निर्देश	ऑपरेशन	उदाहरण	स्पष्टीकरण
NOP	नो ऑपरेशन	NOP	काहीही करत नाही; विलंबासाठी वापरले जाते
HLT	हॉल्ट	HLT	कार्यवाही थांबवणे
SETB EA	इंटरप्ट एनबल करा	SETB EA	सर्व इंटरप्ट्स एनबल करा
CLR EA	इंटरप्ट्स डिसेबल करा	CLR EA	सर्व इंटरप्ट्स क्लिअर करा
SETB P1.0	बिट सेट करा	SETB P1.0	P1.0 बिट सेट करा (पोर्ट 1, बिट 0).

Table No.4.5 Example

4.2.6. स्टॅक ऑपरेशन इन्स्ट्रक्शन (Stack Operation Instructions)

या निर्देशांचा वापर स्टॅक ऑपरेशन्ससाठी केला जातो, जसे की डेटा पुश करणे किंवा पॉप करणे

निर्देश	ऑपरेशन	उदाहरण	स्पष्टीकरण
PUSH 05H	स्टॅकवर पुश करा	PUSH 05H	रजिस्टर 05H स्टॅकवर पुश करा
POP 03H	स्टॅकवरून पॉप करा	POP 03H	स्टॅकचा टॉप रजिस्टर 03H मध्ये पॉप करा
MOV SP, #60H	स्टॅक पॉइंटर लोड करा	MOV SP, #60H	60H मूल्य स्टॅक पॉइंटरमध्ये (SP) हलवा

Table No.4.6 Example

4.2.7. बुलियन इन्स्ट्रक्शन (Boolean Instructions)

या इन्स्ट्रक्शनचा वापर व्यक्तिशः बिट्सवर कार्य करण्यासाठी केला जातो

निर्देश	ऑपरेशन	उदाहरण	स्पष्टीकरण
SETB C	कॅरी फ्लॅग सेट करा	SETB C	कॅरी फ्लॅगला 1 वर सेट करा
CLR C	कॅरी फ्लॅग क्लियर करा	CLR C	कॅरी फ्लॅग क्लियर (C = 0)
CPL C	कॅरी फ्लॅग इन्वर्ट करा	CPL C	कॅरी फ्लॅग इन्वर्ट
ANL P1, #0FH	AND ऑपरेशन	ANL P1, #0FH	Port 1 आणि 0FH यांच्यात AND ऑपरेशन करते.
ORL A, P2	OR ऑपरेशन	ORL A, P2	अक्युम्युलेटर A आणि Port 2 यांच्यात OR ऑपरेशन करते

Table No.4.7 Example

4.3 अॅस्सेम्बलर निर्देश: ORG, DB, EQU, END, CODE, DATA. (Assembler Directives: ORG, DB, EQU, END, CODE, DATA.)

अॅस्सेम्बलर निर्देश: (ASSEMBLER DIRECTIVES)

अॅस्सेम्बलर निर्देश हे विशेष निर्देश असतात जे असेंबलरद्वारे प्रोग्राम संरचना निश्चित करण्यासाठी, मेमोरी आरक्षित करण्यासाठी, आणि सिम्बॉलिक नाव कॉन्स्टंटला प्रदान करण्यासाठी वापरले जातात. हे निर्देश नोट एक्झिक्युटेबल (not executable)आहेत, पण ते कार्यक्षमतेत सुधारणा करण्यास मदत करतात.

खालील 8051 असेंबली निर्देश हे अत्यधिक वापरले जाणारे असेंबली निर्देश आहेत:

4.3.1 ORG (ओरिजिन) निर्देश (Origin) Directive

- उद्दीष्ट: प्रोग्राम किंवा डेटाचे प्रारंभिक अड्रेस मेमोरीत निश्चित करणे.
- सिंटॅक्स: ORG अॅड्रेस

उदाहरण:

ORG 0000H ; प्रोग्रामचा प्रारंभ अड्रेस 0000H वर सेट करा

MOVA, #30H ; अॅक्युम्युलेटरमध्ये 30H लोड करा

हे अॅस्सेम्बलरला सांगते की कोड 0000H अड्रेसपासून सुरू होईल.

4.3.2 DB (डिफाईन बाईट्) निर्देश (Define Byte) Directive

- उद्दीष्ट: मेमोरी स्पेस आरक्षित करणे आणि त्यात बाइट व्हॅल्यू असाइन करणे.
- DB डायरेक्टिव्हचा वापर बाइट्सचा क्रम (वर्ण किंवा संख्यात्मक मूल्ये) परिभाषित करण्यासाठी केला जातो.
- हे सामान्यतः मेमरीमध्ये स्ट्रिंग डेटा किंवा कॉन्स्टन्ट व्हॅल्यू ज संग्रहित करण्यासाठी वापरले जाते
- सिंटॅक्स: लेबल DB व्हॅल्यू

उदाहरण:

- DB "HELLO";
- असेंबलर "HELLO" स्ट्रिंगच्या प्रत्येक अक्षराला त्याच्या ASCII इक्विव्हॅलेंट मध्ये रूपांतरित करतो आणि त्यांना क्रमाने मेमरीमध्ये संग्रहित करतो. हे सामान्यतः लुकअप टेबल्स, ASCII स्ट्रिंग्स आणि कॉन्स्टन्ट परिभाषित करण्यासाठी वापरले जाते.

4.3.3 EQU (इक्वेट) निर्देश (Equate) Directive

- उद्दीष्ट: प्रतीकात्मक नावाला एक निश्चित मूल्य असाइन करणे.
- सिंटॅक्स: लेबल EQU व्हॅल्यू

उदाहरण: COUNT EQU 0AH ; COUNT ला 0AH म्हणून व्याख्यायित(Define)करा

MOV R0, #COUNT ; R0 मध्ये 0AH लोड करा

यामुळे COUNT चे सर्व घटक 0AH ने बदलले जातात, ज्यामुळे कोडमध्ये सुधारणा करणे आणि वाचन करणे सोपे होते.

4.3.4 END (एंड ऑफ प्रोग्राम) निर्देश (End of Program) Directive

- उद्दीष्ट: सोर्स कोड संपल्याचे असेम्बलरला इंडिकेट करतो
- सिंटॅक्स: एंड(END)

उदाहरण:
```
ORG 0000H
MOV A, #50H
END ; प्रोग्रॅम सम्पल्याचे दर्शवतो
```

- अॅस्सेम्बलर एंड नंतर असलेल्या कोणत्याही लाईन्स दुर्लक्षित करतो

4.3.5 कोड निर्देश (CODE Directive)

- उद्दीष्ट: प्रोग्रॅम इंस्ट्रक्शन जिथे स्टोर होतात त्या कोड सेगमेंटला डिफायन करतो
- सिंटॅक्स: कोड

उदाहरण: सेगमेंट कोड (SEGMENT CODE)

ORG 0000H

MOV A, #25H

हे अॅस्सेम्बलरला सांगते की एक्सिक्युटेबल कोड या सेक्शनमध्ये आहे.

4.3.6. डाटा निर्देश (DATA Directive)

- उद्दीष्ट: एक डेटा विभाग निश्चित करणे, ज्यात वेरिएबल्स आणि कॉन्स्टंट संग्रहित (store) होतात.
- सिंटॅक्स: डाटा

उदाहरण:

सेगमेंट कोड (SEGMENT DATA)

ORG 30H

VALUE DB 0AH ; ऍड्रेस 30H वर सुरुवातीच्या 0AH व्हॅलूने व्हेरिएबल डिफायन करा

व्हेरिएबल रॅम मध्ये कुठे स्टोर करायचे हे असेम्बलरला सांगते

निर्देश	उद्दीष्ट	उदाहरण
ORG	प्रोग्राम किंवा डेटाचा प्रारंभ ॲड्रेस मेमोरीत निश्चित करणे.	ORG 0000H MOV A, #30H
DB	मेमोरी स्पेस आरक्षित करणे आणि त्यात बाइट व्हॅल्यू असाइन करणे.	MESSAGE DB "HELLO", 0DH, 0AH, 00H
EQU	प्रतीकात्मक नावाला एक निश्चित मूल्य असाइन करणे.	COUNT EQU 0AH MOV R0, #COUNT
END	प्रोग्राम ची समाप्ती.	ORG 0000H MOV A, #50H END
CODE	डिफाईन कोड मेमरी सेगमेंट	SEGMENT CODE MOV A, #25H
DATA	डिफाईन डाटा मेमरी सेगमेंट	SEGMENT DATA VALUE DB 0AH

Table No 4.8 8051 Assembly Instruction

प्रश्न (Questions):

1. एड्रेसिंग मोड काय आहे? 8051 मायक्रोकंट्रोलरमध्ये वेगवेगळ्या प्रकारच्या एड्रेसिंग मोड्सचे उदाहरणांसह स्पष्टीकरण करा.
2. MOV A, #55H या इन्स्ट्रक्शन मध्ये कोणता एड्रेसिंग मोड वापरला जातो? त्याचे स्पष्टीकरण करा.
3. 8051 मध्ये लॉजिकल आणि अरीथमेटिक इन्स्ट्रक्शन मध्ये काय फरक आहे? प्रत्येक प्रकाराचे दोन उदाहरणे द्या.
4. खालील इन्स्ट्रक्शनचा कार्य स्पष्ट करा:
   (i) ADD A, R1 (ii) SUBB A, #30H
   (iii) MUL AB (iv) DIV AB
5. बूलियन निर्देशांचे(Boolean Directive) वर्णन करा आणि कमीत कमी तीन उदाहरणे द्या.
6. खालील असेंब्ली निर्देश उदाहरणांसह स्पष्ट करा:
   (i) ORG (ii) DB
   (iii) EQU (iv) END

संदर्भ पाठ्यपुस्तके (Reference Books):

Sr. No.	Author	Title	Publisher with ISBN Number
1	Udayashankara M. S. Mallikarjuna Swamy	8051 Microcontroller: Hardware, Software and application.	McGraw Hill Education; 1st edition; ISBN-13 : 978-0070086814
2	Kenneth Ayala	8051 Microcontroller ArchitectureProgramming and Application	Cengage Learning India; 3rd edition ; ISBN-13 : 978-8131502006
3	Mazidi, Mohmad Ali; Mazidi,Janice Gelispe; Mckinlay RolineD.	The 8051 Microcontroller and Embedded system	Pearson Education India; 2nd edition; ISBN-13 : 978-0199681273
4	Ajay Deshmukh	Microcontroller Theory and Application	Mc Graw Hill., New Delhi,2011, ISBN- 9780070585959

माहती संकेतस्थळ (Reference Links)

1. https://youtu.be/CNbCMb8vUo4?si=GXy_P99IfOkQmF74
2. https://youtu.be/iNszvpVRoFQ?si=EfkL_veyQEvdxYXI

युनिट - V
8051 इंटरफेसिंग आणि ॲप्लिकेशन
(8051 Interfacing and Application)

विषय निष्पत्ती (Course Outcome)

मायक्रोकंट्रोलरचा विविध एप्लीकेशन्स मध्ये वापर करा.(Use microcontroller in various applications)

विषय निष्पत्ती बरोबर जुळणारी थिअरी लर्निंग निष्पत्ती (Theory Learning Outcomes aligned to Course Outcome):

TLO 5.1 दिलेली एक्स्टर्नल मेमरी इंटरफेस करण्याच्या प्रक्रियेचे स्केचसह वर्णन करा.

TLO 5.2 दिलेल्या एक्स्टर्नल I/O उपकरणांच्या इंटरफेसिंगचे स्केचसह वर्णन करा.

5.1 मेमरी इंटरफेसिंग - प्रोग्राम आणि डेटा मेमरी (Memory interfacing - Program and Data memory)

रॉम ही एक प्रकारची नॉन-व्होलॅटाइल(Non Volatile) मेमरी आहे.यामध्ये पॉवर सप्लाय जरी बंद केला तरी डेटा लॉस होत नाही.

रॉम चे वर्गीकरण खालील प्रकारे केले जाते

- प्रोम (PROM)
- इप्रोम (EPROM)
- इइप्रोम (EEPROM)
- फ्लॅश इप्रोम (Flash EPROM)
- मास्क रॉम (Mask ROM)

8051 चा 128 KB अॅड्रेस स्पेस, एक्स्टर्नल ऍड्रेस स्पेस करण्यासाठी दोन भागांमध्ये विभागलेला आहे.

1. प्रोग्राम कोड स्पेस(Program Code space)
2. डेटा मेमरी स्पेस(Data Memory Space)

5.1.1 प्रोग्राम कोड स्पेस(Program Code Space)

प्रोग्राम स्पेस ऍक्सेस करण्यासाठी, प्रोग्राम काउंटरची आवश्यकता असते, ते सूचना शोधते आणि आणते. डेटा मिळविण्यासाठी MOVC A, @A+DPTR सूचना वापरली जाते, जिथे C म्हणजे कोड असतो. त्याचा आकार 64K बाइट्स आहे.

प्रोग्राम स्पेसमध्ये खालील प्रमाणे प्रवेश केला जातो.

प्रोग्राम कोड स्टोअर करण्यासाठी, EA पिनच्या स्टेटस प्रमाणे ऑन-चिप रॉम(ROM) किंवा ऑफ-चिप रॉम(ROM) किंवा दोन्हींचा एकत्रित वापर करू शकतो.

1. इंटरनल प्रोग्राम मेमरी (4KB) म्हणजे 0000H ते 0FFFH असते आणि एक्स्टर्नल प्रोग्राम मेमरी (60KB) म्हणजे 1000H ते FFFFH असते. EA = 1 बनवून आपण हा मोड निवडू शकतो. (आकृती 5.1 पहा).
2. एकूण एक्स्टर्नल प्रोग्राम मेमरी (64KB), म्हणजे, 0000H ते FFFFH च्या संपूर्ण रेंज मध्ये असते. EA = 0 करून हा मोड निवडू शकतो.
3. जेव्हा PSEN ग्राउंडला जोडतो, तेव्हा 8051 मायक्रोकंट्रोलर एक्स्टर्नलवरून ऑपकोड(Opcode) आणतो.
4. परंतु, जेव्हा PSEN ला VCC शी जोडतो, तेव्हा PSEN चे स्टेटस ऍक्टिव्ह नसते. कारण ती ऍक्टिव लो पिन असते. म्हणून, प्रोग्राम मेमरी 8051 च्या इंटरनल रॉम मध्ये आपोआप सेव्ह केली जाते.

5.1.1.a एक्स्टर्नल प्रोग्राम रॉम 8051 शी इंटरफेस करण्यासाठी सर्किट डायग्रामच्या स्टेप्स खालीलप्रमाणे आहेत.

(Circuit diagram to interface external program ROM with 8051)

1. स्टेप 1: EA पिन ग्राउंडला जोडा
2. स्टेप 2: PSEN ला CE आणि OE शी जोडा.
3. स्टेप 3: नंतर, पोर्ट 2 (P2.0 - P2.7) एक्स्टर्नल रॉमच्या A8 - A12 ला कनेक्ट करा.
4. स्टेप 4: ते एनेबल(Enable) करण्यासाठी ALE ला 74LS373 लॅचच्या G शी कनेक्ट करा.

5. स्टेप 5: 74LS373 चा OC ग्राउंडला जोडा.
6. स्टेप 6: पोर्ट 0 मध्ये मल्टिप्लेक्स(Multiplexed) केलेला पत्ता(Address) आणि डेटा दोन्हीचा समावेश असतो. हा डेटा डीमल्टिप्लेक्स(Demultiplexed) करण्यासाठी (P0.0 – P0.7), 74LS373 लॅचच्या 1D - 8D पिनला कनेक्ट करा आणि 1Q – 8Q लॅचच्या पिन्स एक्स्टर्नल रॉमच्या A0 – A7 ला कनेक्ट करा.
7. स्टेप 7: पोर्ट 0 (P0.0 – P0.7) एक्स्टर्नल रॉमच्या(External ROM) D0 – D7 ला कनेक्ट करा.
8. स्टेप 8: एक्स्टर्नल रॉमचे(External ROM) VPP, VCC ला कनेक्ट करा.

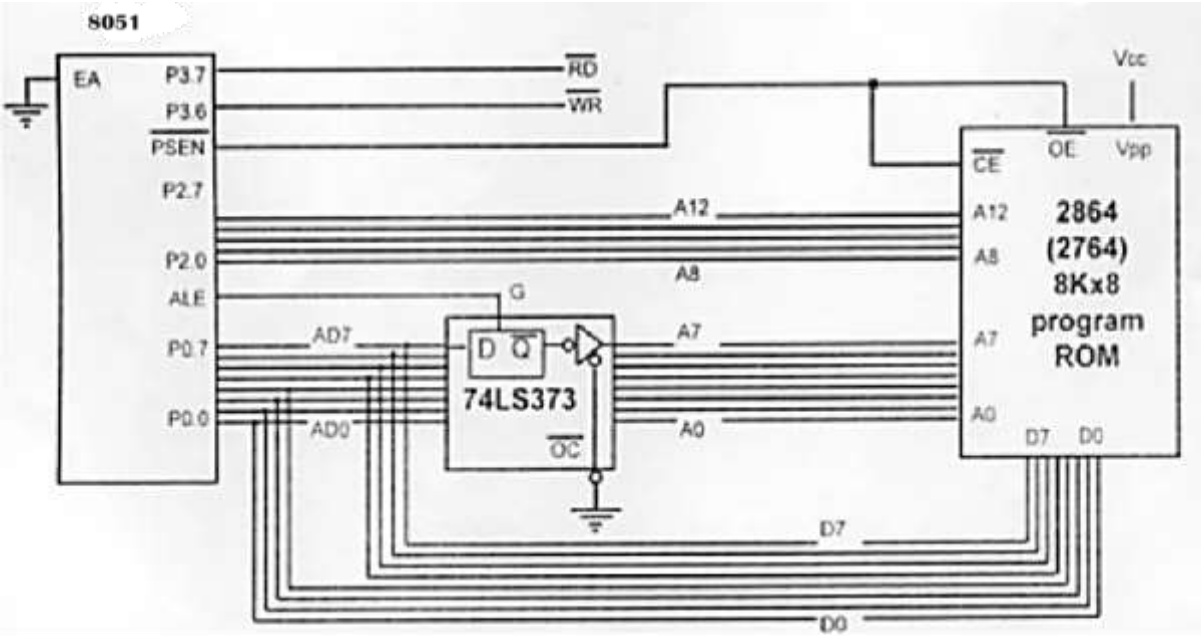


Figure 5.1- interfacing external program ROM with 8051

5.1.2 डेटा मेमरी स्पेस(Data Memory Space)

डेटा मेमरी स्पेसमध्ये प्रवेश करण्यासाठी, MOVX A, @DPTR ही सूचना वापरली जाते. RD पिन (पिन 3.7) डेटा रॉमच्या OE शी कनेक्ट करा आणि डेटा रॉमच्या चिप एनेबल (Chip Enable) पिनला ॲक्टिव्ह लो(Active Low) सिग्नल द्या. येथे डेटा असलेल्या एक्स्टर्नल रॉम मधून डेटा ॲक्सेस(Accsess) करून इंटरनल रॅम मध्ये पाठविला जातो.

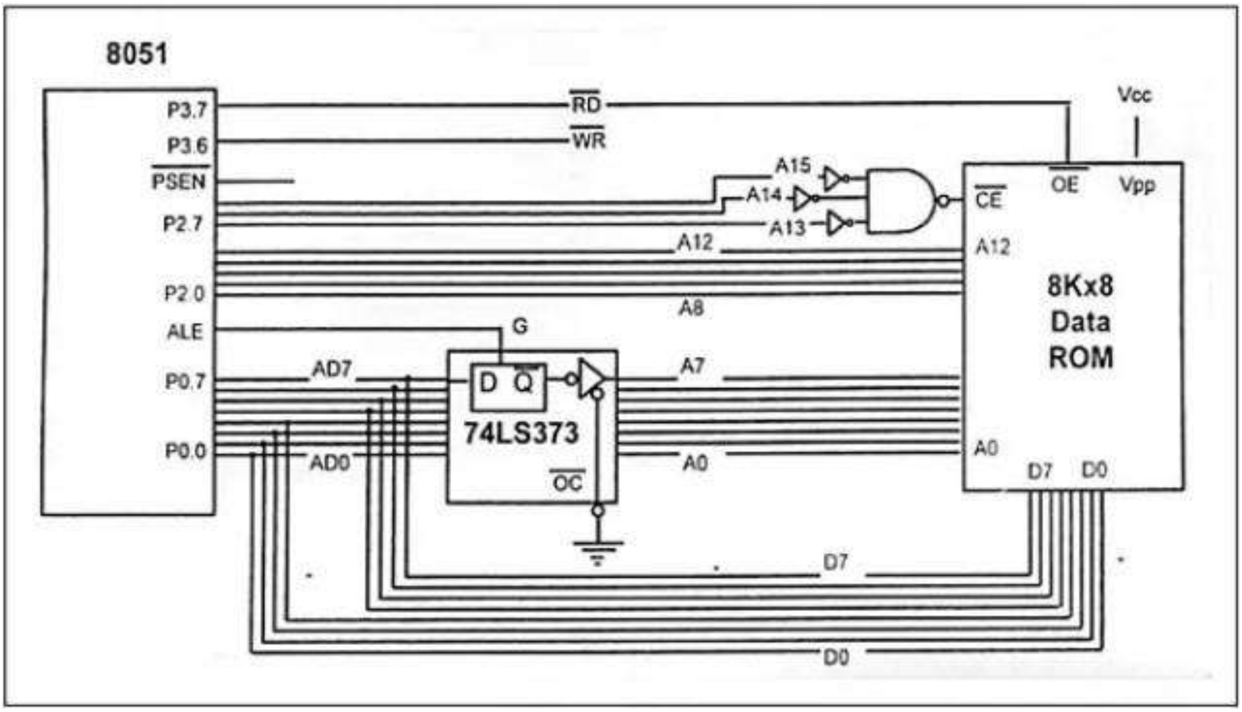


Figure 5.2- interfacing external data ROM with 8051

5.1.2.a एक्स्टर्नल रॅम – डेटासाठी (External RAM – For data)

रॅम(RAM) मेमरीला वोलाटाइल(Volatile) मेमरी असेही म्हटले जाते कारण पावर सप्लाय बंद केला की डेटा लॉस होतो. डीफॉल्ट (Default)RAM ची सुधारणा म्हणजे RAWM (रीड आणि राइट मेमरी), जी रॉमच्या विरूद्ध आहे, ज्यावर आपण कोणताही डेटा लिहू शकत नाही.

रॅमचे तीन प्रकार आहेत:

- एसरॅम (SRAM)
- एनव्हीरॅम (NV-RAM)
- डीरॅम (DRAM)

डेटा स्पेसमध्ये प्रवेश करण्यासाठी(To Access Data Space)

जर एक्स्टर्नल डेटा मेमरी, म्हणजे SRAM कनेक्ट करायची असेल, तर RD (पिन 3.7) आणि WR (पिन 3.6) SRAM डेटा मेमरीशी जोडणे आवश्यक आहे. एक्स्टर्नल डेटा RAM वर डेटा लिहिताना, खालील सूचना वापरल्या जातात.

MOVX @DPTR, A

तसेच NV-RAM देखील वापरू शकतो, कारण हा सर्वात कार्यक्षम प्रकारचा RAM आहे ज्याची मेमरी IC ची पावर बंद झाल्यानंतरही कायम राहते. प्रथम NV-RAM म्हणजे नॉन-व्होलॅटाइल रॅम आहे आणि हा मेमरी घटक आहे जो रॅम आणि रॉम या दोन्हीची उत्कृष्ट वैशिष्ट्ये एकत्र करतो.

NV-RAM ची वैशिष्ट्ये:

- यामध्ये रॅमची(RAM) रीड अंड राईट कॅपॅसिटी(Read and Write Capacity), तसेच रॉमची(ROM) नॉनवोलाटिलिटी(Non volatality) आहे .
- CMOS पासून तयार केलेले अत्यंत उर्जा-कार्यक्षम SRAM सेल असतात.
- बॅकअप उर्जा स्त्रोत म्हणून अंतर्गत लिथियम बॅटरीचा समावेश होतो.
- इंटेलिजेंट कंट्रोल सर्किटरीचा समावेश आहे.

5.1.2.b एक्स्टर्नल रॅम 8051 शी इंटरफेस करण्यासाठी सर्किट डायग्रामच्या स्टेप्स खालीलप्रमाणे आहेत. (Circuit diagram to interface external RAM with 8051)

1. स्टेप 1: RD ला एक्स्टर्नल रॅम च्या OE ला कनेक्ट करा.
2. स्टेप 2: WR ला एक्स्टर्नल रॅम च्या WE ला कनेक्ट करा.
3. स्टेप 3: NAND गेटचे ॲक्टिव्ह लो इनपुट एक्स्टर्नल रॅम च्या CE शी कनेक्ट करा, जेथे NAND गेटचे इनपुट A15, A14 आणि A13 ॲड्रेस लाईन्स आहेत. एक्स्टर्नल रॅम(External RAM) च्या 8000H स्थानामध्ये प्रवेश करण्यासाठी या लाईन्सला 0 1 0 दिले आहेत.

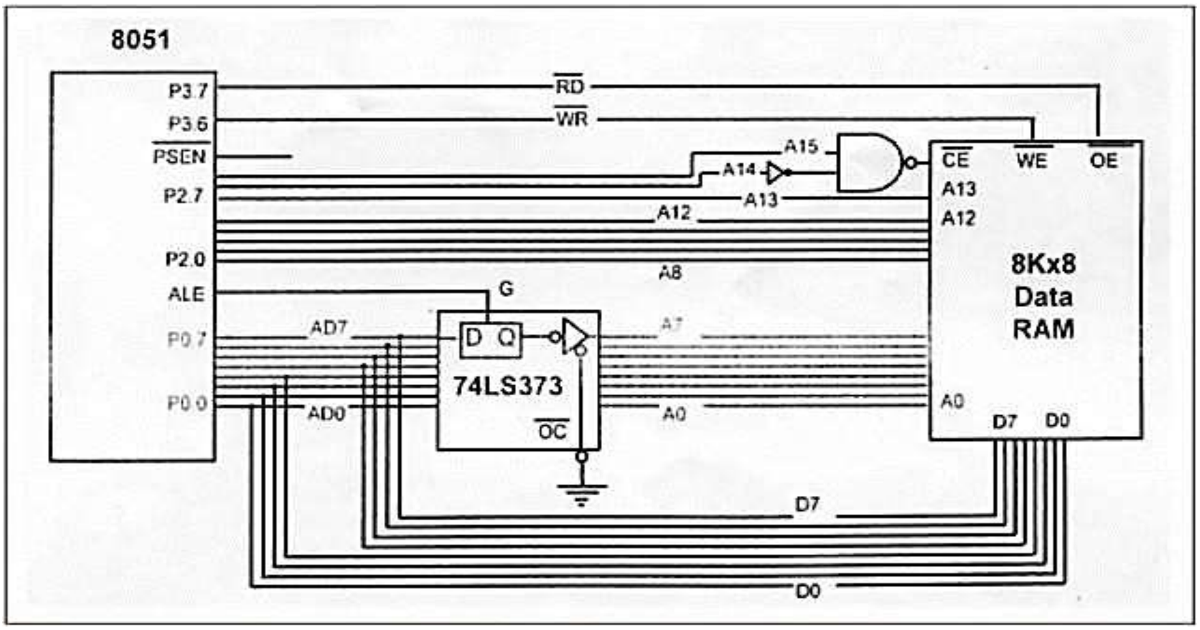


Figure 5.3- interfacing external RAM with 8051

5.1.3 एक्स्टर्नल प्रोग्राम रॉम, डेटा रॉम आणि एक्स्टर्नल रॅमचे 8051 शी इंटरफेसिंग (Interfacing external program ROM, data ROM and external RAM with the 8051)

प्रोग्राम रॉम(Program ROM) आणि डेटा रॅम(Data RAM) दोन्ही 8051 ला इंटरफेस करण्यासाठी, उदाहरणार्थ 16KB डेटा रॅम, 16KB प्रोग्राम रॉम आणि 16KB डेटा रॉम (Data ROM) चा इंटरफेस करायचा असेल, तर त्यासाठी खालील प्रमाणे स्टेप्स आहेत:

1. स्टेप 1: 16KB डेटा ॲक्सेस(Accsess) करण्यासाठी आवश्यक असलेल्या ॲड्रेस लाइनच्या(Address lines) संख्या मोजा, म्हणजे 2^14 = 16KB. येथे, 14 ॲड्रेस लाइन्स A0 - A13 आवश्यक आहेत.
2. स्टेप 2: रॅम आणि रॉम चे स्थान ठरवा, येथे 0000H वरून रॉम आणि डेटा RAM 8000H वरून इंटरफेस करणार आहोत.
3. स्टेप 3: डीकोडर(Decoder) सर्किट निवडा, येथे आपण 74LS138 डीकोडर निवडणार आहोत.
4. स्टेप 4: प्रोग्राम रॉमसाठी डीकोडर सर्किटची आवश्यकता नाही, परंतु 74LS138 डीकोडरला डेटा रॉम आणि डेटा रॅमशी कनेक्ट करावे लागेल.
5. स्टेप 5: G1 ला VCC, G2A आणि G2B ग्राउंडला कनेक्ट करा.
6. स्टेप 6: इनपुट A आणि B ला अनुक्रमे P2.6 आणि P2.7 आणि इनपुट C ग्राउंडला कनेक्ट करा.
7. स्टेप 7: एक्स्टर्नल प्रोग्राम आणि डेटा रॉम(External Program and Data ROM) कनेक्ट करण्यासाठी, यासाठी अँड गेटचा वापर केला जातो. RD आणि PSEN ही सिग्नल्स अँड गेटच्या इनपुटला दिली जातात आणि हे एक्स्टर्नल रॉमच्या OE पिनला दिले जाते
8. स्टेप 8: एक्स्टर्नल रॅम(External RAM) इंटरफेस करण्यासाठी, एक्स्टर्नल रॅमच्या अनुक्रमे WE आणि OE ला RD आणि WR शी जोडतात

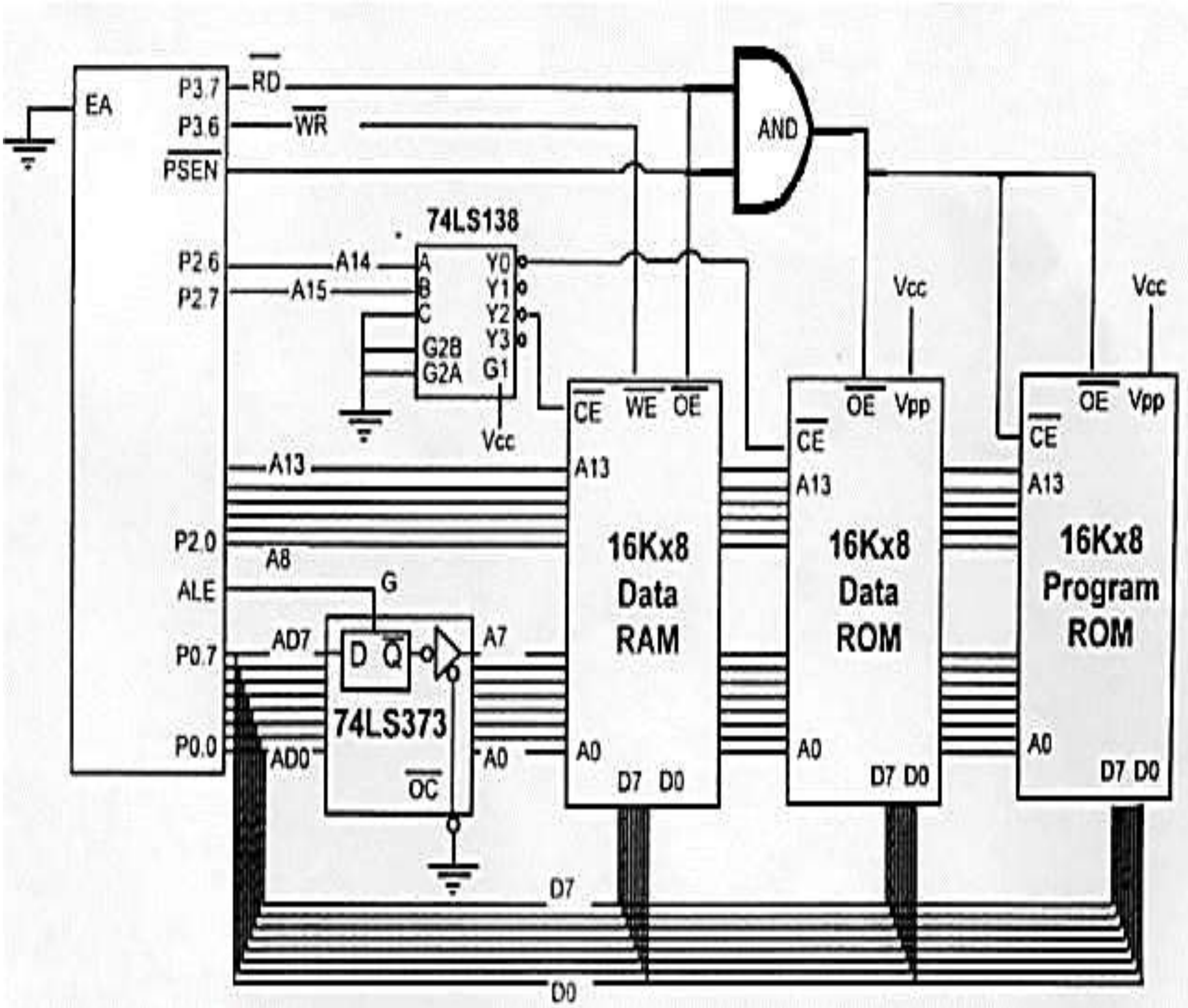


Figure 5.4- Interfacing external program ROM, data ROM and external RAM with the 8051

5.2.1 एलईडी आणि स्विचचे 8051 शी इंटरफेसिंग (Interfacing LED and Switch with 8051)

एलईडी 8051 च्या P2.0 कनेक्ट केलेला आहे आणि हा सेटअप ॲक्टिव्ह लो कॉन्फिगरेशन मध्ये आहे म्हणजेच P2.0 जेव्हा लो असेल तेव्हाच एलईडी ऑन होईल

स्विच 8051 च्या P0.0 कनेक्ट केलेला आहे. हा स्वीच ॲक्टिव्ह लो स्विच म्हणून काम करतो.

स्विच प्रेस केल्यानंतर इंटरप्ट ट्रिगर होईल आणि त्यानुसार मायक्रोकंट्रोलर एलईडी ऑन किंवा ऑफ करेल.

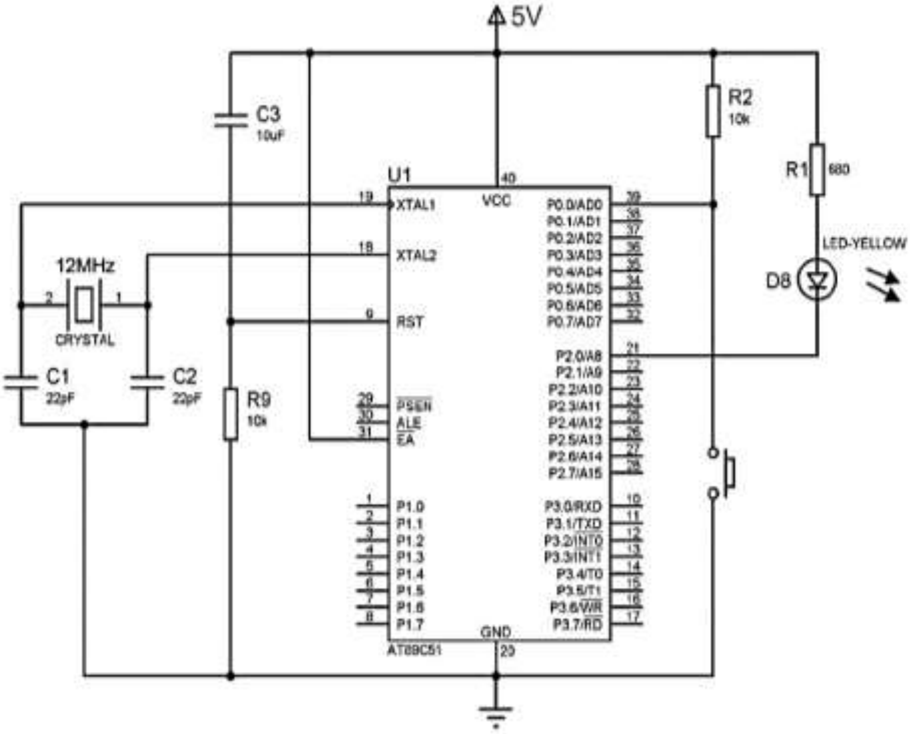
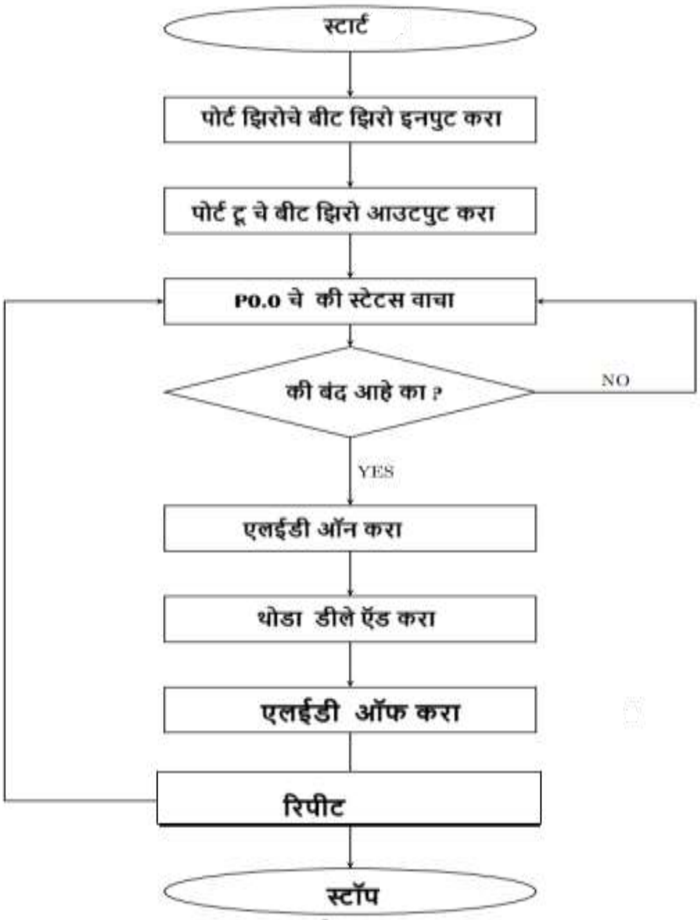


Figure 5.5- 8051 connection to switch and LED



Flowchart 5.1

असेंबली लँग्वेज प्रोग्राम (Assebly Language Program)

```
ORG 0000H
SETB P0.0
CLR P2.0
UP: SETB P2.0
HERE: JB P0.0, HERE
CLR P2.0
ACALL DELAY
SJMP UP
DELAY: MOV R2, #200
L2: MOV R3, #100
L1: DJNZ R3, L1
DJNZ R2, L2
RET
END
```

5.2.2 रिलेचे 8051 शी इंटरफेसिंग (Interfacing Relay with 8051)

8051 च्या P2.0 ला ट्रांजिस्टर(Transistor) कनेक्ट केला आहे आणि हा स्विच म्हणून काम करतो. या ट्रांजिस्टर द्वारे रिले ऑपरेट केला जातो.

जेव्हा P2.0 हाय असते तेव्हा ट्रांजिस्टर(Transistor) ऑन होऊन रिले ऍक्टिव्हेट होतो.

सर्किटमध्ये रिले च्या अक्रॉस(across) एक डायोड(Diode) दिला आहे याला फ्लाय बॅक डायोड(Flyback Diode) असे म्हणतात. हा डायोड जेव्हा रिले कॉइल डीएनर्जाईज(Deenergise) होते तेव्हा वोल्टेज स्पाईक्स मुळे होणारे नुकसान टाळतो.

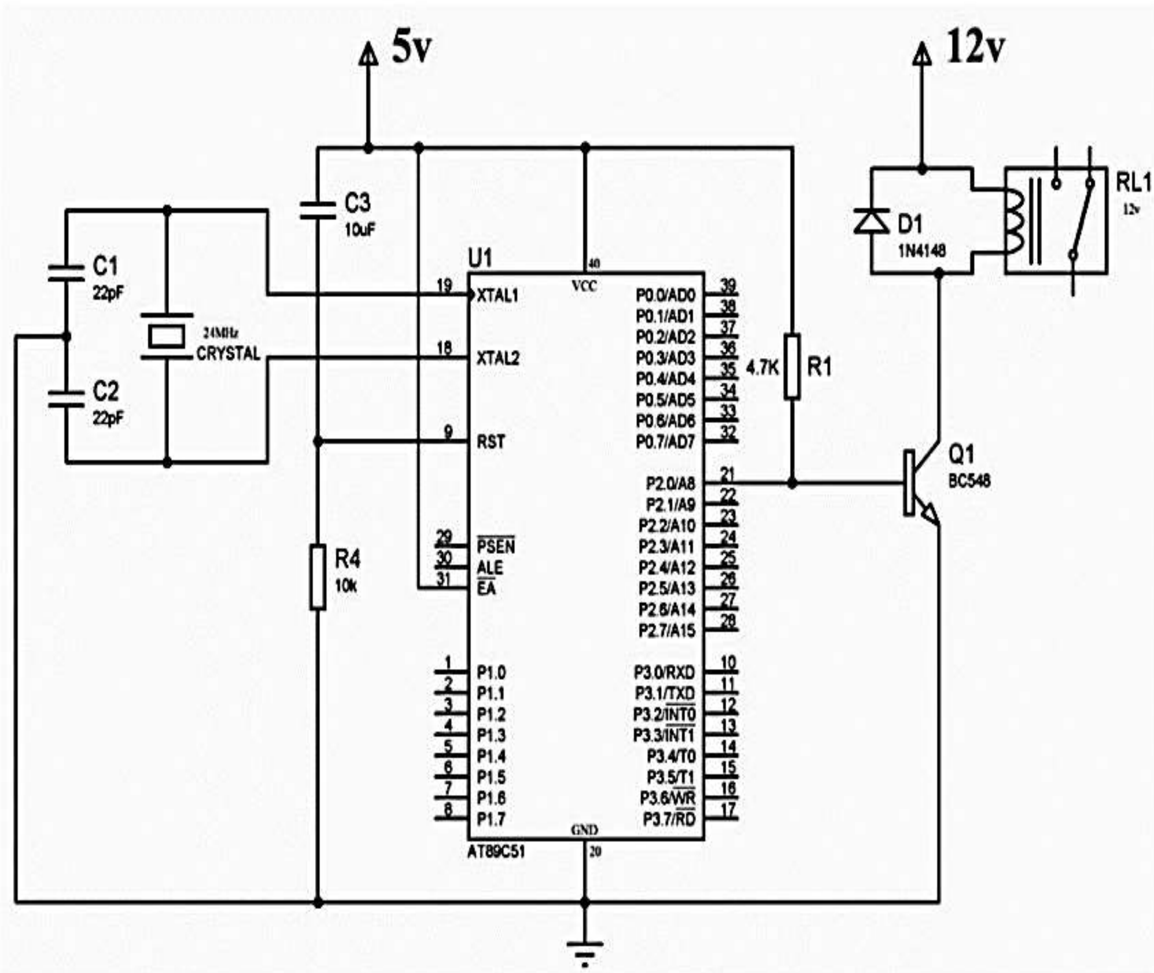
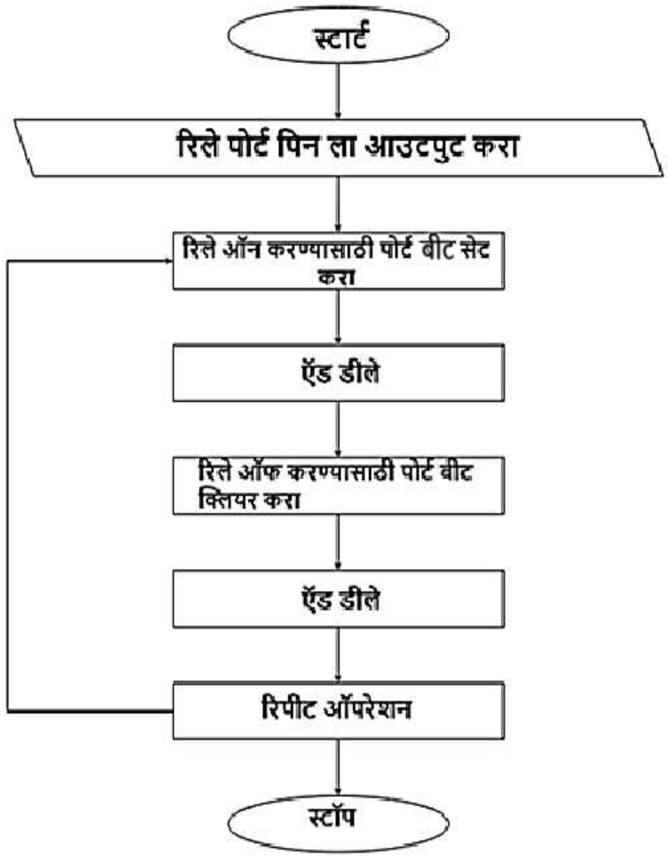


Figure 5.6- 8051 connection to Relay



Flowchart 5.2

असेंबली लॅंग्वेज प्रोग्राम (Assembly Language Program)

```
ORG 0000H
MAIN: SETB P2.0
ACALL DELAY
CLR P2.0
ACALL DELAY
SJMP MAIN
DELAY:MOV R3, #100
L3: MOV R4, #255
L2: MOV R5, #255
L1: DJNZ R5, L1
DJNZ R4, L2
DJNZ R3, L3
RET
```

5.2.3 16×2 LCD चे 8051 शी इंटरफेसिंग (Interfacing 16×2 LCD with 8051)

एलसीडी डिस्प्ले (LCD Display) हा जवळजवळ सर्व एम्बेडेड प्रोजेक्ट मध्येएक अपरिहार्य भाग आहे. 8051 मायक्रोकंट्रोलरसह 16×2 एलसीडी इंटरफेस जाणून घेतल्यास डिजिटल व्होल्टमीटर/अँमिटर, डिजिटल घड्याळ, होम ऑटोमेशन डिस्प्ले, स्टेटस इंडिकेटर डिस्प्ले, डिजिटल कोड लॉक, डिजिटल स्पीडोमीटर/ओडोमीटर इत्यादी एम्बेडेड प्रोजेक्ट तयार करू शकता.

5.2.3.a 16×2 LCD मॉड्यूल (16×2 LCD Module)

16×2 एलसीडी मॉड्यूल (LCD Module) हा एक सामान्य प्रकारचा एलसीडी मॉड्यूल आहे जो 8051 आधारित एम्बेडेड प्रोजेक्ट मध्ये वापरला जातो. यात 5×7 किंवा 5×8 LCD डॉट मॅट्रिक्सच्या 16 रोज(Rows) आणि 2 कॉलम (Column) आहेत. JHD162A या प्रकाराचा मॉड्यूल खूप लोकप्रिय आहे. हे बॅक लाइट(Back light), कॉन्ट्रास्ट ऍडजस्टमेंट(Contrast adjustment) फंक्शनसह 16 पिन पॅकेजमध्ये उपलब्ध आहे आणि प्रत्येक डॉट मॅट्रिक्समध्ये

(Dot Matrix) 5×8 डॉट रिझोल्यूशन आहे. पिन क्रमांक, त्यांचे नाव आणि संबंधित कार्ये खालील तक्त्यामध्ये दर्शविली आहेत.

पिन क्र	कार्य	नाव	पिन क्र	कार्य	नाव
1	ग्राउंड (0V)	ग्राउंड	9	8 बिट डेटा पिन	DB2
2	सप्लाय व्होल्टेज; 5V (4.7V - 5.3V)	Vcc	10	8 बिट डेटा पिन	DB3
3	कॉन्ट्रास्ट ऍडजस्टमेंट; POT द्वारे	VEE	11	8 बिट डेटा पिन	DB4
4	कमांड रजिस्टर निवडते	(RS)	12	8 बिट डेटा पिन	DB4
5	डेटा रजिस्टर निवडा	रीड/ राईट(RW)	13	8 बिट डेटा पिन	DB6
6	जेव्हा हाय ते लो पल्स दिले जाते तेव्हा डेटा पिनवर डेटा पाठवते	एनेबल(E)	14	8 बिट डेटा पिन	DB7
7	8 बिट डेटा पिन	DB0	15	बॅकलाइट Vcc (5V)	LED+
8	8 बिट डेटा पिन	DB1	16	बॅकलाइट ग्राउंड (0V)	LED- (GND)

Table 5.1 - Operating parameters – Pin description of LCD module

VEE पिन हा LCD डिस्प्लेचा कॉन्ट्रास्ट अड्जस्ट करण्यासाठी आहे आणि या पिनवर व्होल्टेज बदलून कॉन्ट्रास्ट अड्जस्ट केला जाऊ शकतो. हे POT चे एक टोक Vcc (5V) ला जोडून, दुसरे टोक ग्राउंडला जोडून आणि POT चे मध्यवर्ती टर्मिनल (Wiper)) VEE पिनला जोडून केले जाते. अधिक चांगल्या प्रकारे समजून घेण्यासाठी सर्किट डायग्राम पहा.

JHD162A मध्ये डेटा रजिस्टर(Data Register) आणि कमांड रजिस्टर(Command Register) असे दोन अंगभूत रजिस्टर आहेत. डेटा रजिस्टर डेटा डिस्प्ले करण्यासाठी आहे, आणि कमांड रजिस्टर कमांड ठेवण्यासाठी आहे. 16×2 LCD मॉड्यूलमध्ये डिस्प्लेसह विशिष्ट कार्य करण्यासाठी प्रत्येक कमांडचा संच असतो. RS पिन स्टेटस हाय असेल तर ते डेटा रजिस्टर निवडेल आणि RS पिनचे स्टेटस लो असेल तर ते कमांड रजिस्टर निवडेल. जर RS पिन हाय केली आणि 8 बिट डेटा लाइनमध्ये डेटा ठेवला (DB0 ते DB7), तर LCD मॉड्यूल ते डिस्प्ले करण्यासाठी डेटा म्हणून ओळखेल. जर आरएस पिन लो केली आणि डेटा लाईनवर डेटा ठेवला, तर मॉड्यूल त्यास कमांड म्हणून ओळखेल.

R/W पिन रीड आणि राईट मोड(Read and Write) निवडण्यासाठी आहे. या पिनवरील हाय लॉजिक रीड मोड एनेबल करते आणि या पिनवरील लो लॉजिक राईट मोड एनेबल करते.

ई पिन मॉड्यूल ऍक्टिव्ह करण्यासाठी आहे. या पिनवरील हाय ते लो ट्रान्झिशन(transition) मॉड्यूल ऍक्टिव्ह करते.

DB0 ते DB7 डेटा पिन आहेत. डिस्प्ले करावयाचा डेटा आणि कमांड सूचना या पिनवर ठेवल्या जातात.

LED+ हा बॅक लाईट LED चा एनोड आहे आणि हि पिन Vcc शी योग्य सिरीज करंट लिमिटिंग रेझिस्टरद्वारे जोडलेला असणे आवश्यक आहे. LED- बॅक लाईट LED चे कॅथोड आहे आणि ही पिन ग्राउंडला जोडलेली असणे आवश्यक आहे.

5.2.3.b 16×2 LCD मॉड्यूल कमांड

16×2 LCD मॉड्यूलमध्ये प्रीसेट कमांड सूचनांचा संच आहे. प्रत्येक कमांड एक विशिष्ट कार्य करण्यासाठी मॉड्यूल बनवेल. सामान्यतः वापरल्या जाणाऱ्या कमांड्स आणि त्यांचे कार्य खालील तक्त्यामध्ये दिले आहेत.

हेक्स मध्ये कोड	LCD इन्स्ट्रक्शन रजिस्टरसाठी कमांड्स	हेक्स मध्ये कोड	LCD इन्स्ट्रक्शन रजिस्टरसाठी कमांड्स
01	डिस्प्ले स्क्रीन क्लियर करा	OE	डिस्प्ले चालू, कर्सर ब्लिंकिंग
02	रिटर्न होम	OF	डिस्प्ले चालू, कर्सर ब्लिंकिंग
04	डिक्रिमेंट कर्सर (कर्सर डावीकडे हलवा)	10	कर्सरची पोझिशन डावीकडे हलवा
06	इन्क्रिमेंट कर्सर (कर्सर उजवीकडे हलवा)	14	कर्सरची पोझिशन उजवीकडे हलवा
05	डिस्प्ले उजवीकडे हलवा	18	संपूर्ण डिस्प्ले डावीकडे शिफ्ट करा
07	डिस्प्ले डावीकडे हलवा	IC	संपूर्ण डिस्प्ले उजवीकडे शिफ्ट करा
08	डिस्प्ले ऑफ, कर्सर बंद	80	कर्सरला पहिल्या लाईन पासून सुरुवात करण्यासाठी फोर्स करणे
0A	डिस्प्ले ऑफ, कर्सर चालू	C0	कर्सरला दुसऱ्या लाईन पासून सुरुवात करण्यासाठी फोर्स करणे
0C	डिस्प्ले चालू, कर्सर बंद	38	2 लाइन्स आणि 5*7 मॅट्रिक्स (8 डेटा लाइन)

Table 5.2- LCD commands

एलसीडी इनिशिलायझेशन (LCD Initialisation)

एलसीडी डिस्प्ले सुरू करण्यासाठी कराव्या लागणाऱ्या स्टेप्स (steps) खाली दिल्या आहेत आणि या पायऱ्या जवळजवळ सर्व ॲप्लिकेशन्ससाठी सामान्य आहेत.

- इनिशिलायझेशनसाठी 8 बिट डेटा लाइनवर 38H पाठवा
- LCD चालू, कर्सर चालू आणि कर्सर ब्लिंकिंग चालू करण्यासाठी 0FH पाठवा.
- कर्सरची स्थिती वाढवण्यासाठी 06H पाठवा.
- डिस्प्ले क्लिअर (clear) करण्यासाठी 01H पाठवा आणि कर्सर परत करा.

एलसीडीला डेटा पाठवणे(Sending data to the LCD.)

एलसीडी मॉड्यूलला (LCD Module)डेटा पाठवण्याच्या स्टेप्स खाली दिल्या आहेत. एलसीडी मॉड्यूलमध्ये RS, R/W आणि E या पिन आहेत. या प्रिन्स वरील लॉजिक स्टेटस नुसार मॉड्युल ठरवते की दिलेला डेटा कमांड आहे की हा डेटा डिस्प्ले करण्यासाठी आहे.

- R/W ला लो करा.
- डेटा बाइट कमांड असल्यास RS=0 करा आणि डेटा बाइट हा डेटा आहे दाखवायचा असल्यास RS=1 करा.
- डेटा रजिस्टरवर डेटा बाइट ठेवा.
- पल्स ई(Pulse E) हाय ते लो करा
- दुसरा डेटा पाठवण्यासाठी वरील स्टेप्स पुनरावृत्ती करा.

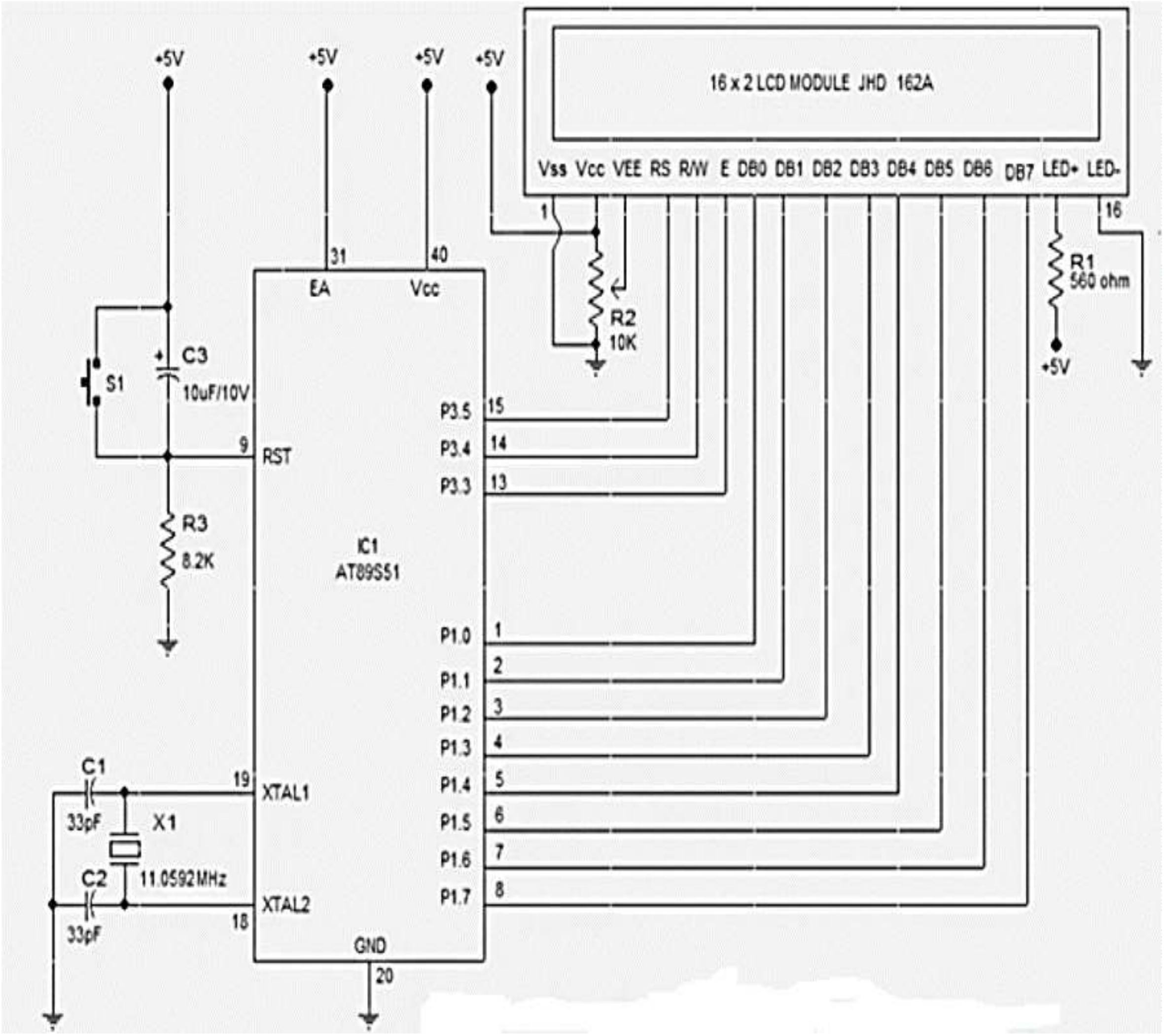
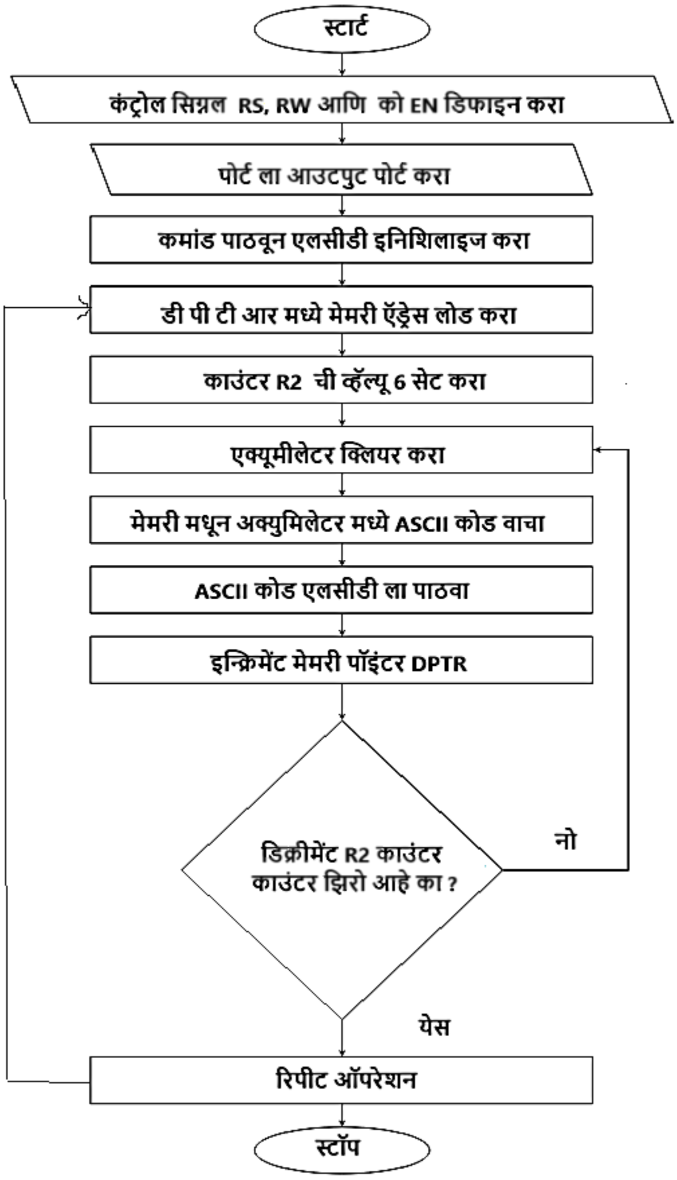


Figure 5.7- 8051 connection to LCD

वर दिलेला सर्किट डायग्राम AT89S1 मायक्रोकंट्रोलरसह 16×2 LCD मॉड्यूलचा इंटरफेस कसा करायचा ते दाखवतो. कॅपेसिटर C3, रेझिस्टर R3 आणि पुश(Push) बटण स्विच S1 रीसेट सर्किटरी(Reset Circuitry) बनवते. सिरॅमिक कॅपेसिटर C1,C2 आणि क्रिस्टल X1 हे क्लॉक सर्किटशी(Clock Circuit) संबंधित आहेत जे सिस्टम क्लॉक ची फ्रिक्वेन्सी तयार करते. मायक्रोकंट्रोलरचे P1.0 ते P1.7 पिन अनुक्रमे मॉड्यूलच्या DB0 ते DB7 जोडलेले असतात आणि या मार्गाने डेटा LCD मॉड्यूलकडे जातो. P3.3, P3.4 आणि P3.5 मायक्रोकंट्रोलरच्या E, R/W, RS पिनशी जोडलेले आहेत आणि या मार्गाद्वारे कंट्रोल सिग्नल LCD मॉड्यूलमध्ये ट्रान्सफर केले जातात. रेझिस्टर R1 बॅक लाईट LED द्वारे करंट कंट्रोल करतो आणि त्याचप्रमाणे बॅक लाइटची तीव्रता कंट्रोल देखील करते. डिस्प्लेचा कॉन्ट्रास्ट अड्जस्ट(Contrast adjust) करण्यासाठी POT R2 वापरला जातो. LCD ते 8051 मायक्रोकंट्रोलर इंटरफेस करण्यासाठी प्रोग्राम खाली दर्शविला आहे.



Flowchart 5.3

असेंबली लँग्वेज प्रोग्राम(Assembly Language Program)

```
RS BIT P3.3
RW BIT P3.4
EN BIT P3.5
ORG 0000H
MOV P1, #00H
LCALL LCD_INIT
MOV DPTR, #MSG
MOV R2, #10
DISP_MSG: CLR A
MOVC A, @A+DPTR
LCALL LCDDATA
INC DPTR
DJNZ R2, DISP_MSG
SJMP $
LCD_INIT: MOV A, #38h
ACALL LCDCMD
MOV A, #0Eh
```

```
ACALL LCDCMD
MOV A, #06h
ACALL LCDCMD
MOV A, #01h
ACALL LCDCMD
MOV A, #80H
ACALL LCDCMD
RET
LCDCMD:MOV P1, A
CLR RS
CLR RW
SETB EN
CLR EN
ACALL DELAY
RET
LCDDATA:MOV P1, A
SETB RS
CLR RW
SETB EN
CLR EN
ACALL DELAY
RET
DELAY: MOV R3, #50
L2: MOV R4, #255
L1: DJNZ R4, L1
DJNZ R3, L2
RET
ORG 0050H
MSG:DB "uC AT89c51"
END
```

5.2.4 स्टेपर मोटरचे 8051 शी इंटरफेसिंग(Interfacing of 8051 to stepper motor)

स्टेपर मोटर 360 डीग्री रोटेशन पूर्ण करण्यासाठी लहान अँगल्सने(angle) फिरते, या लहान अँगल्सना स्टेप्स म्हणतात, म्हणून स्टेपर मोटर असे नाव आहे. सामान्यतः, स्टेपर मोटरमध्ये 200 स्टेप्स असतात.

200 स्टेप्स = 360 डीग्री

1 स्टेप = 'x' डीग्री

x = 360 / 200 = 1.8 डीग्री

म्हणून, प्रत्येक स्टेप 1.8 डीग्री असते.

स्टेपर मोटरच्या स्टेटर कॉइलला हाय आणि लो पल्सेस दिल्या जातात.येथे 4 कॉइल स्टेटर वापरला आहे. प्रत्येक स्टेपचा अँगल रोटरमधील स्टेप्स आणि स्टेटरच्या अलाइनमेंट द्वारे ठरवले जाते.

200 स्टेप्स मोटरच्या बाबतीत, स्टेप एंगल 360/200 = 1.8 डिग्री असतो आणि 8 स्टेप रोटरच्या बाबतीत, स्टेप अँगल 360/8 = 45 डिग्री असतो.

8051शी स्टेपर मोटरला इंटरफेस करण्यासाठी स्टेप्स खालीलप्रमाणे आहेत

1. स्टेप 1: Proteus किंवा इतर सिम्युलेशन सॉफ्टवेअर किंवा अगदी हार्डवेअर वापरत असल्यास, AT89C51 किंवा AT89S51 मायक्रोकंट्रोलर किंवा इतर कोणतेही सुसंगत प्रकार निवडा. (AT89C51 हे Atmel फॅमिलीमधील 8 बिट मायक्रोकंट्रोलर आहे जे 8051 आर्किटेक्चरसह कार्य करते.)
2. स्टेप 2: पिन 18 आणि 19 दरम्यान 12 मेगाहर्ट्झ ऑसिलेटर(12 MHz Oscillator) कनेक्ट करा.

3. स्टेप 3: खाली दाखवल्याप्रमाणे 22pF चे दोन कॅपेसिटर कनेक्ट करा, एक टर्मिनल ऑसिलेटरच्या कोणत्याही एका बाजूला आणि दुसरे टर्मिनल ग्राउंडला आहे.
4. स्टेप 4: +5V DC स्त्रोताशी पिन 31 ला जोडा, म्हणजेच EA पिन हाय सेट करा.
5. स्टेप 5: रिसेट सर्किट बनवण्यासाठी, 10μF च्या कॅपेसिटरद्वारे पिन 9 (RST) ला +5V ला कनेक्ट करा आणि त्याच पिनला 10kΩ रेझिस्टर(Resistor) किंवा पोटेंशियोमीटरद्वारे(Potentiometer) +0V (GND) ला कनेक्ट करा.
6. स्टेप 6: पोर्ट 1 च्या 4 पिन खालीलप्रमाणे ULN2003 IC च्या इनपुट बाजूशी जोडा,
P1.0 ते 1
P1.1 ते 2
P1.2 ते 3
P1.3 ते 4
7. स्टेप 7: खालील सर्किट आकृतीत दाखवल्याप्रमाणे ULN2003 चे आउटपुट स्टेपर मोटरच्या 4 वायर्सशी जोडा.
8. स्टेप 8: स्टेपर मोटरच्या दोन कॉमन वायर आणि IC चा पिन 9, 12V DC सप्लायला जोडा.

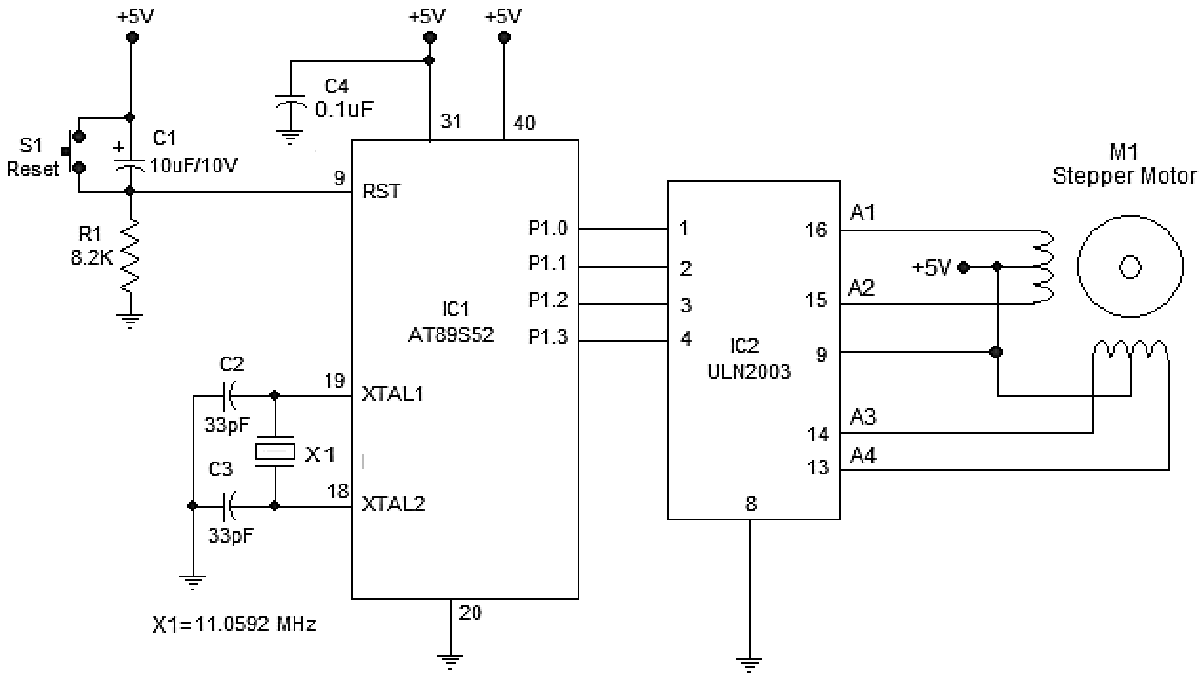


Figure 5.8 - interfacing of stepper motor to 8051

5.2.4.a स्टेपर मोटरला क्लॉक वाईज दिशेने फिरवण्यासाठी असेंबली लँग्वेज प्रोग्राम(ALP)

स्टेप अँगल = 1.80 , आवश्यक स्टेप्स नंबर= 360/1.8 = 200d =C8H

```
ORG 0000H
MOV R0,#0C8H
MOV A,#99H L1: MOV P1,A
ACALL DELAY
RR A
DJNZ R0,L1
DELAY: MOV R7,#4
WAIT2: MOV R6,#0FFH
WAIT1: MOV R5,#0FFH
WAIT: DJNZ R5,WAIT
DJNZ R6,WAIT1
DJNZ R7,WAIT2
RET
END
```

5.2.4.b स्टेपर मोटरला अँटीक्लॉक वाईज दिशेने फिरवण्यासाठी असेंबली लँग्वेज प्रोग्राम(ALP)

```
ORG 0000H
MOV A, #66H ; load step sequence
BACK: MOV P1, A ; issue sequence to motor
AGAIN: RL A ; rotate left anticlockwise
ACALL DELAY ; wait
SJMP BACK ; keep going
DELAY ; delay subroutine.
MOV R2, #100
H1: MOV R3, #255
H2: DJNZ R3, H2
DJNZ R2, H1
RET
```

प्रश्न (Questions):

1. 2KB साईज च्या दोन रॅम चीप, 8051 मायक्रोकंट्रोलर सोबत इंटरफेस करण्या साठी स्केच ड्रॉ करा.
2. 8051 च्या पोर्ट 1 शी LEDs इंटरफेस करण्यासाठी ALP लिहा तसेच इंटरफेसिंग डायग्राम सुद्धा ड्रॉ करा.
3. 8051 च्या पोर्ट 1 शी रिले इंटरफेस करण्यासाठी ALP लिहा तसेच इंटरफेसिंग डायग्राम सुद्धा ड्रॉ करा.
4. खालील दिलेल्या एलसीडी डिस्प्ले पिनचे फंक्शन्स लिहा

(i) R/W (ii) RS

5. स्टेपर मोटर, 8051 शी इंटरफेस करा आणि स्टेपर मोटर क्लॉकवाईज दिशेने फिरवण्यासाठी ALP लिहा.
6. स्टेपर मोटर, 8051 शी इंटरफेस करा आणि स्टेपर मोटर अँटीक्लॉकवाईज दिशेने फिरवण्यासाठी ALP लिहा.

संदर्भ पाठ्यपुस्तके (Reference Books):

Sr No	Author	Title	Publisher with ISBN Number
1	Mazidi Muhammad Ali, Mazidi Janice Gillispe, Mckinlay Rolin D	The 8051 Microcontroller and Embedded Systems: Using Assembly and C	Pearson Publication, 2017 ISBN: 9788131710265
2	Ayala Kenneth J	The 8051 Microcontroller	Thomson Delmar Learning, 2004 ISBN: 9781401861582
3	Deshmukh Ajay V	Microcontroller: Theory and Application	McGraw Hill,2011 ISBN: 9780070585959
4	Pal Ajit	Microcontrollers: Principle and Application	PHI Learning, 2014 ISBN: 978812034394
5	Chattopadhyay Santanu	Microcontroller and Applications	All India Council for Technical Education, 2023 ISBN: 9788196057602

माहती संकेतस्थळ (Reference Links)

1.https://www.keil.com/download/ Simulation software
2.https://archive.nptel.ac.in/courses/108/105/108105102/
3.https://nptel.ac.in/courses/117104072
4.https://play.google.com/store/apps/details?id=com.coderbro.tutorial.a8051microcontroller&hl=en_IE
5.**Diagram Courtesy** https://technobyte.org/8051-external-memory-interfacing-ram-rom/
6.**Diagram Courtesy** https://www.circuitstoday.com/interfacing-16x2-lcd-with-8051
7.**Diagram Courtesy** https://technobyte.org/8051-stepper-motor-interfacing/