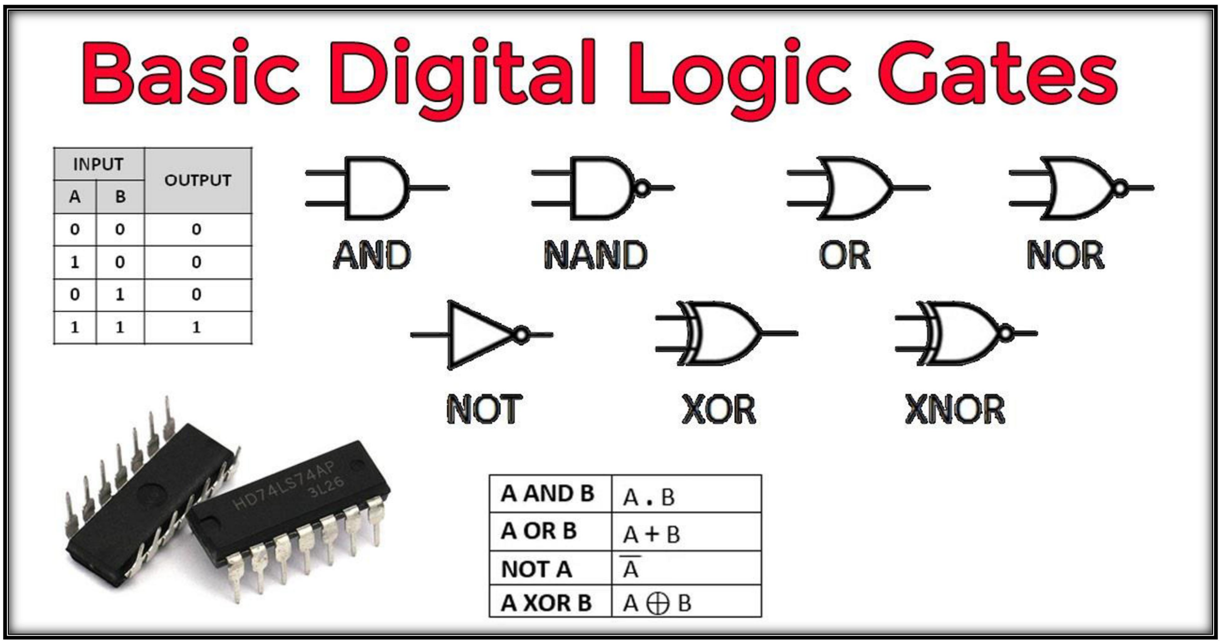
~ Digital Device ~

- এটা কোনো নোট/গাইড/সাজেশন না । এটা ৩য় অধ্যায় (২য় অংশ) ডিজিটাল ডিভাইসের লেকচার মাত্র । এই লেকচারের মাধ্যমে কঠিন বিষয়গুলো সহজে বোঝার জন্য তৈরি করা হয়েছে । যা একটি ক্ষুদ্র প্রচেষ্টা ।
- এই লেকচার পড়ে পরীক্ষায় পাশের আশা না করাই ভালো । এই লেকচার কখনই মূল বইয়ের বিকল্প নয় ।
- ডিজিটাল ডিভাইসকে একটু ভিন্নভাবে উপস্থাপন করার প্রচেষ্টা ।



Md. Minhazul Kabir
https://minhazulkabir.com/
mdminhazulkabir@gmail.com
CSE, SUST

ডিজিটাল ডিভাইস

 # লজিক গেইটঃ বুলিয়ান অ্যালজেবরায় মৌলিক কাজগুলো বাস্তবায়নের জন্য যে সমস্ত ডিজিটাল ইলেকট্রনিক সার্কিট এক বা একাধিক ইনপুট গ্রহন করে এবং একটিমাত্র আউটপুট প্রদান করে এবং যুক্তিভিত্তিক সংকেতের প্রবাহ নিয়ন্ত্রন করে তাদেরকে লজিক গেইট বলে।

মৌলিক লজিক গেইটঃ

→ বুলিয়ান অ্যালজেবরার মৌলিক অপারেশনগুলো বাস্তবায়ন করা যায়।

→ এর সাহায্যে সকল যৌগিক গেইট ও যেকোনো সার্কিট করা যায়।

* 0 এবং 1 দুইটি সংখ্যা হলেও ডিজিটাল ডিভাইসে 0 এবং 1 দুটি প্রতিক। যাদের মান~

0 → No, False, Off, Wrong / না, মিথ্যা, বন্ধ, ভুল, Low

1 → Yes, True, On, Right / হ্যাঁ, সত্য, চালু, শুদ্ধ, High

1. AND Gate:

→ গুনের কাজ করে

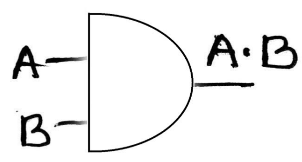
→ যৌক্তিক গুনের নিয়ম মেনে চলে

→ ন্যূনতম দুটি ইনপুট এবং সর্বোচ্চ অসীম সংখ্যক ইনপুট নেয়।

দুটি ইনপুট (A ও B) বিশিষ্ট AND গেট:

a. AND এর লাস্ট অক্ষর D. সেজন্য AND গেট দেখতে ইংরেজি অক্ষর D এর মতন।

b. এখানে ইনপুটে A, B দেয়া হয়েছে। যেহেতু এটা AND গেইট। সেজন্য, আউটপুটে গুন অবস্থায় AB পাওয়া যায়।



AND gate এর লজিক চিহ্ন

আমরা জানি,
কোনো কিছুর সাথে শূন্য গুন করলে শূন্য হয় গুনফল। যেমন:

$110409 \times 1027 \times 0 = 0$

হাম্বু × জাম্বু × গাম্বু × 0 = 0

সত্যক সারনীর মাধ্যমে বুঝতে পারব।

input		output
A	B	A·B
0	0	0
0	1	0
1	0	0
1	1	1

$0 \times 0 = 0$

$0 \times 1 = 0$

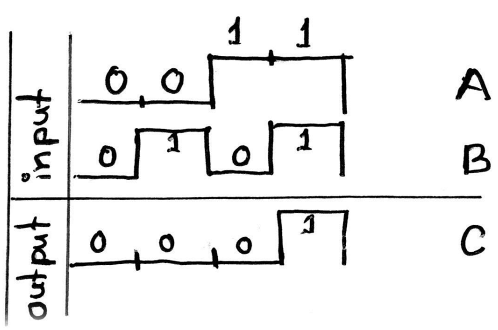
$1 \times 0 = 0$

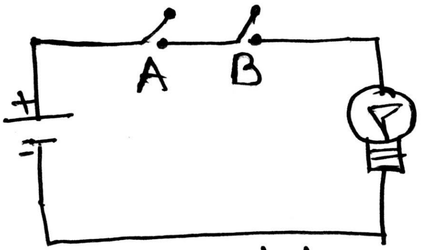
$1 \times 1 = 1$

গুনের ক্ষেত্রে input একটি মাত্র শূন্য হলে output শূন্য হবে। গুনে শূন্য শক্তিশালী।

AND গেটের সত্যক সারনী

ডিজিটাল সিগনালের মাধ্যমে আমরা AND gate বুঝতে পারি।





AND গেইটের সুইচিং সার্কিট

→ তারে যদি সমস্যা থাকে বা তারে যদি কাটা থাকে তবে বিদ্যুৎ চলবে না।

→ AND gate এর সুইচিং সার্কিট শ্রেণী/Series circuit হয়।

AND gate এর গল্প (ছেলেরাই সবসময় ভুল করে):

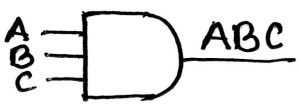
জনাব, আক্কাস মিয়া একজন বোকাসোকা মানুষ। আক্কাস মিয়ার বউএর নাম বিলকিস বেগম। বিলকিস বেগম একজন রাগী মহিলা। আক্কাস (A) এবং বিলকিস (B) এর দাম্পত্য জীবন এরূপ~

আক্কাস (A)	বিলকিস (B)	দাম্পত্য জীবন
ভুল (0)	ভুল (0)	আক্কাস মিয়ার দোষ
ভুল (0)	সঠিক (1)	আক্কাস মিয়ার দোষ
সঠিক (1)	ভুল (0)	আক্কাস মিয়ার দোষ
সঠিক (1)	সঠিক (1)	বিলকিস বেগম সঠিক

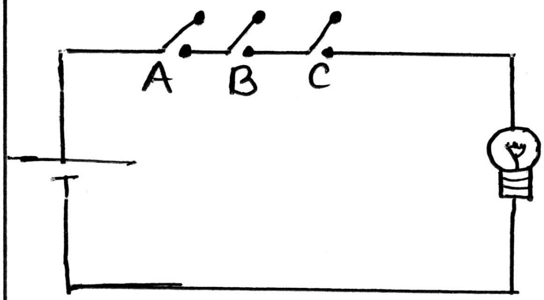
মেয়েরা ভুল করলেও ছেলের দোষ. কারন মেয়েরা কোনোদিন ভুল করে না

// switching circuit এ চাবি খোলা (0) এর মানে হল সেখানে সংযোগ বিচ্ছিন্ন বা তারের অনুপস্থিতি।

তিন ইনপুট (A, B ও C) বিশিষ্ট AND গেইটঃ



AND গেইটের লজিক চিহ্ন



AND গেইটের সুইচিং সার্কিট

Input			Output
A	B	C	ABC
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

AND গেইটের সত্যক সারনী

2. OR gate:

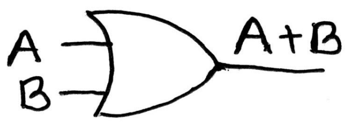
→ যোগের কাজ করে।

→ যৌক্তিক যোগের নিয়ম মেনে চলে

→ ন্যূনতম দুটি ইনপুট এবং সর্বোচ্চ অসীম সংখ্যক ইনপুট নেয়।

দুটি ইনপুট (A ও B) বিশিষ্ট OR গেটঃ

a. OR gate এ input এ A, B দেয়া হয়েছে। যেহেতু, এটা OR gate সেহেতু output যোগ অবস্থায় পাওয়া যাবে A+B.



OR gate এর লজিক চিহ্ন

আমরা জানি,
ডিজিটাল ডিভাইসে 0,1 হচ্ছে প্রতীক। কোনো কিছুর সাথে 1 যোগ করলে যোগফল সর্বদা 1 হয়। ডিজিটাল ডিভাইসে 0, 1 এর বাইরে কিছু হয় না। যেমনঃ

হাম্বু + গাম্বু + 1 = 1

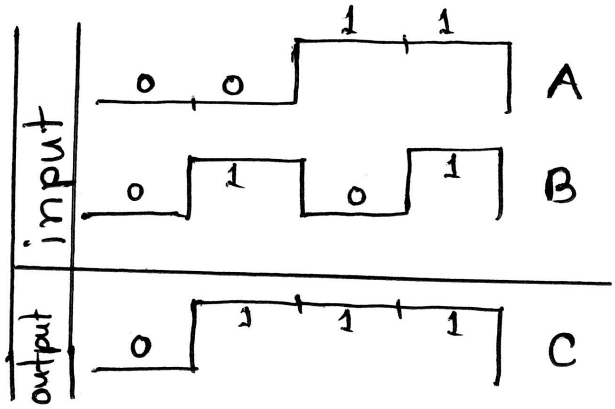
সত্যক সারনীর মাধ্যমে বুঝতে পারবো।

input		output	
A	B	A+B	
0	0	0	0+0=0
0	1	1	0+1=1
1	0	1	1+0=1
1	1	1	1+1=1

OR গেইটের সত্যক সারনী

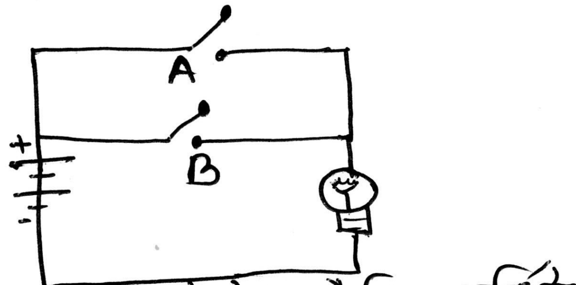
যোগের ক্ষেত্রে input একটিমাত্র 1 হলে output 1 হবে। যোগে 1 শক্তিশালী।

ডিজিটাল সিগনালের মাধ্যমে আমরা OR gate বুঝতে পারি।



Switching Circuit:

→OR gate এর সুইচিং সার্কিট সমান্তরাল/parallel circuit হয়।



OR গেইটের সুইচিং সার্কিট

OR gate এর গল্প (মেয়েরাই সবসময় সঠিক):

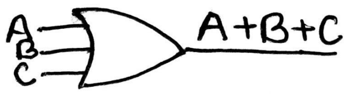
জনাব, আক্কাস মিয়ার বউ বিলকিস বেগম সবসময় বাড়ির সব সিদ্ধান্ত নেয়। বেচারা আক্কাস (A) আর রাগী বিলকিস (B) এর দাম্পত্য জীবন এরূপ~

আক্কাস (A)	বিলকিস (B)	দাম্পত্য জীবন
ভুল (0)	ভুল (0)	আক্কাস মিয়ার দোষ
ভুল (0)	সঠিক (1)	বিলকিস বেগম সঠিক
সঠিক (1)	ভুল (0)	বিলকিস বেগম সঠিক
সঠিক (1)	সঠিক (1)	বিলকিস বেগম সঠিক

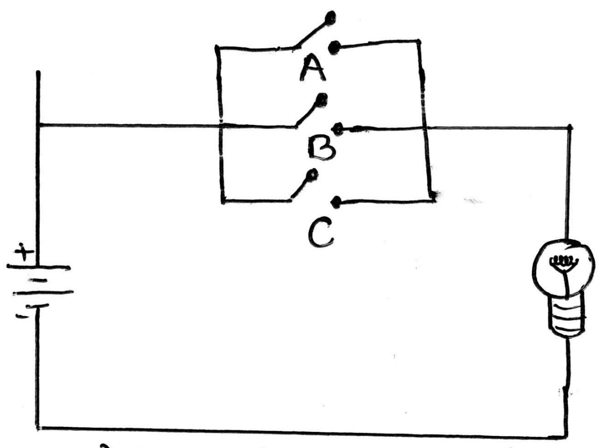
ছেলেরা ভুলগুলোকে মেনে নেয় তখন সংসার ঠিক থাকে।

সংসার সুখের হয় রমনের গুনে

তিন ইনপুট (A, B ও C) বিশিষ্ট OR গেইটঃ



OR গেইটের লজিক চিহ্ন



OR গেইটের সুইচিং সার্কিট

OR গেইটের সত্যক সারণী

Input			output
A	B	C	A+B+C
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

3. NOT gate:

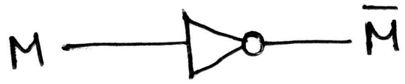
→ পূরকের কাজ করে। অর্থাৎ, $\bar{0}=1$, $\bar{1}=0$ করে।

→ যৌক্তিক পূরকের কাজ করে। NOT gate এর অপর নাম ইনভার্টার।

→ কেবলমাত্র (সর্বোচ্চ বা সর্বনিম্ন) 1 টি মাত্র ইনপুট নেয়।

M ইনপুট বিশিষ্ট NOT গেইট:

a. NOT gate এ input এ M দেয়া হয়েছে। যেহেতু, এটা NOT gate সেহেতু output এ $\bar{M}$ পূরক অবস্থায় পাওয়া যাবে।



আমরা জানি,
ডিজিটাল ডিভাইসে 0,1 এর বাইরে আর চিহ্ন সম্ভব না। 0 হতে 1 বা 1 হতে 0 যাওয়া যায় শুধুমাত্র পূরকের মাধ্যমে।

$$\bar{0}=1,\ \bar{1}=0,\ \bar{\bar{0}}=\bar{1}=0,\ \bar{\bar{1}}=\bar{0}=1$$

কোনো মানকে দুইবার পূরক করালে মান আগের অবস্থায় আসে।

M	$\bar{M}$
0	1
1	0

NOT গেইটের সত্যক সারণী

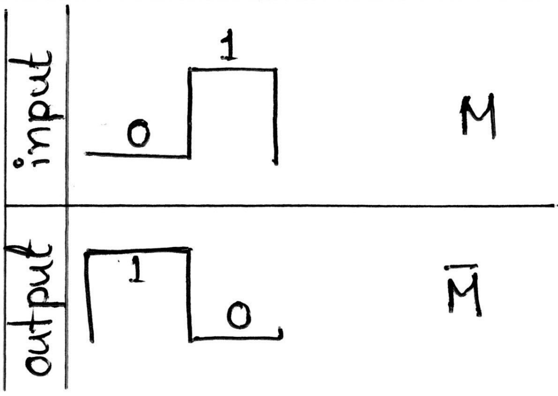
$\bar{1}=0$

$\bar{0}=1$

* জোড় সংখ্যক NOT থাকলে মানের পরিবর্তন হবে না।

* বিজোড় সংখ্যক NOT থাকলে মানের পরিবর্তন হবে।

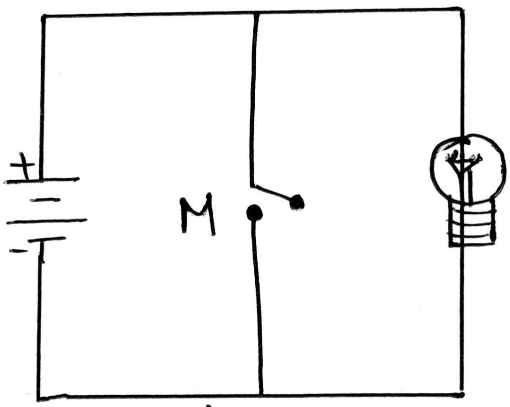
ডিজিটাল সিগন্যালের মাধ্যমে আমরা NOT gate বুঝতে পারি।



Switching Circuit:

→ NOT gate এর সুইচিং সার্কিট সংক্ষিপ্ত বর্তনী (short circuit) হয়।

→ NOT gate এ চাবি (Key) বর্তনীর ভিতরে থাকে।



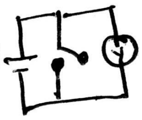
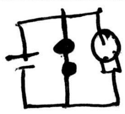
NOT গেইটের সুইচিং সার্কিট

* input এর যৌক্তিক বিপরীত output হয় NOT গেইটে।

NOT gate এর গল্প (অত্যাচারী মেয়ের গল্প):

বিজ্ঞানী Albert Einstein এর প্রথম বৌয়ের নাম Mileva MariC এবং দ্বিতীয় বৌয়ের নাম Elsa. বিজ্ঞানী Einstein কে MariC (MC) প্রচুর মারত। এতবড় বিজ্ঞানী হওয়ার দরুন লোক লজ্জার ভয়ে Einstein কাউকে কিছু বলত না। আশেপাশের মানুষ তাকে জিজ্ঞাস করত Eimstein

এর বাসা থেকে তার কষ্টের কান্না শুনতে পারার কারন কি?
Einstein আর সহ্য করতে না পেরে সে MariC (MC) কে তালাক দিয়ে Elsa (E) কে বিয়ে করে।
Elsa ছিল অনেক নির্দয় মহিলা। সে Einstein কে মারত। মারার পরে Einstein চিৎকার করে কাঁদলে তার গলা চেপে ধরত। গলা চেপে ধরলে Einstein এর মুখ থেকে আর শব্দ বের হত না। তখন, পাশের বাসার কেউ আর শুনতে পেত না।
Einstein খেয়াল করে দেখল Elsa তাকে MariC (MC) এর তুলনায় দ্বিগুন কষ্ট দেয়। MariC অনেক ভালো ছিল। তখন, Einstein আবিষ্কার করল, $\text{Elsa} = (\text{MariC})^2$ বা $E = MC^2$
এই সূত্র আবিষ্কারের জন্য Einstein নোবেল পুরস্কার পায়।
Einstein ও Elsa এর দাম্পত্য জীবন এরূপঃ

Einstein এর গলার অবস্থা	চিৎকার?
গলা খোলা (0) 	Einstein এর চিৎকার (1)
গলা বন্ধ (1) 	Einstein চুপ (0)

NOT gate বিদ্যুতের গলা চাপ দিয়ে ধরে। NOT gate active হলে বিদ্যুৎ প্রবাহ বন্ধ হয়। NOT gate off হলে বিদ্যুৎ প্রবাহ চালু হয়।

ভুল থেকে যদি নোবেল পুরস্কার পায়, তাহলে কী নোবেল পুরস্কার ভালো!

বুলিয়ান অ্যালজেবরা:

গানিতবিদ জর্জ বুল গনিত ও যুক্তির মধ্যে সম্পর্ক আবিষ্কার করে এর উপরে ভিত্তি করে যে অ্যালজেবরা তৈরী করেন।

মৌলিক অপারেশন ৩টি:

AND Operation/যৌক্তিক গুন	OR operation/যৌক্তিক যোগ
$0 \cdot 0 = 0$	$0+0=0$
$0 \cdot 1 = 0$	$0+1=1$
$1 \cdot 0 = 0$	$1+0=1$
$1 \cdot 1 = 1$	$1+1=1$

NOT operation/যৌক্তিক পূরক: $\bar{0}=1, \bar{1}=0$

বুলিয়ান ধ্রুবক:

বুলিয়ান অ্যালজেবরায় 0/1 থাকলে তাকে বুলিয়ান ধ্রুবক বলে।

বুলিয়ান দ্বৈতনীতি:

a. 0 এর পরিবর্তে 1/1 এর পরিবর্তে 0 হবে।

b. + থাকলে · হবে/ · থাকলে + হবে।

বুলিয়ান অ্যালজেবরায় উপপাদ্যঃ

উপপাদ্য/সূত্র/শক্রকে চিনে রাখার টেকনিকঃ

আমরা জানি, 0 বা 1 এর বাইরে কোনো চিহ্ন ব্যবহৃত হবে না। সেজন্য আমরা চলক (A) এর স্থলে একবার 0 আরেকবার 1 বসাবে।

a. যদি output দুইবারই 0 বা 1 অর্থাৎ একই হয় তাহলে সে মান বসাবো।

b. যদি output একবার 0 একবার 1 হয় অর্থাৎ ভিন্ন হয় তাহলে চলক (A) লিখবো।

$A+1=1$	$A+0=A$	$A\cdot 1=A$	$A\cdot 0=0$
$0+1=1$ $1+1=1$	$0+0=0$ $1+0=1$	$0\cdot 1=0$ $1\cdot 1=1$	$0\cdot 0=0$ $1\cdot 0=0$
$A+A=A$	$A\cdot A=A$	$A+\bar{A}=1$	$A\cdot\bar{A}=0$
$0+0=0$ $1+1=1$	$0\cdot 0=0$ $1\cdot 1=1$	$0+1=1$ $1+0=1$	$0\cdot 1=0$ $1\cdot 0=0$
$A=\bar{\bar{A}}=A$			
$0=\bar{\bar{0}}=0$ $1=\bar{\bar{1}}=1$	$\bar{0}=1$ $\bar{1}=0$	$\bar{\bar{0}}=\bar{1}=0$ $\bar{\bar{1}}=\bar{0}=1$	$\bar{\bar{\bar{\bar{0}}}}=0$ $\bar{\bar{\bar{1}}}=0$, $\bar{\bar{\bar{\bar{1}}}}=1$

ডি মরগ্যান উপপাদ্যঃ

গনিতবিদ বুলিয়ান ফাংশানকে মারার জন্য একটা জ্ঞান দিয়েছেন। যা ডি মরগ্যান উপপাদ্য। মরার পরেও মরগ্যান

ব্যাটার জ্ঞান বিতরন শেষ হয় না। এজন্যই বলা হল, জ্ঞান অমূল্য সম্পদ।

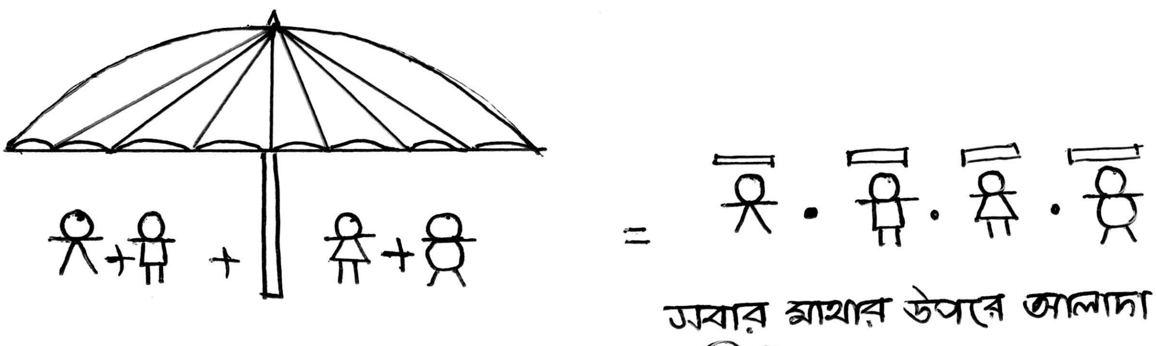
<u>প্রথম উপপাদ্যঃ</u> অনেক চলক যোগ দিয়ে যুক্ত আছে এবং তাদের উপরে একটিমাত্র NOT আছে ($\overline{A+B+C}$) তাহলে প্রত্যেকের উপরে আলাদা আলাদা NOT হবে এবং তারা গুন অবস্থায় থাকবে ($\bar{A}\cdot\bar{B}\cdot\bar{C}$)

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

<u>দ্বিতীয় উপপাদ্যঃ</u> অনেক চলক গুন দিয়ে যুক্ত আছে এবং তাদের উপরে একটিমাত্র NOT আছে ($\overline{A\cdot B\cdot C}$) তাহলে প্রত্যেকের উপরে আলাদা আলাদা NOT হবে এবং তারা যোগ অবস্থায় থাকবে ($\bar{A}+\bar{B}+\bar{C}$)

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

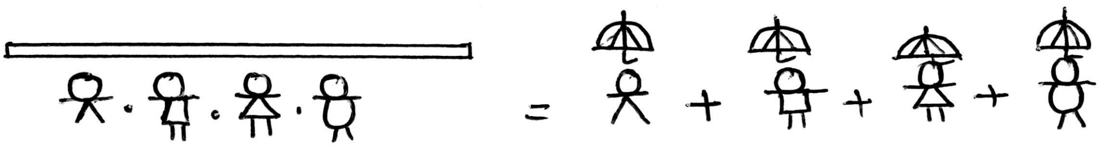
<u>প্রথম উপপাদ্যঃ</u>
শীতকালে বৃষ্টি হচ্ছে। সবাই একটি বড় ছাতার নিচে আছে। সবাইকে ছাতার নিচ থেকে বের হবার সময়ে একটি পলিথিন দেয়া হল। সবাই তখন নিজনিজ পলিথিন মাথায় ধরবে এবং কাছাকাছি থাকবে যেন শীত কম লাগে।



সবার মাথার উপরে আলাদা আলাদা পলিথিন

দ্বিতীয় উপপাদ্যঃ

শীতকালে বৃষ্টি হচ্ছে। সবাই একটি বড় লম্বা ছাতার নিচে আছে। সবাইকে বড় লম্বা পলিথিনের নিচ থেকে বের হবার সময়ে একটি একটি ছাতা দেয়া হল। সবাই তখন নিজ নিজ ছাতা মাথায় ধরবে এবং ছাতার পানি যেন কারো শরীরে না পড়ে সেজন্য দূরে হয়ে থাকবে।



সবার মাথার উপরে আলাদা আলাদা ছাতা

সত্যক সারনীঃ

সারনীর মাধ্যমে কোনো একটা সমীকরনের চলকের বিভিন্ন মানের জন্য বিভিন্ন আউটপুট প্রদর্শিত হয়।

ইনপুট এর সংখ্যা n হলে সত্যক সারনীতে মানের কম্বিনেশান হবে 2^n.

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

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

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

A	B	$\bar{A}$	$\bar{B}$	AB	$\overline{AB}$	$\bar{A}+\bar{B}$
0	0	1	1	0	1	1
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	1	0	0

ইনপুট/চলকের সংখ্যা 2 টি (A, B) সেজন্য সত্যক সারনীতে $2^2=4$ টা কম্বিনেশান।

A, B, C তিনটি চলকের জন্য মরগ্যানের উপপাদ্য

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

$\overline{ABC} = \bar{A} + \bar{B} + \bar{C}$

A	B	C	$\bar{A}$	$\bar{B}$	$\bar{C}$	$A+B+C$	$\overline{A+B+C}$	$\bar{A}\cdot\bar{B}\cdot\bar{C}$	ABC	$\overline{ABC}$	$\bar{A}+\bar{B}+\bar{C}$
0	0	0	1	1	1	0	1	1	0	1	1
0	0	1	1	1	0	1	0	0	0	1	1
0	1	0	1	0	1	1	0	0	0	1	1
0	1	1	1	0	0	1	0	0	0	1	1
1	0	0	0	1	1	1	0	0	0	1	1
1	0	1	0	1	0	1	0	0	0	1	1
1	1	0	0	0	1	1	0	0	0	1	1
1	1	1	0	0	0	1	0	0	1	0	0

ইনপুট/চলকের সংখ্যা 3টি (A, B, C) সেজন্য সত্যক সারনীতে $2^3 = 8$ টা কম্বিনেশান।

$(x+y)(x+z) = x+yz$

x	y	z	$x+y$	$x+z$	$(x+y)\cdot(x+z)$	yz	$x+yz$
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	1	1	1	1
1	0	0	1	1	1	0	1
1	0	1	1	1	1	0	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

সত্যক সারনীর মাধ্যমে সমীকরন প্রমান করা যায়।

চলক 3টা (x, y, z) সেজন্য $2^3 = 8$ টা কম্বিনেশান

সমীকরন সরল করার নিয়মঃ

বন্ধনীর ভিতরের কাজ আগে করতে হবে।

আগে NOT পরে AND তারপর OR এর কাজ করতে হবে। আকাশ থেকে বৃষ্টি পড়লে বৃষ্টি আগে ছাতাতে পড়ে। NOT গেট মাথার উপরে থাকে। সেজন্য সেটা ছাতার মতন।

$5\times3+7=15+7=22$ হয়।

কিন্তু $5\times3+7=5\times10=50$ হয় না।

বাস্তব জীবনে আমরা আগে গুনের (AND) কাজ করি। পরে যোগ (OR) এর কাজ করি। আগে যোগ ও পরে গুন করি না।

সরল কর: $(x+y)\cdot(x+z)$ V.V.I.

$= x\cdot x + xz + xy + yz$

$= x + xz + xy + yz$ $[\because A\cdot A = A]$

$= x(1+z+y) + yz$

$= x\cdot 1 + yz$ $[\because A+B+C+1 = 1]$

$= x + yz$ $[\because A\cdot 1 = A]$

একই ধরনের চলক দিয়ে কমন নিয়ে সরল সংক্ষিপ্ত করতে হয়।

সরল কর: $AB + \bar{A}B + A\bar{B}$

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

$= B\cdot 1 + A\bar{B}$ $[\because A+\bar{A}=1]$

$= B + A\bar{B}$ $[B\cdot 1 = B]$

$= (B+A)\cdot(B+\bar{B})$ $[\because x+yz=(x+y)\cdot(x+z)]$

$= (A+B)\cdot 1$ $[\because (B+\bar{B}=1)]$

$= A+B$ $[x\cdot 1 = x]$

সরল কর: $A(A+\bar{B}C) + A(\bar{B}+C)$

$= AA + A\bar{B}C + A\bar{B} + AC$

$= A + A\bar{B}C + A\bar{B} + AC$ $[\because A\cdot A = A]$

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

$= A\cdot 1$ $[\because x+y+z+1 = 1]$

$= A$ $[\because A\cdot 1 = A]$

সরল কর: $ABC + A\bar{B}C + AB\bar{C}$

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

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

$= A(C\cdot 1 + B\bar{C})$ $[\because B+\bar{B} = 1]$

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

$= A(B+C)$ $[\because C+B\bar{C} = (C+\bar{C})\cdot(C+B) = B+C]$

যদি $x+yz$ বা $x+\bar{x}z$ বা $c+b\bar{c}$ এরূপ থাকে তবে সেটা $(x+y)\cdot(x+z)$ বা $(x+\bar{x})(x+z)$ বা $(c+\bar{c})(c+b)$ হবে।

কোনো সমীকরনের উপরে NOT থাকলে ডিমরগ্যানের সূত্র ব্যবহৃত হয়।

সরল কর: $A\bar{B}+\overline{(\bar{A}+\bar{B}+C\cdot\bar{C})}$

$= A\bar{B}+\overline{(\bar{A}+\bar{B}+0)}$ $\quad [\because x\cdot\bar{x}=0]$

$= A\bar{B}+\overline{(\bar{A}+\bar{B})}$ $\quad [\because x+0=x]$

$= A\bar{B}+\bar{\bar{A}}\cdot\bar{\bar{B}}$ $\quad [\because \overline{x+y}=\bar{x}\cdot\bar{y}]$

$= A\bar{B}+AB$ $\quad [\because \bar{\bar{x}}=x]$

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

$= A\cdot 1$ $\quad [\because x+\bar{x}=1]$

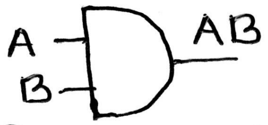
$= A$ $\quad [\because x\cdot 1=x]$

যৌগিক গেইট:
মৌলিক গেইটের সমন্বয়ে যৌগিক গেইট গঠিত হয়। দুই ধরনের~
1. সার্বজনীন (NAND, NOR) 2. বিশেষ গেট (XOR, XNOR)

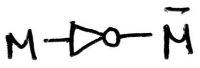
সার্বজনীন গেইট:

যে সকল গেইটের সাহায্যে মৌলিক গেইটসহ (AND, OR, NOT) যেকোনো গেইট এবং সার্কিট বাস্তবায়ন করা যায়, তাকে সার্বজনীন গেইট বলে।

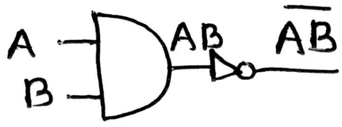
NAND gate: AND + NOT



মৌলিক AND gate



মৌলিক NOT gate



AND + NOT

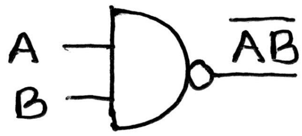
→ NAND গেইট গুন (AND) করার পরে মানকে পূরক (NOT) করে।

→ NAND গেইট ন্যূনতম দুটি ইনপুট এবং সর্বোচ্চ অসীম সংখ্যক ইনপুট নেয়।

→ NAND এর লাস্ট অক্ষর D, এবং AND গেইট দিয়েই NAND গেইট বানানো হয়। NAND দেখতে ইংরেজি অক্ষর D এর মতন।

ইনপুট দুটি (A ও B) বিশিষ্ট NAND গেইট:

এখানে ইনপুট A, B দেখা হয়েছে। যেহেতু এটা NAND গেইট। সেজন্য, আউটপুটে গুন এর উপরে NOT অর্থাৎ $\overline{AB}$ পাওয়া যাবে।



NAND গেইটের লজিক চিহ্ন

AND গেইটের মানকে বিপরীত করলে NAND গেইট পাওয়া যায়।

NAND গেইট যৌগিক এবং সার্বজনীন গেইট।

NAND হলো Not AND.

সত্যক সারনীর মাধ্যমে বুঝতে পারব।

input		রাশি	output
A	B	AB	$\overline{AB}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

NAND গেইটের সত্যক সারনী

$\overline{0\times0}=\bar{0}=1$

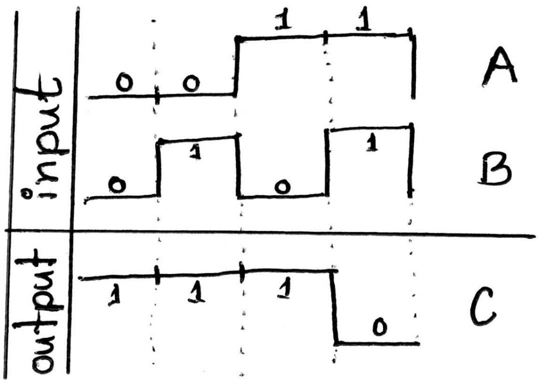
$\overline{0\times1}=\bar{0}=1$

$\overline{1\times0}=\bar{0}=1$

$\overline{1\times1}=\bar{1}=0$

AND gate এ input এ সবগুলো 1 হলে তখন output এ 1 হয়। সুতরাং, NAND gate এ input এ সবগুলো 1 হলে তখন শূন্য পাওয়া যায়।

ডিজিটাল সিগন্যালের মাধ্যমে আমরা NAND gate বুঝতে পারি

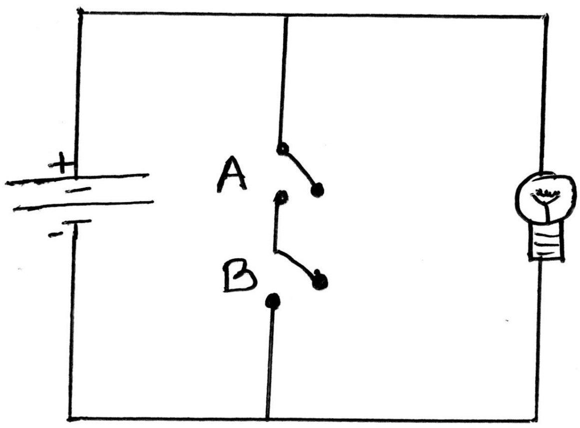


● NAND গেইট AND ও NOT গেইটের সমন্বয়ে গঠিত। সেজন্য NAND gate এর switching circuit এ AND ও NOT গেইটের উভয় বৈশিষ্ট্যই বিদ্যমান।

→ NAND গেইটে NOT থাকার জন্য সে সংক্ষিপ্ত বর্তনী (short circuit) করবে।

→ NAND গেইটে চাবি (Key) বর্তনীর ভিতরে থাকে।

→ NAND গেইটের বর্তনীর ভিতরে চাবি শ্রেণী/series circuit হয়। কারন, AND gate এ শ্রেণী সমবায়।



NAND গেইটের switching circuit

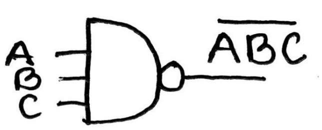
NAND gate এর গল্প (Einstein এর গলা Elsa চেপে ধরে):

কারো গলা কেউ চেপে ধরলে তার হাতের সব আঙ্গুলের ছাপ পড়ে। Elsa Einstein এর গলা চেপে ধরে। তখন বিজ্ঞানী Einstein চুপ হয়ে যায়। শুধুমাত্র একটা আংগুল দিয়ে গলা চেপে ধরা যায় না। সবগুলো আংগুল দিয়ে গলা চেপে ধরতে হয়। সবগুলো আংগুল বলতে হাতের যেকোনো আংগুল এবং বৃদ্ধ অংগুল সহকারেই Elsa বিজ্ঞানীর গলা চেপে ধরে।

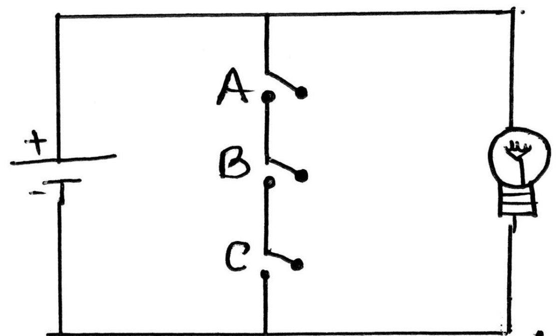
যেকোনো আংগুল	বৃদ্ধ আংগুল	Einstein এর অবস্থা
খোলা (0)	খোলা (0)	Einstein এর গলা খোলা (1)
খোলা (0)	চেপে ধরা (1)	শুধুমাত্র এক পাশের আংগুল দিয়ে গলা চাপা যায় না। চিৎকার (1)
চেপে ধরা (1)	খোলা (0)	শুধুমাত্র এক পাশের আংগুল দিয়ে গলা চাপা যায় না। চিৎকার (1)
চেপে ধরা (1)	চেপে ধরা (1)	দুই পাশের আংগুল দিয়ে গলা চেপে ধরে চিৎকার বন্ধ (0)

AND ও NAND গেইট একা কিছু করতে পারে না। সে অন্য মানের উপরে নির্ভরশীল।

তিন ইনপুট (A, B, C) বিশিষ্ট NAND গেইট:



NAND gate এর লজিক চিহ্ন

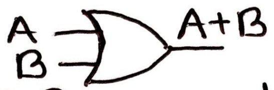


NAND gate এর switching circuit

input			রাফ	output
A	B	C	ABC	$\overline{ABC}$
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

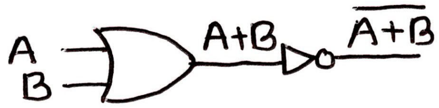
NAND গেইটের সত্যক সারনী

NOR gate: OR + NOT



মৌলিক OR gate

মৌলিক NOT gate



OR+NOT

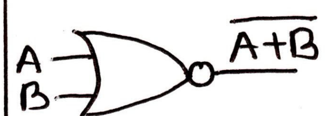
→ NOR গেইট যোগ (OR) করার পরে মানকে পূরক (NOT) করে।

→ NOR গেইট ন্যূনতম দুটি ইনপুট এবং সর্বোচ্চ অসীম সংখ্যক ইনপুট নেয়।

দুটি ইনপুট (A ও B) বিশিষ্ট NOR গেইটঃ

এখানে ইনপুটে A, B দেয়া হয়েছে। যেহেতু এটা NOR গেইট। সেজন্য, আউটপুটে যোগ এর উপরে NOT অর্থাৎ $\overline{A+B}$ পাওয়া যাবে।

OR গেইটের মানের বিপরীত NOR গেইট



NOR গেইটের লজিক চিহ্ন

NOR গেইট যৌগিক এবং সার্বজনীন গেইট।

NOR হলো Not OR.

সত্যক সারনীর মাধ্যমে বুঝতে পারবো।

input		রাশি	output
A	B	A+B	$\overline{A+B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

NOR গেইটের সত্যক সারনী

$\overline{0+0} = \overline{0} = 1$

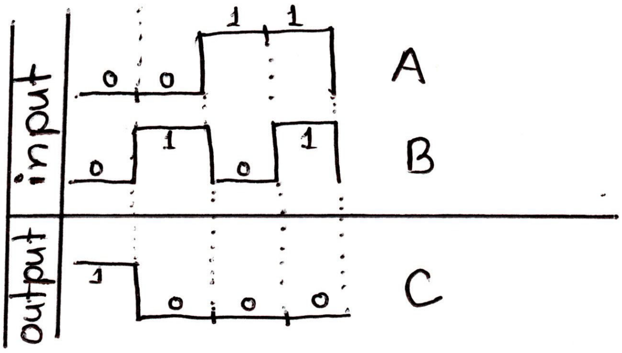
$\overline{0+1} = \overline{1} = 0$

$\overline{1+0} = \overline{1} = 0$

$\overline{1+1} = \overline{1} = 0$

OR gate এ input এ সবগুলো 0 হলে তখন output এ 0 হয়। সুতরাং, NOR gate এ input এ সবগুলো 0 হলে তখন output এ 1 পাওয়া যায়

ডিজিটাল সিগন্যালের মাধ্যমে আমরা NOR gate বুঝতে পারি।

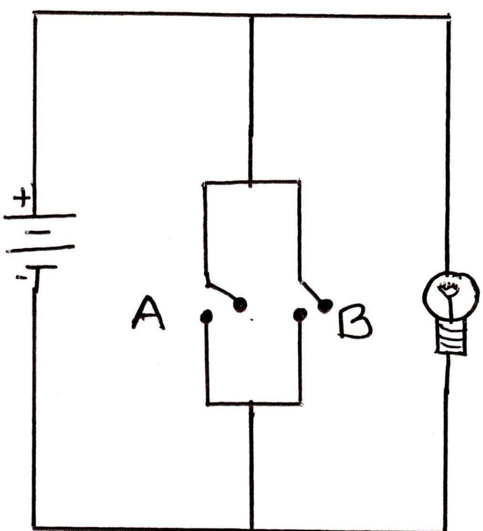


NOR গেইট OR ও NOT গেইটের সমন্বয়ে গঠিত। সেজন্য NOR gate এর switching circuit এ OR ও NOT গেইটের উভয় বৈশিষ্ট্যই বিদ্যমান।

→ NOR গেইটে NOT থাকার জন্য সে সংক্ষিপ্ত বর্তনী (short circuit) করবে।

→ NOR গেইটে চাবি (key) বর্তনীর ভিতরে থাকে।

→ NOR গেইটের বর্তনীর ভিতরে চাবি সমান্তরাল/parallel circuit হয়। কারন, OR gate এ সমান্তরাল সমবায়।



NOR গেইটের switching circuit

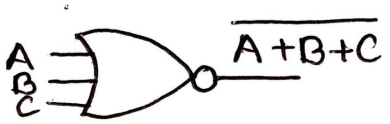
NOR gate এর গল্প (অসহায় Einstein):

একজনের গলা যে কেউ চেপে ধরলে হয়। একজন চাইলে নিজের গলা নিজেই চেপে ধরতে পারে। কেউ নিজে না চেপে ধরলে অন্য কেউও চেপে ধরতে পারে। গলা চেপে ধরলে চিৎকার বন্ধ হয়ে যায়। Elsa/Einstein যে কেউ Einstein এর গলা চেপে ধরতে পারে। কিন্তু কেউই যদি গলা চেপে না ধরে তবে

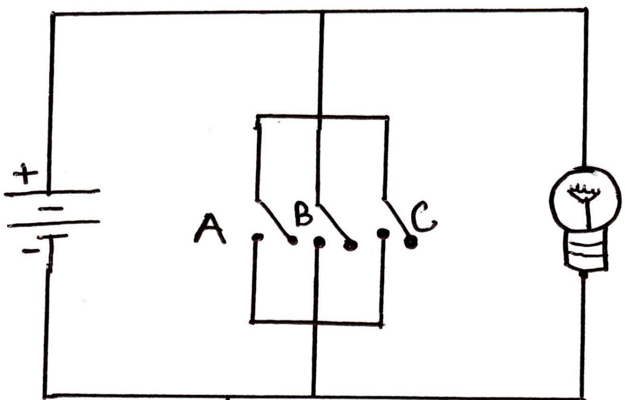
Elsa	Einstein	Einstein এর অবস্থা
হাত খোলা (0)	হাত খোলা (0)	চিৎকার (1)
হাত খোলা (0)	নিজ গলা চেপে ধরা (1)	চুপ (0)
Einstein এর গলা চেপে ধরা (1)	হাত খোলা (0)	চুপ (0)
Einstein এর গলা চেপে ধরা (1)	নিজ গলা চেপে ধরা (1)	চুপ (0)

OR ও NOR ভাইটি একাই একশ। কারো সাহায্য লাগে না।

তিন ইনপুট (A, B, C) বিশিষ্ট NOR গেইট:



NOR gate এর লজিক চিহ্ন



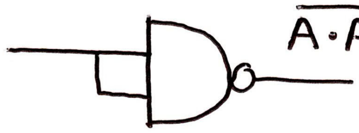
NOR gate এর switching circuit

input			রাশি	output
A	B	C	A+B+C	$\overline{A+B+C}$
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

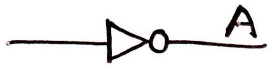
NOR গেইটের সত্যক সারনী

NAND ও NOR গেইট দিয়ে মৌলিক গেইট সহ সকল গেইটসমূহ বাস্তবায়ন করা যায়। আমরা, NAND ও NOR গেইট দিয়ে মৌলিক গেইট বাস্তবায়ন দেখবো।

1. শুধুমাত্র NAND গেইট দিয়ে NOT গেইট বাস্তবায়নঃ

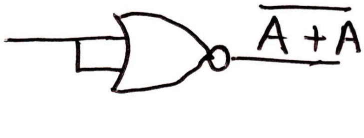
A  $\overline{A\cdot A} = \bar{A}$

[$\because A\cdot A = A$]

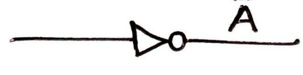
A  $\bar{A}$

মৌলিক NOT গেইট

1. শুধুমাত্র NOR গেইট দিয়ে NOT গেইট বাস্তবায়নঃ

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

$A+A = A$

A  $\bar{A}$

মৌলিক NOT গেইট

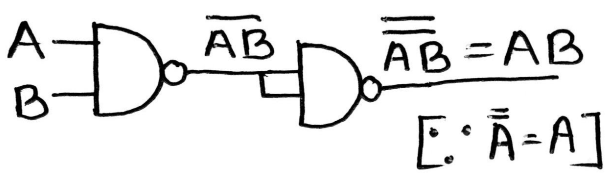
NOT গেইট শুধুমাত্র 1 টি এবং একই ধরনের ইনপুট নেয়। NOT গেইট A ইনপুট নেয় এবং আউটপুটে $\overline{A}$ দেয়।

NAND/NOR এ আমরা দুটি ইনপুট সংযুক্ত করে একটি ইনপুট বানাই। সেজন্য আমরা NAND/NOR কে NOT এর অবস্থায় পাই। দুটি ইনপুট সংযুক্ত NAND/NOR এ আমরা A ইনপুট দিলে আউটপুটে $\overline{A}$ পাই। যা আমরা NOT গেইটে পেতাম।

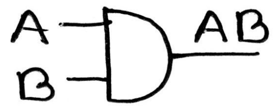
NAND/NOR গেইটের সকল ইনপুট একই হলে গেইটটি মৌলিক গেইট NOT গেইট হিসেবে কাজ করে।

2. শুধুমাত্র NAND গেইট দিয়ে AND গেইট বাস্তবায়নঃ

AB
$= \overline{\overline{AB}}$



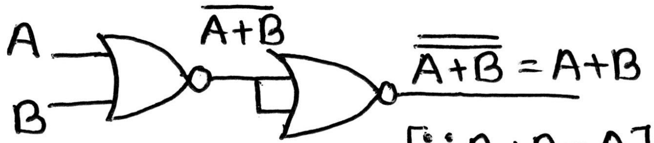
[∵ $\overline{\overline{A}} = A$]



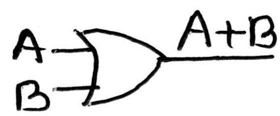
মৌলিক AND গেইট

2. শুধুমাত্র NOR গেইট দিয়ে OR গেইট বাস্তবায়নঃ

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



[∵ $A+A=A$]



মৌলিক OR গেইট

Ninja Technique: কোনো সমীকরন NAND/NOR গেইট দিয়ে বাস্তবায়ন করতে হলে। সমীকরন/মানের উপরে দুটি NOT দিতে হবে।

1. গুনের (AND) এর উপরে NOT থাকলে NAND গেইট দিয়ে এবং যোগের (OR) এর উপরে NOT থাকলে NOR গেইট দিয়ে সমাধান করতে হবে।

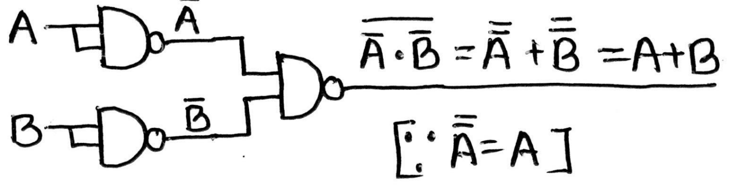
2. গুনের উপরে/যোগের উপরে NOT না থেকে যদি একক চলক (A/B/x) এর উপরে NOT থাকে। কিংবা NAND/NOR এর উপরে NOT থাকে। তাহলে NAND/NOR এর দুটি ইনপুট সংযুক্ত করে NOT গেইট বানাতে হয়।

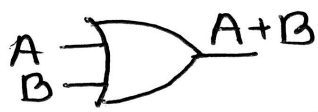
3. শুধুমাত্র NAND গেইট দিয়ে OR গেইট বাস্তবায়ন:

$A+B$

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

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





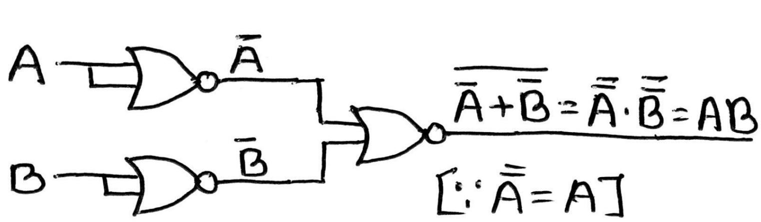
মৌলিক OR গেইট

3. শুধুমাত্র NOR গেইট দিয়ে AND গেইট বাস্তবায়ন:

AB

$= \overline{\overline{AB}}$

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



মৌলিক AND গেইট

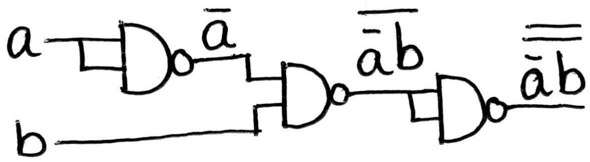
NAND এবং NOR / NAND থেকে OR / NOR থেকে AND এর রূপান্তর শুধুমাত্র De Morgan's Law এর মাধ্যমে।

> গেইট/লজিক গেইট/সার্কিট একই অর্থ বহন করে

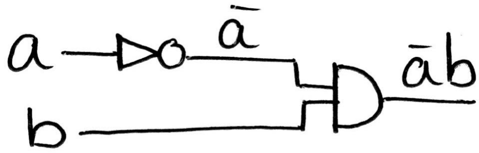
মান/সমীকরণকে মৌলিক গেইট/NAND ও NOR দিয়ে বাস্তবায়নঃ

● $\bar{a}b$ NAND গেইট দিয়ে

$\bar{a}b$
$= \overline{\overline{\bar{a}b}}$

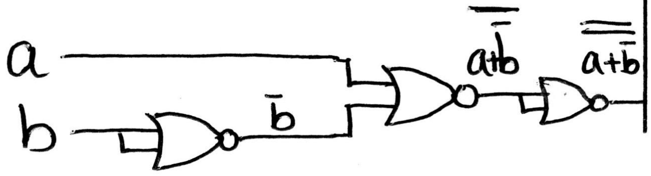


$\bar{a}b$ মৌলিক গেইট

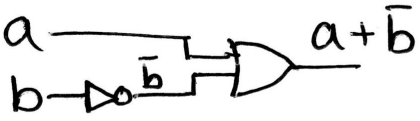


● $a+\bar{b}$ NOR গেইট দিয়ে

$a+\bar{b}$
$= \overline{\overline{a+\bar{b}}}$

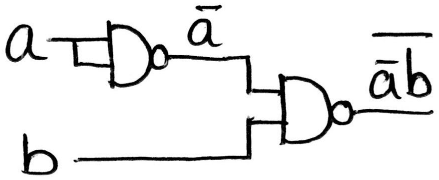


$a+\bar{b}$ মৌলিক গেইট

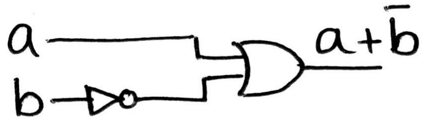


● $a+\bar{b}$ NAND গেইট দিয়ে

$a+\bar{b}$
$= \overline{\overline{a+\bar{b}}}$
$= \overline{\bar{a}\cdot\bar{\bar{b}}}$
$= \overline{\bar{a}b}$

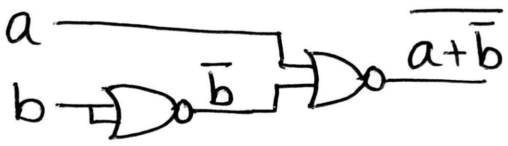


$a+\bar{b}$ মৌলিক গেইট

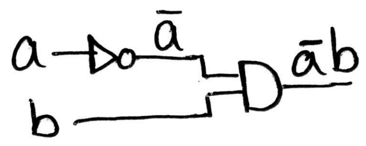


$\bar{a}b$ NOR গেইট দিয়ে

$$\bar{a}b$$
$$=\overline{\overline{\bar{a}b}}$$
$$=\overline{\bar{\bar{a}}+\bar{b}}$$
$$=\overline{a+\bar{b}}$$

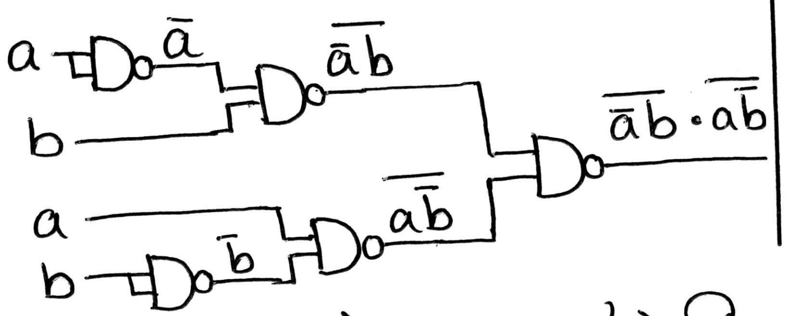


$\bar{a}b$ মৌলিক গেইট

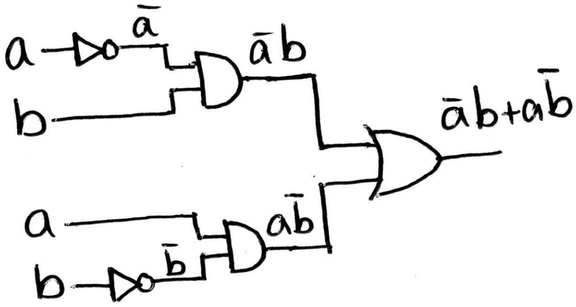


$\bar{a}b+a\bar{b}$ NAND গেইট দিয়ে।

$$\overline{\overline{\bar{a}b+a\bar{b}}}$$
$$=\overline{\overline{\bar{a}b+a\bar{b}}}$$
$$=\overline{\overline{\bar{a}b}\cdot\overline{a\bar{b}}}$$

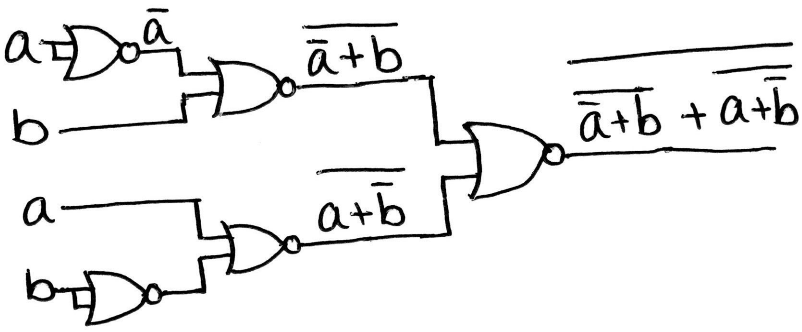


$\bar{a}b+a\bar{b}$ মৌলিক গেইট

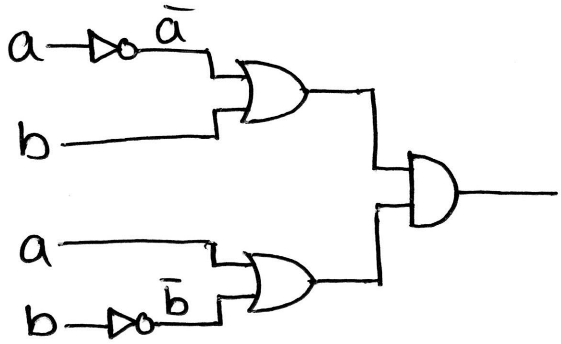


$(\bar{a}+b)\cdot(a+\bar{b})$ NOR গেইট দিয়ে

$$\overline{\overline{(\bar{a}+b)\cdot(a+\bar{b})}}$$
$$=\overline{\overline{(\bar{a}+b)\cdot(a+\bar{b})}}$$
$$=\overline{\overline{(\bar{a}+b)}+\overline{(a+\bar{b})}}$$



$(\bar{a}+b)\cdot(a+\bar{b})$ মৌলিক গেইট দিয়ে



আমরা, বিগত সমীকরণ হতে দেখতে পাই যে,

1. NAND/NOR গেইট দিয়ে বাস্তবায়ন করার সময়ে সমীকরণে/মান এর উপরে দুটি NOT দেয়া হয়েছে।

2. এক চলক (A/B) কে NOT করার জন্য NAND/NOR এর ইনপুটকে সংযুক্ত করে NOT বানানো হয়েছে। $\bar{A}/\bar{B}$

3. গুন (AND)/ যোগ (OR) এর উপরে NOT থাকার জন্য যেসব গেইটকে NAND/NOR দিয়ে বাস্তবায়ন করা হয়েছে। $\overline{A+B}/\overline{AB}/\bar{A}+\bar{B}/\bar{A}\cdot\bar{B}$

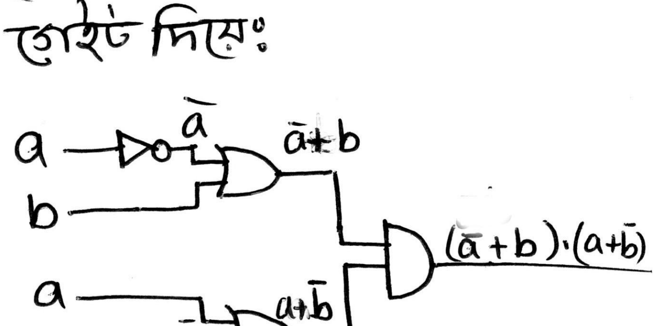
4. NAND/NOR এর উপরে NOT থাকলে ইনপুটকে সংযুক্ত করে NOT বানানো হয়েছে। $\overline{\overline{A+B}}/\overline{\overline{AB}}$

5. গুন (AND)/ যোগ (OR) রুপান্তর করার জন্য De Morgan's Law ব্যবহার করা হয়েছে।

ⓔ $(\bar{a}+b)\cdot(a+\bar{b})$ NAND দিয়ে

নিয়মঃ যদি কোনো সমীকরণে একাধিক অংশ গুন আকারে থাকে। অংশগুলোর ভিতরে যোগ (OR) থাকে। তবে অংশগুলোর উপরে আলাদা আলাদাভাবে NOT হবে। এবং সম্পূর্ন সমীকরনের উপরে আবার দুটি NOT হবে।

$(\bar{a}+b)\cdot(a+\bar{b})$ মৌলিক গেইট দিয়েঃ



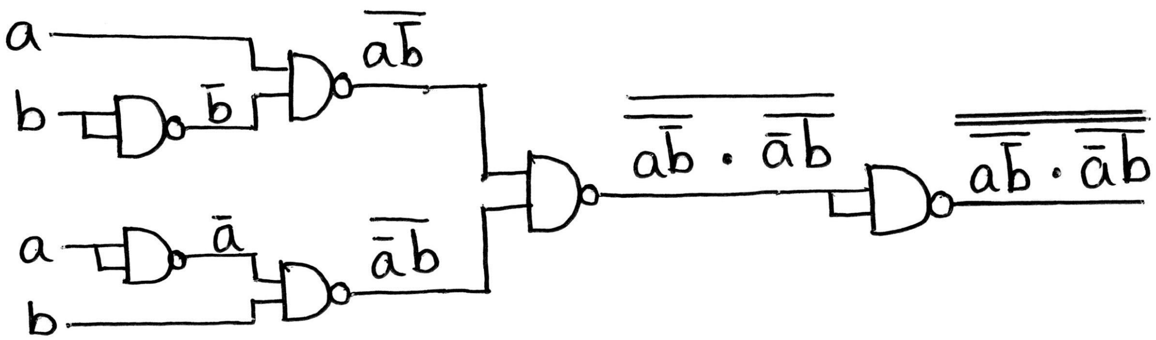
$(\bar{a}+b)\cdot(a+\bar{b})$ [অংশগুলো গুনের সাথে আছে]

$= \overline{\overline{(\bar{a}+b)}}\cdot\overline{\overline{(a+\bar{b})}}$ [অংশের ভিতরে যোগ আছে। সেজন্য দুটি NOT]

$= \overline{\bar{\bar{a}}\cdot\bar{b}}\cdot\overline{\bar{a}\cdot\bar{\bar{b}}}$ [মরগ্যানের মাধ্যমে যোগ হতে গুন]

$= \overline{a\cdot\bar{b}}\cdot\overline{\bar{a}\cdot b}$

$= \overline{\overline{\overline{a\bar{b}}\cdot\overline{\bar{a}\cdot b}}}$ [দুটি NOT]



Note: বিশুদ্ধ NAND গেইটে/সমীকরনে কোনো যোগ (OR) চিহ্ন থাকে না। দ্বিতীয়ত, গুনের উপরে NOT থাকে।

NOR: $\bar{a}b+a\bar{b}$ NOR গেইট দিয়ে।

নিয়ম: যদি কোনো সমীকরনে একাধিক অংশ যোগ আকারে যুক্ত থাকে। অংশগুলোর ভিতরে গুন (AND) থাকে। তাহলে, অংশগুলোর উপরে আলাদা আলাদাভাবে দুটি NOT দিতে হবে। এবং সম্পূর্ন সমীকরনের উপরে পরে আবার দুটি NOT হবে।

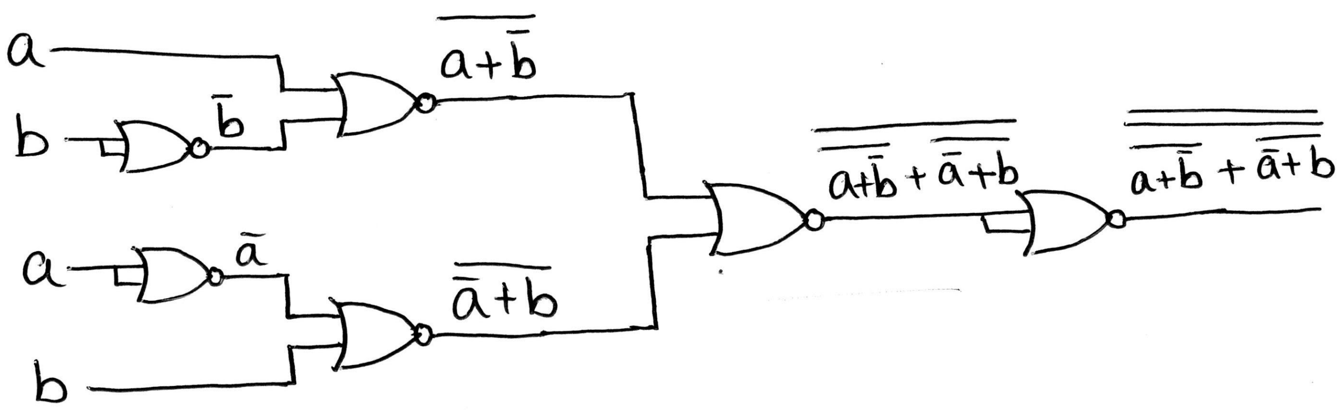
$\bar{a}b + a\bar{b}$ [অংশগুলোর মাঝে যোগ আছে]

$= \overline{\overline{\bar{a}b}} + \overline{\overline{a\bar{b}}}$ [অংশের ভিতরে গুন আছে। সেজন্য দুটি NOT]

$= \overline{\bar{\bar{a}} + \bar{b}} + \overline{\bar{a} + \bar{\bar{b}}}$ [মরগ্যানের মাধ্যমে গুন হতে যোগ]

$= \overline{a + \bar{b}} + \overline{\bar{a} + b}$

$= \overline{\overline{\overline{a + \bar{b}} + \overline{\bar{a} + b}}}$ [দুটি NOT]



Note: বিশুদ্ধ NOR গেইটে/সমীকরণে কোনো গুন (AND) চিহ্ন থাকে না।
দ্বিতীয়ত, যোগের উপরে NOT থাকে।

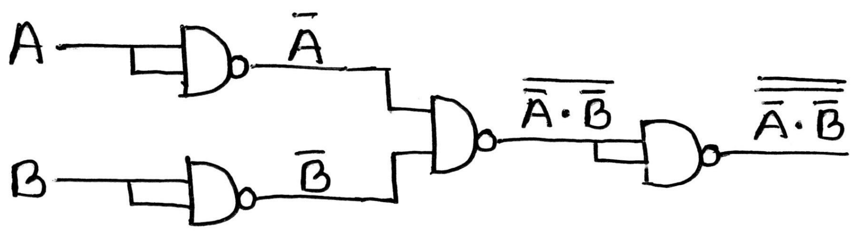
// $\bar{a}$/$\bar{b}$ /এর উপরে NOT এর জন্য input সংযুক্ত করে NAND/NOR দিয়ে NOT বানানো হয়েছে।
$\overline{a+b}$/$\overline{ab}$ এদের যোগ (OR)/গুন (AND) এর উপরে NOT থাকার জন্য এদের বাস্তবায়ন করতে NOR/NAND গেইট লাগে।

● NAND গেইট ব্যবহার করে NOR গেইট বাস্তবায়নঃ

$\overline{A+B}$ [NOR এর মান]

$= \overline{A} \cdot \overline{B}$ [ডিমরগ্যানের মাধ্যমে যোগ হতে গুন]

$= \overline{\overline{\overline{A} \cdot \overline{B}}}$ [গুন থেকে NAND এ যাবার জন্য ২টি NOT]

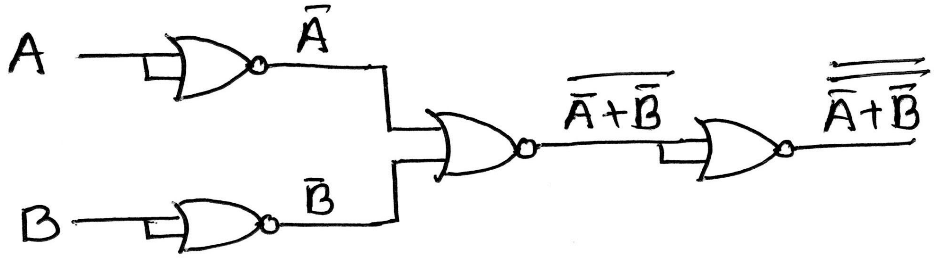


● NOR গেইট ব্যবহার করে NAND গেইট বাস্তবায়নঃ

$\overline{AB}$ [NAND এর মান]

$= \overline{A} + \overline{B}$ [ডিমরগ্যানের মাধ্যমে গুন হতে যোগ]

$= \overline{\overline{\overline{A} + \overline{B}}}$ [যোগ থেকে NOR এ যাবার জন্য 2টি NOT]



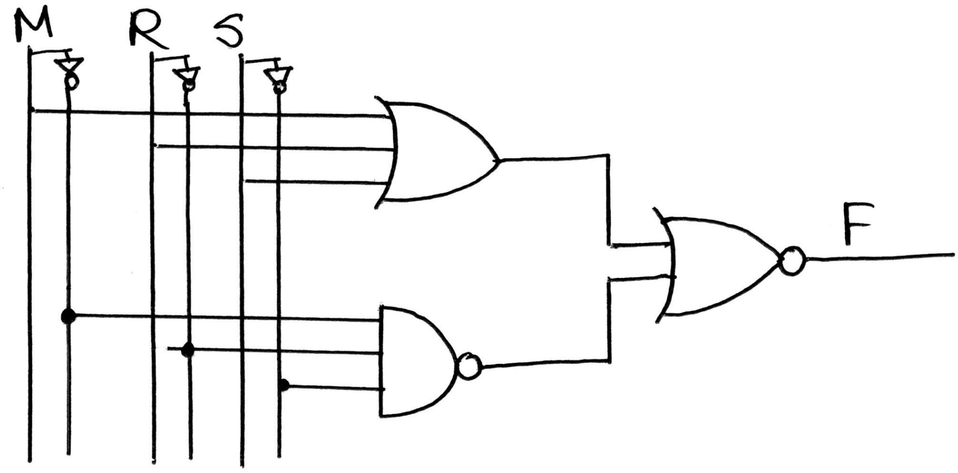
লজিক গেইট হতে সমীকরন বের করে সরল করাঃ

1. ইনপুট চলক খেয়াল করতে হবে। সেটা NOT নাকি NOT মুক্ত।

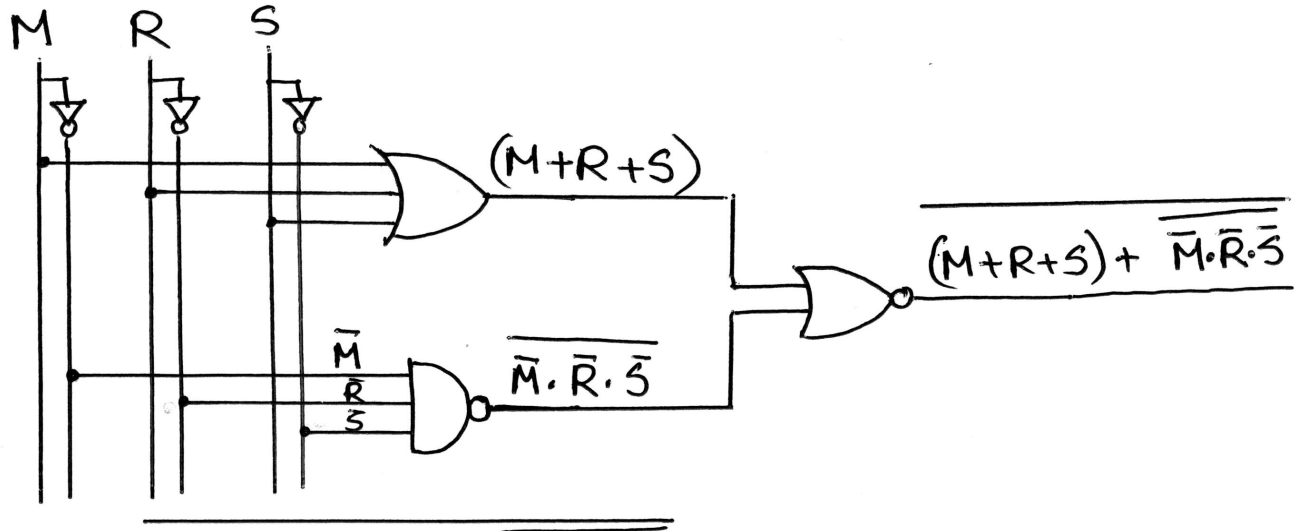
2. প্রতিটি লজিক গেইটের অপারেশান এর পরপর মান লিখতে হবে।

3. সমীকরনকে স্বাভাবিক নিয়মে সরল করতে হবে।

উদাঃ ১



সমাধানঃ



$\therefore F = \overline{(M+R+S) + \overline{\overline{M}\cdot\overline{R}\cdot\overline{S}}}$

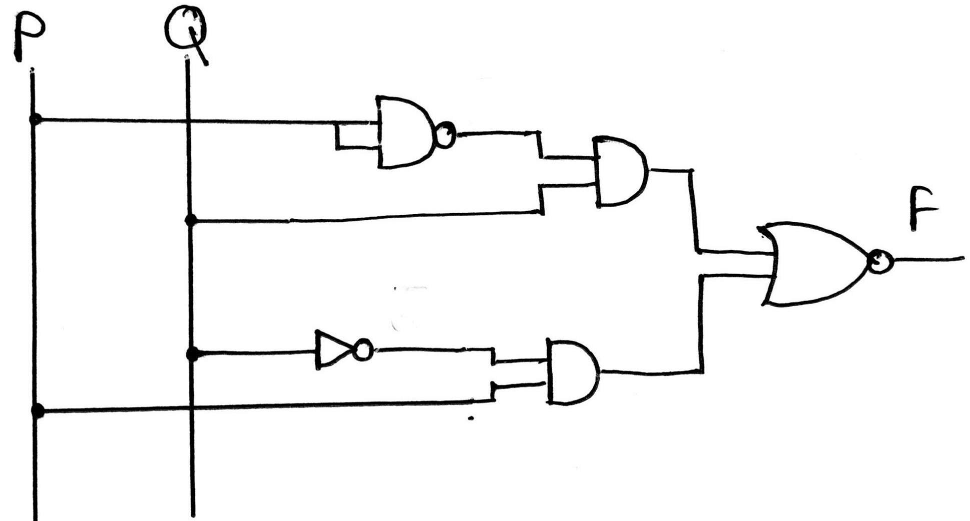
$\therefore F = \overline{M+R+S} \cdot \overline{(\overline{\overline{M}\cdot\overline{R}\cdot\overline{S}})}$ [মরগ্যান উপপাদ্য/সূত্র]

$= \overline{M}\cdot\overline{R}\cdot\overline{S}\cdot(\overline{M}\cdot\overline{R}\cdot\overline{S})$ $[\overline{\overline{A}} = A]$

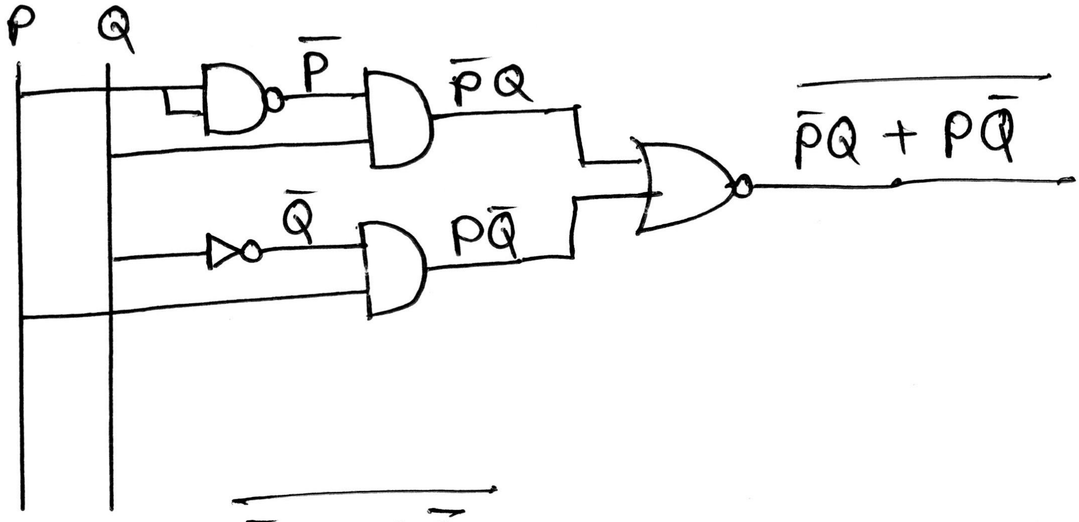
$= \overline{M}\cdot\overline{R}\cdot\overline{S}$ $[A\cdot A = A]$

$= \overline{M+R+S}$

উদা: 02



সমাধানঃ



$\therefore F = \overline{\overline{P}Q + P\overline{Q}}$

$= \overline{P \oplus Q}$ $[\because \overline{A}B + A\overline{B} = A \oplus B]$

বিশেষ গেইটঃ

X-OR ও XNOR গেইট দুটি বিশেষ গেইট হিসেবে পরিচিত। এরা যৌগিক গেইট।

NAND/NOR এরা দুই ধরনের মৌলিক গেইটের সমন্বয়ে গঠিত।

XOR/XNOR এরা তিন ধরনের মৌলিক গেইটের সমন্বয়ে গঠিত।

X-OR gate:

Exclusive OR গেইটকে সংক্ষেপে X-OR গেইট বলা হয়।

→ X-OR গেইট AND, OR, NOT তিনটি মৌলিক গেইটের সমন্বয়ে গঠিত।

→ X-OR গেইট ন্যূনতম দুটি ইনপুট এবং সর্বোচ্চ অসীম সংখ্যক ইনপুট নেয়।

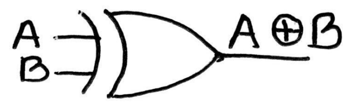
→ X-OR গেইটে OR যুক্ত থাকার জন্য এর সাথে OR এর চিত্রের এবং circuit এ মিল আছে।

দুটি ইনপুট (A ও B) বিশিষ্ট X-OR গেইটঃ

এখানে, ইনপুটে A, B দেয়া হয়েছে।

যেহেতু এটা XOR গেইট। সেজন্য, আউটপুটে ⊕ পাওয়া যাবে। অর্থাৎ, $A \oplus B$ পাওয়া যাবে।

$A \oplus B$ এর মান $\bar{A}B + A\bar{B}$



XOR গেইটের লজিক চিহ্ন

সত্যক সারণীর মাধ্যমে আমরা বুঝতে পারবো।

input		output
A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

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

$1 \cdot 0 + 0 \cdot 1 = 0$

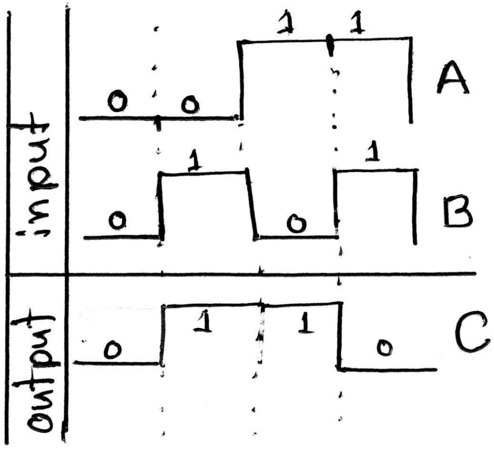
$1 \cdot 1 + 0 \cdot 0 = 1$

$0 \cdot 0 + 1 \cdot 1 = 1$

$0 \cdot 1 + 1 \cdot 0 = 0$

XOR গেইটে input এ বিজোড় সংখ্যক 1 হলে তখন output এ 1 হয়।

ডিজিটাল সিগন্যালের মাধ্যমে আমরা XOR gate বুঝতে পারি

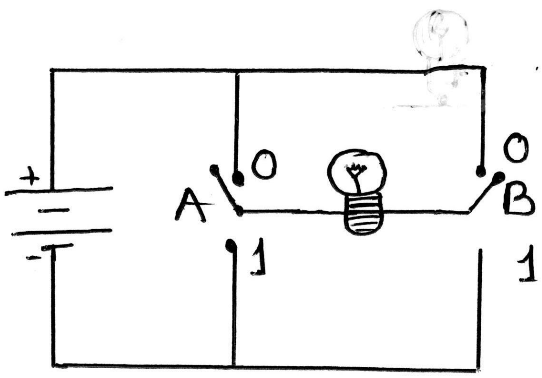


XOR গেইট বিশেষ ধরনের যৌগিক OR গেইট।

→ XOR গেইটে OR গেইট এর বৈশিষ্ট্য বিশেষভাবে বিদ্যমান

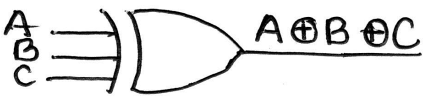
→ XOR গেইটে 0 অর্থাৎ চাবি খোলা থাকলে বিদ্যুৎ প্রবাহ না হয়ে লাইট জ্বলে না। জোড়া 1 থাকলে সর্ট সার্কিট হলে লাইট জ্বলে না।

→ XOR গেইটে সমান্তরাল/ parallel চাবি থাকে।
কারন, OR gate এ সমান্তরাল সমবায়।



XOR গেইটের switching circuit

তিন ইনপুট (A, B, C) বিশিষ্ট X-OR গেইট:



XOR গেইটের লজিক চিহ্ন

XOR গেইটে input এ বিজোড় সংখ্যক 1 হলে তখন output এ 1 হয়।

$A \oplus B \oplus C = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

input			output
A	B	C	$A \oplus B \oplus C$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

XOR গেইটের সত্যক সারণী

X-NOR gate:

Exclusive NOT OR গেইটকে সংক্ষেপে X-NOR গেইট বলা হয়।

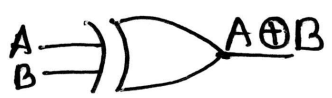
→ X-NOR গেইট AND, OR, NOT তিনটি মৌলিক গেইটের সমন্বয়ে গঠিত।

→ X-NOR গেইট ন্যূনতম দুটি ইনপুট এবং সর্বোচ্চ অসীম সংখ্যক ইনপুট নেয়।

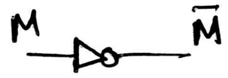
→ X-NOR গেইটে NOR যুক্ত থাকার জন্য এর সাথে OR গেইটের চিত্র এবং circuit এ মিল আছে।

→ NOR গেইটে যেমন OR এর বিপরীত তদ্রূপ X-NOR গেইট XOR গেইটের বিপরীত।

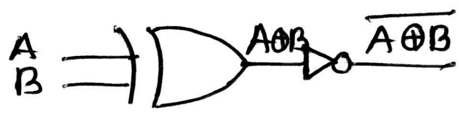
দুটি ইনপুট (A ও B) বিশিষ্ট X-NOR গেইটঃ



XOR গেইট



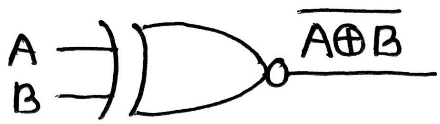
মৌলিক NOT গেইট



XOR + NOT

XOR গেইট এর মানকে পূরক (NOT) করার পরে XNOR পাওয়া যায়।

এখানে, ইনপুটে A, B দেয়া হয়েছে। যেহেতু, এটা XNOR গেইট। সেজন্য আউটপুট XOR এর উপরে NOT অর্থাৎ $\overline{A\oplus B}$ পাওয়া যাবে।



XNOR গেইটের লজিক চিহ্ন

$\overline{A\oplus B}$ এর মান $AB+\bar{A}\bar{B}$

সত্যক সারনীর মাধ্যমে বুঝতে পারবো

input		রাফ	output
A	B	$A\oplus B$	$\overline{A\oplus B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

$AB+\bar{A}\bar{B}$

$0\cdot0+1\cdot1=1$

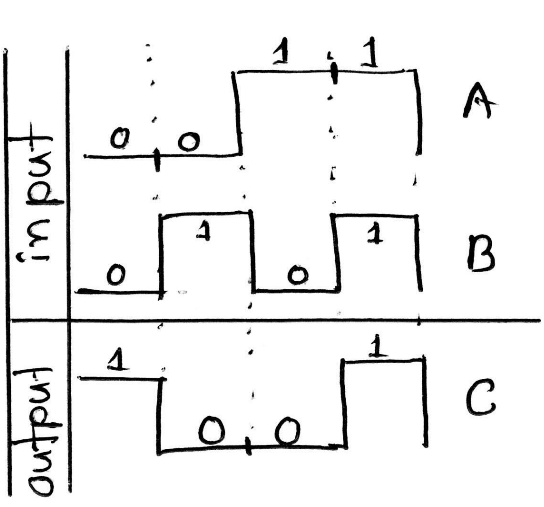
$0\cdot1+1\cdot0=0$

$1\cdot0+0\cdot1=0$

$1\cdot1+0\cdot0=1$

XOR গেইটের উল্টা মান

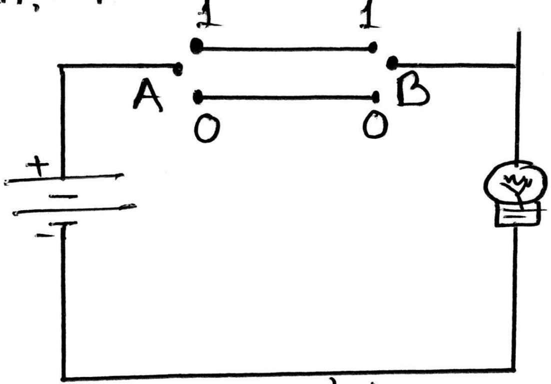
ডিজিটাল সিগন্যালের মাধ্যমে আমরা XNOR gate বুঝতে পারি।



XNOR গেইট বিশেষ ধরনের যৌগিক NOR গেইট।

→ XNOR গেইটে NOR গেইট এর বৈশিষ্ট্য বিশেষভাবে বিদ্যমান।

→ XNOR গেইটে সমান্তরাল/Parallel চাবি থাকে। কারণ, OR gate এ সমান্তরাল সমবায়।

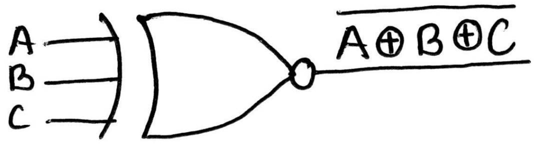


XNOR গেইটের switching circuit

তিন ইনপুট (A, B, C) বিশিষ্ট XNOR:

input			রাফ	output
A	B	C	$A\oplus B\oplus C$	$\overline{A\oplus B\oplus C}$
0	0	0	0	1
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	0
1	1	1	1	1

XNOR গেইটের সত্যক সারণী



XNOR গেইটের লজিক চিহ্ন

$\overline{A\oplus B}$

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

$= \left(\overline{\bar{A}B}\right)\cdot\left(\overline{A\bar{B}}\right)$

$= \left(\bar{\bar{A}} + \bar{B}\right)\cdot\left(\bar{A} + \bar{\bar{B}}\right)$

$= \left(A + \bar{B}\right)\cdot\left(\bar{A} + B\right)$

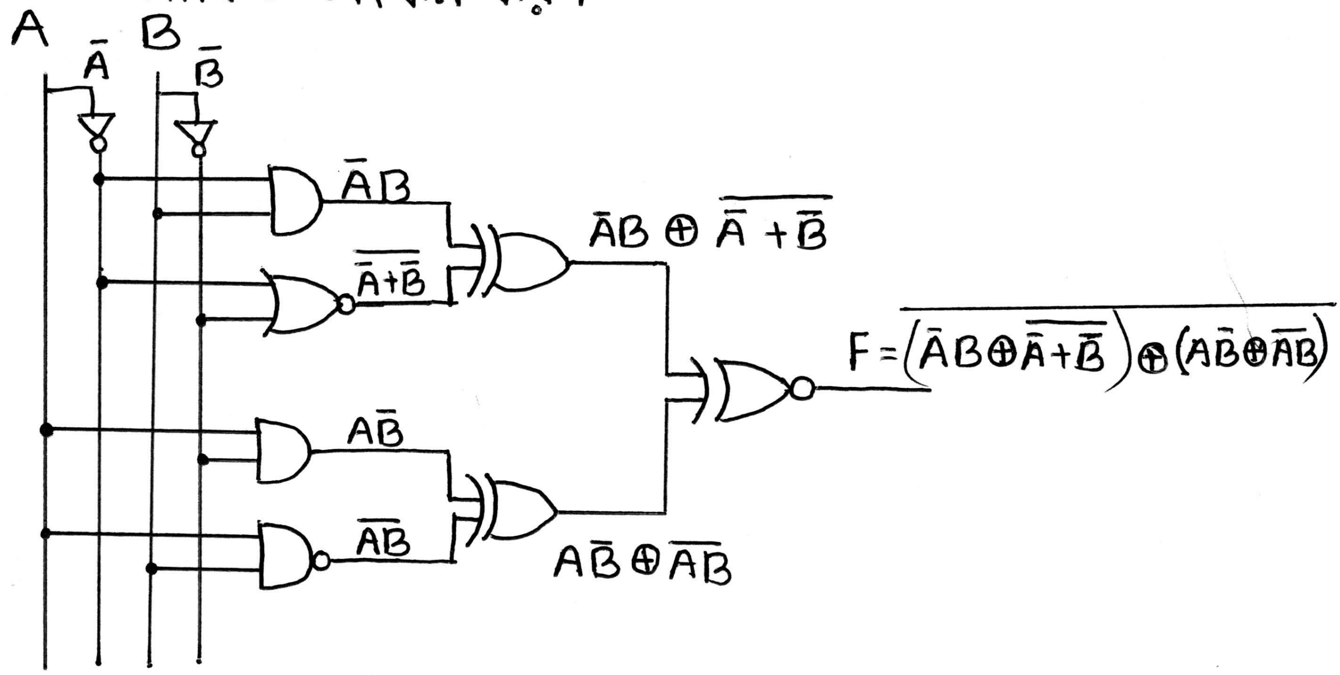
$= A\cdot\bar{A} + AB + \bar{A}\cdot\bar{B} + B\cdot\bar{B}$

$= 0 + AB + \bar{A}\bar{B} + 0$

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

$\therefore \overline{A\oplus B} = AB + \bar{A}\bar{B}$

$F = \overline{\left(\bar{A}B \oplus \overline{\bar{A}+\bar{B}}\right) \oplus \left(A\bar{B} \oplus \overline{AB}\right)}$ ফাংশনটির লজিক সার্কিট অংকন কর।



সত্যক সারনী থেকে সমীকরন নির্নয়ঃ (পরীক্ষায় আসবে না):

নিয়মঃ

১। আউটপুটে যাদের মান 1 শুধুমাত্র তাদের সারি বিবেচিত হবে।

২। ইনপুটে যাদের মান 0 হবে তাদের চলকের উপরে NOT হবে। যাদের মান 1 হবে তাদের চলকের উপরে কিছুই হবে না।

৩। ইনপুটের চলকগুলো গুন অবস্থায় থাকবে। যাদের আউটপুট 1. তারা সবাই একত্রে যোগ হবে।

AND

A	B	AB	
0	0	0	×
0	1	0	×
1	0	0	×
1	1	1	AB

সমীকরণঃ $0+0+0+AB$

$= AB$

OR

A	B	A+B	
0	0	0	0
0	1	1	$\bar{A}B$
1	0	1	$A\bar{B}$
1	1	1	AB

সমীকরণঃ $0+\bar{A}B+A\bar{B}+AB$

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

$= \bar{A}B + A$

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

$= A+B$

NAND

A	B	$\overline{AB}$	
0	0	1	$\bar{A}\bar{B}$
0	1	1	$\bar{A}B$
1	0	1	$A\bar{B}$
1	1	0	0

সমীকরণঃ $\bar{A}\bar{B}+\bar{A}B+A\bar{B}+0$

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

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

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

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

NOR

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

সমীকরণঃ $\bar{A}\bar{B}+0+0+0$

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

$= \overline{A+B}$

X-OR

A	B	$A \oplus B$	
0	0	0	0
0	1	1	$\bar{A}B$
1	0	1	$A\bar{B}$
1	1	0	0

সমীকরনঃ $0 + \bar{A}B + A\bar{B} + 0$

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

X-NOR

A	B	$\overline{A \oplus B}$	
0	0	1	$\bar{A}\bar{B}$
0	1	0	0
1	0	0	0
1	1	1	AB

সমীকরনঃ $\bar{A}\bar{B} + 0 + 0 + AB$

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

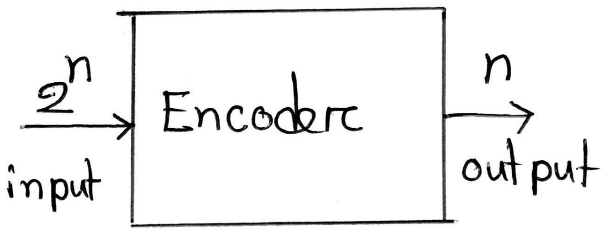
Encoder:

এনকোডার এক ধরনের সমবায় সার্কিট বা ডিজিটাল বর্তনী যা মানুষের ব্যবহৃত বিভিন্ন আলফানিউমেরিক বর্ন, বিশেষ চিহ্ন, টেক্সট, অডিও ও ভিডিও ইত্যাদিকে কম্পিউটার বা ডিজিটাল সিস্টেমের বোধগম্য কোডে রুপান্তর করে।

* মানুষের ভাষায় বর্নমালা বেশি সেজন্য এনকোডার অধিক input নেয়। কম্পিউটারের ভাষায় বর্নমালা কম সেজন্য এনকোডার অল্প output দেয়। এনকোডার 2^n সংখ্যক ইনপুট নেয়। এবং n সংখ্যক আউটপুট দেয়।

* এনকোডার যোগ করে। অর্থাৎ এনকোডারে OR gate ⇒ ব্যবহৃত হয়।

*এনকোডারে শুধুমাত্র একটা 1 হয় input এ. বাকি সব input এর মান 0 হয়।



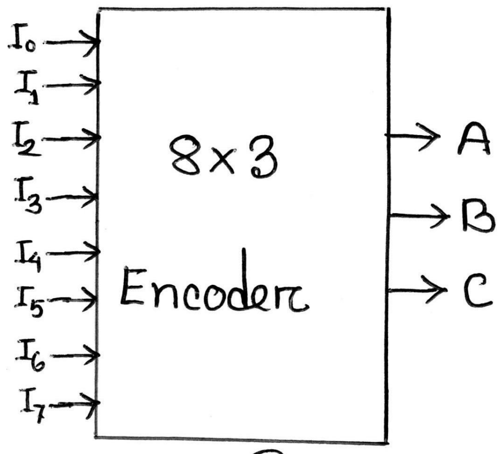
ব্লক চিত্র

input (2^n)	output (n)
$2^2/4$	2
$2^3/8$	3
$2^4/16$	4
$2^5/32$	5

8 (2^3) to 3 লাইন এনকোডার:

*এখানে 8 to 3 এনকোডারের আটটি ইনপুট $I_0, I_1, I_2, I_3, I_4, I_5, I_6, I_7$ এবং তিনটি আউটপুট A, B, C

* input এর প্রথম অক্ষর I সেজন্য I ব্যবহৃত হয়েছে।

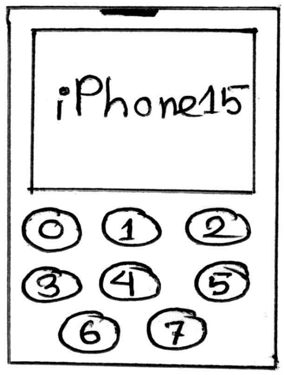


ব্লক চিত্র

8×3 এনকোডারের সত্যক সারণী:

Input								output		
I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	A	B	C
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

* এখানে, 0, 1, 2, 3, 4, 5, 6, 7 বাটনের একটা ফোন দেখতে পাচ্ছি। আমরা যদি 0 বাটনে চাপ দেই তখন স্ক্রীনে 0 দেখতে পাব। যদি 5(I_5) বাটনে চাপ দেই তখন স্ক্রীনে 5(101) দেখতে পাব।



* একটা বাটন শুধুমাত্র সেটা চাপ দিলে চালু হয়। তখন বাকিগুলো অফ থাকে।

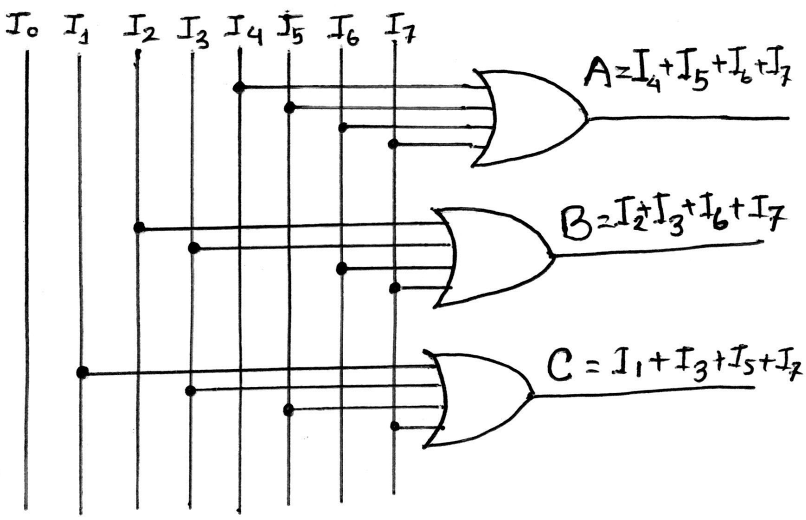
সত্যক সারনী থেকে প্রত্যেকটি আউটপুটের জন্য নিম্নোক্ত বুলিয়ান ফাংশান লিখা যায়:

$A = I_4 + I_5 + I_6 + I_7$

$B = I_2 + I_3 + I_6 + I_7$

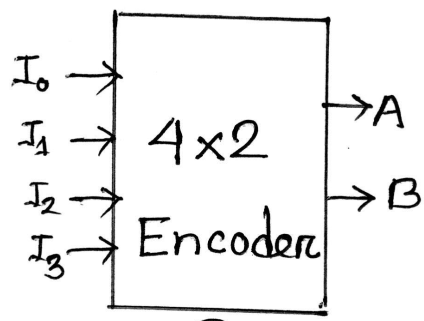
$C = I_1 + I_3 + I_5 + I_7$

8 to 3 লাইন এনকোডারের লজিক সার্কিট/সার্কিট/লজিক গেইট/গেইট: →



4 (2^2) to 2 লাইন এনকোডার:

* এখানে, 4 to 2 এনকোডারের চারটি ইনপুট I_0, I_1, I_2, I_3 এবং দুইটি আউটপুট A, B



ব্লক চিত্র

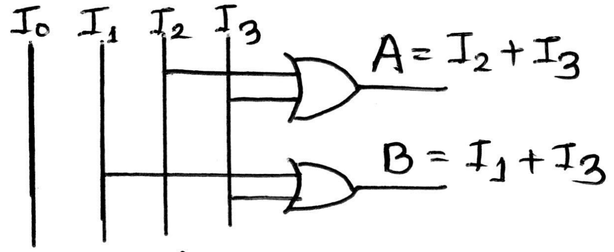
4×2 এনকোডারের সত্যক সারনী:

input				output	
I_0	I_1	I_2	I_3	A	B
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

সত্যক সারনী থেকে প্রত্যেকটি আউটপুটের জন্য নিম্নোক্ত বুলিয়ান ফাংশান লেখা যায়ঃ

$A = I_2 + I_3$

$B = I_1 + I_3$



4to2 লাইন এনকোডারের জন্য লজিক সার্কিট।

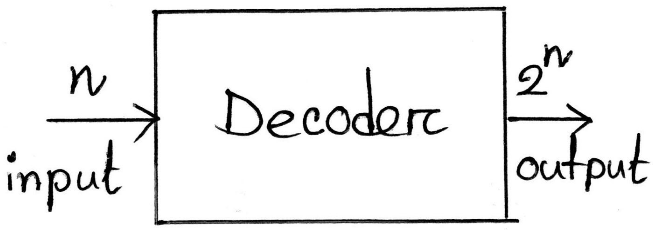
Decoder (ডিকোডার):

ডিকোডার এক ধরনের সমবায় সার্কিট বা ডিজিটাল বর্তনী যা কম্পিউটার বা ডিজিটাল সিস্টেমের বোধগম্য কোডকে মানুষের বোধগম্য ফরম্যাটে রুপান্তরিত করে।

* কম্পিউটারের ভাষায় বর্নমালা অনেক কম (0,1) সেজন্য ডিকোডার অল্প ইনপুট নেয়। মানুষের ভাষায় বর্নমালা অনেক বেশি। সেজন্য, ডিকোডার অধিক সংখ্যক আউটপুট দেয়। ডিকোডার n সংখ্যক ইনপুট নেয় এবং 2^n সংখ্যক আউটপুট দেয়।

* ডিকোডার যৌক্তিক গুন করে। অর্থাৎ ডিকোডারে AND gate ব্যবহৃত হয়

* ডিকোডারে আউটপুটে কেবলমাত্র একটি ঘরে 1 পাওয়া যায়।



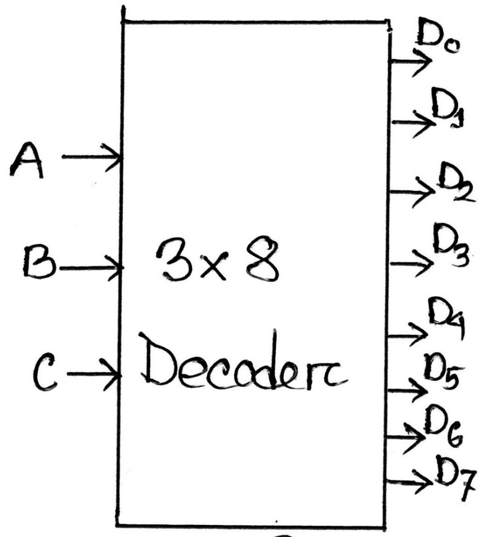
ব্লক চিত্র

input (n)	output (2^n)
2	4 (2^2)
3	8 (2^3)
4	16 (2^4)
5	32 (2^5)

3 to 8 (2^3) লাইন ডিকোডার:

* এখানে, 3 to 8 ডিকোডারের তিনটি A, B, C ইনপুট এবং আটটি আউটপুট $D_0, D_1, D_2, D_3, D_4, D_5, D_6, D_7$.

* output-এর প্রথম অক্ষর D (decoder) সেজন্য D ব্যবহৃত হয়েছে



ব্লক চিত্র

3×8 ডিকোডারের সত্যক সারণী:

input			output							
A	B	C	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

সত্যক সারণী হতে প্রাপ্ত প্রত্যেকটি আউটপুটের জন্য নিম্নোক্ত বুলিয়ান ফাংশান লিখা যায়।

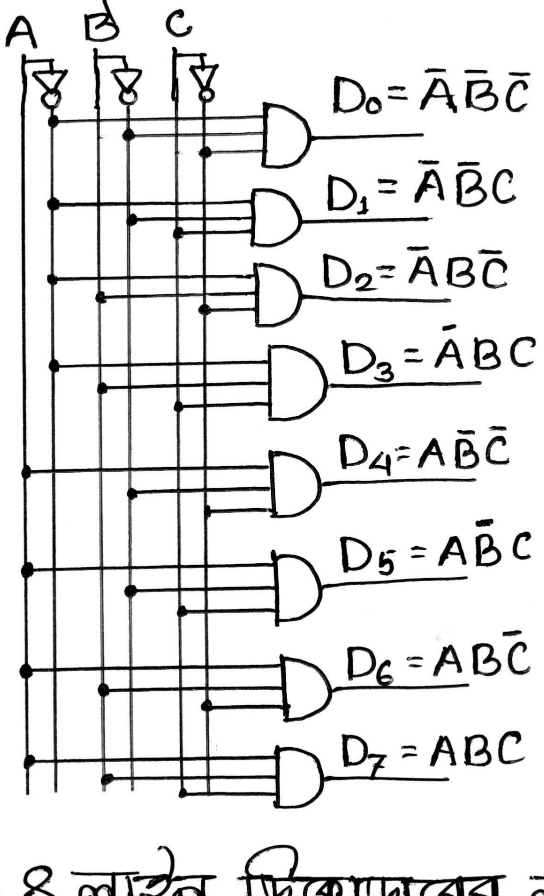
$D_0 = \bar{A}\bar{B}\bar{C}$ $\quad D_4 = A\bar{B}\bar{C}$

$D_1 = \bar{A}\bar{B}C$ $\quad D_5 = A\bar{B}C$

$D_2 = \bar{A}B\bar{C}$ $\quad D_6 = AB\bar{C}$

$D_3 = \bar{A}BC$ $\quad D_7 = ABC$

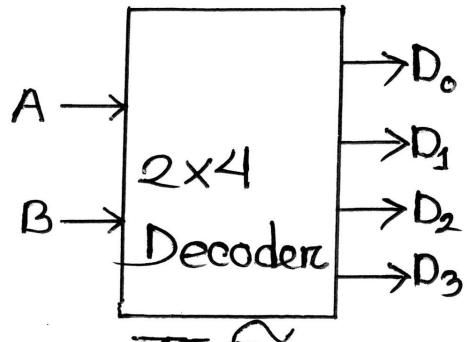
* আউটপুটে যেসব ঘর 1 ($D_0, D_1, D_2, \ldots D_7$) তাদের সাপেক্ষে input (ABC) লিখা হয়েছে।
* input এ মান শূন্য (0) হলে চলক (A, B, C) এর উপরে NOT দেয়া হয়েছে।
* Decoder গুন (AND) করে। যেজন্য A, B, C গুন অবস্থায় যুক্ত।



3 to 8 লাইন ডিকোডারের লজিক সার্কিট / সার্কিট / লজিক ডায়াগ্রাম / গেইট:

2 to 4 (2^2) লাইন ডিকোডার:

* এখানে 2 to 4 ডিকোডার দুইটি ইনপুট A, B এবং চারটি আউটপুট D_0, D_1, D_2, D_3.



ব্লক চিত্র

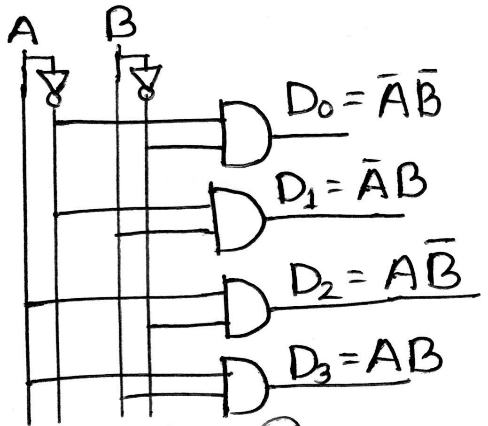
2×4 ডিকোডারের সত্যক সারনীঃ

input		output			
A	B	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

সত্যক সারনী থেকে প্রত্যেকটি আউটপুটের জন্য নিম্নোক্ত বুলিয়ান ফাংশন লিখা যায়ঃ

$D_0 = \bar{A}\bar{B}$ $\quad$ $D_2 = A\bar{B}$

$D_1 = \bar{A}B$ $\quad$ $D_3 = AB$



2 to 4 লাইন ডিকোডারের জন্য লজিক সার্কিট।

Adder Circuit:

যে সমবায় সার্কিট দ্বারা যোগের কাজ সম্পন্ন হয় তাকে অ্যাডার বা যোগের বর্তনী বলে।

কম্পিউটারের সকল গানিতিক কাজ বাইনারী যোগের মাধ্যমে সম্পন্ন হয়। অ্যাডার সার্কিট দুই ধরনেরঃ

Half Adder Circuit বা অর্ধ যোগের বর্তনী

Full Adder Circuit বা পূর্ণ যোগের বর্তনী

$$\begin{array}{r} 3 \\ +4 \\ \hline 07 \end{array}$$

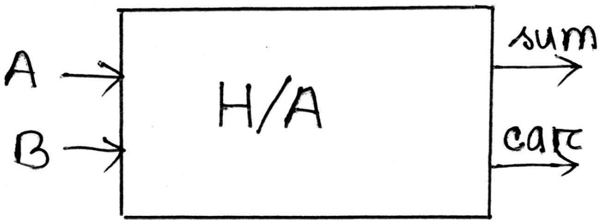
carry ← 0, 7 → summation

$$\begin{array}{r} 9 \\ +7 \\ \hline 16 \end{array}$$

carry ← 1, 6 → summation

হাফ অ্যাডার:

যে সমবায় সার্কিট দুটি বিট যোগ করে আউটপুটে একটি যোগফল (sum) ও একটি ক্যারি (carr) দেয় তাকে হাফ অ্যাডার সার্কিট বা অর্ধযোগের বর্তনী বলে।



হাফ অ্যাডারের ব্লক চিত্র

A	0	0	1	1
B	+0	+1	+0	+1
	00	01	01	10
	c ↑	c ↑	c ↑	c ↑
	S	S	S	S

ইনপুট A, B এবং আউটপুট যোগফল (sum) ও ক্যারি (carr):

input		output		
A	B	sum	carr	
0	0	0	0	
0	1	1	0	$\bar{A}B$
1	0	1	0	$A\bar{B}$
1	1	0	1	AB

সত্যক সারণী

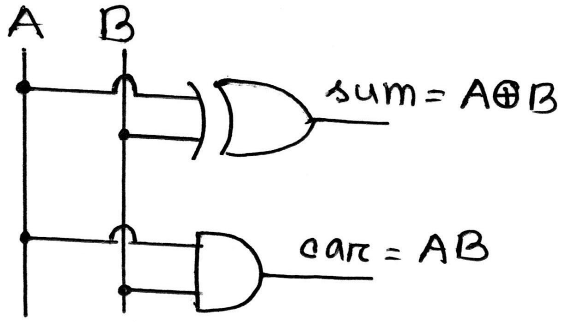
সত্যক সারণী থেকে আউটপুট (sum) ও (carr) এর জন্য নিম্নোক্ত বুলিয়ান ফাংশান লিখা যায়-

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

$$= A \oplus B \quad (XOR)$$

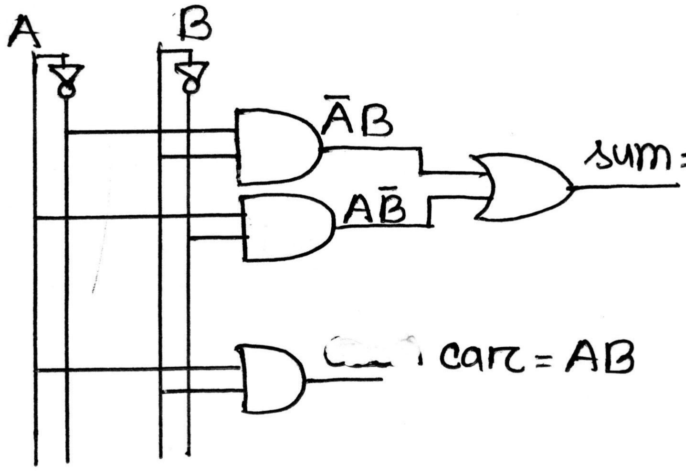
$$carr = AB \quad (AND)$$

* হাফ অ্যাডারের আউটপুটে যোগফলে XOR gate এবং ক্যারি বিটে AND gate পাওয়া যায়।



হাফ অ্যাডারের সার্কিট (যৌগিক)

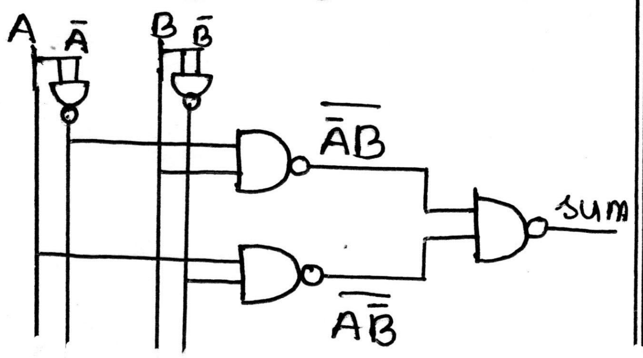
$sum = \bar{A}B + A\bar{B}$ $\qquad$ $carr = AB$



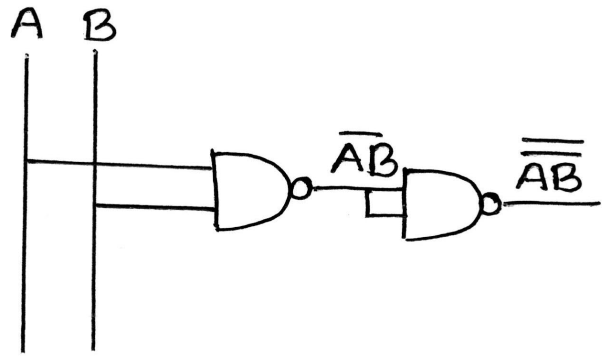
হাফ অ্যাডারের সার্কিট (শুধুমাত্র মৌলিক গেইটের সাহায্যে)

NAND gate দিয়ে হাফ অ্যাডার বাস্তবায়নঃ

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



$$carr = AB$$
$$= \overline{\overline{AB}}$$

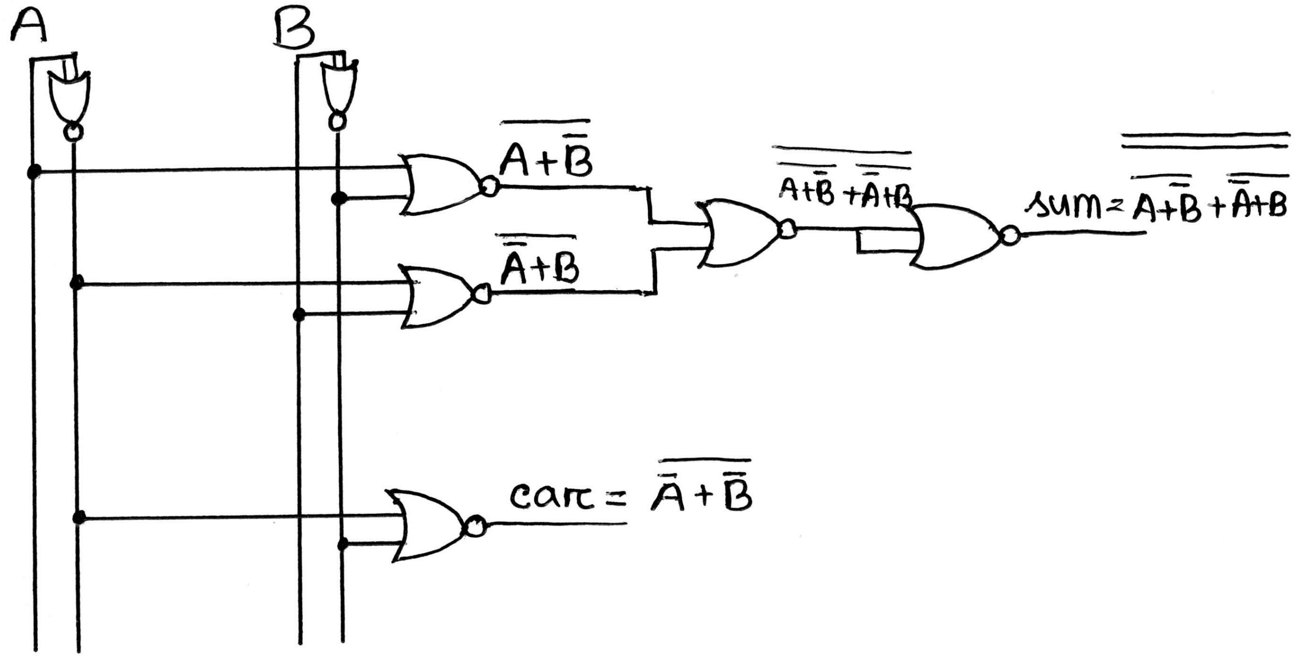


NOR gate দিয়ে হাফ অ্যাডার বাস্তবায়নঃ

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

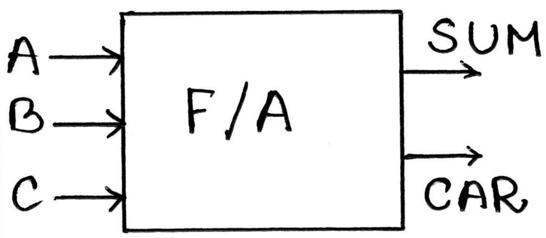
$$carr = AB$$
$$= \overline{\overline{AB}}$$
$$= \overline{\bar{A} + \bar{B}}$$

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



ফুল অ্যাডারঃ

যে সমবায় সার্কিট তিনটি বাইনারী বিট (দুটি ইনপুট বিট ও একটি ক্যারি বিট) যোগ করে যোগফল (SUM) এবং ক্যারি বিট (CAR) আউটপুট দেয় তাকে ফুল অ্যাডার সার্কিট বা পূর্ণ যোগের বর্তনী বলে।



ফুল অ্যাডারের ব্লক চিত্র

A	0	0	0	0	1	1	1	1
B	0	0	1	1	0	0	1	1
C	0	1	0	1	0	1	0	1
	0	1	1	10	1	10	10	11
	↑ S	↑ S	↑ S	↑ S	↑ S	↑ S	↑ S	↑ S

Full Adder 3টি বিট ইনপুট নেয়। output এ যোগফল এবং ক্যারি দেয়।

ইনপুট A, B, C এবং আউটপুট যোগ (SUM) ও ক্যারি (CAR)

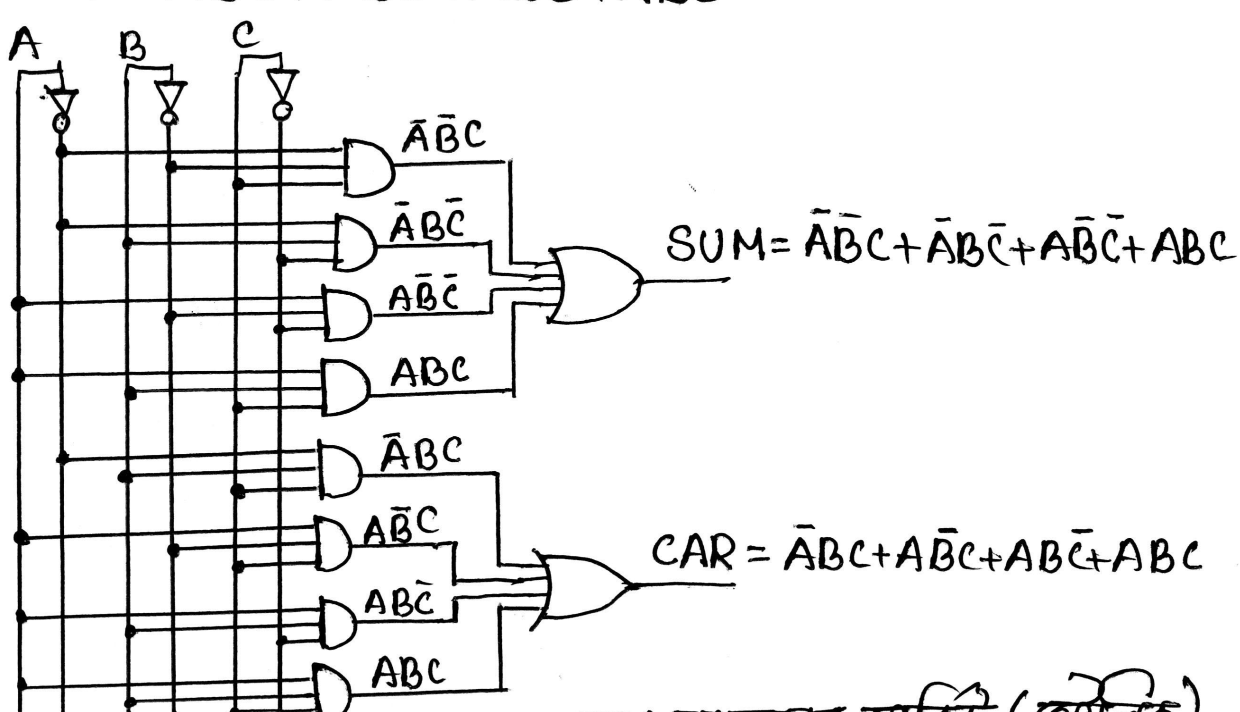
input			output		
A	B	C	SUM	CAR	
0	0	0	0	0	
0	0	1	1	0	$\bar{A}\bar{B}C$
0	1	0	1	0	$\bar{A}B\bar{C}$
0	1	1	0	1	$\bar{A}BC$
1	0	0	1	0	$A\bar{B}\bar{C}$
1	0	1	0	1	$A\bar{B}C$
1	1	0	0	1	$AB\bar{C}$
1	1	1	1	1	ABC

সত্যক সারণী

ফুল অ্যাডারের সত্যক সারণী থেকে আউটপুট (SUM) ও (CAR) এর জন্য নিম্নোক্ত বুলিয়ান ফাংশান লিখা যায় -

$$SUM = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

$$CAR = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$



ফুল অ্যাডারের সার্কিট (মৌলিক)

54

ফুল অ্যাডারের বুলিয়ান ফাংশান সরল করা:

$SUM = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC \downarrow$

NAND/NOR gate দিয়ে বাস্তবায়নে ব্যবহৃত হয়

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

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

$= \bar{A} \cdot Z + A \cdot \bar{Z}$ [ধরি, $B \oplus C = Z$]

$= A \oplus Z$

$= A \oplus B \oplus C \downarrow$ [$\because Z = B \oplus C$]

হাফ অ্যাডার দিয়ে ফুল অ্যাডার বাস্তবায়নে ব্যবহৃত হয়

$CAR = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$

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

$= C(A \oplus B) + AB \cdot 1$ [$\because X + \bar{X} = 1$]

$= (A \oplus B) \cdot C + AB \downarrow$

হাফ অ্যাডার দিয়ে ফুল অ্যাডার বাস্তবায়নে ব্যবহৃত হয়।

$CAR = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$

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

$= (\bar{A}BC + ABC) + (A\bar{B}C + ABC) + (AB\bar{C} + ABC)$

$= BC(\bar{A} + A) + AC(\bar{B} + B) + AB(\bar{C} + C)$

$= BC \cdot 1 + AC \cdot 1 + AB \cdot 1$

$= AB + BC + AC$

$= AB + BC + AC$

NAND/NOR gate দিয়ে বাস্তবায়নে ব্যবহৃত হয়।

Half Adder দিয়ে Full Adder বাস্তবায়ন:

Half Adder:

$sum = A \oplus B$

$car = AB$

Full Adder:

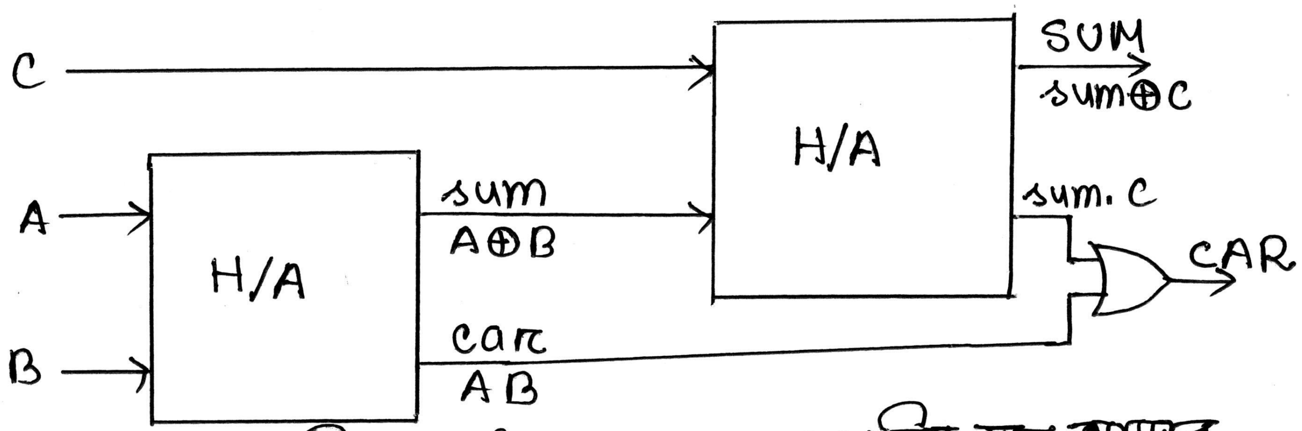
$SUM = A \oplus B \oplus C$

$= sum \oplus C$ $[\because A \oplus B = sum]$

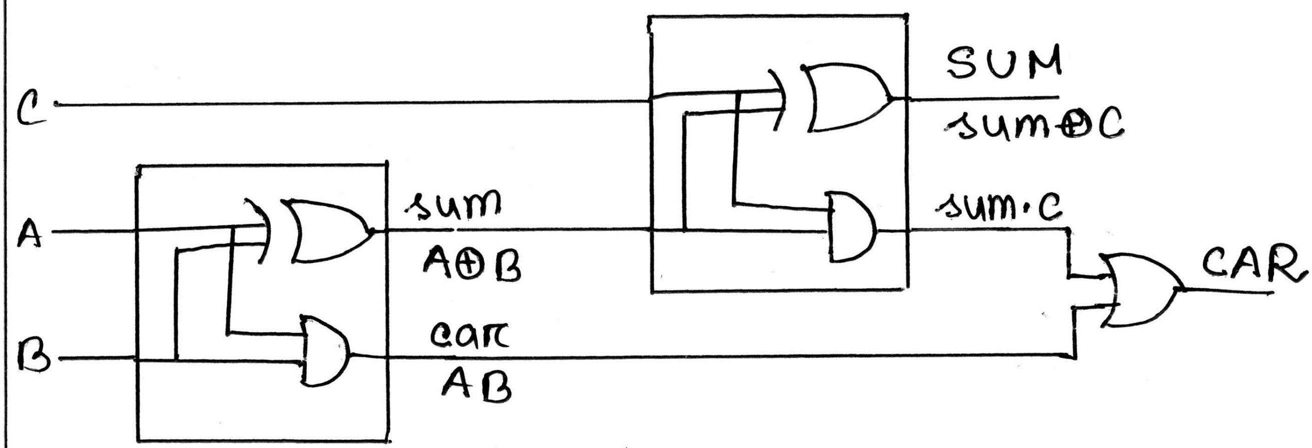
$CAR = (A \oplus B) \cdot C + AB$

$= sum \cdot C + car$

$[\because A \oplus B = sum$
$AB = car]$



ব্লক চিত্রের মাধ্যমে হাফ অ্যাডার দিয়ে ফুল অ্যাডার



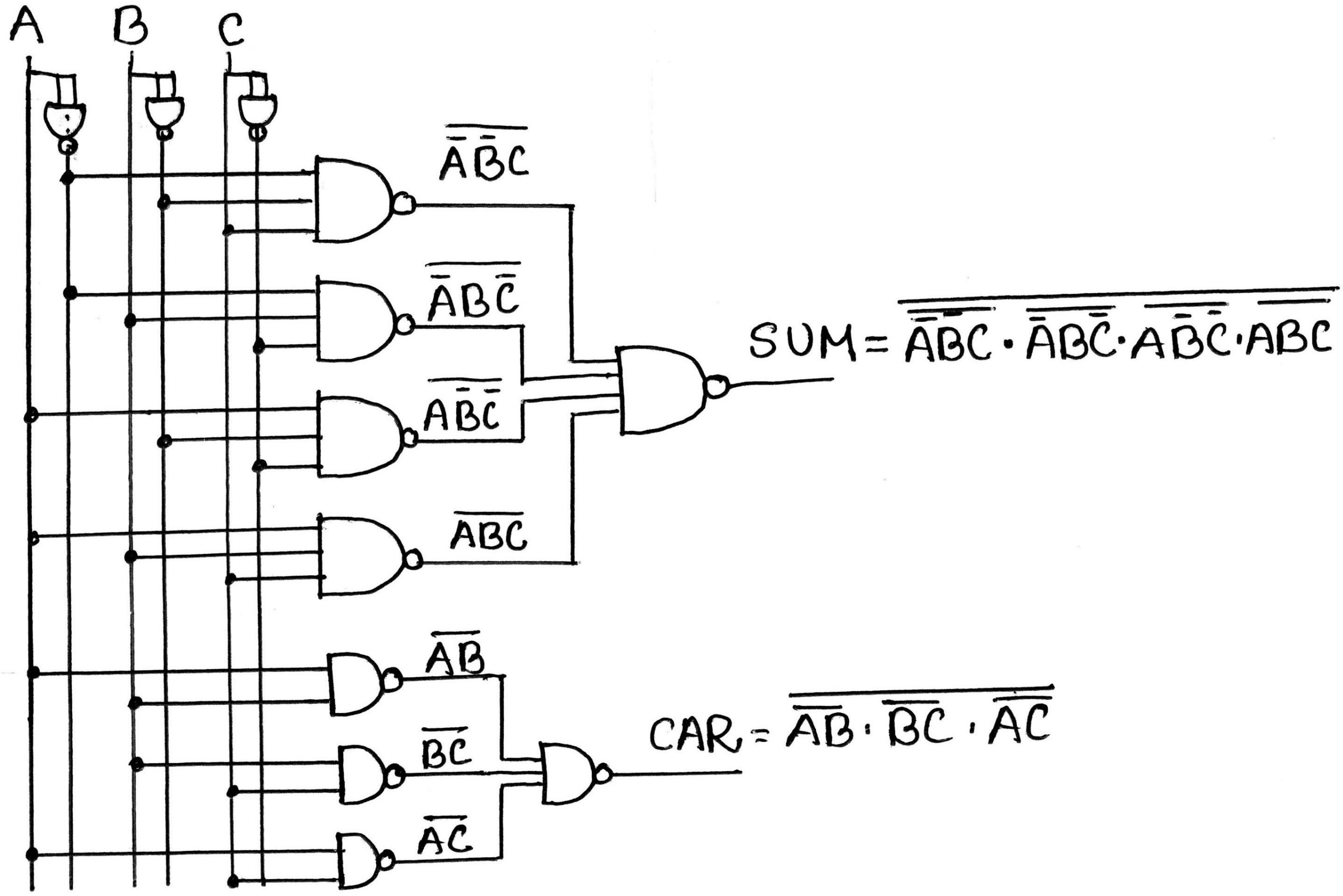
লজিক চিত্রের মাধ্যমে হাফ অ্যাডার দিয়ে ফুল অ্যাডার

NAND gate দিয়ে ফুল অ্যাডার বাস্তবায়ন:

$$SUM = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$
$$= \overline{\overline{\bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC}}$$
$$= \overline{\overline{\bar{A}\bar{B}C} \cdot \overline{\bar{A}B\bar{C}} \cdot \overline{A\bar{B}\bar{C}} \cdot \overline{ABC}}$$

$$CAR = AB + BC + AC$$
$$= \overline{\overline{AB + BC + AC}}$$
$$= \overline{\overline{AB} \cdot \overline{BC} \cdot \overline{AC}}$$

লজিক গেইটের মাধ্যমে

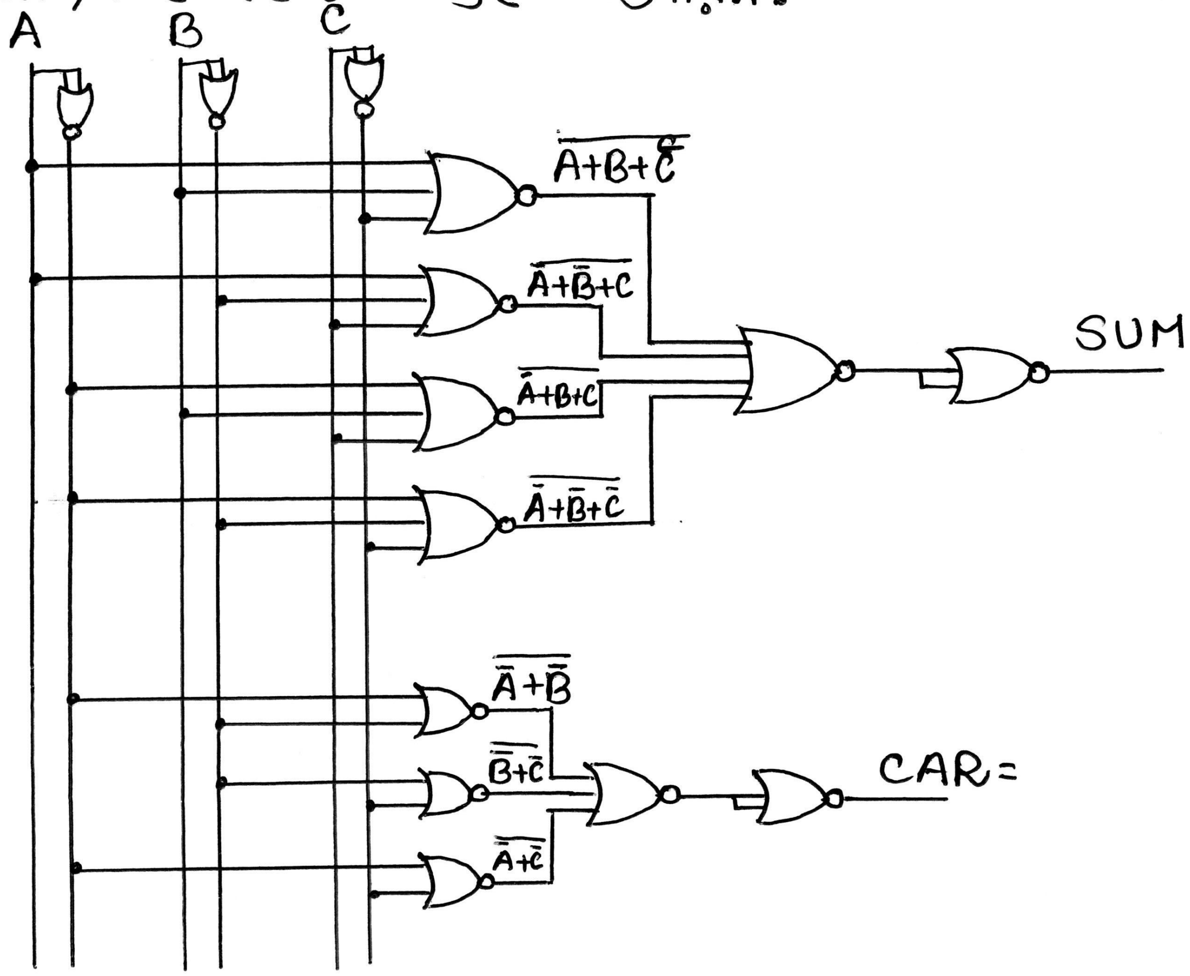


NOR gate ব্যবহার করে ফুল অ্যাডার বাস্তবায়নঃ

$$SUM = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$
$$= \overline{\overline{\bar{A}\bar{B}C}} + \overline{\overline{\bar{A}B\bar{C}}} + \overline{\overline{A\bar{B}\bar{C}}} + \overline{\overline{ABC}}$$
$$= \overline{\bar{\bar{A}}+\bar{\bar{B}}+\bar{C}} + \overline{\bar{\bar{A}}+\bar{B}+\bar{\bar{C}}} + \overline{\bar{A}+\bar{\bar{B}}+\bar{\bar{C}}} + \overline{\bar{A}+\bar{B}+\bar{C}}$$
$$= \overline{A+B+\bar{C}} + \overline{A+\bar{B}+C} + \overline{\bar{A}+B+C} + \overline{\bar{A}+\bar{B}+\bar{C}}$$
$$= \overline{\overline{\overline{A+B+\bar{C}} + \overline{A+\bar{B}+C} + \overline{\bar{A}+B+C} + \overline{\bar{A}+\bar{B}+\bar{C}}}}$$

$$CAR = AB + BC + AC$$
$$= \overline{\overline{AB}} + \overline{\overline{BC}} + \overline{\overline{AC}}$$
$$= \overline{\bar{A}+\bar{B}} + \overline{\bar{B}+\bar{C}} + \overline{\bar{A}+\bar{C}}$$
$$= \overline{\overline{\overline{\bar{A}+\bar{B}} + \overline{\bar{B}+\bar{C}} + \overline{\bar{A}+\bar{C}}}}$$

লজিক গেইটের মাধ্যমে বাস্তবায়নঃ



কাউন্টারঃ

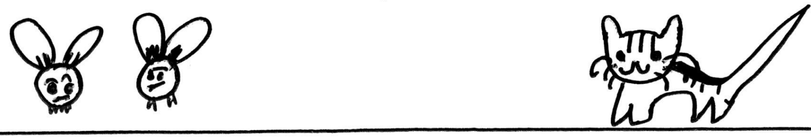
কাউন্টার হল এমন একটি সিকুয়েন্সিয়াল ডিজিটাল ইলেক্ট্রনিক্স সার্কিট যা ফ্লিপ-ফ্লপ এবং লজিক গেইট দিয়ে গঠিত এবং তাতে দেয়া ইনপুট পালসের সংখ্যা গুনতে পারে।

কাউন্টারের মোডঃ

একটি কাউন্টার প্রথম ধাপ থেকে শুরু করে আবার প্রথম ধাপে ফিরে আসতে যতগুলো ধাপ প্রয়োজন হয়, তাকে ঐ কাউন্টারের মোড বলে।

ফ্লিপফ্লপঃ

ফ্লিপফ্লপ হলো লজিক গেইট দিয়ে তৈরী এক ধরনের ডিজিটাল বর্তনী যা একবিট তথ্য ধারন করতে পারে।



রেজিস্টার:

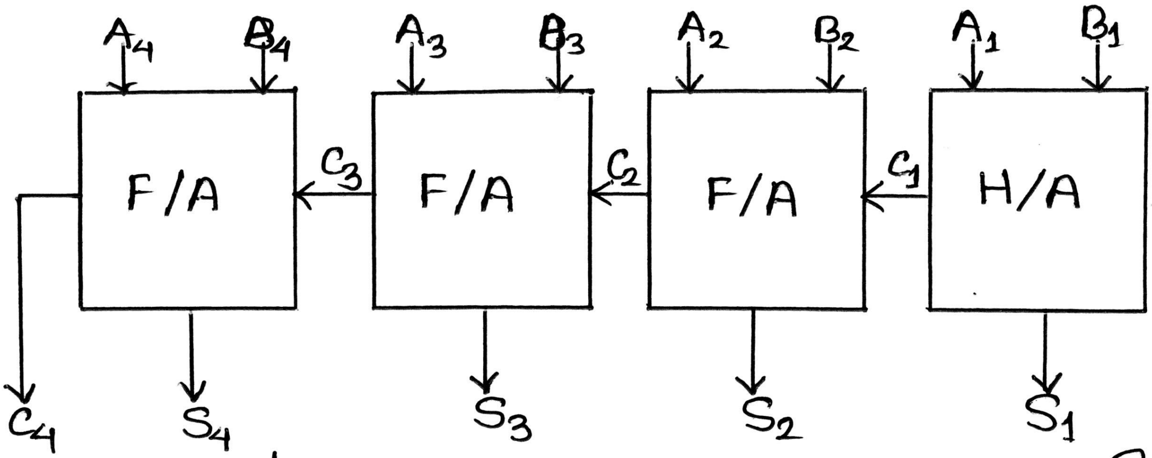
রেজিস্টার হলো একগুচ্ছ ফ্লিপ-ফ্লপ এবং লজিক গেইটের সমন্বয়ে গঠিত সার্কিট যা অস্থায়ী মেমোরি হিসেবে কাজ করে। 4bit 16 bit, 32 bit register।

প্যারালাল বাইনারী অ্যাডার:

$$\begin{array}{r} {}^{+1}\;{}^{+1}{}^{+1}{}^{+1}\\ 8795 \\ +\ 7638 \\ \hline 16433 \end{array}$$

দুইটি বাইনারী সংখ্যা $A_4A_3A_2A_1$ এবং $B_4B_3B_2B_1$

$$\begin{array}{r} +C_4\ +C_3\ +C_2\ +C_1 \\ A_4\ A_3\ A_2\ A_1 \\ +\ B_4\ B_3\ B_2\ B_1 \\ \hline C_4\ S_4\ S_3\ S_2\ S_1 \end{array}$$



Half Adder ও Full Adder ব্যবহার করে 4 bit বাইনারী প্যারালাল অ্যাডার।

ByE