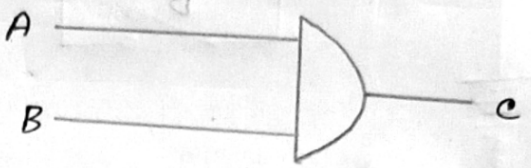
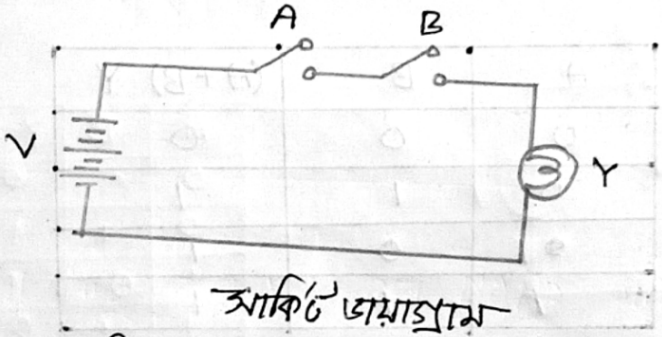
□ লজিক গেইট সমূহের- লজিক প্রতিক, সত্যক সারনি, বুলিয়ান রাশি

AND GATE :-



লজিক প্রতীক

বুলিয়ান রাশি :- $Y = AB$



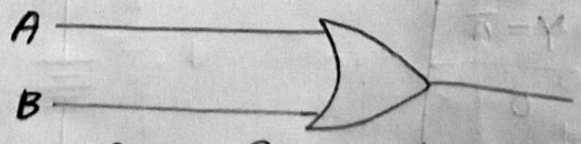
সার্কিট ডায়াগ্রাম

সত্যক সারনি

A	B	Y = AB
0	0	0
0	1	0
1	0	0
1	1	1

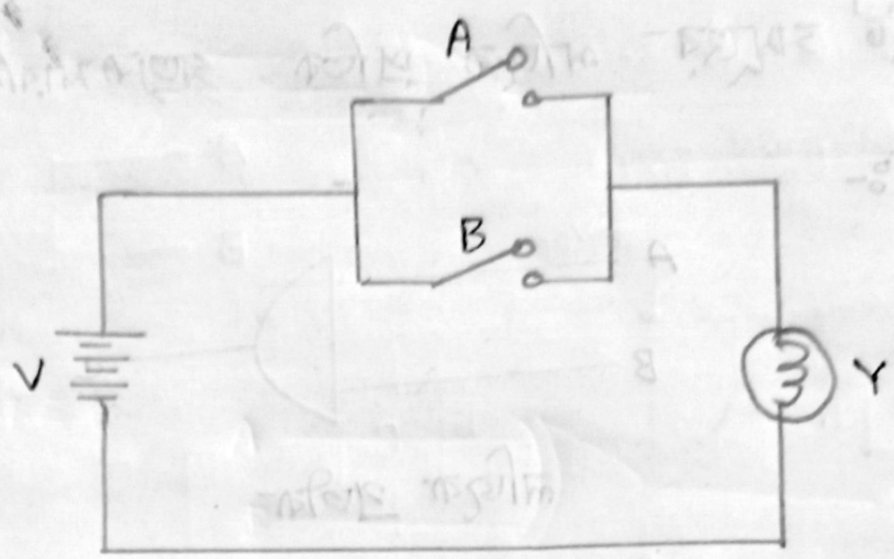
যে কোনো একটি মান (input মান) 0 হলে আউটপুট শূন্য হবে।

OR GATE :-



লজিক প্রতীক

বুলিয়ান রাশি :- $Y = A+B$



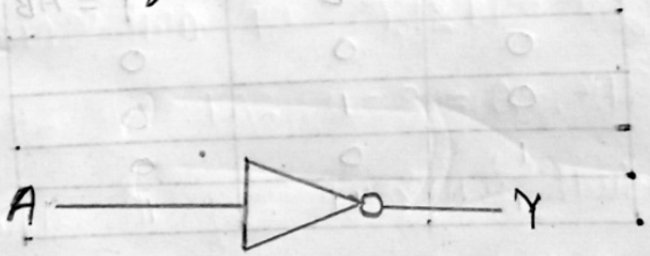
সার্কিট ডায়াগ্রাম

সত্যক সারণি

A	B	(A+B) Y
0	0	0
0	1	1
1	0	1
1	1	1

যে কোনো একটি মান High হলে আউটপুট হাই অন্য হবে

NOT GATE :-



লজিক প্রতিক

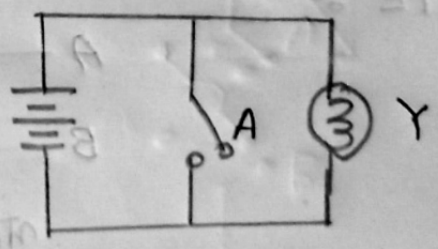
সমীকরন :-

$Y = \bar{A}$

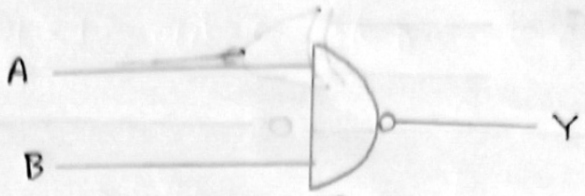
সার্কিট ডায়াগ্রাম

সত্যক সারণি,

A	$Y = \bar{A}$
1	0
0	1

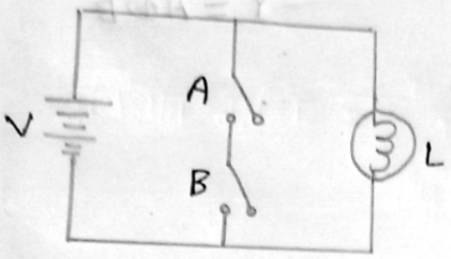


NAND GATE :-



লজিক প্রতীক-

লজিক সার্কিট-



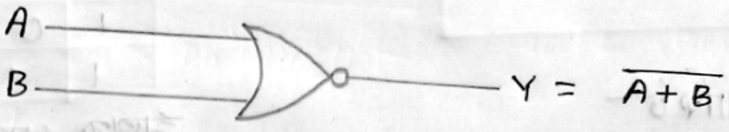
সত্যক সারনি

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

বুলিয়ান রাশি - $\overline{AB}$

যে কোনো একটি বা দুইটি মান LOW হলে Output High হবে।

NOR GATE :-

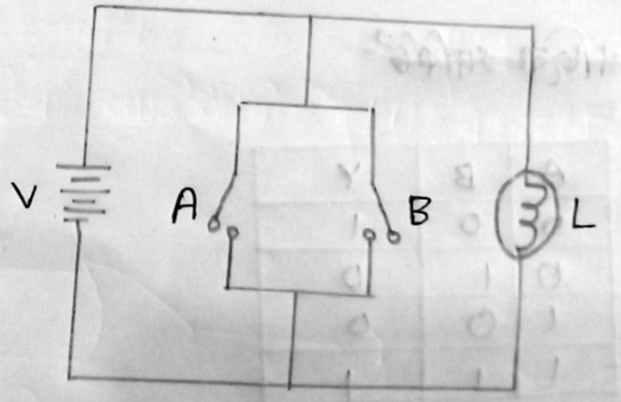


লজিক প্রতিক-

সত্যক সারনি

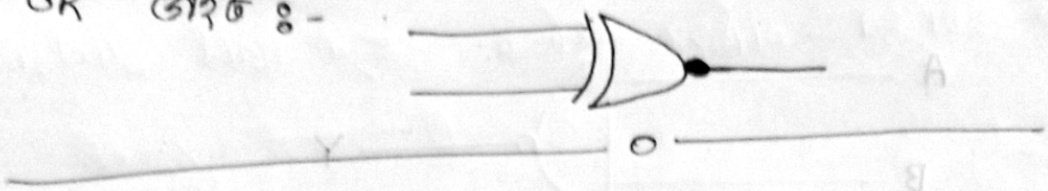
A	B	$Y = \overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

লজিক সার্কিট-

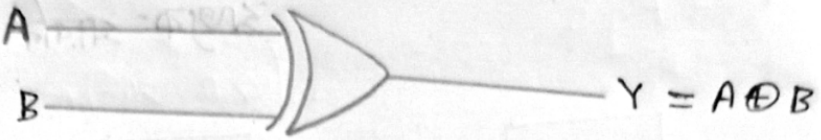


যে কোনো একটা মান High হলে আউটপুট LOW হবে,

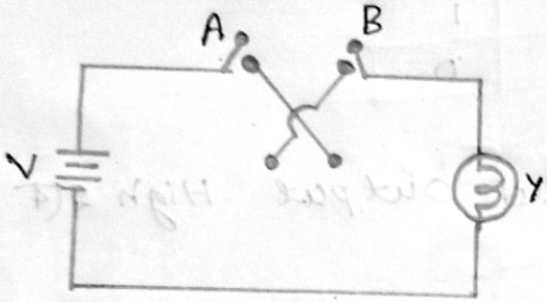
Ex OR গেইট :-



Ex OR GATE :-



লজিক প্রতীক

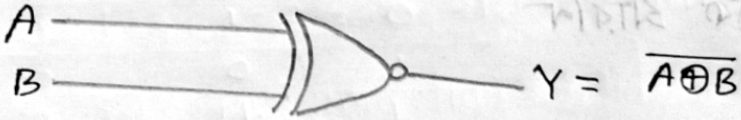


লজিক সার্কিট-

A	B	Y = A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

সত্যক সারণি.

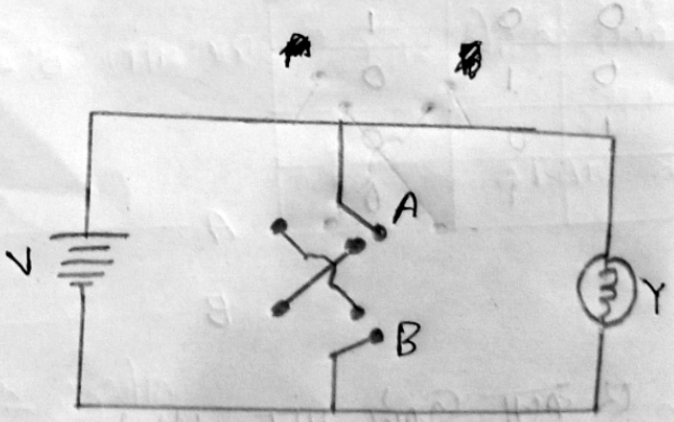
Ex NOR GATE :-



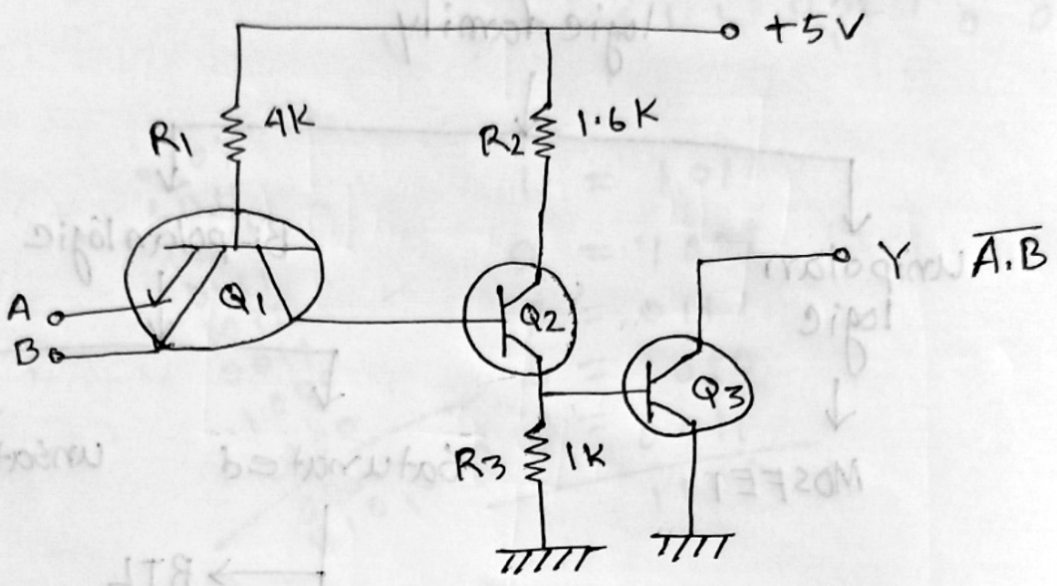
~~লজিক সার্কিট-~~

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

সত্যক সারণি



একটি standard TTL NAND গেটের বর্ণনা ।



TTL NAND গেইট-

উপরের চিত্রে একটি TTL NAND গেইট এর সার্কিট ডায়াগ্রাম দেখানো হয়েছে, Q_1, Q_2 ও Q_3 তিনটি Bipolen ট্রানজিস্টর যা $R_1 = 4K\Omega$, $R_2 = 1.6K\Omega$ ও $R_3 = K\Omega$ রেজিস্ট্যান্স এর সাথে সংযুক্ত। রেজিস্ট্যান্স গুলো ট্রানজিস্টরের ভোল্টেজ লেভেল থাপ আউন করে, সার্কিটের ইনপুট দ্বয়ের মান যদি ~~Hing~~ High হয় $(A.B) = (1.1)$ তাহলে আউটপুট হবে $Y=0$, আর সার্কিটের যেকোনো একটি ইনপুটের মান low হলে ~~আউটপুট~~ আউটপুট High হবে এবং সার্কিট সাধারণ ভাবে ভোল্টেজ সরবরাহ করে।

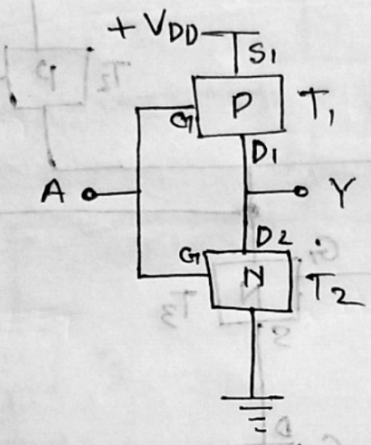
A	B	Y
1	0	1
0	1	1
0	0	1
1	1	0

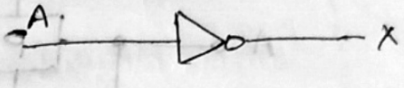
Trouth Table

0 = low
1 = High

CMOS NOT, NAND ও NOR গেইটের প্রত্যেকটির বর্তনী-আরেখন পূর্বক কার্যপ্রণালী বর্ণনা।

CMOS NOT গেইট



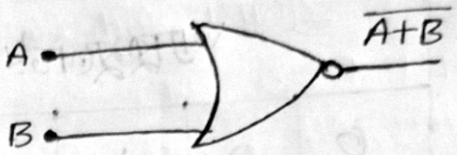


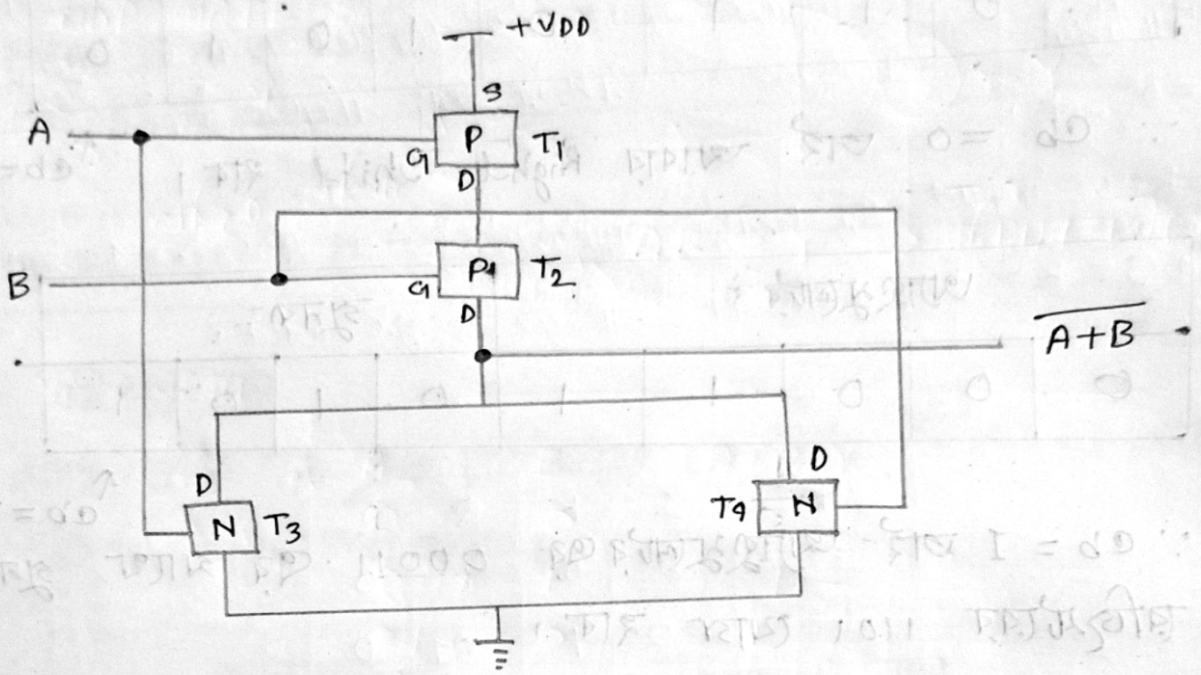
A	T1	T2	X
+VDD	off	on	0
0	on	off	+VDD

উপরের চিত্রে, CMOS NOT গেইটের সার্কিট ডায়াগ্রাম দেখানো হয়েছে, যেখানে দুই type MOS গেইট রয়েছে, PMOS ও N MOS. PMOS কে T1 ও N MOS কে T2 দ্বারা সাংকেতিত করা হয়।

যেখানে, ইনপুট A এর মান যদি +VDD হয় তাহলে T1 [PMOS] off থাকে এবং T2 [N MOS] on থাকে, এবং আউটপুট হয় 0, আবার input A এর মান 0 হলে NMOS বা T2 ~~on~~ থাকবে off থাকে এবং PMOS T1 চালু থাকে, A এর ~~an~~ value high হলে y=low আবার, A এর value low হলে y = high.

☐ CMOS NOR গেইট :



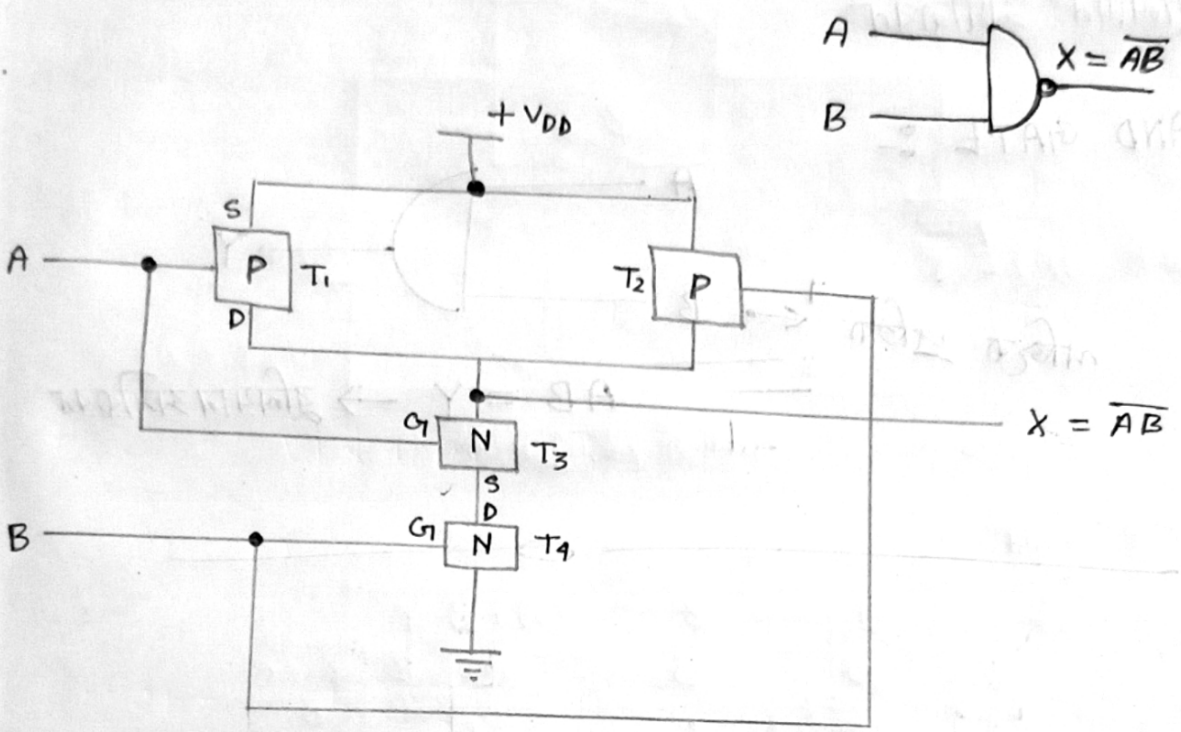


এই গেইটের কার্যপ্রনালী নিচে দেওয়া হলো:

$A=B=0$ ভোল্ট হলে T_1, T_2 চালু এবং T_3, T_4 বন্ধ থাকে ফলে $X=+V_{DD}$ হয়, $A=+V_{DD}$ ও $B=0$ ভোল্ট হলে T_2 ও T_3 চালু ও T_1 ও T_4 বন্ধ থাকে, $A=0$, $B=+V_{DD}$ হলে T_2 ও T_3 অচালু থাকে ও T_1 ও T_4 চালু থাকে, $A=B=+V_{DD}$ হলে ~~T_2 ও T_3~~ T_1 ও T_2 বন্ধ ও T_3, T_4 চালু থাকে।
∴ বর্তনীটি NOR গেইট হিসেবে কাজ করে।

$A = T_1 T_3$
$B = T_2 T_4$

□ CMOS NAND গেইট 8-



NOT গেইট এর কিছু পরিবর্তন করে NAND GATE তৈরি করা যায়।

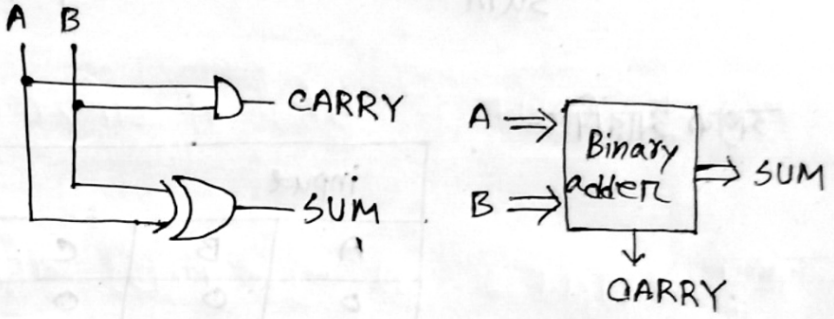
A = 0 ভোল্ট ও B = +VDD অথবা A = +VDD ও B = 0 ভোল্ট এই দুইটি গ্রহনমুখ অবস্থায় যে কোনো একটি P চ্যানেল MOSFET চালু থাকে, কিন্তু A = B = 0 হলে, P চ্যানেল MOSFET চালু থাকে। কিন্তু কোনো ট্রানজিস্টর চালু না থাকায় X = +VDD হয়, আবার, A = B = +VDD হলে N চ্যানেল দুইটি চালু থাকায় X = 0 ভোল্ট হয়। এই কাজের বর্ণনাটি এই NAND গেইট হিসেবে কাজ করে।

[1] হাফ অ্যাডার ও ফুল অ্যাডার আর্কিটেক্টের কার্য প্রনালী বর্ননা

হাফ অ্যাডার :- দুইটি বিটকে যোগ করার জন্যে যোজের যে বর্তনী ব্যবহার করা হয় তাকে হাফ অ্যাডার আর্কিটেক্ট বা বর্তনী বলে।
দুইটি বিট যোগ করার ক্ষেত্রে বর্তনী যোগফল (SUM) এবং (CARRY) প্রদর্শন করে।

(i) 0+0 = 0 (ii) 0+1 = 1 (iii) 0+1 = 1 (iv) 1+1 = 10

আর্কিটেক্ট ডায়াগ্রাম :-

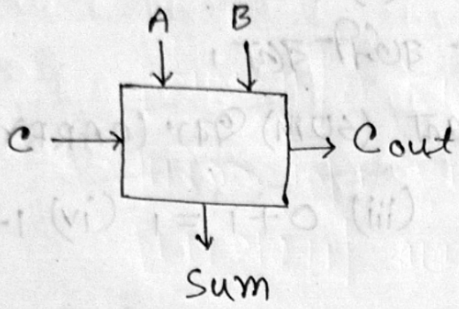


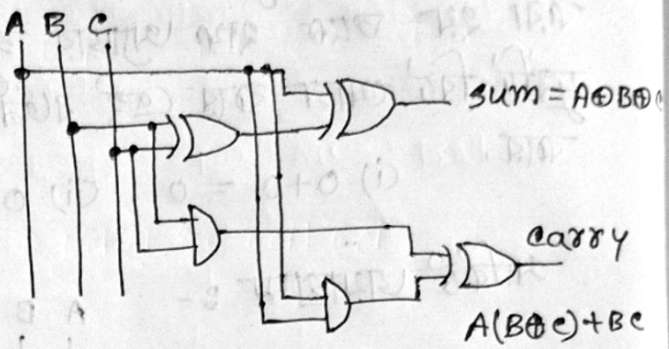
সত্যক সারনি :-

input		output	
A	B	Sum	Carry
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

সত্যক সারনিতে দেখা যাচ্ছে, যখন A=0, B=0 হয়, তখন, Sum=0 Carry =0, A=0, B=1 তখন, Sum = 1, Carry = 0, যখন A=1, B=0 তখন, Sum = 1, Carry =0, এবং যখন, A=1 B=1 তখন, Sum = 0, Carry =1 ।

ফুল অ্যাডার :- যে অ্যাডার সার্কিট তিনটি বাইনারি বিটকে একত্রে যোগ করতে পারে, তাকে ফুল অ্যাডার বলে।





সত্যক সারনি :-

input			output	
A	B	C	S	C_{out}
0	0	0	0	0
0	0	1	1	0
1	0	0	1	0
0	1	0	1	0
1	1	0	0	1
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

যখন, A, B, C = 0 হয় তখন S ও C = 0 । যখন, A = 0, B = 0 C = 1 তখন, S = 1, C = 0 হয়, যখন, A = 0, B = 1, C = 0 তখন, S = 1, C = 0, যখন, A = 1, B = 1 C = 0 তখন S = 0, Carry = 1 , সর্বশেষ আবার যখন A, B, C = 1 হয় তখন, S = 1, Carry = 1 হবে।

* Hamming code এর সাহায্যে ভুল সংশোধন

বাইনারি বা অন্যান্য কোড ব্যবহার করে আমাদের প্রক্রিয়াতে তথ্য প্রেরন করা হয়ে থাকে। নানা কারনে কোড পরিবর্তিত হয়ে গ্রাহক গ্রাহক যন্ত্রে ভুল আকারে ধরা পরে। এই ত্রুটি সংশোধন এর জন্য নানা প্রকার কোড ব্যবহার করা হয়। হামিং কোড এগুলোর মধ্যে উল্লেখ যোগ্য।

ভুল সংশোধনের জন্য কোডের 1, 2, 4, 8, 16, ইত্যাদি স্থান কোডের মধ্যে প্যারিটি-বিট বসানো হয়।

10।।001 এর ভিতরে ১ম, ২য় ও অষ্টম স্থানে হামিং প্যারিটি স্থাপন করা যায়।

1	0	1	h_4	1	0	0	h_3	1	h_2	h_1
11	10	9	8	7	6	5	4	3	2	1

হামিং বিটগুলো নির্ণয় করার জন্য যে যে স্থানে 1 আছে তা নির্ণয় করা হয় এবং এই সংখ্যাগুলোকে বাইনারিতে রূপান্তর করে তার যোগফল নির্ণয় করা হয়।

এখানে, 3, 7, 9, 11 অবস্থানে 1 আছে।

3 এর বাইনারি		0011
7 ,,	,,	0111
9 ,,	,,	1001
11 ,,	,,	1011
		0110
		$h_4h_3h_2h_1$

এ বিষয়টা ১ম এর ধারণা থাকলে hamming code পাওয়া যাবে,

১	০	১	০	১	০	০	১	১	১	০
১১	১০	৯	৮	৭	৬	৫	৪	৩	২	১

প্রাইম থাকে সে সে স্থানে। এখানে তার বাইনারির সোর্সফল যদি ০ হয় তাহলে কোনো দুনা থাকতে না তার যদি কোনো স্থানে ১ আসে তাহলে দুনা থাকতে,

সে সে স্থানে। আছে তা হলো: ২, ৩, ৭, ৮, ৯, ১১

২ এর বাইনারি, ০০১০
৩ " " ০০১১
৭ " " ০১০০
৮ " " ০১১১
৯ " " ১০০১
১১ " " ১০১১

০০০০

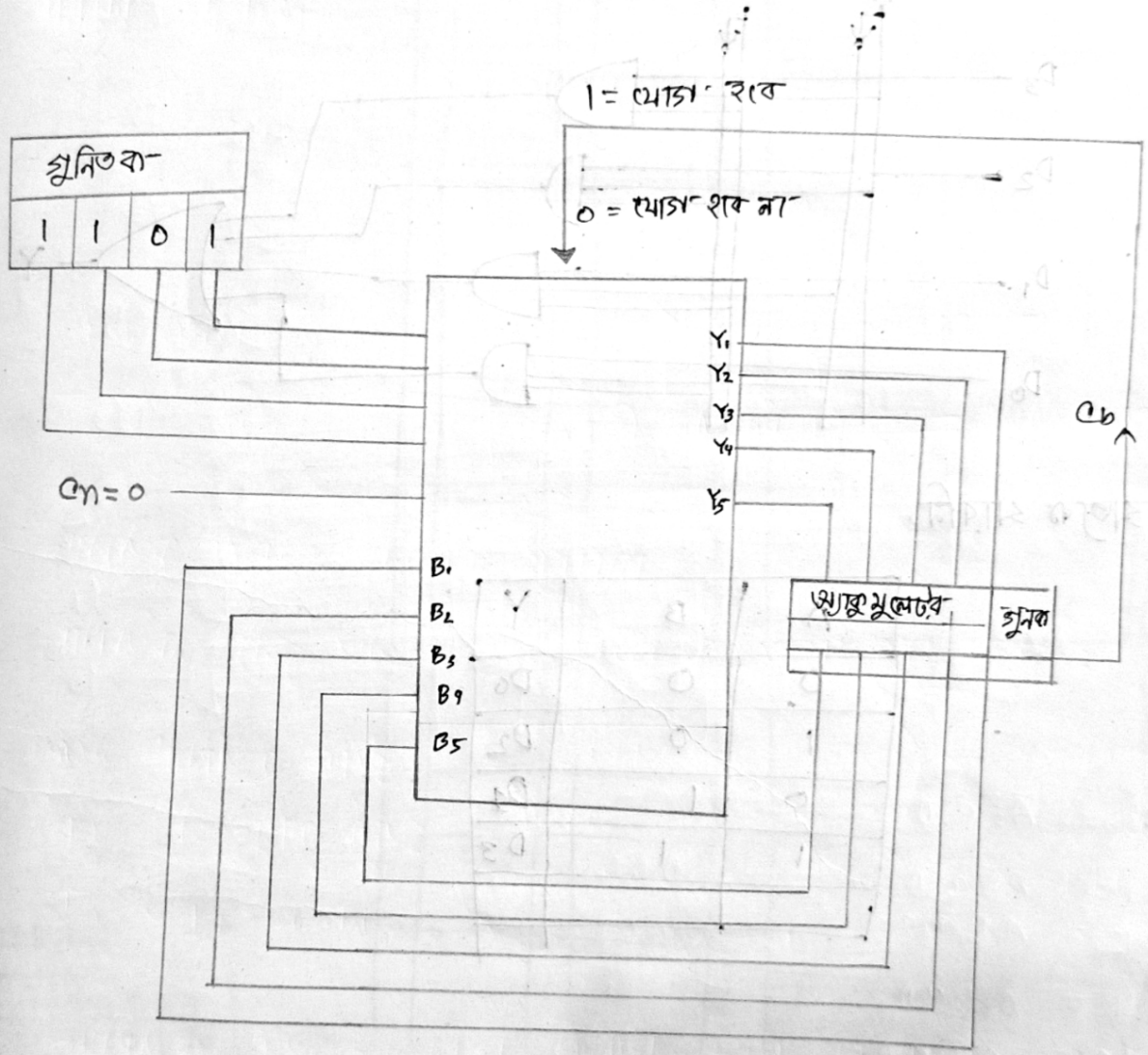
যদি কোনো যায়গা কোনো ডিজিট মুক্ত হয় তাহলে, স্থানের বাইনারির সোর্সফল ০ হবে না,
যেমন করি ৫ এর স্থানে। হবে যেমন,
তাহলে,

২ = ০ ০ ১ ০
৩ = ০ ০ ১ ১
৭ = ০ ১ ০ ০
৫ = ০ ১ ০ ১
৮ = ০ ১ ১ ১
৯ = ১ ০ ০ ১
১১ = ১ ০ ১ ১

০১০১

বাইনারি রেট মাল্টিপ্লায়ার অপারেশন বর্ননা

যে লজিক সার্কিটের সাহায্যে দুইটি বাইনারি সংখ্যার গুন করা হয় তাকে বাইনারি রেট মাল্টিপ্লায়ার বলে। দুইটি সংখ্যার মধ্যে যে সংখ্যাকে গুন করা হয় তাকে গুনিতক বলে, যে সংখ্যা দিয়ে গুন করা হয় তাকে গুনক বা মাল্টিপ্লায়ার বলে।



চিত্র :- বাইনারি রেট মাল্টিপ্লায়ারের লজিক ডায়াগ্রাম

কার্যপ্রনালী :- এ মাল্টিপ্লায়ারের কার্যপ্রনালী বুঝতে হলে অ্যাকুমুলেটর গুনক রেজিস্টারের অবস্থা প্রতিটি ধাপে ধাপে অনুসরন করতে হবে। সাধারন গুনিতক রেজিস্টারে 1101 রাখা হলো, অ্যাকুমুলেটর কে রিসেট করা হলো এবং গুনক রেজিস্টারে 1010 রাখা হলো।

যদি $c_b = 1$ হয় তবে গুনিতক রেজিস্টারের সংখ্যা ও অ্যাকুমুলেটর এর সংখ্যা যোগ হবে। $c_b = 0$ হলে যোগ হবে না।
অ্যাকুমুলেটরকে রিসেট করার পর অ্যাকুমুলেটর গুনক রেজিস্টারের অবস্থা নিম্নরূপ,

অ্যাকুমুলেটর					গুনক			
0	0	0	0	0	1	0	1	0

$\searrow c_b$

∴ ~~c_b এর মান~~
অ্যাকুমুলেটর এর গুনক রেজিস্টারের ৩ বিট বিশিষ্ট সংখ্যাকে Right shift করা হলো।

অ্যাকুমুলেটর					গুনক			
0	0	0	0	0	0	1	0	1

$\searrow c_b$

∵ $c_b = 1$ তাই অ্যাকুমুলেটর এর 00000 এবং গুনিতক রেজিস্টারের 1101 যোগ হবে। ফলাফল 01101 অ্যাকুমুলেটর এ জমা হবে,

অ্যাকুমুলেটর					গুনক			
0	1	1	0	1	0	1	0	1

এবার অ্যাকুমুলেটর এর গুনক রেজিস্টারের সংখ্যাকে Right shift করে পাই,

$\longrightarrow$

অ্যাকুমুলেটর					গুনক			
0	0	1	1	0	1	0	1	0

↑ eb = 0

∵ eb = 0 তাই আবার Right Shift হবে।

অ্যাকুমুলেটর					গুনক			
0	0	0	1	1	0	1	0	1

↑ eb = 1

∵ eb = 1 তাই অ্যাকুমুলেটর এর 00011 এর সাথে গুনক রেজিস্টারের 1101 যোগ হবে।

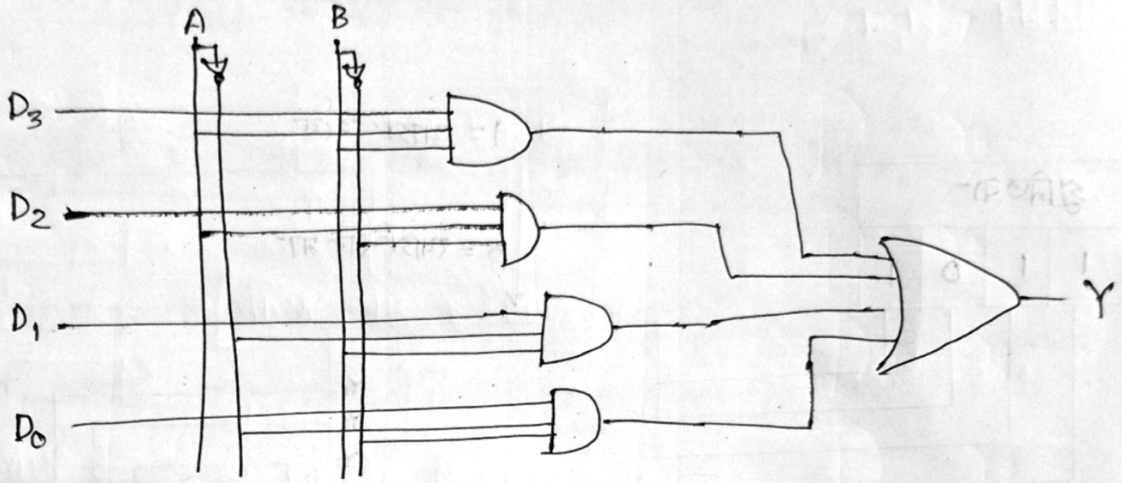
অ্যাকুমুলেটর					গুনক			
1	0	0	0	0	0	1	0	1

গুনক রেজিস্টারের মান আবার Right Shift করে পাই,

অ্যাকুমুলেটর					গুনক			
0	1	0	0	0	0	0	1	0

যদি $C_b = 1$ হয় তবে গুনিতক রেজিস্টারের সংখ্যা ও অ্যাকুমুলেটরের সংখ্যা যোগ হবে। এবং যদি $C_b = 0$ হয় তবে গুনিতক রেজিস্টারের সংখ্যা ও অ্যাকুমুলেটর রেজিস্টারের সংখ্যা যোগ হবে না।

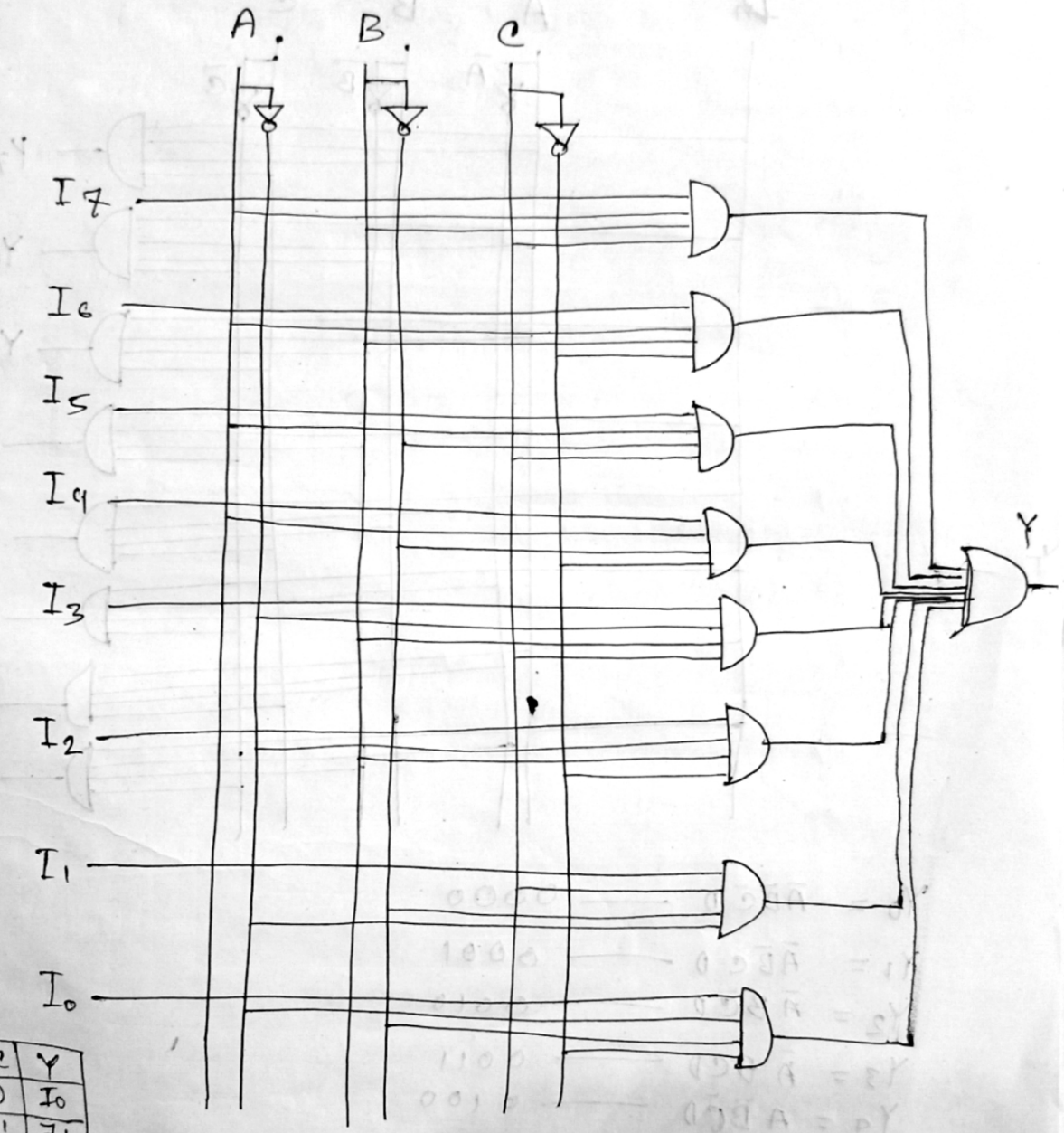
~~4:1~~ (4:1) মাল্টিপ্লেক্সার



সত্যক সারনি,

A	B	Y
0	0	D_0
1	0	D_2
0	1	D_1
1	1	D_3

(8:1) মাল্টিপ্লেক্সার :- 8:1 multiplexer এ ৪টি input মান ও 1টি আউটপুট মান থাকে। নিচে (8:1) মাল্টিপ্লেক্সার এর সার্কিট-ডায়াগ্রাম দেখানো হলো:

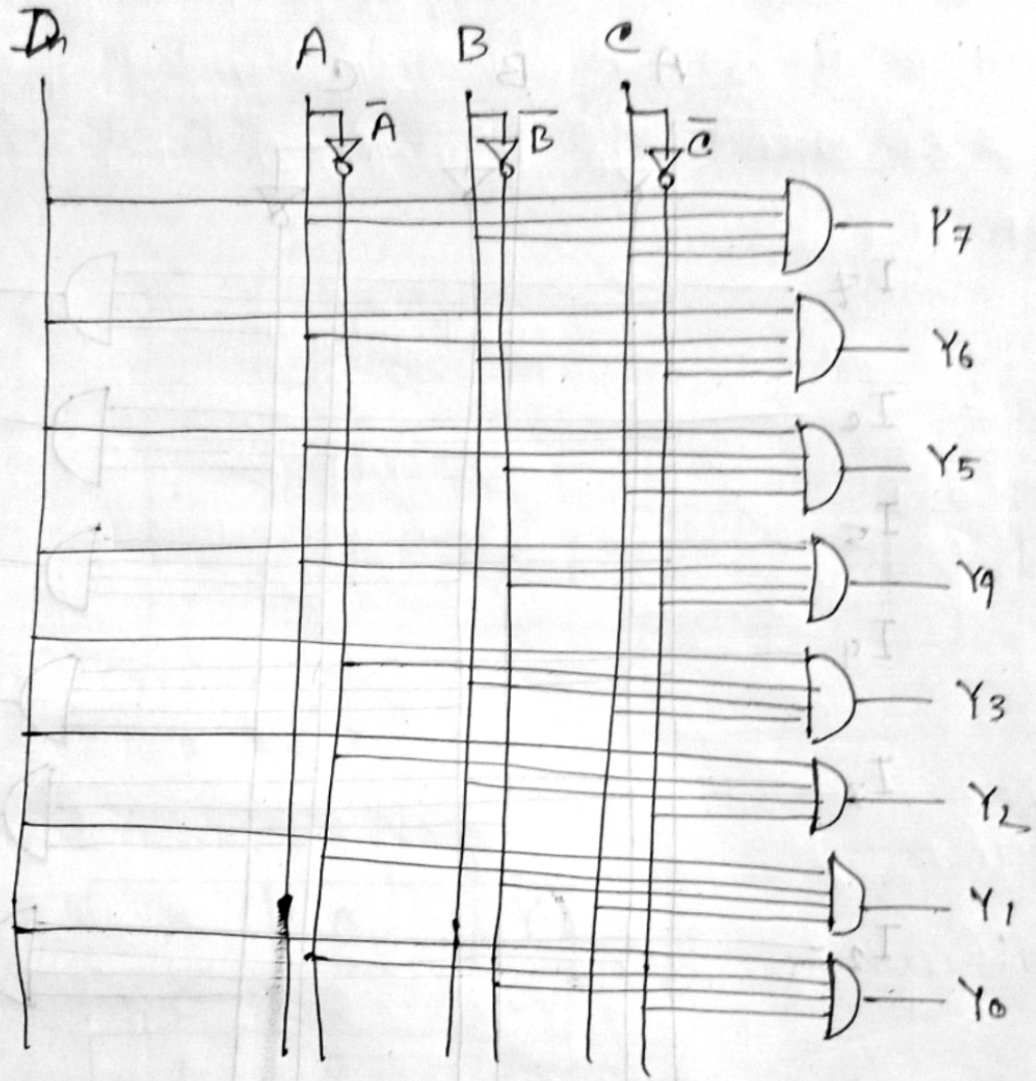


সত্যকসার্ণি :

A	B	C	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

1:8 ডিমাল্টিপ্লেক্সার :-

1টি input মান ও 8টি আউটপুট মান থাকে।



$Y_0 = \bar{A}\bar{B}\bar{C}D$ —— 0000

$Y_1 = \bar{A}\bar{B}CD$ —— 0001

$Y_2 = \bar{A}B\bar{C}D$ —— 0010

$Y_3 = \bar{A}BCD$ —— 0011

$Y_4 = A\bar{B}\bar{C}D$ —— 0100

$Y_5 = A\bar{B}CD$ —— 0101

$Y_6 = AB\bar{C}D$ —— 0110

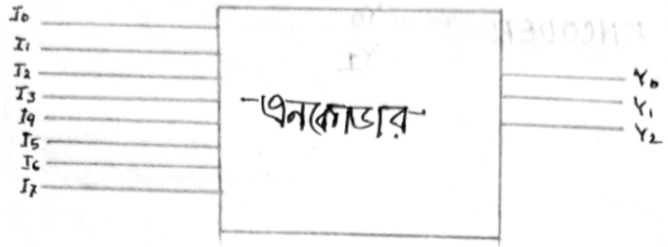
$Y_7 = ABCD$ —— 0111

প্রত্যক্ষ সারনি :-

D	A	B	C	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
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	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	1	0	0	0
1	1	0	1	0	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	0	0	1	0
1	1	1	1	0	0	0	0	0	0	0	1

৪:৩ এনকোডার লজিক সার্কিট ও টুথ টেবিল :-

৪ input ও ৩ output বিশিষ্ট Encoder দ্বারা বাইনারি বিট গুলোর OR গেইট এর সাহায্যে সমাধান করা হয়,

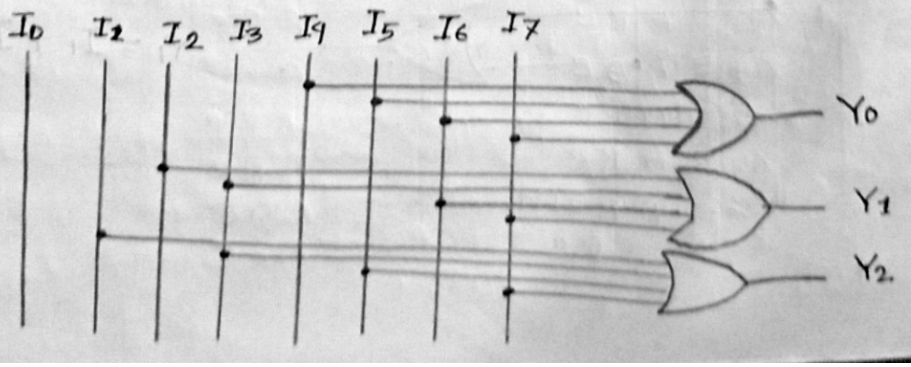


input								output		
I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	Y_0	Y_1	Y_3
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

$$Y_0 = I_4 + I_5 + I_6 + I_7$$

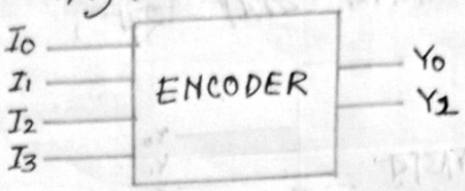
$$Y_1 = I_2 + I_3 + I_6 + I_7$$

$$Y_2 = I_1 + I_3 + I_5 + I_7$$



4:2 এনকোডার লজিক সার্কিট ও ট্রুথ টেবিল

4 input ও 2 output বিশিষ্ট এনকোডার দ্বারা বাইনারি বিট গুলো OR গেইট এর সাহায্যে সমাবেশন করা হয়।

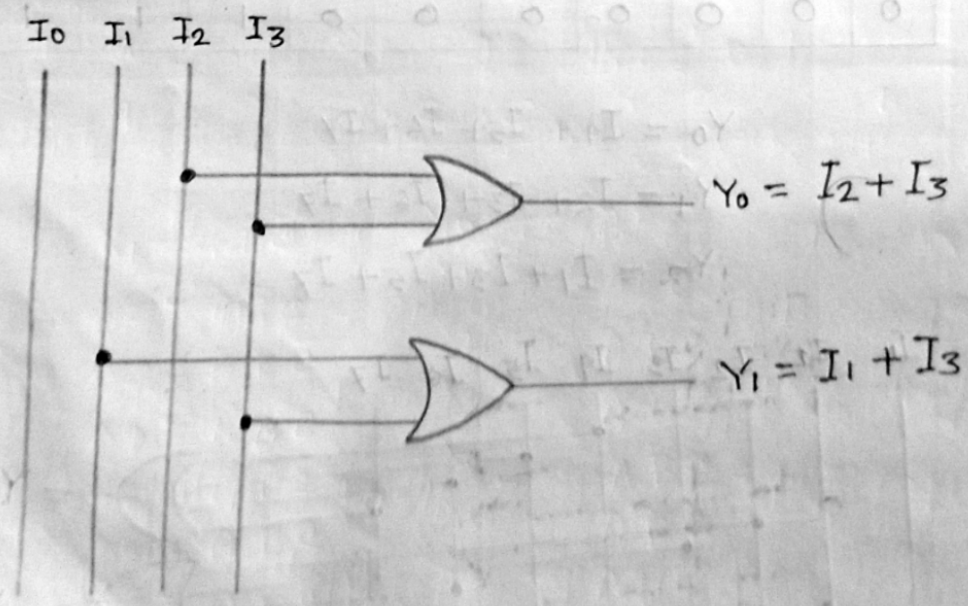


input				output	
I_0	I_1	I_2	I_3	Y_0	Y_1
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

সত্যক সারনি হতে দেখা যায়,

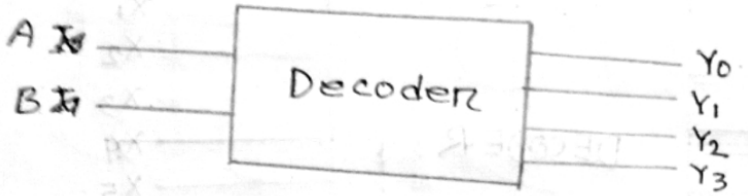
$$Y_0 = I_2 + I_3$$

$$Y_1 = I_1 + I_3$$



2:4 ডিকোডার লজিক সার্কিট ও সত্যক সারনি :-

2:4 বা 2 input ও 4 output বিশিষ্ট Decoder দ্বারা বাইনারি বিট গুলো AND গেইট এর সাহায্যে সমাবিষ্ট করা হয়।



input		Output			
A	B	Y_0	Y_1	Y_2	Y_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

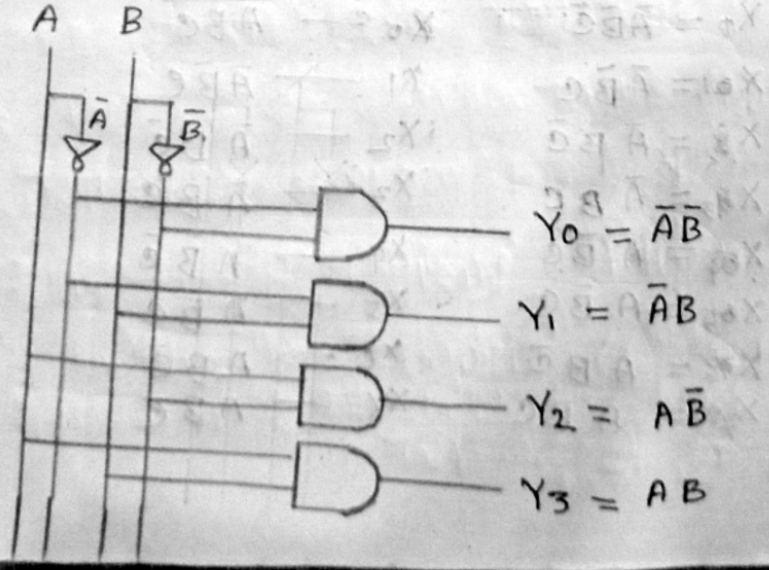
সত্যক সারনি হতে,

$Y_0 = \bar{A}\bar{B}$

$Y_1 = \bar{A}B$

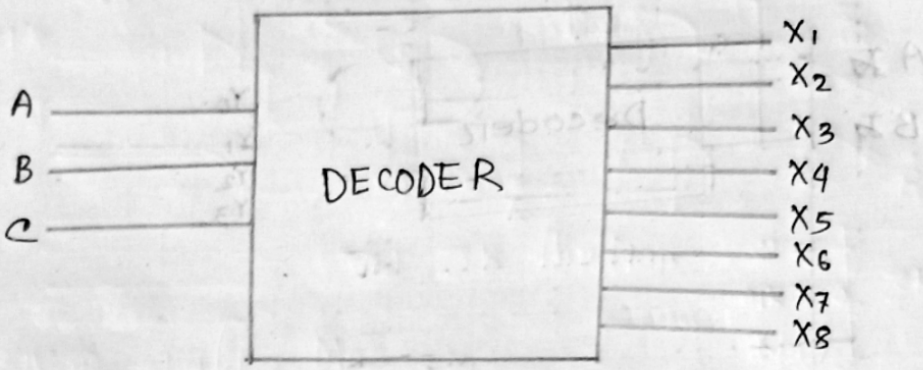
$Y_2 = A\bar{B}$

$Y_3 = AB$



3:8 DECODER লজিক সার্কিট ও সত্যক সারণি :-

3 input ও 8 output বিশিষ্ট DECODER এ বাইনারি বিটগুলো AND গেইটের সাহায্যে সমাধান করা হয়।



input			output							
A	B	C	X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_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

$X_0 = \bar{A}\bar{B}\bar{C}$ $\quad X_0 \text{ —} \overline{ABC}$

$X_1 = \bar{A}\bar{B}C$ $\quad X_1 \text{ —} \bar{A}\bar{B}C$

$X_2 = \bar{A}B\bar{C}$ $\quad X_2 \text{ —} \bar{A}B\bar{C}$

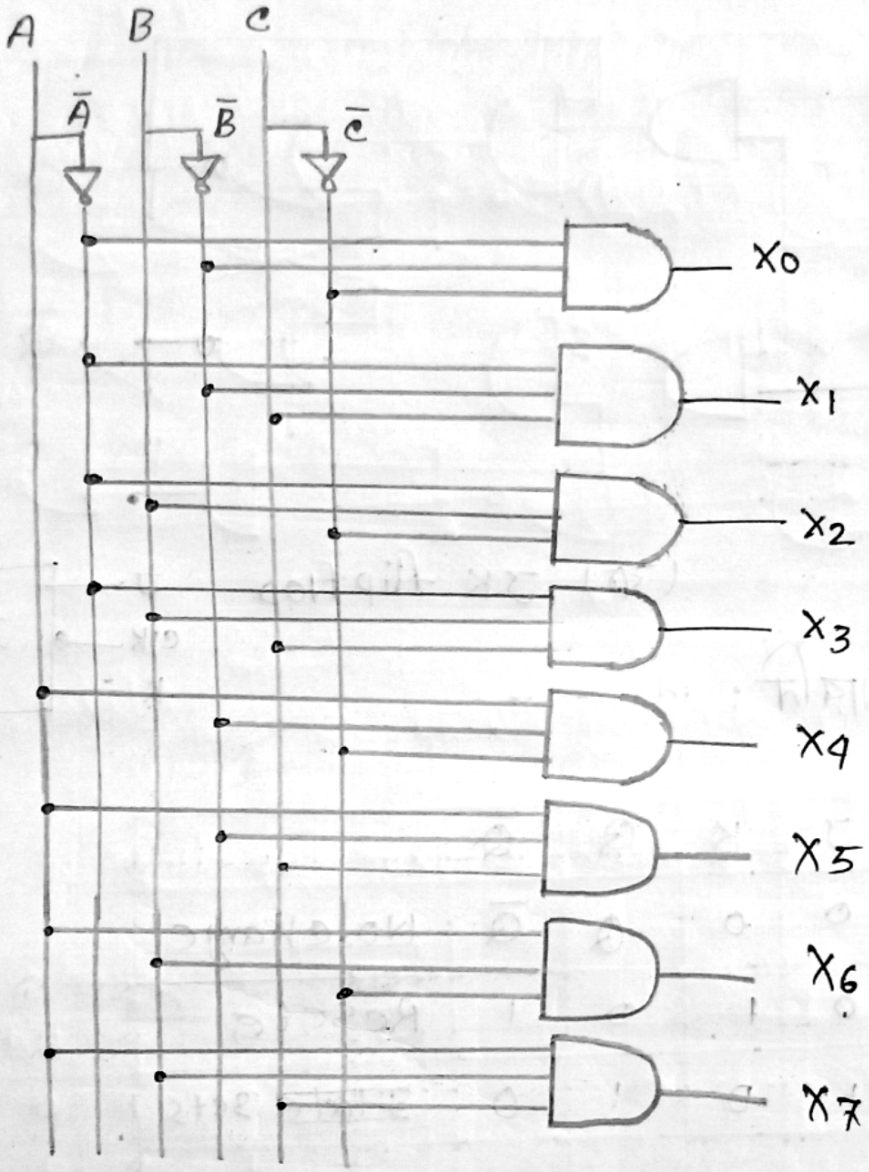
$X_3 = \bar{A}BC$ $\quad X_3 \text{ —} \bar{A}BC$

$X_4 = A\bar{B}\bar{C}$ $\quad X_4 \text{ —} A\bar{B}\bar{C}$

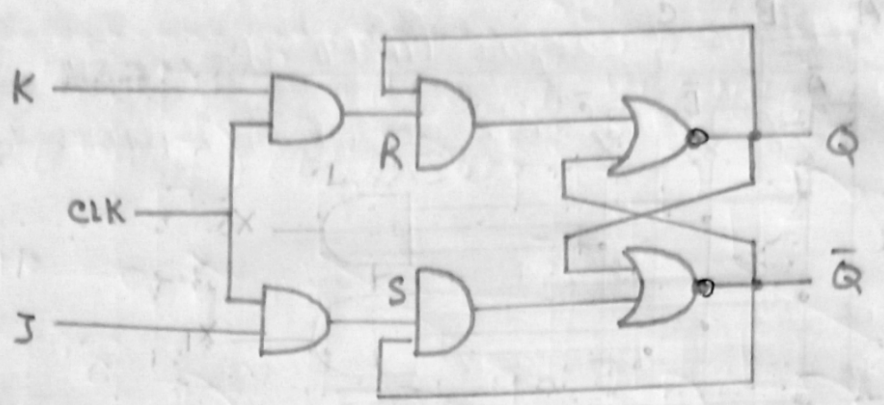
$X_5 = A\bar{B}C$ $\quad X_5 \text{ —} A\bar{B}C$

$X_6 = AB\bar{C}$ $\quad X_6 \text{ —} AB\bar{C}$

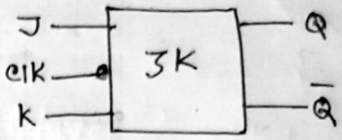
$X_7 = ABC$ $\quad X_7 \text{ —} ABC$


A
B
C
$\bar{A}$
$\bar{B}$
$\bar{C}$
X_0
X_1
X_2
X_3
X_4
X_5
X_6
X_7

JK ফ্লিপ-ফ্লপ :-



(ক) JK flip flop



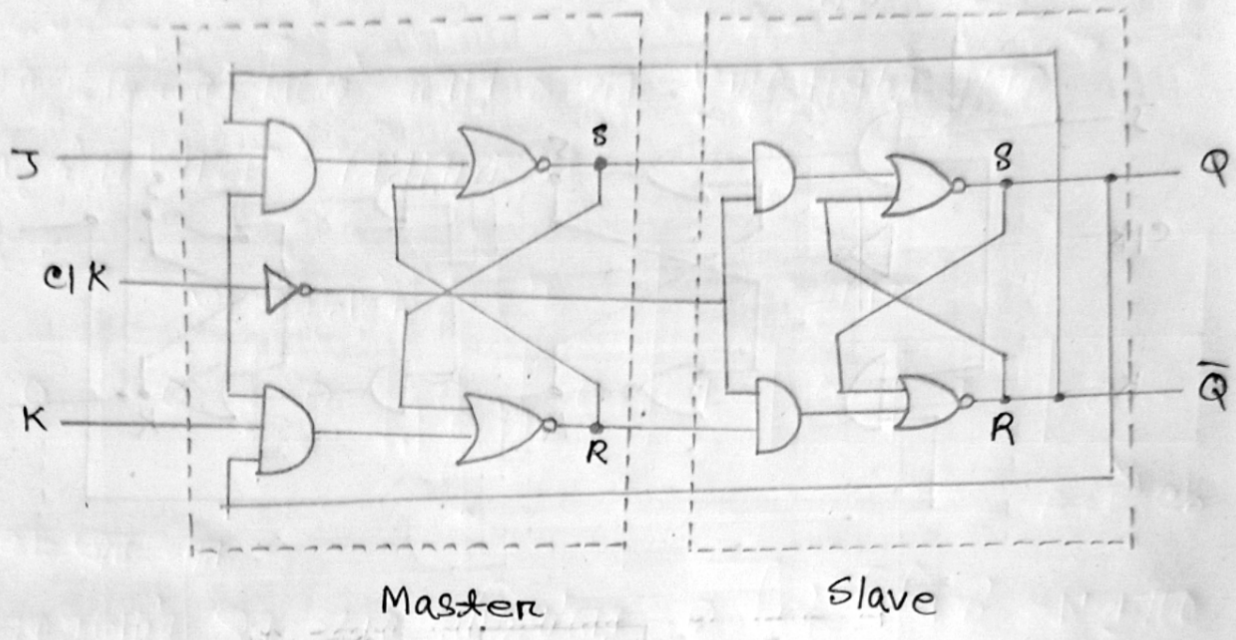
সত্যক সারনি :

CLK	J	K	Q	$\bar{Q}$	
1	0	0	Q	$\bar{Q}$	No change
1	0	1	0	1	Resete
1	1	0	1	0	~~State~~ sets
1	1	1	-	-	Toggles

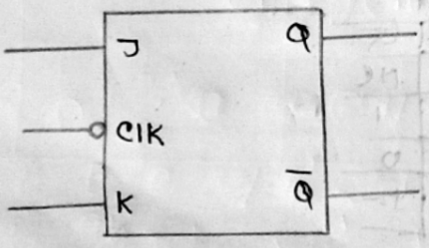
উপরের চিত্রে J-K flip-flop এর সার্কিট ডায়াগ্রাম দেখানো হয়েছে, এর দুইটি input J ও K । J এবং $\bar{Q}$ কে NAND গেইট দিয়ে S এর সাথে এবং K এবং Q কে R এর এর সাথে NAND গেইট দিয়ে যুক্ত করা হয়েছে, এর ফলে J ও K একসাথে 1 প্রয়োগ করলে SR কখোনো একসাথে 1 হবে না, 1 থাকলে 0 হবে এবং 0 থাকলে 1 হবে।

এই অবস্থা কে toggle বলে।

JK মাস্টার স্লেভ ফ্লিপ-ফ্লপ (JK master-slave flip-plop):-



J-K master-slave flip-flop



প্রতীক

CLK	J	K	Q
⎍	0	0	NC
⎍	0	1	0
⎍	1	0	1
⎍	1	1	Toggle

সত্যক সারনি

দুটি CLK যুক্ত ফ্লিপ-ফ্লপ দিয়ে master-slave, flip-flop গঠন করা যায়। ফ্লিপ-ফ্লপ দুটির একটিকে master ও অপরটিকে slave বলা হয়। master এর output slave এর input হিসেবে কাজ করে, master এর CLK পালসটির এবং slave এর ক্লক নেগেটিভ, যখন CLK 1 অবস্থায় বিরাজ করে তখন মাস্টার সক্রিয় এবং স্লেভ নিষ্ক্রিয় থাকে, আবার CLK যখনন 0 অবস্থায় থাকে তখন মাস্টার নিষ্ক্রিয় ও স্লেভ সক্রিয় থাকে,

মনেকরি, $Q=0$ এবং $\bar{Q}=1$ এ অবস্থায় J-K input এ যথাক্রমে 1 এবং 0 প্রয়োগ করা হলো। তখনে. clk 1 হলে master এর আউটপুট $S=1$ ও $R=0$ হবে, এ সময় স্লেভ এর আউটপুট পরিবর্তন হবে না। আবার clk 0 হলে মাস্টার এর আউটপুট পরিবর্তন হবে না। কিন্তু slave এর আউটপুট $S=1$ ও $R=0$ ইনপুটের জন্য পরিবর্তন হয়ে $Q=1$ এবং $\bar{Q}=0$ হবে।

$J=K=0$ input এর জন্য আউটপুট $\bar{Q}=0$ এবং $Q=1$ হবে যদি $J=K=1$ হয় তখন আউটপুট toogle করবে।

clk	J	K	Q	$\bar{Q}$
⎍	0	0	1	0
⎍	0	1	1	0
⎍	1	0	0	1
⎍	1	1	toogle →	