

## BOOLEAN ALGEBRA FORMULA

### Distributive Law:

1.  $A+BC=(A+B)(A+C)$
2.  $A(B+C)=AB+AC$

### De-Morgan's law:

Break the bar change the sign








1.  $\overline{A+B} = \overline{A} \cdot \overline{B}$
2.  $\overline{AB} = \overline{A} + \overline{B}$

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

### Various kinds of operation:

AND	OR	NOT
$A \cdot 0 = 0$	$A + 0 = A$	$\overline{\overline{A}} = A$
$A \cdot 1 = A$	$A + 1 = 1$	
$A \cdot A = A$	$A + A = A$	
$A \cdot \overline{A} = 0$	$A + \overline{A} = 1$	

### Various kind of gate & their equation:

NAME	SYMBOL	OPERATION	TRUTH TABLE															
AND GATE		$F = A.B$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	0	0	1	0	1	0	0	1	1	1
A	B	F																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR GATE		$F = A + B$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	1
A	B	F																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
NOT GATE		$F = \overline{A}$	<table><tr><th>A</th><th>F</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	F	0	1	1	0									
A	F																	
0	1																	
1	0																	
NAND GATE		$F = \overline{A.B}$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	1	0	1	1	1	0	1	1	1	0
A	B	F																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOR GATE		$F = \overline{A + B}$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	1	0	1	0	1	0	0	1	1	0
A	B	F																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
X-OR GATE		1) $Y = A.\overline{B} + \overline{A}.B$ $= (A \oplus B)$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	0
A	B	F																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
X-NOR GATE		1) $Y = A.B + \overline{A}.\overline{B}$ $= \overline{(A \oplus B)}$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	1	0	1	0	1	0	0	1	1	1
A	B	F																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

## Boolean Algebra

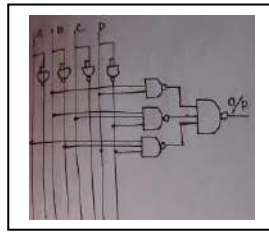
### 1. নিচের Boolean equation simplify করে logic circuit অংকন কর। [DUET]

- (i)  $Y = ABC'D' + A'BCD' + ABCD'$       Ans:  $Y = BD'$   
 $= ABD'(C + \overline{C}) + \overline{A}BD'(C + \overline{C}) = ABD' + \overline{A}BD' = BD'(A + \overline{A}) = BD'$  (Ans)
- (ii)  $Y = ((A+B'+C)+(B+C'))'$       Ans:  $Y=0$



$$=BD + BCD' + AB'D'$$

$$=BD \cdot BCD \cdot ABD$$



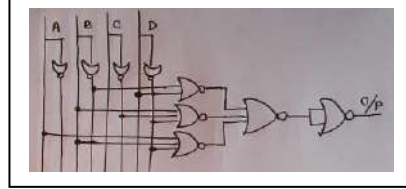
## 2. $Y=BD'+B'CD+A'B'D$ কে শুধুমাত্র NOR gate মাধ্যমে ডিজাইন কর।

$$=BD' + B'CD + A'B'D$$

$$= (\overline{BD}) \cdot (\overline{B'CD}) \cdot (\overline{A'B'D})$$

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

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



## 3. Combinational ও Sequential circuit এর মধ্যে পার্থক্য লিখ।

\*\*\*

Combinational	Sequential
১. যে সকল লজিক সার্কিটের আউটপুট ইহার ইনপুটের যে কোন মানের উপর নির্ভর করে, তাকে বলে	১. যে সকল লজিক সার্কিটের আউটপুট ইহার ইনপুটের তাত্ক্ষণিক মানের উপর নির্ভর করে এবং একই সাথে ইহার পূর্ববর্তী আউটপুটের উপর নির্ভর করে, তাকে বলে
২. উদাহরণ: Adder, Subtractor	2. Flip-flop, Register

## 4. দুটি Half adder ও একটি OR gate দ্বারা Full Adder Design কর।

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

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

$$= A \oplus B \oplus C$$

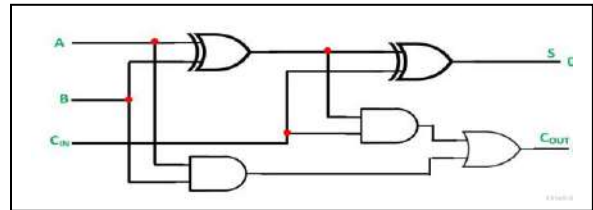
$$\text{Carry} = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

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

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

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

Input			Output	
A	B	Cin	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

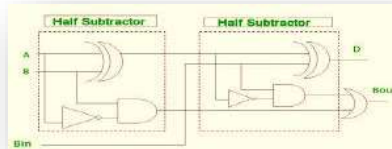


## 5. একটি Full adder ও একটি NOT gate দ্বারা Full subtractor Design কর।

$$\text{Difference} = \overline{A}BC + \overline{A}\overline{B}C + A\overline{B}\overline{C} + ABC = \overline{A}(\overline{B}C + B\overline{C}) + A(\overline{B}\overline{C} + BC) = A \oplus B \oplus C$$

$$\text{Borrow} = \overline{A}BC + \overline{A}\overline{B}C + A\overline{B}\overline{C} + ABC = C(\overline{A}B + A\overline{B}) + \overline{A}B(C + \overline{C}) = C(A \oplus B) + \overline{A}B$$

Input			Output	
A	B	C	Difference	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



## 6. NAND gate দ্বারা Full Adder Circuit Design কর।

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

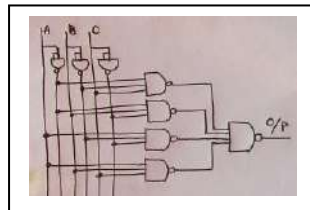
$$= \overline{A}BC + \overline{A}BC + \overline{A}BC + \overline{A}BC$$

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

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

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

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



## 7. NOR gate দ্বারা Full adder Circuit Design কর। [DUET]

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

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

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

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

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

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

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

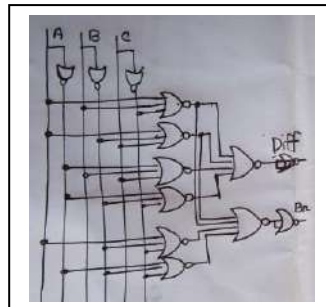
$$= \overline{A}BC + \overline{A}BC + \overline{A}BC + \overline{A}BC$$

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

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

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

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



### 8. NAND gate দ্বারা Full Subtractor Design কর।

$$\text{Difference} = \overline{A}BC + \overline{A}B\overline{C} + \overline{A}B\overline{C} + ABC$$

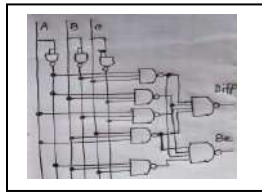
$$= \overline{A}BC + \overline{A}B\overline{C} + \overline{A}B\overline{C} + ABC$$

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

$$\text{Borrow} = \overline{A}BC + \overline{A}B\overline{C} + \overline{A}B\overline{C} + ABC$$

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

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



### 9. NOR gate দ্বারা Full Subtractor Design কর।

$$\text{Difference} = \overline{A}BC + \overline{A}B\overline{C} + \overline{A}B\overline{C} + ABC$$

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

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

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

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

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

$$\text{Borrow} = \overline{A}BC + \overline{A}B\overline{C} + \overline{A}B\overline{C} + ABC$$

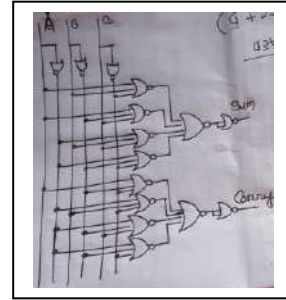
$$= \overline{A}BC + \overline{A}B\overline{C} + \overline{A}B\overline{C} + ABC$$

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

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

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

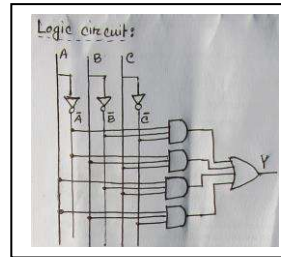
$$= (A + B + C)(A + \overline{B} + \overline{C})(\overline{A} + B + C)(\overline{A} + \overline{B} + \overline{C})$$



### 10. তিনটি ইনপুট বিশিষ্ট লজিক circuit design কর যার বিজোড় সংখ্যক ইনপুট হাই হলে আউটপুট লো হবে।

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

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



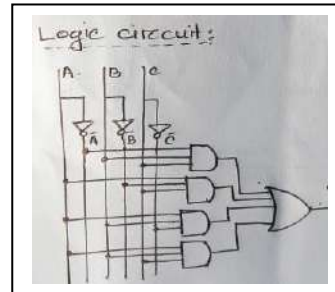
### 10. তিনটি ইনপুট বিশিষ্ট এমন একটি লজিক circuit design কর যার বিজোড় সংখ্যক ইনপুট হাই হলে আউটপুট হাই হবে।

(SELF STUDY)

### 11. তিনটি ইনপুট বিশিষ্ট লজিক circuit design কর, যার যেকোন দুটি ইনপুট লো হলে আউটপুট লো হবে। Boolean function টি লিখে সিম্পলিফাই করে logic circuit design কর।

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

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



## বিবিধ

### 19. Register, Latch, Flip flop কাকে বলে?

**Register:** কতগুলো ফ্লিপ ফ্লপ এ সমন্বয়ে গঠিত এমন একটি ইউনিট যা যা একত্রিভাবে একটি ইউনিটের মত কাজ করে এবং বাইনারি ওয়ার্ডকে ধারণ করতে পারে তাহাকে রেজিস্টার বলে।

**Flip flop:** ফ্লিপ ফ্লপ এমন একটি ইলেক্ট্রনিক সার্কিট যার মাত্র দুইটি অবস্থা রয়েছে।

**Latch circuit:** যে সার্কিট সেট রিসেট পদ্ধতি ডাটা জমা রাখে।

### 20. Toggle ও Race condition এর পার্থক্য লিখ।

Toggle	Race
১. Toggle means 0 to change 1 & 1 to change 0	১. Race means S,R=(1,1). যা অসম্ভব।
২. J-K flip flop এ ঘটে।	২. R-S flip flop এ ঘটে।

### 21. R-S, J-K flip flop truth table লিখ।

R-S flip-flop:

J-k flip flop:

S	R	Q
0	0	N.C
0	1	0
1	0	1
1	1	Race condition

J	K	Q
0	0	N.C
0	1	0
1	0	1

## 22. Multiplexer, De-multiplexer, Encoder, Decoder কাকে বলে এবং এদের ব্যবহার লিখ।

**Multiplexer:** যে সার্কিট এর একাধিক ইনপুট সিগন্যাল এবং একটি মাত্র আউটপুট সিগন্যাল থাকে, Multiplexer তাকে বলে।

**ব্যবহার:**

- একাধিক ইনপুট সিগন্যাল হতে নির্দিষ্ট আউটপুট সিগন্যালকে নির্বাচন করার জন্য ডাটা সিলেক্টর হিসেবে।
- সিদ্ধান্ত গ্রহণকারী ডিভাইস হিসেবে।
- বিভিন্ন ইলেক্ট্রনিক ও কম্পিউটার ডিভাইসে।

**De-multiplexer:** যে সার্কিট এর একটি ইনপুট সিগন্যাল এবং একাধিক আউটপুট সিগন্যাল থাকে, তাকে De-multiplexer বলে।

**ব্যবহার:**

- একটি ইনপুট সিগন্যাল হতে যথাযথ আউটপুট লাইনকে নির্বাচন করার জন্য ডাটা সিলেক্টর হিসেবে।
- সিদ্ধান্ত গ্রহণকারী ডিভাইস হিসেবে।
- বিভিন্ন ইলেক্ট্রনিক ও কম্পিউটার ডিভাইসে।

**Encoder:** যে সার্কিটের সাহায্যে Man to machine যোগাযোগ প্রতিষ্ঠা করা যায়, তাকে Encoder বলে।

**ব্যবহার:**

- ডেসিমাল থেকে বাইনারীতে রূপান্তর করার জন্য।
- ইনপুট ডিভাইসে ব্যবহৃত হয়।
- ফিজিক্যাল সিগন্যালকে মেশিন কোডে রূপান্তর করতে

**Decoder:** যে সার্কিটের সাহায্যে Machine to man যোগাযোগ প্রতিষ্ঠা করা যায়, তাকে Decoder বলে।

**ব্যবহার:**

- বাইনারী থেকে ডেসিমাল এ রূপান্তর করার জন্য।
- আউটপুট ডিভাইসে ব্যবহৃত হয়।
- মেশিন কোডকে ফিজিক্যাল সিগন্যাল এ রূপান্তর করতে

## 23. লজিক ফ্যামিলির সংজ্ঞা ও বৈশিষ্ট্য গুলোর সংজ্ঞা লিখ।

**Logic family:** ডিজিটাল সিস্টেম এ বিভিন্ন আইসি গুলোর সমন্বয়ে লজিক ফ্যামিলি বলে।

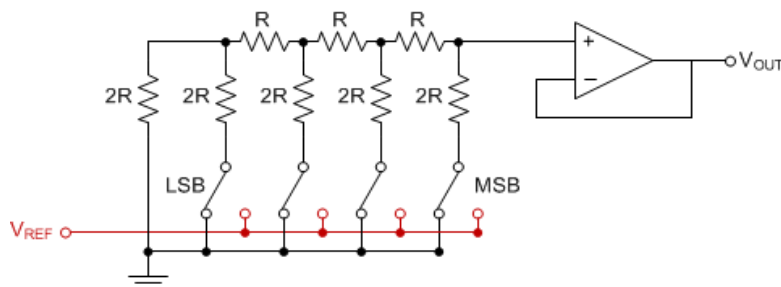
**বৈশিষ্ট্য:**

- Fan in :** যতগুলো ইনপুট সুষ্ঠুভাবে পরিচালনা করতে পারে তাকে ফ্যান ইন বলে।
- Fan out:** যতগুলো আউটপুট সুষ্ঠুভাবে পরিচালনা করতে পারে তাকে ফ্যান ইন বলে।
- Noise margin:** যে পরিমাণ নয়েজ থাকলে লজিক গেইট সুষ্ঠুভাবে আউটপুট প্রদান করতে পারে, তাকে Noise margin বলে।
- Propagation delay:** ইনপুট দেওয়ার পর আউটপুট পেতে যে সময়ের প্রয়োজন হয়, তাকে Propagation delay বলে।
- Power dissipation:** ইনপুট দেওয়ার পর যে পাওয়ার অপচয় হয়, তাকে Power dissipation বলে।
- Operating frequency:** যে ফ্রিকুয়েন্সি পর্যন্ত উহার বৈশিষ্ট্য অক্ষুর রেখে পরিচালনা করতে পারে, তাকে Operating frequency বলে।

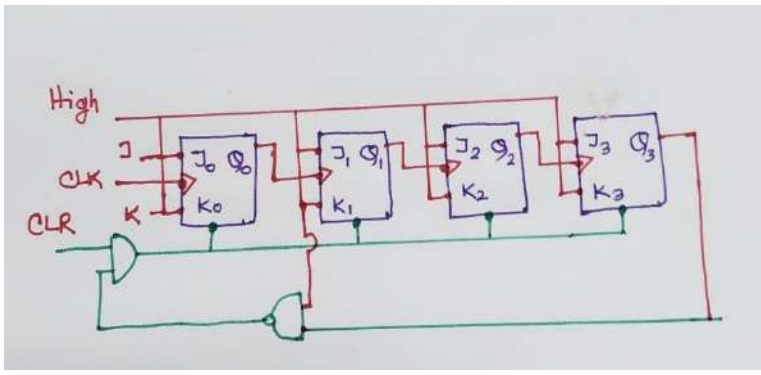
## 24. সিনক্রোনাস ও অ্যাসিনক্রোনাস কাউন্টারের পার্থক্য লিখ। \*\*\*

সিনক্রোনাস	অ্যাসিনক্রোনাস
১. যে কাউন্টার এ সকল ফ্লিপ-ফ্লপ কে একসাথে ক্লক ইনপুট দ্বারা ট্রিগার করা হয়, তাকে সিনক্রোনাস বলে।	১. যে কাউন্টার এ একটি ফ্লিপ-ফ্লপ এর আউটপুট পরবর্তী ফ্লিপ ফ্লপ ক্লকের জন্য ব্যবহৃত হয়, তাকে সিনক্রোনাস বলে।
২. প্রোপাগেশন ডিলে কম।	২. প্রোপাগেশন ডিলে বেশী।
৩. ইহাকে প্যারালাল কাউন্টার বলে।	৩.

## 25. R-2R ladder Design কর।

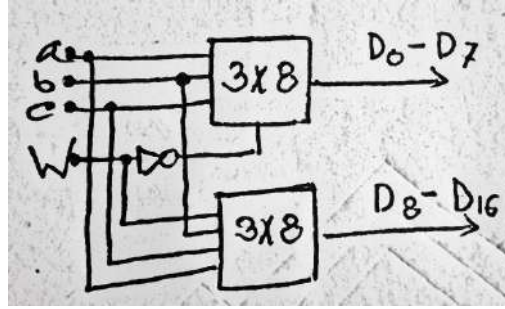


## 26. MOD-9 Counter/Programmable/Decode/Ripple/Frequency Asynchronous Counter Design কর।



2<sup>n</sup> সূত্র ব্যবহার করে ট্রিপ ফ্লপ হিসাব করা হয়। যেহেতু মোড ৭, তাই  $2^4 = 16$ । তাই ৪ টি ট্রিপ ফ্লপ ব্যবহার করা হয়েছে। যদি মোড ১৭ কাউন্টার ডিজাইন করতে বলে তাহলে তোমরাই হিসাব কর কয়টা কাউন্টার লাগবে?

27. ৩x৮ ডিকোডার ব্যবহার করে ৪ x ১৬ ডিকোডার ডিজাইন কর।



রচনা ও সম্পাদনায়:

আল ফয়সাল রাব্বী, ডুয়েট, সি.এস.ই

Special Thanks to  
Meson chakma  
EEE, DUET