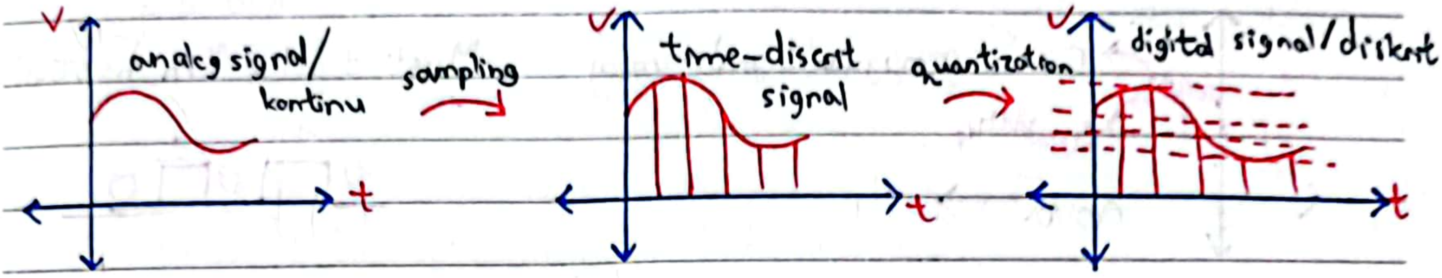
Digital systems can represent and manipulate discrete elements of information. Discrete elements of information are represented in a digital system by physical quantities called signals.



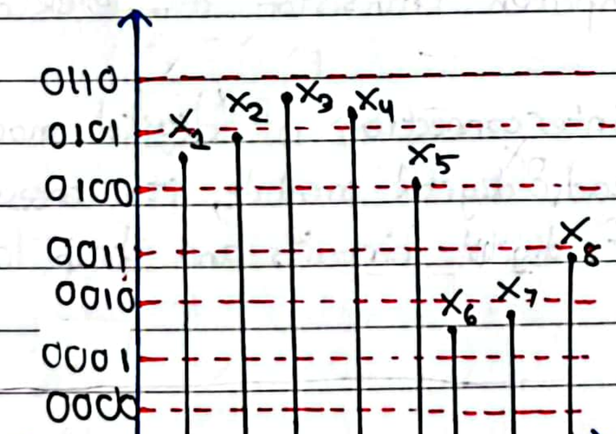
ADC = Analog to Digital Converter
DAC = Digital to Analog Converter

ADC adalah merubah signal/data kontinyu terhadap waktu menjadi sinyal/data diskrit terhadap waktu.

Proses sampling / pencuplikan yaitu proses pengambilan data diskrit dari data kontinyu.

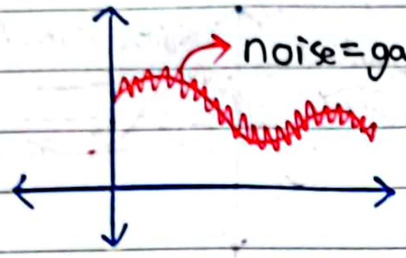
Teknologi sehebat apapun akan kembali ke alam melalui ADC/DAC.

Proses kuantifisasi yaitu proses pengelompokkan data diskrit.

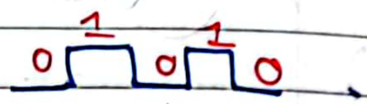


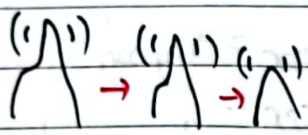
0011 = x_6, x_7
0100 = x_8
0101 = x_1, x_3, x_4
0110 = x_2, x_3

Why digital signal?
- Can convey information with less noise, distortion, and interferen.
- More flexible, secure, accurate



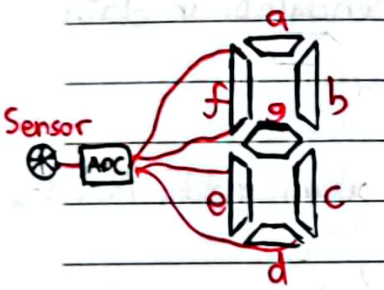
Must digital krn bs stabil





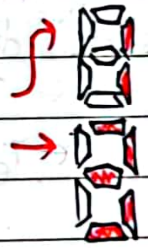
= Interferensi

Analog Input → ADC → Digital Systems → DAC → Analog output
↳ Sensor ↳ Sbg modul ↳ Sbg proses & butuh program utk memproses ↳ Sbg modul
Cth: Kelembapan, Suhu, Gerak



: 7 Segments

a	b	c	d	e	f	g	
0	1	1	0	0	0	0	= 1
1	1	1	1	0	0	1	= 3



Kode biner ada di komponen transistor dlm elektronika

A digital system is an interconnection of digital modules. To understand the operation of each digital module, it's necessary to have a basic knowledge of digital circuits and their logical function.

Name	Radix	Digits
Binary	2	0,1
Octal	8	0,1,2,3,4,5,6,7
Decimal	10	0,1,2,3,4,5,6,7,8,9
Hexadecimal	16	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Number System and Codes

- Weighted → There is weight in the position
  - e.g. : Decimal, binary, octal, hexadecimal
- Unweighted
  - e.g. : Gray code

$56_{(10)}$: $111000_{(2)}$

↳ 2 | 56 — 0   MSB/MSD

2 | 28 — 0

2 | 14 — 0

2 | 7 — 1

2 | 3 — 1

1

LSD/LSB → Least Significant Bit / Decimal

Most Significant Bit/Digit

Untuk membuat Gray Kode, memerlukan gerbang ex-or.

22,5°

→ Gray Kode

Tabel Kebenaran ex-or (⊕) :

A	B (Input)	X (output)
0	0	0
0	1	1
1	0	1
1	1	0

	MSB	⊕	⊕	⊕ LSB
Biner	0 →	0 →	1 →	0
Gray	0	0	1	1

Worry is like a down payment to problem you may never had.

kuncimatahari.com

Mesin gerak

0011 → Kl ketemu di mesin → 0110

menggunakan gray code krn lbh efektif & cepat

Biner MSB 0 1 1 1 LSB

Gray 0 1 0 0

- $235_{(10)} = 2\cdot10^2 + 3\cdot10^1 + 5\cdot10^0 = 235_{(10)}$
  - ↳ Contoh Decimal ke Decimal

- $11010.11_{(2)} = 1\cdot2^4 + 1\cdot2^3 + 0\cdot2^2 + 1\cdot2^1 + 0\cdot2^0 + 1\cdot2^{-1} + 1\cdot2^{-2}$
  $= 16 + 8 + 0 + 2 + 0 + 0,5 + 0,25$
  $= 26 + 0,75$
  $= 26,75_{(10)}$
  - ↳ Contoh Biner ke Decimal

- $26,75_{(10)} = 11010.11_{(2)}$
  - ↳ 26 : 2 = 13 sisa 0 → LSB
    13 : 2 = 6 sisa 1
    6 : 2 = 3 sisa 0
    3 : 2 = 1 sisa 1
    1 → MSB
  - ↳ $0,75 \cdot 2 = 1,5 \rightarrow 1 + 0,5$ → LSB
    $0,5 \cdot 2 = 1 \rightarrow 1 + 0$ → MSB
    $0 \cdot 2 = 0$
  - ↳ Contoh Decimal ke Biner

- $127,4_{(8)} = 87,5_{(10)}$
  - ↳ $\overset{2}{1}\overset{1}{2}\overset{0}{7},\overset{-1}{4} = 1\cdot8^2 + 2\cdot8^1 + 7\cdot8^0 + 4\cdot8^{-1}$
    $= 1\cdot64 + 2\cdot8 + 7\cdot1 + 0,5$
    $= 87 + 0,5$
    $= 87,5_{(10)}$
  - ↳ Contoh Octal ke Decimal

- Oktal ⟷ Heksa
  Oktal ⟷ Biner
  Heksa ⟷ Biner

- $B\,6\,5\,F_{(16)} = 46687_{(10)}$

B	6	5	F
11	2	1	0

$$= 11 \cdot 16^3 + 6 \cdot 16^2 + 5 \cdot 16^1 + 15 \cdot 16^0$$
$$= 45056 + 1536 + 80 + 15$$
$$= 46687_{(10)}$$

↳ Contoh Hexadecimal ke Decimal

- $41_{(10)} = 101001_{(2)}$

2^5	2^4	2^3	2^2	2^1	2^0
32	16	8	4	2	1
1	0	1	0	0	1

↳ Contoh Decimal ke Biner

- $1001_{(10)} = 1111101001_{(2)}$

2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
512	256	128	64	32	16	8	4	2	1
1	1	1	1	1	0	1	0	0	1

↳ Contoh Decimal ke Biner

- $45_{(10)} = 2D_{(16)}$

128	64	32	16	8	4	2	1
8	4	2	1	8	4	2	1
0	0	1	0	1	1	0	1

2

$8 + 4 + 1 = 15 = D$

↳ Contoh Decimal ke Hexadecimal

- $3912_{(10)} = 7510_{(8)}$

2048	1024	512	256	128	64	32	16	8	4	2	1
4	2	1	4	2	1	4	2	1	4	2	1
1	1	1	1	0	1	0	0	1	0	0	0

$4+2+1=7$ $\quad 4+1=5 \quad 1 \quad 0$

↳ Contoh Decimal ke Octal

- $3912_{(10)} = F48_{(16)}$

↳

2048	1024	512	256	128	64	32	16	8	4	2	1
8	4	2	1	8	4	2	1	8	4	2	1
1	1	1	1	0	1	0	0	1	0	0	0
8+4+2+1 = 15 = F				4				8			

↳ Contoh Decimal ke Hexadecimal

- $010\ 110.110_{(2)} = 26,6_{(8)}$

↳ 010 | 110. | 110

010	110.	110
4 2 1	4 2 1	4 2 1
2	4+2=6	4+2=6

↳ Contoh Biner ke Octal

- $37,45_{(8)} = 011\ 111.100101_{(2)}$

↳ 3 7 . 4 5

2+1 4+2+1 (4) 4+1

011 111 . 100 101

↳ Contoh Octal ke Biner

- $0001\ 1000\ 1001.\ 1100_{(2)} = 189.C_{(16)}$

↳ 0001	1000	1001.	1100
8 4 2 1	8 4 2 1	8 4 2 1	8 4 2 1
1	8	8+1=9	8+4=12=C

↳ Contoh Biner ke Hexadecimal

- $254_{(8)} = 010\ 101\ 100_{(2)}$

↳ 2 5 4

2 4+1 4

010 101 100

↳ Contoh Octal ke Biner

- $(CAD)_{16} = CAD_{(16)} = 110\ 010\ 101\ 101_{(2)} = 6255_{(8)}$

110	010	101	101
4 2 1	4 2 1	4 2 1	4 2 1
4+2=6	2	4+1=5	4+1=5

↳ Contoh Hexadecimal ke Biner ke Octal

Arithmetic Number Systems

- Decimal :

$$\begin{array}{r} 25 \\ 47 + \\ \hline 72 \end{array} \qquad \begin{array}{r} 125 \\ 72 - \\ \hline 53 \end{array}$$

- Biner :

$$\begin{array}{r} 110 \\ 101 + \\ \hline 1011 \end{array} \qquad \begin{array}{r} 11011 \\ 10110 - \\ \hline 00101 \end{array}$$

∘) 0 + 0 = 0
1 + 0 = 0 + 1 = 1
1 + 1 = 0 → Carry 1

$$\begin{array}{r} 111 \\ 111 + \\ \hline 1110 \end{array} \qquad \begin{array}{rl} 101000 & = 40_{(8)} \\ 100100 & = 36_{(8)} - \\ \hline 000100 & = 4 \end{array}$$

∘) 0 − 0 = 0 = 1 − 1
0 − 1 = 0 → Borrow 1
1 − 0 = 1

- Octal :

$$\begin{array}{r} {}^{1}5\,{}^{1}6\,7 \\ 2\,4\,3 + \\ \hline 10\;3\;2 \end{array}$$

→ 7+3−8
6+4+1−8
5+2+1−8
1

$$\begin{array}{r} 7\,{}^{-1}4\,{}^{1}3 \\ 5\,6\,4 - \\ \hline 1\;5\;7 \end{array}$$

→ 8+3−4
8 +3−6

- Hexadecimal :

$$\begin{array}{r} 5\;6\;8\;9 \\ 4\;5\;7\;4 + \\ \hline 9\;B\;F\;D \end{array}$$

$$\begin{array}{r} {}^{1}A\,{}^{1}D\,{}^{1}D \\ D\,A\,D + \\ \hline {}^{1}18\;8\;A \end{array}$$

→ D + D = 13+13 = 26 − 16 = 10 = A
13+10+1−16
10+13+1 = 24 − 16 = 8

$$\begin{array}{r} 9\,{}^{-1}7\,{}^{-1}4\,{}^{-1}B \\ 5\;8\;7\;C - \\ \hline 3\;E\;C\;F \end{array}$$

→ 16 + B − C = 16+11−12 = 15 = F
16+3 −7 = 12 = C
16+6 −8 = 14 = E

DATE PAGE

Sistem komplemen :

digunakan oleh komputer untuk mengoperasikan arithmetic

1) Komplemen-1
- Akan mengubah nilai 0 menjadi 1, atau
- Akan mengubah nilai 1 menjadi 0

2) Komplemen-2
- Hasil komplemen-1 ditambah dengan 1

yg biasa dibaca oleh komputer

Contoh Komplemen-1 :

- 11001
  10110 − $\xrightarrow{\text{Komp.-1}}$ 01001

```
      11001
      10110 -
carry1 11001
      01001 +
carry1 00010
          1 +
      00011
```

Contoh Komplemen-2 :

```
  11001
- 10110 +  →Komp.2→  01001
                        1 +  → Komplemen-2
                     01010
                     11001
           carry ← (1)00011
```

atau

```
   11001
   01010 +  →Komp-2
 (1)00011
  carry
```

BCD / Binary Code Decimals → hrs berbasis 4 / 4 byte

```
o 0100 1001 0101     ↔   495
  0010 0111 1000 +       278 +
  ----------------       -----
  0110 10000 1101        773
       K     K ← Kurangi
       0110  0110
       -----------
       10110 10011 → BCD
     1↵    1↵
  ----------------
  0111 0111 0011
   7    7    3
```

Jika penjumlahan melebihi dari 10 (>10), maka harus dikoreksi dengan menambahkan 0110.

```
  o 0010 0111 1000       495  ↔  0100 1001 0101
Komp-3 { 1101 1000 0111 -Komp-1  278 -  0111 0010 0001 +
    1010 1010 1010 +1010  ----   ---------------
    ----------------      217    1011 1011 0110
    10111 10010 10001            0110 0110
      ↓     ↓     ↓              ---------------
    0111  0010  0001             10001 10001 0110
                                        1↵       1
                                 ---------------
                                 0010  0001  0111
```

Untuk menjadikan komp-3, bilangan pengurang di komp.-1 kemudian ditambahkan 10 (1010).

Bilangan Bertanda

- **Bilangan Tidak Bertanda / Unsigned**

b_{n-1}	b_{10}	b_9	oooooo	b_2	b_1	b_0

Magnitude

- **Bilangan Bertanda**

S	b_{n-1}	b_9	oooooo	b_2	b_1	b_0

Sign: − '1', + '0'   Magnitude

- **Sign Magnitude**
  - Bilangan sign magnitude menggunakan bit paling kiri (MSB) untuk menyatakan tanda, sisanya sebagai nilai besarnya magnitude.
  -

	0	1	2	3	....
Positif	0000	0001	0010	0011	...
Negatif	1000	1001	1010	1011	...

Bilangan bertanda ada:
- Sign magnitude
- 1' complement
- 2' complement

- **1' Complement**
  - Akan merubah bit magnitude juga bit bertanda / sign.
    - '0' menjadi '1'
    - '1' menjadi '0'

DATE . . PAGE

	0	1	2	3	...
Positif	0000	0001	0010	0011	....
Negatif	1111	1110	1101	1100	....

Contoh : 16 8 4 2 1
1 0 1 0 0

0 1 1 0 0 → Ktmu 1 plg kiri baru berubah

- 2' Complement
  - Mengkomplemenkan bilangan tiap bit kemudian ditambah dengan '1'
  - $\boxed{\text{2' Complement} = \text{1' Complement} + 1}$

	0	1	2	3	....
Positif	0000	0001	0010	0011	....
Negatif	10000	1111	1110	1101	....

Contoh : Kurangkan nilai 5 dengan -2
x -y → x+(-y)

0101 (5)
1010 + (-2)
1111 (Salah)

→

0101 (5)
1010 + (-2)
Komp-1 0101
0101 +
1010
1
011

Worry is like a down payment to problem you may never had. kuncimatahari.com

- SOP (Sum Of Product)

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

OR
- lambang concnical (m) - Minform
- lambang persamaan Σ
- bilangan dasar adlh 1
- inversnya adlh 0

A \ B	0	1
0	1	1
1		

$F(A,B) = \Sigma m(0,1)$
$= \Sigma(0,1)$

$f(A,B,C) = \Sigma(0,1,3,4,6,7)$

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

- POS (Product Of Sum)

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

AND
- lambang canonical (M)
- lambang perkalian Π
- bilangan dasar 0
- inversnya adlh 1

kuncimatahari.com

Person who chases two rabbits catches neither.

Take Home 1

1. Konversikan K-Map berikut $F(A,B,C,D) = \Sigma m(5,7,10,8,11,12,13,14,15)$

→ c)

AB \ CD	00	01	11	10
00	0_0	0_1	0_3	0_2
01	0_4	1_5	1_7	0_6
11	1_{12}	1_{13}	1_{15}	1_{14}
10	1_8	0_9	1_{11}	1_{10}

III

c). $F = \text{I} + \text{II} + \text{III}$

$= BD + AC + AD'$

$= BD + A(C + D')$

2. Konversikan K-Map berikut $F(A,B,C,D) = \Sigma m(5,6,9,8,13,15) + d(1,7,14)$

→ c)

AB \ CD	00	01	11	10
00	0_0	d_1	0_3	0_2
01	0_4	1_5	d_7	1_6
11	0_{12}	1_{13}	1_{15}	d_{14}
10	1_8	1_9	0_{11}	0_{10}

c). $F = \text{I} + \text{II} + \text{III}$

$= BC + C'D + AB'C'$

$= BC + C'(D + AB')$

3. Sederhanakan persamaan K-Map berikut, $F(A,B,C,D,E) = \Sigma m(0,2,4,6,9,13,21,23,25,29,31) + d(1,5,7,10,14,17)$

→ c) A

BC \ DE	00	01	11	10
00	0_0	d_1	0_3	0_2
01	0_4	1_5	1_7	0_6
11	0_{12}	1_{13}	1_{15}	0_{14}
10	0_8	1_9	0_{11}	0_{10}

c) A'

BC \ DE	00	01	11	10
00	1_0	d_1	0_3	1_2
01	1_4	d_5	d_7	1_6
11	0_{12}	1_{13}	0_{15}	d_{14}
10	0_8	1_9	0_{11}	d_{10}

c). $F = \text{I} + \text{II} + \text{III} + \text{IV} + \text{V}$

$= A'B'D' + D'E + ACE + A'B'C + A'DE'$

$= A'B'(D' + C) + E(D' + AC) + A'DE'$

Worry is like a down payment to problem you may never had.

kuncimatahari.com

4. Sederhanakan fungsi Boolean berikut dengan menggunakan metode tabulasi $F(A,B,C,D) = \Sigma m\,(1,4,6,7,8,9,10,11,15)$

a) 1 = 0001 b) 7 = 0111 c) 10 = 1010
4 = 0100 8 = 1000 11 = 1011
6 = 0110 9 = 1001 15 = 1111

b)

Group	Minterm	A B C D		A B C D
1	m_1	0 0 0 1 ✓	$m_1 \to m_4$	- 0 0 1
	m_4	0 1 0 0 ✓	$m_4 \to m_6$	0 1 - 0
	m_8	1 0 0 0 ✓	$m_8 \to m_9$	1 0 0 -
2	m_6	0 1 1 0 ✓	$m_9 \to m_{10}$	1 0 - 0
	m_9	1 0 0 1 ✓	$m_6 \to m_7$	0 1 1 -
	m_{10}	1 0 1 0 ✓	$m_9 \to m_{10}$	1 0 - 1
3	m_7	0 1 1 1 ✓	$m_{10} \to m_7$	1 0 1 -
	m_{11}	1 0 1 1 ✓	$m_7 \to m_{15}$	- 1 1 1
4	m_{15}	1 1 1 1 ✓	$m_{11} \to m_{15}$	1 - 1 1

c)

1	4	6	7	8	9	10	11	15	
✓					✓				1,9
	✓	✓							4,6
		✓	✓						6,7
			✓					✓	7,15
							✓	✓	11,15
				✓	✓	✓	✓		8,10,9,11

d) $F = B'C'D + A'BD' + AB' + BCD$

5. Sederhanakan fungsi Boolean berikut dengan menggunakan metode tabulasi $F(A,B,C,D) = \Sigma m\,(2,6,7,8,9,13,15)$

→ ◦) 2 = 0010 ◦) 8 = 1000 ◦) 13 = 1101

6 = 0110 9 = 1001 15 = 1111

7 = 0111

◦) 2 0010 ✓

8 1000 ✓

6 0110 ✓

9 1001 ✓

7 0111 ✓

13 1101 ✓

15 1111 ✓

	AB CD	2	6	7	8	9	13	15
2→6	0 – 1 0	✓	✓					
8→9	1 0 0 –				✓	✓		
6→7	0 1 1 –		✓	✓				
9→13	1 – 0 1					✓	✓	
7→15	– 1 1 1			✓				✓
13→15	1 1 – 1						✓	✓

◦)◦ $F = A'CD' + AB'C' + BCD + ABD$

Nama : Jackson Lawrence

NIM : 00000070612

Kelas : CE 232 -D

Worry is like a down payment to problem you may never had.

kuncimatahari.com

DATE Week 10 PAGE

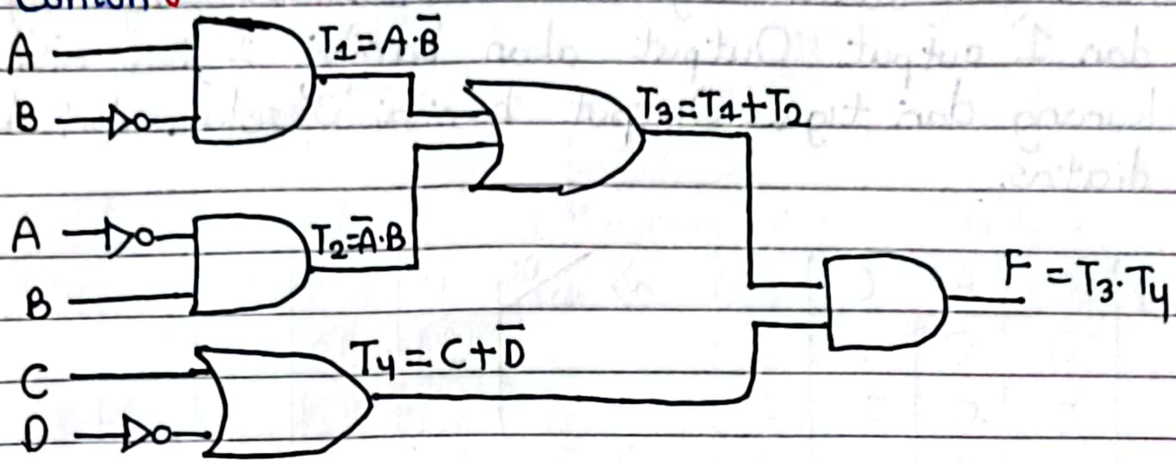
- Rangkaian Logika
  - Rangkaian Kombinasional → Rangkaian yang memerlukan input saat ini sehingga tidak memerlukan memori.
  - Rangkaian Sekuensial → Rangkaian yang memerlukan input saat ini dan output sebelumnya. Sehingga rangkaian ini memerlukan memori.
    - Rang. Sinkron
    - Rang. Asinkron
      - Rang. Fundamental
      - Pulse mode

- Rangkaian Kombinasional
  - Aritmatika dan fungsi logika
    - ↳ Adder < Half adder, Full adder
    - ↳ Subtractor < Half subtractor, Full subtractor
    - ↳ Comparator
  - Transmisi data
    - ↳ Multiplexer (Mux)
    - ↳ Demultiplexer (DMux)
    - ↳ Encoder
    - ↳ Decoder
  - Konversi kode
    - ↳ BCD
    - ↳ 7 Segment
    - ↳ Binary

- Analisa logika rangkaian kombinasional :
  1) Berilah label sembarang di setiap output gate
  2) Tentukan fungsi Boolean dari gate-gate tersebut
  3) Ulangi langkah 2) sampai ditemukan output dari rangkaian
  4) Dengan substitusi berulang, tentukan output dari variabel input yang dinyatakan sebelumnya.

kuncimatahari.com
Person who chases two rabbits catches neither.

Contoh :



- Perancangan rangkaian kombinasional :
  1) Dari rangkaian yang ada, tentukan input dan output
  2) Buatlah tabel kebenaran, berdasarkan input/output yang ada
  3) Susun fungsi Boolean untuk setiap output dengan menggunakan K-map
  4) Gambar logik diagramnya

Contoh Soal :

1. Suatu sistem sirkulasi udara bekerja. Untuk menggerakkan kipas angin dalam ruangan. Kipas angin akan bekerja ("1"), jika kondisi a. dan b. terpenuhi :
   a. Temperature lebih besar dari 28°C ("1")
   b. Jendela tertutup ("0") atau pintu tertutup ("0") (Minimal salah satu tertutup)

→ a)

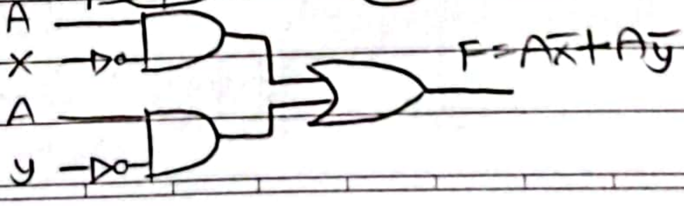
A	x	y	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

b)

A \ xy	00	01	11	10
0	0	0	0	0
1	1	1	0	1

→ $F = A\overline{x} + A\overline{y}$

c)



2. Rancanglah suatu rangkaian kombinasional dengan tiga input dan 1 output. Output akan bernilai '1' jika nilai desimalnya kurang dari tiga. Output bernilai '0' selain dari ketentuan diatas.

→ a)

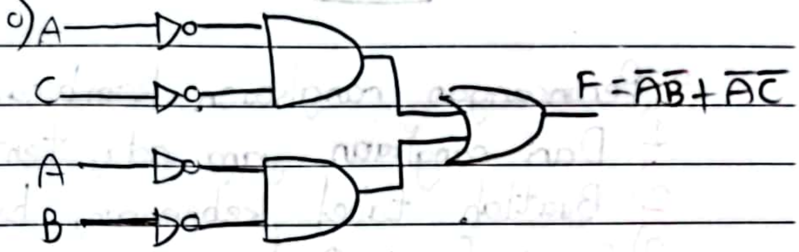
A	B	C	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

b)

A \ BC	00	01	11	10
0	1	1		$\boxed{1}$
1				

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

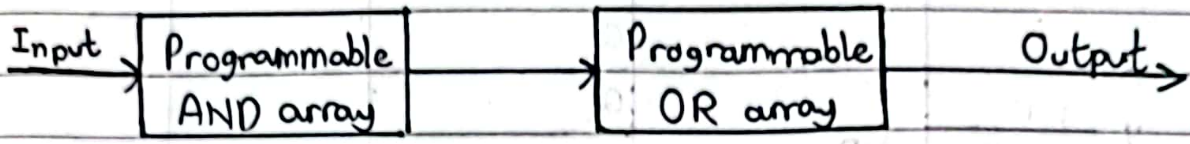
d)



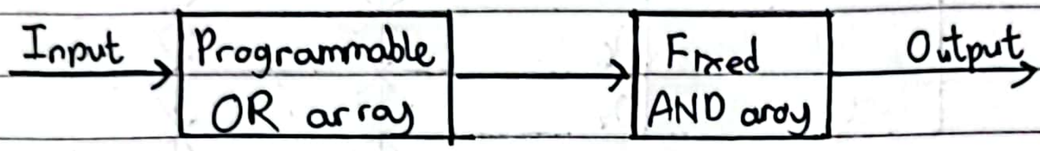
kuncimatahari.com
Person who chases two rabbits catches neither.

PLD (Programmable Logic Devices) ada 3 yaitu :

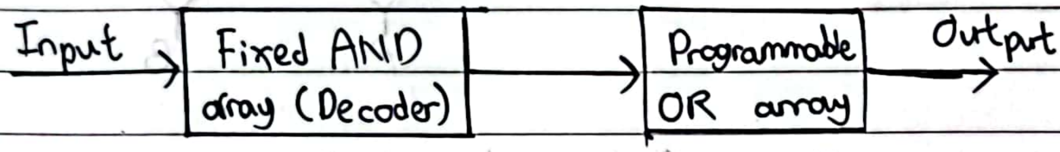
1. PLA / Programmable Logic Array



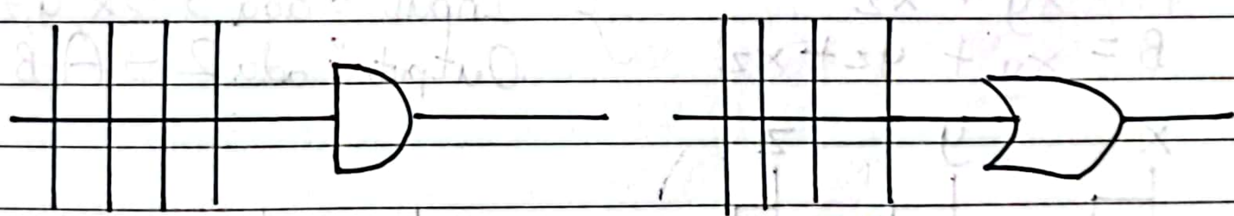
2. PAL/Programmable Array Logic



3. PROM / Programmable Read Only Memory



Simbol array AND dan OR yaitu :



AND array OR array

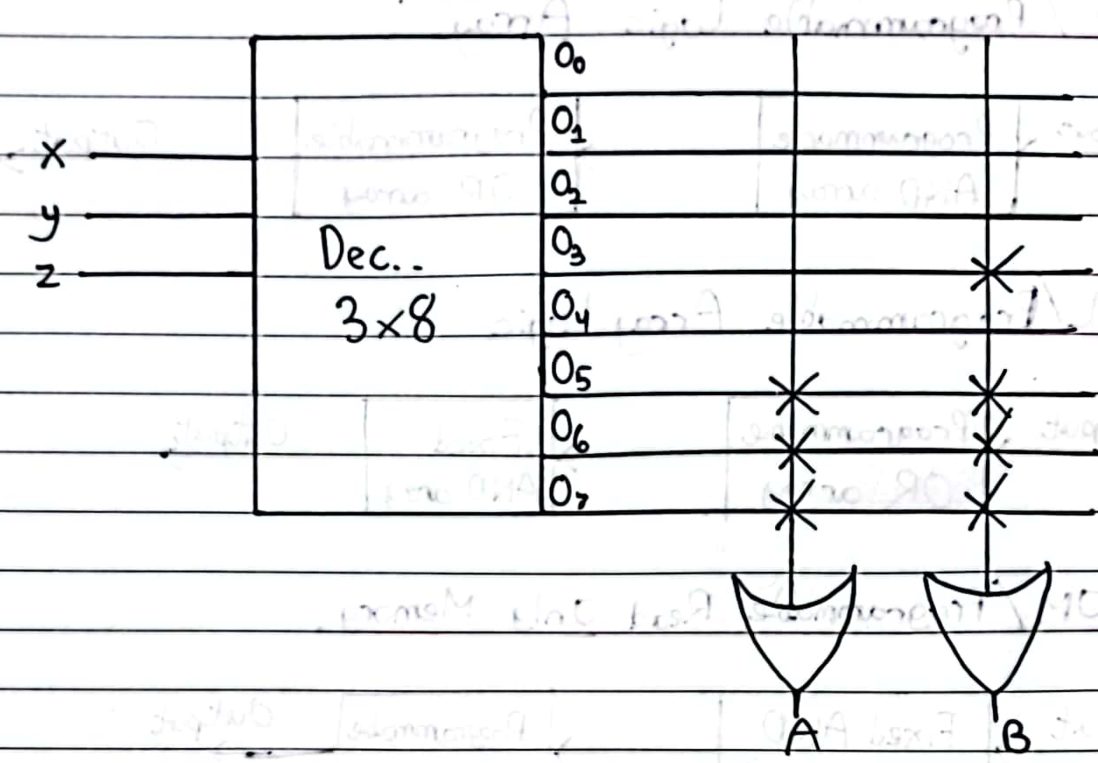
A. Implementasi Fungsi Boolean dengan Menggunakan PROM

- Decoder mempunyai banyak input dan output
  Input < Output
  $n < 2^n$
  $3 < 2^3$
  $3 < 8$

Worry is like a down payment to problem you may never had.

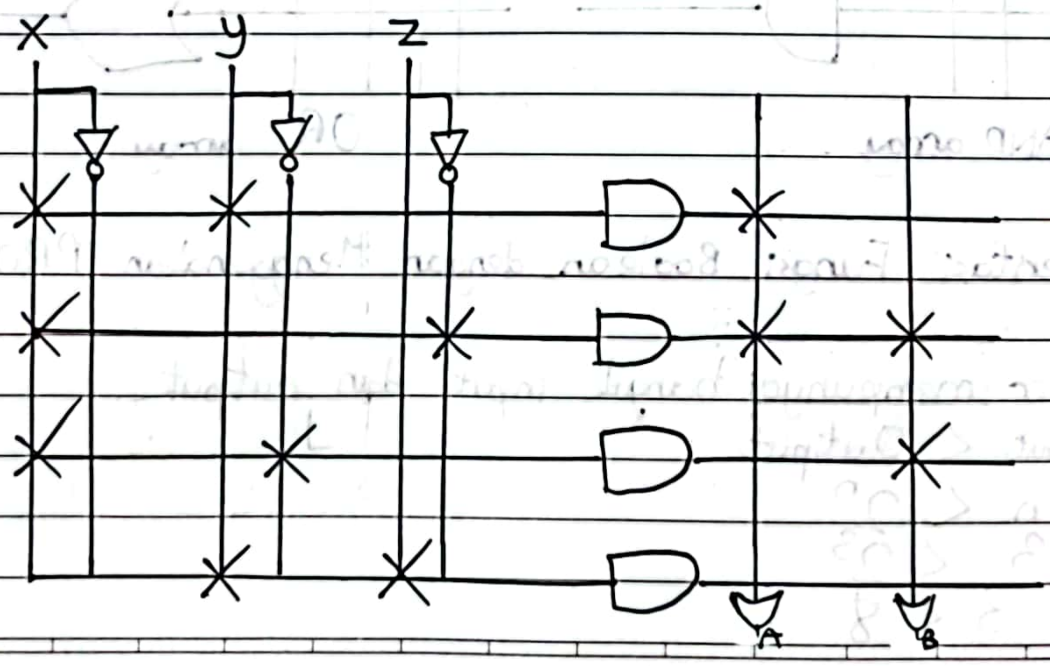
kuncimatahari.com

Misalkan $A(x,y,z) = \sum m(5,6,7)$
$B(x,y,z) = \sum m(3,5,6,7)$ } 2 output, 3 input



B. Implementasikan Fungsi Boolean ke PLA

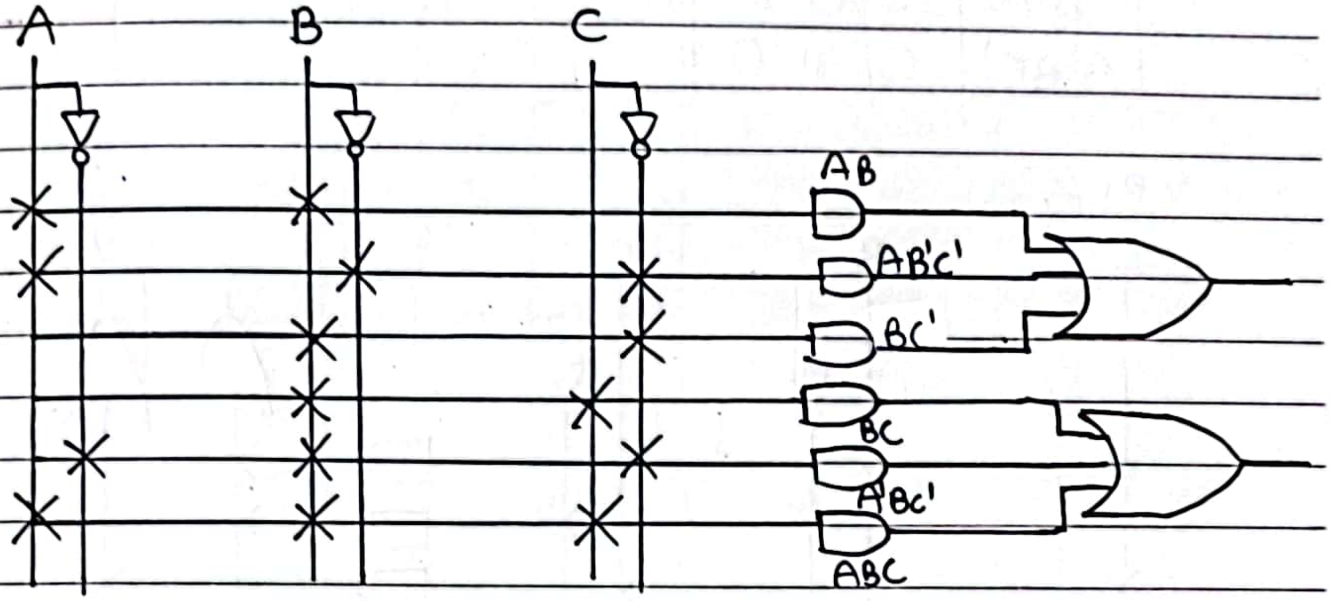
$A = xy + xz'$
$B = xy' + yz + xz'$ } Input : ada 3 → x, y, z; Output : ada 2 → A, B



C. Implementasi Fungsi Boolean Menggunakan PAL

$x = AB + AB'C' + BC'$ } Input : ada 3 → A, B, C

$y = BC + A'BC' + ABC$ } Output : ada 2 → x, y



Contoh Soal :

1. Implementasikan fungsi boolean berikut ke dalam PLA, PAL, PROM.

$x = \Sigma m(0,3,4,7)$

$y = \Sigma m(1,2,5,7)$ $\quad 2^3 \quad 2^4$

→ 7 lbh dkt dgn 8 drpd 16

→ a) Input ada 3, misalkan A, B, C

Output ada 2, x, y

a) x

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

$x = B'C' + BC$

a) y

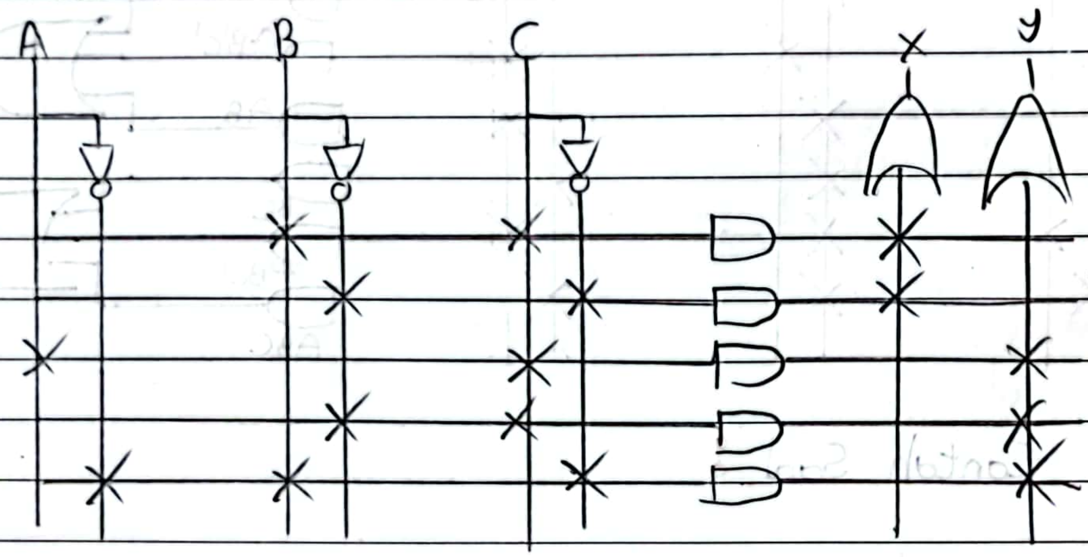
C \ AB	00	01	11	10
0		1		
1	1		1	1

$y = AC + B'C + A'BC'$

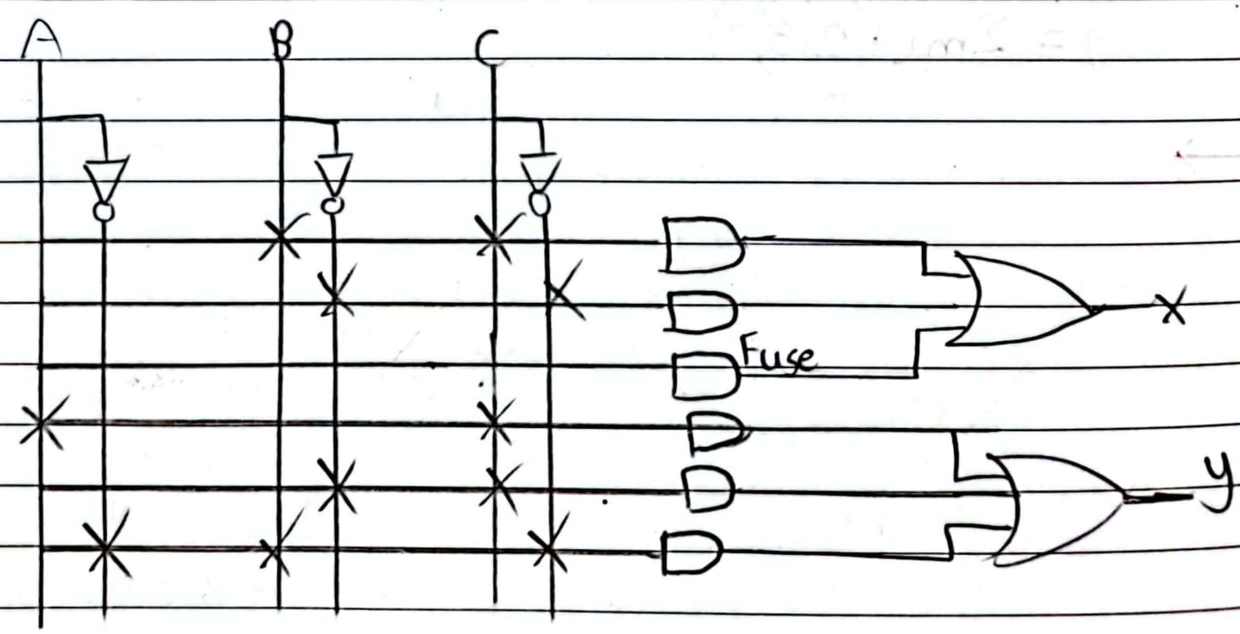
c)

Term	A	B	C	output x	output y
BC	-	1	1	1	-
B'C'	-	0	0	1	-
AC	1	-	1	-	1
B'C	-	0	1	-	1
A'BC'	0	1	0	-	1

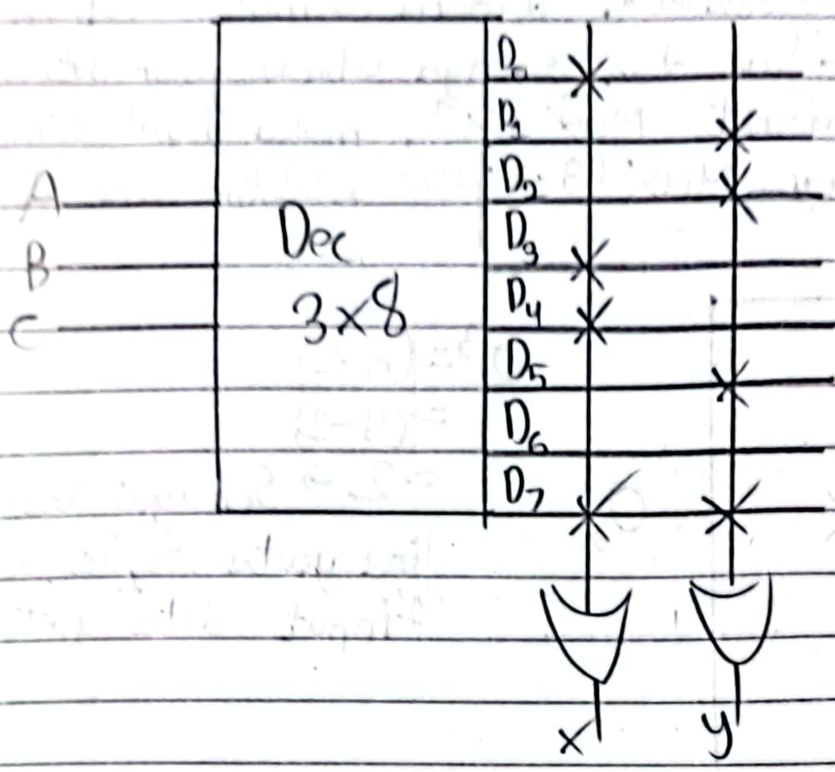
c) PLA:



c) PAL:

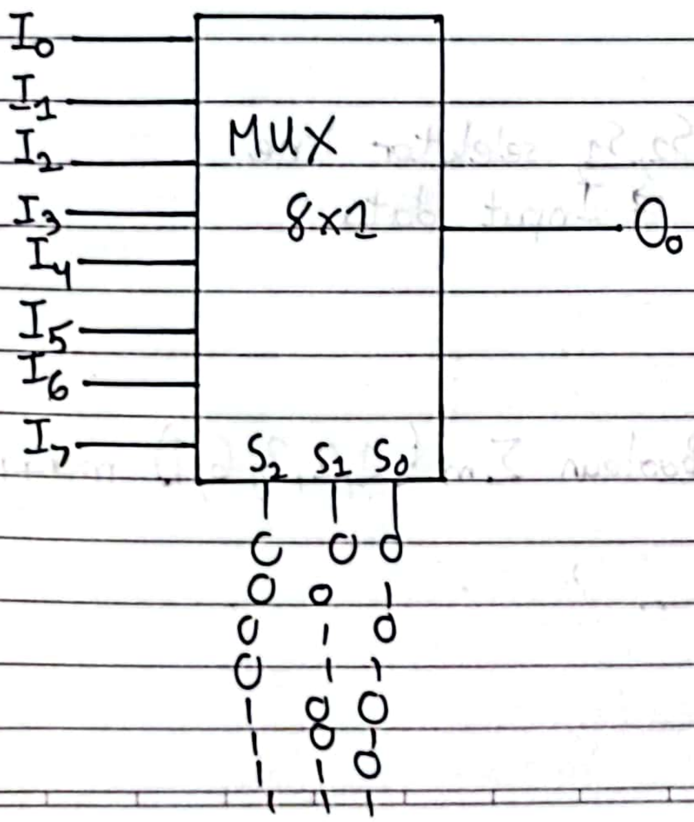


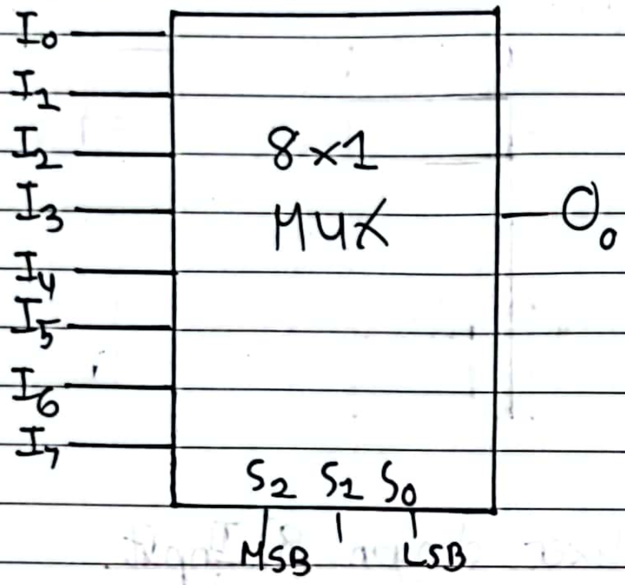
c) PROM : $2^3 = 8$



D. Membuat MUX / Multiplexer dengan 8 Input

- 8 input $= 2^n \rightarrow n = 3 =$ Selektor



- Fungsi boolean dengan 'n' variabel dapat implementasi dengan MUX, (n-1) selektor, dimana dimulai MSB (n-1) sebagai selektor line dan sisinya sebagai jalur data/input.
- Misal, akan membuat MUX 8×1, maka dapat diimplementasikan dengan MUX (3-1 = 2) selektor line dan 1 data input.
- 

$2^3 = (n-1)$
$= (3-1)$
$= 2 \rightarrow$ Sebagai selektor line yaitu S_2, S_1, dan '1' Input data yaitu S_0

- 16 MUX

$16 = 2^4 = (n-1)$
$= 4-1$
$= 3 \rightarrow S_3, S_2, S_1$ selektor line
$S_0 \rightarrow$ Input data

Contoh Soal :

1. Implementasi fungsi Boolean $\Sigma m(1,2,3,6,7)$ menggunakan MUX.

→ o) Berarti menggunakan MUX 8×1
↳ Input ada 3
↳ $2^3 = (n-1)$
$= 3-1 = 2 \rightarrow$ selektor line
'1' sbg input jalur data

DATE Bocalke kirl PAGE

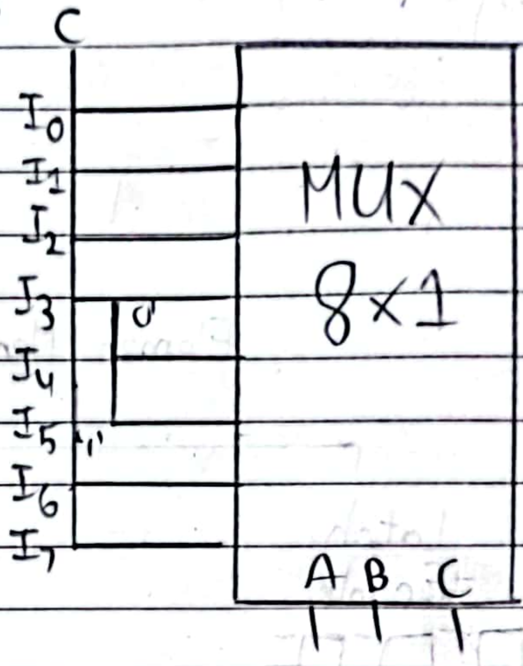
a)

A	B	C	X
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

b) Bro K-Map

	C \ AB	00	01	11	10
0	C	0	2	4	6
1	$\bar{C}$	1	3	5	7
		$C+0$	$\bar{C}+C$	0	$\bar{C}+C$
		C	1	0	1

c)

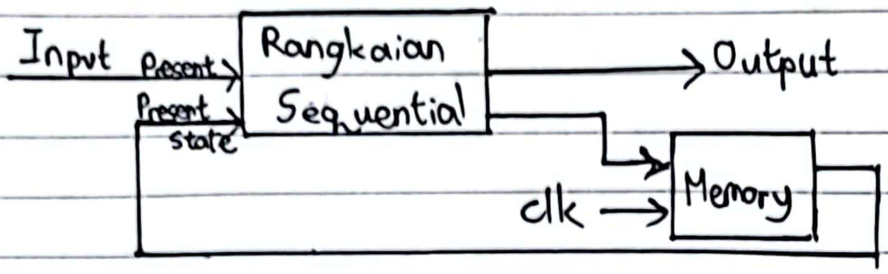


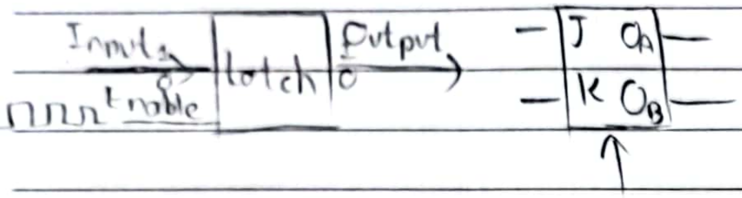
Worry is like a down payment to problem you may never had.
kuncimatahari.com

Week 15

DATE PAGE

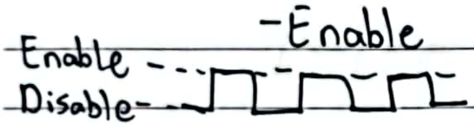
Rangkaian Sequential

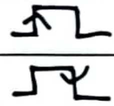




Elemen Memory

- Latch
  - Enable
- Flip-Flop
  - Clock / clk
    - ↳ Trigger edge
    - ↳ Falling edge

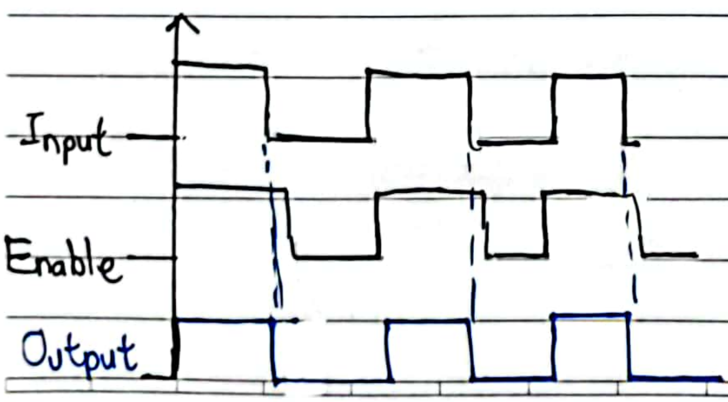




Positive triggered edge

Negative falling edge

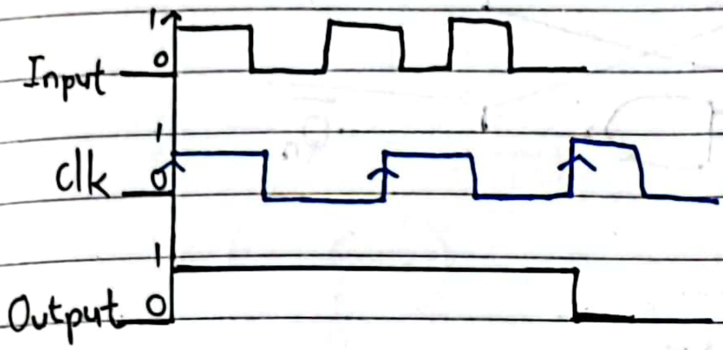
Latch :



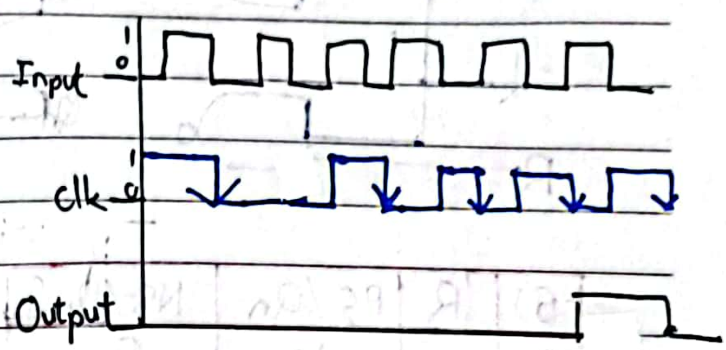
kuncimatahari.com
Person who chases two rabbits catches neither.

◦ Clock :

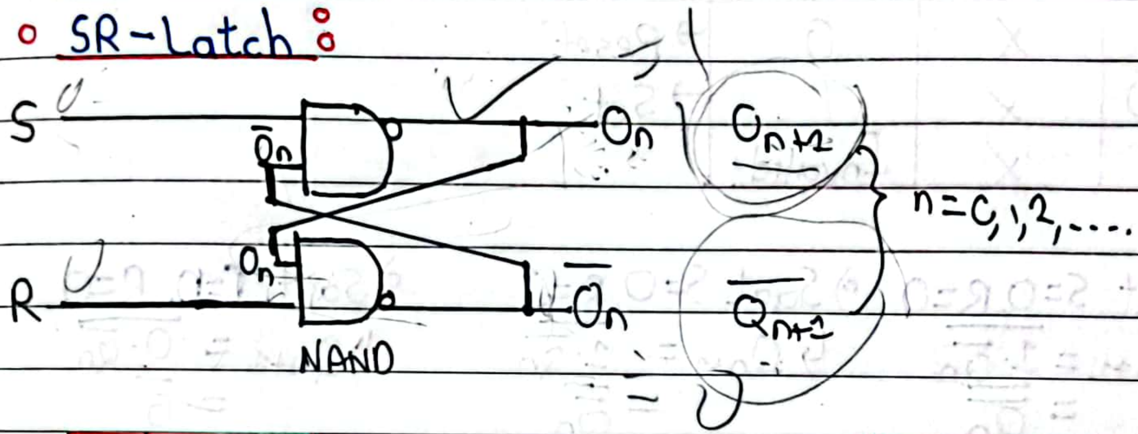
1) Positive Triggered Edge



2) Negative Falling Edge



◦ SR-Latch :



◦

S	R	Q_n
0	0	Invalid
0	1	1 → set
1	0	0 → reset
1	1	Memory

◦

S	R	Present State Q_n	Next State Q_{n+1}
0	0	$x <^0_1$	Invalid
0	1	$x <^0_1$	1
1	0	$x <^0_1$	0
1	1	$x <^0_1$	Memory

◦ Saat S=0, R=0

↳ $Q_{n+1} = \overline{0 \cdot \overline{Q_n}}$
$= \overline{0}$
$= 1$

↳ $\overline{Q_{n+1}} = \overline{0 \cdot Q_n}$
$= \overline{0}$
$= 1$

◦ Saat S=0, R=1

↳ $Q_{n+1} = \overline{0 \cdot \overline{Q_n}}$
$= \overline{0}$
$= 1$

↳ $\overline{Q_{n+1}} = \overline{1 \cdot Q_n}$
$= \overline{Q_n}$

◦ Saat S=1, R=0

↳ $Q_{n+1} = \overline{1 \cdot \overline{Q_n}}$
$= \overline{\overline{Q_n}}$
$= Q_n = 0$

↳ $\overline{Q_{n+1}} = \overline{0 \cdot Q_n}$
$= \overline{0}$
$= 1$

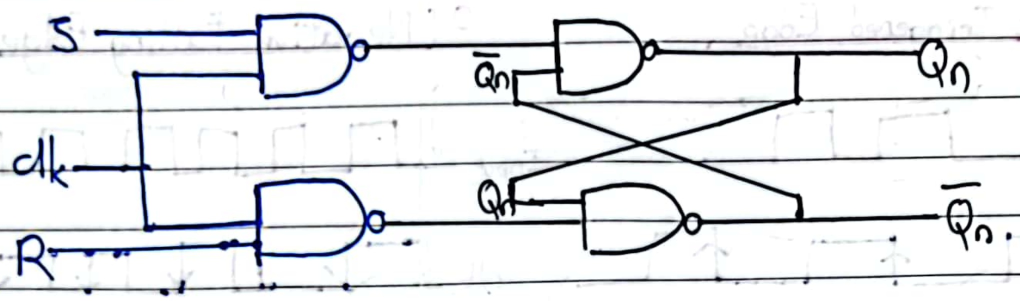
◦ Saat S=1, R=1

↳ $Q_{n+1} = \overline{1 \cdot \overline{Q_n}}$
$= \overline{\overline{Q_n}}$
$= Q_n$

↳ $\overline{Q_{n+1}} = \overline{1 \cdot Q_n}$
$= \overline{Q_n}$

Worry is like a down payment to problem you may never had.
kuncimatahari.com

◦ SR-FlipFlop :



S	R	PS / Q_n	NS / Q_{n+1}	
0	0	X	Q_n	→ Memory
0	1	X	0	→ Reset
1	0	X	1	→ Set
1	1	X	Invalid	

◦ Saat S=0, R=0

↳ $Q_{n+1} = \overline{1 \cdot \overline{Q_n}}$
$= \overline{\overline{Q_n}}$
$= Q_n$

◦ Saat S=0, R=1

↳ $Q_{n+1} = \overline{1 \cdot \overline{Q_n}}$
$= \overline{\overline{Q_n}}$
$= Q_n = 0$

↳ $\overline{Q_{n+1}} = \overline{0 \cdot Q_n}$
$= \overline{0}$
$= 1$

◦ Saat S=1, R=0

↳ $Q_{n+1} = \overline{0 \cdot \overline{Q_n}}$
$= \overline{0}$
$= 1$

◦ Saat S=1, R=1

↳ $Q_{n+1} = \overline{0 \cdot \overline{Q_n}}$
$= \overline{0}$
$= 1$

↳ $\overline{Q_{n+1}} = \overline{0 \cdot Q_n}$
$= \overline{0}$
$= 1$

Equation Table :

S	R	PS/Q_n	NS/Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	X
1	1	1	X

Excitation Table :

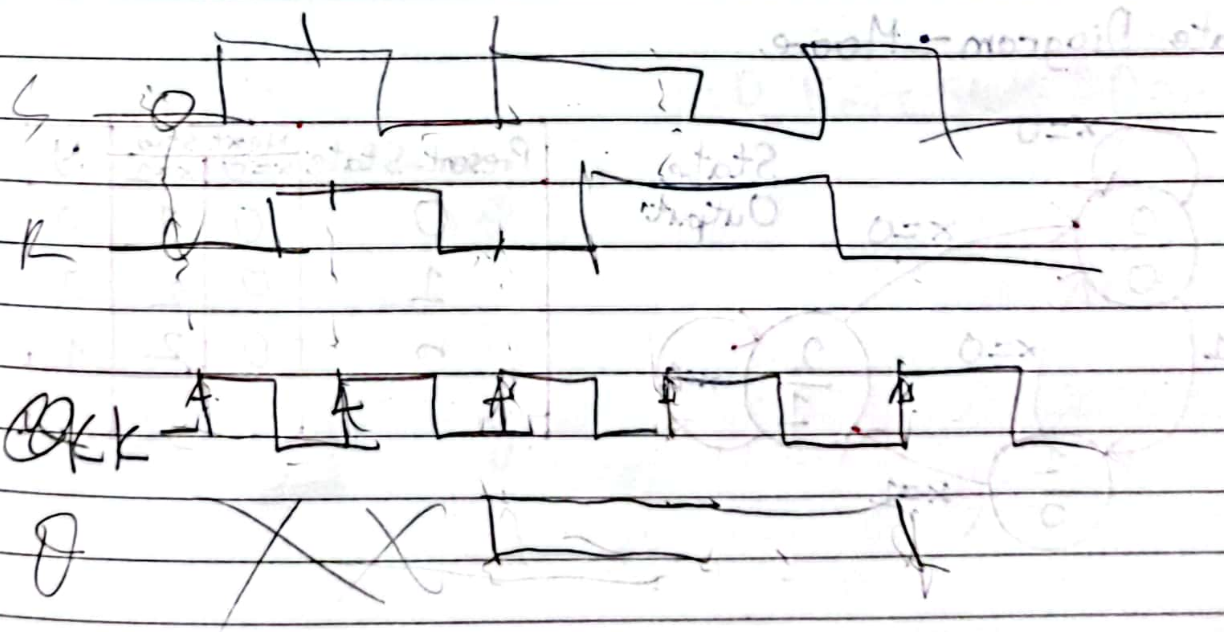
PS/Q_n	NS/Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

S R
00
01
0X

00
10
X0

- $Q_{n+1} = S + Q_n R'$

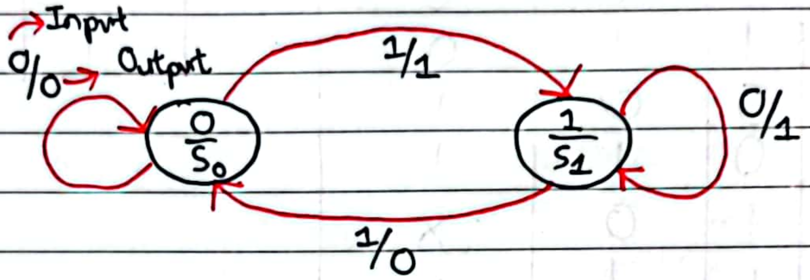
Q_n \ SR	00	01	11	10
0			X	1
1	1		X	1



Aturan penulisan state diagram yaitu :

1) Digambarkan dengan lingkaran
2) ◦ x = Input
   ◦ y = Output
3) Panah = Arah perpindahan

Off On
(S_0) (S_1)

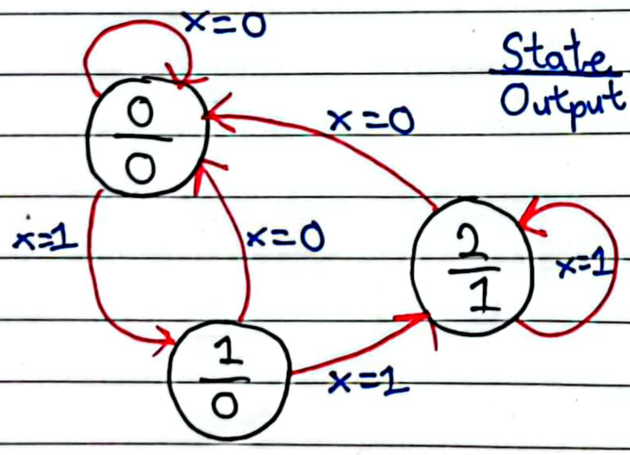


Present State	X	Next State	y
S_0	0	S_0	Padam
S_0	1	S_1	Nyala
S_1	0	S_1	Nyala
S_1	1	S_0	Padam

=

P S	Input x=0	Input x=1	N S x=0	N S x=1	y

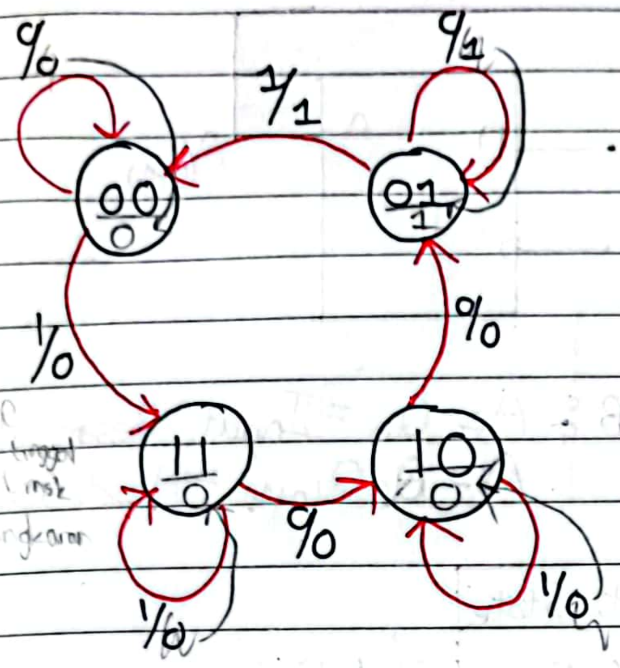
A. State Diagram - Moore



State
Output

Present State	Next State x=0	Next State x=1	y
0	0	1	0
1	0	2	0
2	0	2	1

B. State Diagram – Mealy



Present State	Input	Next State	Output
00	0	00	0
01	0	01	1
10	0	01	0
11	0	10	0
00	1	11	0
01	1	00	1
10	1	10	0
11	1	11	0

Circuit To State Diagram

Langkahnya ada 3 yaitu :

1) Analisa rangkaian / circuit
   - Input
   - Output
   - Jenis Flip-Flop : T-FF, D-FF, JK-FF, SR-FF
   - Persamaan fungsi
2) Membuat state table
3) Membuat State diagram

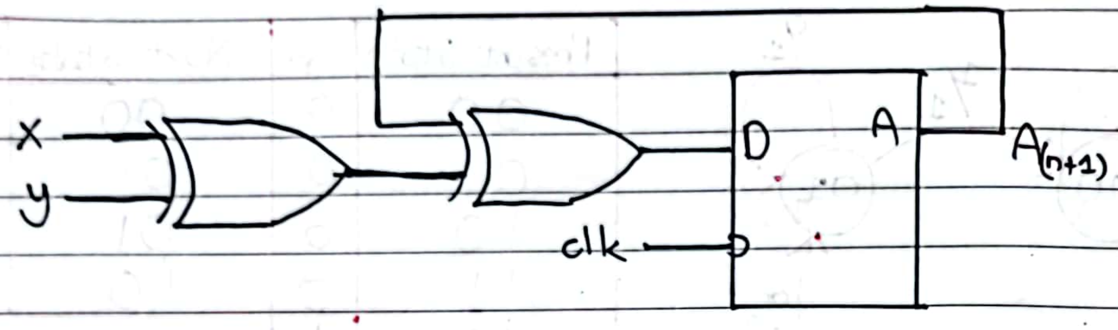
JK Next State : $Q_{n+1} = \overline{Q_n} \cdot J + Q_n \cdot K$

D Next State : $D = A \oplus (x \oplus y)$

$Q_{n+1} = D$

T Next State : $Q_{n+1} = Q_n \oplus T$

Misal :



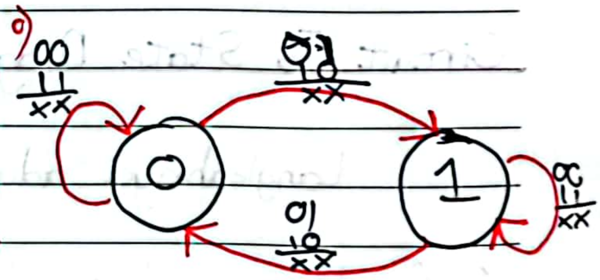
→ Input = x & y

- FF = 1 → D-FF

- NB : $A = Q_n$ = Input Present State $<^0_1$

  $A_{(n+1)} = Q_{(n+1)}$ = Next State

Present State	Input FF		Next State
A	x	y	$A_{(n+1)}$
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

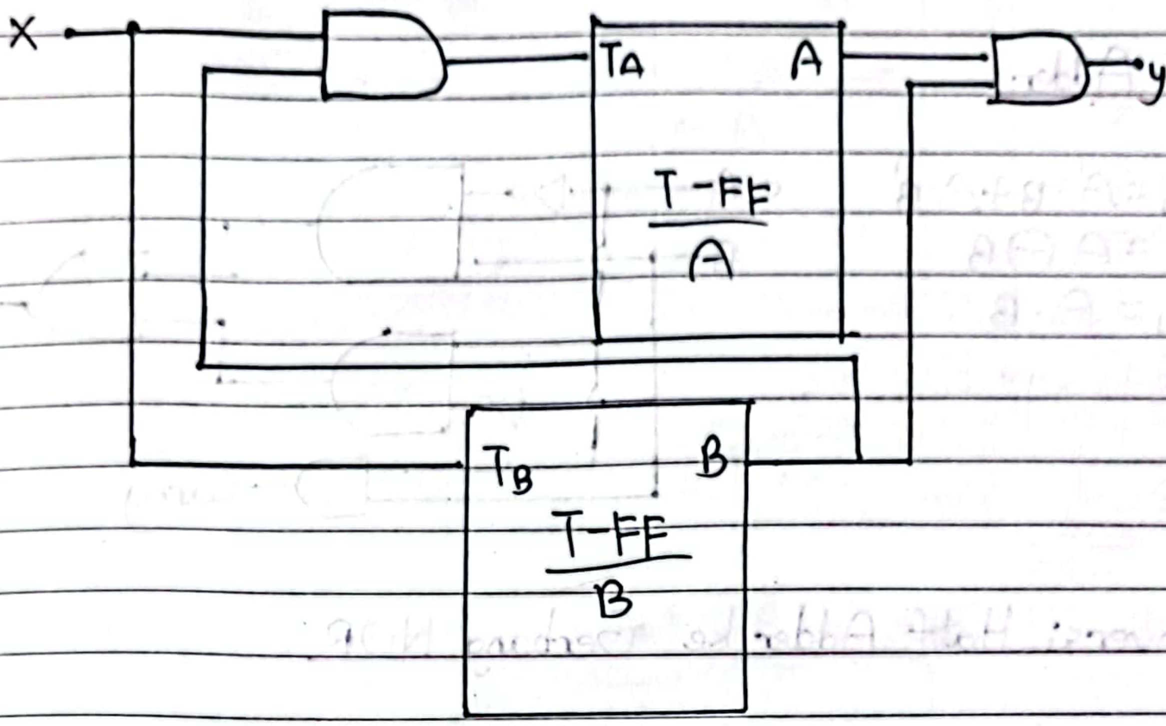


$2^n \geq M$

$2^1 \geq 2$

- $2^n \geq M$
  - ↳ M = Banyaknya state
  - ↳ n = Banyaknya FF/Bit

Misal :



- Input = 1 = x
- Output = 1 = y
- FF → 2 → T-FF
- $Q_{(n+1)} = Q_n \oplus T$

  $A_{(n+1)} = A \oplus T_A$

  $= A \oplus (B \cdot X)$

- $2^n \geq M$

  $2^2 = 4 <$ state 00, 01, 10, 11

- $B_{(n+1)} = B \oplus T_B$

  $= B \oplus X$

- Pers. fungsi : $T_A = B \cdot T_A$

  $T_B = X$

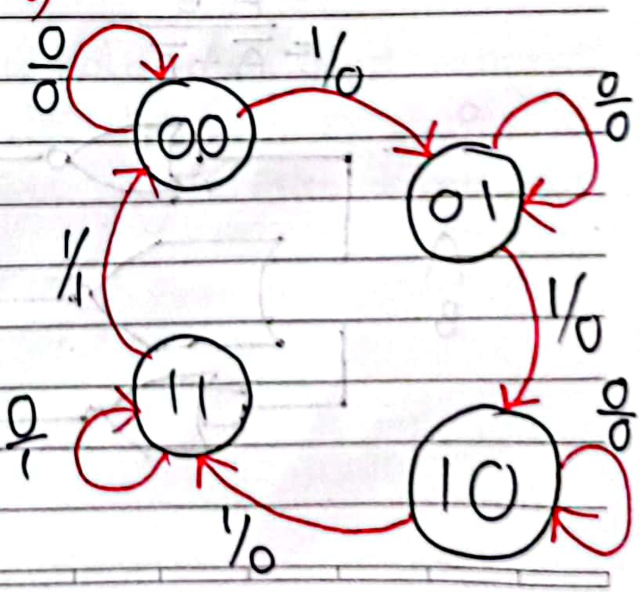
  $y = A \cdot B$

- $y = AB$

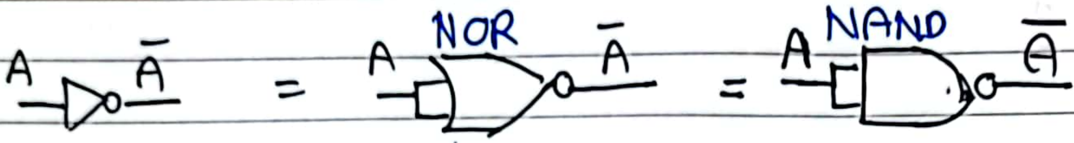
-

Present State			Next State		Output
A B		X	$A_{(n+1)}$	$B_{(n+1)}$	y
0	0	0	0	0	0
0	1	0	0	1	0
1	0	0	1	0	0
1	1	0	1	1	1
0	0	1	0	1	0
0	1	1	1	0	0
1	0	1	1	1	1
1	1	1	0	0	0

-



DATE PAGE

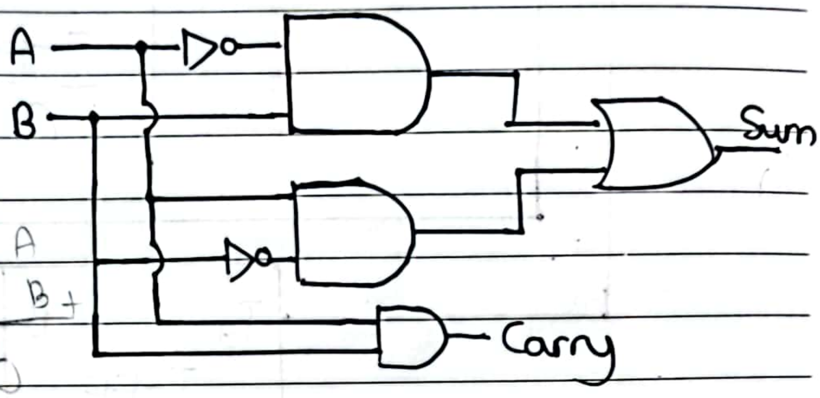


A. Half Adder

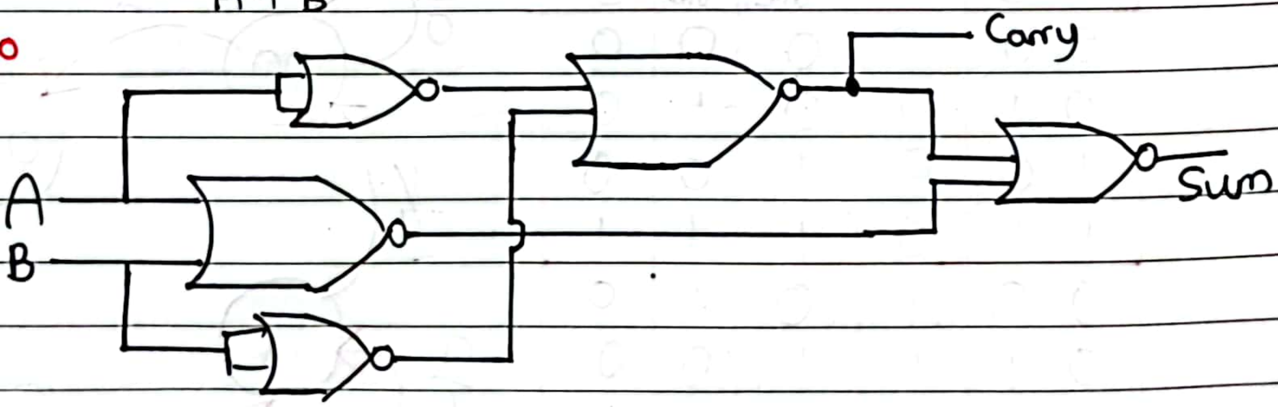
- $Sum = A' \cdot B + A \cdot B'$
  $= A \oplus B$
- $Carry = A \cdot B$

A	B	sum	carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

1001 A
0010 B +
1011



1) Konversi Half Adder ke Gerbang NOR

- $Sum = A' \cdot B + A \cdot B'$
  $= A' \cdot B + A \cdot B' + A' \cdot A + B \cdot B'$ → Hukum komplemen
  $= A' \cdot B + B \cdot B' + A \cdot B' + A' \cdot A$
  $= (A' + B')(A + B)$
  $= \overline{\overline{(A' + B')(A + B)}}$
  $= \overline{\overline{(A' + B')} + \overline{(A + B)}}$
- $Carry = A \cdot B$
  $= \overline{\overline{A \cdot B}}$
  $= \overline{\overline{A} + \overline{B}}$
- 

2) Konversi Half Adder ke Gerbang NAND

- $Sum = A' \cdot B + A \cdot B'$

$= A' \cdot B + A \cdot B' + A' \cdot A + B' \cdot B$ → Hukum komplemen

$= A' \cdot B + B \cdot B' + A \cdot B' + A' \cdot A$

$= B(A' + B') + A(B' + A')$

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

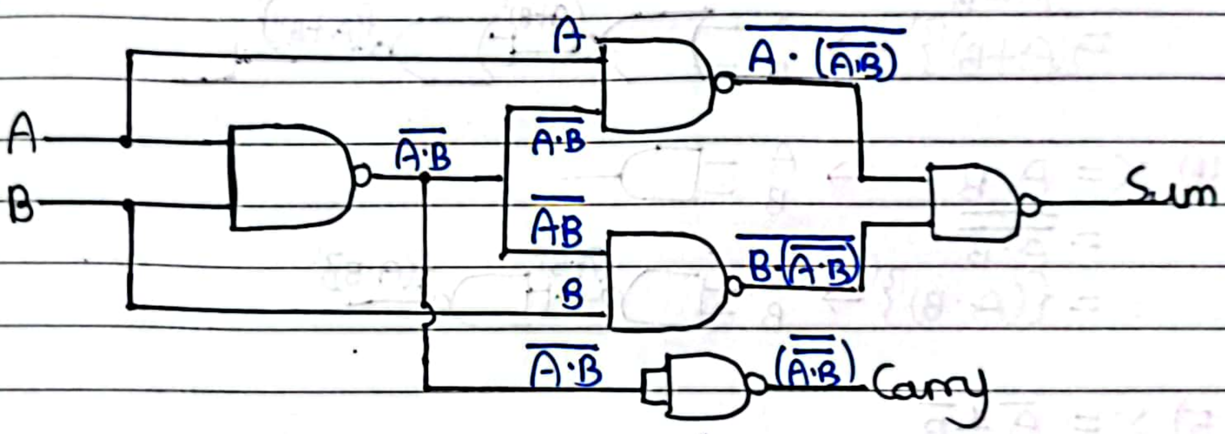
$= \overline{\overline{B(\overline{A \cdot B}) + A(\overline{A \cdot B})}}$ → Hukum involusi

$= \overline{\overline{B \cdot (\overline{A \cdot B})} \cdot \overline{A(\overline{A \cdot B})}}$

- $Carry = A \cdot B$

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

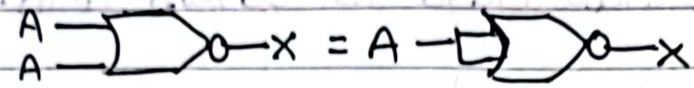
-



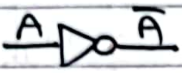
Universal Gate (NAND dan NOR)

1) Dengan menggunakan universal gate maka akan dapat mengganti fungsi dari basic gate (AND, OR, NOT)
2) Sehingga akan menjadi rangkaian yang efisien dengan cost yang murah.

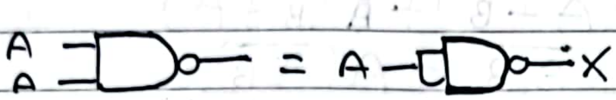
1) $X = A+A$

$= \overline{A+A}$ → 

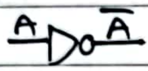
$= \overline{A}\cdot\overline{A}$

$= \overline{A}$ → 

2) $X = A\cdot A$

$= \overline{A\cdot A}$ → 

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

$= \overline{A}$ → 

3) $X = A+B$ → 

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

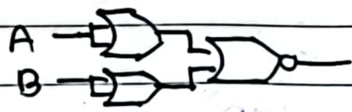
$= \{(A+B)'\}'$ → 

4) $X = A\cdot B$ → 

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

$= \{(A\cdot B)'\}'$ → 

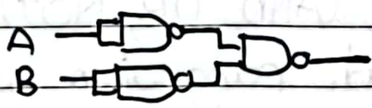
5) $X = \overline{A}+\overline{B}$

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

$= \overline{AB}$

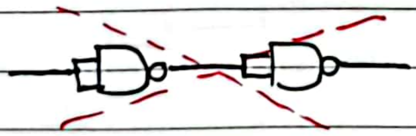
$= AB$ → 

6) $X = \overline{A}\cdot\overline{B}$

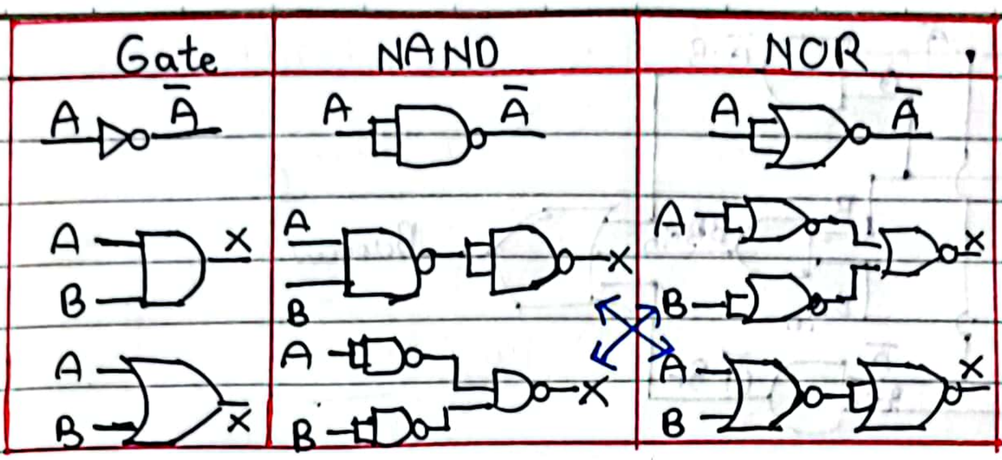
$= \overline{\overline{\overline{A}\cdot\overline{B}}}$ → 

$= \overline{A+B}$

$= A+B$ → 

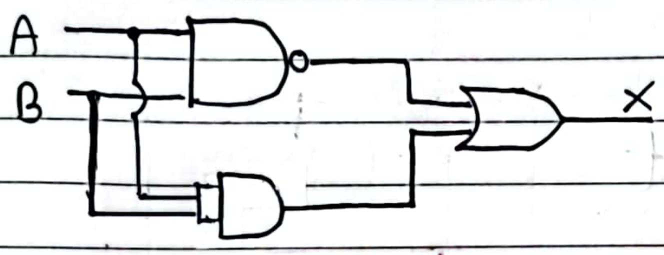


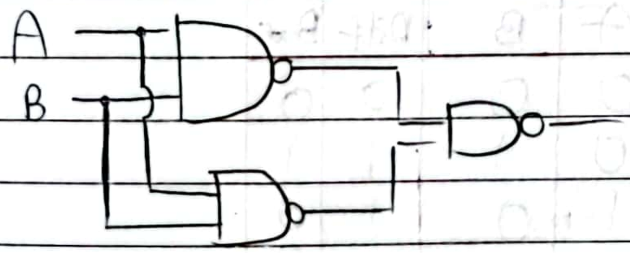
Jika gatenya sama dan seri, boleh dihilangkan.



Contoh Soal :

1. Buatlah menjadi NAND semua





Subtractor

Diff

Br_{in} \ AB	00	01	11	10
0		1		1
1	1		1	

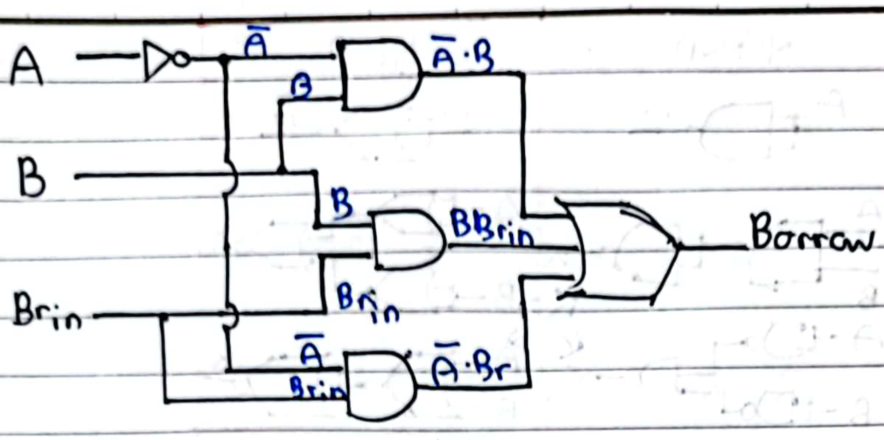
- $Diff = \bar{A}\bar{B}C + \bar{A}B\bar{C} + ABC + A\bar{B}\bar{C}$
  $= C(\bar{A}\cdot\bar{B} + AB) + \bar{C}\cdot(\bar{A}\cdot B + A\bar{B})$
  $= C(\overline{A \oplus B}) + \bar{C}(A \oplus B)$
  $= C \oplus A \oplus B$

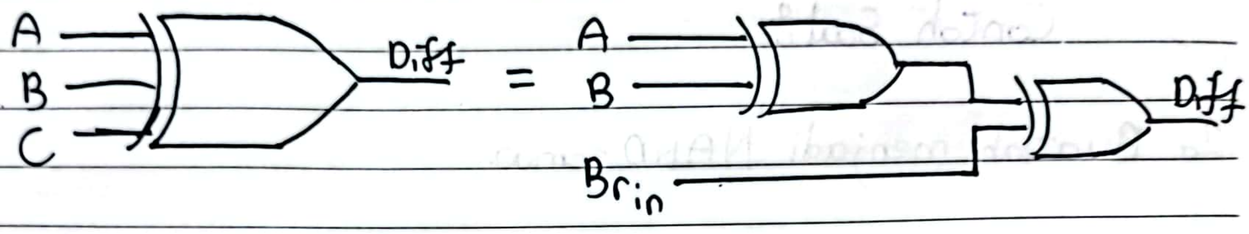
- Borrow

Br_{in} \ AB	00	01	11	10
0				
1	1	1	1	

- $Borrow = \bar{A}\cdot Br_{in} + \bar{A}\cdot B + B\cdot Br_{in}$

Worry is like a down payment to problem you may never had.
kuncimatahari.com





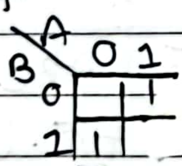
B. Half Subtractor

↳ Pengurangan single bit

- 0101 A
  0001 B −
  0100 Diff
  0000 Borr

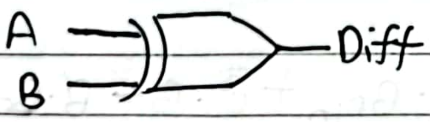
Input		Output	
A	B	Diff	Borr
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

- Diff

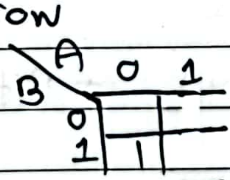


↳ $Diff = \overline{A}B + A\overline{B}$

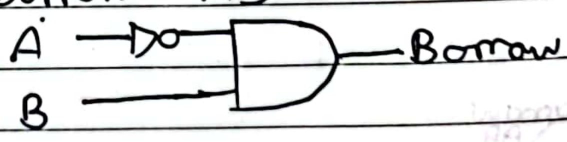
$= A \oplus B$



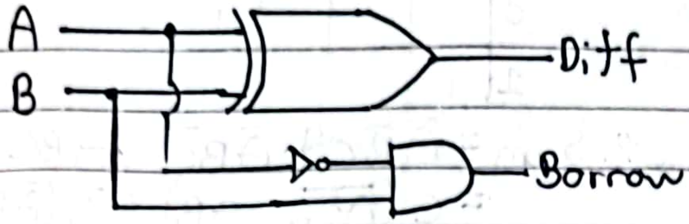
- Borrow



↳ $Borrow = \overline{A}B$



- Gambar Half Subtractor :



C. Full Subtractor

↳ $A - B - Br_{in} = A - (B + Br_{in})$

Input			Output	
A	B	Br_{in}	Diff	Bor
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

D. Full Adder

↳ Digunakan untuk penjumlahan single bit dengan carry

```
 1001   A
 1011   B +
10100   Sum
 1011   c_{i+1} = output
```

Worry is like a down payment to problem you may never had.
kuncimatahari.com

Input			Output	
A_i	B_i	C_i	Sum	C_{i+1}
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

- Sum

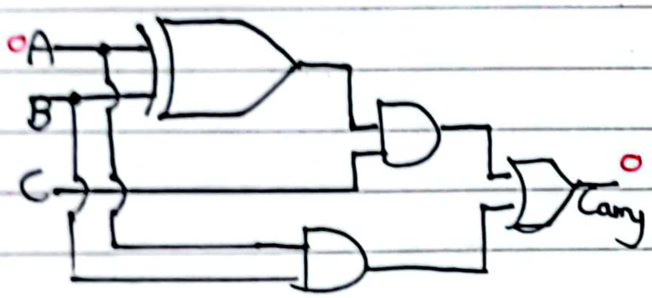
C_i \ AB	00	01	11	10
0		1		1
1	1		1	

- $Sum = \bar{A}\bar{B}C + \bar{A}B\bar{C} + ABC + A\bar{B}\bar{C}$

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

- Carry

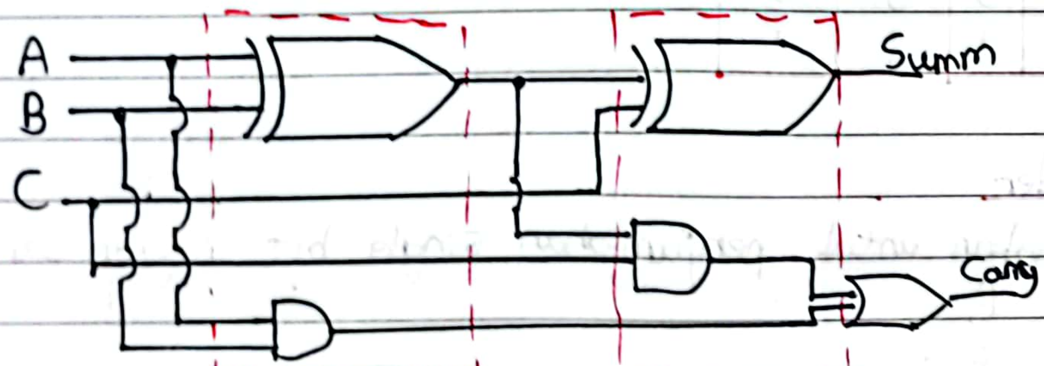
C \ AB	00	01	11	10
0			1	
1		1	1	1



- $Carry = \bar{A}BC_i + AB + A\bar{B}C_i$

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

- Gambar rangkaian:



Half Adder Half Adder

kuncimatahari.com

Person who chases two rabbits catches neither.

1) Full Adder menjadi NAND

↳ $Sum = A \oplus (B \oplus C)$

$= \underbrace{A \oplus B} \oplus C$

$= S_1 \oplus C$

↳ $Carry = C(A \oplus B) + AB$

$= C S_1 + AB$

∘ $S_1 = A \oplus B$

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

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

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

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

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

$= \overline{\overline{B(\overline{AB})}\;\overline{A(\overline{AB})}}$

∘ $Carry = \overline{\overline{CS_1 + AB}}$

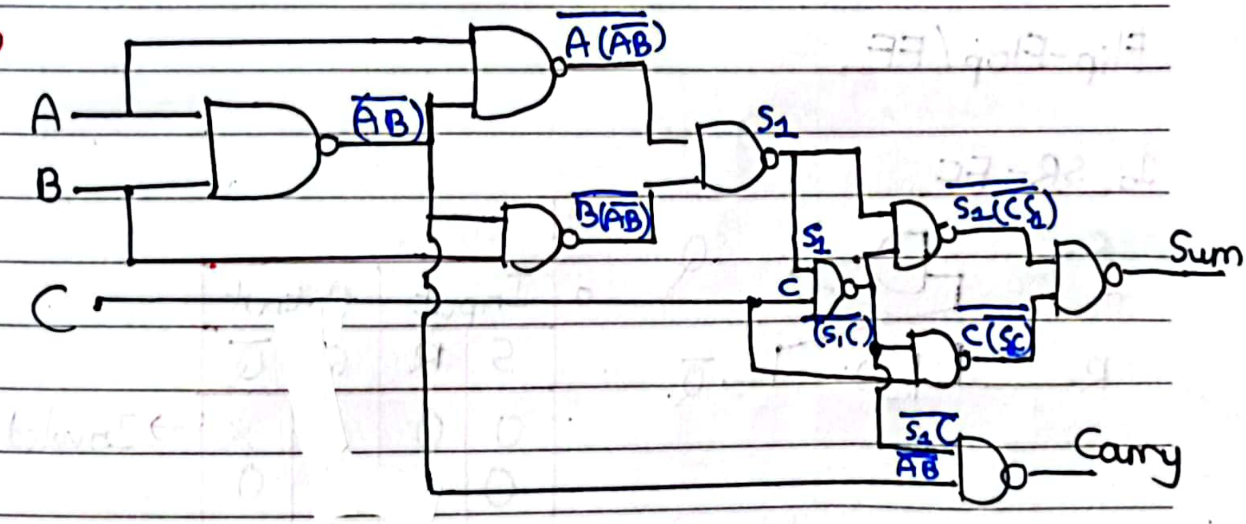
$= \overline{\overline{CS_1}\;\overline{AB}}$

∘ $Sum = \overline{\overline{C(\bar{S_1} + \bar{C}) + S_1(\bar{C} + \bar{S_1})}}$

$= \overline{\overline{C(\bar{S_1} + \bar{C})}\;\overline{S_1(\bar{C} + \bar{S_1})}}$

$= \overline{\overline{C(\overline{S_1 C})}\;\overline{S_1(\overline{C S_1})}}$

∘

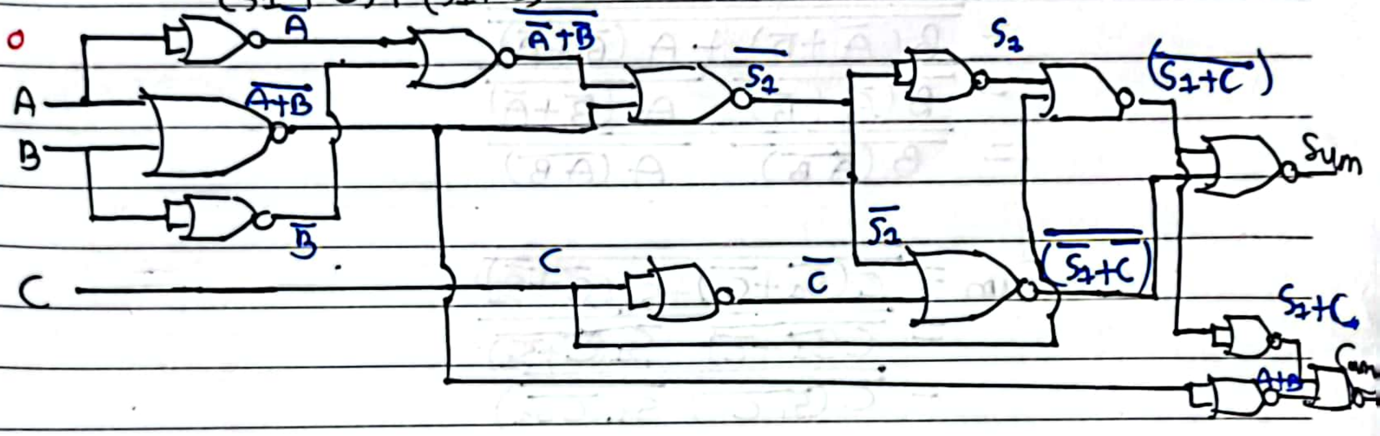


2) Full Adder menjadi NOR

- $S_1 = A \oplus B$
  $= \bar{A}B + A\bar{B}$
  $= \bar{A}B + A\bar{B} + A\bar{A} + \bar{B}B$
  $= (\bar{A}+\bar{B})(A+B)$
  $= \overline{\overline{(\bar{A}+\bar{B})(A+B)}}$
  $= \overline{\overline{(\bar{A}+\bar{B})} + \overline{(A+B)}}$

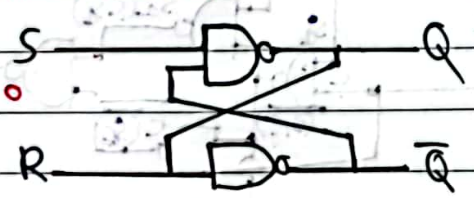
- $Carry = \overline{\overline{C S_1 + AB}}$
  $= \overline{\overline{CS_1} \cdot \overline{A+B}}$
  $= \overline{\overline{\overline{C+S_1}} + \overline{\overline{A+B}}}$
  $= C + S_1 + A + B$

- $Sum = \overline{\overline{(\bar{S}_1+\bar{C})(S_1+C)}}$
  $= \overline{\overline{(\bar{S}_1+\bar{C})} + \overline{(S_1+C)}}$

- 

Flip-Flop/FF

1. SR-FF



- SR latch

Input		Output		
S	R	Q	$\bar{Q}$	
0	0	×	×	→ Invalid
0	1	1	0	
1	0	0	1	
1	1	Q_n	$\bar{Q}_n$	→ Memory

clk (↓)	clk (↑)	S	R	Q_n	Q_{n+1}	
1	0	×	×	0		
0	1	0	0	0	No change	
0	1	0	1	0	0	
0	1	1	0	0	1	
0	1	1	1	×	Invalid	→Toggle

- Characteristic Table :

clk	S	R	Q_n	Q_{n+1}
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	Invalid
1	1	1	1	Invalid

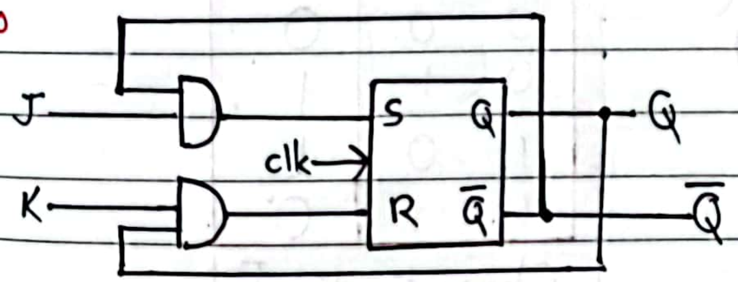
- Excitation Table :

Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

- Truth table :

clk	S	R	Q	$\overline{Q}$
0	X	X	Memory	
1	0	0	Memory	
1	0	1	0	1
1	1	0	1	0
1	1	1	Invalid	

2. JK-FF

- 

-

Input		Output		Keterangan
J	K	Q	$\overline{Q}$	
0	0	Q_n	$\overline{Q_n}$	No change
0	1	0	1	Reset 0
1	0	1	0	Set 1
1	1	$\overline{Q_n}$	Q_n	Toggle → Perlawanan dgn sblmnya

- Truth table :

clk	J	K	Q_{n+1}	
0	X	X	Q_n	} Memory
1	0	0	Q_n	
1	0	1	0	
1	1	0	1	
1	1	1	$\overline{Q_n}$	→ Toggle

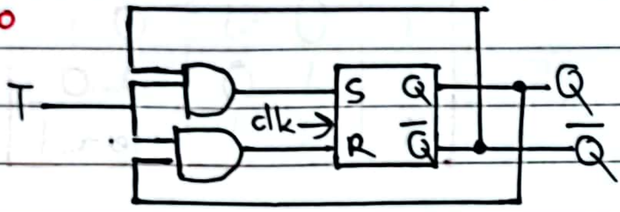
Q_n	J	K	Q_{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Characteristic Table

Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Excitation Table

3. T-FF



Input	Output	
T	Q	$\overline{Q_n}$
0	0	1
1	1	0

clk	T	Q_{n+1}
0	X	Q_n
1	0	Q_n
1	1	$\overline{Q_n}$

$\}$ Memory

$\rightarrow$ Toggle

Truth Table

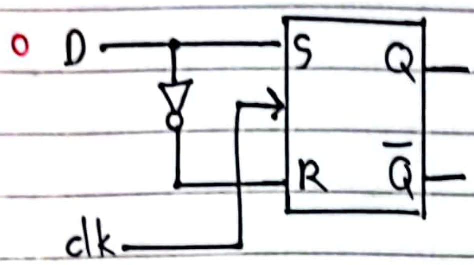
Q_n	T	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

Characteristic Table

Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

Excitation Table

4. D-FF



Input	Output	
clk	Q	$\overline{Q}$
1	1	0
0	0	1

- Truth table :

clk	D	Q_{n+1}
0	x	Q_n
1	0	0
1	1	1

- Characteristic table :

Q_n	D	Q_{n+1}
0	0	0
0	1	1
1	0	0
1	1	1

- Excitation table :

Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

- NB : Synchronous memakai excitation table. Utk mod 6 mod

The true sign of intelligence is not knowledge but imagination..

kuncimatahari.com