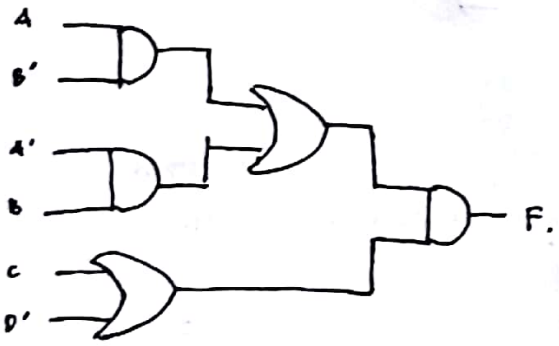
KB

Radiq.

Rangkaian Kombinasional 1

- Rangkaian logika.
  1. kombinasional – tanpa memory.
  2. sequential – dengan memory.
- 2 hal dalam rangkaian kombinasional.
  1. Rangkaian → analisis rangkaian kombinasional
     - truth table :
  2. truth table → dengan rangkaian kombinasional.
     - Rangkaian :



$F = (AB' + A'B)(C + D')$

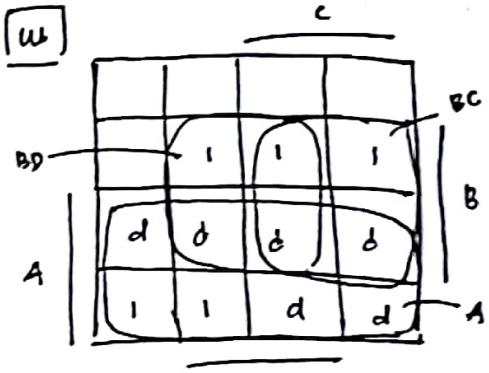
Tabel kebenaran

A	B	C	D	A'	B'	D'	AB'	A'B	(AB'+A'B)	F	C+D'
0	0	0	0	1	1	1	0	0	0	0	1
0	0	0	1	1	1	0	0	0	0	0	0
0	0	1	0	1	1	1	0	0	0	0	1
0	0	1	1	1	1	0	0	0	0	0	1
0	1	0	0	1	0	1	0	1	1	1	1
0	1	0	1	1	0	0	0	1	1	0	0
0	1	1	0	1	0	1	0	1	1	1	1
0	1	1	1	1	0	0	0	1	1	1	1
1	0	0	0	0	1	1	1	0	1	1	1
1	0	0	1	0	1	0	1	0	1	0	0

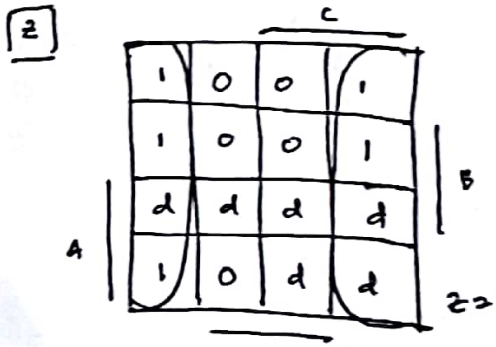
A	B	C	D	A'	B'	D'	AB'	A'B	(AB'+A'B)	F	C+D'
1	1	0	1	0	0	0	1	0	1	1	1
1	1	0	1	1	0	0	1	0	1	1	1
1	1	1	0	0	0	1	0	0	0	0	0
0	1	1	0	1	0	0	0	0	0	0	0
1	1	1	1	0	0	0	1	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0

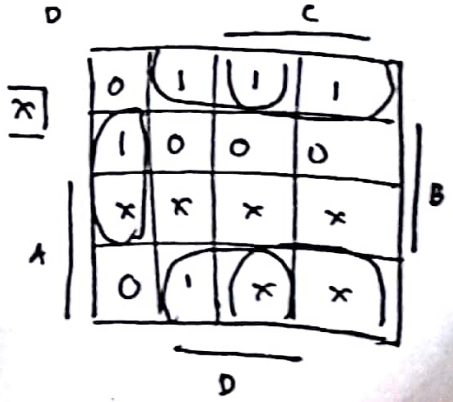
Contoh 2

1. Input BCD – 8421 → output Excess 3 code

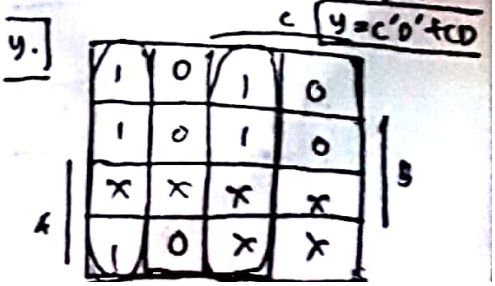


$W = A + BD + BC$





$x = B'D + BC'D' + B'C$



Kg. Radrq.

Rangkaian Kombinasional 2.

latihan 1

- input

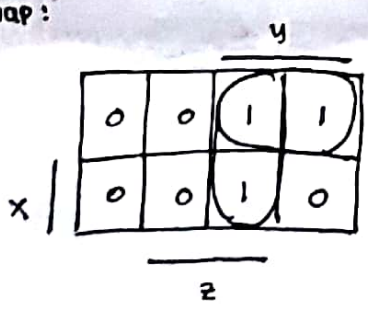
x	y	z
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

- output

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

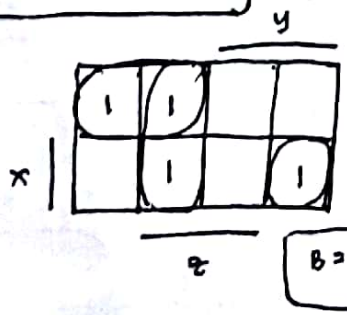
- Kmap:

A.



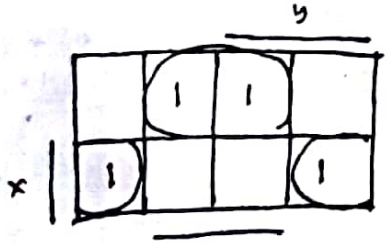
$A = x'y + yz$

B.



$B = x'y' + y'z + xyz'$

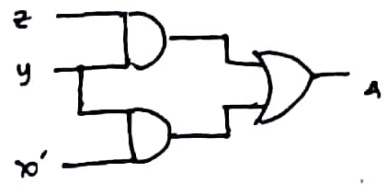
C.



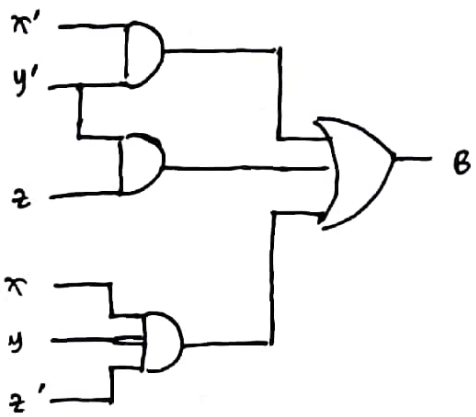
$C = x'z + xz'$ — x nor z

logic gate:

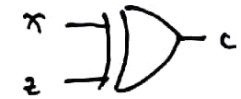
A.



B.



C.



latihan 2

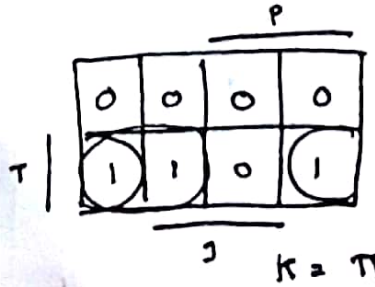
- input

temp	pintu	jendela
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

- output

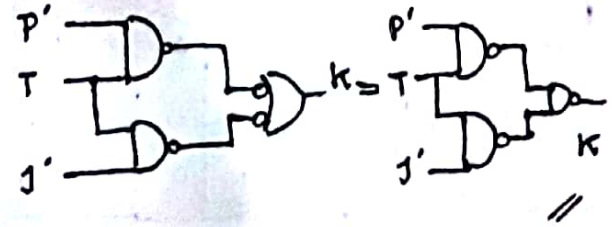
kipas
0
0
0
0
1
1
1
0

kmap:



$K = TP' + TJ'$

logic gate: nand ?



radrg.

Decoder

↳ mengkonversikan informasi biner dari n input lines ke maksimum 2^n output lines yg unik.

- active low → yg nyala yg 0.
- active high → nyala yg bernilai 1.

demultiplexer → decoder dg enable input.

contoh:

1. 2 to 4 line decoder with E, active high.

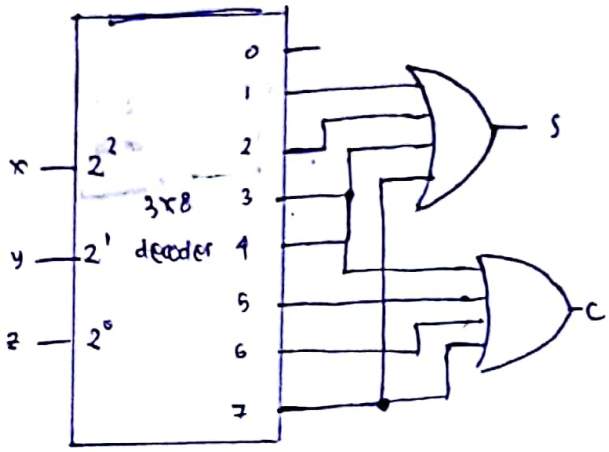
E	A	B	D_0	D_1	D_2	D_3
0	x	x	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

- Full Adder menggunakan 3 x 8 decoder.

x	y	z	s	c	
0	0	0	0	0	m_0
0	0	1	1	0	m_1
0	1	0	1	0	m_2
0	1	1	0	1	m_3
1	0	0	1	0	m_4
1	0	1	0	1	m_5
1	1	0	0	1	m_6
1	1	1	1	1	m_7

$S = \Sigma (1,2,4,7)$

$C = \Sigma (3,5,6,7)$



Encoder

↳ kebalikan dari decoder.

- Priority Encoder.
  - Jika 2 input / lebih menyala, yg dipertimbangkan hanya 1.
  - Tambahan output V yg menandakan validitas output (nilai 1 jika ada 1 / lebih input)

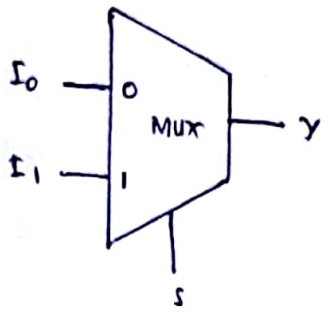
contoh:

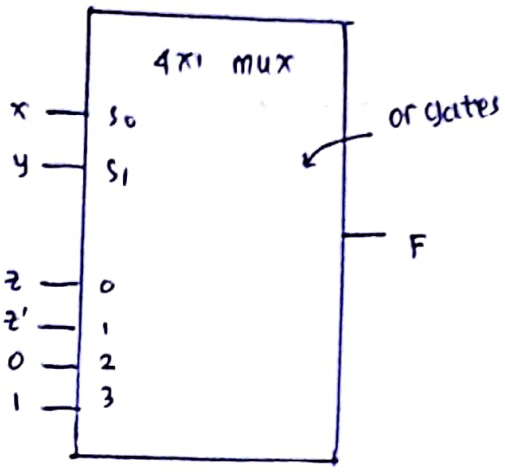
input				output		
D_0	D_1	D_2	D_3	x	y	V
0	0	0	0	x	x	0
1	0	0	0	0	0	1
x	1	0	0	0	1	1
x	x	1	0	1	0	1
x	x	x	1	1	1	1

Multiplexer.

↳ Data Selector.

Block diagram:





Implementasi fungsi dg Multiplexer.

↳ fungsi dg n variabel.

multiplexer dg n-1 selection lines (2^{n-1} pilihan input).

contoh:

3 variabel → mux 2 selection lines dan 4 input lines.

① variabel input : x, y : selection lines, S_1 dan S_0

Variabel z : input

x	y	z	F	
0	0	0	0	F = z
0	0	1	1	
0	1	0	1	F = z'
0	1	1	0	
1	0	0	0	F = 0
1	0	1	0	
1	1	0	1	F = 1
1	1	1	1	

implementasi fungsi dg 2x1 multi plexer.

variabel fungsi = 4

selection lines mux = 3

input lines mux = 8

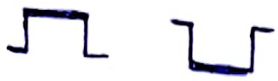
1. lengkapi truth table dari sop.
2. n-1 variabel yg pertama dipilih sebagai selection inputs dari multiplexer.
3. Untuk setiap kombinasi variabel selection, can outputnya sbg fungsi dari variabel terakhir.
4. nilai ini digunakan untuk data input dg urutan yg sesuai.

radiq.

Rangkaian Sekuensial 1

- Sequential Logic

1. Asynchronous



active high / active low.

2 Synchronous

positive edge / negative edge

- Latches → berubah selama high.

. active high :

S	R	Q	Q'
1	0	1	0
0	0	1	0
0	1	0	1
0	0	0	1
1	1	undefined	

no change

- SR Latches.

sama dg latches, tapi undefined saat inputnya 0 semua.

- Flip-flops → berubah sementara

o D flip flop.

D	Q_{t+1}
0	0
1	1

$Q_{t+1} = D$

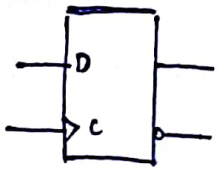
- Jk flip flop.

J	K	Q_{t+1}	
0	0	Q_t	no change $Q_{t+1} = JQ_t' + K'Q_t$
0	1	0	reset.
1	0	1	set
1	1	Q_t'	complement

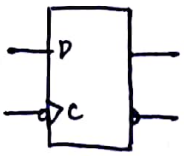
- T flip-flop. (toggle)

T	Q_{t+1}	
0	Q_t	$Q_{t+1} = Q_t \oplus T$
1	Q_t'	

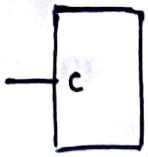
Rangkaian Sekuensial 2



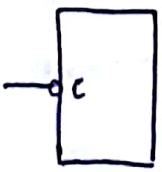
→ positive edge



→ negative edge.



→ active high.



→ active low.

Analisis Rangkaian Sekuensial dengan clock

Istilah:

- State equation / persamaan equation.
  menentukan next state
- state Table
  terdiri dari: present state, input, next state dan output.
- State diagram.
  informasi dalam state table direpresentasikan secara grafis. dinyatakan dg lingkaran dan transisi antar state

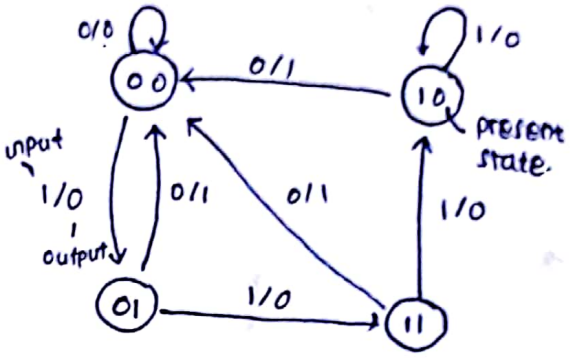
prosedur:

1. tentukan pers. input flip-flop dalam present state dan variabel input.
2. substitusikan pers. input ke dalam pers. karakteristik flip-flop untuk memperoleh pers. state.
3. Gunakan pers. state yg sesuai u/ menentukan nilai next state dalam state table.

contoh:

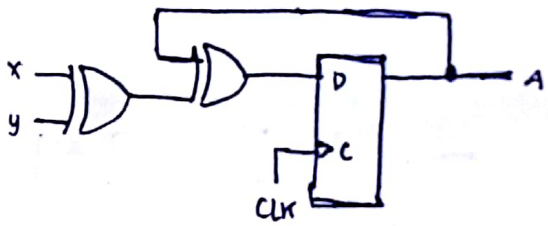
Present state		input	next state		output.	
A	B	x	A	B	y	
0	0	0	0	0	0	
0	0	1	0	1	0	a
0	1	0	0	0	1	
0	1	1	1	1	0	b
1	0	0	0	0	1	
1	0	1	1	0	0	c
1	1	0	0	0	1	
1	1	1	1	0	0	

State diagram:



contoh:

①



- input: $D_A = A \oplus x \oplus y$

  D-flip flop → $Q(t+1) = D$

- Pers. state:

  $A(t+1) = A \oplus x \oplus y$

- state table:

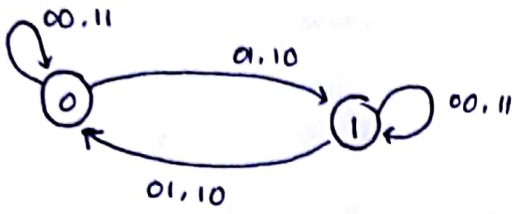
Present State	inputs		next state
A	x	y	A
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

K.11.

radio.

lanjutan soal:

- State diagram



- Mealy model :

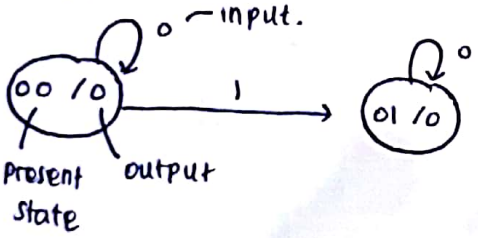
output bergantung pada input dan present state.

- Moore model → T-flip flop.

output hanya bergantung pada present state.

↳ jadi, tanda / berada di dalam lingkaran.

contoh:



K.12

Rangkaian Sekuensial 3.

- Prosedur perancangan.

1. diagram state
2. penyederhanaan state
3. nilai biner trap state.
4. tabel state
5. tipe flip-flop
6. pers. input output flip flop.
7. diagram logic rancangan.

K.13.

Rangkaian Sekuensial 3.

- Penyederhanaan state.

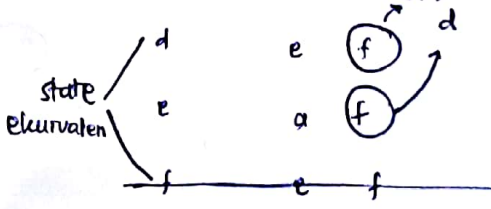
algoritme:

2 state adalah ekuivalen, apabila untuk setiap anggota input :

- output yg diberikan sama
- next state sama.

misal :

Present state	next state x=0	next state x=1	output x=0	output x=1
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	e	f	0	1



ket: baris f dihapus, state f diganti d

- penentuan jumlah FF.

5 state → 3 bit.

4 state → 2 bit.

Jumlah FF = jumlah bit.

- Perancangan Sequence Detector.

contoh:

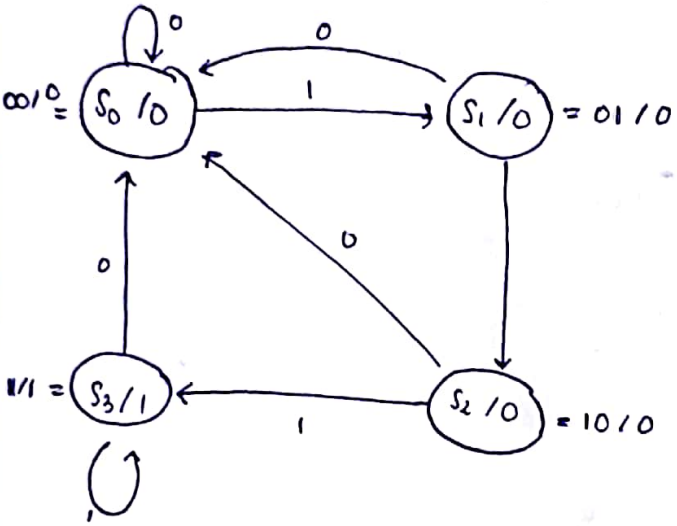
① - mulai dg state S_0.

- Jika input: 0 → tetap pada state yg sama.
- Jika input = 1 → pindah ke state S_1.
- Selanjutnya, input = 1 lanjut ke S_2.

- Namun jika input=0, kembali ke S_0.
- Jika muncul 3 bit 1 secara berurutan, Pindah ke state S_3 dimana output=1.
- Jika > 3 buah bit 1 berurutan, tetap pada state S_3.

• diagram state?

Jawab:



nilai biner tiap state:

$S_0 = 00$, $S_1 = 01$, $S_2 = 10$, $S_3 = 11$

• tabel state

↳ diturunkan dari diagram state memiliki 1 input (x) & 1 output (y).

Present State		input	next State		output
A	B	x	A	B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

• Tentukan pers. input output flip flop.

↳ menggunakan D flip flop?

Pers. state D flip-flop:

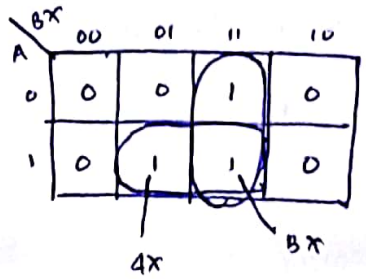
$A(t+1) = D_A(A, B, x) = \Sigma(3, 5, 7)$

$B(t+1) = D_B(A, B, x) = \Sigma(1, 5, 7)$

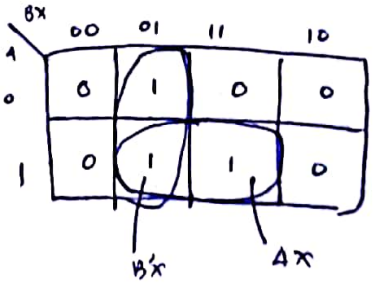
$y(A, B, x) = \Sigma(6, 7)$

Kmap → dari next state & output.

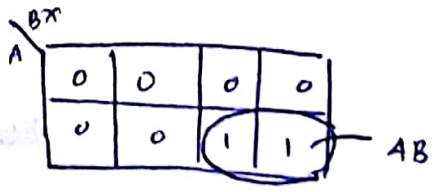
D_A



D_B



y



Sehingga, diperoleh:

$D_A = Ax + Bx$

$D_B = Ax + B'x$

$y = AB$ //

K. B

radra.

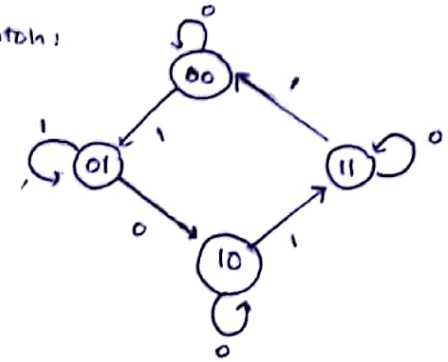
Perancangan dg JK Flip-Flop.

ingat! dihapalkan.

Q(t)	Q(t+1)	J	K	
0	0	0	x	- don't care.
0	1	1	x	
1	0	x	1	JK. flip-flop.
1	1	x	0	

Q(t)	Q(t+1)	T	
0	0	0	
0	1	1	T
1	0	1	flip flop
1	1	0	

contoh:



tabel state.

present state		input	next state		flip-flop input.			
A	B	x	$A_{(t+1)}$	$B_{(t+1)}$	J_A	K_A	J_B	K_B
0	0	0	0	0	0	x	0	x
0	0	1	0	1	0	x	1	x
0	1	0	1	0	1	x	x	1
0	1	1	0	1	0	x	x	0
1	0	0	1	0	x	0	0	x
1	0	1	1	1	x	0	1	x
1	1	0	1	1	x	0	x	0
1	1	1	0	0	x	1	x	1

Kmap. → mencari pers dari input flip-flopnya.

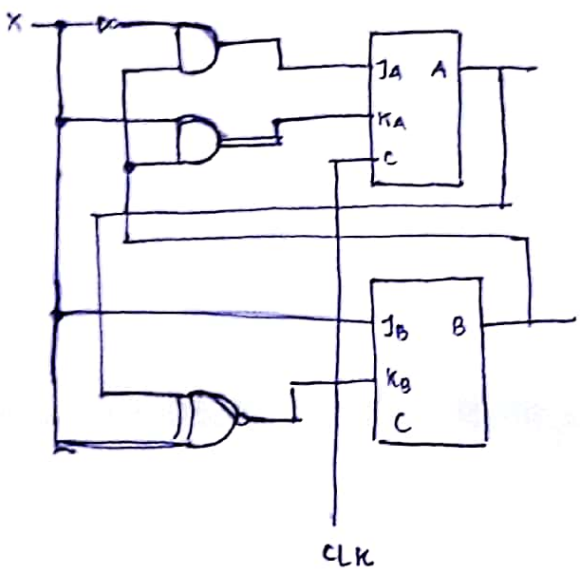
didapat: $J_A = Bx'$

$K_A = Bx$

$J_B = x$

$K_B = (A \oplus x)'$

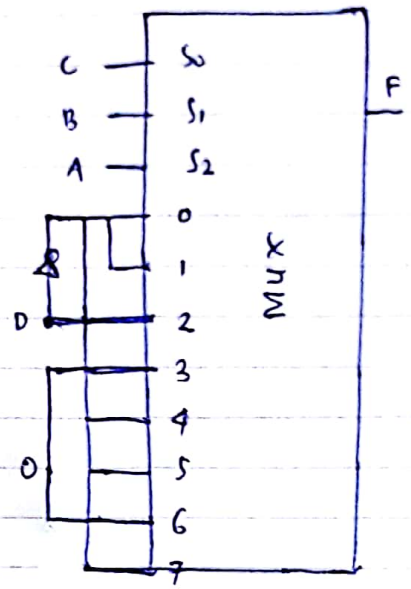
Logic drogram

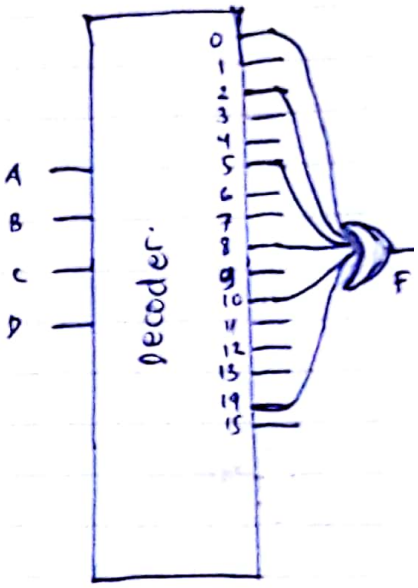


Tugas.

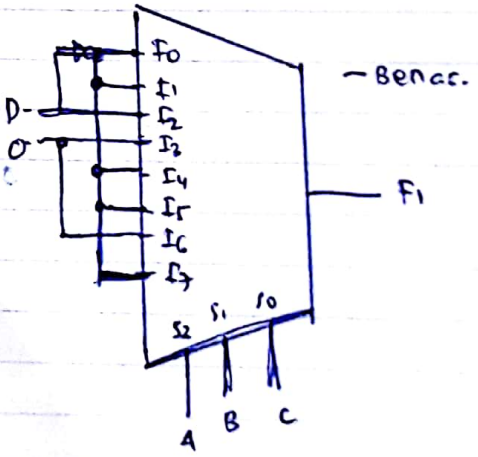
1. $F_1(A,B,C,D) = \Sigma\,(0,2,5,8,10,14)$ $\quad f_2 = (A,B,C,D) = \Pi\,(2,6,11)$

0	0	0	0	0	1	$I_0 = D'$
	0	0	0	1	0	$I_1 = D'$
1	0	0	1	0	1	$I_2 = D$
	0	0	1	1	0	$I_3 = 0$
2	0	1	0	0	0	$I_4 = D'$
	0	1	0	1	1	$I_5 = D'$
3	0	1	1	0	0	$I_6 = 0$
	0	1	1	1	0	$I_7 = D'$
4	1	0	0	0	1	
	1	0	0	1	0	
5	1	0	1	0	1	
	1	0	1	1	0	
6	1	1	0	0	0	
	1	1	0	1	0	
7	1	1	1	0	1	
	1	1	1	1	0	
	A	B	C	D	F	





D' → karena f bertolak belakang dengan D.



2. $F_2 = (A,B,C,D) = \Pi\,(2,6,11)$

Soal UAS.

5.

Input register.

10110 (1) → 1011 (1) → jadi hilang.

digeser ke kanan

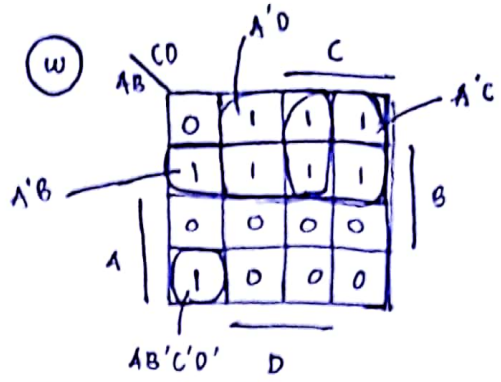
010110	1101
001011	0110
000101	1011
000010	1101
000001	0110
000000	1011

2. truth table. — 2's complement.

A	B	C	D	w	x	y	z	
0	0	0	0	0	0	0	0	m_0
0	0	0	1	1	1	1	1	m_1
0	0	1	0	1	1	1	0	m_2
0	0	1	1	1	1	0	1	m_3
0	1	0	0	1	1	0	0	m_4
0	1	0	1	1	0	1	1	m_5
0	1	1	0	1	0	1	0	m_6
0	1	1	1	1	0	0	1	m_7
1	0	0	0	1	0	0	0	m_8
1	0	0	1	0	1	1	1	m_9
1	0	1	0	0	1	1	0	m_{10}
1	0	1	1	0	1	0	1	m_{11}
1	1	0	0	0	1	0	0	m_{12}
1	1	0	1	0	0	1	1	m_{13}
1	1	1	0	0	0	1	0	m_{14}
1	1	1	1	0	0	0	1	m_{15}

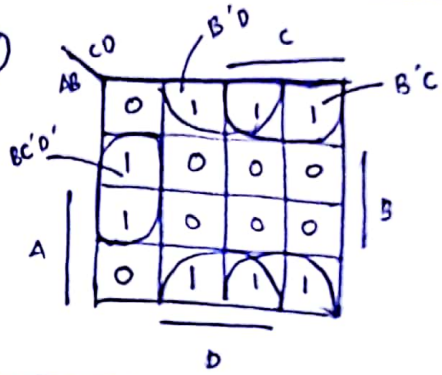
Kmap

(w)



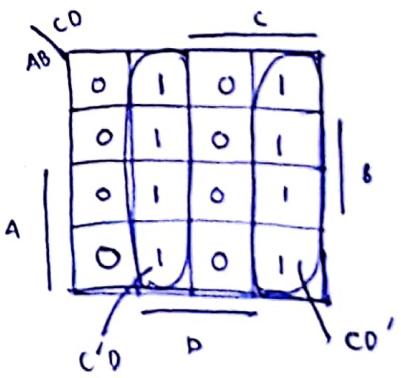
$w = A'B + A'D + A'C + AB'C'D'$

(x)



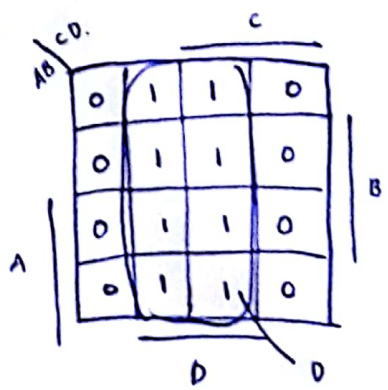
$x = B'C + B'D + BC'D'$

(y)



$y = C'D + CD'$

(z)



$z = D$.