

RANGKAIAN KOMBINASI

Rangkaian Logika dapat digolongkan dalam 2 jenis, yaitu :

1. Rangkaian Kombinasi (Combinatorial Circuit)
2. Rangkaian Berurut (Sequential Circuit)

Keluaran rangkaian kombinasi ditentukan oleh masukannya pada saat itu.

Keluaran rangkaian berurut selain ditentukan oleh masukannya pada saat itu, ditentukan juga oleh keluarannya sebelumnya.

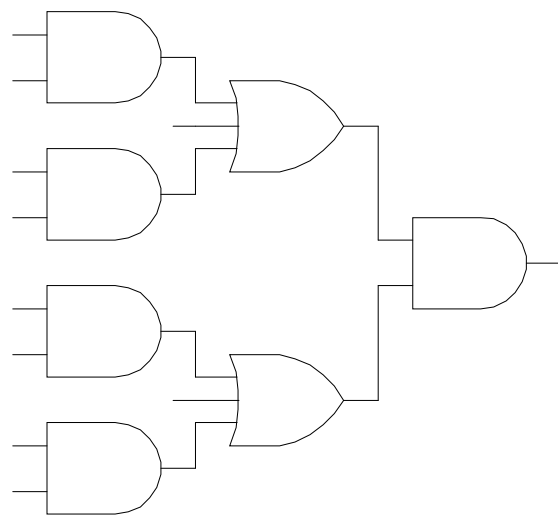
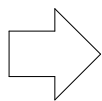
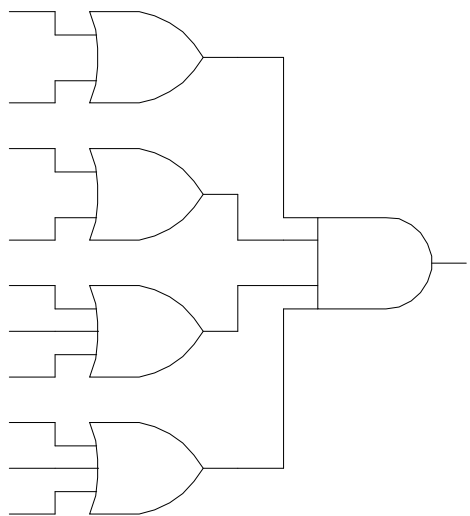
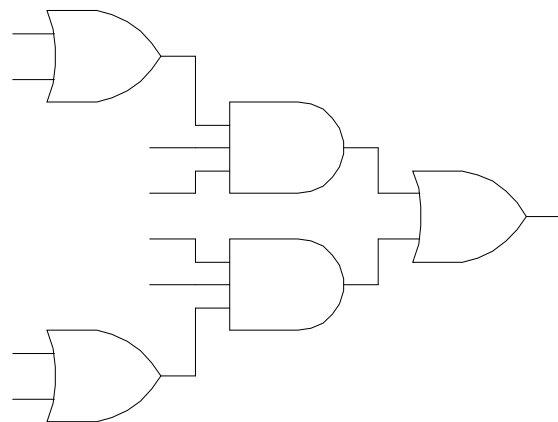
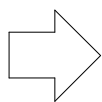
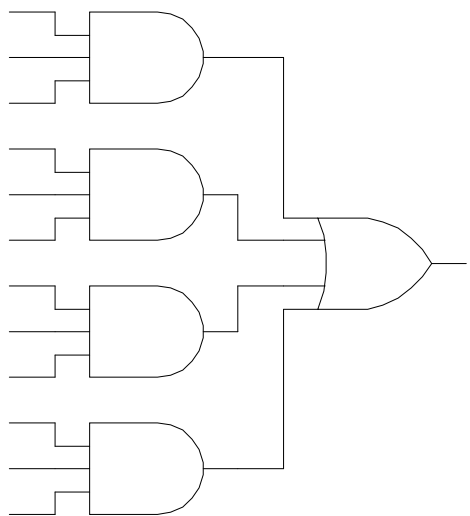
Jumlah Tingkatan

Setiap tingkat dari rangkaian memiliki kelambanan atau tundaan waktu tertentu. Semakin banyak jumlah tingkatan semakin besar tundaan waktu.

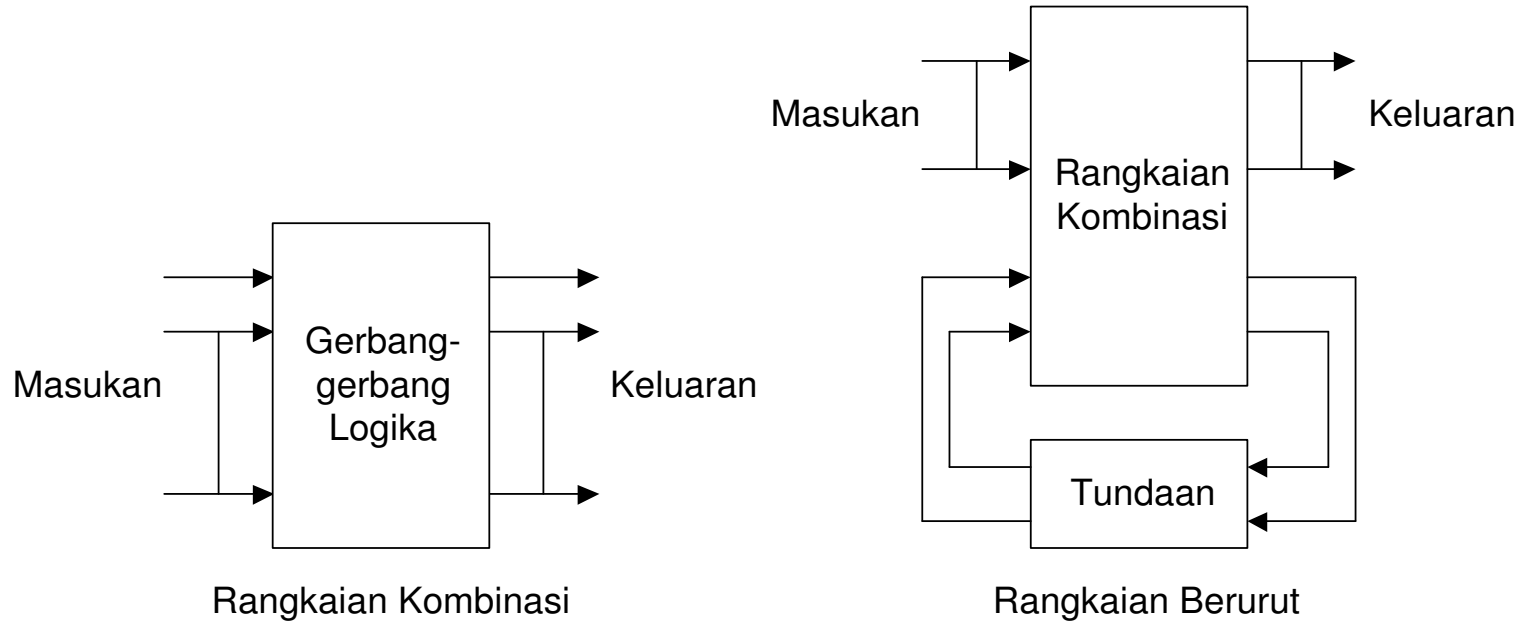
Keberhasilan suatu rangkaian seringkali ditentukan oleh keserempakan sinyal-sinyal masukannya.

Kadang-kadang jumlah tingkatan dari rangkaian harus ditambah untuk menambah tundaan waktu guna mendapatkan keserempakan.

Contoh :



Rangkaian Umum



Perancangan Rangkaian Kombinasi

1. Langkah pertama adalah menentukan apa yang diinginkan dalam bentuk uraian kata-kata. Dari kalimat-kalimat ini dapat diketahui jumlah masukan dan keluaran.
2. Langkah kedua adalah membuat tabel kebenaran (*Truth Table*) yang menyatakan hubungan antara masukan dengan keluaran.
3. Dari tabel kebenaran dapat diperoleh persamaan fungsi Boolean yang selanjutnya dapat disederhanakan secara matematis, dengan peta Karnaugh atau dengan metoda Quine-McCluskey.

Rangkaian Penjumlah Paruh

Fungsi : Menjumlahkan 1-bit dengan 1-bit.

Contoh : $0 + 0 = 0$
 $0 + 1 = 1$
 $1 + 0 = 1$
 $1 + 1 = 0 \rightarrow \text{carry } 1$

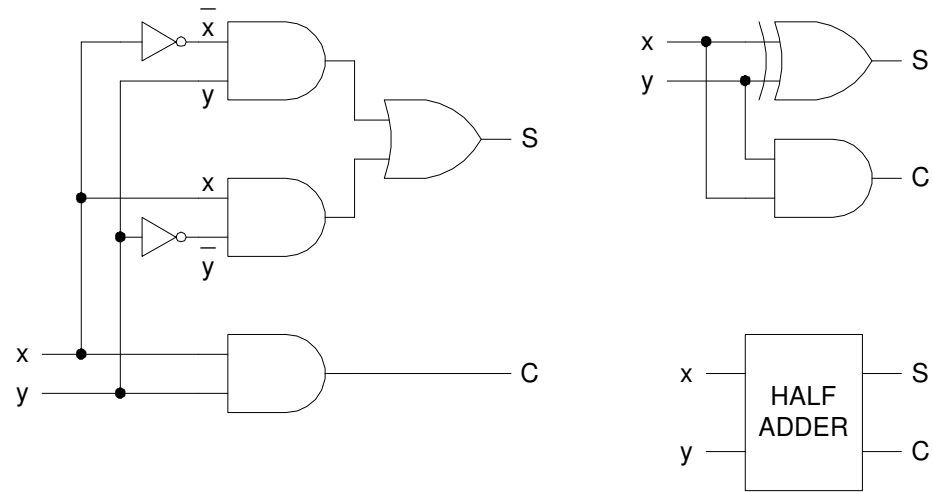
atau dalam bentuk tabel kebenaran :

x	y	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Dari tabel kebenaran ini diperoleh persamaan :

$$S = x \oplus y$$

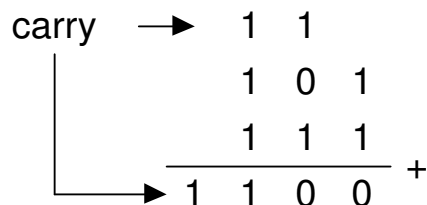
$$C = x \cdot y$$



Rangkaian dan Simbol Penjumlah Paruh

Rangkaian Penjumlah Penuh

Jika pada suatu proses penjumlahan timbul limpahan (carry) maka limpahan ini harus dapat ditampung oleh penjumlah berikutnya.

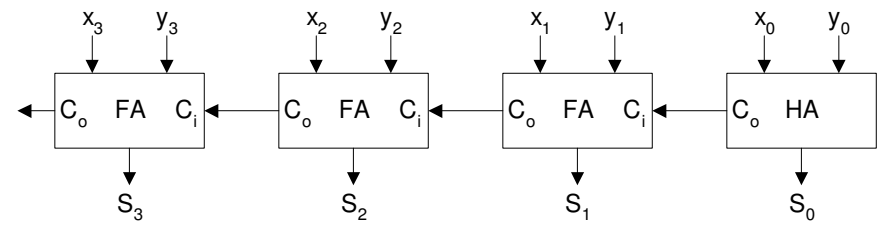
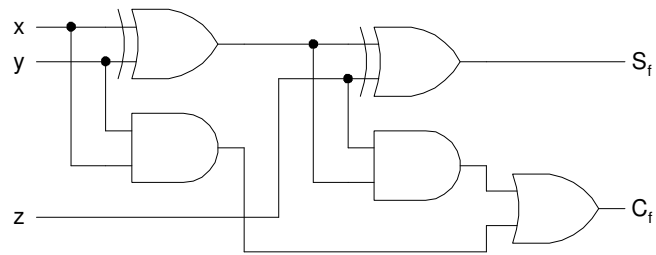


Tabel Kebenaran

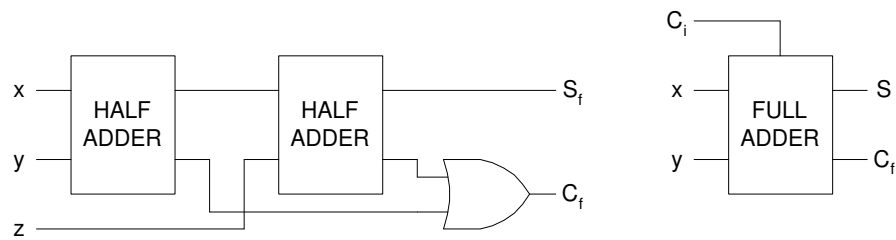
x	y	z	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$S_f = \bar{x}.\bar{y}.z + x.y.z + \bar{x}.y.\bar{z} + x.\bar{y}.\bar{z} = (x \oplus y) \oplus z = S_h \oplus z$$

$$C_f = x.y + x.z + y.z = x.y + (x \oplus y).z = z.S_h + C_h$$



Hubungan Penjumlah Penuh

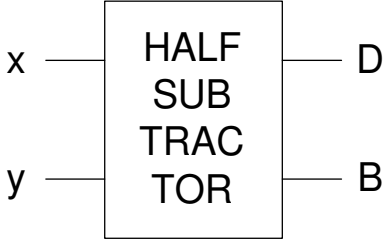
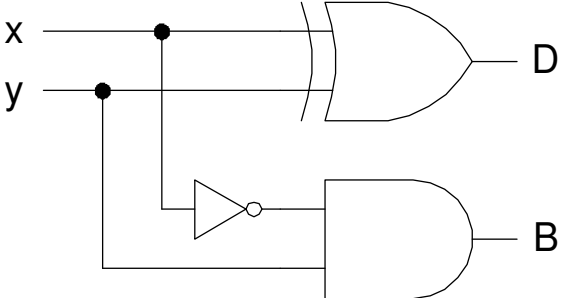


Rangkaian dan Simbol Penjumlah Penuh

Rangkaian Pengurang Paruh

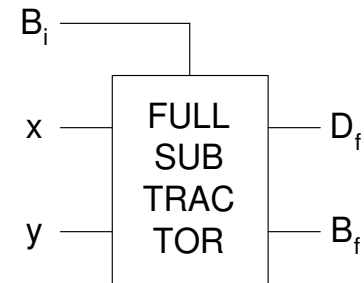
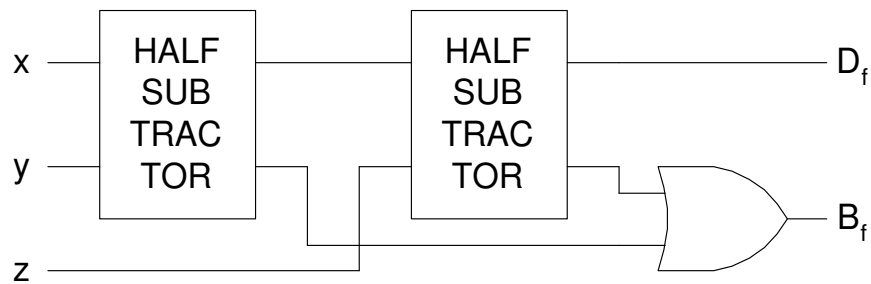
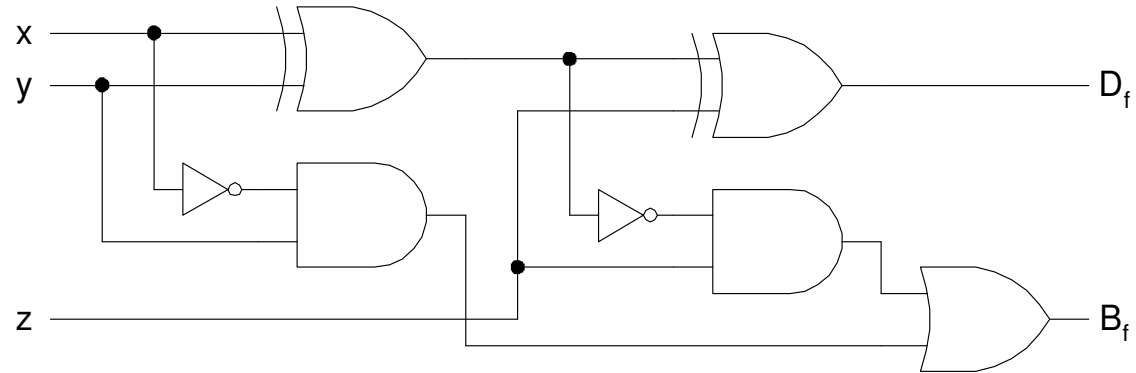
0 - 0 = 0
0 - 1 = 1 → borrow = 1
1 - 0 = 1
1 - 1 = 0

x	y	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0



Rangkaian Pengurang Penuh

x	y	z	D	B
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



Pengubah Kode BCD ke XS-3

BCD				XS-3			
a	b	c	d	P	Q	R	S
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

	ab				
cd		00	01	11	10
00				x	1
01			1	x	1
11			1	x	x
10			1	x	x

P

	ab				
cd		00	01	11	10
00			1	x	
01		1		x	1
11		1		x	x
10		1		x	x

Q

	ab				
cd		00	01	11	10
00		1	1	x	1
01				x	
11		1	1	x	1
10				x	

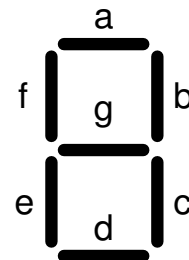
R

	ab				
cd		00	01	11	10
00		1	1	x	1
01				x	
11				x	x
10		1	1	x	x

S

Pengubah Kode BCD ke 7-Segmen

BCD				7-Segmen							Tampilan
D	C	B	A	a	b	c	d	e	f	g	
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	2
0	0	1	1	1	1	1	1	0	0	1	3
0	1	0	0	0	1	1	0	0	1	1	4
0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	0	0	1	1	1	1	1	6
0	1	1	1	1	1	1	0	0	0	0	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	1	0	0	1	1	9



DC	BA			
	00	01	11	10
00	1		1	1
01		1	1	
11	x	x	x	x
10	1	1	x	x

a

	BA				
DC		00	01	11	10
00		1	1	1	1
01		1		1	
11		x	x	x	x
10		1	1	x	x

b

	BA				
DC		00	01	11	10
00		1	1	1	
01		1	1	1	1
11		x	x	x	x
10		1	1	x	x

c

	BA				
DC		00	01	11	10
00		1		1	1
01			1		1
11		x	x	x	x
10		1		x	x

d

	BA				
DC		00	01	11	10
00		1			1
01					1
11		x	x	x	x
10		1		x	x

e

	BA				
DC		00	01	11	10
00		1			
01		1	1		1
11		x	x	x	x
10		1	1	x	x

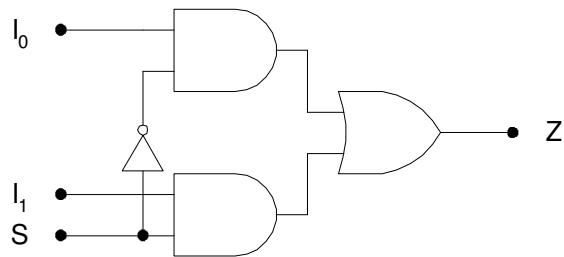
f

	BA				
DC		00	01	11	10
00				1	1
01		1	1		1
11		x	x	x	x
10		1	1	x	x

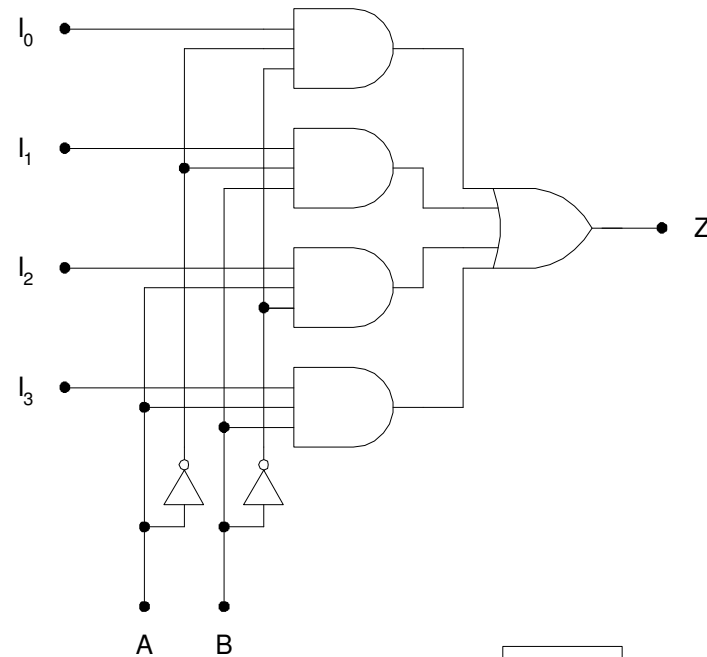
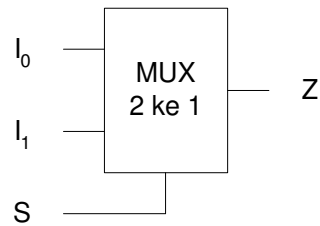
g

Multiplexer

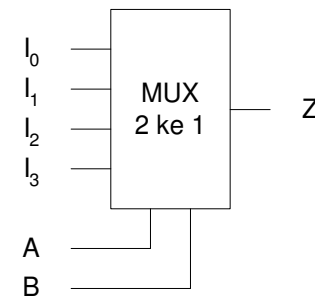
Berfungsi untuk memilih salah satu dari masukan



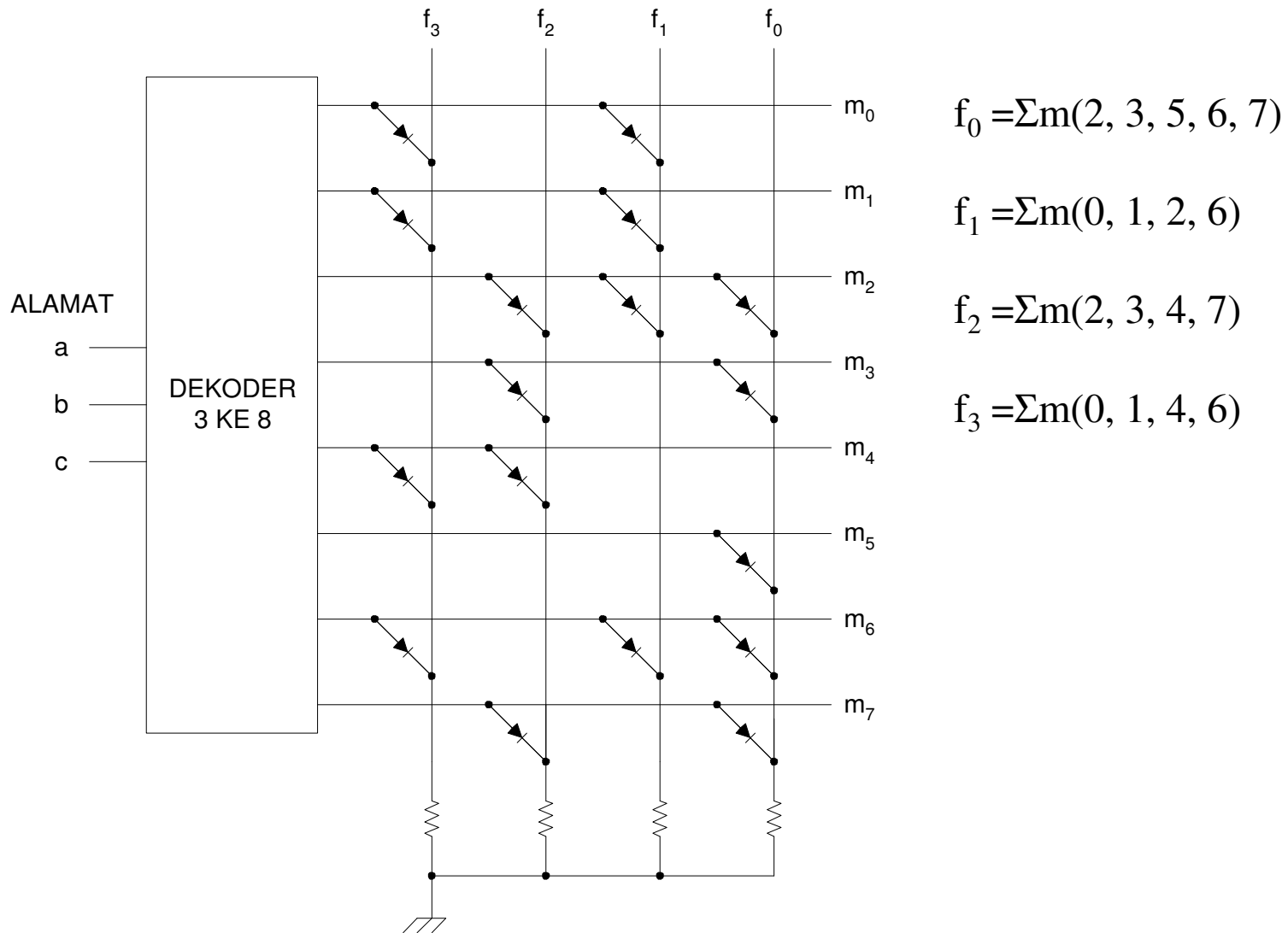
S	Z
0	I_0
1	I_1

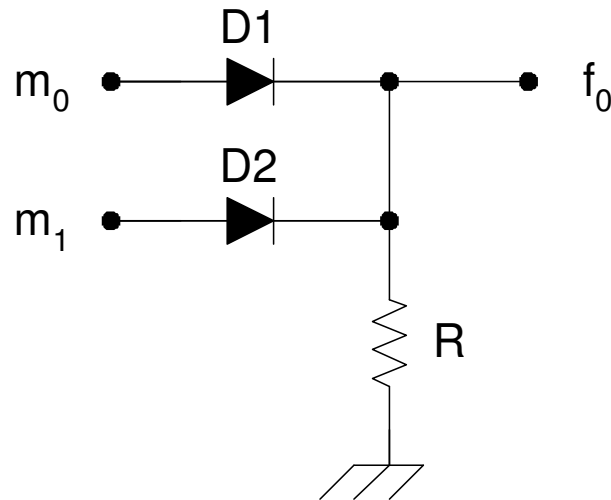


A	B	Z
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3



Realisasi dengan EPROM





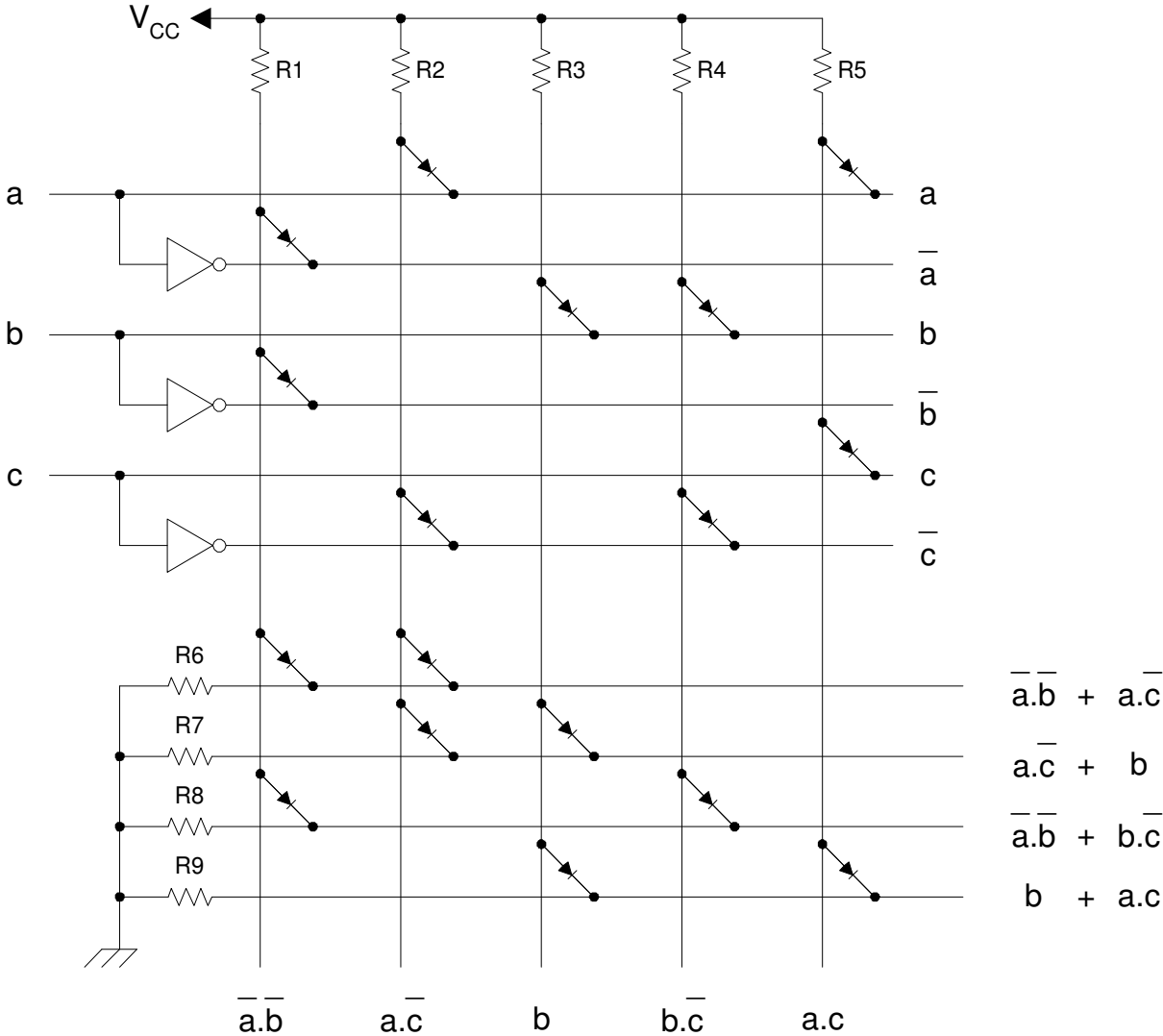
Dioda D1 dan D2 membentuk gerbang OR.

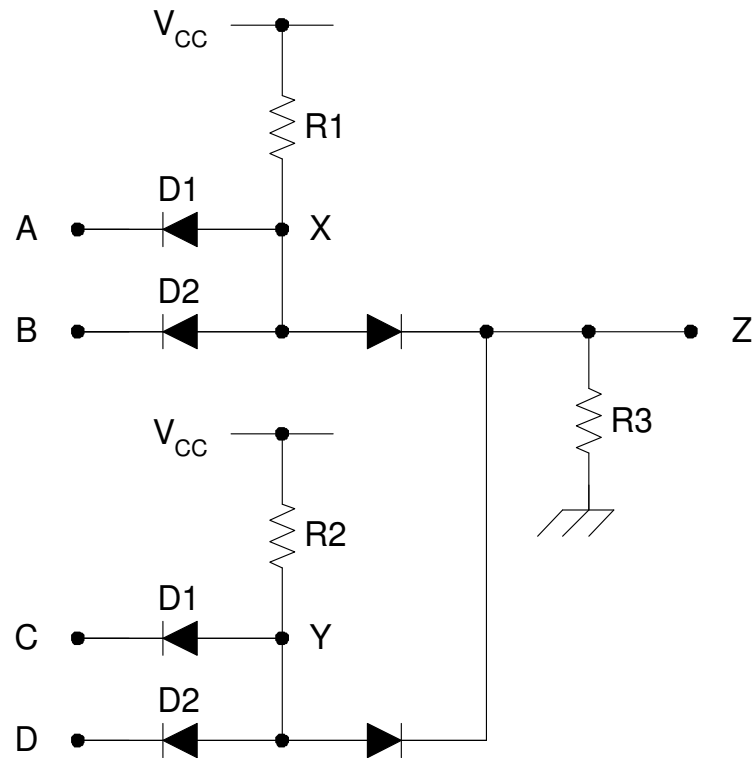
Jika logika m_0 tinggi dan logika m_1 rendah maka D1 akan menghantar dan D2 reverse sehingga logika pada titik C tinggi.

Sebaliknya jika logika m_1 tinggi dan logika m_0 rendah maka D2 menghantar dan D1 reverse sehingga logika pada titik C tinggi.

Logika f_0 akan tinggi jika m_0 atau m_1 atau keduanya berlogika tinggi.

Realisasi dengan PLA





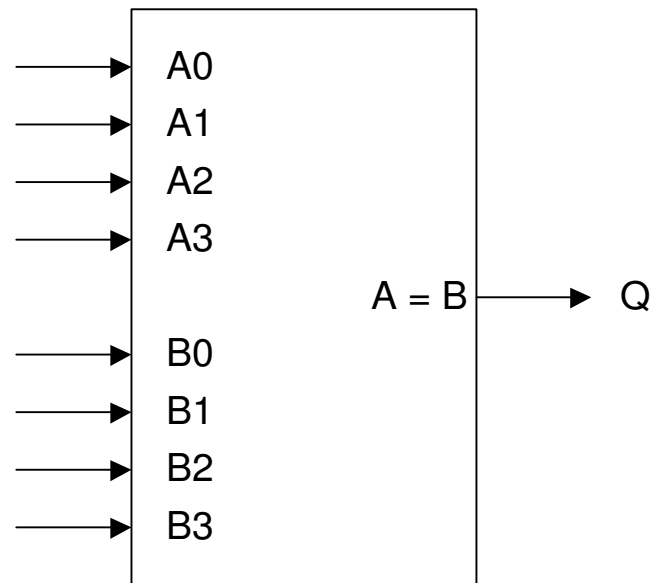
$$\begin{aligned}
 X &= A.B \\
 Y &= C.D \\
 Z &= X + Y \\
 &= A.B + C.D
 \end{aligned}$$

Logika titik X akan tinggi jika logika titik A dan titik B keduanya tinggi.

Logika titik X akan rendah jika salah satu dari titik A atau titik B atau keduanya bernilai rendah.

Rangkaian Pembanding Logika

Masukan A dibandingkan dengan masukan B. Jika sama maka keluaran Q akan tinggi.



Keluaran $Q = 1$ jika $A_0 = B_0$ dan $A_1 = B_1$ dan $A_2 = B_2$ dan $A_3 = B_3$

Persamaannya adalah :

$$Q = (A_0 \oplus B_0).(A_1 \oplus B_1).(A_2 \oplus B_2).(A_3 \oplus B_3)$$

Rangkaiannya menjadi

