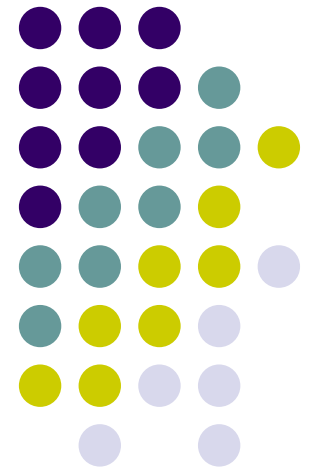


# LOGIKA MESIN



# PENDAHULUAN



- Data dan instruksi ditransmisikan diantara berbagai bagian prosesor atau diantara prosesor dan peripheral dgn menggunakan **PULSE TRAIN**.
- Berbagai tugas dijalankan dgn cara menyampaikan pulse train melalui switch-switch elektronik yg disebut **GATES**.
- Tiap gates adalah sirkuit elektronis yg mungkin mempunyai ketentuan utk menerima atau mengirimkan beberapa pulsa sekaligus
- Tiap gates bisa dianggap sebagai kotak hitam yg mengontrol arus pulsa dgn cara tertentu.
- Tiap gates biasanya ,menjalankan beberapa fungsi sederhana misalnya AND, OR, NOT, dan karena itu gate sering disebut **ELEMEN LOGIKA**.



# LATAR BELAKANG



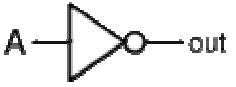
- Dalam komputer modern sejumlah besar komponen dibuat dari bagian kecil bahan semi konduktor yg disebut **chip silikon**. Satu chip berisi beberapa komponen, dan karena sifatnya yg elektronis dapat dianggap sebagai **SIRKUIT TERPADU (IC)**
- Chip dapat diklasifikasikan menurut ukurannya :
  - Small scale integration (SSI) : berisi paling banyak 10 gate
  - Medium scale integration (MSI) : berisi antara 10 & 100 gate
  - Large scale integration (LSI) : berisi antara 100 & 100.000 gate
  - Very large scale integration (VLSI): berisi lebih dari 100.000 gate



# DIAGRAM LOGIKA

- Dalam diagram logika gate direpresentasikan dgn **simbol**, dan input dan output direpresentasikan dgn **garis panah** yg dilabeli dgn huruf.

Tabel simbol logika

Operator	Operasi	Simbol
AND	$X.Y$	
OR	$X + Y$	
NOT	$\overline{X}$	

# MEMECAHKAN MASALAH DALAM DESAIN LOGIKA



- Dalam merancang sirkuit logika tujuannya adalah utk mengurangi jumlah gate yg digunakan, sehingga akan mengurangi biaya.
- Langkah dalam pemecahan :
  - a. Deskripsikan masalah yg telah ditetapkan dgn baik
  - b. Pilih simbol utk merepresentasikan input dan output
  - c. Buat tabel kebenaran menurut simbol
  - d. Buat pernyataan logis dari tabel kebenaran yg memberi variabel output berdasarkan variabel input
  - e. Sederhanakan pernyataan output dgn peta karnaugh atau aljabar
  - f. Buat diagram logika utk output yg disederhanakan.

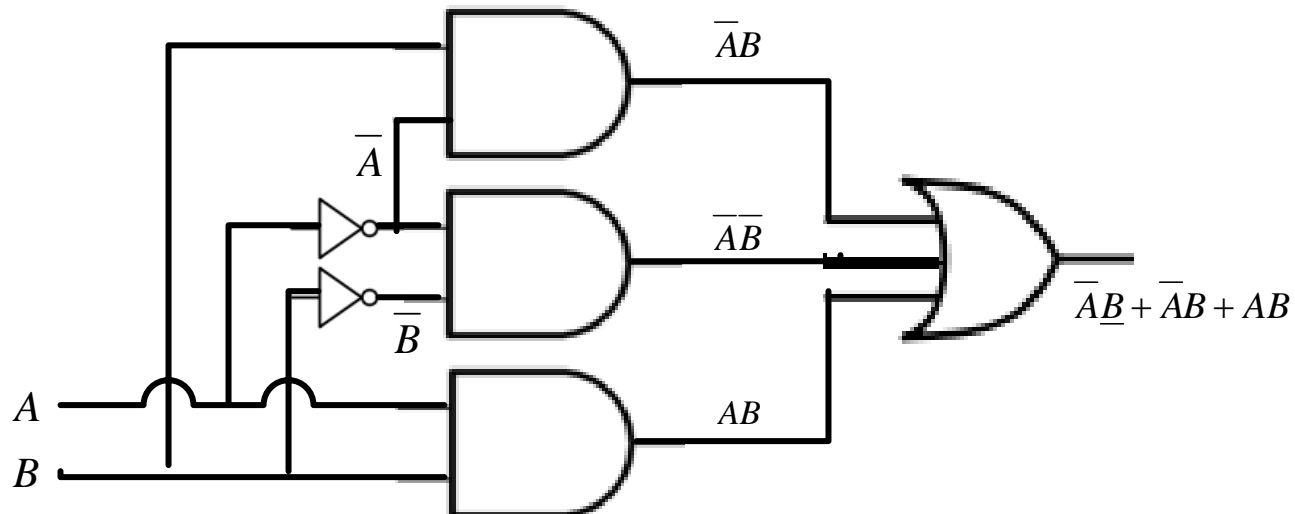
Contoh :  $Q = \bar{A}\bar{B} + \bar{A}B + A.B$



## PENYELESAIAN :

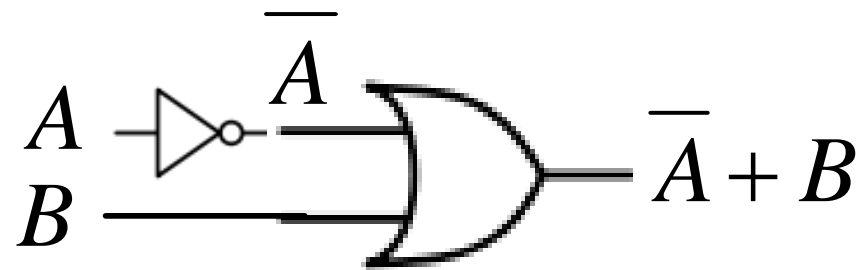
### •Karnaugh

	A	$\bar{A}$
B	1	1
$\bar{B}$		1



- Aljabarik

$$\begin{aligned} Q &= \overline{A}.\overline{B} + \overline{A}.B + A.B \\ &= \overline{A}.\overline{(B + B)} + B.\overline{(A + A)} \\ &= \overline{A} + B \end{aligned}$$





# PENAMBAHAN 2 BIT

4 penambahan yg mungkin direpresentasikan dalam tabel kebenaran, 2 inputnya X dan Y, dan outputnya S utk SUM dan C utk CARRY.

X	Y	SUM S	CARRY C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Dari tabel dapat dilihat :

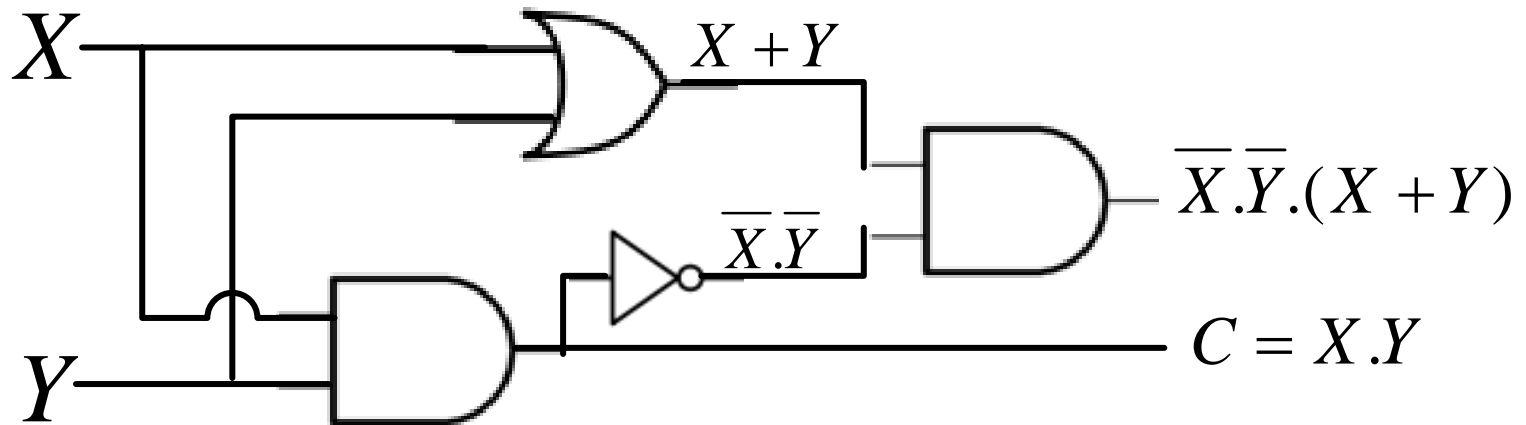
$$C = X.Y \text{ dan } S = \overline{X}.Y + X.\overline{Y}$$





# HALF ADDER

gate dapat dikombinasikan utk menghasilkan perangkat yg memberi output SUM maupun CARRY, perangkat ini disebut **HALF ADDER**





# ADDER (FULL ADDER)

dapat menangani carry yg berasal dari SUM 2 bit sebelumnya dan mempunyai 3 input

Input				output
P	X	Y	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

Tabel kebenaran disamping mendefinisikan operasi ADDER Inputnya adalah X,Y dan CARRY nya berasal dari SUM P sebelumnya. Outputnya adalah S utk SUM dan C utk CARRY



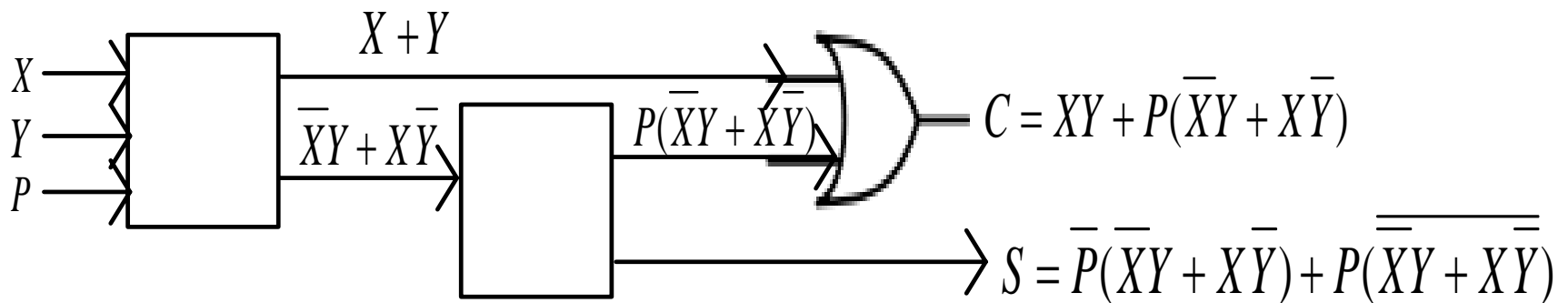
# ADDER (FULL ADDER)

$$S = \bar{P}.\bar{X}.Y + \bar{P}.X.\bar{Y} + P.\bar{X}.\bar{Y} + P.X.Y$$

$$S = \bar{P}.(X.Y + X.\bar{Y}) + P.(X.\bar{Y} + X.Y)$$

$$C = \bar{P}.X.Y + P.\bar{X}.Y + P.X.\bar{Y} + P.X.Y$$

$$C = X.Y.(P + P) + P.(X.Y + X.\bar{Y}) = X.Y + P.(X.Y + X.\bar{Y})$$





# PENAMBAHAN PARALEL & SERIAL

## PENAMBAHAN PARALEL

kolom-kolom ditambahkan sisi demi sisi sekaligus

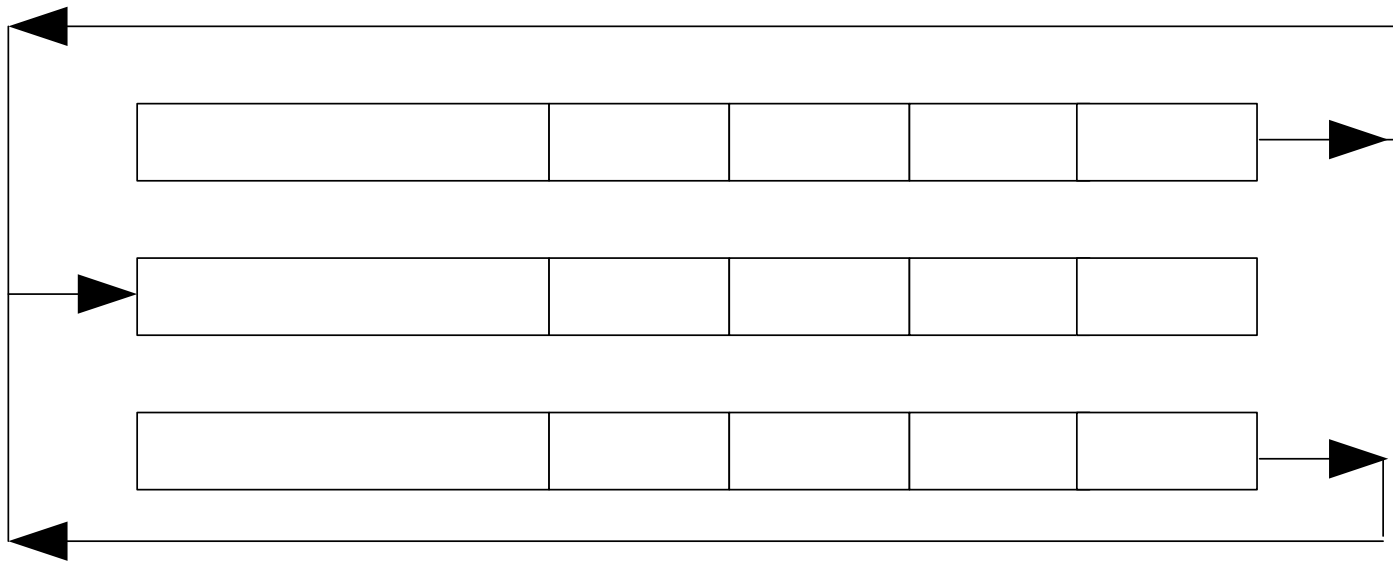
First number	0	1	1	1
	▼	▼	▼	▼
SUM	1	0	0	1
	▲	▲	▲	▲
Second number	0	0	1	0



# PENAMBAHAN PARALEL & SERIAL

## PENAMBAHAN SERIAL

kolomnya ditambahkan sekali tiap waktu carry-nya disangga dulu dan digunakan dalam penambahan kolom berikutnya



# REGISTER



data yg diinputkan ke adder harus disangga dalam register yg tepat

- a. Dalam **register paralel** semua bit dari register ke adder paralel secara serentak
- b. Dalam **register serial** semua bit dari register diterapkan ke adder serial sekaligus tiap waktu. Bit yg tersisa dalam register digeser ke ujung tiap kali sampai ia di outputkan

# ADDER SERIAL



- a. **Satu full adder**, menerima pasangan bit  $x, Y$  dari 2 register serial yg dimulai dgn the least significant
- b. **Penyimpanan delay**, penyimpanan biner yg menyangga bit carry sampai diperlukan utk penambahan berikutnya. Penyimpanan di set ke nol sebelum penambahan pertama

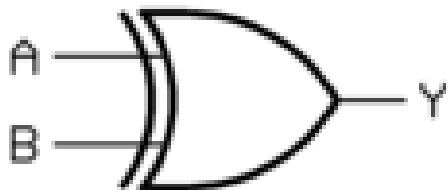


# KOMPARATOR

Exclusive OR dan ekuivalensi merupakan contoh komparator (XOR)

X	Y	$X \neq Y$
0	0	0
0	1	1
1	0	1
1	1	0


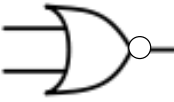
X	Y	$X \equiv Y$
0	0	1
0	1	0
1	0	0
1	1	1



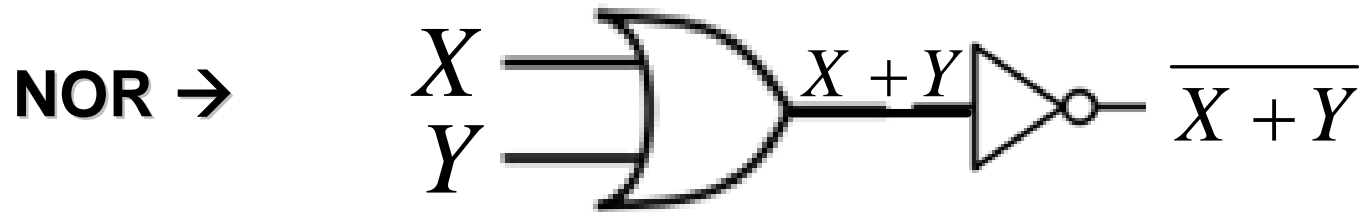
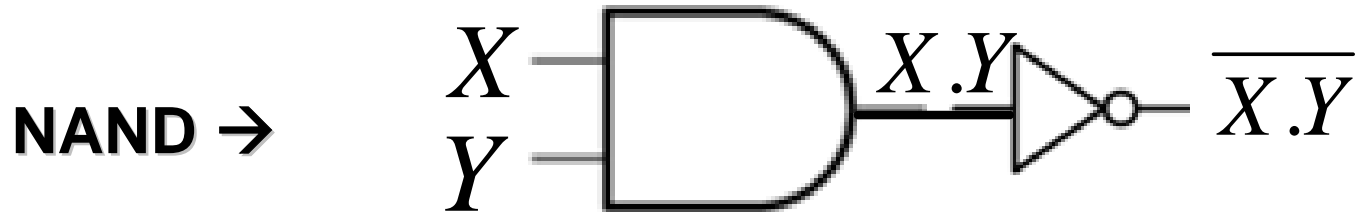




# METODE NAND & NOR

OPERATOR	OPERASI	SIMBOL	TABEL KEBENARAN		
			X	Y	
NAND	$\overline{X.Y}$		X	Y	
			0	0	1
			0	1	1
			1	0	1
			1	1	0
NOR	$\overline{X.+Y}$		X	Y	
			0	0	1
			0	1	0
			1	0	0
			1	1	0

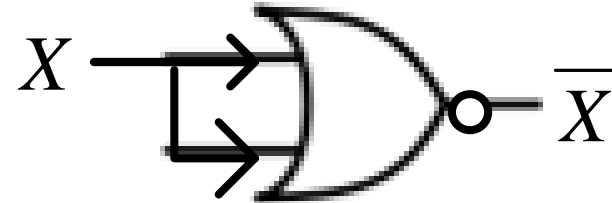
# REPRESENTASI AND-OR-NOT dari NAND dan NOR



# REPRESENTASI NAND & NOR STANDAR

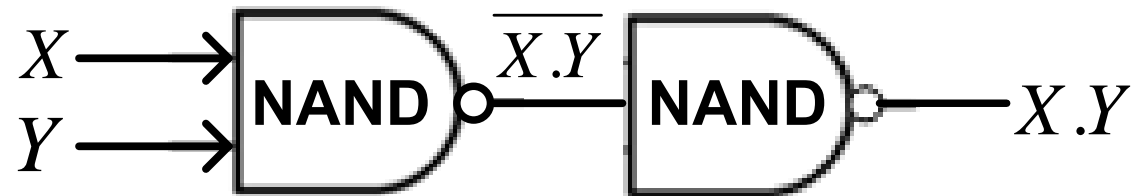


## a. NOT



## b. AND

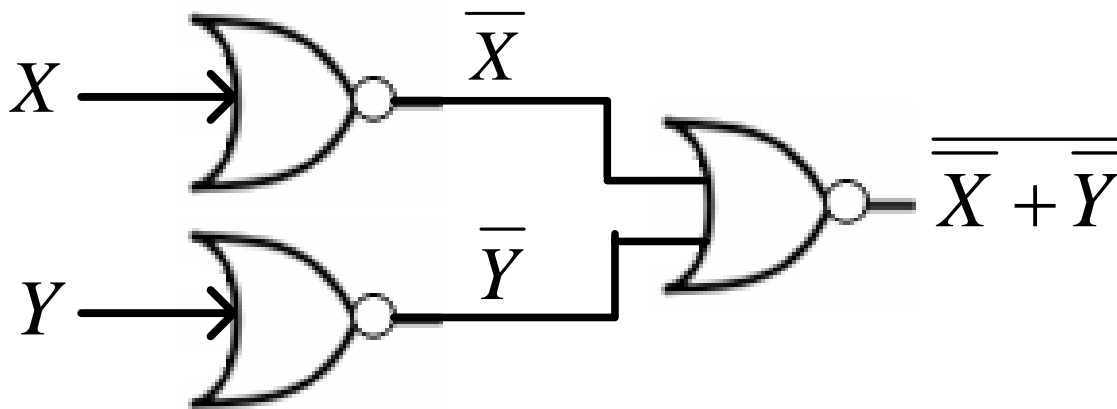
### 1. dengan NAND



# REPRESENTASI NAND & NOR STANDAR



2. dengan NOR  $\rightarrow X.Y = \overline{\overline{X} + \overline{Y}}$

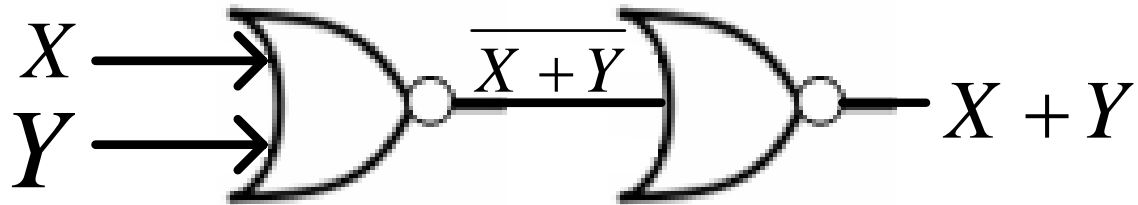


# REPRESENTASI NAND & NOR STANDAR

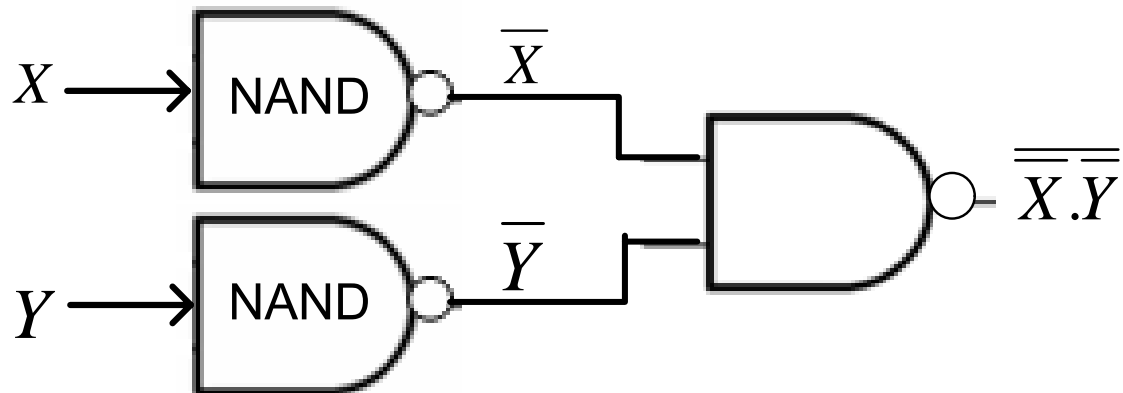


## c. OR

### 1. dengan NOR



### 2. dengan NAND $\rightarrow X + Y = \overline{\overline{X} \cdot \overline{Y}}$



# FLIP-FLOP (BISTABLE MULTIVIBRATOR / BISTABLE) DAN LATCH



Flip-flop adalah perangkat 2 state, mempunyai 2 state pengoperasian stabil yg bersesuaian dgn 0 dan 1 biner.

## LATCH

Flip flop yg state-state output tergantung pada tingkat sinyal input (0 atau 1).

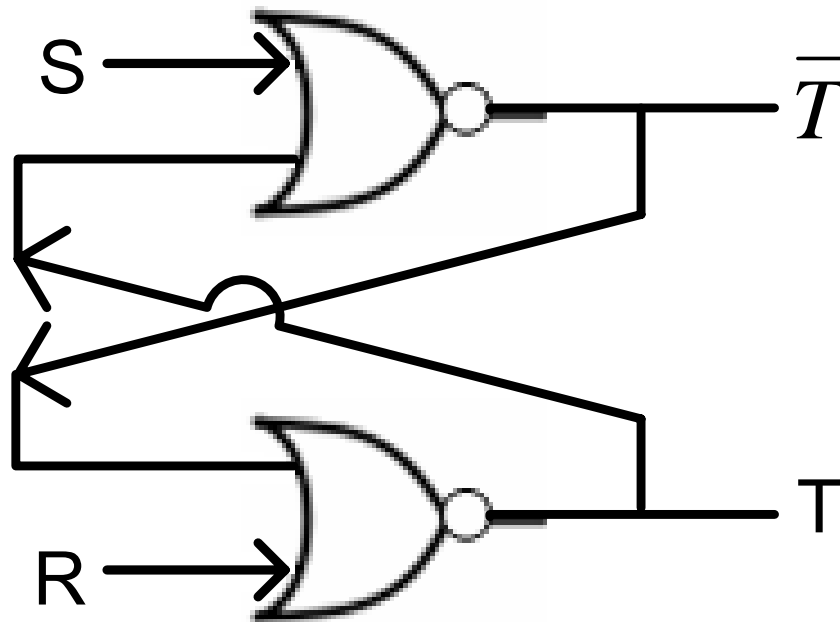
## EDGE-TRIGGERED FLIP-FLOP

Flip flop yg state-state output tergantung pada transisi dalam tingkatan (0 ke 1 atau 1 ke 0)

# FLIP-FLOP (BISTABLE MULTIVIBRATOR / BISTABLE) DAN LATCH



Jenis flip-flop yg umum adalah flip flop set=reset yg mempunyai 2 input dimana yg satu memuat 1 (set) yg satunya membuat output 0 (reset)

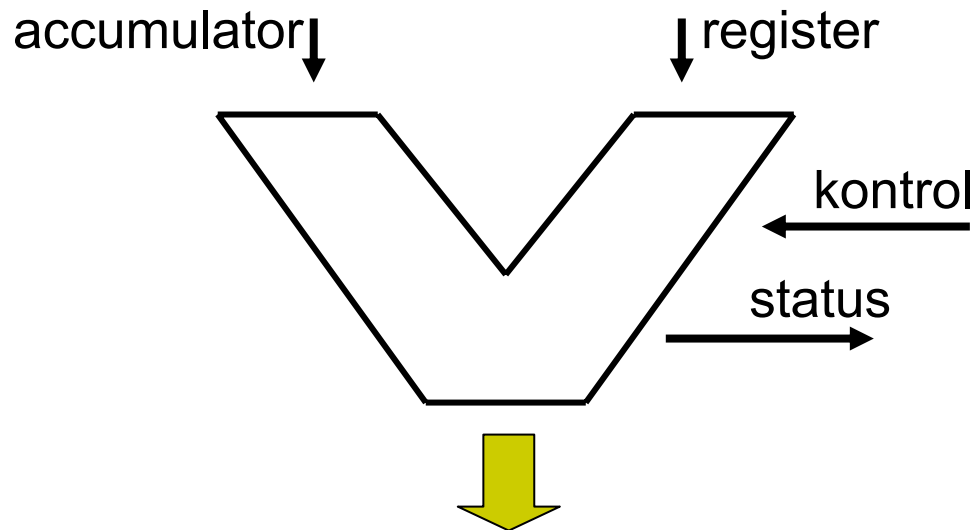




# ALU & CONTROL

ALU terbentuk dari sejumlah besar kombinasi gate, namun biasanya dirancang secara menyeluruh agar dapat bekerja secara lebih efisien

Representasi diagramatis biasa utk ALU







# PENKODEAN & PENDEKODEAN

Input										Output			
0	1	2	3	4	5	6	7	8	9	W	X	Y	Z
1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	1
0	0	0	0	0	0	1	0	0	0	0	1	1	0
0	0	0	0	0	0	0	1	0	0	0	1	1	1
0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	1	0	0	1

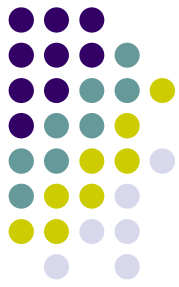
1 = HIGH SIGNAL      0 = LOW SIGNAL

Input

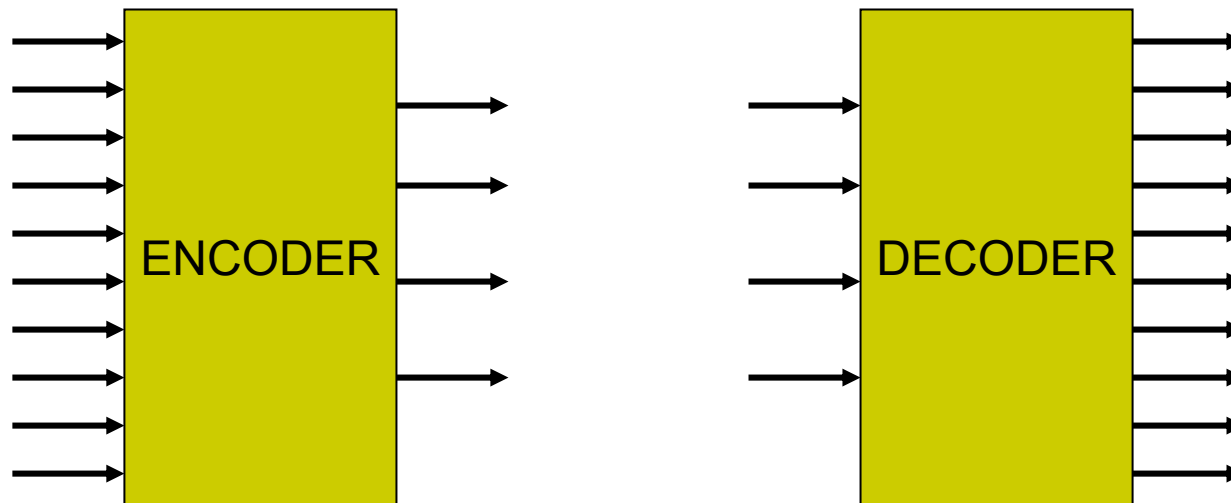
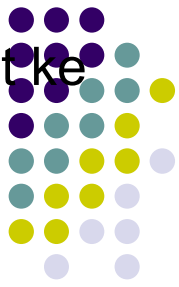
W	Y	Z
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Input

0	1	2	3	4	5	6	7
1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1



- Pengkodean melibatkan pengkonversian satu sinyal sari satu set ke dalam kelompok sinyal dari set lain. Perangkat yg menjalankan tugas ini adalah **ENCODER**  
    contoh : perangkat keyboard
- Pendekodean adalah operasi yg berlawanan dari pengkodean dan perangkat yg menjalankan tugas ini disebut **DECODER**.
- Jika  $n$  wire yg disambungkan, maka menjulur  $2^n$ , maka hanya satu yg akan berupa logika 1



# INTERFACING



Interface standar adalah spesifikasi sinyal-sinyal logika, koneksi elektrik dan aturan logika dari bagian interface prosesor. Interface adalah bagian dari hardware yg mengkonversi ke salah satu sisinya tempat interface standar, sedang prosesornya dikoneksikan ke sisi interface yg satunya

Pengkonversian dapat dilakukan dgn mengambil alih bagian dari pekerjaan pengontrolan perangkat, tempat dimana interface dikoneksikan interfacing memberikan fleksibilitas dan meningkatkan efisiensi.

# INTERFACING



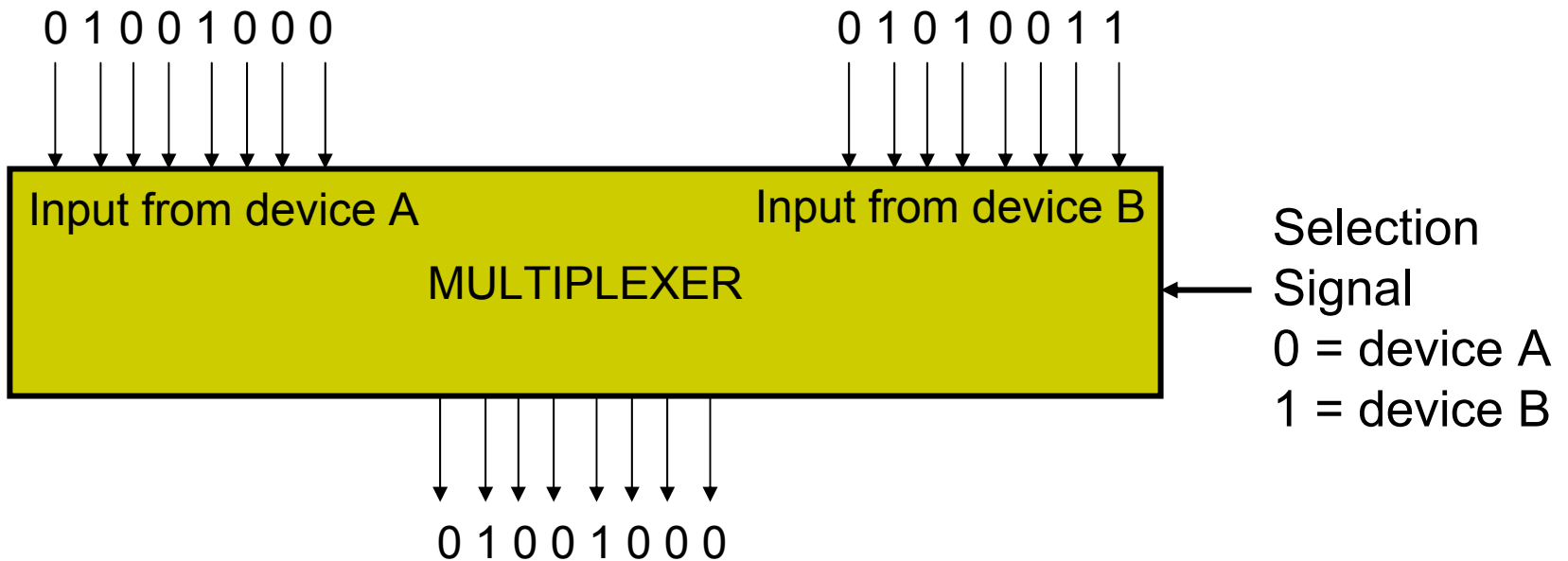
Chip-chip yg umum digunakan :

- **UART – universal asynchronous receiver transmitter**  
Menangani komunikasi dua arah antara komputer dgn perangkat serial
- **USART - universal synchronous asynchronous receiver transmitter**  
Menangani pentransmisi blok data secara sinkron antara perangkat dan komputer
- **PIO – parallel input/output**  
menangani I/O antara komputer dan perangkat dan menerima data secara paralel



# MULTIPLEXER

mekanisme seleksi utk menseleksi arus data dan instruksi di dalam komputer modern



# LOGIKA KOMBINASIONAL & BERANGKAI



- **Gate logika kombinasional**

mempunyai output yg sepenuhnya ditetapkan oleh kombinasi input.

contoh : gate AND,OR,NOT

- **Gate logika berangkai**

outputnya tidak hanya bergantung pada kombinasi input namun juga pada rangkaian tempat terjadinya output

contoh : flip-flop

# TINJAUAN SEDERHANA TERHADAP SISTEM BERDASARKAN MIKROKOMPUTER



- dalam sistem **mikroprosesor 16 bit**, alamat akan mempunyai lebar 24 bit, namun pada mikroprosesor yg lebih modern bus data internalnya akan selebar 32 bit utk mencocokkan diri dgn ukuran register internal
- dalam **mikroprosesor 8 bit** biasa, bus alamat akan mempunyai luas 16 bit dan bus datanya hanya 8 bit. Register internal biasanya mempunyai luas 8 bit kecuali MAR dan PC(SCR) yg mempunyai luas 16 bit





# PENGGORGANISASIAN MEMORY

- Mikroprosesor 1 bit biasa mempunyai ukuran memory maksimum teoritis yg cukup besar, hingga tidak menimbulkan kendala praktis
- Mikroprosesor 8 bit akan mempunyai ukuran memory maksimum 64 K word (65,36 lokasi memory) yg masing-masing akan direpresentasikan oleh salah satu dari 64 K kode

# MICROCODE MACHINE



- Dimulai pada point ketika pada tingkat konvensional mesin, instruksi berikutnya yg akan di eksekusi baru saja dimuatkan ke dalam IR (CIR). Bagian instruksi yg fungsinya menetapkan operasi yg akan dijalankan yg disebut **function code** atau **op code di decode** utk memperoleh alamat awal
- Multiplexer menyampaikan alamat ke MSCR
- Instruksi microcode berikutnya yg disangga pada alamat yg ditetapkan dalam MSCR dimuatkan ke dalam MMDR
- Tiap instruksi microcode menyangga kombinasi sinyal-sinyal kontrol tertentu. Ketika MCIR menerima suatu pulsa maka sinyal-sinyal ini akan ditransmisikan
- Di dalam MICR terdapat alamat utk instruksi microcode berikutnya
- Instruksi microcode akan dieksekusi secara berangkai menurut isi MSCR yg sedang dinaikkan sebesar 1 dan sedang dimuatkan ke dalam MSCR melalui multiplexer.