

# I. DASAR RANGKAIAN SEKUENSIAL

## Tujuan :

1. Memahami perbedaan antara rangkaian kombinasional dan sekuensial
2. Mengerti State Diagram
3. Mengerti maksud dan tujuan Elemen Penyimpan Biner
4. Dapat membuat SR Flip-flop dari gerbang NOR
5. Dapat membuat SR Flip-flop dari gerbang NAND
6. Mengerti Elemen Penyimpan dengan Clock
7. Dapat melakukan Analisa Rangkaian Sekuensial
8. Dapat melakukan Sintesa Rangkaian Sekuensial

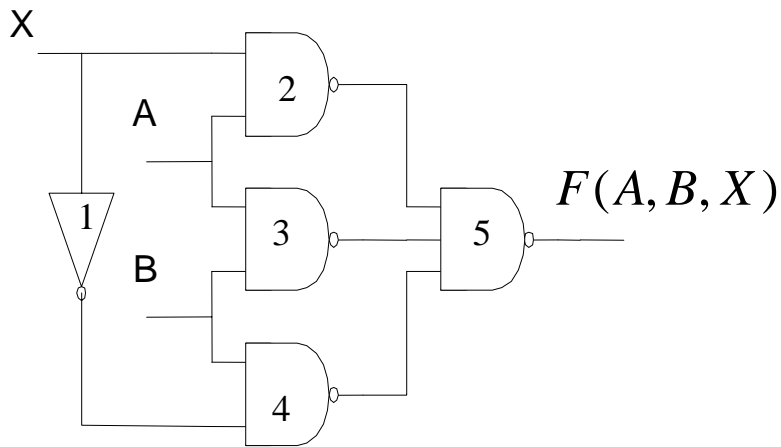
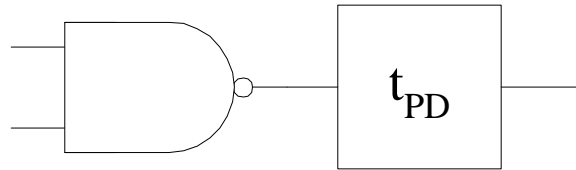


a) Blok Diagram Rangkaian Kombinasional

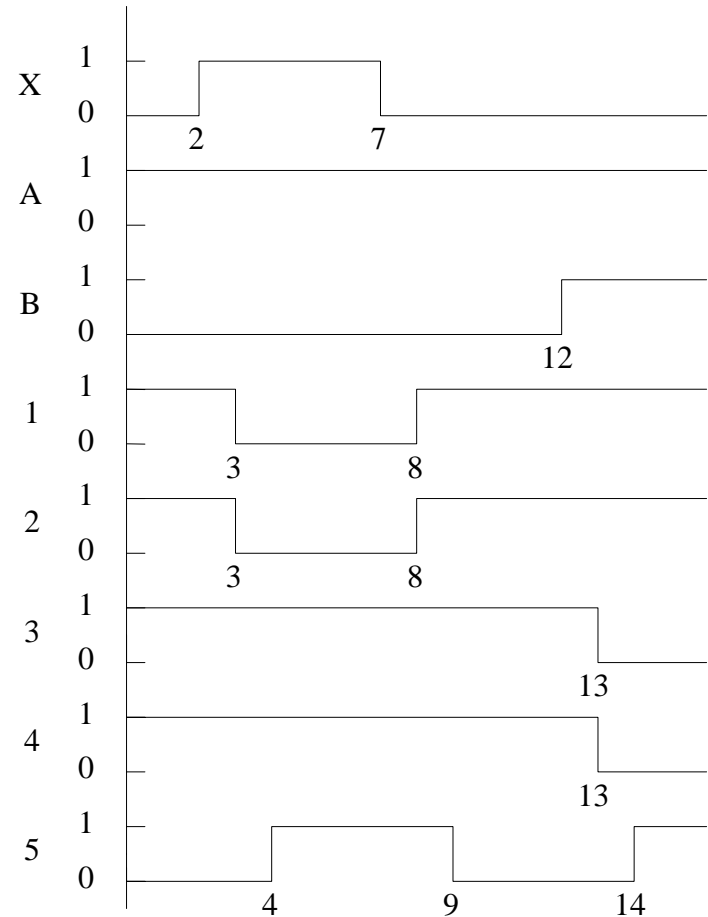


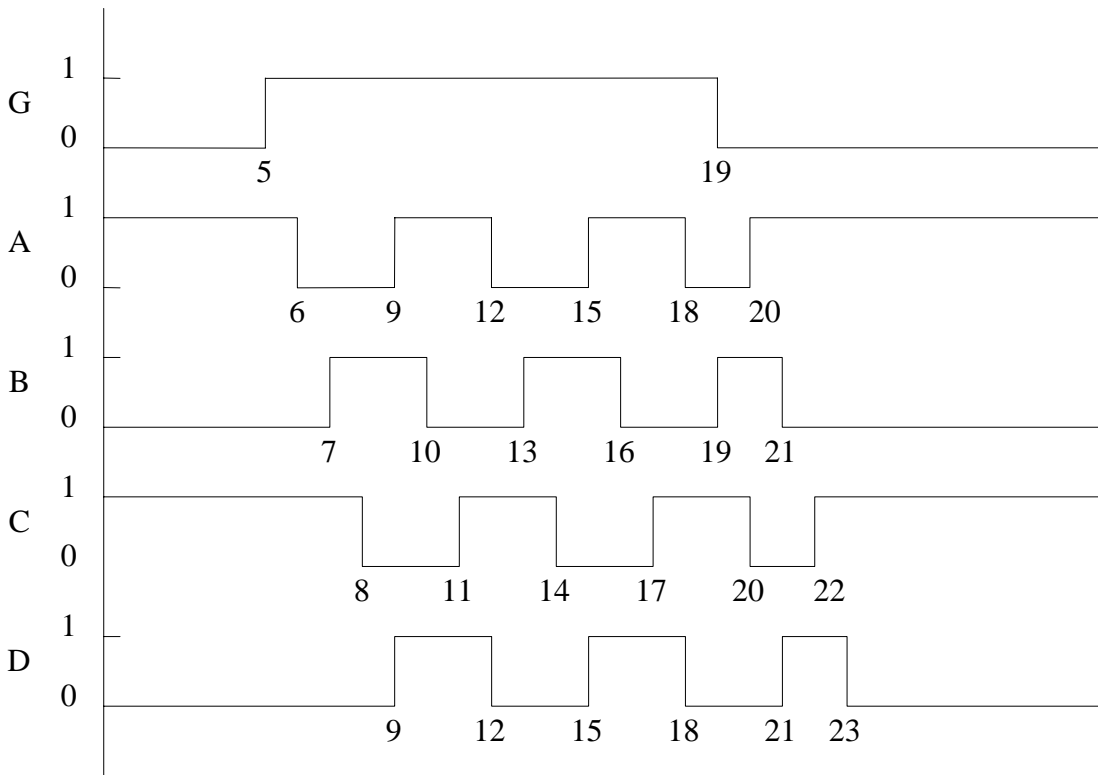
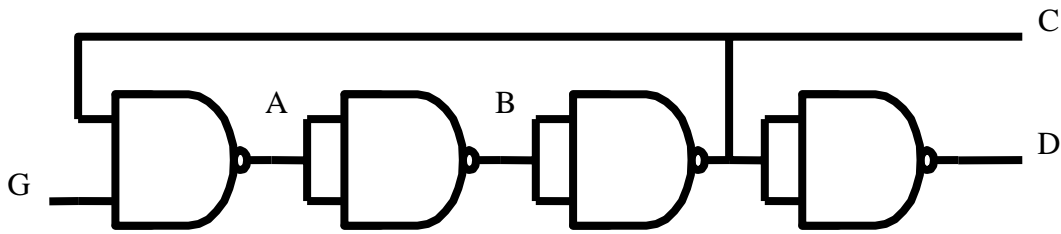
b) Blok Diagram Rangkaian Sekuensial

# Fungsi Delay pada Elemen Penyimpanan

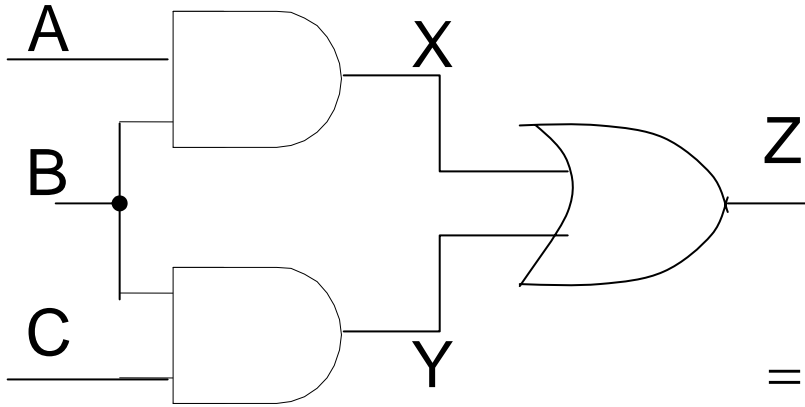


$$f(A, B, X) = \overline{\overline{XA} \cdot \overline{AB} \cdot \overline{BX}}$$





## OSILASI DARI 4 BUAH GERBANG NAND

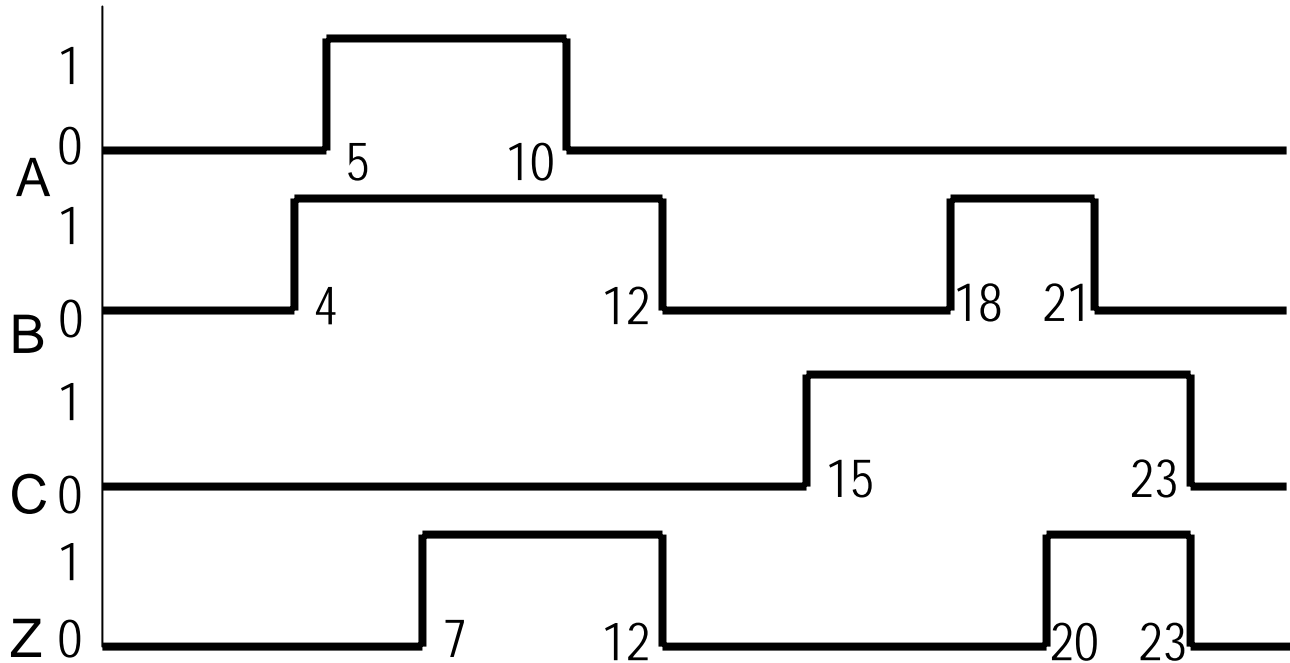


$$X(t + \Delta) = A(t).B(t)$$

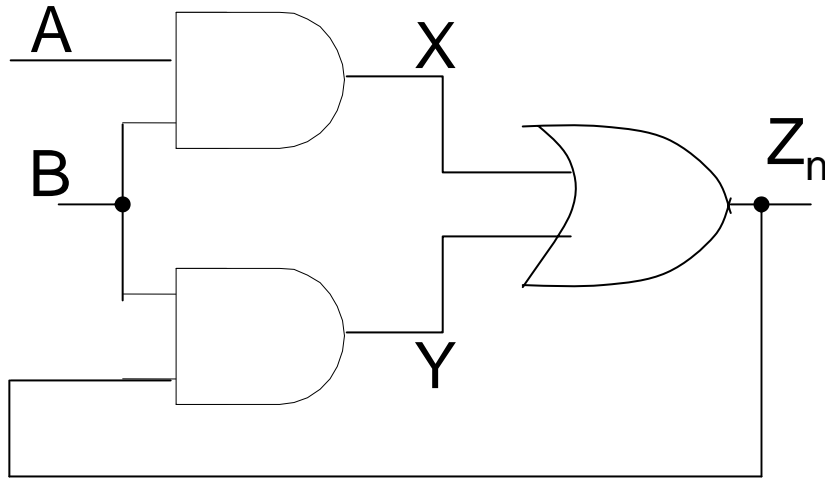
$$Y(t + \Delta) = B(t).C(t)$$

$$Z(t + 2\Delta) = X(t + \Delta) + Y(t + \Delta)$$

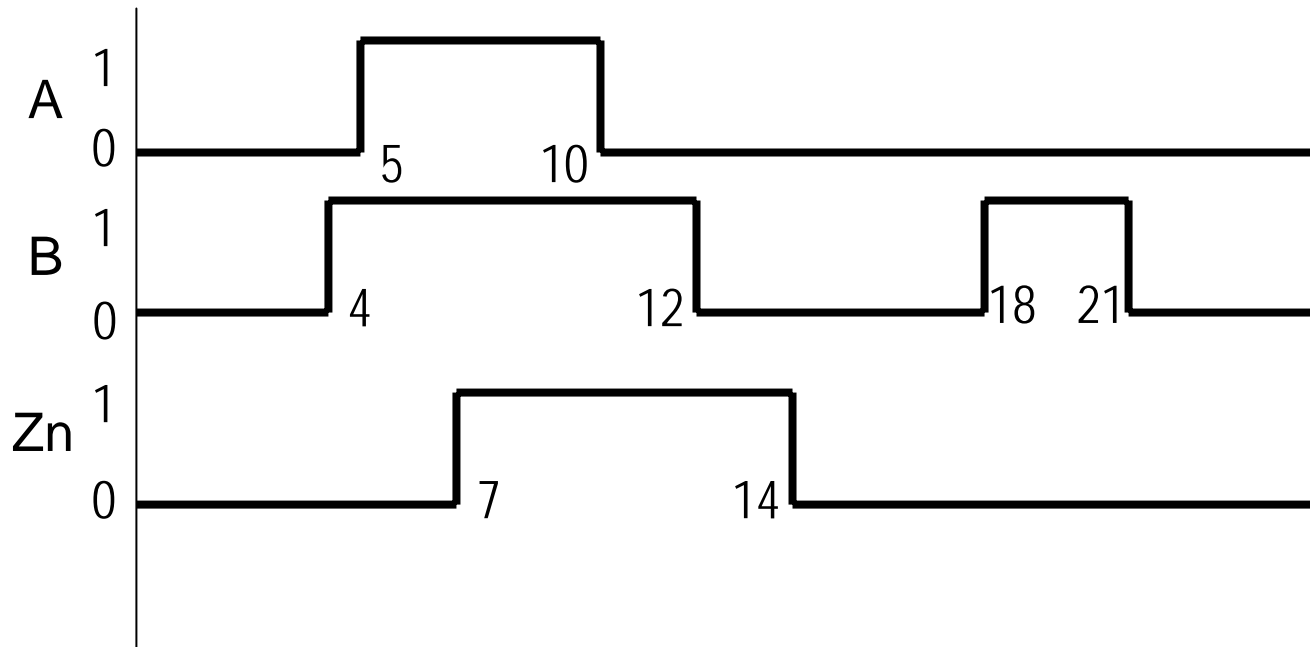
$$= A(t)B(t) + B(t)C(t) = B(t)[A(t) + C(t)]$$



## RANGKAIAN KOMBINASIONAL & TIMING DIAGRAM



$$Z(t + 2\Delta) = X(t + \Delta) + Y(t + \Delta) \\ = B(t)[A(t) + Z(t)]$$



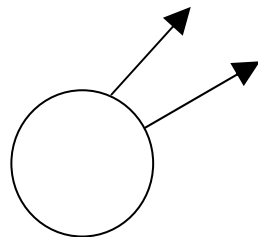
## RANGKAIAN SEKUENSIAL & TIMING DIAGRAM

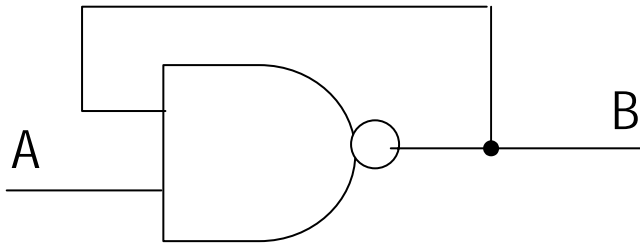
# STATE DIAGRAM

- Menggambarkan perubahan kondisi sebuah variabel (output) dari kondisi sekarang (*Present State*) ke kondisi berikutnya (*Next State*).
- Kondisi tersebut berubah karena adanya pengaruh input dari luar

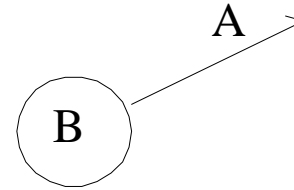
State diagram terdiri dari

- a. **Lingkaran (node)** yang jumlahnya satu untuk tiap-tiap keadaan.
- b. **Anak panah transisi**, yang meninggalkan tiap-tiap keadaan dan berakhir pada keadaan berikutnya.

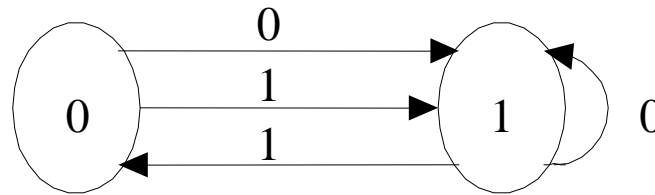




Rangkaian sekuensial gerbang NAND

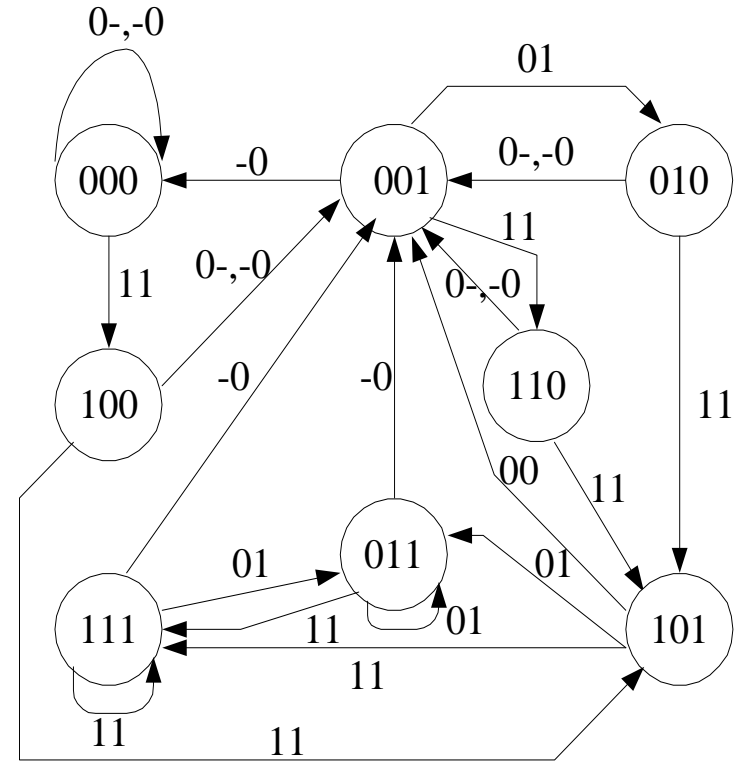
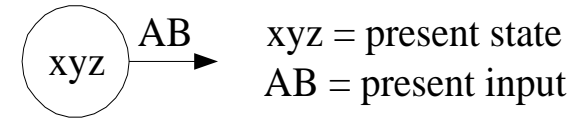
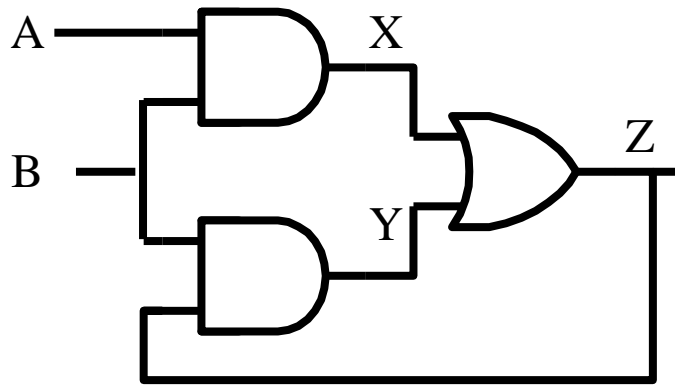


A = Present input    B = Present output



State Diagram untuk rangkaian NAND di atas





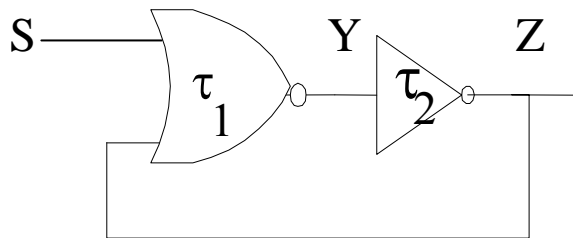
Tabel PS/NS utk rangkaian di atas

Present State X(t)Y(t)Z(t)	Next State X(t + τ).Y(t + τ). Z(t + τ)			
	Present input A(t). B(t)			
	00	01	10	11
000	000	000	000	100
001	000	010	000	110
010	001	001	001	101
011	001	011	001	111
100	001	001	001	101
101	001	011	001	111
110	001	001	001	101
111	001	011	001	111

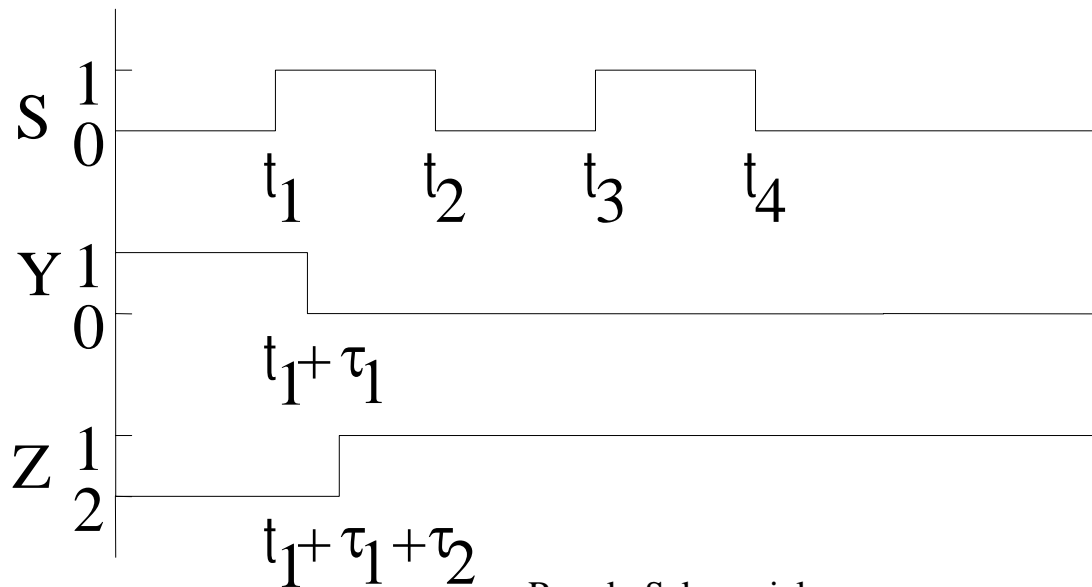
State diagram utk rangkaian di atas

# ELEMEN PENYIMPAN BINER

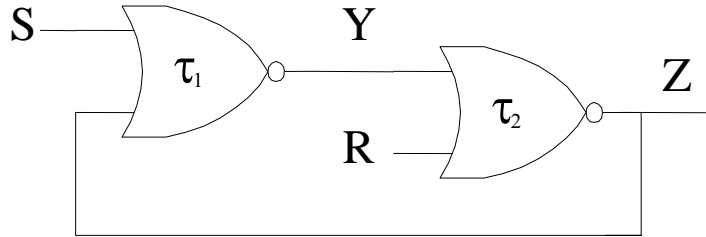
Bentuk sederhana dari elemen penyimpan biner adalah sebuah rangkaian yang dapat mengingat sebuah sinyal biner sebelumnya, terutama nilai logika.



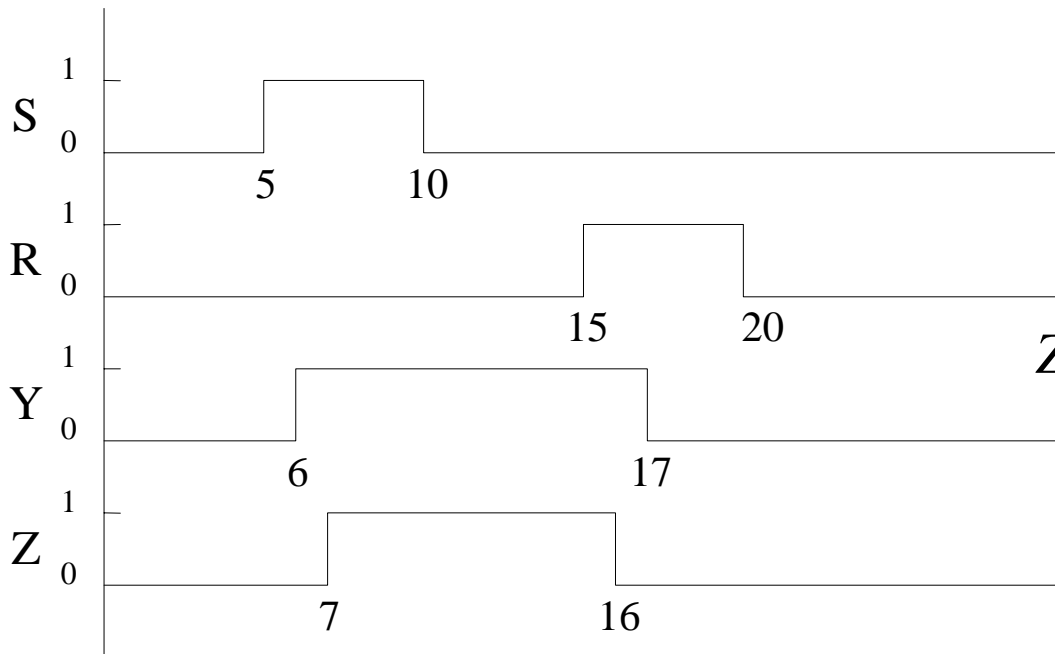
$$\begin{aligned} Z(t) &= 0, t \leq 0 \\ S(t) &= 0, t \leq 0 \\ Y(t) &= 0, t \leq 0 \end{aligned}$$



# NOR SR FLIP-FLOP



$S(t) = 0$   
 $Y(t) = 1$   
 $R(t) = 0$   
 $Z(t) = 0, t \leq 0$   
 $\tau_1 = \tau_2 = 1$  unit



$$Y(t + \tau_1) = \overline{Z(t) + S(t)}$$

$$Z^*(t + \tau_2) = \overline{Y(t) + R(t)}$$

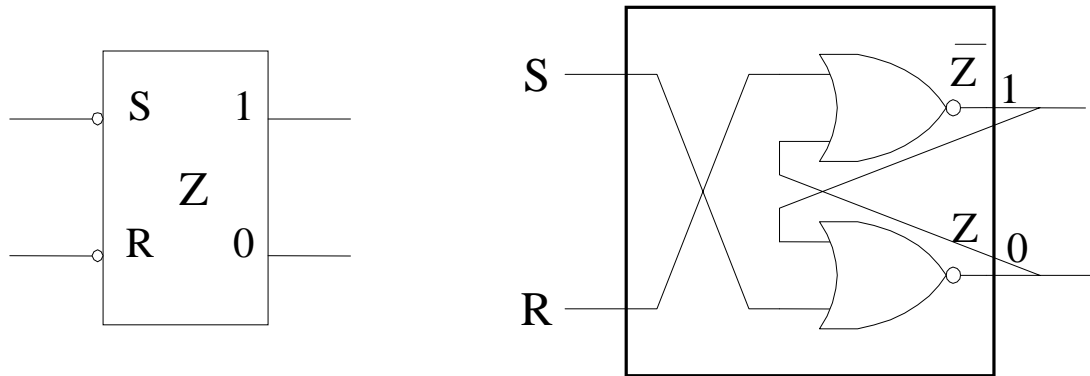
$$Z(t + \tau_1 + \tau_2) = \overline{\overline{Y(t + \tau_1) + R(t + \tau_1)}}$$

$$= \overline{Z(t) + S(t) + R(t + \tau_1)}$$

$$= \overline{R(t + \tau_1)} [Z(t) + S(t)]$$

Rangk. Sekuensial

Persamaan NOR SR Flip-flop

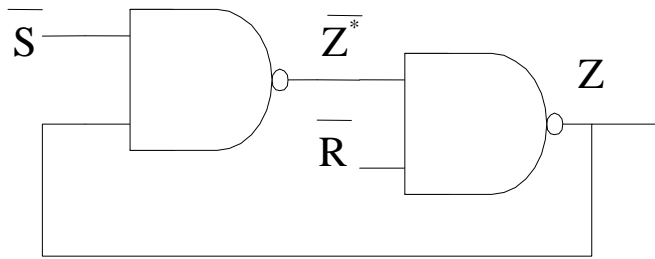


Simbol Logika NOR SR Flip-flop

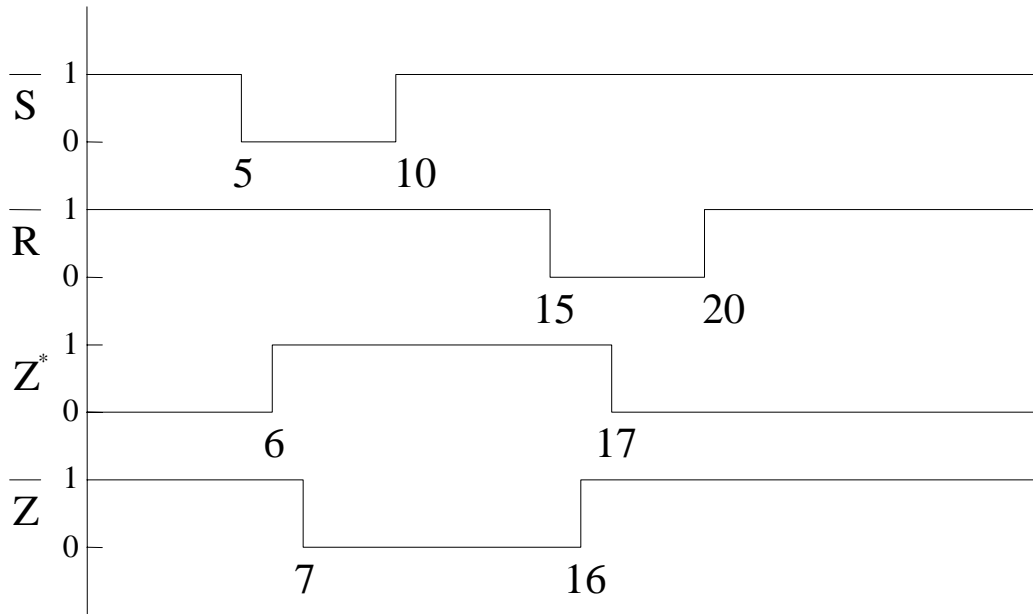
Tabel Kebenaran NOR SR Flip-Flop

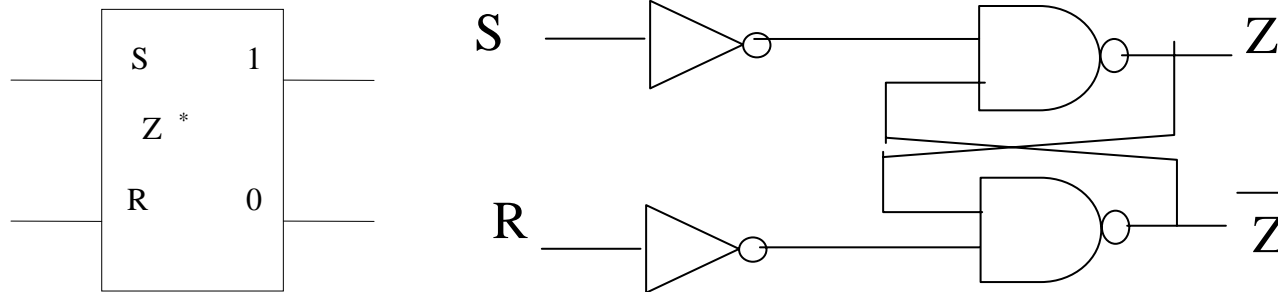
<b>S</b>	<b>R</b>	<b>Z*</b>	<b>Z̄</b>	<b>Kondisi</b>
0	0	Zn	Z̄n	Hold
0	1	0	1	Reset
1	0	1	0	Set
1	1	0	0	Not used

# NAND SR FLIP-FLOP



$$\begin{aligned} \bar{S}(t) &= 1 \\ \bar{R}(t) &= 1 \\ \bar{Z}^*(t) &= 1 \\ \bar{Z}(t) &= 1 \end{aligned}$$





Simbol Logika NAND SR Flip-flop

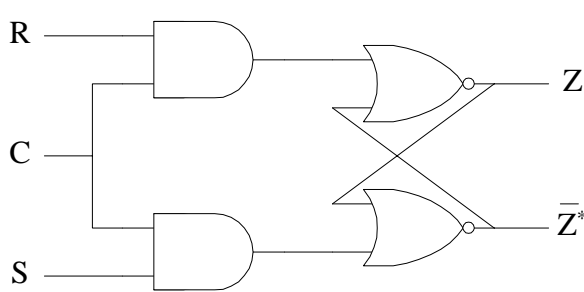
Tabel Kebenaran NAND SR Flip-flop

<b>S</b>	<b>R</b>	<b>Z*</b>	<b>Z</b>	<b>Kondisi</b>
0	0	Zn	Zn	Hold
0	1	0	1	Reset
1	0	1	0	Set
1	1	1	1	Not used

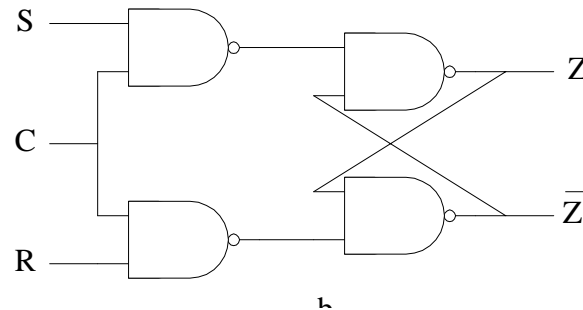
# ELEMEN PENYIMPAN DENGAN CLOCK

Di dalam sistem digital sering terjadi beberapa buah SR flip flop yang bekerja secara bersamaan (*synchron*). Untuk mengatasi hal itu, maka diperlukan suatu alat pengontrol yang bekerja untuk mengatur proses dari rangkaian tersebut.

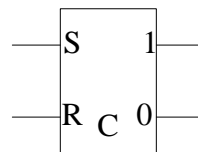
*Clock SR flip flop* yaitu menambahkan sinyal enable pada gerbang SR. Tujuan dari suatu sinyal clock adalah agar user dapat *menahan* dan *mengembalikan* SR flip flop untuk berhenti sejenak (*rest state*) selama perubahan terjadi pada input SR.



*NOR SR-FF dengan Clock*



*NAND SR-FF dengan Clock*



Rangk. Sekuensial Simbol Logika SR-FF dengan Clock

Persamaan output untuk gerbang NOR :

$$Z(t + \Delta) = [\overline{R}(t) + \overline{C}(t)][S(t)C(t) + Z(t)]$$

$$\overline{Z}(t + \Delta) = [\overline{S}(t) + \overline{C}(t)][R(t)C(t) + \overline{Z}(t)]$$

Jika  $C(t) = 0$

$$Z(t + \Delta) = Z(t), \quad \overline{Z}^*(t + \Delta) = \overline{Z}^*(t)$$

Jika  $C(t) = 1$

$$\begin{aligned} Z(t + \Delta) &= \overline{R}(t)[S(t) + Z(t)], \\ \overline{Z}(t + \Delta) &= \overline{S}(t)[R(t) + \overline{Z}^*(t)] \end{aligned}$$



# ANALISA RANGKAIAN SEKUENSIAL

Digunakan untuk mendapatkan hasil keluaran dari sebuah rangkaian sekuensial yang diketahui

## Langkah-langkah melakukan analisa :

a. Tentukan persamaan logika kombinasional untuk input S dan R, serta anggap gerbangnya dalam keadaan ideal.

b. Tentukan apakah  $S.R = 0$

Catatan : Jika  $S.R \neq 0$ , prosedur harus dihentikan.

c. Gunakan persamaan gerbang NAND atau NOR untuk menentukan persamaan next state.

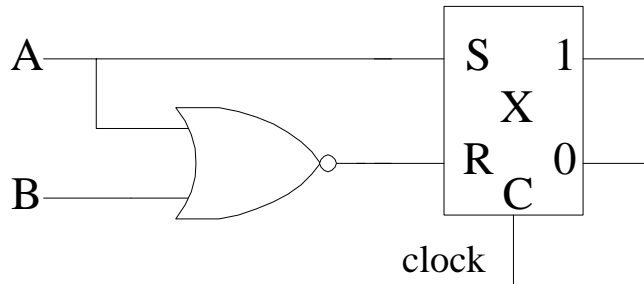
$$Z(t + \Delta) = \underline{S}(t) + \overline{R}(t)Z(t) \quad \text{NAND}$$

$$Z(t + \Delta) = \overline{R}(t)[S(t) + Z(t)] \quad \text{NOR}$$

Catatan : Jika  $S.R = 0$ , kedua persamaan ini adalah ekivalen.<sup>17</sup>

### **Contoh 1 :**

Tentukan persamaan next state dan tabel present state/next state untuk clock SR flip flop di bawah ini.



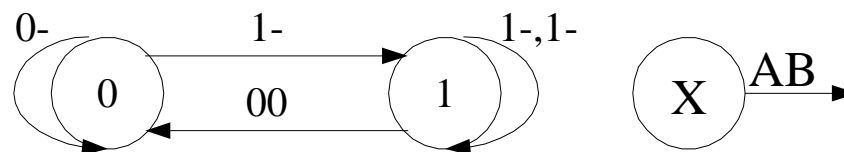
### **Jawab :**

1.  $S(t) = A(t)$  ,  $R(t) = \overline{A(t) + B(t)} = \overline{A(t)} \cdot \overline{B(t)}$
2.  $S(t) \cdot R(t) = A(t) \cdot \overline{A(t)B(t)} = 0$
3.  $X(t + \Delta) = S(t) \overline{R(t)} X(t)$   
 $= A(t) + [A(t) + B(t)] X(t)$   
 $= A(t) + B(t)X(t)$

*Tabel Present State / Next State untuk soal contoh 1 :*

Present Input		Present State	Next State
A(t)	B(t)	X(t)	X(t + Δ)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

*State Diagram untuk soal contoh 1*



Rangk. Sekuensial

# SINTESA RANGKAIAN SEKUENSIAL

Digunakan untuk mendisain sebuah rangkaian logika sekuensial, jika diberikan deskripsi dari fungsi rangkaian tersebut

## *Prosedur sintesa dengan menggunakan clock SR flip flop :*

1. Dengan menggunakan persamaan next state yang diketahui, buatlah tabel present state/next state untuk rangkaian yang akan dibangun.
2. Tambahkan kolom pasangan eksitasi  $S_{X_i}(t)$  dan  $R_{X_i}(t)$  untuk setiap variabel keadaan. Masukkan ke dalam kolom ini, dengan menggunakan pasangan :  $[X_i(t), X_i(t + \Delta)]$
3. Tuliskan persamaan logika untuk kolom eksitasi  $S_{Q_i}(t)$  dan  $R_{Q_i}(t)$ .
4. Buatlah tabel eksitasi dan persamaan outputnya.
5. Periksa kembali dan analisa setiap flip flop dengan menggunakan persamaan umum next state, yaitu :

$$Q_i(t + \Delta) = S_{Q_i}(t) + \bar{R}_{Q_i}(t)Q_i(t)$$

Kemudian :  $S_{Q_i}(t) \cdot R_{Q_i}(t) = 0$

## Contoh 2 :

Rancanglah rangkaian sekuensial dengan menggunakan clock SR flip flop dimana persamaan next statenya adalah :

$$X(t + \Delta) = A(t)X(t) + B(t)$$

### Jawab :

Dengan menggunakan persamaan next state, maka dapat dibangun tabel present state/next statenya

**Tabel 8-12** Tabel present state/next state contoh 8-4

Present Input		Present State	Next State	Nilai eksitasi	
A(t)	B(t)	X(t)	X(t + Δ)	S <sub>x</sub> (t)	R <sub>x</sub> (t)
0	0	0	0	0	d
0	0	1	0	0	1
0	1	0	1	1	0
0	1	1	1	d	0
1	0	0	0	0	d
1	0	1	1	d	0
1	1	0	1	1	0
1	1	1	1	d	0

# Mencari persamaan logika menggunakan K-Map :

Untuk  $S_x$

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

$$S_x(t) = B(t)$$

Untuk  $R_x$

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

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

Bentuk rangkaian adalah sbb :

