

## PERCOBAAN 2. DASAR-DASAR RANGKAIAN SEKUENSIAL 2

### 2.1. TUJUAN :

Setelah melaksanakan percobaan ini mahasiswa diharapkan mampu :

- Membuat SR Flip-flop dari gerbang NOR
- Membuat SR Flip-flop dari gerbang NAND
- Membuat SR Flip-flop dengan Clock
- Melakukan analisa rangkaian sekuensial dengan SR Flip-flop
- Mendisain rangkaian sekuensial dengan SR flip-flop

### 2.2. PERALATAN :

- Modul Trainer ITF 02 / DL 02

### 2.3. TEORI :

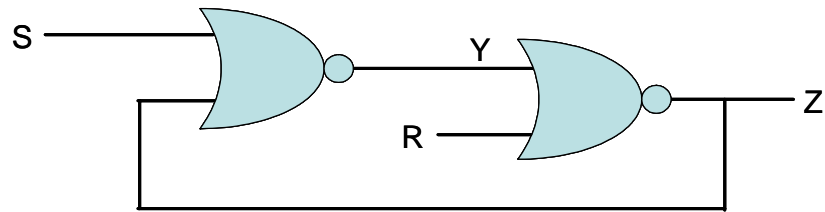
#### 2.3.1. SR Flip-flop dari gerbang NOR dan NAND

Rangkaian Sekuensial dapat dibuat dari gerbang kombinasional yang dimodifikasi sedemikian rupa sehingga menghasilkan kondisi Present State dan Next State. Ada dua jenis gerbang yang bisa digunakan : gerbang NOR dan gerbang NAND.

SR Flip-flop adalah jenis rangkaian sekuensial yang mempunyai dua input, yaitu input S (Set) dan input R (Reset), serta mempunyai dua output yaitu output Z dan  $\bar{Z}$ . Nilai dari Z selalu berlawanan dengan  $\bar{Z}$ , sehingga rangkaian ini disebut sebagai *Flip – flop* (Z sebagai Flip dan  $\bar{Z}$  sebagai Flop).

#### 1) SR *Flip-flop* dari gerbang NOR

Untuk membuat sebuah SR Flip-flop dari gerbang NOR, dibentuk rangkaian seperti gambar 2.1.



Gambar 2.1. SR Flip-flop dari gerbang NOR

Jika output Y dianggap mempunyai nilai yang berlawanan dengan output Z, maka  $Y = \bar{Z}$ . Dengan kombinasi nilai biner dari input S dan R maka didapatkan Tabel PS/NS untuk SR Flip-flop dari gerbang NOR adalah seperti pada Tabel 2.1.

Tabel 2.1. Tabel PS / NS untuk SR Flip-flop dari gerbang NOR

S	R	Z*	$\bar{Z}$	Kondisi
0	0	Zn	$\bar{Z}$ n	Hold
0	1	0	1	Reset
1	0	1	0	Set
1	1	0	0	Not used

Untuk melakukan analisa rangkaian sekuensial, diperlukan nilai dari Next Outputnya. Cara mendapatkan Next Output dari rangkaian di atas adalah sebagai berikut :

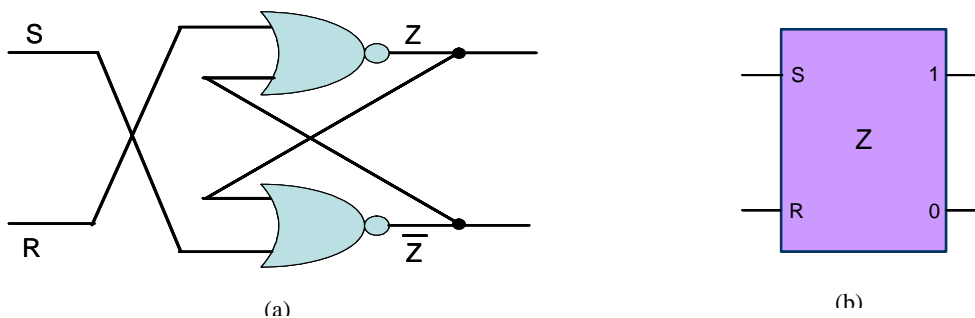
$$Z(t + \Delta) = \overline{\bar{Z}(t) + R(t)}$$

$$\bar{Z}(t + \Delta) = \overline{S(t) + Z(t)}$$

$$Z(t + 2\Delta) = \overline{\bar{Z}(t + \Delta) + R(t + \Delta)} \text{ atau } Z(t + 2\Delta) = \overline{\overline{\overline{S(t) + Z(t)} + R(t + \Delta)}}$$

Jika  $\Delta \ll 0$  maka  $Z(t + \Delta) = \bar{R}(t) \cdot [S(t) + Z(t)] \rightarrow$  persamaan SR FF dengan NOR

SR Flip-flop bisa dirangkai dengan cara lain seperti ditunjukkan pada gambar 2.2.

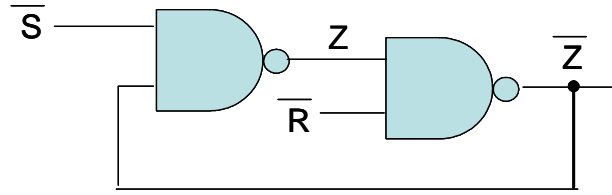


Gambar 2.2. Bentuk lain SR Flip-flop dari gerbang NOR

(a) Gambar rangkaian (b) Simbol logika

2) SR Flip-flop dari gerbang NAND

Untuk membuat sebuah SR Flip-flop dari gerbang NAND, dibentuk rangkaian seperti gambar 2.3.



Gambar 2.3. SR Flip-flop dari gerbang NAND

Tabel 2.2. Tabel PS / NS untuk SR Flip-flop dari gerbang NAND

S	R	Z*	Z-bar	Kondisi
0	0	Zn	Zn	Hold
0	1	0	1	Reset
1	0	1	0	Set
1	1	0	0	Not used

Nilai Next Output dari gerbang NAND didapatkan dari persamaan sebagai berikut :

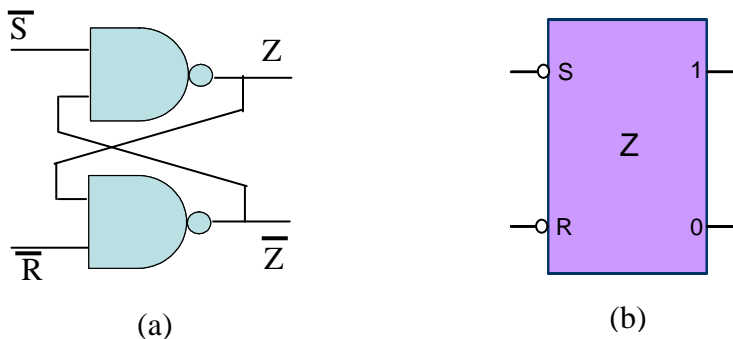
$$Z(t + \Delta) = \bar{S}(t) \cdot \bar{Z}(t)$$

$$\bar{Z}(t + \Delta) = \overline{\bar{R}(t) \cdot Z(t)}$$

$$Z(t + 2\Delta) = \overline{\bar{Z}(t + \Delta) \cdot \bar{S}(t + \Delta)} \quad \text{atau} \quad Z(t + 2\Delta) = \overline{\overline{\bar{R}(t) \cdot Z(t)} \cdot \bar{S}(t + \Delta)}$$

Jika  $\Delta \ll 0$  maka  $Z(t + \Delta) = S(t) + [\bar{R}(t) \cdot Z(t)] \rightarrow$  persamaan SR FF dengan NAND

Rangkaian SR Flip-flop yang lain ditunjukkan pada gambar 2.4.

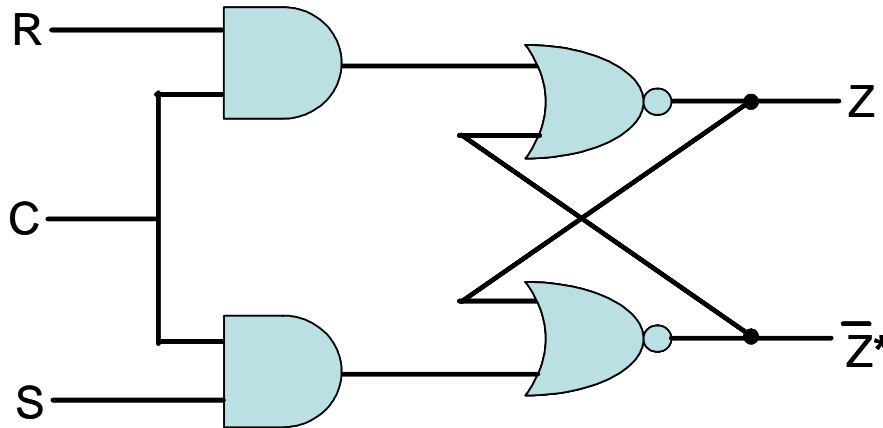


Gambar 2.4. Bentuk lain SR Flip-flop dari gerbang NAND

(a) Gambar rangkaian (b) Simbol logika

### 2.3.2. SR Flip-Flop dengan Clock

Sebuah rangkaian Sekuensial dapat diatur sebagai elemen penyimpan jika diberi input kontrol. Input kontrol tersebut akan mengatur kapan Next Output boleh dikeluarkan atau tidak. Pemberian input kontrol (untuk selanjutnya disebut Clock) ditunjukkan pada gambar 2.5.



Gambar 2.5. SR Flip-flop dari gerbang NOR dengan Clock

Input C merupakan input kontrol yang akan mengatur nilai R dan S yang masuk ke Flip-flop. Jika C bernilai 1, output Flip-flop akan berubah ke kondisi Next-nya sesuai dengan kombinasi input R dan S nya, sehingga  $Z(t + \Delta) = \overline{R}(t).[S(t) + Z(t)]$ . Jika C bernilai 0, output Flip-flop tidak berubah, artinya kondisi Next sama dengan kondisi Present-nya, atau  $Z(t + \Delta) = Z(t)$ . Dengan kondisi ini maka flip-flop dapat dikatakan sebagai elemen penyimpan.

### 2.3.3. Analisa Rangkaian Sekuensial

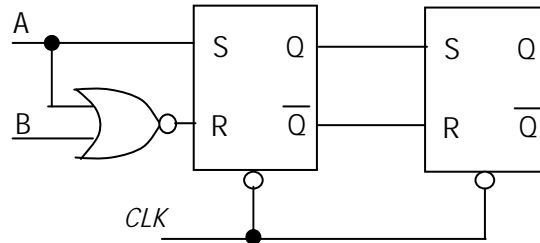
Menganalisa rangkaian adalah mengamati cara kerja sebuah rangkaian untuk mendapatkan hasilnya. Untuk menganalisa sebuah rangkaian sekuensial diperlukan langkah-langkah sebagai berikut :

1. Tentukan persamaan logika untuk input-input Flip-flopnya
2. Untuk jenis SR-FF, yakinkan bahwa persamaan logika input  $S.R = 0$ , jika tidak, hentikan analisa ini (tidak sesuai dengan sifat SR-FF, dimana nilai input S dan R keduanya tidak pernah = "1").

3. Tentukan persamaan Next State untuk output masing-masing flip-flop yang dianalisa :

$$Z(t + \Delta) = S(t) + \bar{R}(t)Z(t) \quad \text{Untuk SR Flip-flop dengan gerbang NAND}$$

$$Z(t + \Delta) = \bar{R}(t).[S(t) + Z(t)] \quad \text{Untuk SR Flip-flop dengan gerbang NOR}$$



Gambar 2.6. Contoh Rangkaian Sekuensial dari SR-FF

### 2.3.4. Sintesa Rangkaian Sekuensial

Untuk mendisain sebuah rangkaian sekuensial yang dapat memberikan respons tertentu sesuai dengan yang kita kehendaki, maka dilakukan proses sintesa rangkaian. Pada proses sintesa rangkaian, yang diketahui adalah perubahan kondisi dari satu kondisi awal ke kondisi berikutnya. Proses sintesa berkebalikan dengan proses analisa, oleh karena itu diperlukan Tabel Eksitasi, yang merupakan tabel kebalikan dari Tabel State. Pada Tabel Eksitasi, nilai output sekarang (*Present Output*) dan output berikutnya (*Next Output*) sudah diketahui. Nilai *Present Input* dicari dari hubungan kedua nilai output tadi. Tabel Eksitasi dari SR- flip-flop seperti ditunjukkan pada Tabel 2.3

Tabel 2.3. Tabel Eksitasi SR-FF

<i>PS</i>	<i>NS</i>	<i>Eksitasi</i>	
<b>Q(t)</b>	<b>Q(t+Δ)</b>	<b>S(t)</b>	<b>R(t)</b>
0	0	0	d
0	1	1	0
1	0	0	1
1	1	d	0

Untuk melakukan proses sintesa rangkaian, ikuti langkah–langkah sebagai berikut :

1. Dapatkan bentuk Tabel PS/NS dari kasus yang diketahui (bisa dalam bentuk soal cerita, maupun persamaan next state)
2. Buat Tabel Eksitasi sesuai dengan jenis Flip-flop yang akan digunakan.

3. Buat K-map untuk masing-masing input Flip-flop.
4. Cari Persamaan Logika dari input Flip-flop sesuai hasil dari K-Map.
5. Buat gambar rangkaian dan jalankan.

## 2.4. PROSEDUR PERCOBAAN

### 2.4.1. SR Flip-flop dari gerbang NOR dan NAND

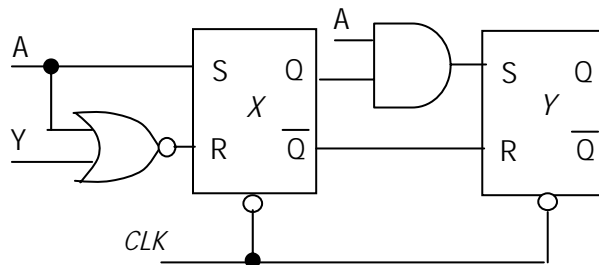
1. Buat rangkaian SR Flip-flop dari gerbang NOR seperti gambar 2.1.
2. Dapatkan Tabel Present State / Next Statenya
3. Buat rangkaian SR Fip-flop dari gerbang NAND seperti gambar 2.3.
4. Dapatkan Tabel Prsent State / Next Statenya

### 2.4.2. SR Flip-flop dengan Clock

1. Buat rangkaian SR Flip-flop dengan Clock seperti gambar 2.5.
2. Input C berasal dari switch input.
3. Dapatkan Tabel Kebenarannya.

### 2.4.3. Analisa Rangkaian Sekuensial

1. Pada Trainer, buatlah rangkaian seperti yang ditunjukkan pada gambar 2.7.



Gambar 2.7. Percobaan Analisa Rangkaian menggunakan SR-FF

2. Sebelum menjalankan rangkaian, periksakan dulu ke dosen / asisten .
3. Buat Tabel PS/NS sebagai hasil pengamatan.
4. Bandingkan hasilnya apabila menggunakan persamaan Next-State untuk SR-FF.

### 2.4.2. Sintesa Rangkaian Sekuensial (dengan JK-FF)

1. Disain sebuah rangkaian sekuensial yang terdiri dari 1 buah SR-FF dimana flip-flop tersebut mempunyai persamaan next-state sebagai berikut :

$$X(t + \Delta) = A(t) + \overline{A}(t)X(t)$$

2. Carilah nilai eksitasinya sesuai langkah-langkah yang telah dijelaskan sebelumnya.
3. Gambarkan hasilnya dan rangkai di trainer.
4. Catat hasilnya pada Tabel PS/NS.

## **2.5. TUGAS**

1. Dapatkan State Diagram dari rangkaian SR Flip-flop dengan gerbang NOR maupun dengan gerbang NAND yang sudah diamati.
2. Disain sebuah rangkaian sekuensial dari SR Flip-flop yang memiliki persamaan next state sebagai berikut :

$$W(t + \Delta) = \overline{B(t) + Y(t)}$$

$$Y(t + \Delta) = W(t) \cdot \overline{B}(t)$$