

PERCOBAAN 3. FLIP-FLOP

3.1. TUJUAN :

Setelah melaksanakan percobaan ini mahasiswa diharapkan mampu :

- Membedakan sifat dasar SR-FF dengan dan tanpa clock
- Membuat rangkaian Master Slave JK-FF
- Menggunakan input-input Asinkron pada JK-FF
- Membuat D-FF dan T-FF dari JK-FF dan SR-FF
- Mendisain beberapa macam rangkaian sekuensial menggunakan ke-4 jenis Flip-flop

3.2. PERALATAN :

- Modul Trainer ITF 02

3.3. TEORI :

3.3.1. Pendahuluan

Flip-flop merupakan suatu rangkaian sekuensial yang dapat menyimpan data sementara (*latch*) dimana bagian outputnya akan me-respons input dengan cara mengunci nilai input yang diberikan atau mengingat input tersebut. Flip-flop mempunyai dua kondisi output yang stabil dan saling berlawanan.

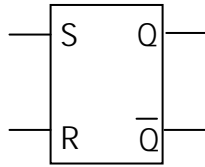
Perubahan dari setiap keadaan output dapat terjadi jika diberikan *trigger* pada flip-flop tersebut. *Triger* –nya berupa sinyal logika “1” dan “0” yang kontinyu.

Ada 4 tipe Flip-flop yang dikenal, yaitu SR, JK, D dan T Flip-flop. Dua tipe pertama merupakan tipe dasar dari Flip-flop, sedangkan D dan T merupakan turunan dari SR dan JK Flip-flop.

3.3.2. SR-Flip-Flop (SET & RESET Flip-Flop)

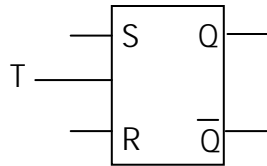
SR-Flip-flop dapat dibentuk dengan dua cara; dari gerbang NAND atau dari gerbang NOR. Proses pembentukan dasar SR-FF telah dijelaskan dalam teori. Pada percobaan ini kita akan mengamati dua jenis SR-FF, yang tanpa menggunakan *Clock* dan

dengan menggunakan *Clock*. Perbedaan dasar dari kedua jenis SR tersebut adalah perubahan output berikutnya akan terjadi dengan atau tanpa adanya *clock / trigger*.



Gambar 3.1. Simbol Logika SR-FF tanpa Clock

Pada jenis SR-FF yang disimbolkan pada gambar 3.1, setiap perubahan yang diberikan pada input S dan R akan menyebabkan terjadinya perubahan output menuju keadaan berikutnya.



Gambar 3.2. Simbol Logika SR-FF dengan *Clock / Positive-edge Trigger*

SR-FF dengan simbol seperti pada gambar 3.2, outputnya baru akan memberikan respons menuju output berikutnya jika input T diberi *trigger*.

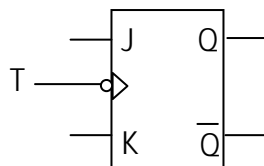
Tabel 3.1. menunjukkan perubahan kondisi output dari SR-FF dengan Clock. Jika *clock* bernilai “1”, maka kondisi output akan berubah sesuai dengan perubahan input SR-nya, jika *clock* bernilai “0”, kondisi output tetap pada kondisi sebelumnya, meskipun nilai input S dan R-nya diubah-ubah.

Tabel 3.1. Tabel State SR-FF dengan Clock

Clock	Present State		Present Output	Next Output
	S	R	Q	Qn
0	0	0	0	Hold "0" saja atau "1" saja
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	0
1	0	1	1	
1	1	0	0	1
1	1	0	1	
1	1	1	0	*
1	1	1	1	*

3.3.3. JK-FLIP-FLOP

Sebuah JK-FF adalah SR-FF yang telah dimodifikasi sedemikian rupa. Pada SR-FF, jika kedua input S dan R-nya sama-sama bernilai "1", flip-flop tidak mampu merespons kondisi output berikutnya (pelajari lagi sifat SR-FF). Sebuah JK-FF dibentuk dari SR-FF dengan tambahan gerbang AND pada sisi input SR-nya. Dengan tambahan tersebut, apabila input J dan K keduanya bernilai "1" akan membuat kondisi output berikutnya menjadi kebalikan dari kondisi output sebelumnya. Keadaan ini dinamakan *Toggle*.

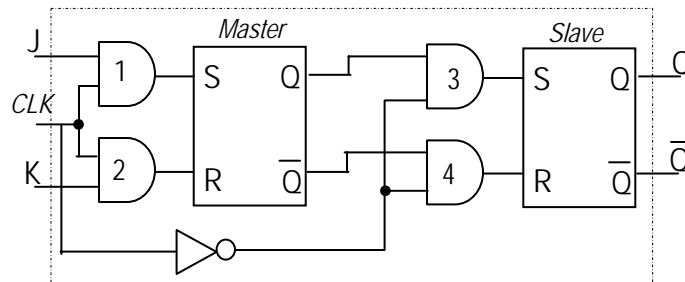


Gambar 3.3. Simbol Logika JK-FF dengan *negative-edge trigger*

Tabel 3.2. Tabel State JK-FF

Clock	Present Input		Present Output	Next Output
	T	J	K	Qn
1	0	0	0	Hold "0" saja atau "1" saja
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	
0	0	0	0	Hold
0	0	0	1	
0	0	1	0	0
0	0	1	1	
0	1	0	0	1
0	1	0	1	
0	1	1	0	Toggle
0	1	1	1	

Sebuah *Master-Slave* JK-FF dibentuk dari dua buah SR-FF, dimana operasi dari kedua SR-FF tersebut dilakukan secara bergantian, dengan memberikan input *Clock* yang berlawanan pada kedua SR-FF tersebut. *Master-Slave* JK-FF ditunjukkan pada gambar 3.4.

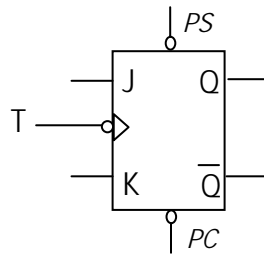


Gambar 3.4. Sebuah Master-Slave JK-FF disusun dari SR-FF

Prinsip dasar dari *Master-Slave* JK-FF adalah sebagai berikut : jika *Clock* diberi input "1", gerbang AND 1 dan 2 akan aktif, SR-FF ke-1 (*Master*) akan menerima data yang dimasukkan melalui input J dan K, sementara gerbang AND 3 dan 4 tidak aktif (menghasilkan output = "0"), sehingga SR-FF ke-2 (*Slave*) tidak ada respons (kondisinya sama dengan kondisi sebelumnya). Sebaliknya jika *Clock* diberi input "0", gerbang 3 dan

4 aktif, *Slave* akan mengeluarkan output di Q dan Q', sementara *Master* tidak me-respons input, karena gerbang AND 1 dan 2 tidak aktif.

Selain mempunyai input *Clock*, sebuah JK-FF juga dilengkapi dengan input-input Asinkron. Kedua input Asinkron ini dikenal sebagai *Preset (PS)* dan *Preclear (PC)*. IC JK-FF yang mempunyai input Asinkron adalah 74LS76. Kedua input Asinkron ini digunakan untuk mengoperasikan JK-FF dimana kondisi perubahan outputnya tidak hanya bergantung kepada nilai input J dan K-nya, melainkan juga pada nilai input Asinkron tersebut. Contoh pemakaian input Asinkron ini adalah untuk me-reset JK-FF ke kondisi "0" maupun men-set JK-FF ke kondisi "1", tanpa harus menunggu J dan K bernilai "0" dan "1" atau sebaliknya. Input-input Asinkron akan diaplikasikan dalam pembuatan *Counter* dan *Shift Register*.



Gambar 3.5. JK-FF dengan input Asinkron

Tabel 3.3. Tabel PS/NS JK-FF menggunakan Input Asinkron

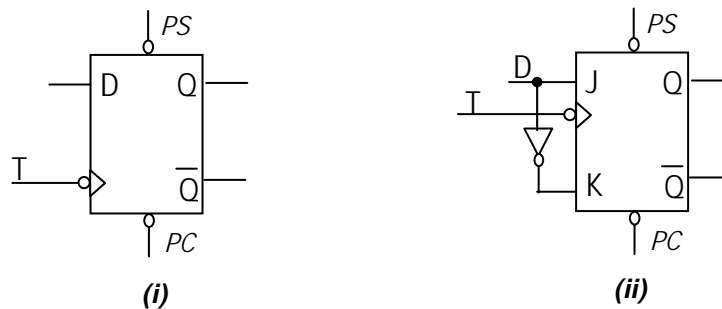
Clock T	Input Asinkron		Present Input		Present Output	Next Output
	PS	PC	J	K	Q	Qn
0	0	0	x	x	x	not used
0	0	1	x	x	x	1
0	1	0	x	x	x	0
0	1	1	0	0	0	Hold
0	1	1	0	0	1	
0	1	1	0	1	0	0
0	1	1	0	1	1	
0	1	1	1	0	0	1
0	1	1	1	0	1	
0	1	1	1	1	0	Toggle
0	1	1	1	1	1	

x = don't care

3.3.4. D-FLIP FLOP (Delay/Data Flip-Flop)

Sebuah D-FF terdiri dari sebuah input D dan dua buah output Q dan Q'. D-FF digunakan sebagai Flip-flop pengunci data. Prinsip kerja dari D-FF adalah sebagai berikut : berapapun nilai yang diberikan pada input D akan dikeluarkan dengan nilai yang sama pada output Q. D-FF diaplikasikan pada rangkaian-rangkaian yang memerlukan penyimpanan data sementara sebelum diproses berikutnya. Salah satu contoh IC D-FF adalah 74LS75, yang mempunyai input Asinkron.

D-FF juga dapat dibuat dari JK-FF, dengan mengambil sifat Set dan Reset dari JK-FF tersebut. Rangkaian D-FF ditunjukkan pada gambar 3.6.



Gambar 3.6. D-Flip Flop

(i) Simbol Logika D-FF 74LS75 (ii) D-FF dari JK-FF

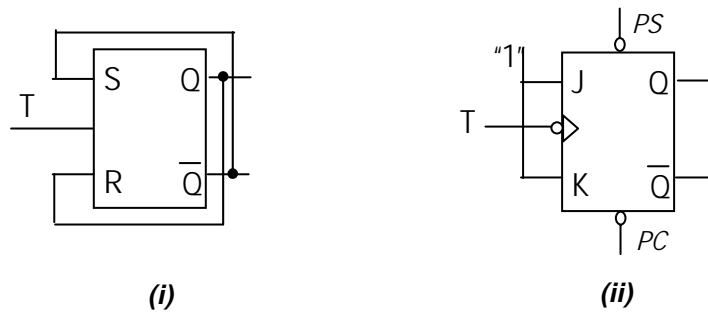
Tabel 3.4. Tabel State D-FF

Clock	Present Input	Present Output	Next Output
T	D	Q	Qn
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	Hold
1	0	1	
1	1	0	
1	1	1	

3.3.5. T-FLIP-FLOP (Toggle Flip-Flop)

Sebuah T-FF dapat dibentuk dari SR-FF maupun dari JK-FF, karena pada kenyataan, IC T-FF tidak tersedia di pasaran. T-FF biasanya digunakan untuk rangkaian yang memerlukan kondisi output berikut yang selalu berlawanan dengan kondisi sebelumnya, misalkan pada rangkaian pembagi frekuensi (*Frequency Divider*).

Rangkaian T-FF dibentuk dari SR-FF dengan memanfaatkan hubungan *Set* dan *Reset* serta output Q dan Q' yang diumpan balik ke input S dan R. Sedangkan rangkaian T-FF yang dibentuk dari JK-FF hanya perlu menambahkan nilai "1" pada input-input J dan K (ingat sifat *Toggle* dari JK-FF).



Gambar 3.7. Rangkaian T-Flip-Flop

(i) Dari SR-FF (ii) Dari JK-FF

Tabel 3.5. Tabel State dari T-FF

Clock	Present Output	Next Output
T	Q	Qn
0	0	1
0	1	0
1	0	Hold
1	1	

3.3.6. Sintesa Rangkaian Sekuensial

Pada subbab ini kita akan mencoba mengaplikasikan semua jenis Flip-flop yang sudah dibahas diatas untuk membuat sebuah rangkaian sekuensial yang dapat menghasilkan respons output tertentu. Untuk itu perlu diketahui Tabel Eksitasi dari masing-masing jenis Flip-flop di atas (Tabel ini dibuat hanya dengan membalik Tabel PS/NS masing-masing flip-flop yang sudah diketahui).

Tabel 3.6. Tabel Eksitasi untuk SR-FF, JK-FF, D-FF dan T-FF

PS	NS	Eksitasi					
Q	Qn	S	R	J	K	D	T
0	0	0	x	0	x	0	1
0	1	1	0	1	x	1	0
1	0	0	1	x	1	0	0
1	1	x	0	x	0	1	1

Contoh :

Buat sebuah rangkaian sekuensial Down Counter sinkron 2 bit menggunakan D-FF.

Sesuai dengan prosedur sintesa rangkaian (Percobaan 1), cari dahulu Tabel PS/NS, Eksitasi, buat K-Map dan temukan persamaan Logika. Selesaikan dengan gambar rangkaian menggunakan D-FF.

State diagram dari Down Counter tersebut adalah $\overset{\rightarrow}{3} \overset{\rightarrow}{2} \overset{\rightarrow}{1} \overset{\rightarrow}{0} \overset{\rightarrow}{3} \overset{\rightarrow}{2} \overset{\rightarrow}{1} \overset{\rightarrow}{0} \dots$

Tabel 3.7 . Tabel PS/NS dan Eksitasi D-FF untuk contoh soal Down Counter 2 bit

PS		NS		Eksitasi	
A2	A1	A2n	A1n	D2	D1
1	1	1	0	1	0
1	0	0	1	0	1
0	1	0	0	0	0
0	0	1	1	1	1

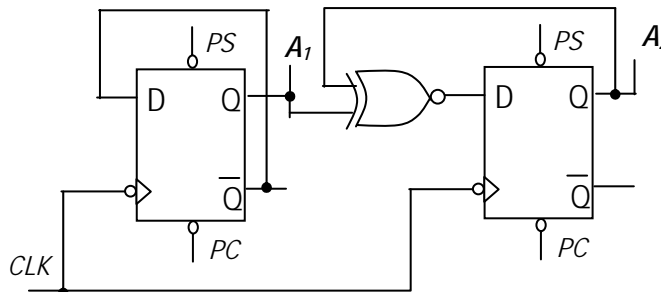
Bentuk K-Map :

$A_2 \backslash A_1$	0	1
0	1	0
1	0	1

$$D_2 = \overline{A_2} \oplus A_1$$

$A_2 \backslash A_1$	0	1
0	(1)	0
1	(1)	0

$$D_1 = \overline{A_1}$$



Gambar 3.8. Rangkaian Down Counter 2 bit dengan D-FF

3.4. PROSEDUR PERCOBAAN

3.4.1. SR-FF dengan dan tanpa Clock

1. Cari simbol logika SR-FF tanpa Clock dan dengan Clock pada trainer ITF-02.
2. Berikan nilai melalui switch pada input-input S dan R-nya.
3. Amati hasilnya pada display output Q-nya.
4. Untuk SR-FF dengan Clock, respons berikut dari output Q baru nampak jika input Clock sudah ditekan.
5. Tuliskan hasilnya pada Tabel PS/NS.

3.4.2. Master-Slave JK-FF dan JK-FF dengan input Asinkron

1. Buat rangkaian *Master Slave* JK-FF dari SR-FF seperti pada gambar 3.4.
2. Berikan nilai pada input J dan K melalui *switch* input yang tersedia.
3. Berikan input *manual Clock*.
4. Amati hasilnya dan catat dalam Tabel PS/NS.
5. Cari simbol logika JK-FF dengan input Asinkron pada trainer ITF-02.
6. Pada masing-masing input J, K, PS dan PC, berikan nilai yang didapat dari *switch input*. Beri input *manual Clock* pada T. Ubah-ubahlah nilai-nilai tersebut untuk mengetahui respons outputnya.
7. Amati hasilnya dan catat dalam Tabel PS/NS.

3.4.3. D-FF

1. Buat sebuah rangkaian D-FF dari salah satu JK-FF dengan input Asinkron seperti gambar 3.6(ii).
2. Pada masing-masing input D, PS dan PC, berikan nilai yang didapat dari *switch input*. Beri input *manual Clock* pada T. Ubah-ubahlah nilai-nilai tersebut untuk mengetahui respons outputnya.
3. Amati hasilnya dan catat dalam Tabel PS/NS.
4. Ulangi langkah 1. s/d 3. dengan mengganti JK-FF dengan SR-FF. Bandingkan hasilnya dengan yang menggunakan JK-FF sebelumnya.

3.4.4 T-FF

1. Buat sebuah rangkaian T-FF dari salah satu SR-FF dengan Clock seperti gambar 3.7(i).
2. Berikan nilai pada input T menggunakan *switch input*.
3. Amati hasilnya dan catat dalam Tabel PS/NS.
4. Ulangi langkah 1. s/d 3. dengan menggunakan JK-FF seperti gambar 3.7(ii). Bandingkan hasilnya dengan yang menggunakan SR-FF sebelumnya..

3.4.5. Sintesa Rangkaian Sekuensial

1. Buat sebuah rangkaian *Odd-Up and Even-Down Counter Sinkron 3 bit*, dengan urutan : 1,3,5,7,6,4,2,0,1,3,5,7,6,4,2,0,1,3,..
2. Gunakan SR-FF untuk merealisasikan rangkaian tersebut.
3. Amati hasilnya. Catat dalam Tabel PS/NS.

3.5. TUGAS

1. Ubah rangkaian yang telah dibuat pada prosedur 3.4.5 (*Rangkaian Odd-Up and Even-Down*) di atas menggunakan D-FF. Lengkapi dengan Tabel PS/NS, Eksitasi dan K-Map untuk mendapatkan rangkaian tersebut.
2. Carilah bentuk gelombang output dari masing-masing flip-flop di bawah ini.

