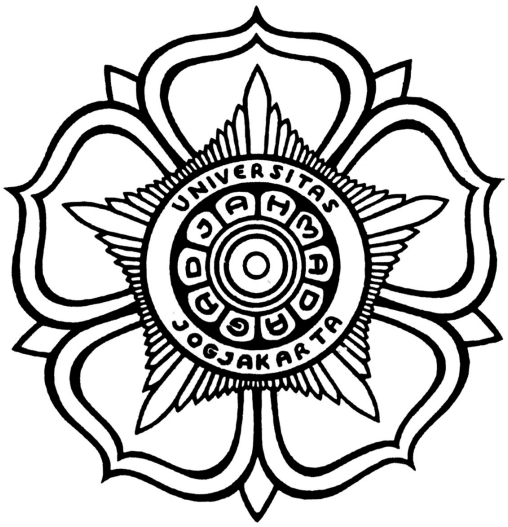
MODUL 04

PRAKTIKUM SISTEM DIGITAL

TNF 2178 - 1 SKS



Disusun oleh:
Prof. Ir. Sunarno, M.Eng., Ph.D.
dan
Tim Asisten Praktikum Sistem Digital

LABORATORIUM SENSOR DAN SISTEM TELEKONTROL
DEPARTEMEN TEKNIK NUKLIR DAN TEKNIK FISIKA
FAKULTAS TEKNIK
UNIVERSITAS GADJAH MADA
2018

Chapter 1

RANGKAIAN SEKUENSIAL: FLIP-FLOP & LATCH

1.1 Tujuan

a. Mengenal *flip-flop* dan *latch* dan pembagian kelompoknya

b. Mempelajari prinsip kerja dari *flip-flop* dan *latch*

c. Membuktikan tabel kebenaran dari *flip-flop* dan *latch*

d. Mengetahui aplikasi dari *flip-flop* dan *latch*

1.2 Materi

a. *Flip-Flop*

b. Rangkaian dasar *Flip-Flop*

c. *Set-Reset Flip-Flop*

d. *Data Flip-Flop*

e. *J-K Flip-Flop*

f. Tabel eksitasi *Flip-Flop*

g. Prosedur desain

1.3 Dasar Teori

1.3.1 Rangkaian Sekuensial

Rangkaian sekuensial adalah rangkaian yang *output*-nya tidak saja bergantung pada *input* pada saat itu saja tetapi juga bergantung pada keadaan *output*

sebelumnya.

1.3.2 Flip-Flop

Flip-flop adalah rangkaian utama dalam logika sekuensial. *Flip-flop* mempunyai nama lain *bistable multivibrator*. Rangkaian ini adalah rangkaian sel biner yang mempunyai dua buah *output* yang saling berkebalikan keadaanya. Disebut *bistable* karena rangkaian ini punya dua keadaan stabil, yaitu 0 dan 1 dan selalu berubah-ubah secara stabil.

Counter, register, shift register, serta rangkaian sekuensial lain disusun dengan menggunakan *flip-flop* sebagai komponen utama. *Flip-flop* adalah rangkaian yang mempunyai fungsi pengingat (*memory*). Artinya rangkaian ini mampu melakukan proses penyimpanan data sesuai dengan kombinasi masukan yang diberikan kepadanya. Data yang tersimpan itu dapat dikeluarkan sesuai dengan kombinasi masukan yang diberikan. Rangkaian *flip-flop* dapat menyimpan *state biner* (sepanjang masih terdapat tenaga pada rangkaian) sampai terjadi perubahan pada sinyal *input*-nya.

Ada beberapa macam *flip-flop* yang akan dibahas, yaitu *Set-Reset Flip-Flop, Data Flip-Flop,* dan *J-K Flip-Flop*. Sebagai tambahan akan dibahas pula masalah pemicuan yang akan mengaktifkan kerja *flip-flop*.

Hubungan *input-output* ideal yang dapat terjadi pada *flip-flop* adalah:

1. *Set*, yaitu jika suatu kondisi masukan mengakibatkan keluaran (Q) bernilai logika positif (1) saat dipicu, apapun kondisi sebelumnya.

2. *Reset*, yaitu jika suatu kondisi masukan mengakibatkan keluaran (Q) bernilai logika negatif (0) saat dipicu, apapun kondisi sebelumnya.

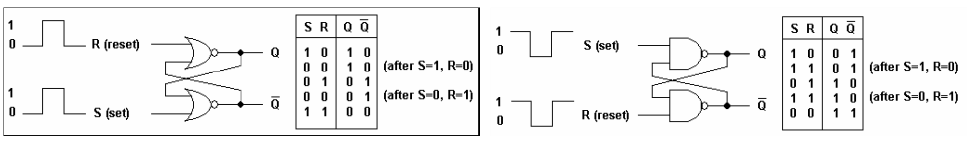
3. Tetap, yaitu jika suatu kondisi masukan mengakibatkan keluaran (Q) tidak berubah dari kondisi sebelumnya saat dipicu.

4. *Toggle*, yaitu jika suatu kondisi masukan mengakibatkan logika keluaran (Q) berkebalikan dari kondisi sebelumnya saat dipicu.

Secara ideal berdasar perancangan kondisi keluaran Q' selalu berkebalikan dari kondisi keluaran Q.

1.3.3 Rangkaian Dasar Flip-Flop

Flip-flop dapat dibuat dari dua buah gerbang NAND atau NOR



Gambar 1.1: Rangkaian dasar *flip-flop* dengan gerbang NOR dan NAND

1.3.4 Pemicuan Flip-Flop

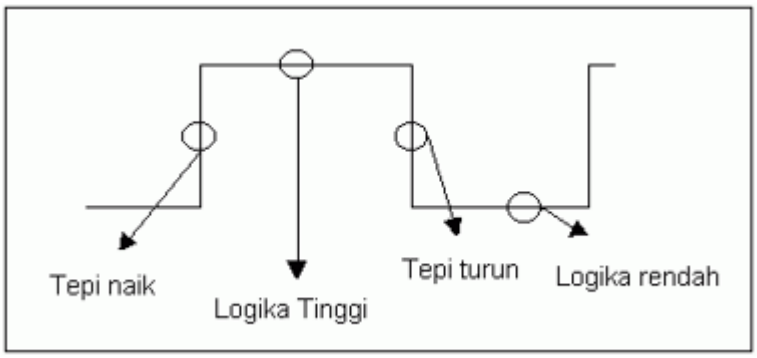
Pada *flip-flop* untuk menyerempakkan masukan yang diberikan pada kedua masukannya maka diperlukan sebuah *clock* untuk memungkinkan hal itu terjadi. *Clock* yang dimaksud di sini adalah sinyal pulsa yang beberapa kondisinya dapat digunakan untuk memicu *flip-flop* untuk bekerja. Ada beberapa kondisi *clock* yang biasa digunakan untuk menyerempakkan kerja *flip-flop* yaitu :

1. Tepi naik : yaitu saat perubahan sinyal *clock* dari logika rendah (0) ke logika tinggi (1).

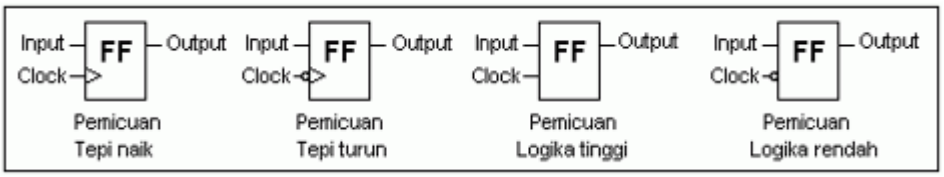
2. Tepi turun : yaitu saat perubahan sinyal *clock* dari logika tinggi (1) ke logika rendah (0).

3. Logika tinggi : yaitu saat sinyal *clock* berada dalam logika 1.

4. Logika rendah : yaitu saat sinyal *clock* berada dalam logika 0.



Gambar 1.2: Kondisi pemicuan *clock*



Gambar 1.3: Simbol-simbol pemicuan

Selanjutnya cara pengujian pemicuan suatu *flip-flop* akan dijelaskan dalam Tabel 1.1. Pada tabel tersebut, kita gunakan penerapan logika positif. Kondisi *Clock High*, yaitu saat *clock* ditekan sama artinya dengan logika 1, sedangkan saat *clock* dilepas sama artinya dengan logika 0. Jika pada langkah pengujian pertama keadaan sudah sesuai dengan tabel, pengujian dapat dihentikan, demikian seterusnya.

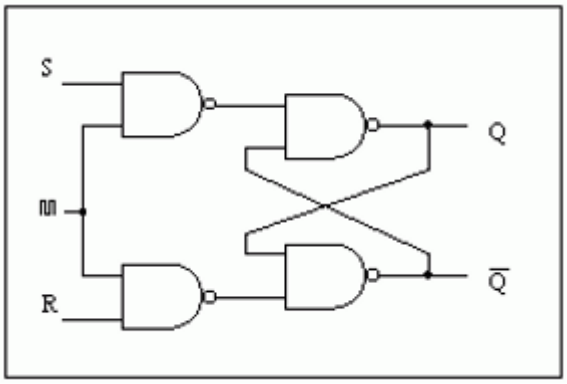
Table 1.1: Pengujian pemicuan *clock*

Langkah Pengujian	*Clock*	*Input*	*Output*	Jenis Pemicuan
1.	1	Diubah-ubah	Berubah	Logika Tinggi
2.	0	Diubah-ubah	Berubah	Logika Rendah
	0	Diubah-ubah	Tetap	
3.	0 ke 1 (ditekan)	Diubah-ubah	Berubah	Tepi naik
	1	Diubah-ubah	Tetap	
	1	Diubah-ubah	Tetap	
4.	1 ke 0 (dilepas)	Diubah-ubah	Berubah	Tepi turun
	0	Diubah-ubah	Tetap	

1.3.5 Set-Reset Flip-Flop (SRFF)

Flip-flop R-S adalah rangkaian dasar dari semua jenis *flip-flop* yang ada. Terdapat berbagai macam rangkaian *flip-flop* R-S, pada percobaan ini *flip-flop* R-S disusun dari empat buah gerbang NAND 2 masukan. Dua masukan *flip-flop* ini adalah S (*set*) dan R (*reset*), serta dua keluarannya adalah Q dan Q'.

Kondisi keluaran akan tetap ketika kedua masukan R dan S berlogika 0. Sedangkan pada kondisi masukan R dan S berlogika 1 maka kedua keluaran akan berlogika 1, hal ini sangat dihindari karena bila kondisi masukan diubah menjadi berlogika 0 kondisi kelurannya tidak dapat diprediksi (bisa 1 atau 0). Keadaan ini disebut kondisi terlarang. Selanjutnya kondisi terlarang, pacu, dan tak tentu akan dijelaskan melalui Tabel 1.2 dan 1.3.



Gambar 1.4: Rangkaian percobaan SRFF dengan *clock*

Table 1.2: Kondisi terlarang, pacu, dan tak tentu karena perubahan *clock*

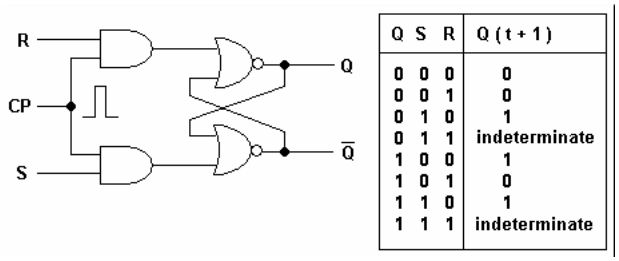
No.	S	R	*Clock*	Keterangan
1.	1	1	Aktif (1)	Kondisi terlarang
2.	1	1	Tepi turun (Berubah dari 1 ke 0)	Kondisi pacu
3.	1	1	Tidak aktif (0)	Kondisi tak tentu

Table 1.3: Kondisi terlarang, pacu, dan tak tentu karena perubahan *clock* dan masukan yang serempak

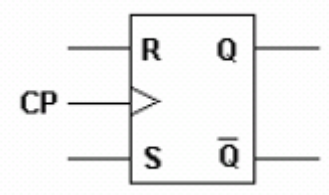
No.	S	R	*Clock*	Keterangan
1.	1	1	Aktif (1)	Kondisi terlarang
2.	1	1	Tepi turun	Kondisi pacu
3.	1	1	Tidak aktif (0)	Kondisi tak tentu

1.3.6 Set-Reset Flip-Flop dengan Clock (SRFF dengan Clock)

Dengan menambah beberapa gerbang pada bagian *input* rangkaian dasar, *flip-flop* tersebut hanya dapat merespon *input* selama terdapat *clock* pulsa. Output dari *flip-flop* tidak akan berubah selama *clock* pulsanya 0 meskipun terjadi perubahan pada *input*-nya. *Output flip-flop* hanya akan berubah sesuai dengan perubahan *input*-nya jika *clock* pulsa bernilai 1.



Q	S	R	Q (t + 1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	indeterminate
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	indeterminate

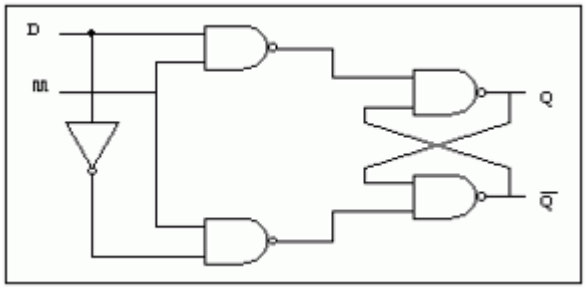


Gambar 1.5: SRFF dengan *clock*

1.3.7 Data Flip-Flop dengan Clock (DFF dengan Clock)

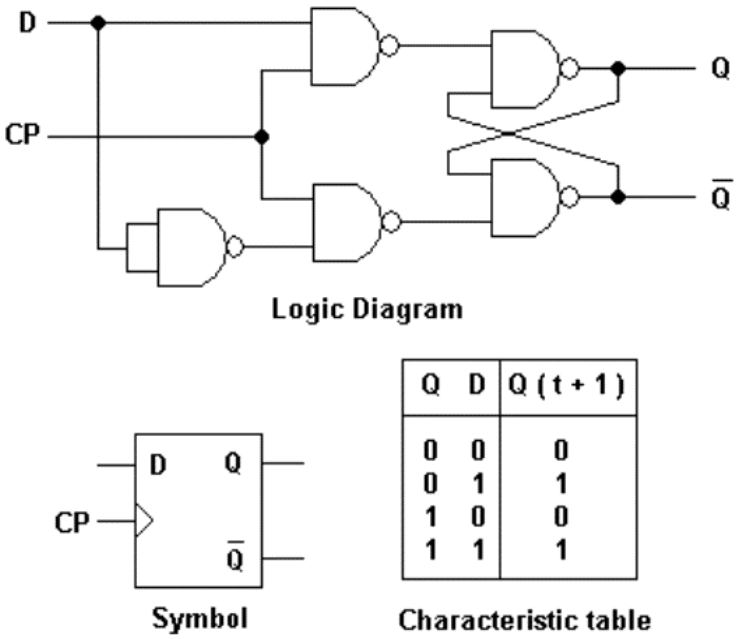
Flip-flop D dapat disusun dari *flip-flop* S-R atau *flip-flop* J-K yang masukannya saling berkebalikan. Hal ini dimungkinkan dengan menambahkan

salah satu masukannya dengan *inverter* agar kedua masukan *flip-flop* selalu dalam kondisi berlawanan. *Flip-flop* ini dinamakan dengan *flip-flop* data karena keluarannya selalu sama dengan masukan yang diberikan. Saat *flip-flop* pada keadaan aktif, masukan akan diteruskan ke saluran keluaran.



Gambar 1.6: Contoh rangkaian *flip-flop* DFF (Picu logika tinggi)

D *flip-flop* merupakan modifikasi dari RS *flip-flop* memakai *clock*. *Input* D disalurkan secara langsung ke S.

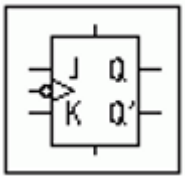


Gambar 1.7: DFF dengan *clock*

1.3.8 J-K Flip-Flop (JKFF)

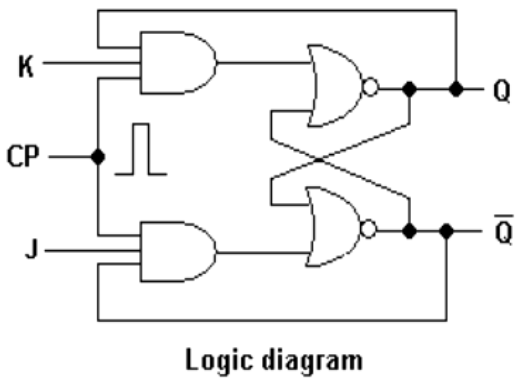
Flip-flop J-K merupakan penyempurnaan dari *flip-flop* R-S terutama untuk mengatasi masalah osilasi, yaitu dengan adanya umpan balik, serta masalah kondisi terlarang seperti yang telah dijelaskan di atas, yaitu pada kondisi masukan J dan K berlogika 1 yang akan membuat kondisi keluaran menjadi

berlawanan dengan kondisi keluaran sebelumnya atau dikenal dengan istilah *toggle*. Sementara untuk keluaran berdasarkan kondisi-kondisi masukan yang lain semua sama dengan *flip-flop* R-S.



Gambar 1.8: J-K *flip-flop*

State-state yang tidak didefinisikan pada RS *flip-flop*, pada JK *flip -flop* ini *state* tersebut didefinisikan. Jika pada RS *flip-flop* kondisi R dan S sama dengan 1, maka kondisi seperti ini tidak didefinisikan, maka pada JK *flip-flop* jika kondisi J dan K sama dengan 1 maka *output* JK *flip-flop* tersebut adalah komplemen dari *output* sebelumnya. Dalam hal ini J setara dengan S dan K setara dengan R. Untuk lebih jelasnya kita perhatikan diagram dibawah ini.

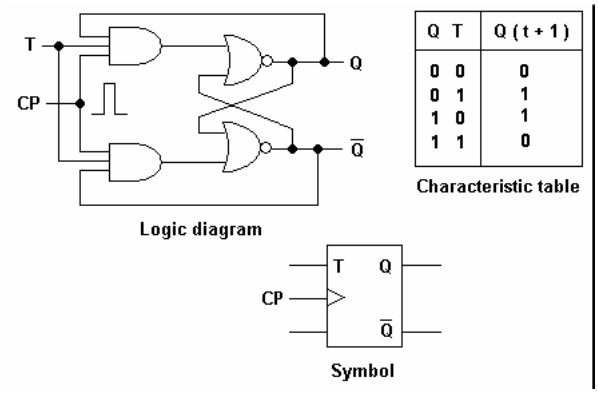


Q	J	K	Q (t + 1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Gambar 1.9: JKFF dengan *clock*

1.3.9 Toggle Flip-Flop (TFF)

Adalah versi JK *flip-flop* dengan *single input*. T *flip-flop* mempunyai kemampuan yaitu membuat *toggle* seperti pada Gambar 1.10.



Gambar 1.10: TFF dengan *clock*

1.3.10 Tabel Eksitasi Flip-Flop

Dibawah ini adalah karakteristik tabel dari berbagai tipe *flip-flop*. Nilai X menandakan bahwa nilainya dapat diisi kedua-duanya yaitu 0 dan 1.

Q (t)	Q (t+1)	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

(a) RS flip-flop

Q (t)	Q (t+1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

(b) JK flip-flop

Q (t)	Q (t+1)	D
0	0	0
0	1	1
1	0	0
1	1	1

(c) D flip-flop

Q (t)	Q (t+1)	T
0	0	0
0	1	1
1	0	1
1	1	0

(d) T flip-flop

Gambar 1.11: TFF dengan *clock*

1.4 Alat dan Bahan

a. Buku praktikum

b. Modul PSoC

c. Kabel *jumper*

d. Laptop

e. Kabel USB *male to* USB *female*

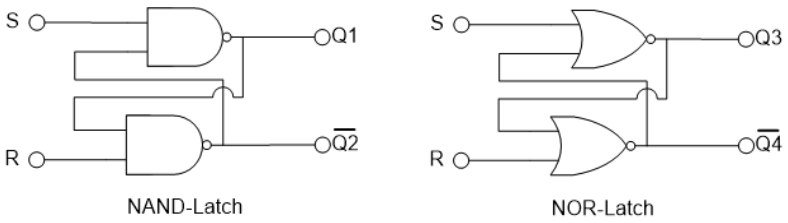
1.5 Metodologi

a. Peserta terlebih dahulu membaca dan mempelajari materi praktikum.

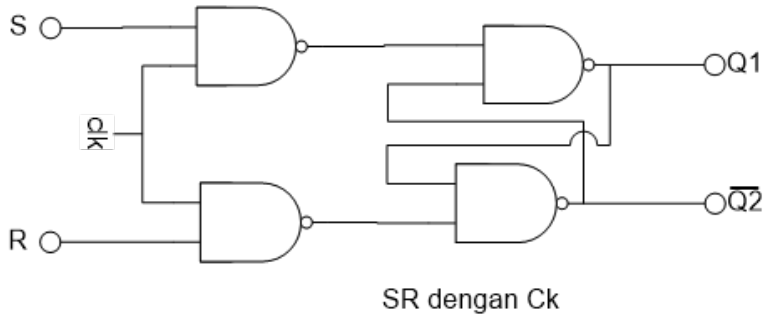
b. Pada awal pertemuan asisten menerangkan teori dan cara kerja percobaan.

c. Jenis-jenis *flip-flop* yang digunakan pada praktikum ini :

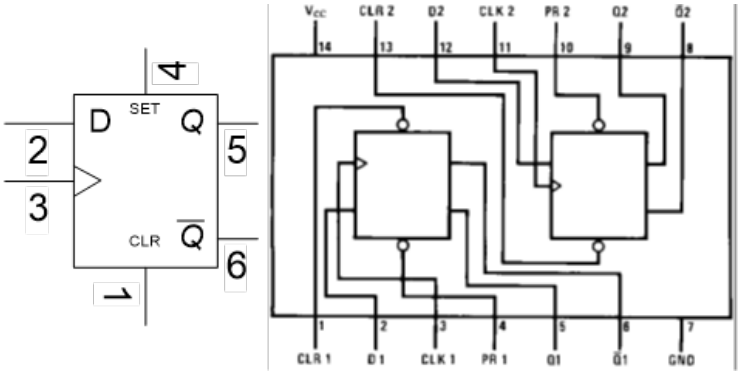
(a) NAND-*Latch* dan NOR-*Latch*
Latch SR pada dasarnya merupakan suatu piranti asinkron. Peralatan ini tidak beroperasi serempak dengan detak. Bila masukan (*set*) diaktifkan, maka keluaran normal segera diaktifkan seperti pada rangkaian kombinasional. Beda dari NAND-*latch* dengan NOR-*latch* hanya pada komponen penyusunnya, yaitu NAND-*latch* dengan gerbang NAND sedangkan NOR-*latch* dengan gerbang NOR.



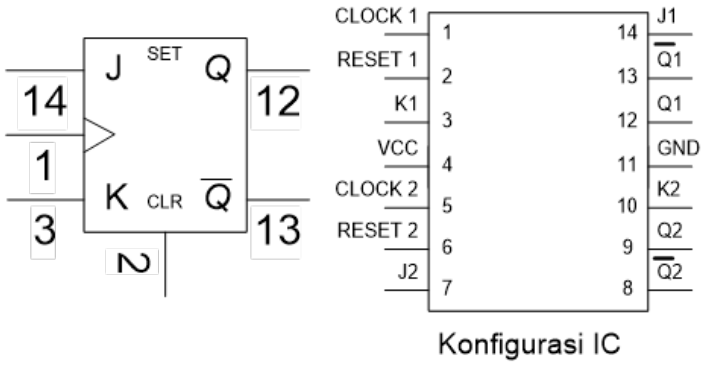
(b) SRFF dengan *Clock*
Rangkaian ini mirip dengan dengan *Latch* SR, hanya saja di sini terdapat *input* ekstra yaitu detak (*clock*). Flip-flop ini beroperasi secara sinkron yaitu selangkah dengan detak. Karakteristik dari SRFF yang berdetak ialah karakteristik memorinya. Ketika sekali di-*set* atau di-*reset* akan tetap pada keadaaanya kecuali bila masukannya dirubah.



(c) DFF dengan *Clock* (IC 74LS74)
Di dalam IC 74LS74 terdapat dua DFF yang independen. Pakai salah satu. Operasi yang diamati adalah bagaimana data dapat berpindah ke *output* Q. Pin *reset*, *data*, *clock* dan *set* adalah *input* dari saklar sedangkan Q dan Q' berupa LED.

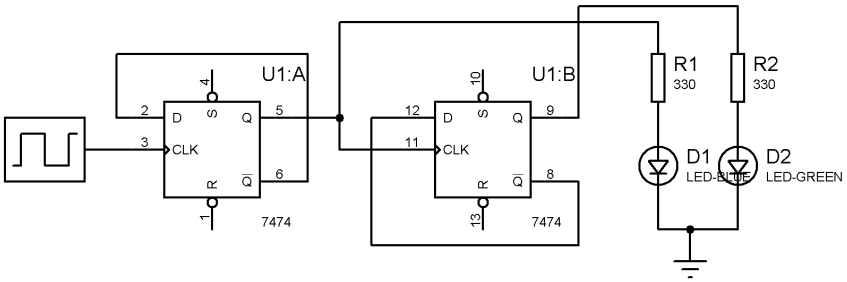


(d) JKK dengan *Clock* (IC 74HC73)
Seperti pada IC 74LS74 (DFF), IC 74HC73 (JKFF) juga memiliki 2 buah JKFF yang independen. Operasi yang diamati adalah saat *output* Q, dapat di-*reset*, di-*set*, di-*clear*, ataupun operasi *toggle*. Pin *reset*, J, dan K adalah *input* saklar, *clock* adalah *push-on (bounceless)*, *output* Q dan Q' berupa LED.



Konfigurasi IC

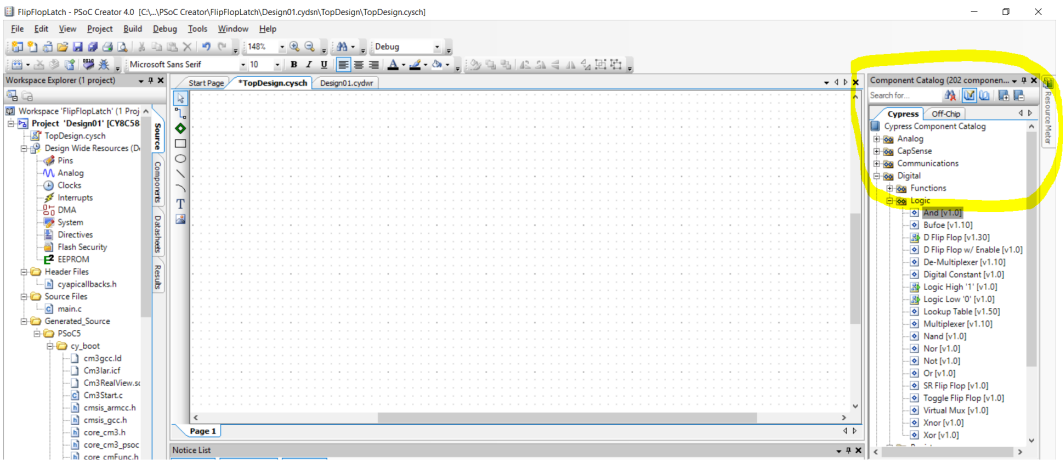
(e) Pembagi Frekuensi dengan DFF (IC 74LS74)
Rangkaian ini berfungsi untuk membagi frekuensi berapapun masukan ke CLK 1 menjadi setengahnya pada keluaran Q2, dan seterusnya akan dibagi 2 setiap di-*cascade* dengan *flip-flop* baru.



1.6 Tata Laksana

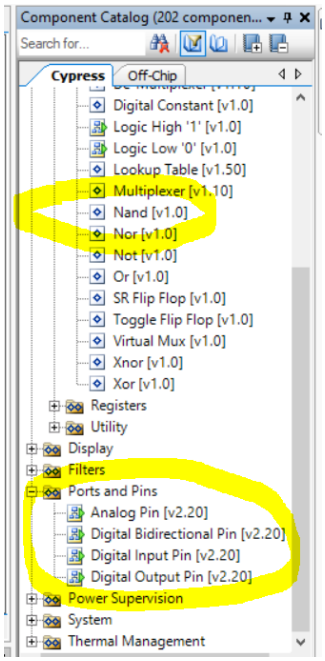
1.6.1 Praktikum 1 NAND-Latch dan NOR-Latch

1. Pertama adalah memilih komponen dari Component Catalog seperti Gambar 1.12 dibawah ini



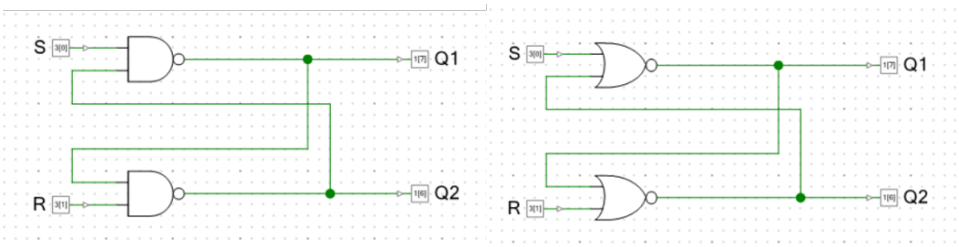
Gambar 1.12: *Component Catalog*

2. Pilih komponen yang dibutuhkan dengan cara di-*drag* ke kanvas. Untuk bagian pertama ini komponen yang dibutuhkan adalah komponen digital logic NAND, Ports and Pins → Digital Input Pin, dan Ports and Pins → Digital Output Pin seperti Gambar 1.13.



Gambar 1.13: *Drag Component*

3. Kemudian hubungkan komponen menggunakan *wire* sampai seperti Gambar 1.14 dibawah ini

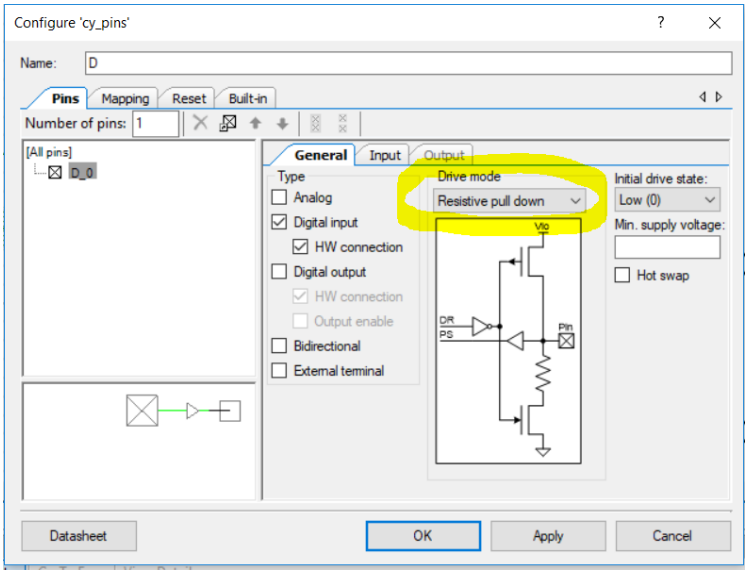


Gambar 1.14: Skema Rangkaian

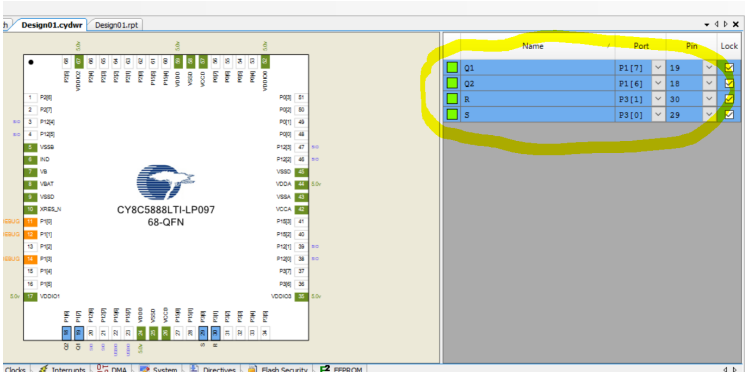
4. Untuk pengaturan *input* digital, klik 2 kali pada komponen *input* digital kemudian akan muncul jendela seperti dibawah ini. Atur Drive mode menjadi Resistive Pull Down dengan Initial drive state : Low (0). Untuk setiap *input* digital selanjutnya pengaturannya sama seperti Gambar 1.15.

5. Setelah itu dilanjutkan mengatur pin, dilakukan dengan klik 2 kali pada Pins Kemudian atur *port* sesuai dengan keinginan seperti Gambar 1.16.

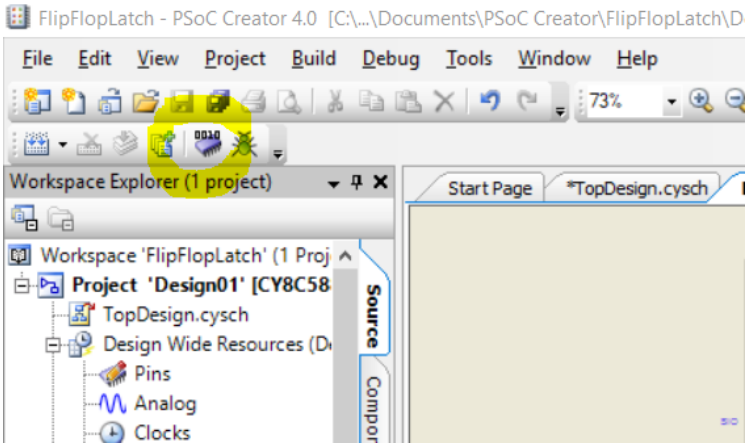
6. Untuk melakukan pemrograman dapat diklik *icon* seperti Gambar 1.17, atau bisa juga melalui Debug → Programs



Gambar 1.15: Pengaturan *Drive Mode*



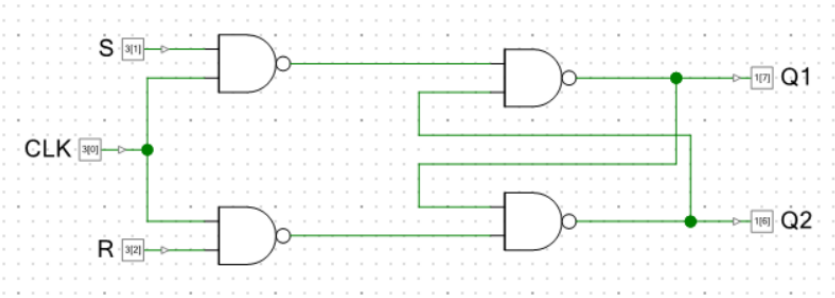
Gambar 1.16: Pengaturan Port



Gambar 1.17: Icon Program

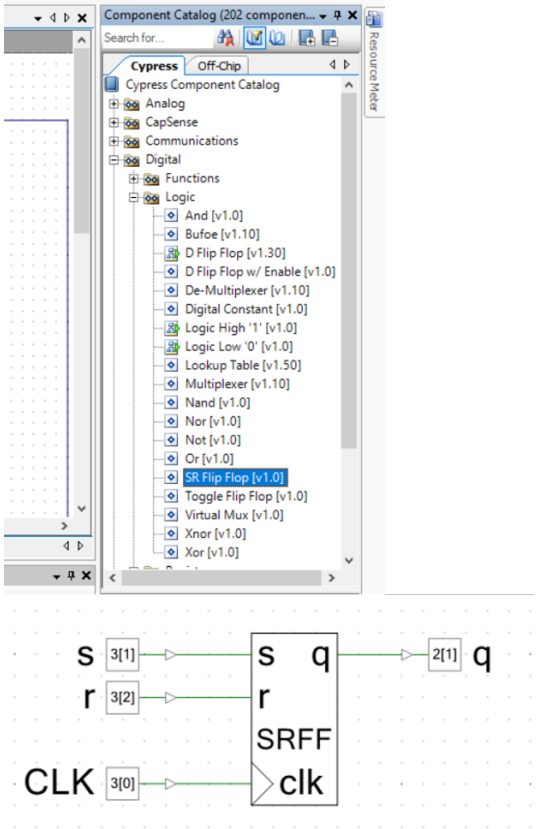
1.6.2 SRFF dengan Clock

1. Rangkai rangkaian seperti Gambar 1.18, dan atur dengan cara seperti yang dijelaskan di penjelasan sebelumnya



Gambar 1.18: Skema Rangkaian SRFF

2. Pada rangkaian SRFF dapat juga menggunakan blok diagram SRFF yang telah disediakan oleh PSoC seperti Gambar 1.19.

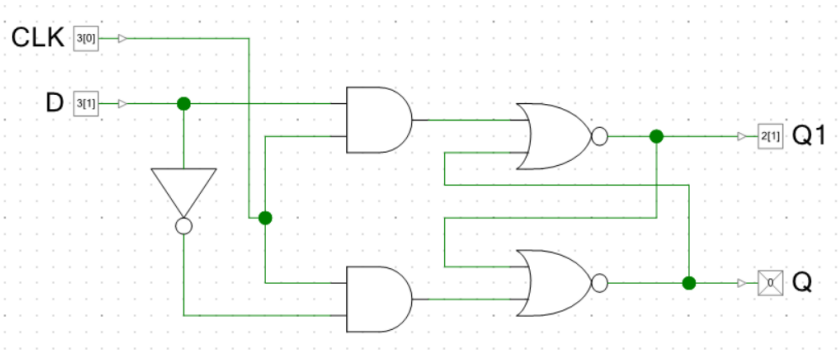


Gambar 1.19: Blok Skema Rangkaian SRFF

3. Rangkaian dapat dibuat dengan gerbang logika ataupun blok diagram yang ada secara langsung. Catat hasil sesuai dengan tabel yang disediakan.

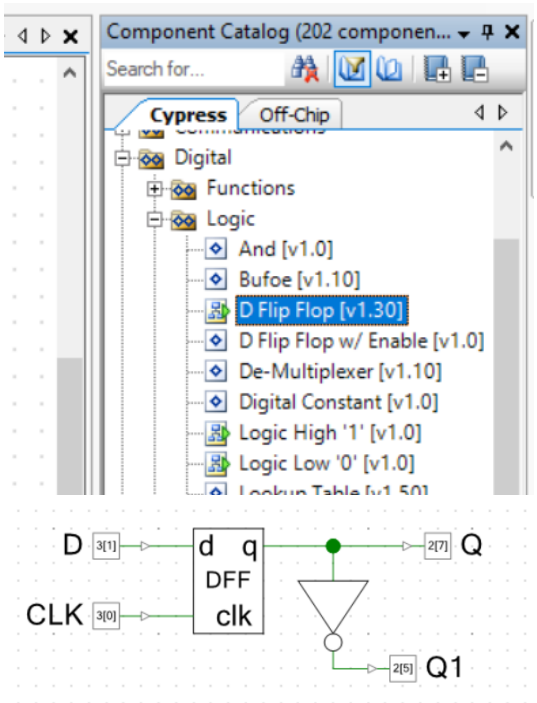
1.6.3 DFF dengan Clock

1. Rangkai rangkaian seperti dibawah ini, dan atur dengan cara seperti yang dijelaskan dipenjelasan sebelumnya



Gambar 1.20: Skema rangkaian DFF dengan *Clock*

2. Pada rangkaian DFF dengan *clock* dapat juga menggunakan blok diagram DFF yang telah disediakan oleh PSoC seperti Gambar 1.21

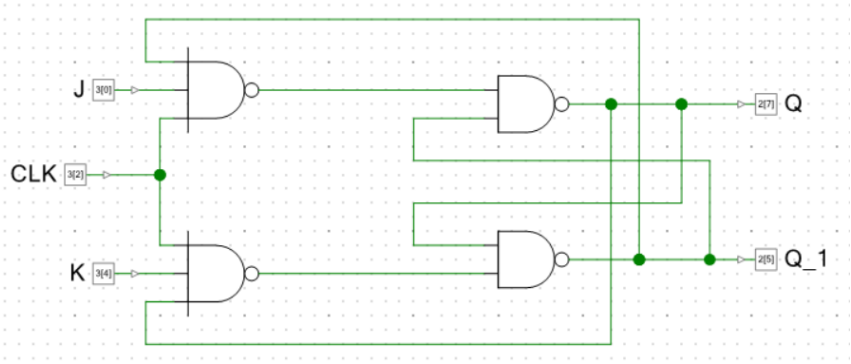


Gambar 1.21: Blok Skema Rangkaian DFF dengan *Clock*

3. Rangkaian dapat dibuat dengan gerbang logika ataupun blok diagram yang ada secara langsung. Catat hasil sesuai dengan tabel yang disediakan.

1.6.4 JK Flip-Flop

1. Rangkai rangkaian seperti Gambar 1.22, dan atur dengan cara seperti yang dijelaskan dipenjelasan sebelumnya

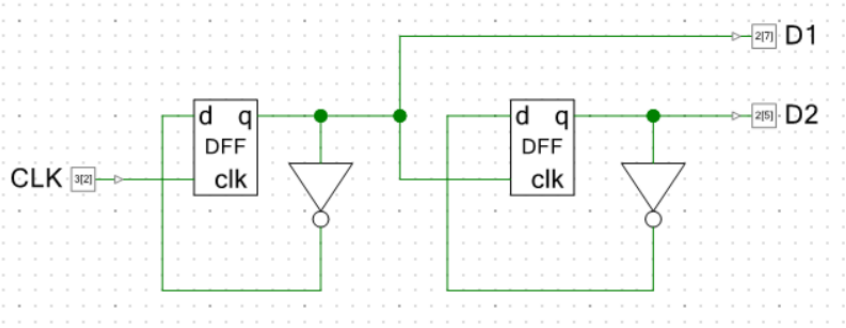


Gambar 1.22: Skema Rangkaian JKFF

2. Catat hasil sesuai dengan tabel yang disediakan

1.6.5 Pembagi Frekuensi dengan DFF

1. Rangkai rangkaian seperti Gambar 1.23, dan atur dengan cara seperti yang dijelaskan dipenjelasan sebelumnya



Gambar 1.23: Blok Skema Rangkaian Pembagi Frekuensi dengan DFF

2. Catat hasil sesuai dengan tabel yang disediakan

1.7 Tabel Pengamatan

1. NAND-Latch dan NOR-Latch

Step	INPUT		OUTPUT			
			NAND-Latch		NOR-Latch	
	S	R	Q1	Q2	Q3	Q4
1	0	0				
2	0	1				
3	1	1				
4	1	0				
5	0	0				
6	1	1				
7	0	1				
8	0	0				
9	1	0				
10	1	1				

2. SRFF dengan *Clock*

No	INPUT			OUTPUT	
	CLK	S	R	Q1	Q2
1	0	0	0		
2		0	1		
3	0	1	1		
4		1	1		
5	0	1	0		
6		1	0		
7	0	0	0		
8		0	0		
9		1	0		
10		1	1		

3. DFF dengan *Clock*

No	INPUT		OUTPUT	
	CLK	DATA	Q	$\overline{Q}$
1	0	1		
2	1	0		
3	1	1		
4	0	0		
5	1	0		
6	1	1		
7	0	1		
8	1	0		
9	1	1		
10	0	0		
11	1	0		
12	1	1		
13	0	1		
14	1	1		
15	0	0		
16	1	0		

4. JKFF dengan *Clock*

No	INPUT				OUTPUT	
	RESET	CLK	J	K	Q	$\overline{Q}$
1	**0**	**0**	**0**	**0**		
2	**0**	**1**	**0**	**1**		
3	**0**		**1**	**0**		
4	**0**	**0**	**1**	**1**		
5	**1**	**0**	**1**	**0**		
6	**1**		**1**	**0**		
7	**1**		**0**	**1**		
8	**1**		**1**	**0**		
9	**1**		**1**	**1**		
10	**0**		**1**	**1**		
11	**1**		**1**	**1**		
12	**1**		**1**	**1**		

5. Pembagi Frekuensi dengan DFF dan *Clock*

Timeline Table

Clk																
Q1																
Q2																

DAFTAR PUSTAKA

David Bucchlah, Wayne McLahan, "Applied Electronic Instrumentation And Measurement", MacMilian Publishing Company, 1992.

Eggebrecht, Lewis C., Interfacing to The IBM PC, Howard W. Sams & Co., Indianapolis, 1987.

Hall, Douglas V., Microprocessor and Interfacing : Programming and Hardware, McGraw-Hill Book Company, Singapore, 1987.

Hodges D. , Jacson, Nasution S."Analisa dan Desain Rangkaian Terpadu Digital", Erlangga, Jakarta, 1987.

Ian Robertson Sinclair, Suryawan,"Panduan Belajar Elektronik Digital", ElexMedia Komputindo, Jakarta, 1993.

K.F. Ibrahim,"Teknik Digital", Andi Offset , Jakarta, 1996.

Sendra, Smith, Keneth C.," Rangkaian Mikroelektronika", Penerbit Erlangga, Jakarta, 1989.

Singh, Avtar & Walter A. Triebel, The 8088 Microprocessor : Programming, Interfacing, Software, Hardware and Applications, Prentice-Hall International Inc., New Jersey, 1987.

Sofyan H. Nasution, "Analisa dan Desain Rangkaian Terpadu Digital", Penerbit Erlangga, Jakarta, 1987.

Sutrisno,"Rangkaian Digital dan Rancangan Logika", Erlangga, Jakarta, 1990.

Tokheim. R., "Elektronika Digital", Edisi Kedua, Erlangga, Jakarta, 1995.

Wijaya Widjanarka N., "Teknik Digital", Erlangga, Jakarta, 2006.