



LAPORAN PRAKTIKUM

IF310303

PRAKTIKUM SISTEM DIGITAL

MODUL: 7

FLIP - FLOP

NAMA : Muhammad Alwiza Ansyar

NIM : M0520051

HARI : Jumat

TANGGAL : 27 November 2020

WAKTU : 10.15 – 11.05 WIB

ASISTEN : Akhtar Bariq Rahman

PROGRAM STUDI INFORMATIKA

UNIVERSITAS SEBELAS MARET

2020

Modul 7

FLIP - FLOP

Muhammad Alwiza Ansyar (M0520051) / 27 November 2020

Email : alwiza21@student.uns.ac.id

Asisten : Akhtar Bariq Rahman

Abstraksi— Berikut merupakan laporan praktikum untuk modul 7 yang memiliki fokus bahasan tentang berbagai macam Flip - Flop . Flip – Flop adalah salah satu rangkaian logika sekuensial yang dapat menyimpan informasi sebesar 1 bit. Pada laporan praktikum ini, akan dibahas tentang berbagai pengekspresian Flip – Flop dalam rangkaian logika serta tabel kebenaran dan tabel karakteristiknya.

Kata kunci— Flip – Flop

I. PENDAHULUAN

Ada saatnya dimana kita tidak mau keluaran terus berubah karena perubahan *input*. Kita ingin supaya keadaan *output* dapat ditahan/*hold*. Maka, diciptakanlah rangkaian memori. Rangkaian memori adalah rangkaian sekuensial yang dapat menyimpan informasi. Rangkaian memori tersebut salah satunya adalah Flip – Flop.

Flip – Flop adalah rangkaian memori terkecil yaitu hanya bisa menyimpan informasi sebesar 1 bit. Rangkaian Flip - Flop menjadi dasar yang mana akan dikombinasikan supaya memori dapat menyimpan data yang lebih besar lagi.

II. DASAR TEORI

2.1 Rangkaian Sekuensial

Rangkaian sekuensial merupakan rangkaian kombinasional yang dipadukan dengan unit *Memory*. *Output* selanjutnya (Q_{n+1}) dari rangkaian sekuensial dipengaruhi oleh keadaan-keadaan *input* dan keadaan-keadaan *output* pada saat yang sama (Q_n). Unit *Memory* menyimpan keadaan-keadaan *output* saat itu (Q_n) lalu menyalurkannya kembali menjadi *input* dari rangkaian sekuensial (disebut *feedback*).

Rangkaian sekuensial dapat bekerja secara sinkronus maupun asynchronous. Sinkronus berarti *input* disalurkan masuk ke rangkaian secara serentak/waktu yang sinkron sedangkan asinkronus berarti *input* disalurkan masuk ke rangkaian secara independen/waktu yang tidak sinkron. Cara kerja asinkronus lebih sering dihindari karena berpotensi terjadinya perubahan *input* yang tidak disengaja (yang mana langsung mengubah keadaan *output*) dan sukar dicari kesalahannya. Cara kerja sinkronus memiliki *input* tambahan yaitu *control input* yang bisa berupa Enable maupun Clock. Pada saat *control input* berlogika rendah, maka tidak akan terjadi perubahan pada *output* walaupun keadaan *input* berubah-ubah.

2.2 Flip – Flop

Flip – Flop atau FF adalah rangkaian sekuensial yang dapat menyimpan informasi sebesar 1 bit sehingga FF adalah unit *Memory* terkecil. *Output* dari FF berjumlah dua yang keduanya saling berkebalikan/komplemen terbalik yaitu Q dan Q'. FF memiliki *output* “*Memory*” yang berarti rangkaian akan menampilkan *output* sebelumnya.

2.3 Flip – Flop RS Clocked

Flip – Flop RS Clocked atau FF RSC adalah FF sinkron yang merupakan dasar dari semua jenis FF. Rangkaian ini terdiri dari *input* Set (S) dan Reset (R). Berikut tabel kebenarannya

Clock	S	R	Q	Q'
0	×	×	<i>Memory</i>	
1	0	0	<i>Memory</i>	
1	0	1	0	1
1	1	0	1	0
1	1	1	<i>Not Used</i>	

FF RSC memiliki kelemahan yaitu saat S = R = 1 maka Q dan Q' bernilai 1. Hal ini tentu bertentangan terhadap definisi dari Q dan Q' yaitu saling berkebalikan. Maka dari itu, keadaan *input* tersebut tidak dipakai (*Not Used*).

2.4 Flip – Flop RS Clocked dengan Preset dan Clear

Flip – Flop ini merupakan modifikasi dari FF RSC dimana ditambahkan *input* Preset dan Clear. Kedua *input* tersebut bekerja secara asinkronus terhadap *output* saat itu sehingga tidak terpengaruh oleh clock. Modifikasi ini bertujuan supaya kita dapat mengubah terlebih dahulu keadaan *output* sebelum dioperasikan pada FF RSC. Berikut tabel kebenarannya.

Preset	Clear	Clock	S	R	Q	Q'
1	1	0	×	×	<i>Memory</i>	
1	1	1	0	0	<i>Memory</i>	
1	1	1	0	1	0	1
1	1	1	1	0	1	0
1	1	1	1	1	<i>Not Used</i>	

Preset	Clear	Clock	S	R	Q	Q'
0	1	×	×	×	1	0
1	0	×	×	×	0	1
0	0	×	×	×	<i>Not Used</i>	

Preset dan Clear bekerja secara inverted. Keadaan Preset dan Clear harus dikembalikan menjadi 11 supaya FF RSC bisa dioperasikan. Seperti pada FF RSC, keadaan 00 perlu dihindari karna akan mengakibatkan keadaan Q dan Q' bernilai 1.

2.5 Flip – Flop JK

Flip – Flop JK atau FF JK adalah modifikasi dari FF RSC dengan tujuan supaya membuat *input* S=R=1 dapat digunakan. *Input* S berubah menjadi J dan *input* R berubah menjadi K. Di FF ini, saat J=K=1, akan terjadi *toggle* yaitu keluaran menampilkan komplemen dari memori (Qn'). Keadaan *toggle* ini hanya akan

terjadi bila *Clock On's time* lebih besar daripada *Flip – Flop's propagation delay*. Berikut tabel kebenarannya.

Clock	J	K	Q	Q'
0	×	×	Memory	
1	0	0	Memory	
1	0	1	0	1
1	1	0	1	0
1	1	1	Toggle*	

Toggle= Toggle belum sempurna*

2.6 Flip – Flop T

Flip – Flop T atau FF T adalah modifikasi dari FF JK dimana hanya digunakan satu *input* yang lalu dihubungkan ke setiap *terminal input* dari J dan K. Dengan ini didapat FF JK yang hanya memiliki *input* 00 dan 11. Fungsi dari FF T adalah sebagai rangkaian *toggle*. Seperti pada FF JK, keadaan *toggle* ini hanya akan terjadi bila *Clock On's time* lebih besar daripada *Flip – Flop's propagation delay*. T dari nama FF ini berarti “Toggle”. Berikut tabel kebenarannya.

Clock	T	Q	Q'
0	×	Memory	
1	0	Memory	
1	1	Toggle*	

Toggle= Toggle belum sempurna*

2.7 Flip – Flop JK Master-Slave

Flip – Flop JK Master-Slave atau FF JKMS adalah modifikasi dari FF JK. Seperti yang sudah dijelaskan diatas, keadaan *toggle* hanya akan terjadi bila *Clock On's time* lebih besar daripada *Flip – Flop's propagation delay*. Pada praksisnya, hal tersebut sukar dilakukan. Jika tidak memenuhi syarat tersebut, maka keluaran akan menjadi *race around*, yaitu keadaan di mana *output* Q dan Q' akan terus menerus bernilai 01, 10, 10 10, ... sampai clock bernilai 0. Keadaan *race around* juga bisa disebut “*toggle* yang tak terkendali”. Maka dari itu, alternatif lain supaya keadaan *toggle* dapat bekerja adalah dengan menggunakan modifikasi Master-Slave. Berikut tabel kebenarannya.

Clock	J	K	Q
0	×	×	Memory
1	0	0	Memory
1	0	1	0
1	1	0	1
1	1	1	Toggle

Master-Slave atau Tuan-Budak adalah modifikasi dimana dibuat dua FF yang mirip lalu ditentukan FF yang sebagai *master* dan FF yang sebagai *slave*. FF master hanya akan bekerja bila clock bernilai 1 sedangkan FF slave akan bekerja bila clock bernilai 0. Modifikasi Master-Slave membuat rangkaian menjadi bersifat *negative-edge triggered*. Dengan itu, keadaan *race around* bisa diatasi.

2.8 Flip – Flop D

Flip – Flop D atau FF D adalah modifikasi dari FF JK dimana hanya digunakan satu *input* yang akan disambungkan ke *terminal input* J secara normal dan *terminal input* K secara berkebalikan. Dengan ini didapat FF JK yang hanya memiliki *input* 01 dan 10 (yang menjadi *input* D = 0 dan D = 1). D pada FF ini adalah singkatan dari “Data”. Berikut tabel kebenarannya.

Clock	D	Q	Q'
0	×	Memory	
1	0	0	1
1	1	1	0

III. ALAT DAN LANGKAH PERCOBAAN

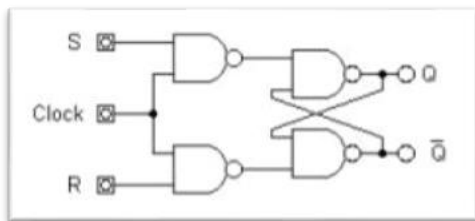
3.1 Alat

1. PC/Laptop
2. Aplikasi Digital Works

3.2 Langkah Percobaan

Flip - Flop RSC

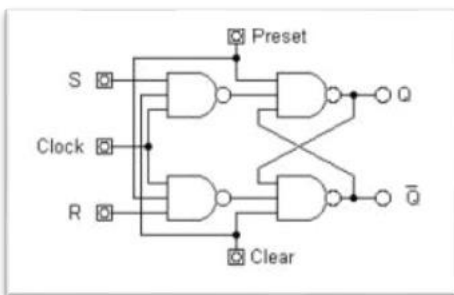
1. Buka aplikasi Digital Works
2. Buatlah rangkaian berikut



3. Amati dan catat *output* terhadap kombinasi keadaan *input*

Flip - Flop RSC dengan Preset dan Clear

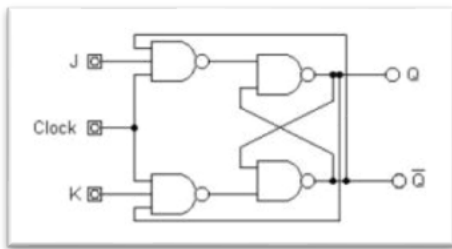
4. Buat rangkaian berikut



5. Amati dan catat *output* terhadap kombinasi keadaan *input*

Flip - Flop JK

6. Buat Rangkaian berikut



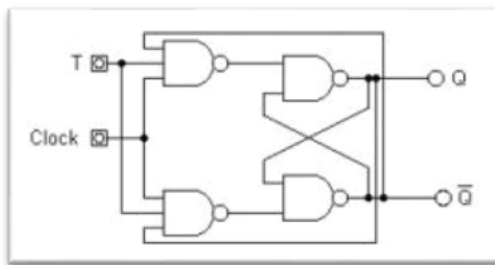
7. Amati dan catat *output* terhadap kombinasi keadaan *input*

8. Buat rangkaian menggunakan IC 7473

9. Amati dan catat *output* terhadap kombinasi keadaan *input*

Flip - Flop T

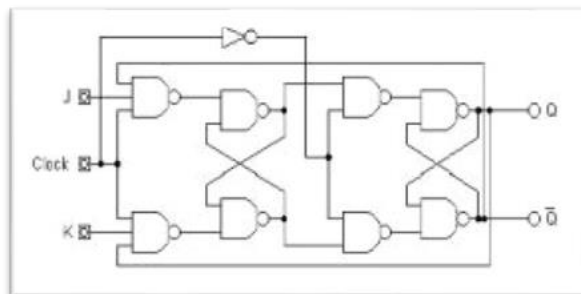
10. Buat rangkaian berikut



11. Amati dan catat *output* terhadap kombinasi keadaan *input*

Flip - Flop JK Master-Slave

12. Buatlah rangkaian berikut



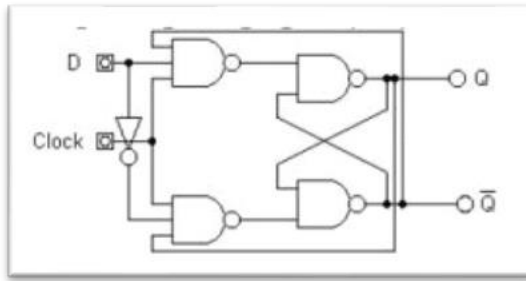
13. Amati dan catat *output* terhadap kombinasi keadaan *input*

14. Buatlah rangkaian menggunakan IC 7472

15. Amati dan catat *output* terhadap kombinasi keadaan *input*

Flip - Flop D

16. Buatlah rangkaian berikut

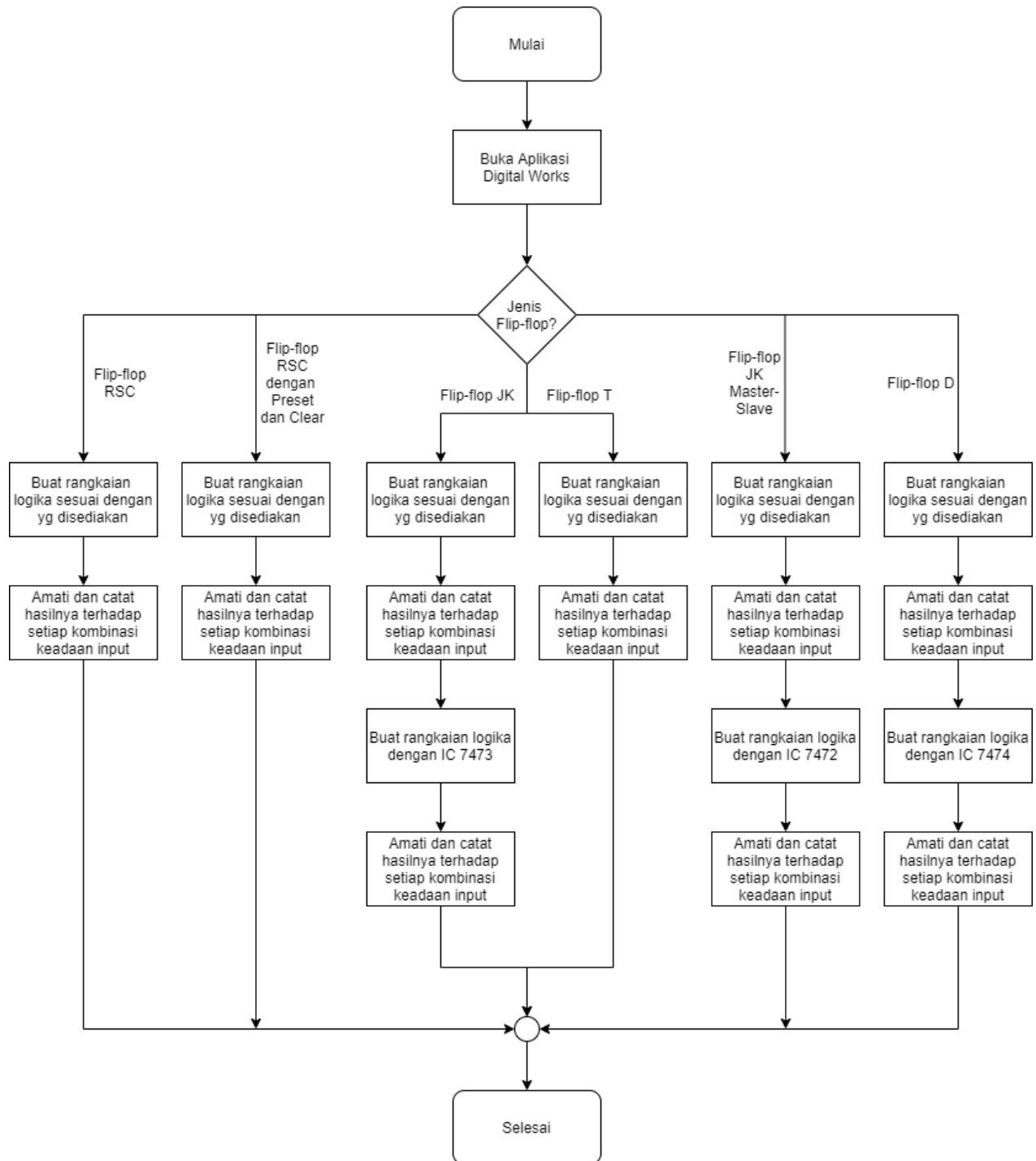


17. Amati dan catat *output* terhadap kombinasi keadaan *input*

18. Buatlah rangkaian menggunakan IC 7474

19. Amati dan catat *output* terhadap kombinasi keadaan *input*

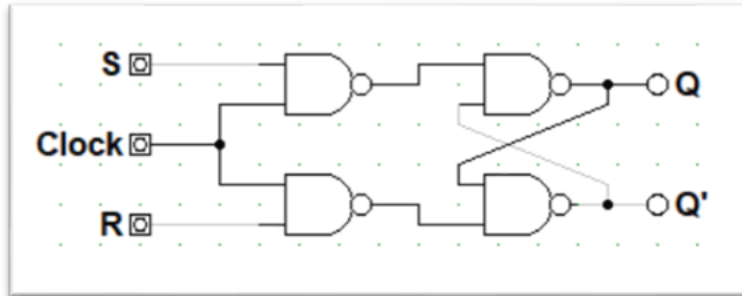
Diagram alur:



IV. HASIL DAN ANALISIS PERCOBAAN

4.1 Flip - Flop RSC

Rangkaian logika:



Tabel kebenaran:

Clock	S	R	Q	Q'
0	x	x	Memory	
1	0	0	Memory	
1	0	1	0	1
1	1	0	1	0
1	1	1	Not Used	

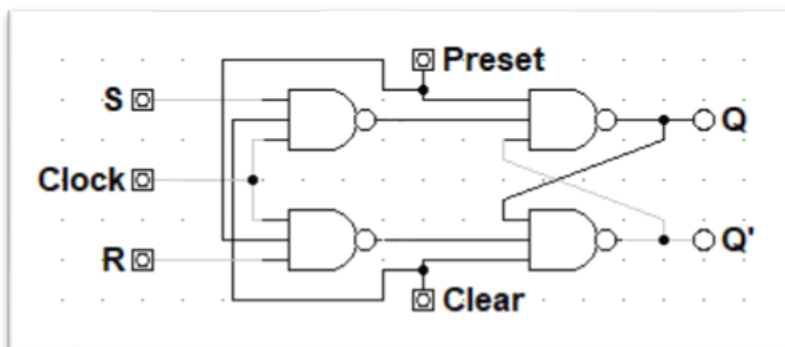
Tabel karakteristik (dengan clock on secara konstan):

Q _n	S	R	Q _{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	Not Used
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	Not Used

FF RSC memiliki kelemahan di mana saat $S=R=1$, *output* menjadi $Q=Q'=1$ yang mana hal tersebut adalah bertentangan dengan definisi komplementer. Maka dari itu, keadaan *input* tersebut tidak dipakai (*Not Used*).

4.2 Flip - Flop RSC dengan Preset dan Clear

Rangkaian logika:



Tabel kebenaran:

Preset	Clear	Clock	S	R	Q	Q'
0	1	×	×	×	1	0
1	0	×	×	×	0	1
0	0	×	×	×	Not Used	

Preset	Clear	Clock	S	R	Q	Q'
1	1	0	×	×	Memory	
1	1	1	0	0	Memory	
1	1	1	0	1	0	1
1	1	1	1	0	1	0
1	1	1	1	1	Not Used	

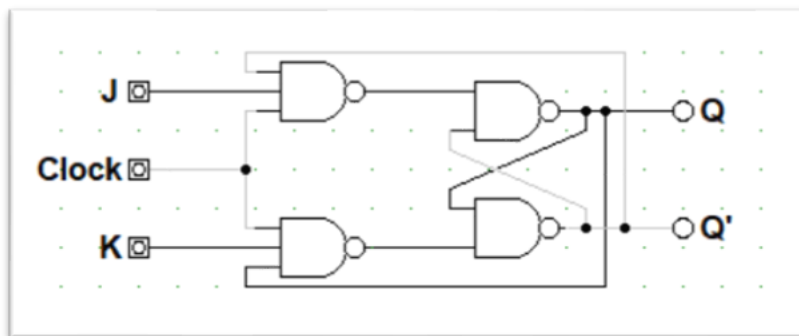
Tabel karakteristik (dengan clock on secara konstan):

Q _n	S	R	Q _{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	Not Used
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	Not Used

Flip - Flop ini merupakan modifikasi dari FF RSC dengan menambahkan Preset dan Clear. Preset dan Clear dapat memodifikasi Q_n secara asinkronus.

4.3 Flip - Flop JK

Rangkaian logika:



Tabel kebenaran:

Clock	J	K	Q	Q'
0	×	×	Memory	
1	0	0	Memory	
1	0	1	0	1
1	1	0	1	0
1	1	1	Toggle*	

Toggle* = Toggle belum sempurna

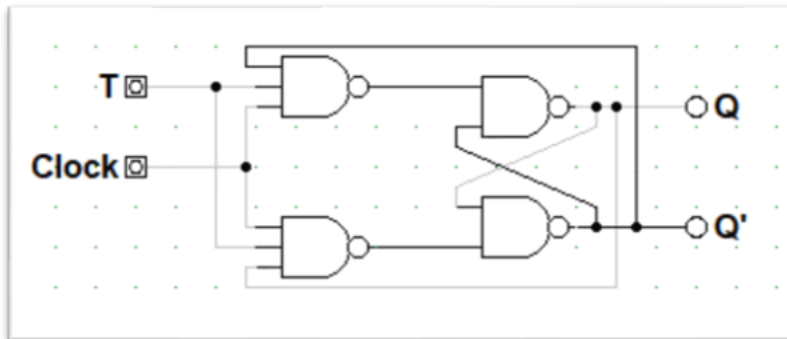
Tabel karakteristik (dengan clock on secara konstan):

Q _n	J	K	Q _{n+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

Flip - Flop ini merupakan modifikasi FF RSC yang membuat keadaan S=R=1 dapat digunakan, yaitu keadaan *toggle*. Namun keadaan *toggle* sukar didapatkan sehingga berakibat terjadinya *race around*.

4.4 Flip - Flop T

Rangkaian logika:



Tabel kebenaran:

Clock	T	Q	Q'
0	x	Memory	
1	0	Memory	
1	1	Toggle*	

Toggle= Toggle belum sempurna*

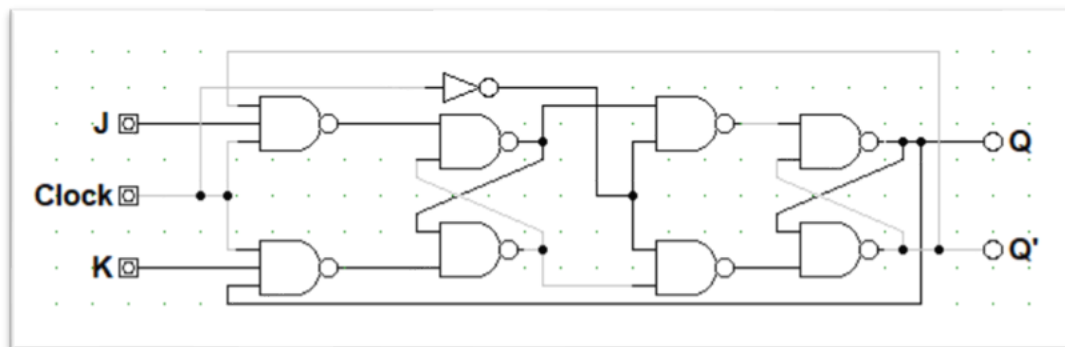
Tabel karakteristik (dengan clock on secara konstan):

Q _n	T	Q _{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

Flip - Flop T merupakan modifikasi dari FF JK dimana hanya dipakai satu *input* yang lalu dihubungkan di kedua *terminal input* J dan K. FF T hanya dapat melakukan *toggle*. Seperti FF JK, keadaan *toggle* juga sukar didapat sehingga berakibat terjadinya *race around*.

4.5 Flip - Flop JK Master-Slave

Rangkaian logika:



Tabel kebenaran:

Clock	J	K	Q	Q'
0	×	×	Memory	
1	0	0	Memory	
1	0	1	0	1
1	1	0	1	0
1	1	1	Toggle	

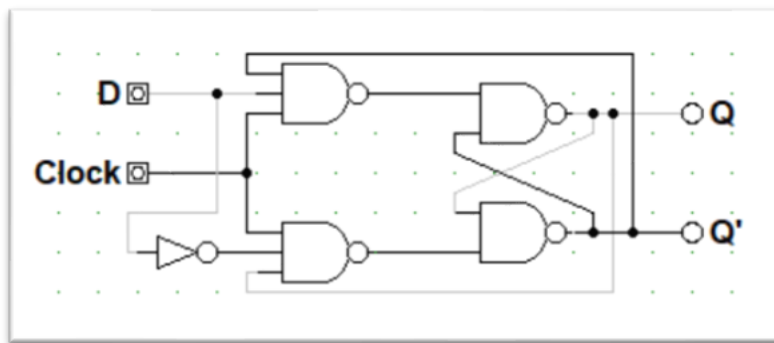
Tabel karakteristik (dengan clock on secara konstan):

Q _n	J	K	Q _{n+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

Flip - Flop ini merupakan modifikasi FF JK yang mengatasi masalah *toggle*. Dengan FF berbentuk Master-Slave, clock akan bersifat *negative-edge triggered*. Hal tersebut membuat keadaan *toggle* dapat diraih.

4.6 Flip - Flop D

Rangkaian logika:



Tabel kebenaran:

Clock	D	Q	Q'
0	×	Memory	
1	0	0	1
1	1	1	0

Tabel karakteristik (dengan clock on secara konstan):

Q _n	D	Q _{n+1}
0	0	0
0	1	1
1	0	0
1	1	1

Flip - Flop ini merupakan modifikasi dari FF JK di mana hanya digunakan satu *input* dan dihubungkan ke *terminal input* J secara normal dan *terminal input* K secara berkebalikan/*inverted*. FF D dapat melakukan kinerja FF RSC tanpa harus khawatir dengan keadaan S=R=1 karena tidak mungkin terjadi di FF D.

V.KESIMPULAN

Flip – Flop atau FF adalah rangkaian sekuensial yang juga rangkaian memori. FF dapat menyimpan informasi sebesar 1 bit karena itu FF adalah unit memori terkecil. FF biasanya akan dirangkai lagi secara

kompleks supaya dapat menampung informasi yang lebih besar lagi. FF dapat bekerja secara sinkronus (yaitu menggunakan clock) dan asinkronus

FF yang paling sederhana adalah RS (Reset dan Set). Dari FF RS ini, terciptalah berbagai modifikasi-modifikasi FF lainnya. Pada praktikum ini, dibahas FF RSC, FF RSC dengan Preset dan Clear, FF JK, FF T, FF JK Master-Slave, dan FF D.

FF RSC adalah FF RS dengan menggunakan clock. FF RSC dengan Preset dan Clear adalah modifikasi dengan menambahkan Preset dan Clear yang mampu mengubah *output* secara asinkronus. FF JK adalah modifikasi yang menangani keadaan *Not Used* yang dimiliki FF RSC, FF T adalah modifikasi yang hanya memiliki fungsi *toggle*, FF JK Master-Slave adalah modifikasi yang menyempurnakan FF JK supaya tidak terjadi keadaan *race around*, dan FF D adalah modifikasi yang menghindari mendapati *output Not Used* pada FF RSC.

Pada praktikum ini, dibuat keenam Flip - Flop tadi pada aplikasi Digital Works serta tabel kebenaran dan tabel karakteristik. Tabel karakteristik adalah tabel dengan *input* clock selalu dalam keadaan 1 dan menganggap *output* saat itu (Q_n) sebagai *input*. Setelah dianalisis untuk setiap kombinasi keadaan *input*, didapatkan tabel-tabel tersebut yang lalu disajikan pada laporan praktikum ini.

Keadaan *toggle* adalah keadaan yang merubah Q_n menjadi Q_n' . Keadaan ini sukar didapat karena keadaan *toggle* ini hanya akan terjadi bila *Clock On's time* lebih besar daripada *Flip – Flop's propagation delay*. *Flip – Flop's propagation delay* adalah senjang waktu yang dibutuhkan Flip - Flop untuk dapat memproses *input* dan menampilkan *output*. Hal tersebut tentulah sulit karena kita tidak bisa dengan mudah mengatur *delay* Flip - Flop tersebut. Maka dari itu, masalah tadi ditangani dengan memodifikasi rangkaian menjadi Master-Slave atau MS. Bentuk MS membuat *Clock On's time* menjadi bersifat *negative-edge triggered* atau perubahan terjadi saat clock diubah dari *High* ke *Low*. Dengan itu, akan didapat keadaan dimana *Clock On's time* lebih besar daripada *Flip – Flop's propagation delay*. *Flip – Flop's propagation delay* sehingga keadaan *toggle* dapat diraih.

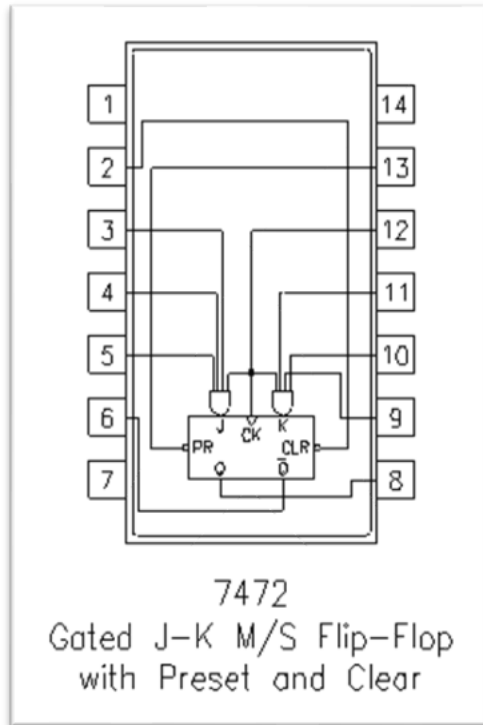
VI. DAFTAR PUSTAKA

- Anonim. 2019. *Pengertian dan Macam-Macam Rangkaian Flip - Flop*, diakses dari <https://katakoala.com/pengertian-dan-macam-macam-rangkaian-Flip - Flop />, pada 30 November 2020
- Maulani, Bismar. 2011. *Flip - Flop Input Sinkron dan Asinkron*, diakses dari <http://bismarmaulani.blogspot.com/2011/09/Flip - Flop -input-sinkron-dan-asinkron.html>, pada 30 November 2020

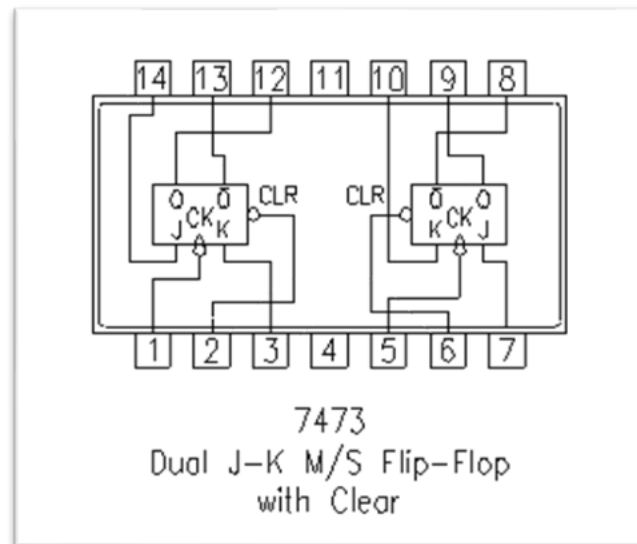
[Muhammad Alwiza Ansyar. Saya adalah seorang mahasiswa yang berasal dari Bogor. Saat ini, saya sedang menempuh pendidikan di Universitas Sebelas Maret jurusan Informatika.....]

VII. LAMPIRAN

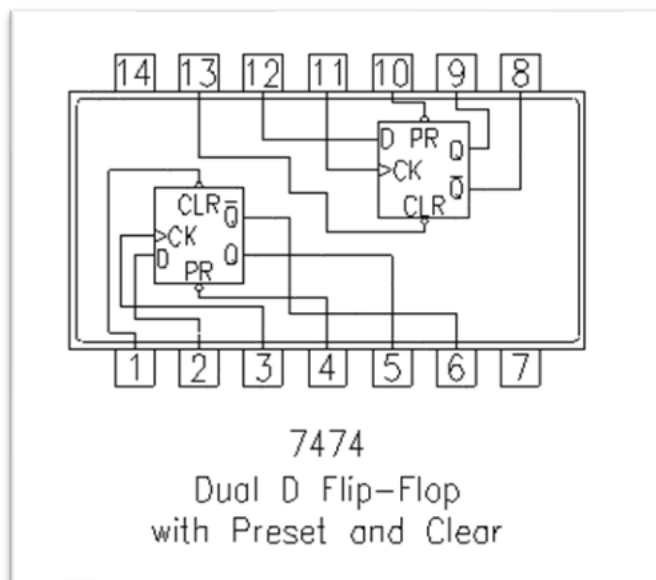
Pinout IC 74HC72:



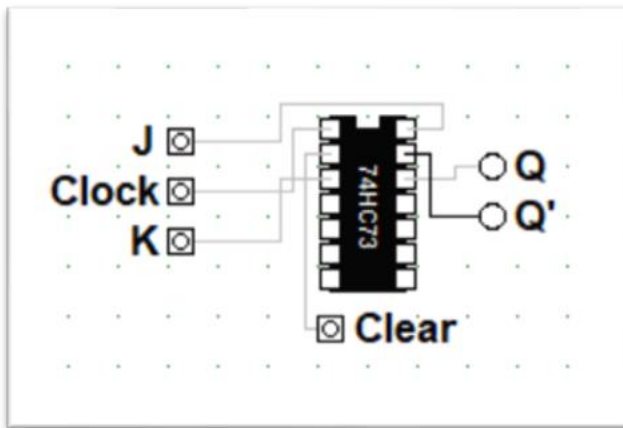
Pinout IC 74HC73:



Pinout IC 74HC74:



Flip - Flop JK menggunakan IC 74HC73:



Flip - Flop D menggunakan IC 74HC74:

