



LAPORAN PRAKTIKUM

IF310303

PRAKTIKUM SISTEM DIGITAL

MODUL: 9

SHIFT REGISTER

NAMA : Muhammad Alwiza Ansyar
NIM : M0520051
HARI : Jumat
TANGGAL : 11 Desember 2020
WAKTU : 10.15 – 11.05 WIB
ASISTEN : Akhtar Bariq Rahman

PROGRAM STUDI INFORMATIKA

UNIVERSITAS SEBELAS MARET

2020

Modul 9

SHIFT REGISTER

Muhammad Alwiza Ansyar (M0520051) / 11 Desember 2020

Email : alwiza21@student.uns.ac.id

Asisten : Akhtar Bariq Rahman

Abstraksi— Berikut merupakan laporan praktikum untuk modul 9 yang memiliki fokus bahasan tentang *shift register*. *Shift register* adalah salah satu rangkaian logika sekuensial yang berfungsi untuk menyimpan data dalam biner yang banyak serta mampu menggeser data tersebut. Pada laporan praktikum ini, akan dibahas tentang pengekspresian *shift register* SISO, SIPO, PISO, dan PIPO dalam rangkaian logika serta tabel kebenarannya.

Kata kunci— *shift register*, SISO, SIPO, PISO, PIPO

I. PENDAHULUAN

Pada sistem digital terdapat berbagai informasi. Informasi tersebut tidak selalu digunakan di saat itu juga. Maka dari itu, diperlukan alat untuk dapat menyimpan informasi. Salah satu alat tersebut adalah *register*.

Register adalah rangkaian logika sekuensial yang dapat menyimpan informasi dalam binary lebih dari 1 bit, berbeda dari Flip-flop yang hanya dapat menyimpan informasi sebanyak 1 bit saja. *Register* terdiri dari susunan Flip-flop JK atau D. Salah satu jenis *register* adalah *shift register*, yaitu *register* yang mampu menggeser data yang tersimpan di dalamnya. Fungsi pergeseran ini banyak diterapkan di berbagai alat-alat elektronik

II. DASAR TEORI

2.1 Register

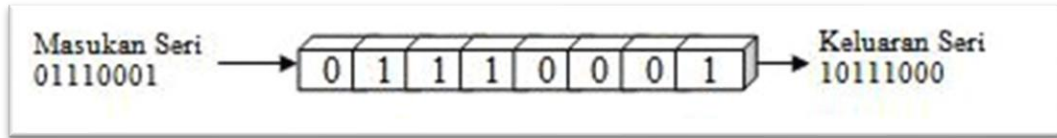
Register adalah rangkaian sekuensial yang dapat menyimpan data dalam bentuk *binary* dalam jumlah yang banyak. *Register* terdiri dari susunan Flip-flop JK atau Flip-flop D. Cara kerja *register* ialah menyusun secara khusus Flip-Flop tadi sehingga sifat Flip-flop yang hanya dapat menyimpan data 1 Bit dapat menyimpan data yang lebih banyak lagi. Jumlah bit yang dapat disimpan sama dengan jumlah Flip-flop yang digunakan.

2.2 Shift Register

Shift register adalah jenis dari *register* di mana ia dapat menyimpan dan *shifting*/menggeser data. *Shift* ini dapat dilakukan ke arah LSB (kanan) atau ke arah MSB (kiri). Penggeseran data dilakukan menggunakan *clock*. Setiap kali *pulse* dilakukan maka data akan bergeser. Fungsi geser yang dimiliki *shift register* dimanfaatkan untuk menyederhanakan rangkaian logika. *Register* ini juga dapat digunakan untuk mengubah format data dari seri ke paralel atau dari paralel ke seri.

2.3 Shift Register: Serial in Serial Out (SISO)

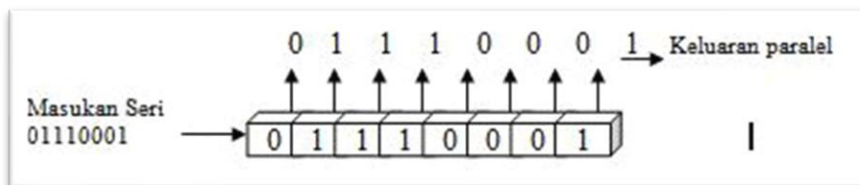
Rangkaian ini adalah jenis dari *Shift Register* yang mana *input* masuk secara seri dan *output* ditampilkan secara seri pula. *Output* yang *register* adalah *output* dari Flip-flop terakhir. Jenis SISO ini tidak mengubah format data



Dikarenakan *input* masuk secara seri, maka data masuk secara satu per satu dan *clock pulse* digunakan untuk mengijinkan data masuk sekaligus melakukan *shifting*. Contoh: jika ingin memasukkan data 1011 pada *Shift Register* SISO 4 bit, maka dilakukan *input* 1 → *pulse* → *input* 1 → *pulse* → *input* 0 → *pulse* → *input* 1 → *pulse*. *Clock* pada SISO bekerja secara *falling edge*.

2.4 Shift Register: Serial in Paralel Out (SIPO)

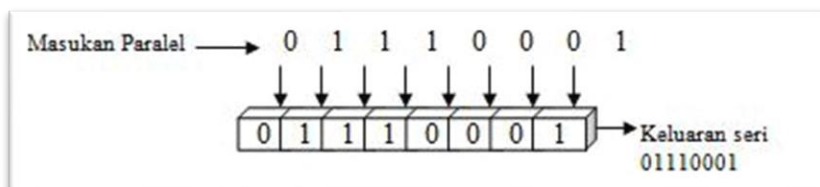
Rangkaian ini adalah jenis dari *Shift Register* yang mana *input* masuk secara seri dan *output* ditampilkan secara paralel. Maka dari itu, setiap Flip-flop ditampilkan *outputnya*. Jenis SIPO ini mengubah format data dari seri ke paralel.



Sama seperti jenis SISO, dikarenakan *input* masuk secara seri, maka data masuk secara satu per satu dan *clock pulse* digunakan untuk mengijinkan data masuk sekaligus melakukan *shifting*. *Clock* pada SIPO bekerja secara *falling edge*.

2.5 Shift Register: Paralel in Serial Out (PISO)

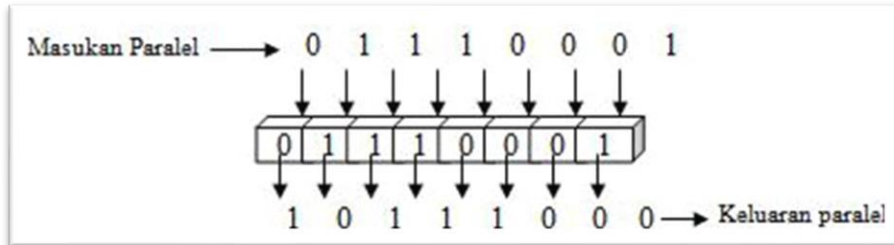
Rangkaian ini adalah jenis dari *Shift Register* yang mana *input* masuk secara paralel dan *output* ditampilkan secara seri. Maka dari itu, seluruh data dapat secara bersamaan dan dimasukkan dalam *register*. Jenis SIPO ini mengubah format data dari paralel ke seri.



Register ini tetap dapat melakukan *shift* menggunakan *clock* tetapi *input* yang baru akibat pergeseran adalah 0 secara konstan. Contoh: saat keadaan *register* 1011 lalu dilakukan *shift*, maka hasilnya ialah 0101. *Clock* pada PISO bekerja secara *leading edge*.

2.6 Shift Register: Paralel in Paralel Out (PIPO)

Rangkaian ini adalah jenis dari *Shift Register* yang mana *input* masuk secara paralel dan *output* ditampilkan secara paralel pula. Maka dari itu, seluruh data dapat secara bersamaan dan dimasukkan dalam *register*. Jenis PIPO ini tidak mengubah format data.



Sama seperti PISO, *register* ini tetap dapat melakukan *shift* menggunakan *clock* tetapi *input* yang baru akibat pergeseran adalah 0 secara konstan. *Clock* pada PIPO bekerja secara *leading edge*.

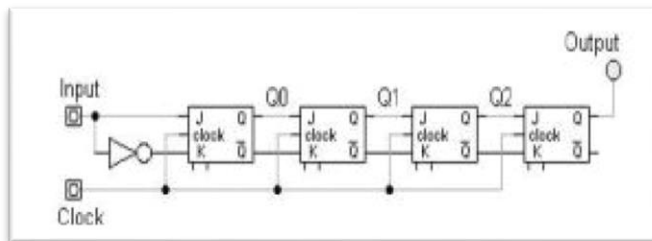
III. ALAT DAN LANGKAH PERCOBAAN

3.1 Alat

1. PC/Laptop
2. Aplikasi Digital Works

3.2 Langkah Percobaan

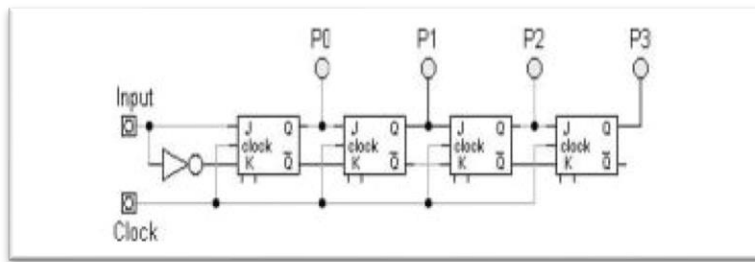
1. Buka aplikasi Digital Works
2. Buat rangkaian berikut



3. Amati dan catat *output* terhadap kombinasi keadaan *input*
4. Lengkapi tabel kebenaran berikut

Clock ke	Input	Q0	Q1	Q2	Q3	Output	SISO
0	1						
1	1						
2	1						
3	1						
4	1						
0	0						
1	0						
2	0						
3	0						
4	0						

5. Buat rangkaian berikut

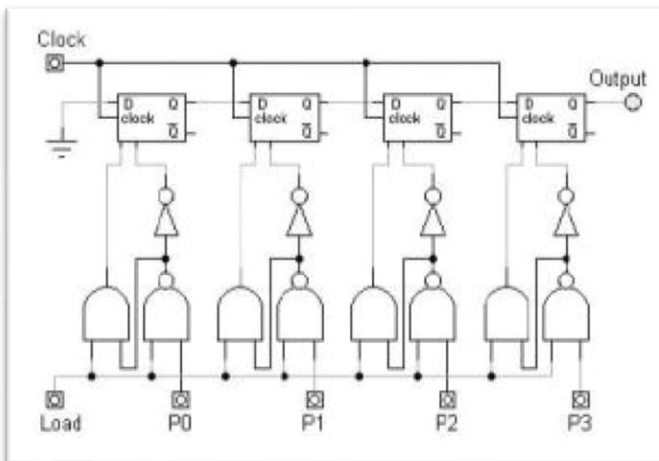


6. Amati dan catat *output* terhadap kombinasi keadaan *input*

7. Lengkapi tabel kebenaran berikut

Clock ke	Input	Q1	Q2	Q3	Q4	SIPO
0	1					
1	1					
2	1					
3	1					
4	1					
0	0					
1	0					
2	0					
3	0					
4	0					

8. Buat rangkaian berikut

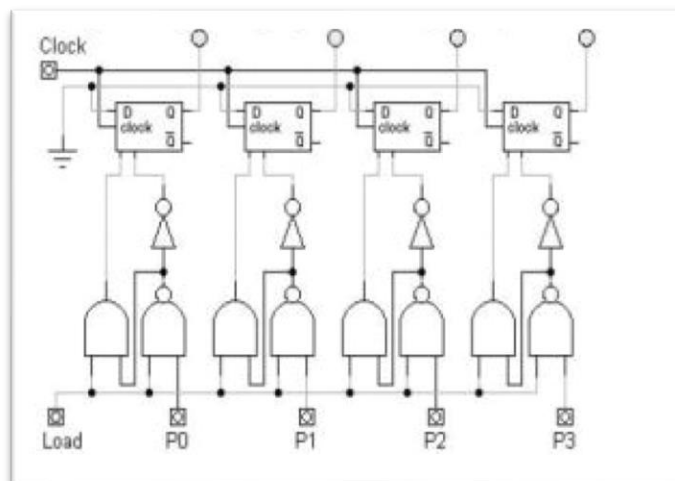


9. Amati dan catat *output* terhadap kombinasi keadaan *input*

10. Lengkapi tabel kebenaran berikut

Clock	Load	P0	P1	P2	P3	Output	PISO
1	0	x	x	x	x		
1	1	0	0	0	1		
1	1	0	0	1	0		
1	1	0	0	1	1		
1	1	0	1	0	0		
1	1	0	1	0	1		
1	1	0	1	1	0		
1	1	0	1	1	1		
1	1	1	0	0	0		
1	1	1	0	0	1		
1	1	1	0	1	0		
1	1	1	0	1	1		
1	1	1	1	0	0		
1	1	1	1	0	1		
1	1	1	1	1	0		
1	1	1	1	1	1		

11. Buat rangkaian berikut

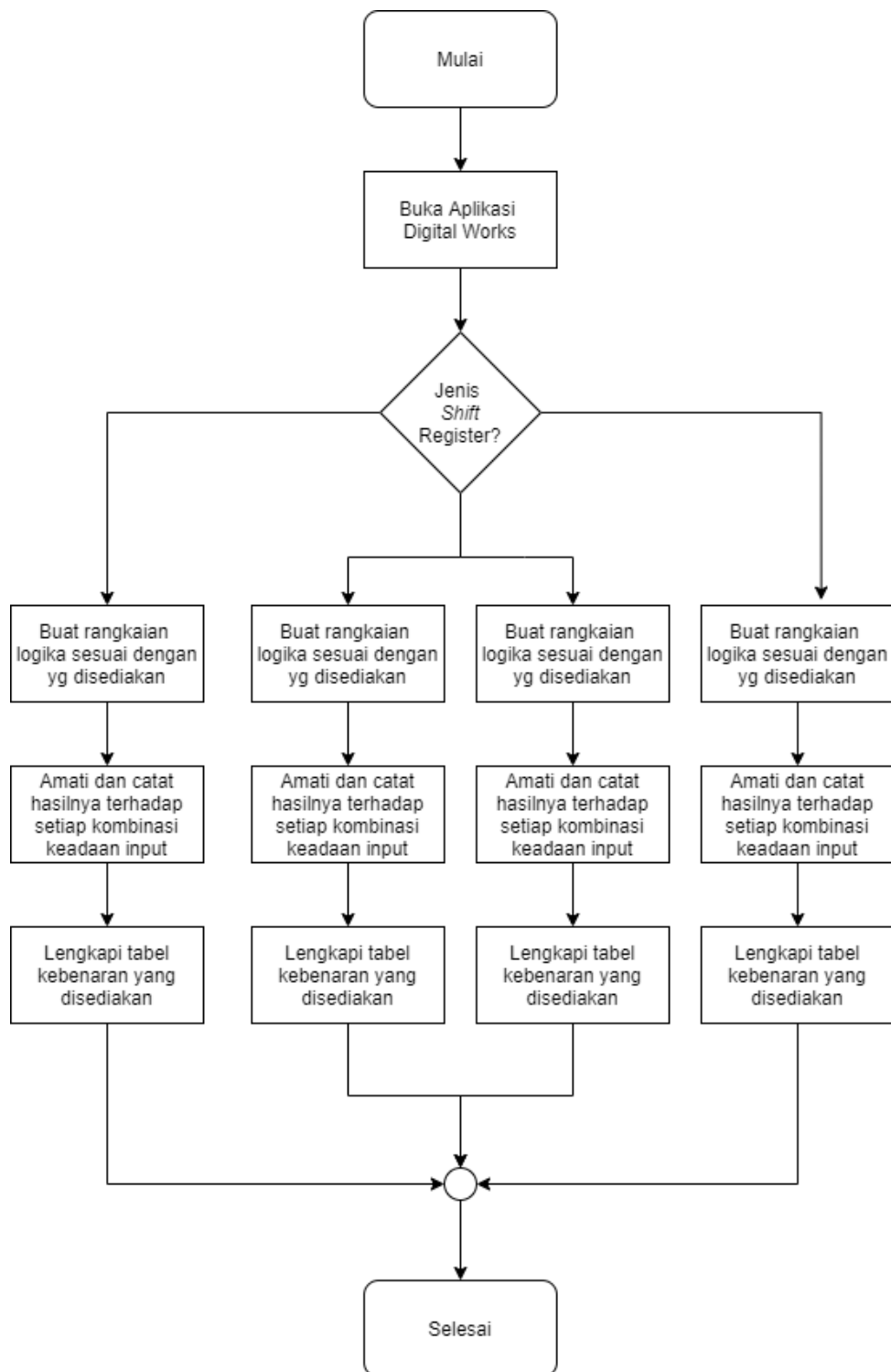


12. Amati dan catat *output* terhadap kombinasi keadaan *input*

13. Lengkapi tabel kebenaran berikut

Clock	Load	P0	P1	P2	P3	O1	O2	O3	O4	PIPO
1	0	x	x	x	x					
1	1	0	0	0	1					
1	1	0	0	1	0					
1	1	0	0	1	1					
1	1	0	1	0	0					
1	1	0	1	0	1					
1	1	0	1	1	0					
1	1	0	1	1	1					
1	1	1	0	0	0					
1	1	1	0	0	1					
1	1	1	0	1	0					
1	1	1	0	1	1					
1	1	1	1	0	0					
1	1	1	1	0	1					
1	1	1	1	1	0					
1	1	1	1	1	1					

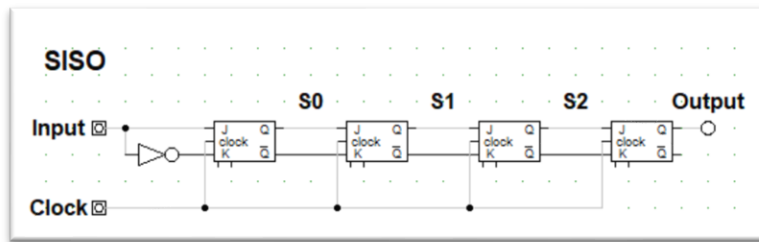
Diagram alur:



IV. HASIL DAN ANALISIS PERCOBAAN

4.1 Shift Register SISO

Rangkaian logika:



Tabel kebenaran:

Pulse	Input	S0	S1	S2	Output
0	#1	Initial	Initial	Initial	Initial
1	#2	#1	Initial	Initial	Initial
2	#3	#2	#1	Initial	Initial
3	#4	#3	#2	#1	Initial
4	#5	#4	#3	#2	#1
5	#6	#5	#4	#3	#2

Contoh: memasukkan 1001

Pulse	Input	S0	S1	S2	Output
0	1	Initial	Initial	Initial	Initial
1	0	1	Initial	Initial	Initial
2	0	0	1	Initial	Initial
3	1	0	0	1	Initial
4	X	1	0	0	1

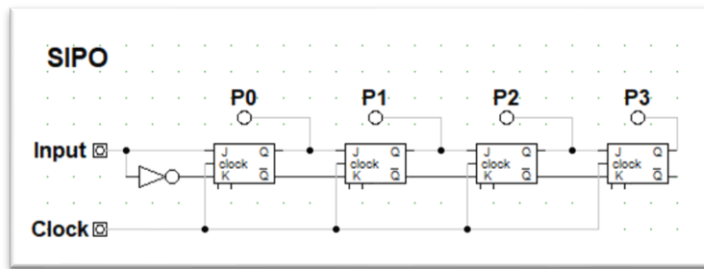
- Pada SISO, *output* hanya menampilkan keadaan S3 menggunakan lampu LED (seri). Untuk S0, S1, dan S2 pada Digital Works dapat dilihat dari keadaan kabel yang menghubungkannya. Apabila putih berarti berlogika 0 dan apabila hitam berarti berlogika 1.
- *Input* masuk ke *register* apabila dilakukan *pulse*. *Input* masuk satu per satu (seri)
- *Initial* merujuk pada keadaan *register* saat itu juga. Pada keadaan *default*, *register* berada pada keadaan 0000.

Melengkapi tabel kebenaran:

Clock ke	Input	Q0	Q1	Q2	Output
0	1	0	0	0	0
1	1	1	0	0	0
2	1	1	1	0	0
3	1	1	1	1	0
4	1	1	1	1	1
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

4.2 Shift Register SIPO

Rangkaian logika:



Tabel kebenaran:

Pulse	Input	P0	P1	P2	P3
0	#1	Initial	Initial	Initial	Initial
1	#2	#1	Initial	Initial	Initial
2	#3	#2	#1	Initial	Initial
3	#4	#3	#2	#1	Initial
4	#5	#4	#3	#2	#1
5	#6	#5	#4	#3	#2

Contoh: memasukkan 1001

Pulse	Input	P0	P1	P2	P3
0	1	Initial	Initial	Initial	Initial
1	0	1	Initial	Initial	Initial
2	0	0	1	Initial	Initial
3	1	0	0	1	Initial
4	X	1	0	0	1

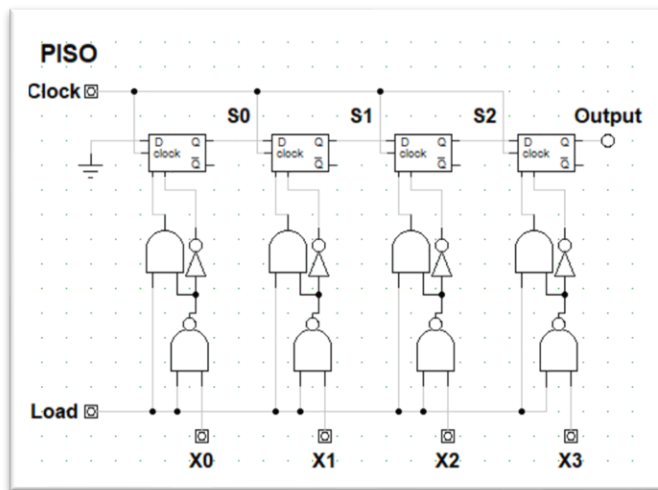
- Pada SIPO, setiap *output* yaitu P0, P1, P2, dan P3 ditampilkan menggunakan lampu LED (paralel)
- *Input* masuk ke *register* apabila dilakukan *pulse*. *Input* masuk satu per satu (seri)
- *Initial* merujuk pada keadaan *register* saat itu juga. Pada keadaan *default*, *register* berada pada keadaan 0000.

Melengkapi tabel kebenaran:

Clock ke	Input	Q1	Q2	Q3	Q4
0	1	0	0	0	0
1	1	1	0	0	0
2	1	1	1	0	0
3	1	1	1	1	0
4	1	1	1	1	1
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

4.3 Shift Register PISO dengan Load

Rangkaian logika:



Tabel kebenaran:

Pulse	Load	X0	X1	X2	X3	S0	S1	S2	Output
0	0	X	X	X	X	Initial	Initial	Initial	Initial
0	1	#1	#2	#3	#4	#1	#2	#3	#4
1	0	X	X	X	X	0	#1	#2	#3
2	0	X	X	X	X	0	0	#1	#2
3	0	X	X	X	X	0	0	0	#1
4	0	X	X	X	X	0	0	0	0
5	0	X	X	X	X	0	0	0	0

Contoh: memasukkan 1001 lalu menggeser sebanyak 2x

Pulse	Load	X0	X1	X2	X3	S0	S1	S2	Output
0	0	X	X	X	X	Initial	Initial	Initial	Initial
0	1	1	0	0	1	1	0	0	1
1	0	X	X	X	X	0	1	0	0
2	0	X	X	X	X	0	0	1	0

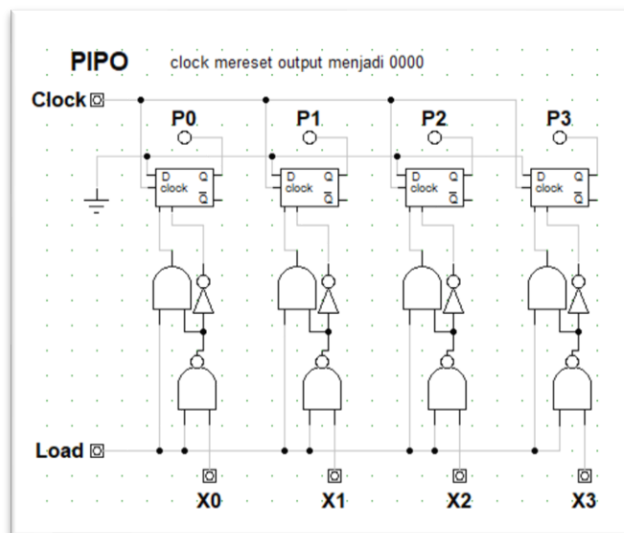
- Pada PISO, *output* hanya menampilkan keadaan S3 menggunakan lampu LED (seri). Untuk S0, S1, dan S2 pada Digital Works dapat dilihat dari keadaan kabel yang menghubungkannya. Apabila putih berarti berlogika 0 dan apabila hitam berarti berlogika 1.
- Terdapat *input Load* yang berfungsi supaya *input* masuk ke *register* secara serentak. Saat *Load* berlogika 0, *input-input* ditahan untuk masuk ke *register*. Saat *Load* berlogika 1, *input-input* dapat masuk ke *register*
- *Clock* hanya bisa digunakan saat *Load* berlogika 0
- Setiap *input* masuk ke masing2 bagiannya (paralel)
- *Initial* merujuk pada keadaan *register* saat itu juga. Pada keadaan *default*, *register* berada pada keadaan 0000.

Melengkapi tabel kebenaran:

<i>Clock</i>	<i>Load</i>	P0	P1	P2	P3	<i>Output</i>
1	0	X	X	X	X	0
1	1	0	0	0	1	1
1	1	0	0	1	0	0
1	1	0	0	1	1	1
1	1	0	1	0	0	0
1	1	0	1	0	1	1
1	1	0	1	1	0	0
1	1	0	1	1	1	1
1	1	1	0	0	0	0
1	1	1	0	0	1	1
1	1	1	0	1	0	0
1	1	1	0	1	1	1
1	1	1	1	0	0	0
1	1	1	1	0	1	1
1	1	1	1	1	0	0
1	1	1	1	1	1	1

4.4 Shift Register PIPO dengan Load

Rangkaian logika:



Tabel kebenaran:

<i>Clock</i>	<i>Load</i>	X0	X1	X2	X3	P0	P1	P2	P3
0	0	X	X	X	X	<i>Initial</i>	<i>Initial</i>	<i>Initial</i>	<i>Initial</i>
0	1	#1	#2	#3	#4	#1	#2	#3	#4
1	0	X	X	X	X	0	0	0	0

Contoh: memasukkan 1001 lalu melakukan Reset

<i>Clock</i>	<i>Load</i>	X0	X1	X2	X3	P0	P1	P2	P3
0	0	X	X	X	X	<i>Initial</i>	<i>Initial</i>	<i>Initial</i>	<i>Initial</i>
0	1	1	0	0	1	1	0	0	1
1	0	X	X	X	X	0	0	0	0

- Pada PIPO, setiap *output* yaitu P0, P1, P2, dan P3 ditampilkan menggunakan lampu LED (paralel)
- Terdapat *input Load* yang berfungsi supaya *input* masuk ke *register* secara serentak. Saat *Load* berlogika 0, *input-input* ditahan untuk masuk ke *register*. Saat *Load* berlogika 1, *input-input* dapat masuk ke *register*
- *Clock* pada rangkaian ini berfungsi sebagai Reset, yaitu mengubah semua *output* menjadi 0
- *Clock* hanya bisa digunakan saat *Load* berlogika 0
- Setiap *input* masuk ke masing2 bagiannya (paralel)
- *Initial* merujuk pada keadaan *register* saat itu juga. Pada keadaan *default*, *register* berada pada keadaan 0000.

Melengkapi tabel kebenaran:

<i>Clock</i>	<i>Load</i>	P0	P1	P2	P3	O1	O2	O3	O4
1	0	X	X	X	X	X	X	X	X
1	1	0	0	0	1	0	0	0	1
1	1	0	0	1	0	0	0	1	0
1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	0	0	1	0	0
1	1	0	1	0	1	0	1	0	1
1	1	0	1	1	0	0	1	1	0
1	1	0	1	1	1	0	1	1	1
1	1	1	0	0	0	1	0	0	0
1	1	1	0	0	1	1	0	0	1
1	1	1	0	1	0	1	0	1	0
1	1	1	0	1	1	1	0	1	1
1	1	1	1	0	0	1	1	0	0
1	1	1	1	0	1	1	1	0	1
1	1	1	1	1	0	1	1	1	0
1	1	1	1	1	1	1	1	1	1

V.KESIMPULAN

Register adalah rangkaian sekuensial yang dapat menyimpan data dalam biner lebih dari 1 bit. *Register* terdiri dari berbagai Flip-flop JK atau D dengan jumlah Flip-flop sama dengan jumlah bit yang bisa disimpan. *Shift register* adalah *register* yang dapat menyimpan data dan mampu melakukan *shifting*/penggeseran. Penggeseran adalah menggeser data ke arah LSB atau MSB. Fungsi geser ini dimanfaatkan untuk menyederhanakan rangkaian logika. Penggeseran dilakukan dengan melakukan *pulse* pada *clock*.

Shift Register memiliki 4 jenis, yaitu Serial In Serial Out (SISO), Serial In Paralel Out (SIPO), Paralel In Serial Out (PISO), dan Paralel In Paralel Out (PIPO). Keempat jenis tersebut berhubungan dengan cara *shift register* menerima *input* dan menampilkan *output*. SISO menerima *input* secara seri yaitu satu per satu dan menampilkan *output* secara seri juga yaitu hanya menampilkan *output* dari Flip-flop terakhir. SIPO menerima *input* secara seri yaitu satu per satu dan menampilkan *output* secara paralel yaitu menampilkan setiap *output* dari Flip-flop. SIPO mengubah format data dari seri ke paralel. PISO menerima *input* secara paralel yaitu semua *input* masuk ke *register* di satu waktu dan menampilkan

output secara seri juga yaitu hanya menampilkan *output* dari Flip-flop terakhir. PISO mengubah format data dari paralel ke seri. PIPO menerima *input* secara paralel yaitu semua *input* masuk ke *register* di satu waktu dan menampilkan *output* secara paralel yaitu menampilkan setiap *output* dari Flip-flop.

Pada praktikum ini, dibuat keempat jenis *shift register* di atas pada aplikasi Digital Works lalu dilakukan analisis. Setelah itu, hasil dari analisis disajikan dalam tabel kebenaran di laporan praktikum ini. Lalu ditambahkan pula contoh kasus pada masing-masing jenis *shift register* beserta tabel kebenarannya sebagai penjelas. Lalu, tabel kebenaran dari video pembahasan modul 9 yang dijadikan soal tersebut ditulis dan dilengkapi (jawaban ada pada bagian tabel yang memiliki *shading* hijau muda).

Saat melakukan *shifting*, data yang baru akan masuk dan data yang terakhir akan “keluar” dari Flip-flop sehingga bisa dikatakan data tersebut dihapus. Untuk jenis SISO dan SIPO, data yang dimasukkan saat *shifting* dapat diatur, yaitu 1 atau 0. Untuk jenis PISO dan PIPO, data yang dimasukkan saat *shifting* adalah 0.

VI. DAFTAR PUSTAKA

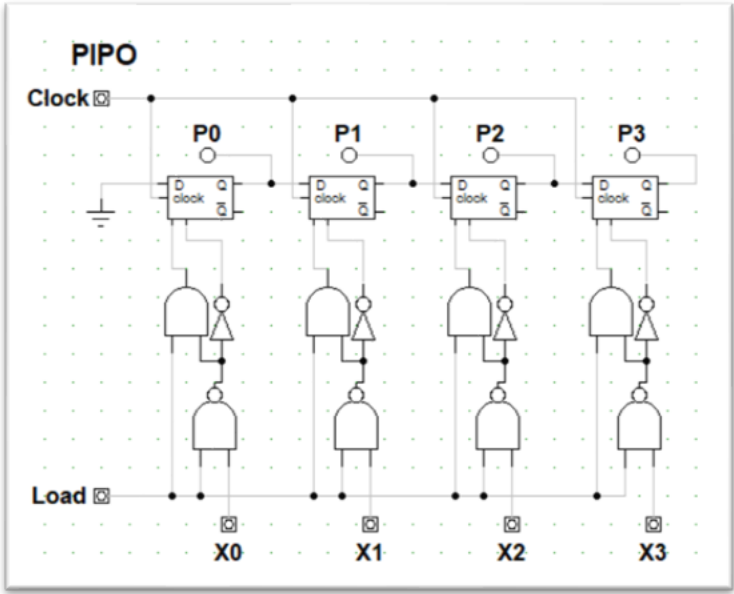
Anonim. 2020. *Shift Register (Register Geser)*, diakses dari <https://elektronika-dasar.web.id/shift-register-register-geser/>, pada 16 Desember 2020.

Rustamaji, Tras. 2016. *Shift Register Buat Apa Sih?*, diakses dari <http://www.rustamaji.net/id/arduino/shift-register-buat-apa-sih>, pada 16 Desember 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

Shift Register PIPO dengan clock untuk melakukan shifting:



Tabel kebenaran:

Pulse	Load	X0	X1	X2	X3	P0	P1	P2	P3
0	0	X	X	X	X	Initial	Initial	Initial	Initial
0	1	#1	#2	#3	#4	#1	#2	#3	#4
1	0	X	X	X	X	0	#1	#2	#3
2	0	X	X	X	X	0	0	#1	#2
3	0	X	X	X	X	0	0	0	#1
4	0	X	X	X	X	0	0	0	0
5	0	X	X	X	X	0	0	0	0

Contoh: Memasukkan 1001 lalu melakukan shift 2x

Pulse	Load	X0	X1	X2	X3	P0	P1	P2	P3
0	0	X	X	X	X	Initial	Initial	Initial	Initial
0	1	1	0	0	1	1	0	0	1
1	0	X	X	X	X	0	1	0	0
2	0	X	X	X	X	0	0	1	0