

## **LAB #4**

# **RANGKAIAN LOGIKA SEKUENSIAL**

### **TUJUAN**

1. Untuk mempelajari bagaimana dasar rangkaian logika sekuensial bekerja
2. Untuk menguji dan menyelidiki pengoperasian berbagai *Latch* dan sirkuit *Flip-Flop*

### **PENDAHULUAN**

Rangkaian sekuensial adalah sirkuit dengan *output* yang diperoleh adalah dari fungsi *state* masukan serta *state* keluaran sebelumnya, disebut juga sebagai rangkaian sirkuit sekuensial dengan elemen memori. Disebut juga elemen memori *Flip-Flop*.

*Latch* dan *Flip-Flop* dapat menyimpan 1 bit informasi biner dan memiliki 2 terminal *output*. Jika salah satu terminal *output* adalah "1", *output* lain "0" dan menjaga kedua statusnya berlawanan. Rangkaian *Latch* tidak menggunakan pulsa *clock* sehingga setiap kali *output* berubah, status *output* berubah. Namun *Flip-Flop* mengubah status keluaran hanya ketika pulsa *clock* lainnya dimasukkan.

*Latch* adalah alat berguna yang berisi umpan balik dan memungkinkan penyimpanan memori. Mereka digunakan dalam aplikasi seperti sirkuit penyaringan *noise* sederhana dan *Flip-Flop*. *Latch* pertama yang telah dibahas di kelas adalah latch SR / S'R' yang memungkinkan kita untuk *Set* atau *Reset output*. Kelemahan *Latch* adalah berisi sebuah transisi yang menyebabkan *state metastable* (tak tentu). Sebuah *clock* / sinyal *enable* ditambahkan untuk mencegah transisi ketika sinyal *clock* rendah. Terakhir, D-*Latch* sama sekali digantikan dengan mencegah S dan R dari mengubah pada saat yang sama. Oleh karena itu, dalam diskusi ini, *Latch* telah menjadi semakin stabil.

### **KEBUTUHAN**

1. Koper lengkap HBE-LogicCircuit-Digital
2. Kabel tembaga

### **TUGAS PENDAHULUAN**

1. Baca dan pahami *Technical Guide* terlebih dahulu!
2. Pelajari semua *datasheet* dari semua IC yang digunakan pada percobaan ini!
3. Apa itu *Clock*?
4. Jelaskan perbedaan antara *Latch* dengan *clock* dan tanpa *clock*!
5. Jelaskan cara kerja D *Latch* dan sirkuit diagramnya!
6. Apa yang akan terjadi jika kedua *input Latch* dan *Flip-Flop* bernilai "1"?
7. Bagaimana cara sebuah rangkaian sekuensial menyimpan memori?

### **EKPERIMEN 1 : LATCH**

#### **[Percobaan 1] SR Latch Sederhana Tanpa Clock**

[Persiapan]

|           |   |
|-----------|---|
| Alat I/O  | Slide Switch (SW1, SW2), LED (D1, D2)             |
| Modul     | Modul NAND  |
| Lain-lain | Kabel (untuk menghubungkan alat I/O dengan modul) |

[Prosedur]

1. Bangun sebuah rangkaian SR Latch sederhana seperti pada Fig. IV-1.

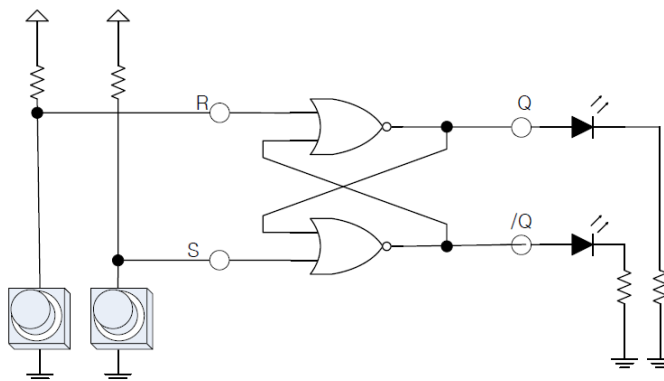


Fig. IV-1. Diagram Operasi SR Latch Sederhana

2. Catat hasilnya pada Tabel IV-1.

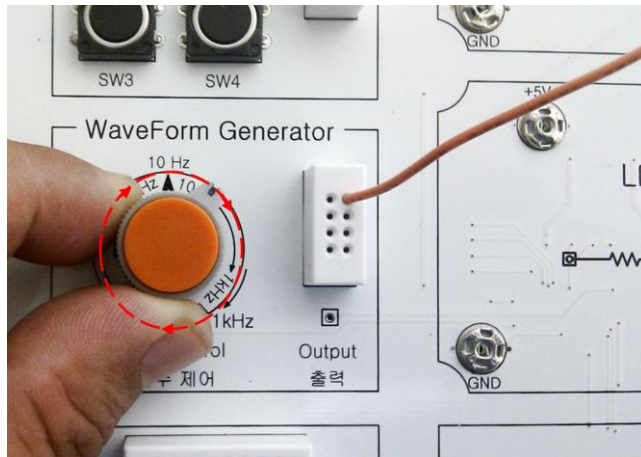
| Tabel IV-1. Hasil Operasi SR Latch Sederhana |         |        |         |
|--|---------|--------|---------|
| Input  |         | Output |         |
| R (SW1)                                      | S (SW2) | Q (D1) | Q' (D2) |
| 0  | 0       |        |         |
| 0  | 1       |        |         |
| 1  | 0       |        |         |
| 1  | 1       |        |         |

3. Simulasikan menggunakan DAQ. Lalu simpan waveform-nya dan lampirkan dalam laporan.

## **EKSPERIMEN 2 : FLIP-FLOP**

### **[Percobaan 2] RS Flip-Flop**

**Catatan :** . SW4 dan clock (clk) dihubungkan dengan Wave Form Generator. Atur frekwensinya supaya keluaran rangkaian bisa anda amati.



[Persiapan]

|           |   |
|-----------|---|
| Alat I/O  | Slide Switch (SW1, SW2, SW4), LED (D1, D2)        |
| Modul     | Modul AND, Modul NOR                              |
| Lain-lain | Kabel (untuk menghubungkan alat I/O dengan modul) |

[Prosedur]

- Gunakan semua modul untuk merangkai RS *Flip-Flop* seperti pada Fig. IV-3

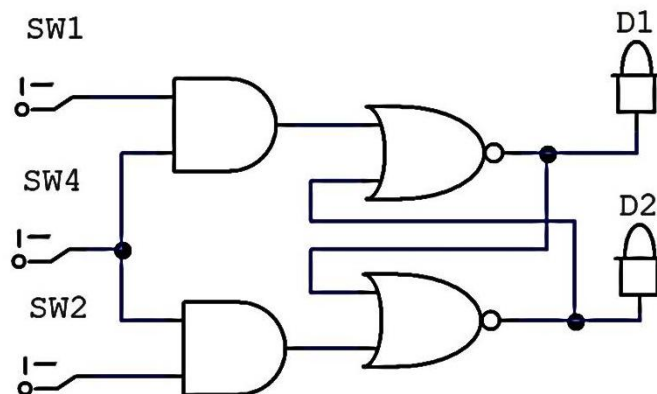


Fig. IV-2. Diagram Operasi RS Flip-Flop

- Catat hasilnya pada Tabel IV-2.

| Tabel IV-2. Hasil Operasi RS Flip-Flop |         |             |        |         |
|--|---------|-------------|--------|---------|
| Input                                  |         |             | Output |         |
| R (SW1)                                | S (SW2) | Clock (SW4) | Q (D1) | Q' (D2) |
| 0                                      | 0       | -           |        |         |
| 0                                      | 0       | ^           |        |         |
| 0                                      | 1       | -           |        |         |
| 0                                      | 1       | ^           |        |         |
| 1                                      | 0       | -           |        |         |
| 1                                      | 0       | ^           |        |         |
| 1                                      | 1       | -           |        |         |
| 1                                      | 1       | ^           |        |         |

3. Simulasikan menggunakan DAQ. Lalu simpan *waveform*-nya dan lampirkan dalam laporan.

### [Percobaan 3] JK Flip-Flop

[Persiapan]

|           |   |
|-----------|---|
| Alat I/O  | Slide Switch (SW1, SW2, SW4), LED (D3, D4)        |
| Modul     | Modul AND, Modul NOR                              |
| Lain-lain | Kabel (untuk menghubungkan alat I/O dengan modul) |

[Prosedur]

1. Gunakan semua modul gate dasar dan modul JK flip flop untuk merangkai JK *Flip-Flop* seperti pada Fig. IV-3. Hubungkan clock(clk) dengan wave from generator.

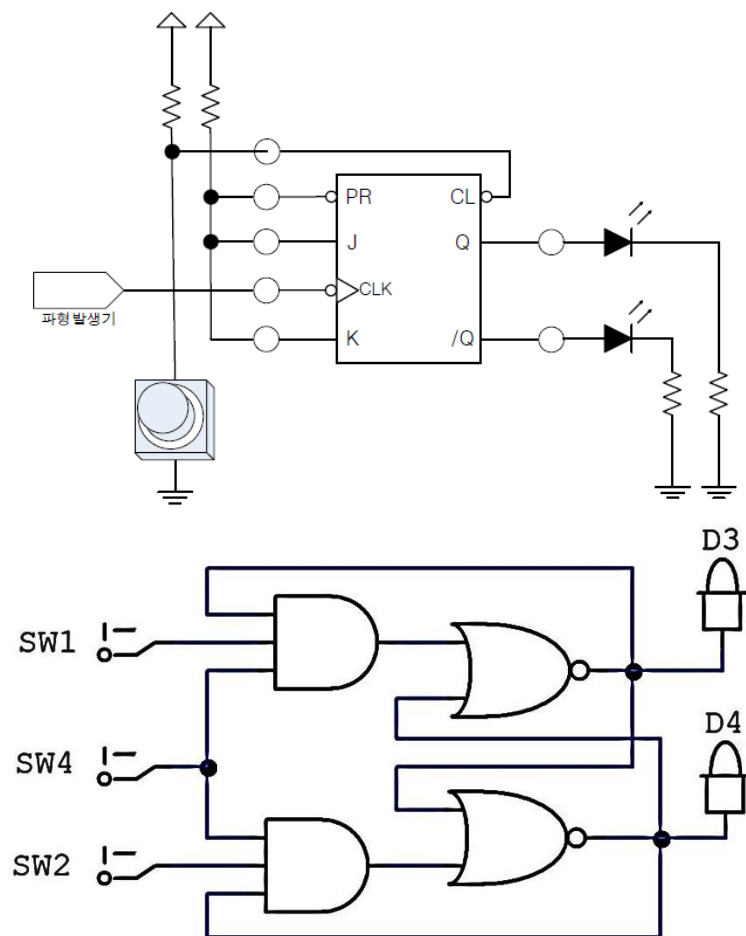


Fig. IV-3. Diagram Operasi JK Flip-Flop

2. Catat hasilnya pada Tabel IV-3.

| Tabel IV-3. Hasil Operasi JK Flip-Flop |         |             |        |         |
|--|---------|-------------|--------|---------|
| Input                                  |         |             | Output |         |
| K (SW1)                                | J (SW2) | Clock (SW4) | Q (D1) | Q' (D2) |
| 0                                      | 0       | -           |        |         |
| 0                                      | 0       | ^           |        |         |

|   |   |   |  |  |
|---|---|---|--|--|
| 0 | 1 | - |  |  |
| 0 | 1 | ^ |  |  |
| 1 | 0 | - |  |  |
| 1 | 0 | ^ |  |  |
| 1 | 1 | - |  |  |
| 1 | 1 | ^ |  |  |

3. Simulasikan menggunakan DAQ. Lalu simpan *waveform*-nya dan lampirkan dalam laporan.

#### [Percobaan 4] D Flip-Flop

[Persiapan]

|           |   |
|-----------|---|
| Alat I/O  | Slide Switch (SW2, SW4), LED (D3, D4)             |
| Modul     | Modul AND, Modul NOR, Modul NOT                   |
| Lain-lain | Kabel (untuk menghubungkan alat I/O dengan modul) |

[Prosedur]

1. Gunakan semua modul gate dasar atau modul JK flip flop untuk merangkai D *Flip-Flop* seperti pada Fig. IV-4.

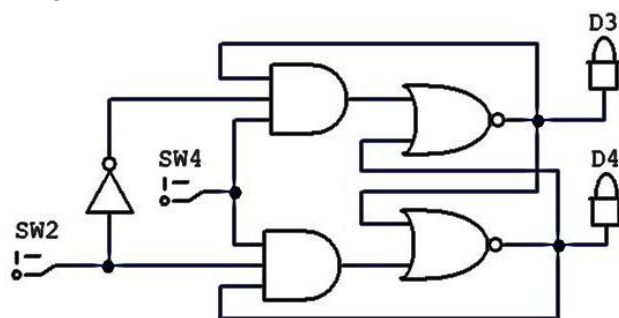


Fig. IV-4. Diagram Operasi D Flip-Flop

2. Catat hasilnya pada Tabel IV-4.

| Tabel IV-4. Hasil Operasi D Flip-Flop |             |        |        |
|---------------------------------------|-------------|--------|--------|
| Input                                 |             | Output |        |
| D (SW1)                               | Clock (SW3) | Q (D1) | Q'(D2) |
| 0                                     | -           |        |        |
| 0                                     | ^           |        |        |
| 1                                     | -           |        |        |
| 1                                     | ^           |        |        |

3. Simulasikan menggunakan DAQ. Lalu simpan *waveform*-nya dan lampirkan dalam laporan.

#### [Percobaan 5] T Flip-Flop

[Persiapan]

|           |   |
|-----------|---|
| Alat I/O  | Slide Switch (SW2, SW4), LED (D3, D4)             |
| Modul     | Modul AND, Modul NOR                              |
| Lain-lain | Kabel (untuk menghubungkan alat I/O dengan modul) |

[Prosedur]

- Gunakan semua modul gate dasar atau modul JK flip flop untuk merangkai T *Flip-Flop* seperti pada Fig. IV-5.

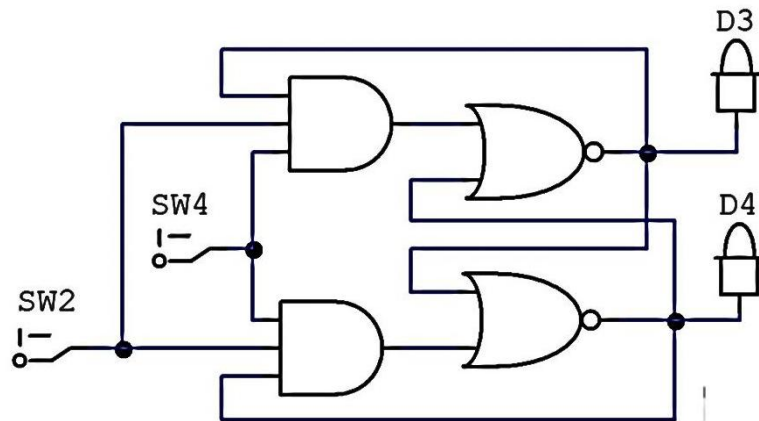


Fig. IV-5. Diagram Operasi T Flip-Flop

- Catat hasilnya pada Tabel IV-5.

| Tabel IV-5. Hasil Operasi T Flip-Flop |             |        |         |
|---------------------------------------|-------------|--------|---------|
| Input                                 |             | Output |         |
| S <sub>1</sub> (SW1)                  | Clock (SW3) | Q (D1) | Q' (D2) |
| 0                                     | -           |        |         |
| 0                                     | ^           |        |         |
| 1                                     | -           |        |         |
| 1                                     | ^           |        |         |

- Simulasikan menggunakan DAQ. Lalu simpan *waveform*-nya dan lampirkan dalam laporan.

## TUGAS MODUL

- Simulasikan semua percobaan pada Circuit Maker!
- Apa itu *Edge-Triggered Flip-Flop*!
- Apa itu *Master-Slave Flip-Flop*, jelaskan keuntungan *Master-Slave Flip-Flop*!
- Buat rangkaian *Master-Slave SR Flip-flop* dan *JK Flip-Flop*! Jelaskan cara kerjanya! Simulasikan!
- Dari rangkaian yang kalian buat pada nomor 4, simpulkan penggunaan *Master-Slave Flip-Flop* dibandingkan rangkaian *Flip-Flop* biasa!