

# ***Panduan Praktikum*** ***Teknik Digital***

***Untuk Mahasiswa Teknik Elektro  
dan Program Studi Serumpun***

**Dr. Muchlas, M.T.**

**Universitas Ahmad Dahlan  
Yogyakarta, 2022**

Panduan Praktikum

# **Teknik Digital**

*Menggunakan Simulator Breadboard*

Dr. Muchlas, M.T.

Universitas Ahmad Dahlan  
Yogyakarta, 2022

Panduan Praktikum Teknik Digital  
Menggunakan Simulator *Breadboard*

Penulis: Dr. Muchlas, M.T.

Hak cipta milik penulis. Buku ini dapat diperoleh secara bebas tanpa dipungut biaya. Dilarang memperbanyak karya ini dalam bentuk apapun untuk kepentingan komersial, kecuali ditujukan bagi pengembangan ilmu dan pendidikan yang bersifat non-komersial dengan menyebutkan identitas buku sebagai sumbernya.

Universitas Ahmad Dahlan  
Alamat: Jalan Kapas 9, Semaki, Yogyakarta  
Telepon: 0274-563515

Kontak Penulis: [muchlas.te@uad.ac.id](mailto:muchlas.te@uad.ac.id)  
Website Penulis: <http://muchlas.ee.uad.ac.id>

## PENGANTAR

Materi-materi praktik yang ada pada buku panduan ini disesuaikan dengan materi-materi kuliah Teknik Digital pada Program Studi Teknik Elektro mencakup Watak Gerbang Logika Dasar, Minimalisasi Rangkaian Logika, Komparator dan Penjumlah Biner, Multiplexer dan Demultiplexer, Enkoder dan Dekoder, Flip-flop, Pencacah, dan Register.

Panduan ini tidak memberikan uraian teori lengkap, melainkan hanya sebagian saja berupa suatu deskripsi topik praktik. Hal ini dilakukan agar sebelum suatu praktik dilaksanakan, para peserta terdorong untuk belajar secara mandiri tentang teori yang digunakan. Tanpa persiapan pemahaman yang baik terhadap teori pendukungnya, praktik ini tidak dapat dilaksanakan dengan baik dan benar. Metode pelaksanaan praktik yang digunakan adalah inkuiri terbimbing, dengan demikian mahasiswa diberi kebebasan untuk menentukan sendiri prosedur percobaan yang dijalannya. Mahasiswa peserta praktik sangat disarankan untuk berdiskusi secara intensif dengan instruktur pendamping sebelum, selama dan sesudah pelaksanaan praktik. Sebelum praktik dilaksanakan, mahasiswa wajib berlatih dan memastikan diri dapat mengoperasikan dengan baik simulator *breadboard* sebagai perangkat *virtual lab* yang akan digunakan.

Kepada semua pihak yang telah membantu penyusunan panduan ini dan membantu uji coba peralatan praktik, diucapkan terima kasih. Penghargaan yang setinggi-tingginya diberikan kepada Dr. Chris Bailey, Dr. Michael Freeman, Nicholas Glass beserta tim pengembang simulator dari Departemen Ilmu Komputer, *the University of York*, Inggris atas dedikasinya menyediakan simulator *breadboard* yang dapat diakses secara bebas oleh masyarakat.

Dengan berbagai kekurangannya, panduan ini diharapkan dapat memberikan manfaat sesuai dengan fungsinya. Masukan-masukan dari siapapun akan sangat dinanti demi perbaikan panduan praktik ini.

Yogyakarta, Februari 2022

Muchlas

## DAFTAR ISI

	<b>Halaman</b>
HALAMAN JUDUL	i
PERNYATAAN HAK CIPTA	ii
KATA PENGANTAR	iii
DAFTAR ISI	iv
DAFTAR <i>LINK</i> UNTUK <i>DOWNLOAD</i> SIMULATOR <i>BREADBOARD</i>	v
PERCOBAAN I WATAK GERBANG LOGIKA DASAR	1
PERCOBAAN II LOGIKA KOMBINASI	8
PERCOBAAN III KOMPARATOR DAN PENJUMLAH BINER	14
PERCOBAAN IV MULTIPLIKESER	23
PERCOBAAN V DEMULTIPLIKESER	28
PERCOBAAN VI ENKODER	33
PERCOBAAN VII DEKODER	37
PERCOBAAN VIII FLIP-FLOP	41
PERCOBAAN IX PENCACAH	50
PERCOBAAN X REGISTER	58
DAFTAR PUSTAKA	64
LAMPIRAN 1: LAPORAN SEMENTARA	65
LAMPIRAN 2: DAFTAR NAMA <i>FILE</i> PERCOBAAN	87
LAMPIRAN 3: CONTOH HALAMAN SAMPUL LAPORAN PRAKTIKUM	89
LAMPIRAN 4: RUJUKAN TUGAS PENDAHULUAN	90

### DAFTAR *LINK* UNTUK *DOWNLOAD* SIMULATOR *BREADBOARD*

Simulator *breadboard* yang digunakan di dalam panduan praktikum ini merupakan aplikasi berbasis pemrograman Java. Perangkat lunak ini pertama kali dibuat pada tahun 2001 oleh Nicholas Glass yang didedikasikan kepada Departemen Ilmu Komputer, *the University of York*, Inggris. Selanjutnya, dilakukan *update* oleh Stephen Halstead, Robert Page, Michael Freeman, dan Chris Bailey hingga pada 1 Juli 2010 menghasilkan *Java Breadboard Simulator Version 1.11*. Aplikasi simulator *breadboard* ini dapat diunduh di link berikut ini.

**<http://muchlas.ee.uad.ac.id/downloads/JavaBreadBoard1-11.zip>**

Simulator *breadboard* dapat berjalan dengan baik jika tersedia komputer dengan spesifikasi standar untuk pengoperasian aplikasi *desktop*. Pada komputer harus terpasang sistem operasi keluarga Windows seperti Windows XP, Windows Vista atau Windows 7 dan program *Java Runtime Environment (JRE)*, yang dapat diunduh melalui link berikut ini.

**<http://muchlas.ee.uad.ac.id/downloads/jre-7u21-windows-i586.exe>**

## PERCOBAAN I WATAK GERBANG LOGIKA DASAR

### A. DESKRIPSI

Gerbang logika dasar OR, AND, dan NOT maupun gerbang universal NOR dan NAND sangat penting peranannya di dalam perancangan suatu rangkaian digital. Oleh sebab itu, pemahaman secara komprehensif terhadap watak setiap gerbang-gerbang logika tersebut sangat diperlukan.

Watak gerbang logika atau juga rangkaian logika pada umumnya, diketahui atau dipelajari melalui tabel kebenaran (*truth table*) yang dimilikinya, yakni suatu tabel yang mengekspresikan hubungan antara masukan (input) dengan keluaran (output) gerbang atau rangkaian logika. Secara teoritis, tabel kebenaran suatu gerbang atau rangkaian logika dapat disusun berdasarkan ekspresi Boole yang dimilikinya. Selain menggunakan tabel kebenaran, watak gerbang atau rangkaian logika juga dapat dikenali melalui diagram waktunya.

Pada percobaan ini Anda akan menyelidiki watak setiap gerbang logika dasar dan gerbang-gerbang universal dengan menentukan tabel kebenarannya secara eksperimen. Gerbang-gerbang yang diselidiki semuanya dari jenis TTL (*transistor-transistor logic*) dalam bentuk rangkaian terintegrasi.

### B. TUJUAN

Praktikum ini ditujukan untuk:

1. menentukan watak gerbang-gerbang logika dasar dan universal secara eksperimen; serta
2. membuktikan universalitas gerbang NOR dan NAND.

### C. ALAT DAN BAHAN

Alat dan bahan praktikum ini meliputi:

1. IC TTL 7432 (OR), 7408 (AND), 7404 (NOT), 7402 atau 7428 (NOR), 7400 (NAND), masing-masing 1 buah;
2. *DIP Switch* ganda dan tunggal masing-masing 1 buah, indikator LED 5 buah;

3. *Breadboard*, catu daya +5V dan kabel-kabel penghubung (sudah tersedia di dalam simulator).

#### D. TUGAS PENDAHULUAN

1. Gambarkan simbol gerbang-gerbang logika dasar OR, AND dan NOT serta gerbang universal NOR dan NAND!
2. Berdasarkan operasinya, sebutkan definisi masing-masing gerbang logika tersebut!
3. Tuliskan ekspresi Boole dari masing-masing gerbang logika tersebut!
4. Isilah tabel kebenaran berikut ini:

Tabel 1. Tabel pengamatan watak gerbang logika dasar

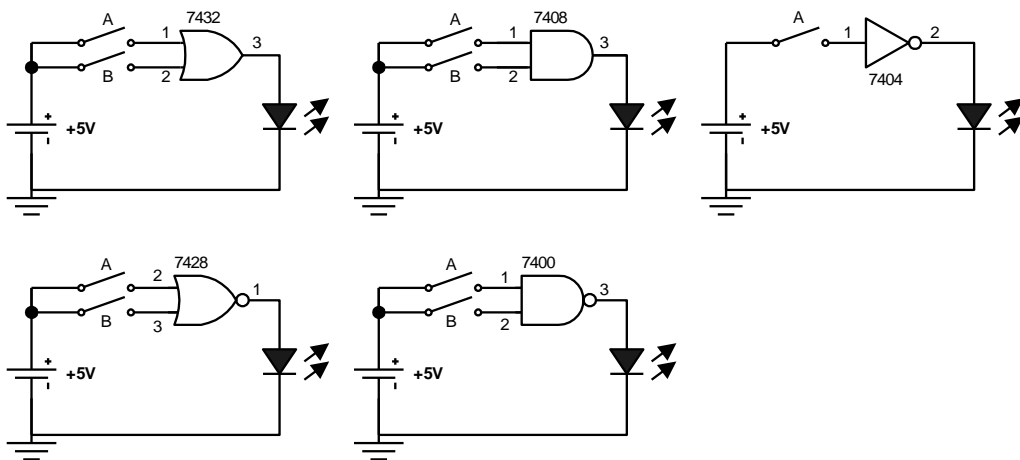
INPUT		OUTPUT				
A	B	OR	AND	NOT A	NOR	NAND
0	0					
0	1					
1	0					
1	1					

5. Susunlah rangkaian logika untuk menggantikan fungsi gerbang OR-2 input dan gerbang AND-2 input, serta gerbang NOT masing-masing menggunakan gerbang NAND dan NOR!
6. Sebutkan nomor seri IC TTL untuk gerbang OR, AND, NOT, NOR dan NAND serta gambarkan konfigurasi pin masing-masing IC tersebut!
7. Berdasarkan tujuan praktikum, susun rangkaian dan tabel-tabel pengamatan percobaan yang diperlukan untuk:
  - a. menentukan watak gerbang OR, AND, NOT, NOR dan NAND; serta
  - b. membuktikan bahwa semua gerbang logika dasar (OR, AND dan NOT) dapat dibuat dari NAND atau NOR!

#### E. PROSEDUR

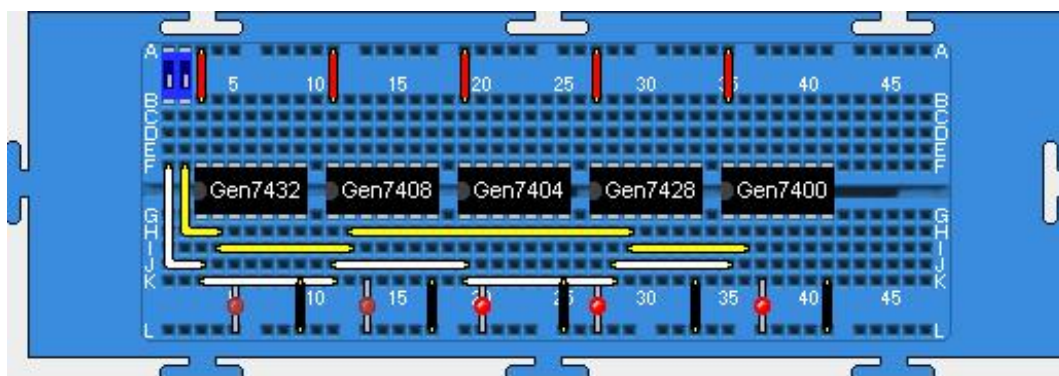
1. Menentukan watak gerbang logika dasar dan universal:
  - a. Untuk menyelidiki watak gerbang OR-2 input, AND-2 input, NOT, NOR-2 input dan NAND-2 input, gunakan rangkaian berikut ini!





Gambar 1. Rangkaian untuk menentukan watak gerbang logika dasar

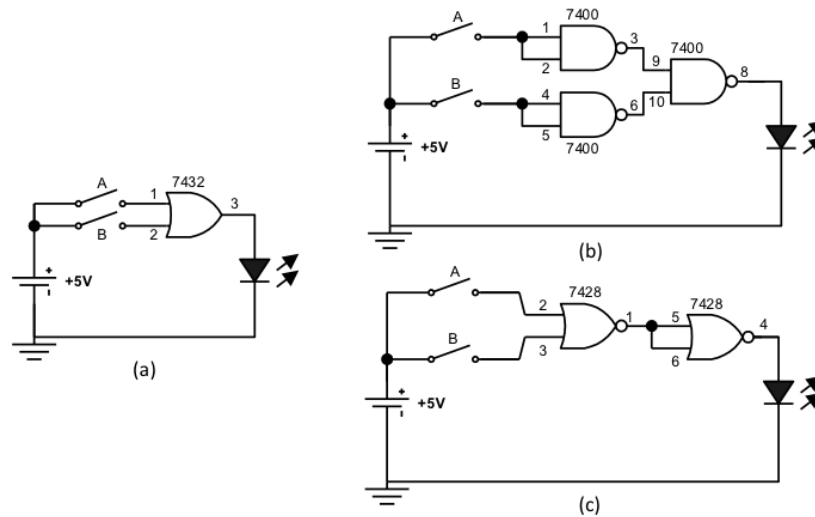
- b. Tempatkan IC 7432 (OR), 7408 (AND), 7404 (NOT), 7428 (NOR), 7400 (NAND), sebuah saklar ganda, dan 5 buah LED pada papan *breadboard*.
- c. Pasang catu daya pada semua IC yang digunakan, pasang saklar logika pada setiap input gerbang dan indikator LED pada setiap output gerbang. Salah satu bentuk susunan rangkaian untuk menyelidiki watak gerbang logika dasar menggunakan simulator *breadboard* ditunjukkan pada gambar berikut ini.



Gambar 2. Rangkaian untuk menyelidiki watak gerbang logika dasar menggunakan simulator *breadboard*

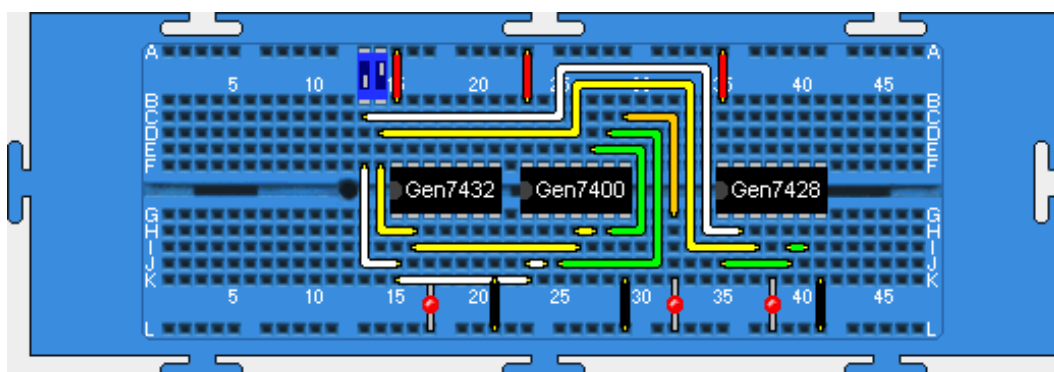
- d. Melalui saklar logika, berikan kombinasi nilai logika yang mungkin pada inputnya dan amati outputnya pada setiap gerbang yang ada.
- e. Catat hasilnya ke dalam tabel pengamatan.

2. Membuktikan bahwa gerbang OR dapat dibuat dari NAND atau NOR
- a. Gunakan rangkaian OR asli, rangkaian OR menggunakan gerbang NAND, dan rangkaian OR menggunakan gerbang NOR berikut ini untuk percobaan.



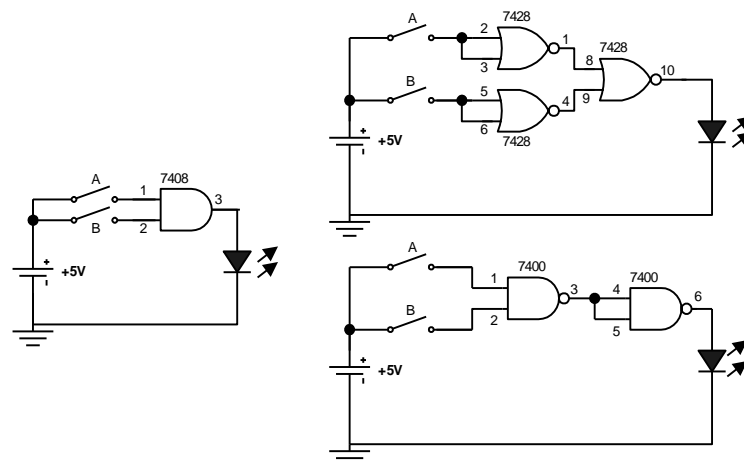
Gambar 3. Rangkaian: (a) OR asli; (b) OR menggunakan NAND; dan (c) OR menggunakan NOR

- b. Tempatkan IC 7432 (OR), 7400 (NAND), 7428 (NOR), 1 saklar ganda, dan 3 buah LED pada papan *breadboard*.
- c. Pasang catu daya pada semua IC yang digunakan, pasang saklar logika pada setiap input gerbang dan indikator LED pada setiap output gerbang. Salah satu bentuk susunan rangkaian untuk membuktikan gerbang OR dapat dibuat dari gerbang NAND atau NOR menggunakan simulator *breadboard* ditunjukkan pada gambar berikut ini.



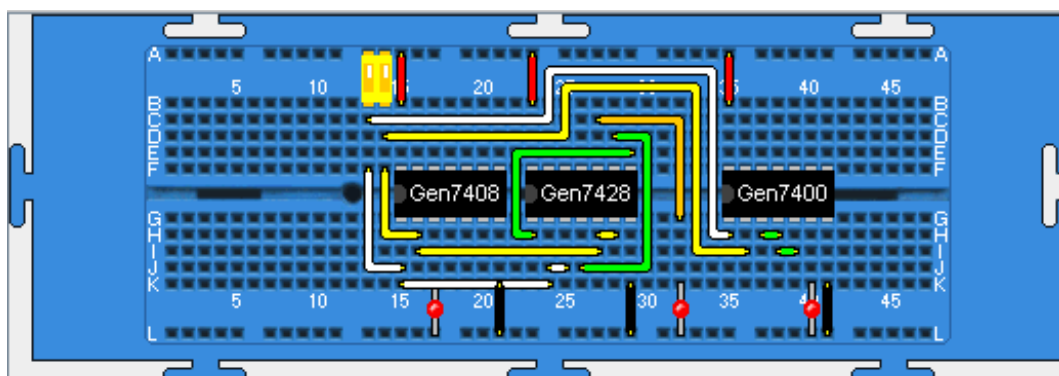
Gambar 4. Rangkaian gerbang OR dari NAND atau NOR menggunakan simulator *breadboard*

- d. Melalui saklar logika, berikan kombinasi nilai logika yang mungkin pada inputnya dan amati outputnya pada setiap gerbang yang ada!
  - e. Catat hasilnya pada tabel pengamatan!
3. Membuktikan bahwa gerbang AND dapat dibuat dari NAND atau NOR
    - a. Lakukan langkah-langkah seperti pada butir 2.a. sampai dengan 2.e. di atas untuk rangkaian gerbang AND.
    - b. Rangkaian untuk eksperimen ini ditunjukkan pada gambar berikut.



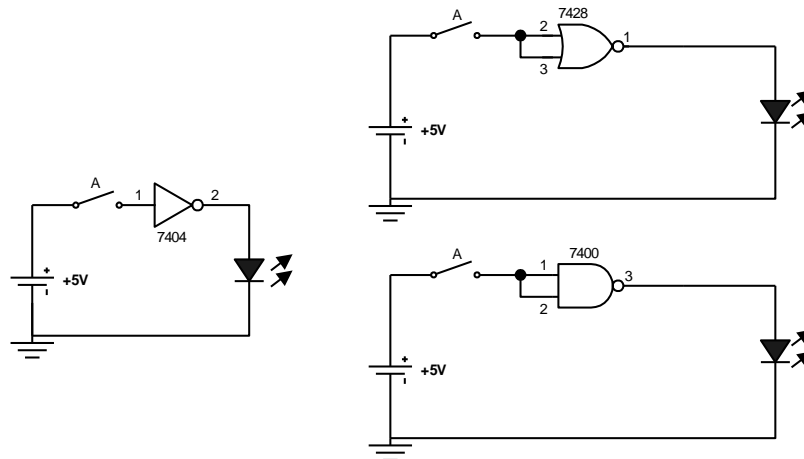
Gambar 5. Rangkaian: (a) AND asli; (b) AND menggunakan NOR; dan (c) AND menggunakan NAND

- c. Sedangkan susunan rangkaian menggunakan simulator *breadboard* adalah sebagai berikut.



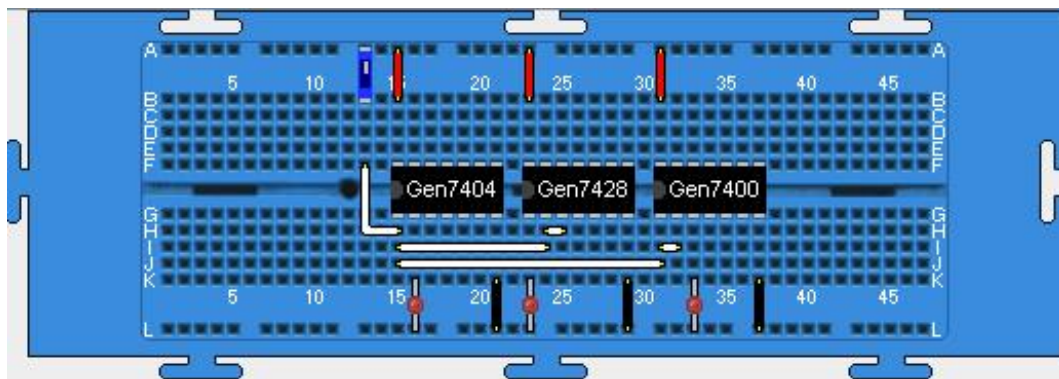
Gambar 6. Rangkaian gerbang AND dari NOR atau NAND menggunakan simulator *breadboard*

4. Membuktikan bahwa gerbang NOT dapat dibuat dari NAND atau NOR
- Lakukan langkah-langkah butir 2.a. sampai dengan 2.e. di atas untuk rangkaian gerbang NOT.
  - Rangkaian untuk eksperimen ini ditunjukkan pada gambar berikut.



Gambar 7. Rangkaian: (a) NOT asli; (b) NOT menggunakan NOR; dan (c) NOT menggunakan NAND

- Rangkaian menggunakan simulator *breadboard* adalah sebagai berikut.



Gambar 8. Rangkaian gerbang NOT dari NOR atau NAND menggunakan simulator *breadboard*

## F. TUGAS LAPORAN

Buatlah tugas laporan praktik dengan sistematika: (1) tujuan, (2) alat dan bahan, (3) langkah percobaan, (4) hasil dan pembahasan, dan (5) kesimpulan. Bagian tujuan, alat dan bahan diuraikan sesuai dengan uraian pada panduan ini.

Uraian tentang langkah percobaan diambil dari panduan ini dan pengalaman anda dalam melaksanakan percobaan. Hasil dan pembahasan berisi penyajian data yang diperoleh melalui percobaan baik yang bersifat kuantitatif maupun kualitatif, perhitungan sesuai tujuan praktik, dan diskusi terhadap hasil analisis data.

Kesimpulan merupakan rangkuman dari hasil pembahasan. Dalam percobaan ini pembahasan diarahkan untuk menjawab pertanyaan-pertanyaan sebagai berikut:

1. Bagaimanakah watak gerbang logika dasar OR, AND dan NOT?
2. Bagaimanakah watak gerbang logika universal NOR dan NAND?
3. Dari hasil membandingkan antara tabel kebenaran yang diperoleh secara eksperimen dengan tabel kebenaran secara teoritik berdasarkan ekspresi Boole untuk masing-masing gerbang, apakah diperoleh hasil yang sama? Jadi tabel kebenaran dapat diperoleh melalui cara apa saja?
4. Dari hasil membandingkan tabel kebenaran OR asli dengan tabel kebenaran OR menggunakan gerbang NAND dan OR menggunakan gerbang NOR, apakah diperoleh hasil yang sama? Apakah gerbang OR bisa digantikan oleh rangkaian dengan gerbang NAND saja? Apakah gerbang OR bisa digantikan oleh rangkaian dengan gerbang NOR saja?
5. Dari hasil membandingkan tabel kebenaran AND asli dengan tabel kebenaran AND menggunakan gerbang NOR, dan AND menggunakan gerbang NAND, apakah diperoleh hasil yang sama? Apakah gerbang AND bisa digantikan oleh rangkaian dengan gerbang NAND saja? Apakah gerbang AND bisa digantikan oleh rangkaian dengan gerbang NOR saja?
6. Dari hasil membandingkan tabel kebenaran gerbang NOT asli dengan tabel kebenaran NOT menggunakan gerbang NAND dan NOT menggunakan gerbang NOR, apakah diperoleh hasil yang sama? Apakah gerbang NOT bisa digantikan oleh rangkaian dengan gerbang NAND saja? Apakah gerbang NOT bisa digantikan oleh rangkaian dengan gerbang NOR saja?
7. Berdasarkan jawaban pertanyaan 4, 5, dan 6, bersifat apakah gerbang NOR dan NAND itu? Bagaimana kesimpulan Anda terhadap peranan kedua gerbang tersebut dalam perancangan rangkaian digital?

## PERCOBAAN II MINIMALISASI RANGKAIAN LOGIKA

### A. DESKRIPSI

Perancangan rangkaian logika yang kompleks akan melibatkan banyak gerbang logika dasar sebagai pembentuk rangkaian yang dirancang. Untuk keperluan efisiensi, baik dari segi biaya maupun teknis perancangan, seringkali dilakukan penyederhanaan atau minimalisasi rangkaian logika. Prosedur minimalisasi merupakan salah satu azas perancangan rangkaian logika yang selalu digunakan untuk memperoleh rancangan rangkaian yang kompak dan murah. Prosedur ini dapat mencakup minimalisasi dalam jumlah gerbang maupun dalam hal jenis gerbang yang dilibatkan.

Minimalisasi rangkaian logika dapat dilakukan dengan bantuan aljabar Boole dan peta Karnaugh. Pada praktikum ini Anda akan melakukan minimalisasi suatu rangkaian logika yang cukup kompleks menjadi rangkaian logika yang lebih sederhana. Tabel kebenaran keduanya dibandingkan untuk memperoleh informasi tentang kesamaan wataknya.

### B. TUJUAN

Melalui praktikum ini diharapkan Anda memiliki kemampuan:

1. merancang rangkaian logika dengan jumlah gerbang minimum; serta
2. merancang rangkaian logika dengan satu jenis gerbang saja.

### C. ALAT DAN BAHAN

Alat dan bahan pada praktikum ini terdiri atas:

1. IC TTL OR (7432), AND (7408), NOR (7428), dan NAND (7400) masing-masing 1 buah;
2. *DIP Switch Triple* 1 buah;
3. Indikator LED 1 buah; serta
4. *Breadboard*, catu daya +5V dan kabel-kabel penghubung (sudah tersedia di dalam simulator).

#### D. TUGAS PENDAHULUAN

1. Tuliskan ekspresi hukum-hukum aljabar Boole yang mencakup hukum-hukum dasar untuk OR, AND, dan NOT, hukum-hukum asosiatif, komutatif, distributif, absorbtif dan de Morgan!
2. Jelaskan pengertian *sum of product*, *standard sum of product*, *product of sum*, *standard product of sum*, *minterm*, dan *maxterm*! Berikan pula contoh-contohnya!
3. Perhatikan persamaan logika berikut ini:

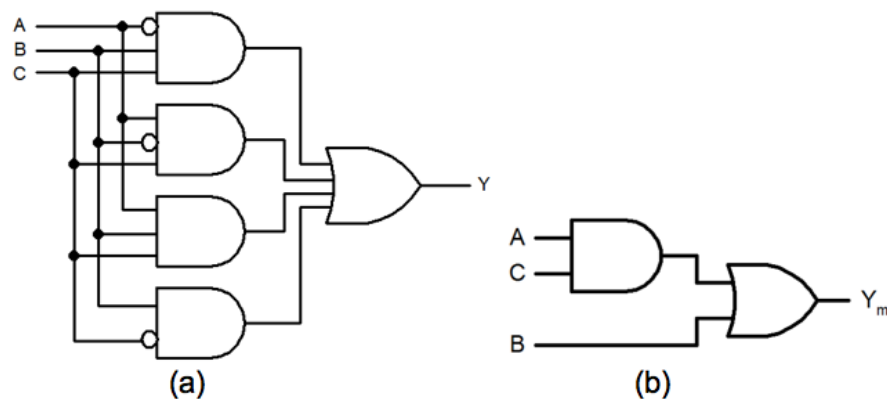
$$Y = \bar{A}BC + A\bar{B}C + ABC + B\bar{C}$$

- a. Gambarkan rangkaian dan susun tabel kebenaran dari persamaan logika tersebut!
  - b. Lakukan minimalisasi menggunakan aljabar Boole dan Peta Karnaugh! Apakah menghasilkan persamaan logika yang sama?
  - c. Gambarkan rangkaian dan susun tabel kebenaran dari persamaan hasil minimalisasi tersebut! Apakah tabel kebenarannya sama dengan rangkaian sebelum diminimalisasikan?
  - d. Ubah persamaan hasil minimalisasi ke dalam bentuk NAND saja! Susun rangkaian dan tabel kebenarannya! Apakah tabel kebenarannya sama dengan rangkaian sebelum diminimalisasikan?
  - e. Ubah persamaan hasil minimalisasi ke dalam bentuk NOR saja! Susun rangkaian dan tabel kebenarannya! Apakah tabel kebenarannya sama dengan rangkaian sebelum diminimalisasikan?
4. Berdasarkan tujuan praktikum, susun rangkaian dan tabel-tabel pengamatan percobaan yang diperlukan untuk:
    - a. merancang rangkaian  $Y = \bar{A}BC + A\bar{B}C + ABC + B\bar{C}$  dengan jumlah gerbang minimum; dan
    - b. merancang rangkaian  $Y = \bar{A}BC + A\bar{B}C + ABC + B\bar{C}$  dengan satu jenis gerbang saja yakni NAND saja dan NOR saja!

## E. PROSEDUR

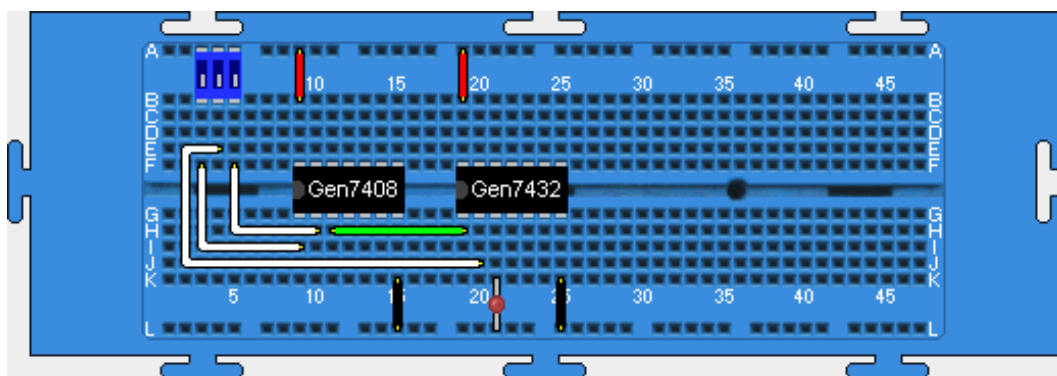
### 1. Merancang rangkaian logika dengan jumlah gerbang minimum

- a. Lakukan minimalisasi persamaan  $Y = \bar{A}BC + A\bar{B}C + ABC + B\bar{C}$ . Jika Anda benar dalam melakukan proses minimalisasi, maka akan dihasilkan rangkaian dengan jumlah gerbang minimal yakni  $Y_m = AC + B$  dan rangkaian seperti berikut ini.



Gambar 9. Minimalisasi rangkaian: (a)  $Y = \bar{A}BC + A\bar{B}C + ABC + B\bar{C}$ ; (b) hasil penyederhanaan  $Y_m = AC + B$

- b. Susun rangkaian yang telah diminimalisasi tersebut di atas *breadboard*, diawali dengan menempatkan terlebih dahulu IC 7408 (AND), 7432 (OR), saklar *triple*, dan sebuah LED di atas *breadboard*.
- c. Pasang catu daya pada semua IC yang digunakan, pasang saklar logika pada input rangkaian dan indikator LED pada output rangkaian. Salah satu bentuk susunan rangkaian  $Y_m = AC + B$  menggunakan simulator *breadboard* ditunjukkan pada gambar berikut ini.

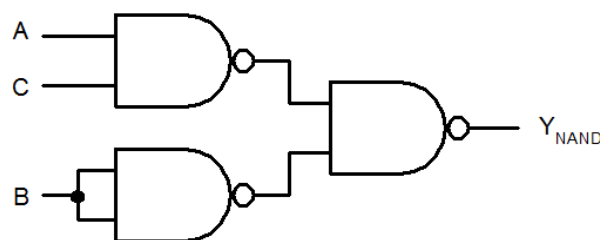




Gambar 10. Rangkaian dalam bentuk minimum  $Y_m = AC + B$  di atas *breadboard*

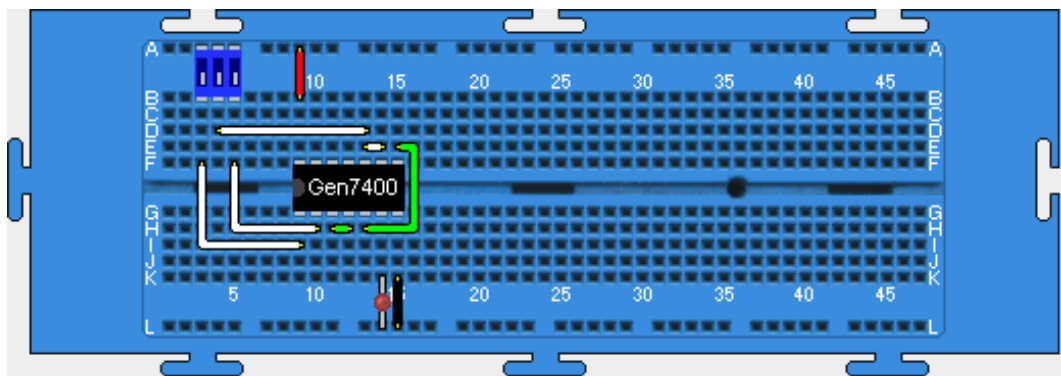
- d. Melalui saklar logika, berikan kombinasi nilai logika yang mungkin pada input rangkaian dan amati outputnya.
  - e. Catat hasilnya ke dalam tabel pengamatan.
2. Merancang rangkaian logika dengan gerbang NAND saja

- a. Ubah hasil minimalisasi  $Y = \overline{A}BC + A\overline{B}C + ABC + \overline{B}C$  ke dalam bentuk NAND. Jika Anda melakukannya dengan benar, maka akan diperoleh persamaan  $Y_{\text{NAND}} = \overline{\overline{AC} \cdot \overline{B}}$  dan rangkaian sebagai berikut.



Gambar 11. Rangkaian hasil minimalisasi menggunakan NAND saja ( $Y_{\text{NAND}} = \overline{\overline{AC} \cdot \overline{B}}$ )

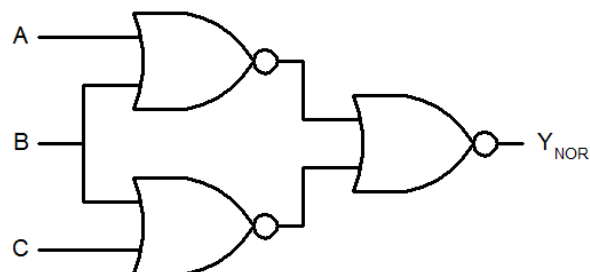
- b. Susun rangkaian dalam bentuk NAND tersebut di atas papan *breadboard*, diawali dengan menempatkan terlebih dahulu IC 7400 (NAND), saklar *triple*, dan sebuah LED di atas *breadboard*.
- c. Pasang catu daya pada IC yang digunakan, pasang saklar logika pada input rangkaian dan indikator LED pada output rangkaian. Salah satu bentuk susunan rangkaian  $Y_{\text{NAND}} = \overline{\overline{AC} \cdot \overline{B}}$  menggunakan simulator *breadboard* ditunjukkan pada gambar berikut ini.



Gambar 12. Rangkaian dalam bentuk minimum  $Y_{\text{NAND}} = \overline{\overline{AC} \cdot \overline{B}}$

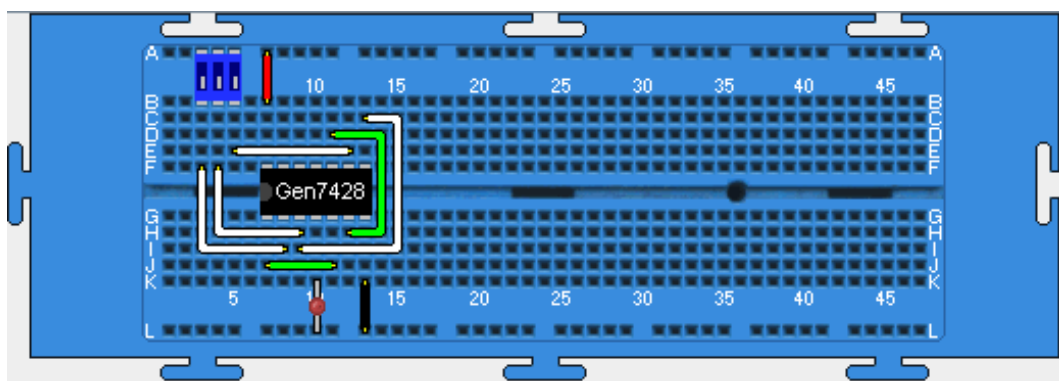
menggunakan simulator *breadboard*

- d. Melalui saklar logika, berikan kombinasi nilai logika yang mungkin pada input rangkaian dan amati outputnya.
  - e. Catat hasilnya ke dalam tabel pengamatan.
3. Merancang rangkaian logika dengan gerbang NOR saja
- a. Ubah hasil minimalisasi  $Y = \overline{A}BC + A\overline{B}C + ABC + \overline{B}\overline{C}$  ke dalam bentuk NOR. Jika Anda melakukannya dengan benar, maka akan diperoleh persamaan  $Y_{\text{NOR}} = \overline{\overline{A + B}} + \overline{\overline{B + C}}$  dan rangkaian sebagai berikut.



Gambar 13. Rangkaian hasil minimalisasi menggunakan NOR saja

- b. Selanjutnya, ikuti langkah seperti pada nomor 2.b. sampai dengan 2.e. di atas. Untuk rangkaian dengan NOR saja, susunan rangkaian menggunakan simulator *breadboard* dapat berbentuk seperti gambar berikut ini.



Gambar 14. Rangkaian dalam bentuk minimum  $Y_{\text{NOR}} = \overline{\overline{A + B}} + \overline{\overline{B + C}}$  menggunakan simulator *breadboard*

## F. TUGAS LAPORAN

Buatlah tugas laporan praktik dengan sistematika: (1) tujuan, (2) alat dan bahan, (3) langkah percobaan, (4) hasil dan pembahasan, dan (5) kesimpulan. Bagian tujuan, alat dan bahan diuraikan sesuai dengan uraian pada panduan ini. Uraian tentang langkah percobaan diambil dari panduan ini dan pengalaman anda dalam melaksanakan percobaan. Hasil dan pembahasan berisi penyajian data yang diperoleh melalui percobaan baik yang bersifat kuantitatif maupun kualitatif, perhitungan sesuai tujuan praktik, dan diskusi terhadap hasil analisis data.

Kesimpulan merupakan rangkuman dari hasil pembahasan. Dalam percobaan ini pembahasan diarahkan untuk menjawab pertanyaan-pertanyaan sebagai berikut:

1. Berdasarkan percobaan, sebutkan jenis minimalisasi yang anda lakukan? Seberapa besar peranan gerbang NOR dan NAND dalam proses perancangan rangkaian logika?
2. Seberapa penting prosedur minimalisasi rangkaian itu dilakukan dalam proses perancangan rangkaian logika? Kemukakan alasan-alasannya?

### PERCOBAAN III KOMPARATOR DAN PENJUMLAH BINER

#### A. DESKRIPSI

Rangkaian komparator atau pembanding biner adalah salah satu rangkaian logika kombinasi, yakni rangkaian logika yang keluarannya suatu saat tidak tergantung pada nilai keluaran saat yang lalu. Komparator biner berfungsi membandingkan dua data biner pada inputnya.

Terdapat 2 jenis rangkaian komparator 1-bit yakni *non-equality comparator* dan *equality comparator*. *Non-equality comparator* merupakan rangkaian logika yang memberikan keadaan output tinggi (logika 1) jika keadaan input-inputnya berbeda, dan *equality comparator* akan memberikan keadaan output tinggi jika keadaan input-inputnya sama. Kecuali dapat diperoleh dengan cara membangun menggunakan gerbang logika dasar, *non-equality comparator* juga dapat diperoleh dalam bentuk gerbang exclusive-OR (EX-OR) dan *equality comparator* dalam bentuk gerbang exclusive-NOR (EX-NOR), keduanya tersedia dalam bentuk IC. Pada percobaan ini Anda akan menyelidiki watak komparator 1-bit yang disusun dengan menggunakan gerbang logika dasar maupun komparator dalam bentuk IC TTL.

Selain komparator, jenis rangkaian logika kombinasi lainnya adalah penjumlah biner (*binary adder*). Rangkaian penjumlah biner merupakan penerapan dari rangkaian komparator biner. Rangkaian ini sangat penting peranannya di dalam bidang komputer digital, karena operasi-operasi aritmatika yang lebih kompleks seperti perkalian hakekatnya merupakan penjumlahan yang diulang-ulang sehingga memerlukan rangkaian penjumlah dan pembagian adalah pengurangan yang diulang-ulang sehingga memerlukan rangkaian pengurang.

Pada percobaan ini Anda akan menyelidiki watak penjumlah biner jenis *half adder*, dan *full adder*, yang dibangun dari gerbang logika dasar. Anda akan menyelidiki pula watak *parallel full adder* yang merupakan landasan pembangunan rangkaian *arithmetic logic unit* pada komputer digital.

## B. TUJUAN

Melalui percobaan ini diharapkan anda dapat:

1. menentukan watak rangkaian *non-equality comparator* dan *equality comparator*;
2. menentukan watak *half adder* dan *full adder* 1-bit; serta
3. merancang dan menentukan watak *parallel full adder* 4-bit.

## C. ALAT DAN BAHAN

Alat dan bahan pada praktikum ini terdiri atas:

1. IC TTL NOT (7404), AND (7408), OR (7432), dan EX-OR (7486) masing-masing 1 buah serta modul *full-adder* 1-bit 4 buah;
2. *DIP switch* tunggal, ganda, dan *triple* masing-masing 1 buah;
3. Indikator LED 2 buah;
4. *Keypad* heksadesimal 2 buah dan peraga heksadesimal 1 buah; serta
5. *Breadboard*, catu daya +5V dan kabel-kabel penghubung (sudah tersedia di dalam simulator).

## D. TUGAS PENDAHULUAN

### Komparator Biner:

1. Sebutkan jenis-jenis komparator biner dan definisi serta tabel kebenarannya!
2. Dari tabel kebenaran yang telah disusun, tuliskan ekspresi Boole dari komparator-komparator tersebut dan gambarkan simbol serta rangkaiannya!
3. Sebutkan nomor seri IC TTL EX-OR dan EX-NOR, serta gambarkan identifikasi pin IC TTL tersebut (lihat *data sheet IC*)!
4. Berdasarkan tujuan praktikum, susun rangkaian dan tabel-tabel pengamatan percobaan yang diperlukan untuk:
  - a. menentukan watak rangkaian *non-equality comparator*; dan
  - b. menentukan watak rangkaian *equality comparator*.

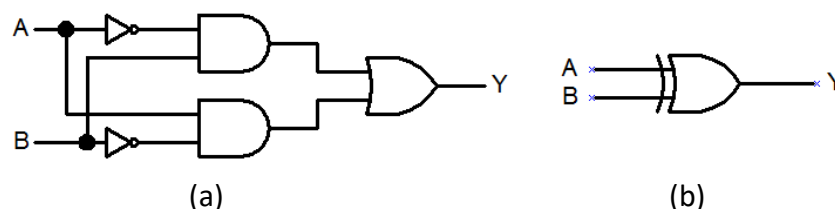
### Penjumlah Biner:

1. Gambarkan rangkaian *half adder* yang dibangun dari gerbang EX-OR dan AND! Tuliskan tabel kebenarannya!

2. Susun tabel kebenaran *full adder*, turunkan ekspresi Boole fungsi *sum* (S) dan lakukan minimalisasi terhadap fungsi *next carry* (Cn) dari rangkaian *full adder* tersebut! Dari fungsi S dan Cn yang diperoleh, gambarkan rangkaian *full adder* tersebut dengan fungsi S diimplementasikan dengan XOR!
3. Gambarkan rangkaian *parallel full adder* 4-bit! Berikan contoh operasi penjumlahan yang dilakukan oleh *parallel full adder* 4-bit jika Augend (A)=1011, Addend (B)= 1001, dan carry input (Ci)=1.
4. Berdasarkan tujuan praktikum, susun rangkaian dan tabel-tabel pengamatan percobaan yang diperlukan untuk:
  - a. menentukan watak rangkaian *half-adder* 1-bit;
  - b. menentukan watak rangkaian *full-adder* 1 bit; serta
  - c. merancang dan menentukan watak rangkaian *parallel full adder* 4-bit

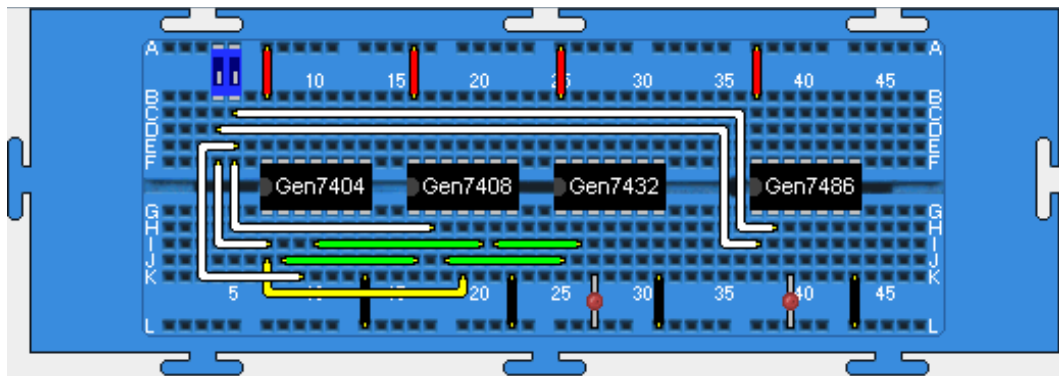
## E. PROSEDUR

1. Menentukan watak rangkaian *non-equality comparator* dan *equality comparator*
  - a. Gunakan rangkaian *non-equality comparator* dalam bentuk *sum of product* (SOP) dan bentuk XOR berikut ini untuk percobaan.



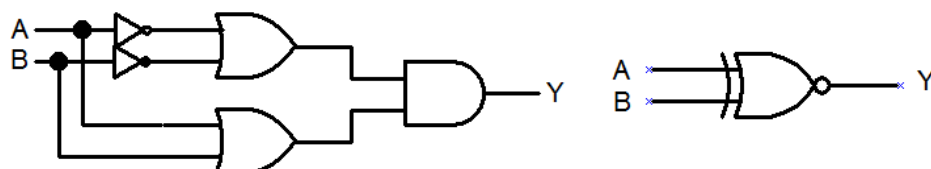
Gambar 15. Rangkaian *Non-Equality Comparator*:  
(a) Bentuk SOP; (b) XOR

- b. Tempatkan IC 7404 (NOT), 7408 (AND), IC 7432 (OR), 7486 (XOR), saklar ganda, dan 2 buah LED pada papan *breadboard*.
- c. Pasang catu daya pada semua IC yang digunakan, pasang saklar logika pada input rangkaian dan indikator LED pada output rangkaian. Salah satu bentuk susunan rangkaian untuk menyelidiki watak *Non-Equality Comparator* menggunakan simulator *breadboard* ditunjukkan pada gambar berikut ini.

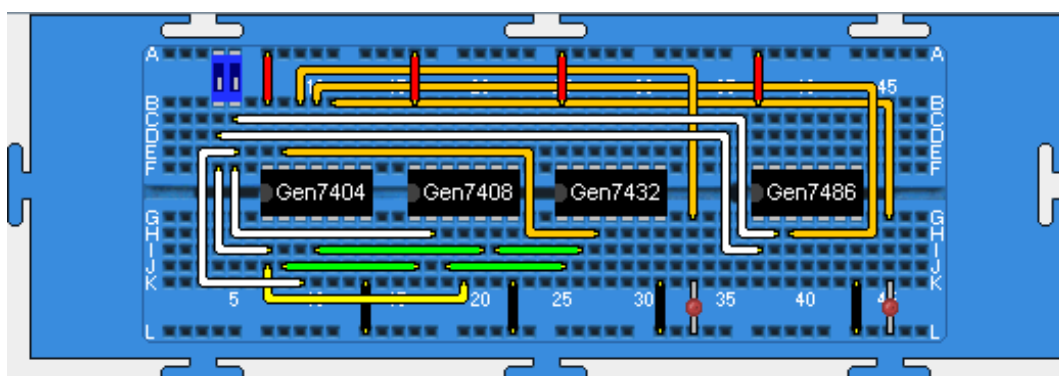


Gambar 16. Susunan *Non-Equality Comparator* menggunakan *breadboard*

- d. Melalui saklar logika, berikan kombinasi nilai logika yang mungkin pada input rangkaian dan amati outputnya
- e. Catat hasilnya ke dalam tabel pengamatan
- f. Lakukan langkah-langkah yang sama seperti di atas untuk rangkaian *equality comparator* dalam bentuk *product of sum* (POS) dan XNOR. Rangkaian dan susunannya dalam papan *breadboard* ditunjukkan pada gambar berikut ini.

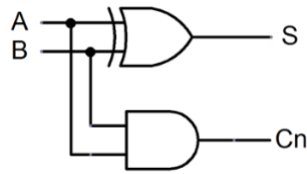


Gambar 17. Rangkaian *Equality Comparator*: (a) Bentuk POS; (b) XNOR



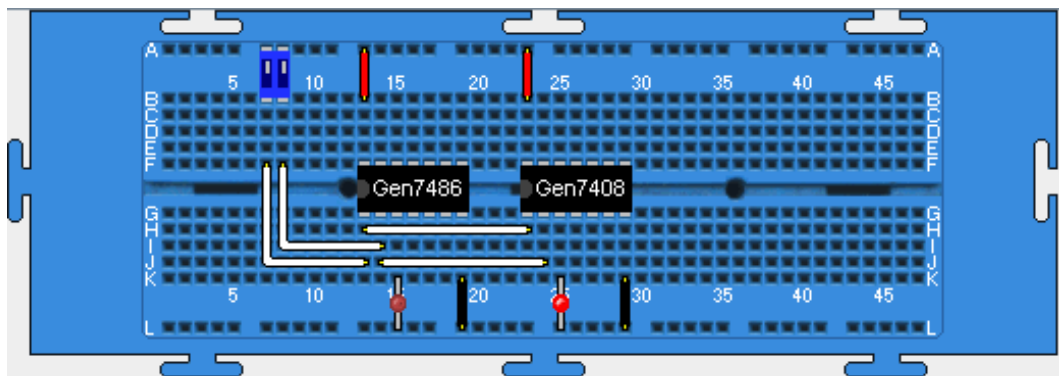
Gambar 18. Rangkaian *Equality Comparator* menggunakan simulator *breadboard*

2. Menentukan watak rangkaian *half-adder* 1-bit dan *full-adder* 1-bit
  - a. Gunakan rangkaian *half-adder* berikut ini untuk percobaan.



Gambar 19. Rangkaian *half-adder* 1-bit

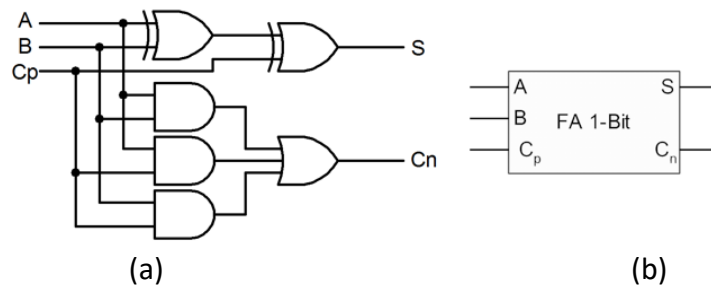
- b. Tempatkan IC 7486 (XOR), 7408 (AND), saklar ganda, dan 2 buah LED pada papan *breadboard*
- c. Pasang catu daya pada semua IC yang digunakan, pasang saklar logika pada input rangkaian dan indikator LED pada output rangkaian. Salah satu bentuk susunan rangkaian untuk menyelidiki watak *half-adder* menggunakan simulator *breadboard* ditunjukkan pada gambar berikut ini.



Gambar 20. Rangkaian *half-adder* menggunakan simulator *breadboard*

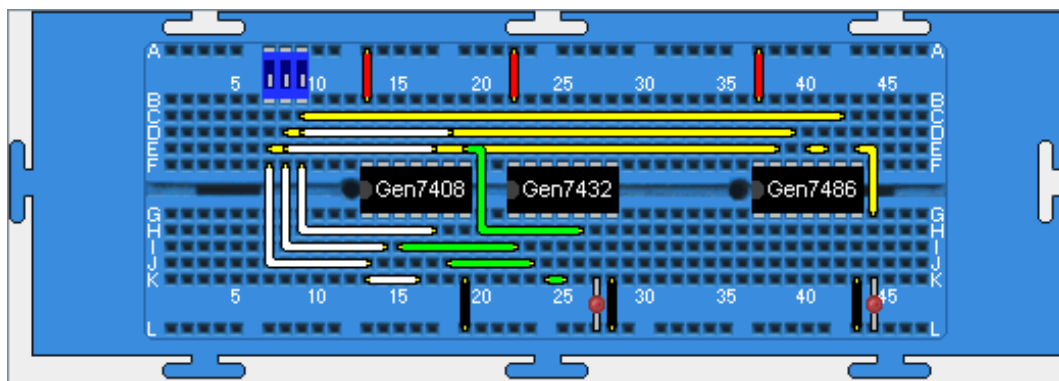
- d. Melalui saklar logika, berikan kombinasi nilai logika yang mungkin pada input rangkaian dan amati outputnya.
- e. Catat hasilnya ke dalam tabel pengamatan.
- f. Lakukan langkah-langkah yang sama seperti di atas untuk rangkaian *full-adder*. Rangkaianya ditunjukkan pada gambar berikut ini.





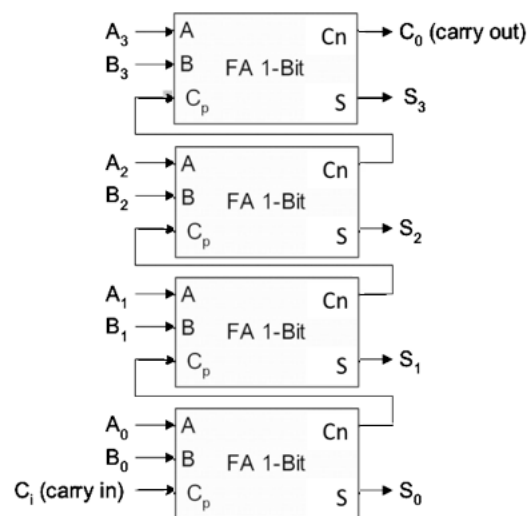
Gambar 21. Full-adder 1-bit: (a) rangkaian dengan XOR; (b) simbol

Salah satu bentuk susunan rangkaian *full-adder* 1-bit menggunakan simulator *breadboard* ditunjukkan pada gambar berikut ini.



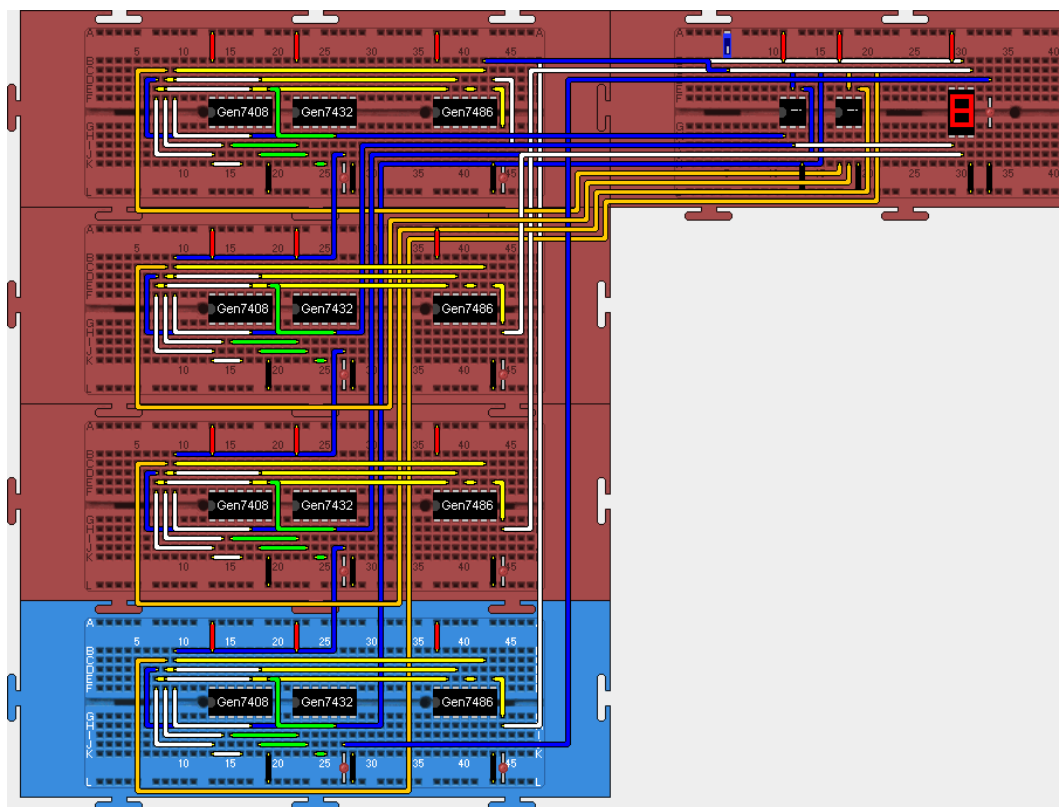
Gambar 22. Rangkaian *full-adder* 1-bit menggunakan simulator *breadboard*

3. Merancang dan menentukan watak rangkaian *parallel full adder* 4-bit
  - a. Gunakan rangkaian *parallel full adder* 4-bit yang terdiri atas 4 buah rangkaian *full adder* 1-bit berikut ini untuk percobaan.



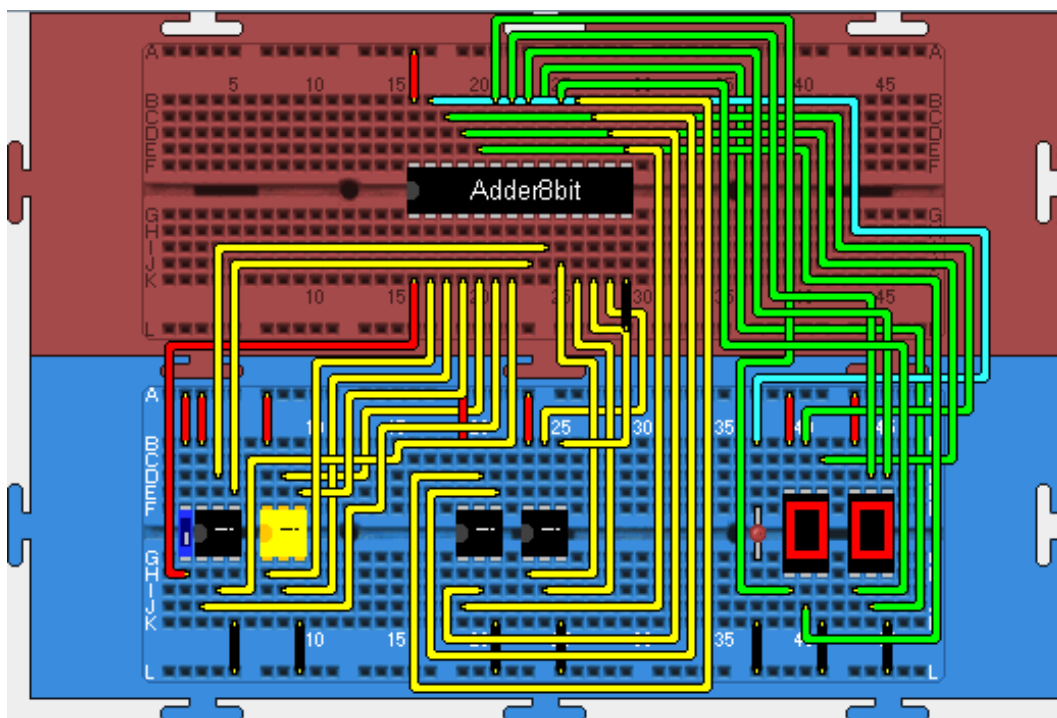
Gambar 23. Full-adder paralel

- b. Tempatkan 4 buah rangkaian *full-adder* 1-bit (Gambar 22), sebuah saklar tunggal, dua buah *keypad*, sebuah peraga heksadesimal 7-segmen, dan sebuah LED di atas *breadboard*.
- c. Pasang catu daya pada semua piranti yang digunakan.
- d. Pasang saklar logika pada *carry-in* (Ci) dan indikator LED pada output *carry* (Cn).
- e. Pasang *keypad* heksadesimal pada input *augend* (A) dan *addend* (B) serta peraga heksadesimal pada output *sum*.
- f. Hubungkan semua kaki IC Cn ke Cp sesuai rangkaian pada Gambar 23.
- g. Salah satu bentuk susunan rangkaian *full-adder* paralel 4-bit menggunakan simulator *breadboard* ditunjukkan pada gambar berikut ini.



Gambar 24. *Full-adder* paralel 4-bit menggunakan simulator *breadboard*

- h. Melalui *keypad* berikan input angka pada *augend* (A) dan *addend* (B), amati dan catat output *sum* dan *carry* ke dalam tabel pengamatan.
- i. Lakukan langkah di atas untuk berbagai input *augend* (A) dan *addend* (B) dan akhiri praktik dengan memberikan nilai  $A=F$ , nilai  $B=1$ , serta  $C_i=0$ .
- j. Ulangi percobaan untuk rangkaian *full-adder* paralel menggunakan IC *full-adder* 8-bit
- k. IC *full-adder* 8-bit diambil dengan cara klik **Insert>Chip>double click CPU>double click jx\_york\_ac\_uk>double click j1>Adder8bit**.
- l. Komponen *keypad* diambil dari klik **Insert>Chip>double click Components>HexKeyPad** dan untuk peraga 7-segmen diambil dari **Insert>Chip>double click Components>HexDisplay**.
- m. Salah satu bentuk susunan rangkaian *full-adder* paralel 8-bit ditunjukkan gambar berikut ini.



Gambar 25. *Full-adder* paralel 8-bit menggunakan simulator *breadboard*

## F. TUGAS LAPORAN

Buatlah tugas laporan praktik dengan sistematika: (1) tujuan, (2) alat dan bahan, (3) langkah percobaan, (4) hasil dan pembahasan, dan (5) kesimpulan. Bagian tujuan, alat dan bahan diuraikan sesuai dengan uraian pada panduan ini. Uraian tentang langkah percobaan diambil dari panduan ini dan pengalaman anda dalam melaksanakan percobaan. Hasil dan pembahasan berisi penyajian data yang diperoleh melalui percobaan baik yang bersifat kuantitatif maupun kualitatif, perhitungan sesuai tujuan praktik, dan diskusi terhadap hasil analisis data.

Kesimpulan merupakan rangkuman dari hasil pembahasan. Dalam percobaan ini pembahasan diarahkan untuk menjawab pertanyaan-pertanyaan sebagai berikut:

1. Apakah secara eksperimen, tabel kebenaran *non-equality comparator* menggunakan SOP sama dengan XOR? Jadi, rangkaian *non-equality comparator* dapat diwujudkan dalam bentuk apa saja? Lebih efisien manakah implementasi *non-equality comparator* menggunakan SOP dan XOR?
2. Apakah secara eksperimen, tabel kebenaran *equality comparator* menggunakan SOP sama dengan XNOR? Jadi, rangkaian *equality comparator* dapat diwujudkan dalam bentuk apa saja? Lebih efisien manakah implementasi *equality comparator* menggunakan SOP dan XNOR?
3. Apakah output *non-equality comparator* (XOR) membentuk fungsi detektor jumlah ganjil? Mengapa rangkaian output *sum* (S) pada *half adder* maupun *full adder* menggunakan fungsi XOR?
4. Dilihat dari tabel kebenarannya, apa perbedaan antara *half adder* dan *full-adder*?
5. Berdasarkan pengamatan terhadap hasil penjumlahan yang dilakukan oleh *full adder* paralel 4-bit, berapa hasil tertinggi yang dapat ditampilkan dalam heksadesimal? Apa artinya?

## PERCOBAAN IV MULTIPLEKSER

### A. DESKRIPSI

Multiplexer (sering disebut dengan MUX saja yang merupakan singkatan dari istilah aslinya yakni *multiplexer*) termasuk rangkaian logika kombinasi. MUX merupakan suatu rangkaian yang berfungsi memilih salah satu dari beberapa sinyal masukan untuk disalurkan melalui keluarannya dengan bantuan sinyal kendali. Pada dasarnya MUX bertugas seperti saklar pemilih sehingga disebut juga sebagai pemilih data (*data selector*).

Melalui percobaan ini Anda akan menyelidiki watak MUX sederhana yang dibangun dengan menggunakan gerbang logika dasar.

### B. TUJUAN

Melalui percobaan ini diharapkan anda dapat menentukan watak:

1. MUX 1-bit 4 ke 1
2. MUX 3-bit 2 ke 1; serta

### C. ALAT DAN BAHAN

Alat dan bahan yang diperlukan dalam praktikum ini adalah:

1. IC TTL OR 2-input (7432), NOT (7404) masing-masing 1 buah, AND 3-input (7411) dan AND 2-input (7408) masing-masing 2 buah,
2. DIP *switch*: tunggal, ganda, *triple* dan *quad* masing-masing 1 buah, *keypad* heksadesimal 2 buah,
3. Indikator LED 4 buah, peraga heksadesimal 1 buah, serta
4. *Breadboard*, catu daya +5 V dan kabel penghubung (sudah tersedia di dalam simulator).

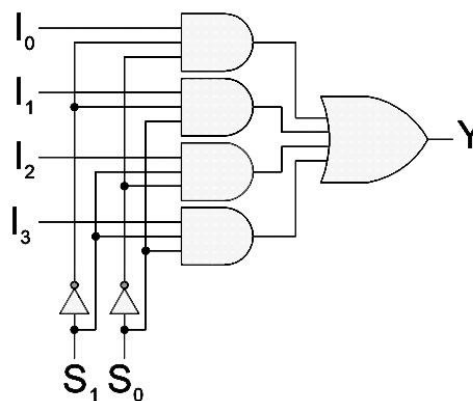
### D. TUGAS PENDAHULUAN

1. Jelaskan pengertian MUX!
2. Apa arti MUX 16 ke 1 dan MUX 3-bit, 4 ke 1?

3. Gambarkan MUX 8 ke 1! Tuliskan ekspresi Boole keluarannya dan jelaskan cara kerjanya!
4. Gambarkan MUX 3-bit, 2 ke 1! Tuliskan ekspresi Boole keluarannya dan jelaskan cara kerjanya!
5. Berdasarkan tujuan praktikum, susun rangkaian dan tabel-tabel pengamatan percobaan yang diperlukan untuk menentukan watak:
  - a. MUX 1 bit 4 ke 1;
  - b. MUX 3-bit, 2 ke 1; dan

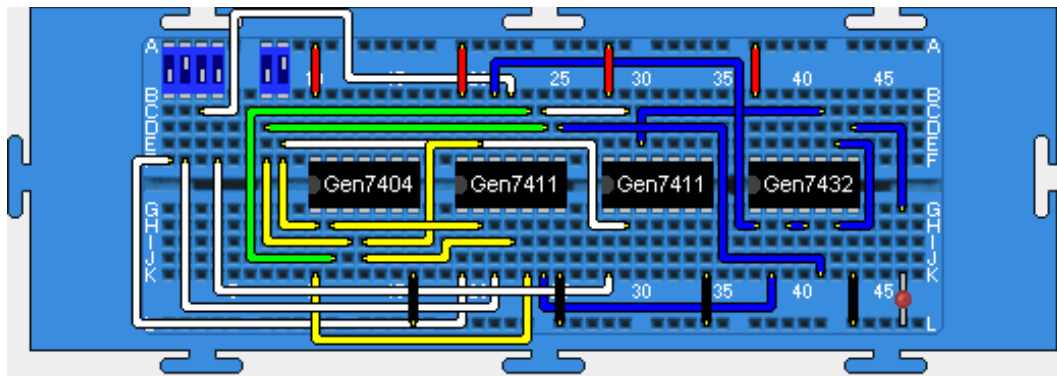
### E. PROSEDUR

1. Menyelidiki watak rangkaian MUX 1-bit 4 ke 1
  - a. Gunakan rangkaian MUX 4 ke 1 berikut ini untuk percobaan.



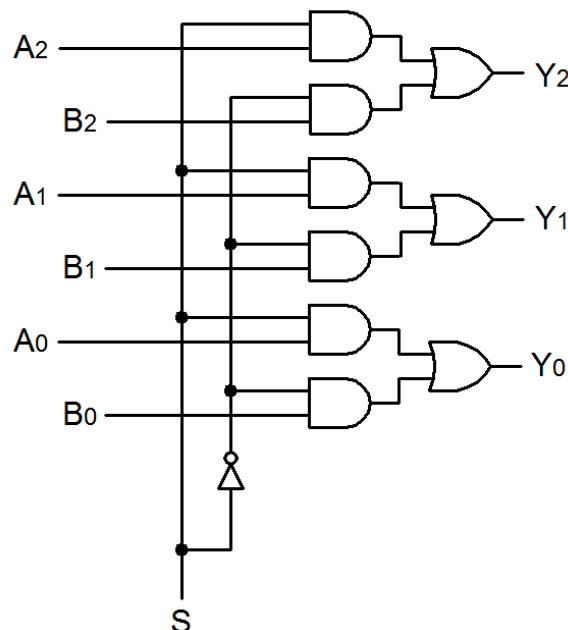
Gambar 26. Multiplexer 4 ke 1

- b. Tempatkan sebuah IC 7404 (NOT), 7432 (OR), dua buah IC 7411 (AND-3 input), 3 buah saklar *double*, dan sebuah LED.
- c. Pasang catu daya pada semua IC yang digunakan, pasang saklar logika pada input rangkaian dan indikator LED pada output rangkaian. Salah satu bentuk susunan rangkaian MUX 4 ke 1 menggunakan simulator *breadboard* ditunjukkan gambar berikut ini.



Gambar 27. MUX 4 ke 1 menggunakan simulator *breadboard*

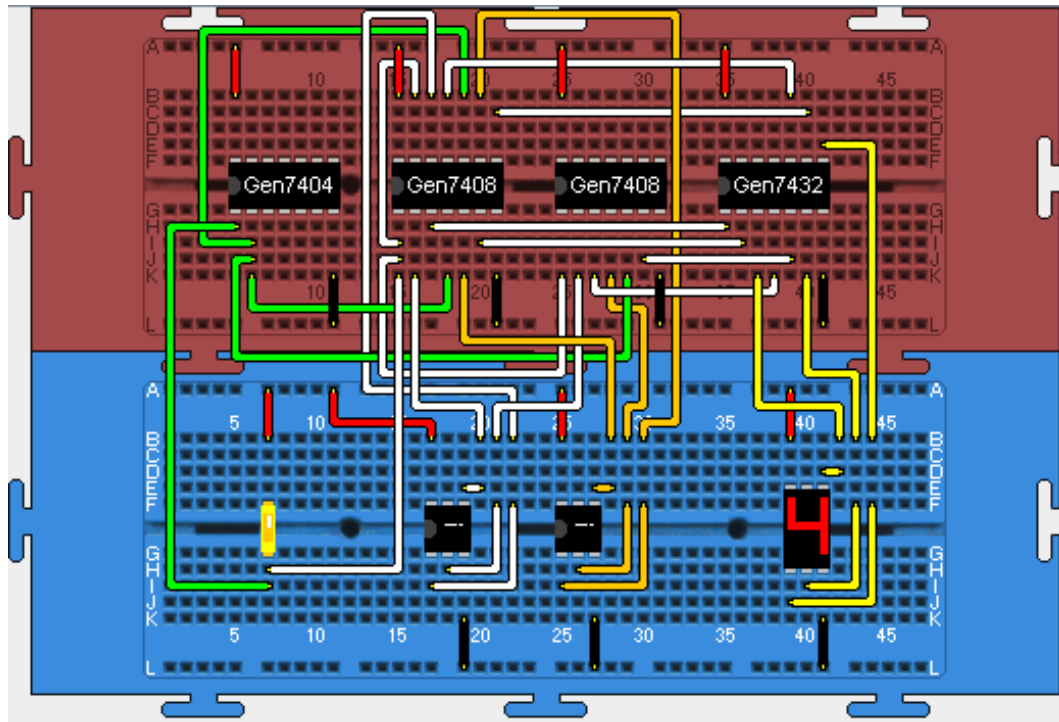
- d. Berikan sinyal kendali  $S_1S_0=00$ , selidiki input mana yang disalurkan ke outputnya. Catat dalam tabel kebenaran dan ulangi untuk sinyal kendali lainnya yakni  $S_1S_0=01$ ,  $S_1S_0=10$  dan  $S_1S_0=11$ .
2. Menyelidiki watak rangkaian MUX 3-bit 2 ke 1
  - a. Gunakan rangkaian MUX 3-bit 2 ke 1 berikut ini untuk percobaan.



Gambar 28. Rangkaian MUX 3 bit 2 ke 1

- b. Tempatkan sebuah IC 7404 (NOT), 7432 (OR), dua buah IC 7408 (AND), sebuah saklar tunggal, dua buah *keypad* dan sebuah peraga 7-segmen di atas *breadboard*.
- c. Pasang catu daya pada semua IC yang digunakan, pasang saklar logika pada input pemilih (S), pasang *keypad* pada input rangkaian dan peraga

7-segmen LED pada output rangkaian. Salah satu bentuk susunan rangkaian MUX 3-bit 2 ke 1 menggunakan simulator *breadboard* ditunjukkan gambar berikut ini.



Gambar 29. Rangkaian MUX 3 bit 2 ke 1 menggunakan simulator *breadboard*

- d. Masukkan angka yang berbeda melalui *keypad* A dan *keypad* B, misalnya A=3 dan B=5. Berikan sinyal kendali S=0 dan amati output rangkaian serta catat hasilnya ke dalam tabel pengamatan. Ulangi untuk S=1 dan data A serta B lainnya.

## F. TUGAS LAPORAN

Buatlah tugas laporan praktik dengan sistematika: (1) tujuan, (2) alat dan bahan, (3) langkah percobaan, (4) hasil dan pembahasan, dan (5) kesimpulan. Bagian tujuan, alat dan bahan diuraikan sesuai dengan uraian pada panduan ini. Uraian tentang langkah percobaan diambil dari panduan ini dan pengalaman anda dalam melaksanakan percobaan. Hasil dan pembahasan berisi penyajian data yang



diperoleh melalui percobaan baik yang bersifat kuantitatif maupun kualitatif, perhitungan sesuai tujuan praktik, dan diskusi terhadap hasil analisis data.

Kesimpulan merupakan rangkuman dari hasil pembahasan. Dalam percobaan ini pembahasan diarahkan untuk menjawab pertanyaan-pertanyaan sebagai berikut: Berdasarkan hasil pengamatan pada watak MUX 4 ke 1, bagaimana pengaruh sinyal kendali atau sinyal pemilih terhadap keluaran MUX? Jadi, apa fungsi rangkaian MUX?

## PERCOBAAN V DEMUPLEKSER

### A. DESKRIPSI

Demultiplekser (biasanya disebut dengan DEMUX saja yakni singkatan dari istilah aslinya *demultiplexer*) termasuk rangkaian logika kombinasi. Fungsi DEMUX kebalikan dari fungsi MUX. Jika rangkaian MUX berfungsi memilih data pada input suatu rangkaian digital, maka DEMUX berfungsi menyalurkan suatu sinyal masukan ke salah satu dari beberapa saluran keluarannya, sehingga disebut pula sebagai penyalur data (data distributor).

Melalui percobaan ini Anda akan menyelidiki watak DEMUX sederhana yang dibangun dengan menggunakan gerbang logika dasar.

### B. TUJUAN

Melalui percobaan ini diharapkan anda dapat menentukan:

1. Watak DEMUX 1 ke 4 yang dibangun dari gerbang logika dasar
2. Kerusakan saluran output DEMUX 1 ke 4 yang dibangun dari IC 74139

### C. ALAT DAN BAHAN

Alat dan bahan yang diperlukan dalam praktikum ini adalah:

1. IC TTL NOT (7404) 1 buah, AND 3-input (7411) 2 buah dan DEMUX 1 ke-4 (74319) 1 buah,
2. DIP *switch*: tunggal, ganda, 1 buah,
3. Indikator LED 4 buah, serta
4. *Breadboard*, catu daya +5 V dan kabel penghubung (sudah tersedia di dalam simulator).

### D. TUGAS PENDAHULUAN

1. Jelaskan pengertian DEMUX!
2. Apa arti DEMUX 1 ke 4?

3. Gambarkan DEMUX 1 ke 4 menggunakan gerbang logika dasar! Tuliskan ekspresi Boole pada keluarannya, dan jelaskan cara kerjanya dengan menggunakan table kebenaran!
4. Gambarkan DEMUX 1 ke-4 menggunakan IC 74139. Jelaskan fungsi-fungsi dari pin 1 sampai dengan pin 7 dari IC 74139. Jelaskan cara kerjanya dengan menggunakan table kebenaran.
5. Berdasarkan tujuan praktikum, susun rangkaian dan tabel-tabel pengamatan percobaan yang diperlukan untuk menentukan watak DEMUX 1 ke 4 menggunakan gerbang logika dasar maupun IC 74139.
6. Perhatikan tabel kebenaran berikut ini!

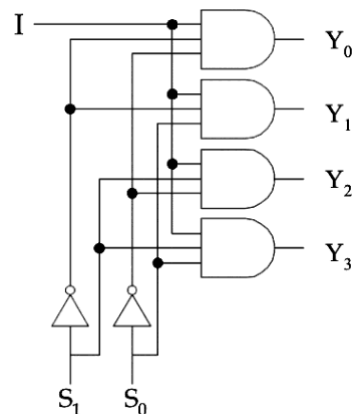
Tabel 2. Tabel kebenaran DEMUX 1-bit 1 ke 4

KENDALI		INPUT	OUTPUT			
$S_1$	$S_0$	$\bar{I}$	$\bar{Y}_0$	$\bar{Y}_1$	$\bar{Y}_2$	$\bar{Y}_3$
0	0	0	0	1	1	1
0	1	0	1	0	1	1
1	0	0	1	0	1	1
1	1	0	1	1	1	0

Apa jenis input dan output dari DEMUX tersebut, ACTIVE-LOW atukah ACTIVE-HIGH? Berdasarkan tabel kebenaran di atas, jelaskan saluran output mana dari DEMUX 1 ke-4 yang mengalami kerusakan.

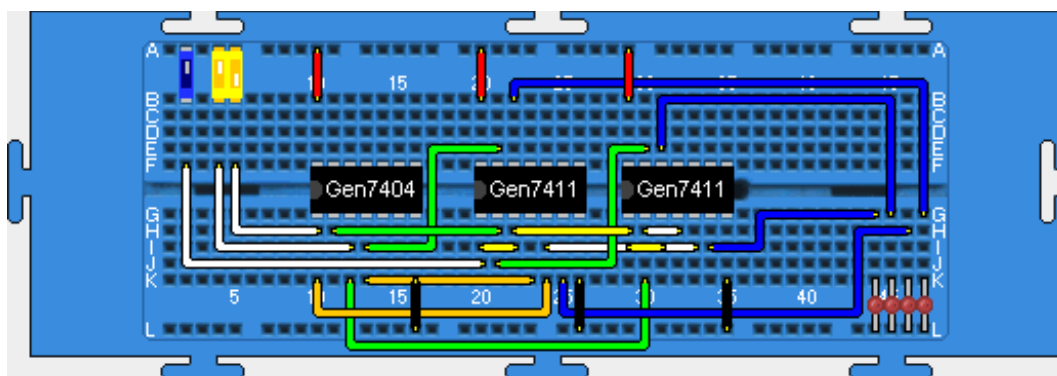
#### E. PROSEDUR

1. Menyelidiki watak rangkaian DEMUX 1 ke 4 yang dibangun dari gerbang logika dasar
  - a. Untuk menyelidiki watak rangkaian DEMUX 1 ke 4, gunakan rangkaian seperti ditunjukkan gambar berikut ini.



Gambar 30. Rangkaian DEMUX 1 ke 4

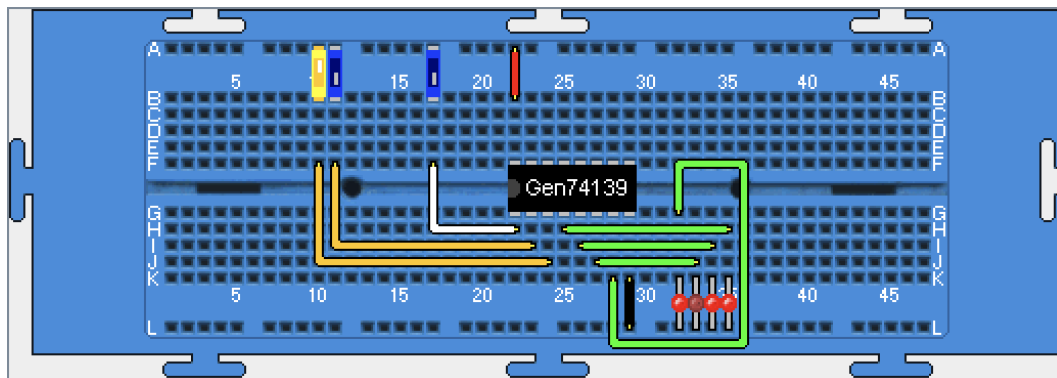
- b. Tempatkan sebuah IC 7404 (NOT), dua buah 7411 (AND-3 input), 1 buah saklar tunggal, 1 buah saklar ganda, dan 4 buah LED di atas *breadboard*.
- c. Pasang catu daya pada semua IC yang digunakan, pasang saklar logika pada input rangkaian (I) dan pemilih ( $S_1S_0$ ), serta LED pada output rangkaian. Salah satu bentuk susunan rangkaian DEMUX 1 ke 4 menggunakan simulator *breadboard* ditunjukkan gambar berikut ini.



Gambar 31. DEMUX 1 ke 4 menggunakan gerbang logika dasar

- d. Berikan input  $I=1$ , dan sinyal pengendali  $S_1S_0=00$ , amati keempat LED pada output rangkaian dan catat hasilnya dalam tabel kebenaran. Ulangi percobaan untuk sinyal pengendali lainnya yakni  $S_1S_0=01$ ,  $S_1S_0=10$  dan  $S_1S_0=11$ .

2. Menentukan kerusakan saluran output DEMUX 1 ke 4 yang dibangun dari IC 74139
  - a. Untuk menyelidiki watak rangkaian DEMUX 1 ke 4 dengan IC 74139, gunakan rangkaian seperti ditunjukkan gambar berikut ini.



Gambar 31b. DEMUX 1 ke 4 menggunakan IC74139

- b. Berikan input  $\overline{1G} = 0$ , dan sinyal pengendali  $\overline{1B} \overline{1A} = 00$ , amati keempat LED pada output rangkaian dan catat hasilnya dalam tabel kebenaran. Ulangi percobaan untuk sinyal pengendali lainnya yakni 01, 10 dan =11

## F. TUGAS LAPORAN

Buatlah tugas laporan praktik dengan sistematika: (1) tujuan, (2) alat dan bahan, (3) langkah percobaan, (4) hasil dan pembahasan, dan (5) kesimpulan. Bagian tujuan, alat dan bahan diuraikan sesuai dengan uraian pada panduan ini. Uraian tentang langkah percobaan diambil dari panduan ini dan pengalaman anda dalam melaksanakan percobaan. Hasil dan pembahasan berisi penyajian data yang diperoleh melalui percobaan baik yang bersifat kuantitatif maupun kualitatif, perhitungan sesuai tujuan praktik, dan diskusi terhadap hasil analisis data.

Kesimpulan merupakan rangkuman dari hasil pembahasan. Dalam percobaan ini pembahasan diarahkan untuk menjawab pertanyaan-pertanyaan sebagai berikut:

1. Berdasarkan tabel pengamatan pada karakteristik DEMUX, bagaimana pengaruh sinyal pemilih terhadap keluaran DEMUX? Jadi, apa fungsi rangkaian DEMUX?
2. Berdasarkan tabel kebenaran yang diperoleh melalui eksperimen, jelaskan saluran output mana dari DEMUX 1 ke-4 yang mengalami kerusakan. Apa jenis input dan output dari DEMUX 74139, ACTIVE-LOW atukah ACTIVE-HIGH?

## PERCOBAAN VI ENKODER

### A. DESKRIPSI

Dalam suatu sistem digital, informasi atau data disajikan dalam bentuk kode biner. Untuk mengkode suatu informasi, misalnya dalam bentuk angka desimal menjadi kode biner dilakukan dengan piranti enkoder

Salah satu jenis sistem kode yang digunakan dalam sistem digital adalah sistem desimal dikode biner (*binary-coded decimal*) atau disingkat BCD. Dalam sistem kode ini, masing-masing digit angka desimal diganti dengan suatu kombinasi 4-bit biner. Salah satu dari jenis kode BCD adalah desimal dikode biner asli atau BCD 8421. Dalam sistem BCD 8421 ini angka 3 desimal akan dikode sebagai 0011, sedangkan angka 264 desimal akan dikode sebagai 0010 0110 0100. Untuk mengkode angka desimal tersebut menjadi kode BCD 8421 digunakan enkoder desimal ke BCD. Pada percobaan ini Anda akan menyelidiki watak enkoder desimal ke BCD.

### B. TUJUAN

Melalui percobaan ini diharapkan Anda dapat merancang dan menentukan watak enkoder desimal ke BCD

### C. ALAT DAN BAHAN

1. IC TTL OR-2-input (7432) 3 buah
2. DIP *switch*: tunggal 2 buah dan *quad* 2 buah dan indikator LED 4 buah.
3. *Breadboard*, catu daya +5 V dan kabel penghubung (sudah tersedia di dalam simulator).

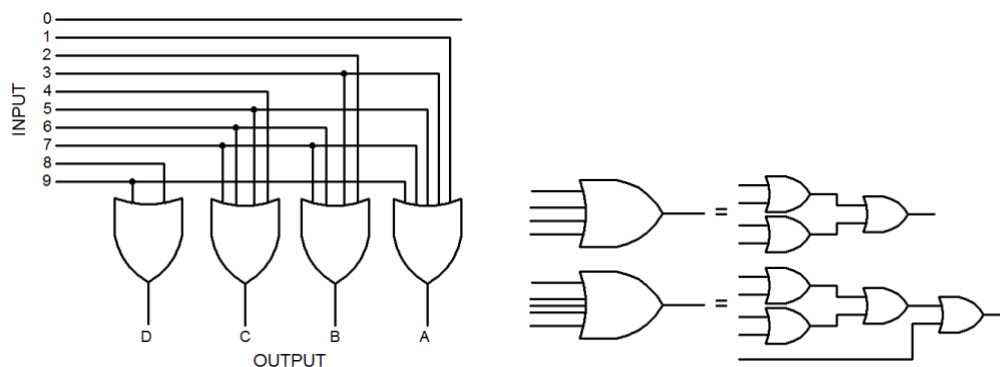
### D. TUGAS PENDAHULUAN

1. Jelaskan pengertian enkoder dalam sistem digital!
2. Ubahlah bilangan desimal 5, 18, dan 379 ke dalam kode BCD 8421!

3. Gambarkan simbol dan susunlah tabel kebenaran dari enkoder desimal ke BCD untuk
  - a. Jenis input dan outputnya ACTIVE-HIGH!
  - b. Jenis input dan outputnya ACTIVE-LOW!
4. Tulislah ekspresi Boole dari setiap keluaran enkoder pada soal nomor 3 untuk jenis input dan outputnya ACTIVE-HIGH!
5. Berdasarkan ekspresi Boole yang diperoleh pada soal nomor 4, susunlah rangkaian enkoder desimal ke BCD!
6. Berdasarkan tujuan praktikum, susun rangkaian dan tabel-tabel pengamatan percobaan yang diperlukan untuk menentukan watak enkoder desimal ke BCD!

#### E. PROSEDUR

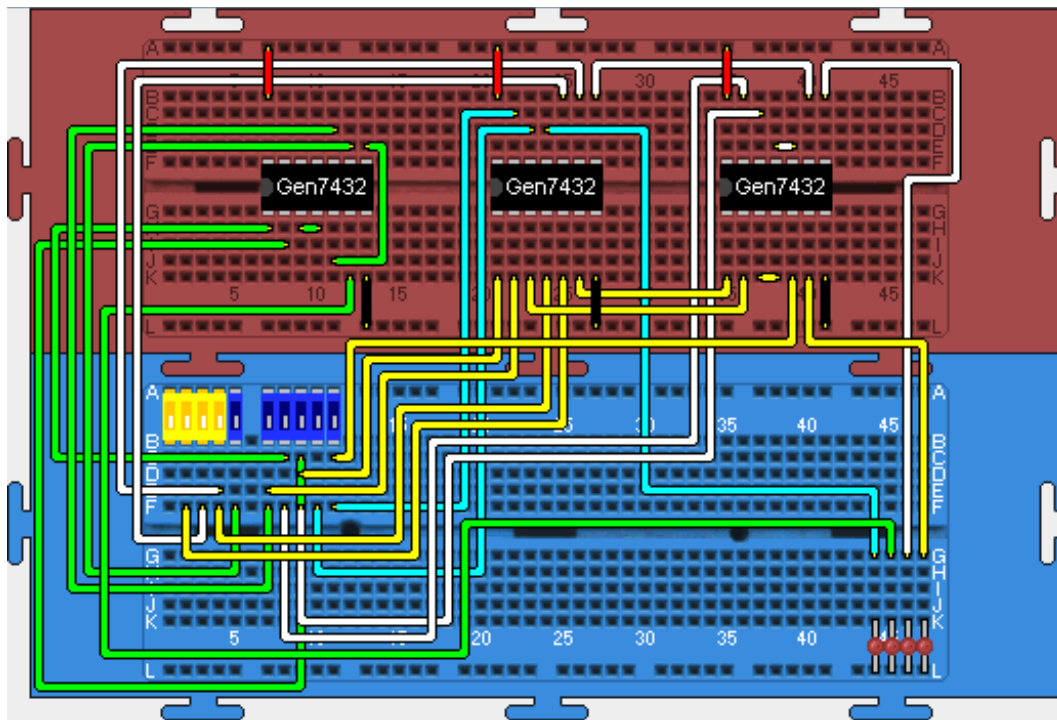
1. Gunakan rangkaian enkoder desimal ke BCD berikut ini untuk percobaan.
2. Tempatkan tiga sebuah IC 7432 (OR), dua buah saklar ganda, dua buah saklar *triple* dan empat buah LED di atas *breadboard*.



Gambar 32. Rangkaian: (a) Enkoder desimal ke BCD; (b) konversi gerbang OR

3. Pasang catu daya pada semua *chip* yang digunakan, pasang saklar logika pada input rangkaian dan indikator LED pada output rangkaian. Salah satu bentuk susunan rangkaian enkoder desimal ke BCD menggunakan simulator *breadboard* ditunjukkan gambar berikut ini.





Gambar 33. Rangkaian enkoder desimal ke BCD menggunakan simulator *breadboard*

4. ON-kan saklar paling kanan (mewakili bilangan 9 desimal), saklar lainnya OFF. Amati kode yang dibangkitkan pada outputnya, catat ke dalam tabel pengamatan. Ulangi untuk 9 saklar input lainnya yang mewakili bilangan desimal 0 s.d. 8. Saklar paling kiri mewakili bilangan desimal 0.

#### F. TUGAS LAPORAN

Buatlah tugas laporan praktik dengan sistematika: (1) tujuan, (2) alat dan bahan, (3) langkah percobaan, (4) hasil dan pembahasan, dan (5) kesimpulan. Bagian tujuan, alat dan bahan diuraikan sesuai dengan uraian pada panduan ini. Uraian tentang langkah percobaan diambil dari panduan ini dan pengalaman anda dalam melaksanakan percobaan. Hasil dan pembahasan berisi penyajian data yang diperoleh melalui percobaan baik yang bersifat kuantitatif maupun kualitatif, perhitungan sesuai tujuan praktik, dan diskusi terhadap hasil analisis data.

Kesimpulan merupakan rangkuman dari hasil pembahasan. Dalam percobaan ini pembahasan diarahkan untuk menjawab pertanyaan-pertanyaan sebagai berikut:

1. Bandingkan tabel kebenaran eksperimen enkoder desimal ke BCD dengan tabel kebenaran teoritiknya! Apakah rangkaian enkoder anda telah bekerja dengan baik? Jadi, apa fungsi enkoder?
2. Berdasarkan tabel kebenarannya, kemukakan alasan mengapa enkoder tersebut termasuk jenis ACTIVE-HIGH!
3. Apa arti ACTIVE-LOW dan ACTIVE-HIGH dari suatu saluran input atau output rangkaian logika?

## PERCOBAAN VII DEKODER

### A. DESKRIPSI

Dekoder digunakan untuk mengenal kembali atau mengkode ulang kode-kode biner yang dihasilkan oleh enkoder. Jika enkoder sistem BCD 8421 menghasilkan kode-kode biner dari angka desimal, maka dekoder akan menghasilkan kembali angka desimal dari kode biner yang dimasukkan ke inputnya. Pada percobaan ini Anda akan menyelidiki watak dekoder BCD ke desimal.

### B. TUJUAN

Melalui percobaan ini diharapkan Anda dapat merancang dan menentukan watak dekoder BCD ke desimal

### C. ALAT DAN BAHAN

1. IC Gerbang AND-4 input (7421) 5 buah, dan NOT (7404) 1 buah.
2. DIP switch: *quad* 1 buah dan indikator LED 10 buah.
3. Breadboard, catu daya +5 V dan kabel penghubung (sudah tersedia di dalam simulator).

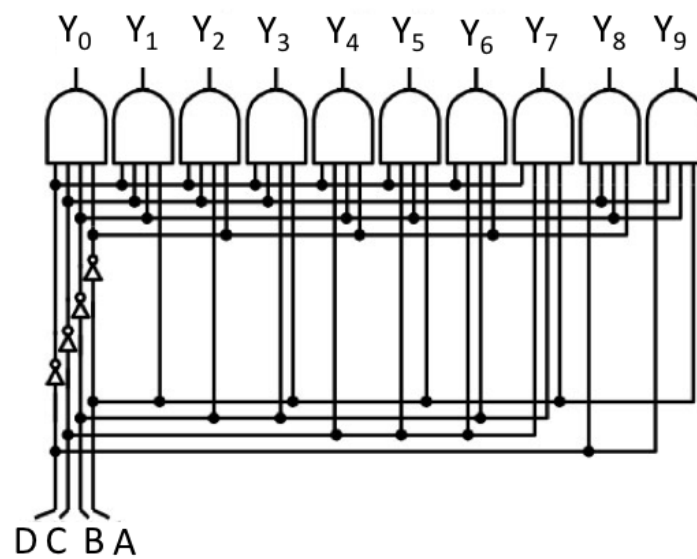
### D. TUGAS PENDAHULUAN

1. Jelaskan pengertian dekoder!
2. Ubahlah kode BCD 8421: 1001, 0110 0100 0010, 0011 0101 ke dalam angka desimal!
3. Gambarkan simbol dan susunlah tabel kebenaran dekoder BCD ke desimal untuk jenis outputnya ACTIVE-HIGH maupun ACTIVE-LOW!
4. Tulislah ekspresi Boole dari setiap keluaran dekoder BCD ke desimal untuk outputnya ACTIVE-HIGH!
5. Berdasarkan ekspresi Boole yang diperoleh pada jawaban soal nomor 9, susunlah rangkaian dekoder BCD ke desimal!

6. Berdasarkan tujuan praktikum, susun rangkaian dan tabel-tabel pengamatan percobaan yang diperlukan untuk menentukan watak dekoder BCD ke desimal!

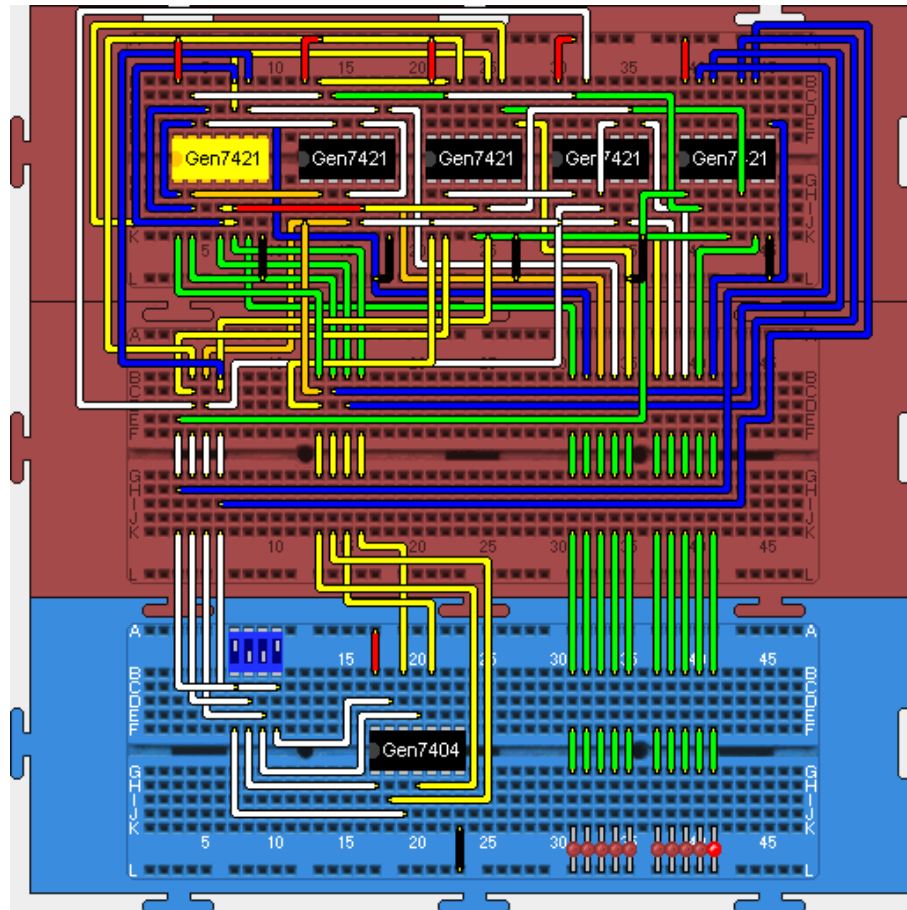
### E. PROSEDUR

1. Gunakan rangkaian pada gambar berikut ini untuk percobaan



Gambar 34. Decoder BCD ke decimal

- Susun rangkaian seperti berikut ini di atas *breadboard*.



Gambar 35. Decoder BCD ke desimal menggunakan simulator breadboard

- Pada rangkaian dekoder BCD ke desimal di atas, saklar input mewakili kode BCD dan indikator LED pada output mewakili bilangan desimal 0 sampai dengan 9 (LED paling kiri menunjukkan angka 0 dan paling kanan angka 9).
- Berikan input kode 1001 melalui saklar input, amati output rangkaian dan catat hasilnya pada tabel pengamatan. Ulangi untuk semua kode BCD yang mungkin.

## F. TUGAS LAPORAN

Buatlah tugas laporan praktik dengan sistematika: (1) tujuan, (2) alat dan bahan, (3) langkah percobaan, (4) hasil dan pembahasan, dan (5) kesimpulan. Bagian tujuan, alat dan bahan diuraikan sesuai dengan uraian pada panduan ini.

Uraian tentang langkah percobaan diambil dari panduan ini dan pengalaman anda dalam melaksanakan percobaan. Hasil dan pembahasan berisi penyajian data yang diperoleh melalui percobaan baik yang bersifat kuantitatif maupun kualitatif, perhitungan sesuai tujuan praktik, dan diskusi terhadap hasil analisis data.

Kesimpulan merupakan rangkuman dari hasil pembahasan. Dalam percobaan ini pembahasan diarahkan untuk menjawab pertanyaan-pertanyaan sebagai berikut:

1. Bandingkan tabel kebenaran eksperimen dekoder BCD ke desimal dengan tabel kebenaran teoritiknya! Apakah rangkaian dekoder anda telah bekerja dengan baik? Jadi, apa fungsi dekoder?
2. Berdasarkan tabel kebenarannya, kemukakan alasan mengapa enkoder tersebut termasuk jenis ACTIVE-HIGH!
3. Apa arti ACTIVE-LOW dan ACTIVE-HIGH dari suatu saluran input atau output rangkaian logika?

## PERCOBAAN VIII FLIP-FLOP

### A. DESKRIPSI

Flip-flop merupakan elemen rangkaian logika sekuensial, yakni rangkaian yang keluarannya tergantung pada keluaran sebelumnya. Karena sifat-sifatnya, flip-flop disebut juga pengingat (memory) 1-bit. Sekumpulan flip-flop dapat membentuk sel pengingat beberapa bit. Flip-flop memegang peranan yang sangat penting di dalam perancangan rangkaian logika sekuensial. Dari flip-flop dapat dibangun rangkaian register yang merupakan pengingat dengan banyak bit, dan rangkaian pencacah (counter). Rangkaian-rangkaian tersebut merupakan dasar bagi perancangan perangkat keras komputer.

Pada percobaan ini Anda akan menyelidiki watak berbagai macam flip-flop yakni jenis flip-flop Set-Reset (FFSR), flip-flop JK (FFJK), flip-flop D (FFD), dan flip-flop T (FFT). Anda juga akan menyelidiki watak rangkaian *toggle* sebagai elemen pencacah, yang dibangun dari FFJK, FFD dan FFT.

### B. TUJUAN

Melalui percobaan ini diharapkan Anda dapat:

1. menentukan watak FFSR, FFJK, FFD dan FFT; serta
2. merancang dan menentukan watak rangkaian *toggle* menggunakan FFJK, FFD dan FFT.

### C. ALAT DAN BAHAN

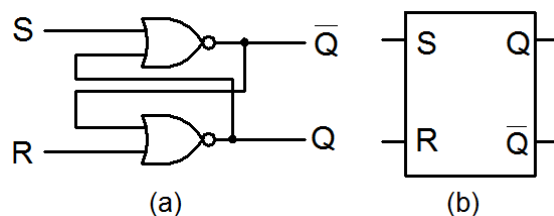
1. IC TTL AND (7408), NOR (7428), NAND (7400), FFJK (7473) dan FFD (7474) masing-masing sebuah
2. DIP *switch*: tunggal dan ganda masing-masing 2 buah dan pembangkit *clock*
3. Indikator LED 2 buah
4. *Breadboard*, catu daya +5 V dan kabel penghubung (sudah tersedia di dalam simulator).

#### D. TUGAS PENDAHULUAN

1. Berikan pengertian flip-flop! Sebutkan jenis-jenisnya dan gambarkan simbolnya!
2. Susun tabel keadaan FFSR dan jelaskan cara kerjanya!
3. Gambarkan rangkaian dan simbol FFJK! Susun pula tabel kebenarannya!
4. Gambarkan rangkaian dan simbol FFD serta tuliskan tabel kebenarannya!
5. Gambarkan rangkaian dan simbol FFT serta susun tabel keadaannya!
6. Apa yang dimaksud dengan keadaan *toggle* dan susunlah rangkaian menggunakan FFJK, FFD, dan FFT untuk menampilkan keadaan *toggle*!
7. Berdasarkan tujuan, alat dan bahan percobaan, susun rangkaian dan tabel-tabel pengamatan percobaan yang diperlukan untuk:
  - a. menentukan watak FFSR, FFJK, FFD dan FFT; serta
  - b. menentukan watak rangkaian *toggle* menggunakan FFJK, FFD dan FFT.

#### E. PROSEDUR

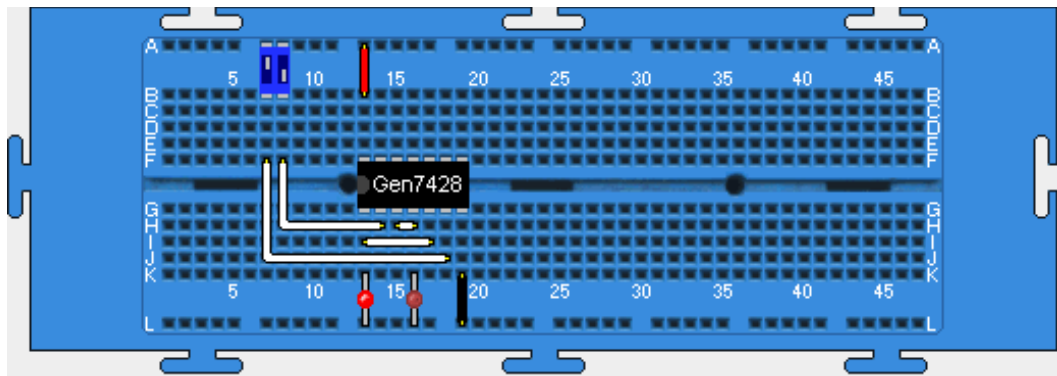
1. Menentukan watak FFSR (Set-Reset)
  - a. Gunakan rangkaian FFSR berikut ini untuk percobaan.



Gambar 36. Flip-flop SR menggunakan gerbang NOR:  
(a) rangkaian; (b) simbol

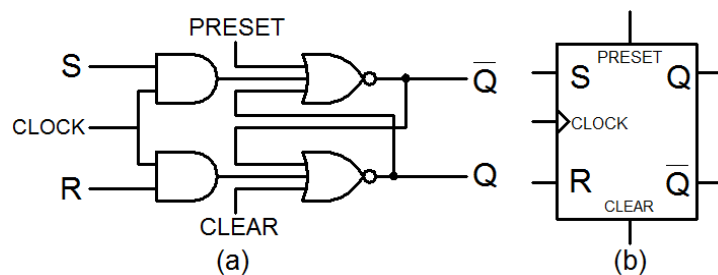
- b. Tempatkan sebuah IC 7428 (NOR), sebuah saklar ganda dan 2 buah LED pada papan *breadboard*.
- c. Pasang catu daya pada IC yang digunakan, pasang saklar logika pada input S dan R dan indikator LED pada output Q dan  $\bar{Q}$ . Salah satu bentuk susunan rangkaian FFSR menggunakan simulator *breadboard* ditunjukkan pada gambar berikut ini.





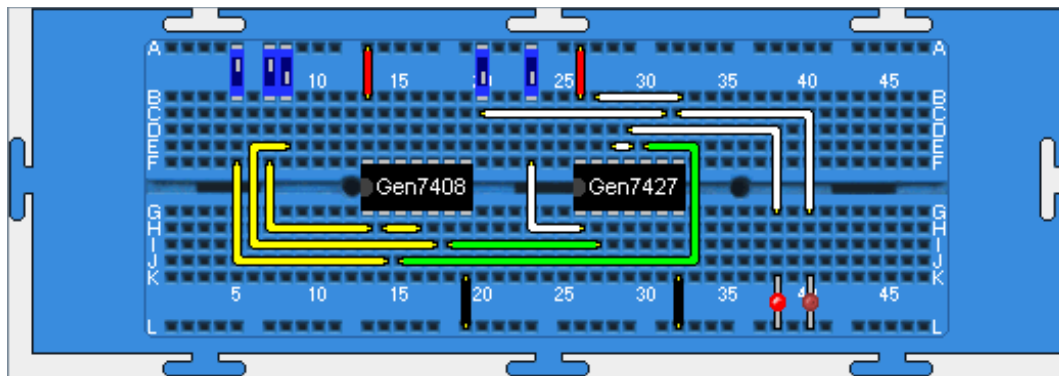
Gambar 37. Susunan FFSR menggunakan simulator *breadboard*

- d. Berikan berbagai kombinasi nilai logika yang mungkin pada input S dan R, dan amati output rangkaian Q dan  $\bar{Q}$  serta catat hasilnya ke dalam tabel pengamatan.
2. Menentukan watak FFSR *clocked*
    - a. Gunakan rangkaian FFSR *clocked* berikut ini untuk percobaan.



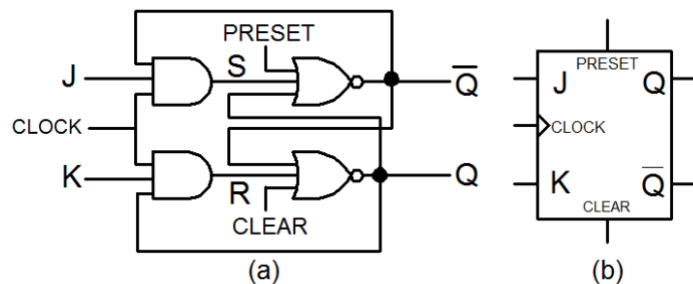
Gambar 38. FFSR *clocked*

- b. Tempatkan sebuah IC 7427 (NOR-3 input), sebuah IC 7408 (AND), 3 buah saklar tunggal, sebuah saklar ganda, dan 2 buah LED pada papan *breadboard*.
- c. Pasang catu daya pada semua IC yang digunakan, pasang saklar logika pada input S, R, *Preset*, *Clear*, dan *clock* serta indikator LED pada output Q dan  $\bar{Q}$ . Salah satu bentuk susunan rangkaian FFSR *clocked* yang dilengkapi *Preset* dan *Clear* menggunakan simulator *breadboard* ditunjukkan pada gambar berikut ini.



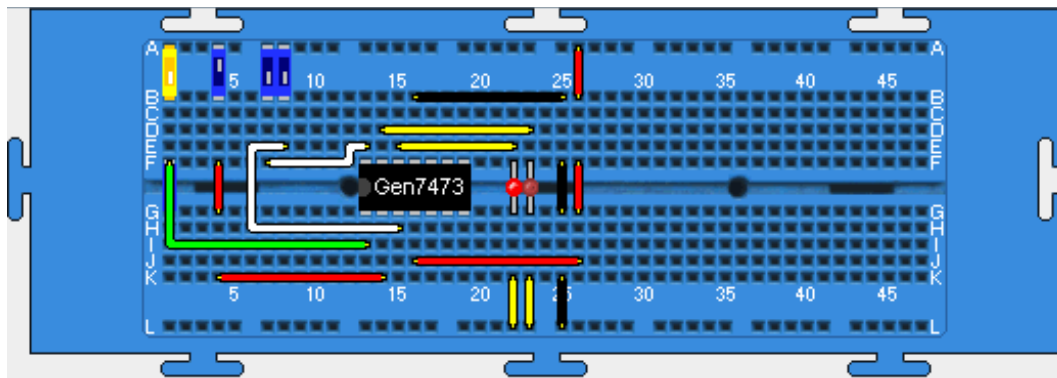
Gambar 39. Susunan FFSR *clocked* menggunakan simulator *breadboard*

- d. Gunakan *Preset* (saklar di-ON-kan, sesaat kemudian di-OFF-kan) untuk memberi nilai  $Q_{n-1}=1$  dan *Clear* untuk  $Q_{n-1}=0$
  - e. Berikan berbagai kombinasi nilai logika yang mungkin pada input S dan R untuk keadaan  $Q_{n-1}=0$  dan  $Q_{n-1}=1$ , dan amati output rangkaian Q dan  $\bar{Q}$  setiap saat sinyal *clock* diberikan serta catat hasilnya ke dalam tabel pengamatan.
3. Menentukan watak FFJK
    - a. Gunakan rangkaian FFJK berikut ini untuk percobaan.



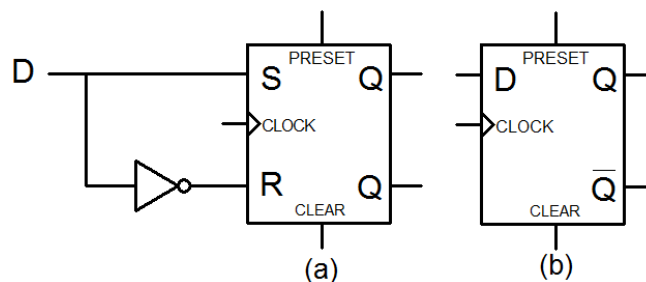
Gambar 40. Flip-flop JK: (a) rangkaian; (b) simbol

- b. Tempatkan sebuah IC 7473 (FFJK), 2 buah saklar tunggal, sebuah saklar ganda, dan 2 buah LED pada papan *breadboard*.
- c. Pasang catu daya pada IC, pasang saklar logika pada input J, K, *Clear* dan *clock* serta indikator LED pada output Q dan  $\bar{Q}$ . Salah satu bentuk susunan rangkaian FFJK menggunakan simulator *breadboard* ditunjukkan pada gambar berikut ini.



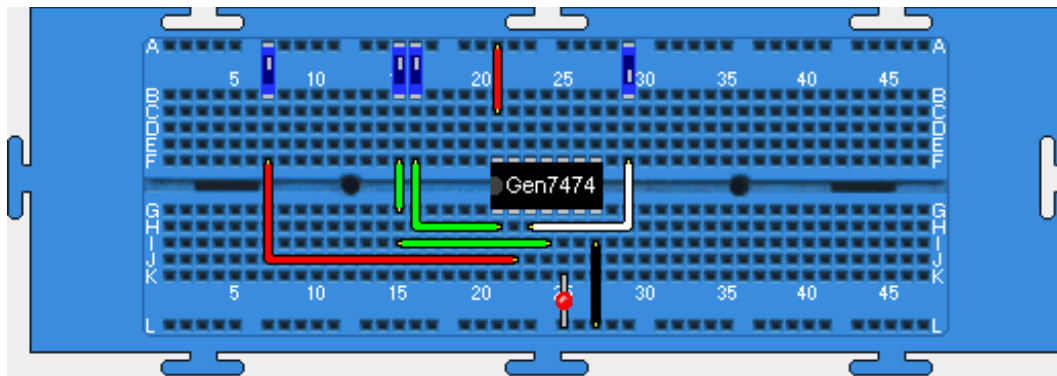
Gambar 41. Susunan rangkaian FFJK menggunakan simulator *breadboard*

- d. Pastikan  $Clear=1$  (saklar ON). Awali percobaan dengan memberikan nilai awal  $Q_{n-1}=0$  dengan meng-OFF-kan saklar *Clear*, sesaat kemudian di-ON-kan. Selanjutnya *Clear* harus selalu bernilai 1 (saklar ON)
  - e. Berikan berbagai kombinasi nilai logika yang mungkin pada input J dan K, dan amati output rangkaian Q dan  $\bar{Q}$  setiap saat sinyal *clock* diberikan serta catat hasilnya ke dalam tabel pengamatan.
4. Menentukan watak FFD
    - a. Gunakan rangkaian FFD berikut ini untuk percobaan.



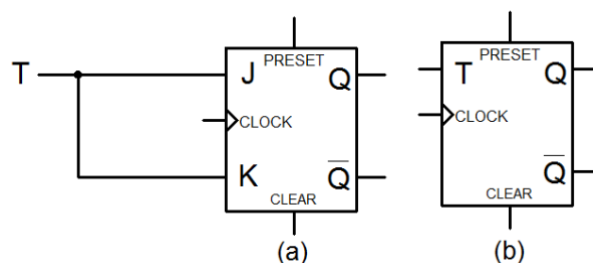
Gambar 42. Flip-flop D: (a) rangkaian; (b) simbol

- b. Tempatkan sebuah IC 7474 (FFD), 2 buah saklar tunggal, 1 buah saklar ganda dan sebuah LED pada papan *breadboard*.
- c. Pasang catu daya pada IC, pasang saklar logika pada input D, Preset, Clear dan *clock* serta indikator LED pada output Q dan  $\bar{Q}$ . Salah satu bentuk susunan rangkaian FFD menggunakan simulator *breadboard* ditunjukkan pada gambar berikut ini.



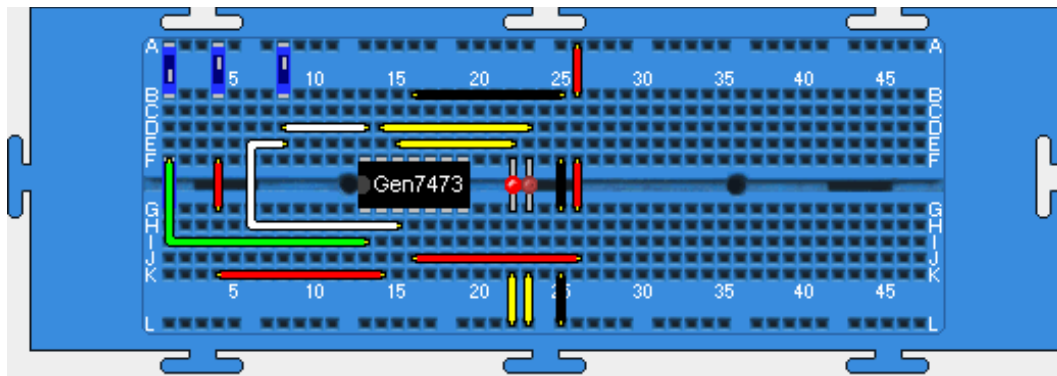
Gambar 43. Susunan Flip-flop D menggunakan simulator *breadboard*

- d. Pastikan saklar Preset dan Clear dalam keadaan ON. Preset dan Clear diaktifkan dengan meng-OFF-kan saklar, sesaat kemudian meng-ON-kannya kembali. Berikan berbagai kombinasi nilai logika yang mungkin pada input D, dan amati output rangkaian Q dan  $\bar{Q}$  setiap saat sinyal *clock* diberikan serta catat hasilnya ke dalam tabel pengamatan.
5. Menentukan watak FFT
  - a. Gunakan rangkaian FFT berikut ini untuk percobaan.



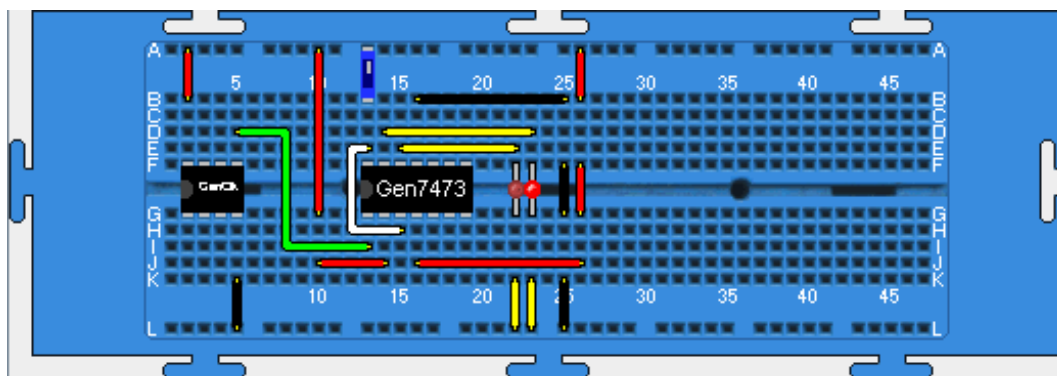
Gambar 44. Flip-flop T: (a) rangkaian; (b) simbol

- b. Tempatkan sebuah IC 7473 (FFJK), 3 buah saklar tunggal, dan dua buah LED pada papan *breadboard*.
- c. Pasang catu daya pada IC yang digunakan, pasang saklar logika pada input T, *Clear* dan *Clock* serta indikator LED pada output Q dan  $\bar{Q}$ . Salah satu bentuk susunan rangkaian FFT menggunakan simulator *breadboard* ditunjukkan pada gambar berikut ini.



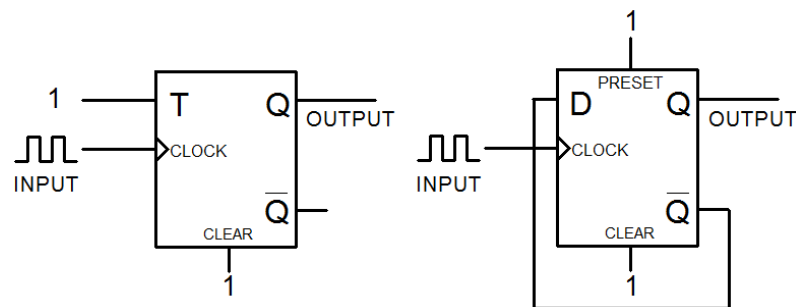
Gambar 45. Susunan Flip-flop T menggunakan simulator *breadboard*

- d. Jalankan simulasi dan pastikan  $Clear=1$  dengan meng-ON-kan saklar *Clear*. Berikan berbagai kombinasi nilai logika 0 dan 1 pada input T, untuk keadaan  $Q_{n-1}=0$  dan  $Q_{n-1}=1$ . Amati output rangkaian Q dan  $\bar{Q}$  setiap saat sinyal *clock* diberikan serta catat hasilnya ke dalam tabel pengamatan.
- e. Ganti saklar *clock* dengan generator *clock* sehingga rangkaian menjadi seperti berikut ini.



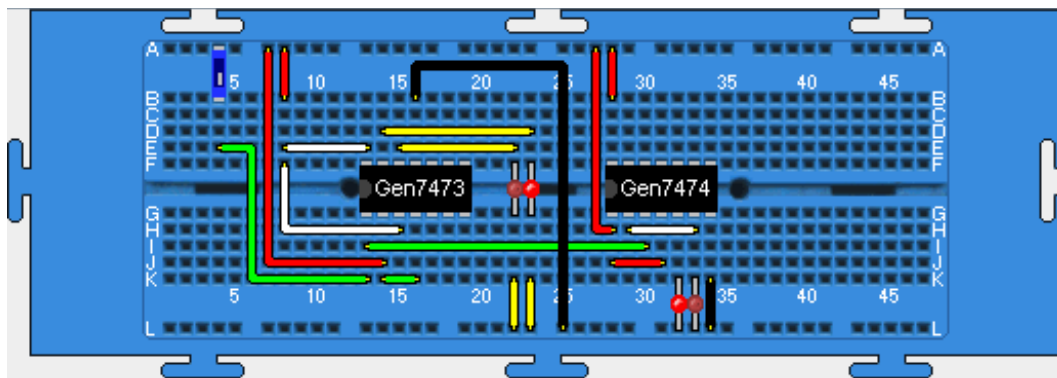
Gambar 46. Flip-flop T dengan input generator *clock*

- f. Jalankan simulasi, berikan input  $J=K=1$  dan atur **kecepatan simulasi (Sim Speed) hingga level 8,5**, amati output rangkaian! Catat keadaan outputnya!
6. Merancang dan menentukan watak rangkaian *toggle*
- a. Gunakan rangkaian *toggle* menggunakan FFT dan FFD berikut ini untuk percobaan.



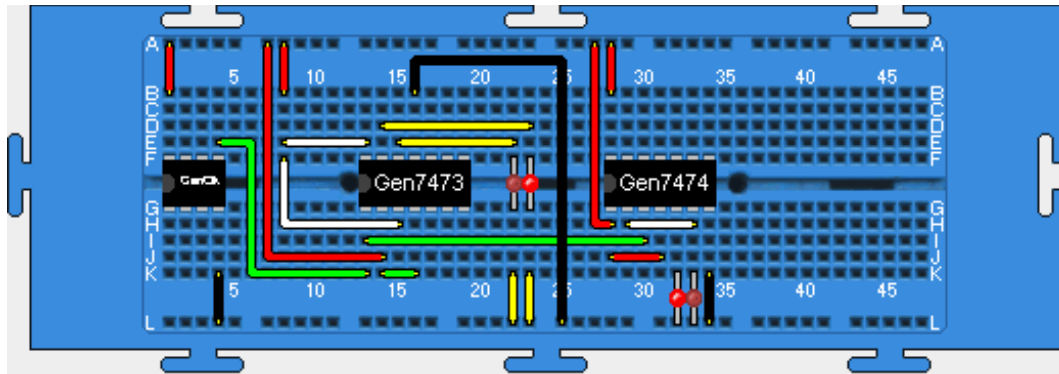
Gambar 47. Rangkaian *Toggle* dengan FFT dan FFD

- b. Tempatkan sebuah IC 7473 (FFJK), sebuah IC 7474 (FFD), sebuah saklar tunggal, dan empat buah LED pada papan *breadboard*.
- c. Pasang catu daya pada semua IC yang digunakan, pasang saklar logika pada input dan *clock* serta indikator LED pada output Q dan  $\bar{Q}$ . Berikan nilai logika 1 pada input *clear* dan *preset*. Salah satu bentuk susunan rangkaian *toggle* dengan FFT dan FFD menggunakan simulator *breadboard* ditunjukkan pada gambar berikut ini.



Gambar 48. *Toggle* dengan FFT dan FFD menggunakan simulator *breadboard*

- d. Berikan sinyal *clock* beberapa kali dengan meng-ON-kan saklar, sesaat kemudian meng-OFF-kannya kembali, dan amati output rangkaian Q dan  $\bar{Q}$  setiap saat sinyal *clock* diberikan serta catat hasilnya ke dalam tabel pengamatan.
- e. Selanjutnya, ganti saklar *clock* dengan generator *clock* sehingga rangkaian menjadi:



Gambar 49. *Toggle* dengan FFT dan FFD menggunakan input generator *clock*

- f. Jalankan simulasi dan atur **kecepatan simulasi (Sim Speed)** hingga level **8,5**, amati output rangkaian! Catat keadaan outputnya!

## F. TUGAS LAPORAN

Buatlah tugas laporan praktik dengan sistematika: (1) tujuan, (2) alat dan bahan, (3) langkah percobaan, (4) hasil dan pembahasan, dan (5) kesimpulan. Bagian tujuan, alat dan bahan diuraikan sesuai dengan uraian pada panduan ini. Uraian tentang langkah percobaan diambil dari panduan ini dan pengalaman anda dalam melaksanakan percobaan. Hasil dan pembahasan berisi penyajian data yang diperoleh melalui percobaan baik yang bersifat kuantitatif maupun kualitatif, perhitungan sesuai tujuan praktik, dan diskusi terhadap hasil analisis data.

Kesimpulan merupakan rangkuman dari hasil pembahasan. Dalam percobaan ini pembahasan diarahkan untuk menjawab pertanyaan-pertanyaan sebagai berikut:

1. Berdasarkan tabel keadaannya, apa syarat keadaan SET dan RESET pada FFSR?
2. Pada kondisi input S dan R seperti apa FFSR dalam keadaan terlarang? Apa maksud keadaan terlarang tersebut?
3. Apa perbedaan antara FFSR dan FSSR *clocked*? Apa fungsi sinyal *clock* pada FFSR *clocked*?
4. Atas dasar tabel keadaannya, tunjukkan persamaan dan perbedaan antara FFSR dengan FFJK!
5. Dari tabel keadaan FFD, tunjukkan bahwa tabel keadaan tersebut merupakan bentuk khusus dari tabel keadaan FFSR!
6. Tunjukkan pula bahwa tabel keadaan FFT merupakan keadaan khusus dari tabel keadaan FFJK!
7. Berdasarkan tabel kebenarannya, tunjukkan bahwa FFJK, FFD, dan FFT dapat digunakan untuk menampilkan keadaan *toggle*!

## PERCOBAAN IX PENCACAH

### A. DESKRIPSI

Pencacah (*counter*) merupakan salah satu rangkaian logika sekuensial, dan dibangun dengan menggunakan beberapa flip-flop. Sebuah flip-flop mempunyai 2 keadaan yakni keadaan 0 (RESET) dan keadaan 1 (SET), sehingga sederetan  $n$  buah flip-flop mempunyai  $2^n$  keadaan yang berbeda. Di dalam penggunaannya sebagai pencacah pulsa, setiap satu keadaan dari  $2^n$  keadaan, digunakan untuk menyatakan jumlah pulsa yang masuk pencacah. Dengan demikian hubungan antara flip-flop yang satu dengan yang lain sedemikian rupa sehingga keadaannya akan berubah secara berurutan setiap kali ada pulsa masuk.

Pencacah terdiri atas *asynchronous counter* (pencacah tak sinkron) dan *synchronous counter* (pencacah sinkron). Melalui percobaan ini Anda akan merancang dan menyelidiki watak pencacah khususnya jenis asinkron.

### B. TUJUAN

Melalui percobaan ini diharapkan Anda dapat:

1. merancang dan menentukan watak rangkaian pencacah menggunakan FFJK dan FFD; serta
2. merancang dan menentukan watak rangkaian pencacah menggunakan IC TTL 74393.

### C. ALAT DAN BAHAN

1. IC TTL 7473 (FFJK), 7474 (FFD) masing-masing 2 buah, dan 74393 (*Binary Counter*), 7408 (AND) masing-masing 1 buah
2. Pembangkit *clock* 1 buah, indikator LED 5 buah dan peraga heksadesimal 1 buah
3. *Breadboard*, catu daya +5 V dan kabel penghubung (sudah tersedia di dalam simulator).

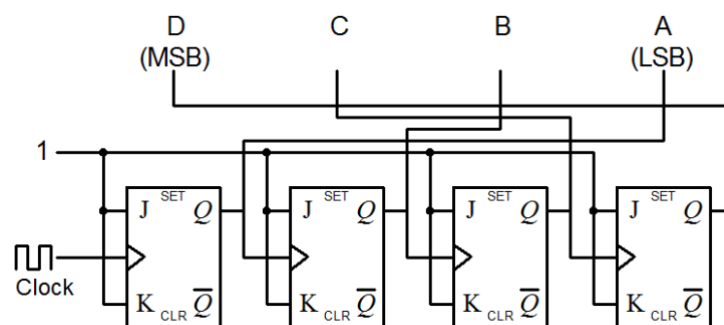


#### D. TUGAS PENDAHULUAN

1. Kemukakan definisi *asynchronous counter* (pencacah tak sinkron) dan *synchronous counter* (pencacah sinkron)!
2. Apa yang dimaksud dengan pencacah modulo-16, modulo-12, dan modulo-6!
3. Susun tabel keadaan pencacah modulo-16, modulo-12, dan modulo-6!
4. Atas dasar tabel keadaannya, rancanglah pencacah tak sinkron modulo-16, menggunakan (a) flip-flop JK, (b) flip-flop T, dan (b) flip-flop D!
5. Rancang pencacah tak sinkron modulo-12 dan modulo-6 menggunakan FFJK!
6. Gambarkan *pinout* IC 74393!
7. Susunlah rangkaian pencacah modulo-16, modulo-12, dan modulo-6 dengan IC 74393!
8. Berdasarkan tujuan praktikum yang ingin dicapai dan alat serta bahan yang disediakan, susun prosedur, rangkaian dan tabel-tabel pengamatan percobaan yang diperlukan untuk:
  - a. merancang dan menentukan watak rangkaian pencacah modulo-16 menggunakan FFJK dan FFD.
  - b. Merancang dan menentukan watak rangkaian pencacah modulo-16, modulo-12, dan modulo-6 menggunakan IC TTL 74393.

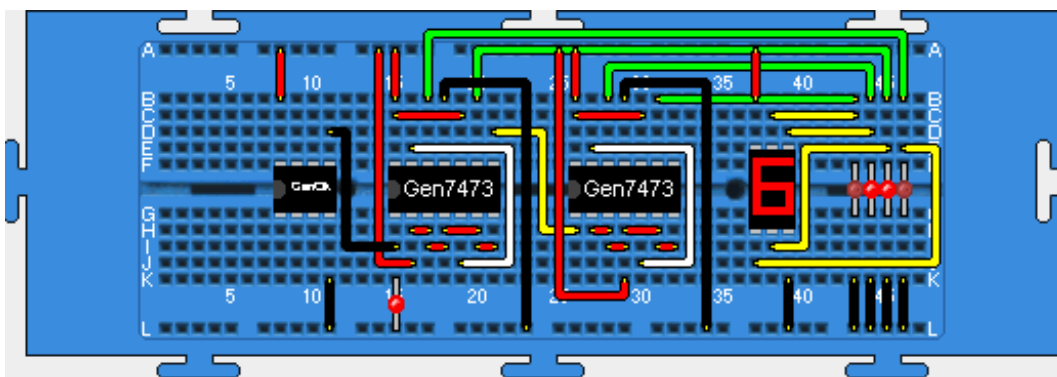
#### E. PROSEDUR

1. Merancang dan menentukan watak rangkaian pencacah modulo-16 menggunakan FFJK dan FFD
  - a. Gunakan rangkaian pencacah modulo-16 menggunakan FFJK berikut ini untuk percobaan.



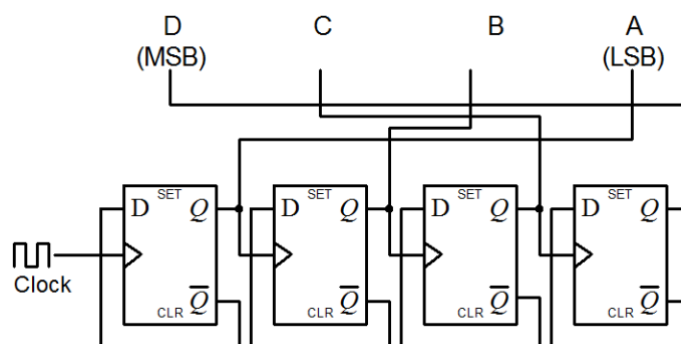
Gambar 50. Pencacah Modulo-16 menggunakan FFJK

- b. Tempatkan dua buah IC 7473 (FFJK), sebuah generator *clock*, sebuah peraga 7-segmen, dan lima buah LED pada papan *breadboard*.
- c. Pasang catu daya pada semua IC yang digunakan, pasang generator *clock* dan LED pada input, serta pasang peraga heksadesimal pada output pencacah. Salah satu bentuk susunan rangkaian pencacah modulo-16 dengan FFJK menggunakan simulator *breadboard* ditunjukkan pada gambar berikut ini.



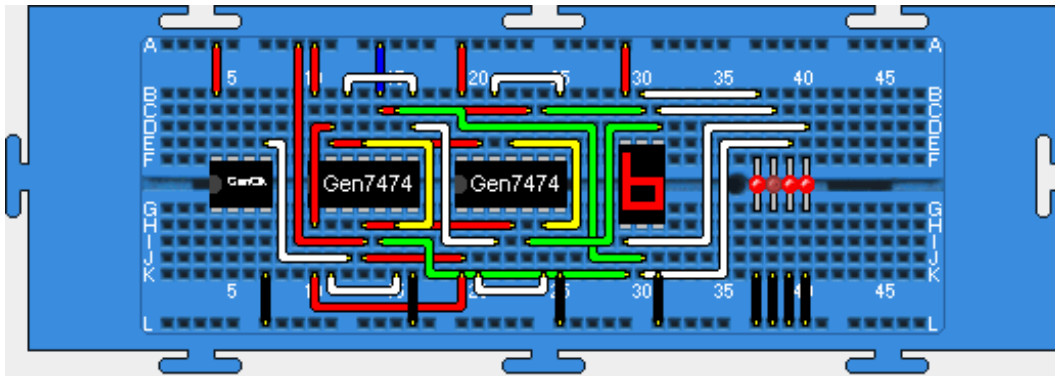
Gambar 51. Pencacah Modulo-16 FFJK Menggunakan Simulator *Breadboard*

- d. Jalankan simulasi dan atur **kecepatan simulasi (Sim Speed) hingga level 8,5!** Amati input pencacah melalui LED dan output pencacah melalui peraga heksadesimal serta catat hasilnya ke dalam tabel pengamatan.
- e. Lakukan prosedur yang sama untuk pencacah modulo-16 dengan FFD. Gunakan rangkaian pencacah modulo-16 dengan FFD berikut ini untuk percobaan.



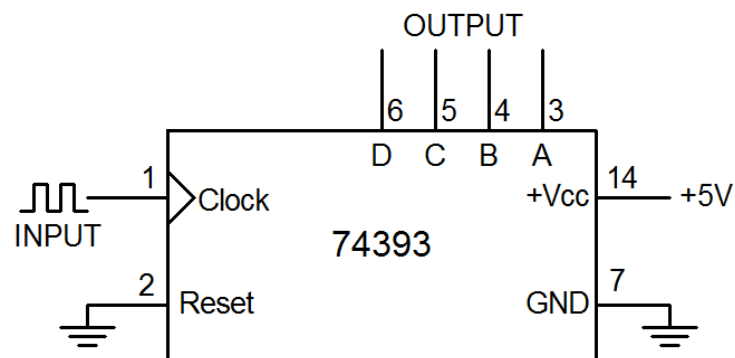
Gambar 52. Pencacah Modulo-16 dengan FFD

Salah satu bentuk susunan rangkaian pencacah modulo-16 dengan FFD menggunakan simulator *breadboard* ditunjukkan pada gambar berikut ini.



Gambar 53. Pencacah modulo-16 dari FFD menggunakan simulator *breadboard*

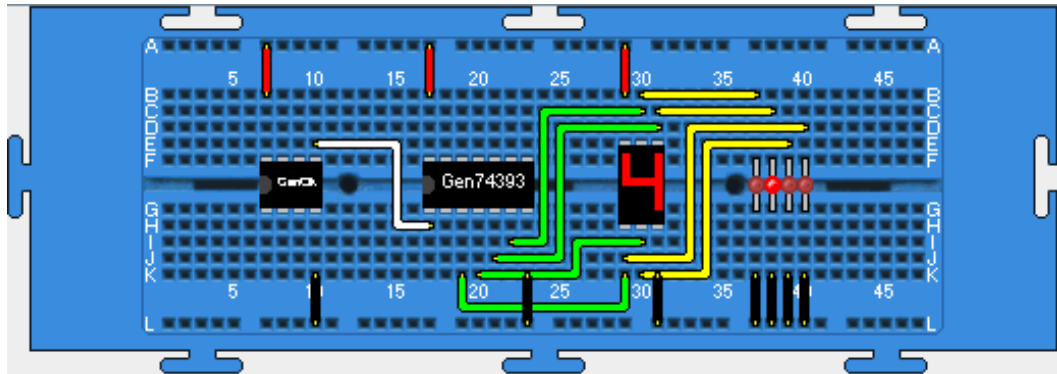
2. Merancang dan menentukan watak rangkaian pencacah modulo-16, modulo-12, dan modulo-6 menggunakan IC TTL 74393
  - a. Gunakan rangkaian pencacah modulo-16 menggunakan IC 74393 berikut ini untuk percobaan.



Gambar 54. Pencacah modulo-16 menggunakan IC 74393

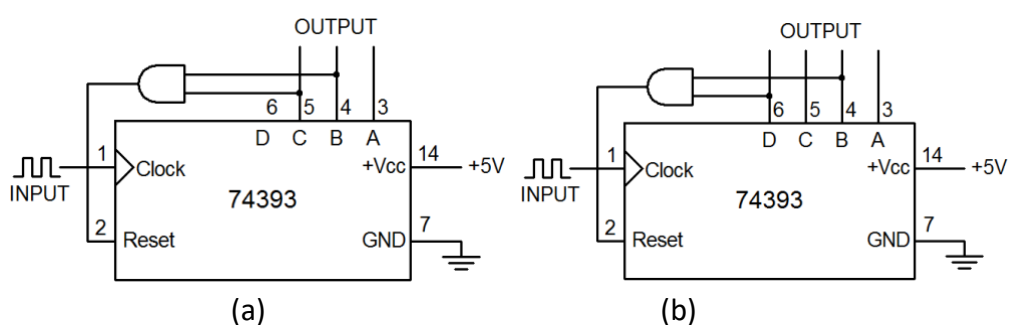
- b. Tempatkan sebuah IC 74393 (Pencacah), sebuah generator *clock*, sebuah peraga 7-segmen, dan empat buah LED pada papan *breadboard*.
- c. Pasang catu daya pada semua IC yang digunakan, pasang generator *clock* pada input, pasang indikator LED dan peraga heksadesimal pada output pencacah. Salah satu bentuk susunan rangkaian pencacah modulo-16

dengan IC 74393 menggunakan simulator *breadboard* ditunjukkan pada gambar berikut ini.



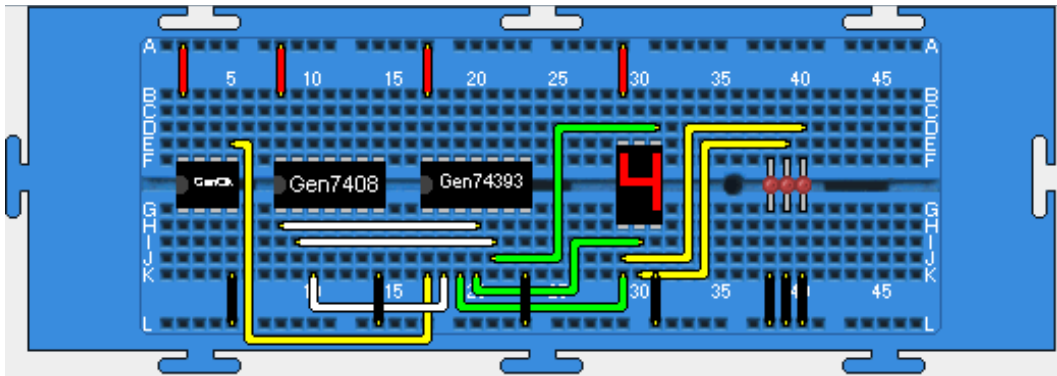
Gambar 55. Pencacah modulo-16 dengan IC 74393 menggunakan simulator *breadboard*

- d. Jalankan simulasi dan atur **kecepatan simulasi (Sim Speed) hingga level 8,5!** Berikan sinyal *clock* pada inputnya, dan amati output rangkaian pencacah pada indikator LED maupun peraga heksadesimal serta catat hasilnya ke dalam tabel pengamatan.
- e. Lakukan prosedur yang sama untuk pencacah modulo-6 dan modulo-12 menggunakan IC 74393. Rangkaian keduanya ditunjukkan pada gambar berikut ini.



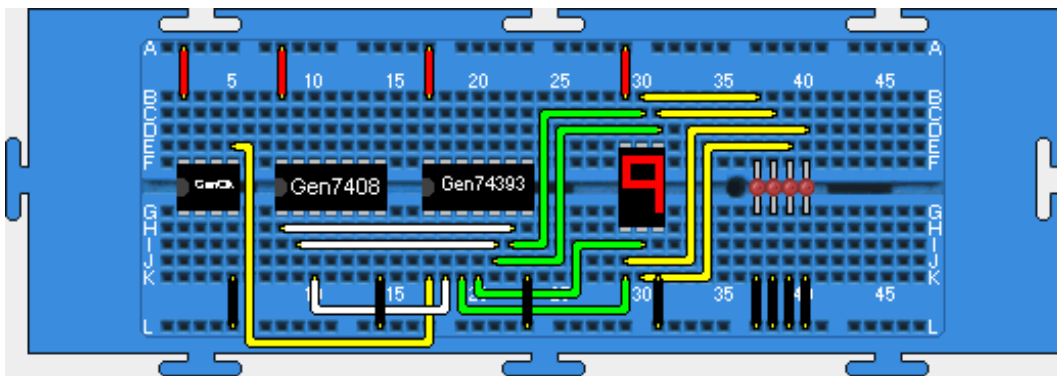
Gambar 56. Pencacah menggunakan IC 74393: (a) modulo-6; (b) modulo-12

- f. Susunan rangkaian pencacah modulo-6 dengan IC 74393 menggunakan simulator *breadboard* ditunjukkan pada gambar berikut ini.



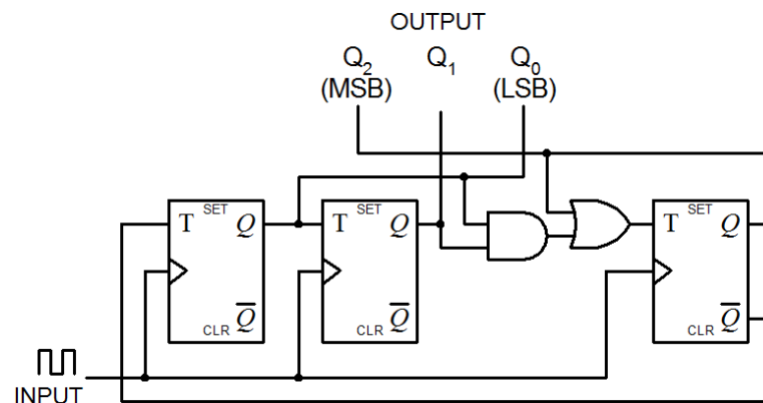
Gambar 57. Pencacah modulo-6 dengan IC 74393 menggunakan simulator *breadboard*

g. Sedangkan untuk pencacah modulo-12 dengan IC 74393, susunan rangkaiannya di atas *breadboard* ditunjukkan gambar berikut ini.



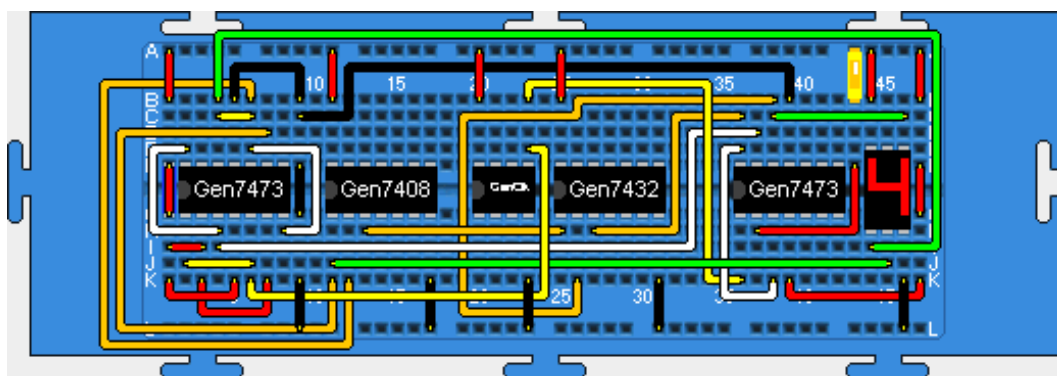
Gambar 58. Pencacah modulo-12 dengan IC 74393 menggunakan simulator *breadboard*

3. Merancang dan menentukan watak rangkaian pencacah sinkron modulo-5
  - a. Gunakan rangkaian berikut ini untuk percobaan!



Gambar 59. Pencacah sinkron modulo-5 menggunakan FFJK

- b. Siapkan di atas papan *breadboard* piranti: 2 buah IC 7473 (FFJK), IC 7408 (AND), IC 7432 (OR), saklar tunggal, generator *clock*, dan peraga heksadesimal.
- c. Selanjutnya, hubungkan titik-titik yang ada sesuai dengan rangkaian di atas. Perlu diingat, anda harus memasang saklar tunggal yang dihubungkan ke pin *Clear* dari FFJK paling kanan untuk memicu flip-flop agar rangkaian mulai bekerja.



Gambar 60. Pencacah sinkron modulo-5 menggunakan simulator *breadboard*

- d. Jalankan simulasi dan atur **kecepatan simulasi (Sim Speed) hingga level 8,5!** Untuk memicu rangkaian agar dapat mulai bekerja, berikan sinyal *Clear* pada FFJK paling kanan dengan meng-OFF-kan saklar, dan sesaat kemudian meng-ON-kannya kembali. Amati output rangkaian pencacah pada peraga heksadesimal dan catat hasilnya ke dalam tabel pengamatan.

## F. TUGAS LAPORAN

Buatlah tugas laporan praktik dengan sistematika: (1) tujuan, (2) alat dan bahan, (3) langkah percobaan, (4) hasil dan pembahasan, dan (5) kesimpulan. Bagian tujuan, alat dan bahan diuraikan sesuai dengan uraian pada panduan ini. Uraian tentang langkah percobaan diambil dari panduan ini dan pengalaman anda dalam melaksanakan percobaan. Hasil dan pembahasan berisi penyajian data yang

diperoleh melalui percobaan baik yang bersifat kuantitatif maupun kualitatif, perhitungan sesuai tujuan praktik, dan diskusi terhadap hasil analisis data.

Kesimpulan merupakan rangkuman dari hasil pembahasan. Dalam percobaan ini pembahasan diarahkan untuk menjawab pertanyaan-pertanyaan sebagai berikut:

1. Berdasarkan percobaan, apakah untuk pencacah modulo-16 menggunakan FFJK, FFD dan IC 74393 memberikan tabel keadaan yang sama? Jadi, pencacah dapat diimplementasikan menggunakan elemen memori dan IC apa saja? Cara mana yang lebih efisien?
2. Pada pencacah modulo-16 menggunakan FFJK (IC 7473) dan FFD (IC 7474), mengapa CLEAR dan PRESET dihubungkan ke +Vcc?
3. Pada percobaan pencacah tak sinkron modulo-6 maupun modulo-12 menggunakan IC 74393, apakah telah memberikan tabel kebenaran yang sesuai dengan tujuan perancangan? Kemukakan alasan-alasannya!
4. Mengapa pada pencacah modulo-16 menggunakan IC 74393, input CLEAR dihubungkan dengan GROUND atau logika 0?
5. Apa fungsi gerbang gerbang AND pada rangkaian pencacah tak sinkron modulo-6 maupun modulo-12? Untuk melakukan CLEAR, mengapa digunakan gerbang AND? Kapan digunakan gerbang NAND?
6. Berdasarkan percobaan, apakah pencacah sinkron modulo-5 telah bekerja dengan baik? Kemukakan alasannya! Apa perbedaan pencacah sinkron dan tak sinkron dari sisi pemberian sinyal *clock* yang diberikan? Lebih mana di antara kedua jenis pencacah tersebut?

## PERCOBAAN X REGISTER

### A. DESKRIPSI

Register merupakan memori n-bit. Jika sebuah flip-flop dapat digunakan untuk menyimpan data 1 bit, maka sederetan dari n flip-flop dapat digunakan untuk menyimpan data n-bit. Ada dua cara untuk memasukkan atau mengeluarkan data ke atau dari suatu register, yakni serial dan paralel. Pada cara serial, data dimasukkan atau dikeluarkan bit demi bit berganti-ganti lewat satu saluran, sedangkan pada cara paralel, n-bit dimasukkan atau dikeluarkan secara bersamaan lewat n saluran. Berdasarkan mekanisme pemasukan dan pengeluaran data tersebut, terdapat 4 macam register yakni *serial in-serial out (SISO)*, *serial in-parallel out (SIPO)*, *parallel in-parallel out (PIPO)*, dan *parallel in-serial out (PISO)*. Dalam keempat jenis register tersebut, terdapat jenis register geser kanan (*right shift register*) dan register geser kiri (*left shift register*).

Pada percobaan ini Anda akan mempelajari cara kerja dan watak beberapa jenis register khususnya SIPO dan PIPO.

### B. TUJUAN

Tujuan percobaan ini adalah:

1. merancang register *parallel in-parallel out (PIPO)* dan register geser *serial in-parallel out (SIPO)*; serta
2. menentukan watak register PIPO dan SIPO

### D. ALAT DAN BAHAN

1. IC TTL 7474 (FFD) 2 buah, dan 74164 (*shift register 8-bit*) 1 buah
2. DIP *switch: triple* dan *quad* masing-masing 1 buah
3. Indikator LED 4 buah
4. *Breadboard*, catu daya +5 V dan kabel penghubung (sudah tersedia di dalam simulator).

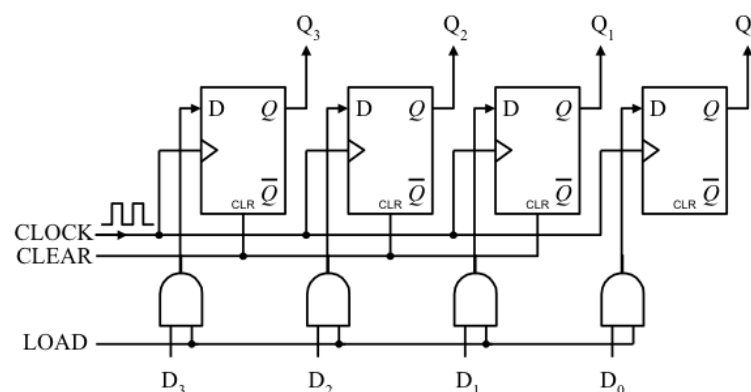


#### D. TUGAS PENDAHULUAN

1. Jelaskan pengertian register!
2. Sebutkan dan jelaskan jenis-jenis register!
3. Susun rangkaian register PIPO 4-bit menggunakan flip-flop D!
4. Susun rangkaian register geser 4-bit menggunakan flip-flop D!
5. Susun rangkaian register geser 4-bit menggunakan IC 74164!
6. Jelaskan cara kerja register-register tersebut dalam menyimpan data 1011!
7. Dengan menggunakan *timing diagram*, jelaskan perbedaan penyimpanan data secara serial dan paralel!
8. Berdasarkan tujuan praktikum, susun rangkaian dan tabel-tabel pengamatan percobaan yang diperlukan untuk merancang dan menentukan watak rangkaian register PIPO 4-bit dan SIPO (*shift register*) 4-bit.

#### E. PROSEDUR

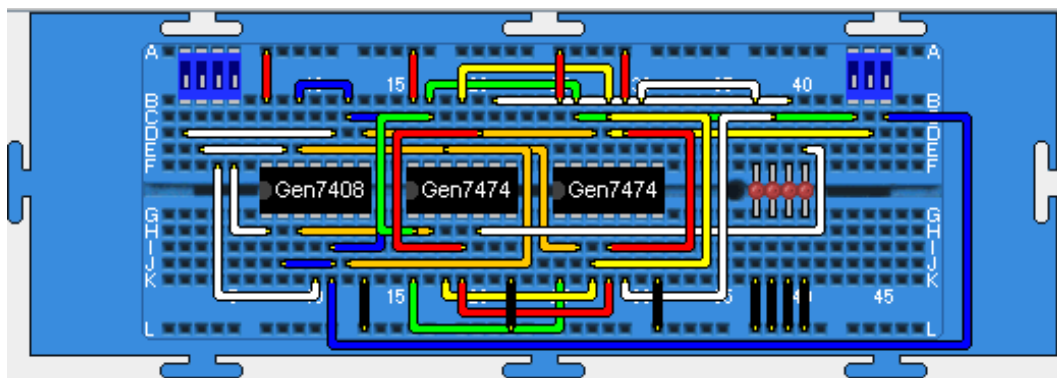
1. Merancang dan menentukan watak rangkaian register PIPO 4-bit menggunakan flip-flop D (IC 7474)
  - a. Gunakan rangkaian register PIPO 4-bit dengan flip-flop D (IC 7474) berikut ini untuk percobaan.



Gambar 61. Rangkaian PIPO 4-bit dengan FFD

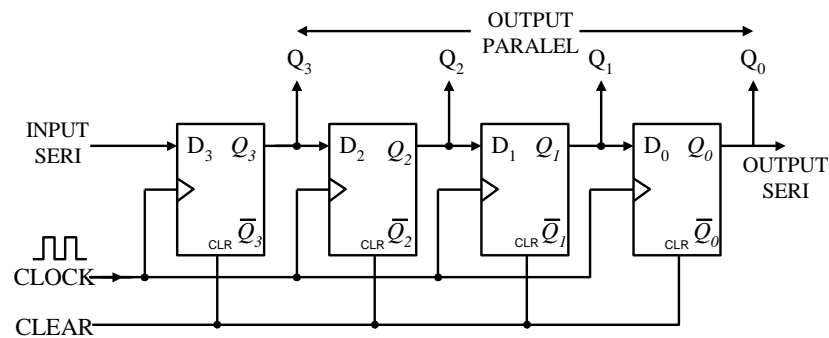
- b. Tempatkan dua buah IC 7474 (FFD), tiga buah saklar *double*, dan empat buah LED di atas *breadboard*.

- c. Pasang catu daya pada semua IC yang digunakan, pasang saklar logika pada input *clock*, *clear* dan data ( $D_3D_2D_1D_0$ ), pasang indikator LED pada output register (DCBA).
- d. Salah satu bentuk susunan rangkaian PIPO 4-bit menggunakan simulator *breadboard*, ditunjukkan pada gambar berikut ini.



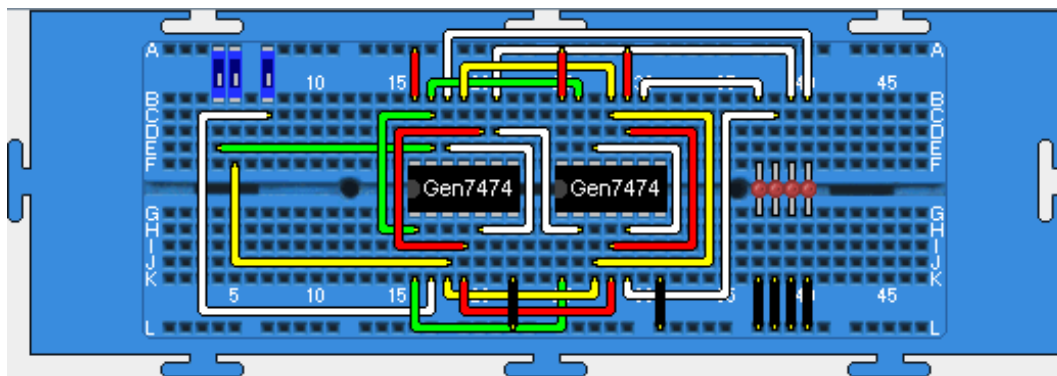
Gambar 62. Rangkaian PIPO 4-bit dengan IC 7474  
Menggunakan simulator *breadboard*

- e. Pastikan *Clear* bernilai 1 (saklar di-ON-kan) dan  $LOAD=1$ . Berikan data paralel pada inputnya ( $D_3D_2D_1D_0$ ) misalnya 1010, kemudian sinyal *clock* (saklar di-ON-kan, sesaat kemudian di-OFF-kan) untuk menyimpan data ke register. Lakukan langkah ini untuk variasi nilai  $LOAD$ . Amati output register dan catat hasilnya ke dalam tabel pengamatan.
  - f. Bersihkan output register dengan memberi sinyal *CLEAR* (saklar di-OFF-kan, sesaat kemudian di-ON-kan kembali). Amati output register dan catat hasilnya ke dalam tabel pengamatan.
  - g. Ulangi langkah-langkah tersebut untuk data input yang lain!
2. Merancang dan menentukan watak rangkaian register SIPO 4-bit menggunakan flip-flop D (IC 7474)
    - a. Gunakan rangkaian register SIPO 4-bit menggunakan flip-flop D (IC 7474) berikut ini untuk percobaan.



Gambar 63. Rangkaian SIPO 4-bit menggunakan FFD

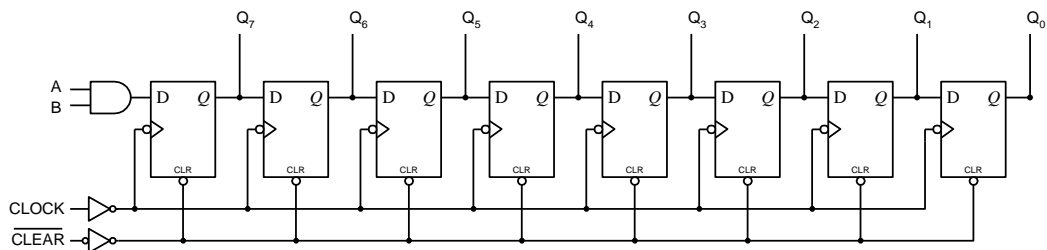
- b. Tempatkan dua buah IC 7474 (FFD), tiga buah saklar tunggal, dan empat buah LED di atas *breadboard*.
- c. Pasang catu daya pada semua IC yang digunakan, pasang saklar logika pada input *clock*, *clear* dan data serial, pasang indikator LED pada output register (DCBA). Salah satu bentuk susunan rangkaian SIPO 4-bit menggunakan simulator *breadboard*, ditunjukkan pada gambar berikut ini.



Gambar 64. Rangkaian SIPO 4-bit menggunakan simulator *breadboard*

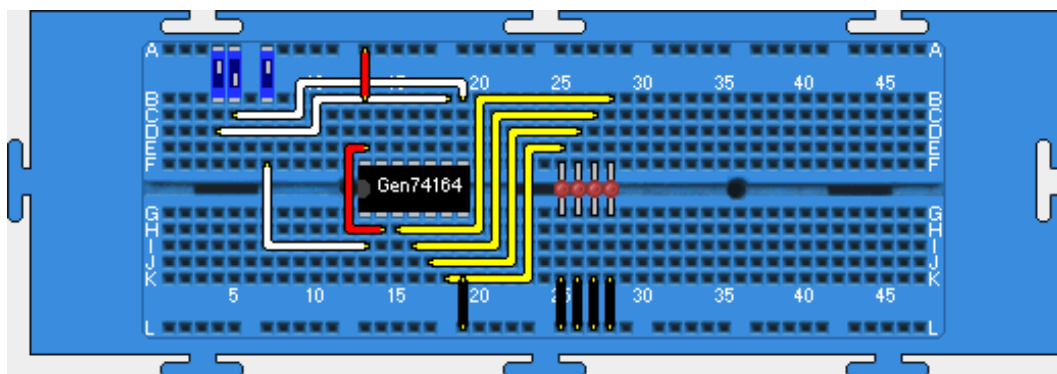
- d. Pasang data serial misalnya 1011 pada inputnya dan berikan sinyal *clock* untuk menyimpan data tersebut ke dalam register. Amati output register dan catat hasilnya ke dalam tabel pengamatan.
- e. Bersihkan output register dengan memberi sinyal CLEAR. Amati output register dan catat hasilnya ke dalam tabel pengamatan.
- f. Ulangi langkah-langkah tersebut untuk data input yang lain!

- g. Lakukan pula langkah-langkah yang sama seperti di atas untuk SIPO 4-bit menggunakan IC 74164!
- h. Rangkaian internal IC 74164 ditunjukkan pada gambar berikut ini.



Gambar 65. Rangkaian internal IC 74164

- i. Agar IC 74164 dapat berfungsi sebagai SIPO 4-bit, gunakan input A sebagai input serial dan input B dihubungkan ke +Vcc.
- j. Salah satu bentuk susunan rangkaian SIPO 4-bit dengan IC 74164 menggunakan simulator *breadboard*, ditunjukkan pada gambar berikut ini.



Gambar 66. Susunan SIPO 4-bit dengan IC 74164 menggunakan simulator *breadboard*

## F. TUGAS LAPORAN

Buatlah tugas laporan praktik dengan sistematika: (1) tujuan, (2) alat dan bahan, (3) langkah percobaan, (4) hasil dan pembahasan, dan (5) kesimpulan. Bagian tujuan, alat dan bahan diuraikan sesuai dengan uraian pada panduan ini. Uraian tentang langkah percobaan diambil dari panduan ini dan pengalaman anda dalam melaksanakan percobaan. Hasil dan pembahasan berisi penyajian data yang diperoleh melalui percobaan baik yang bersifat kuantitatif maupun kualitatif, perhitungan sesuai tujuan praktik, dan diskusi terhadap hasil analisis data.

Kesimpulan merupakan rangkuman dari hasil pembahasan. Dalam percobaan ini pembahasan diarahkan untuk menjawab pertanyaan-pertanyaan sebagai berikut:

1. Berdasarkan tabel keadaan yang diperoleh, tunjukkan bahwa register PIPO yang anda rancang telah dapat menyimpan data biner 4-bit secara paralel!
2. Apa fungsi LOAD, CLEAR dan CLOCK pada register?
3. Apakah register SIPO 4-bit menggunakan FFD menghasilkan tabel keadaan yang sama dengan register SIPO 4-bit menggunakan IC 74164 untuk input yang sama? Jadi, register dapat diperoleh dalam bentuk apa saja? Lebih mudah dan efisien yang mana dari bentuk-bentuk itu?
4. Tunjukkan bahwa register SIPO yang anda rancang telah dapat menyimpan data biner 4-bit secara serial!
5. Jelaskan perbedaan yang mendasar antara PIPO 4-bit dengan SIPO 4-bit! Lengkapi jawaban Anda dengan *timing diagram* kedua register tersebut dalam menyimpan data!

**DAFTAR PUSTAKA**

- Bailey, C. & Freeman, M. J. (2010). A Java bread-board simulator: Digital circuit simulation with an open-source toolset. *IADIS International Journal on Computer Science and Information System*, Volume VV, 1, 13-25.
- Freeman, M. 2010. *Getting started with java bread board in windows*. Heslington: Department of Computer Science The University of York.
- Glass, N. (2002). Java Digital Breadboard Simulator: A simulator for an educational electronics environment.
- Muchlas. 2013. *Dasar-dasar Rangkaian Digital*. Yogyakarta: UAD Press.
- Muchlas, & Novianta, M.A. (2015). An online lab for digital electronics course using information technology supports. *2015 International Conference on Science in Information Technology (ICSITech)*, 299-302.
- Muchlas. 2015. Developing a Teaching Model Using an Online Collaboration Approach for a Digital Technique Practical Work. *The Turkish Online Journal of Educational Technology*, 14(3), 63-69.

## LAMPIRAN 1

LAPORAN SEMENTARA  
PERCOBAAN I: WATAK GERBANG LOGIKA DASAR

Nama Praktikan: .....	Tanggal : .....
Anggota Lain : 1. ....	Pembimbing : .....
2. ....	
_____ Tanda tangan Pembimbing	

1. Data untuk menentukan watak gerbang logika dasar dan universal:

Tabel kebenaran gerbang logika dasar dan gerbang universal

INPUT		OUTPUT				
A	B	OR	AND	NOT A	NOR	NAND
0	0					
0	1					
1	0					
1	1					

- Untuk input, 0: saklar OFF, 1: saklar ON
- Untuk output, 0: LED padam, 1: LED menyala

2. Data untuk membuktikan gerbang OR dapat dibuat dari gerbang NAND atau NOR:

Tabel kebenaran untuk membuktikan gerbang OR dapat dibuat  
menggunakan gerbang NAND atau NOR

INPUT		OUTPUT		
A	B	OR ASLI	OR DENGAN NAND	OR DENGAN NOR
0	0			
0	1			
1	0			
1	1			

- Untuk input, 0: saklar OFF, 1: saklar ON
- Untuk output, 0: LED padam, 1: LED menyala

**LANJUTAN LAMPIRAN 1**

3. Data untuk membuktikan gerbang AND dapat dibuat dari gerbang NAND atau NOR:

Tabel kebenaran untuk membuktikan gerbang AND dapat dibuat menggunakan gerbang NAND atau NOR

INPUT		OUTPUT		
A	B	AND ASLI	AND DENGAN NAND	AND DENGAN NOR
0	0			
0	1			
1	0			
1	1			

- Untuk input, 0: saklar OFF, 1: saklar ON
- Untuk output, 0: LED padam, 1: LED menyala

4. Data untuk membuktikan gerbang NOT dapat dibuat dari gerbang NAND atau NOR:

Tabel kebenaran untuk membuktikan gerbang NOT dapat dibuat menggunakan gerbang NAND atau NOR

INPUT	OUTPUT		
A	NOT ASLI	NOT DENGAN NAND	NOT DENGAN NOR
0			
1			

- Untuk input, 0: saklar OFF, 1: saklar ON
- Untuk output, 0: LED padam, 1: LED menyala



**LANJUTAN LAMPIRAN 1**

LAPORAN SEMENTARA  
 PERCOBAAN II: MINIMALISASI RANGKAIAN LOGIKA

Nama Praktikan: ..... Tanggal : .....

Anggota Lain : 1. .... Pembimbing : .....

2. ....

\_\_\_\_\_  
 Tanda tangan Pembimbing

Data untuk membuktikan watak hasil minimalisasi rangkaian sama dengan watak rangkaian aslinya:

Tabel kebenaran untuk Y, Y<sub>m</sub>, Y<sub>NAND</sub> dan Y<sub>NOR</sub>

INPUT			OUTPUT			
A	B	C	Y	Y <sub>m</sub>	Y <sub>NAND</sub>	Y <sub>NOR</sub>
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

Keterangan:

Y = output yang ditentukan secara teori dari  $Y = \overline{A}BC + A\overline{B}C + ABC + B\overline{C}$

Y<sub>m</sub> = hasil minimalisasi dalam bentuk *Sum of Product* (SoP) =  $Y_m = AC + B$

Y<sub>NAND</sub> = hasil minimalisasi dalam bentuk NAND =  $Y_{NAND} = \overline{\overline{AC} \cdot \overline{B}}$

Y<sub>NOR</sub> = hasil minimalisasi dalam bentuk NOR =  $Y_{NOR} = \overline{(A + B) + (B + C)}$

**LANJUTAN LAMPIRAN 1**

LAPORAN SEMENTARA  
 PERCOBAAN III: KOMPARATOR DAN PENJUMLAH BINER

Nama Praktikan: ..... Tanggal : .....

Anggota Lain : 1. .... Pembimbing : .....

2. ....

\_\_\_\_\_  
 Tanda tangan Pembimbing

1. Data untuk menentukan watak rangkaian *non-equality comparator* dan *equality comparator*:

Tabel kebenaran *non-equality comparator* ( $Y_{SOP}$ ,  $Y_{XOR}$ ) dan *equality comparator* ( $Y_{POS}$ ,  $Y_{XNOR}$ )

INPUT		OUTPUT			
		Non-Equality Comparator		Equality Comparator	
A	B	$Y_{SOP}$	$Y_{XOR}$	$Y_{POS}$	$Y_{XNOR}$
0	0				
0	1				
1	0				
1	1				

2. Data untuk menentukan watak rangkaian *half-adder* 1-bit dan *full-adder* 1-bit:

Tabel kebenaran *half-adder* 1-bit

INPUT		OUTPUT	
A	B	S	Cn
0	0		
0	1		
1	0		
1	1		

## LANJUTAN LAMPIRAN 1

Tabel kebenaran *full-adder* 1-bit

INPUT			OUTPUT	
A	B	Cp	S	Cn
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

3. Data untuk menentukan watak rangkaian *full-adder* 4-bit:Tabel kebenaran *full-adder* 4-bit

INPUT			OUTPUT	
A (Heksadesimal)	B (Heksadesimal)	Cp	S (Heksadesimal)	Cn
2	5	0		
2	5	1		
1	3	0		
1	3	1		
2	6	0		
2	6	1		
3	9	0		
3	9	1		
A	3	0		
A	3	1		
7	B	0		
7	B	1		
9	0	0		
9	0	1		
F	1	0		
F	1	1		

## LANJUTAN LAMPIRAN 1

4. Data untuk menentukan watak rangkaian *full-adder* 8-bit:

Tabel kebenaran *full-adder* 8-bit

INPUT			OUTPUT	
A (Heksadesimal)	B (Heksadesimal)	Cp	S (Heksadesimal)	Cn
12	43	0		
12	43	1		
10	75	0		
10	75	1		
A2	3B	0		
A2	3B	1		
AA	22	0		
AA	22	1		
E7	18	0		
E7	18	1		
D3	1F	0		
D3	1F	1		
BC	44	0		
BC	44	1		
EA	16	0		
EA	16	1		

**LANJUTAN LAMPIRAN 1**

LAPORAN SEMENTARA  
PERCOBAAN IV: MULTIPLEKSER

Nama Praktikan: ..... Tanggal : .....

Anggota Lain : 1. .... Pembimbing : .....

2. ....

\_\_\_\_\_  
Tanda tangan Pembimbing

1. Data untuk menentukan watak rangkaian MUX 1-bit 4 ke 1:

Tabel kebenaran MUX 1-bit 4 ke-1

INPUT				PEMILIH		OUTPUT	
$I_0$	$I_1$	$I_2$	$I_3$	$S_1$	$S_0$	Y	HASIL PEMILIHAN INPUT
1	0	0	0	0	0		Y = $I_{\dots}$
0	0	0	0	0	0		
0	1	0	0	0	1		Y = $I_{\dots}$
0	0	0	0	0	1		
0	0	1	0	1	0		Y = $I_{\dots}$
0	0	0	0	1	0		
0	0	0	1	1	1		Y = $I_{\dots}$
0	0	0	0	1	1		

**LANJUTAN LAMPIRAN 1**

2. Data untuk menentukan watak rangkaian MUX 3-bit 2 ke 1:

Tabel kebenaran MUX 3-bit 2 ke-1

SINYAL KENDALI	INPUT		OUTPUT
S	A (Oktal)	B (Octal)	Y (Oktal)
0	3	5	
1	3	5	
0	0	2	
1	0	2	
0	7	1	
1	7	1	
0	6	4	
1	6	4	

**LANJUTAN LAMPIRAN 1**

LAPORAN SEMENTARA  
PERCOBAAN V: DEMULTIPLESER

Nama Praktikan: ..... Tanggal : .....

Anggota Lain : 1. .... Pembimbing : .....

2. ....

\_\_\_\_\_  
Tanda tangan Pembimbing

1. Data untuk menentukan watak rangkaian DEMUX 1-bit 1 ke 4 menggunakan gerbang logika dasar:

Tabel kebenaran DEMUX 1-bit 1 ke 4

KENDALI		INPUT	OUTPUT			
$S_1$	$S_0$	I	$Y_0$	$Y_1$	$Y_2$	$Y_3$
0	0	1				
0	1	1				
1	0	1				
1	1	1				

2. Data untuk menentukan watak dan kerusakan kanal output rangkaian DEMUX 1-bit 1 ke 4 menggunakan IC 74139:

Tabel kebenaran DEMUX 1-bit 1 ke 4 menggunakan IC 74129

KENDALI		INPUT	OUTPUT			
1B	1A	$\overline{1G}$	Y <sub>0</sub>	Y <sub>1</sub>	Y <sub>2</sub>	Y <sub>3</sub>
0	0	0				
0	1	0				
1	0	0				
1	1	0				



## LANJUTAN LAMPIRAN 1

LAPORAN SEMENTARA  
PERCOBAAN VI: ENKODER

Nama Praktikan: ..... Tanggal : .....

Anggota Lain : 1. .... Pembimbing : .....

2. ....

\_\_\_\_\_  
Tanda tangan Pembimbing

Data untuk menentukan watak rangkaian enkoder desimal ke BCD:

Tabel kebenaran enkoder desimal ke BCD

INPUT										OUTPUT			
0	1	2	3	4	5	6	7	8	9	D	C	B	A
1	0	0	0	0	0	0	0	0	0				
0	1	0	0	0	0	0	0	0	0				
0	0	1	0	0	0	0	0	0	0				
0	0	0	1	0	0	0	0	0	0				
0	0	0	0	1	0	0	0	0	0				
0	0	0	0	0	1	0	0	0	0				
0	0	0	0	0	0	1	0	0	0				
0	0	0	0	0	0	0	1	0	0				
0	0	0	0	0	0	0	0	1	0				
0	0	0	0	0	0	0	0	0	1				

**LANJUTAN LAMPIRAN 1**

LAPORAN SEMENTARA  
PERCOBAAN VII: DEKODER

Nama Praktikan: ..... Tanggal : .....

Anggota Lain : 1. .... Pembimbing : .....

2. ....

\_\_\_\_\_  
Tanda tangan Pembimbing

Data untuk menentukan watak rangkaian dekoder BCD ke desimal:

Tabel kebenaran dekoder BCD ke desimal

INPUT				OUTPUT									
D	C	B	A	0	1	2	3	4	5	6	7	8	9
0	0	0	0										
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										

## LANJUTAN LAMPIRAN 1

LAPORAN SEMENTARA  
PERCOBAAN VIII: FLIP-FLOP

Nama Praktikan: ..... Tanggal : .....

Anggota Lain : 1. .... Pembimbing : .....

2. ....

\_\_\_\_\_  
Tanda tangan Pembimbing

## 1. Data untuk menentukan watak rangkaian FFSR:

Tabel kebenaran FFSR  
menggunakan gerbang NOR

INPUT		OUTPUT			
S	R	$Q_{n-1}$	$Q_n$	$\overline{Q_n}$	KEADAAN
1	0	0			
1	0	1			
0	1	0			
0	1	1			
0	0	0			
0	0	1			
1	1	0			
1	1	1			

- $Q_n$ : output sekarang,  $Q_{n-1}$ : output sebelumnya
- Isi kolom keadaan dengan satu diantara: set ( $Q_n=1$ ), reset ( $Q_n=0$ ), tetap ( $Q_n=Q_{n-1}$ ), atau terlarang

## LANJUTAN LAMPIRAN 1

2. Data untuk menentukan watak rangkaian FFSR *clocked*:

Tabel kebenaran FFSR *clocked*

INPUT					OUTPUT			
S	R	PRESET	CLEAR	Clock	$Q_{n-1}$	$Q_n$	$\overline{Q_n}$	KEADAAN
1	0	0			0			
1	0		0		1			
0	1	0			0			
0	1		0		1			
0	0	0			0			
0	0		0		1			
1	1	0			0			
1	1		0		1			

- $Q_n$ : output sekarang,  $Q_{n-1}$ : output sebelumnya
- Pemberian sinyal *Clock*, *Preset*, *Clear* (tanda ) dilakukan dengan cara: saklar di-ON-kan, sesaat kemudian di-OFF-kan kembali
- Isi kolom keadaan dengan satu diantara: set ( $Q_n=1$ ), reset ( $Q_n=0$ ), tetap ( $Q_n=Q_{n-1}$ ), atau terlarang

3. Data untuk menentukan watak rangkaian FFJK:

Tabel kebenaran FFJK









INPUT				OUTPUT			
J	K	CLEAR	CLOCK	$Q_{n-1}$	$Q_n$	$\overline{Q_n}$	KEADAAN
1	0			0			
1	0	1		1			
0	1			0			
0	1	1		1			
0	0			0			
0	0	1		1			
1	1			0			
1	1	1		1			



- $Q_n$ : output sekarang,  $Q_{n-1}$ : output sebelumnya
- Pemberian sinyal *Clear* (tanda ) untuk membuat  $Q_{n-1}=0$  dengan cara: saklar di-OFF-kan, sesaat kemudian di-ON-kan
- Pemberian sinyal *Clock* (tanda ) dilakukan dengan cara: saklar di-ON-kan, sesaat kemudian di-OFF-kan kembali
- Isi kolom keadaan dengan satu diantara: set ( $Q_n=1$ ), reset ( $Q_n=0$ ), tetap ( $Q_n=Q_{n-1}$ ), atau membalik

**LANJUTAN LAMPIRAN 1**

4. Data untuk menentukan watak rangkaian FFD:







Tabel kebenaran FFD



INPUT				OUTPUT			
D	$\overline{\text{PRESET}}$	$\overline{\text{CLEAR}}$	CLOCK	$Q_{n-1}$	$Q_n$	$\overline{Q_n}$	KEADAAN
0	1			0			
0		1		1			
1	1			0			
1		1		1			

- $Q_n$ : output sekarang,  $Q_{n-1}$ : output sebelumnya
- Pemberian  $\overline{\text{PRESET}}$  dan  $\overline{\text{CLEAR}}$  (tanda ) diberikan dengan cara: saklar di-OFF-kan, sesaat kemudian di-ON-kan kembali
- Pemberian Clock (tanda ) dilakukan dengan cara: saklar di-ON-kan, sesaat kemudian di-OFF-kan kembali
- Isi kolom keadaan dengan satu diantara: set ( $Q_n=1$ ) atau reset ( $Q_n=0$ )

5. Data untuk menentukan watak rangkaian FFT:

Tabel kebenaran FFT

INPUT			OUTPUT			
T	CLEAR	CLOCK	$Q_{n-1}$	$Q_n$	$\overline{Q_n}$	KEADAAN
0			0			
0	1		1			
1			0			
1	1		1			

- $Q_n$ : output sekarang,  $Q_{n-1}$ : output sebelumnya
- *Clear* (tanda ) diberikan dengan cara: saklar di-OFF-kan, sesaat kemudian di-ON-kan kembali
- Pemberian *Clock* (tanda ) dilakukan dengan cara: saklar di-ON-kan, sesaat kemudian di-OFF-kan kembali
- Isi kolom keadaan dengan satu diantara: tetap ( $Q_n=Q_{n-1}$ ), atau membalik





Keadaan output rangkaian FFT yang diberi input T=1 dan pulsa *clock* secara kontinyu menggunakan generator *clock*:


.....  
 .....

**LANJUTAN LAMPIRAN 1**

6. Data untuk menentukan watak rangkaian *toggle*

Tabel kebenaran rangkaian *toggle* menggunakan FFJK dan FFD

INPUT	OUTPUT					
	TOGGLE DENGAN FFJK			TOGGLE DENGAN FFD		
Clock	$Q_{n-1}$	$Q_n$	$\overline{Q_n}$	$Q_{n-1}$	$Q_n$	$\overline{Q_n}$
	1			0		
	0			1		
	1			0		
	0			1		

- $Q_n$ : output sekarang,  $Q_{n-1}$ : output sebelumnya
- Pemberian Clock (tanda ) dilakukan dengan cara: saklar di-ON-kan, sesaat kemudian di-OFF-kan kembali

Keadaan output rangkaian *toggle* yang diberi input pulsa *clock* secara kontinyu menggunakan generator *clock*:

.....

.....

## LANJUTAN LAMPIRAN 1

LAPORAN SEMENTARA  
PERCOBAAN IX: PENCACAH

Nama Praktikan: ..... Tanggal : .....


















Anggota Lain : 1. .... Pembimbing : .....

2. ....

\_\_\_\_\_  
Tanda tangan Pembimbing


















1. Data untuk menentukan watak pencacah tak sinkron modulo-16 dari flip-flop

Tabel kebenaran pencacah modulo-16 menggunakan FFJK

INPUT CLOCK	OUTPUT				
	D	C	B	A	Heksadesimal
 ke-1					
 ke-2					
 ke-3					
 ke-4					
 ke-5					
 ke-6					
 ke-7					
 ke-8					
 ke-9					
 ke-10					
 ke-11					
 ke-12					
 ke-13					
 ke-14					
 ke-15					
 ke-16					
 ke-17					


















## LANJUTAN LAMPIRAN 1

Tabel kebenaran pencacah modulo-16 menggunakan FFD

INPUT CLOCK	OUTPUT				Heksadesimal
	D	C	B	A	
 ke-1					
 ke-2					
 ke-3					
 ke-4					
 ke-5					
 ke-6					
 ke-7					
 ke-8					
 ke-9					
 ke-10					
 ke-11					
 ke-12					
 ke-13					
 ke-14					
 ke-15					
 ke-16					
 ke-17					

## 2. Data untuk menentukan watak pencacah menggunakan IC 74393

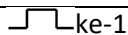
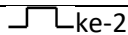
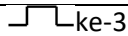
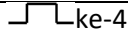
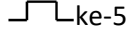
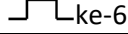

Tabel kebenaran pencacah modulo-16 menggunakan IC 74393

INPUT CLOCK	OUTPUT				Heksadesimal
	D	C	B	A	
 ke-1					
 ke-2					
 ke-3					
 ke-4					
 ke-5					
 ke-6					
 ke-7					
 ke-8					
 ke-9					
 ke-10					
 ke-11					
 ke-12					
 ke-13					
 ke-14					
 ke-15					
 ke-16					
 ke-17					


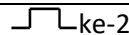
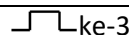

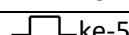
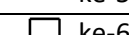
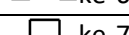
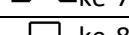
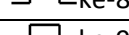
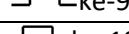
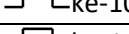
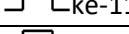
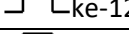


**LANJUTAN LAMPIRAN 1**

Tabel kebenaran pencacah modulo-6 menggunakan IC 74393

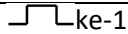
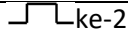
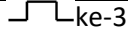
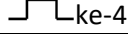
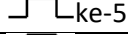
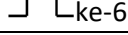
INPUT CLOCK	OUTPUT			
	C	B	A	Desimal
 ke-1				
 ke-2				
 ke-3				
 ke-4				
 ke-5				
 ke-6				
 ke-7				

Tabel kebenaran pencacah modulo-12 menggunakan IC 74393

INPUT CLOCK	OUTPUT				
	D	C	B	A	Heksadesimal
 ke-1					
 ke-2					
 ke-3					
 ke-4					
 ke-5					
 ke-6					
 ke-7					
 ke-8					
 ke-9					
 ke-10					
 ke-11					
 ke-12					
 ke-13					

### 3. Data untuk menentukan watak pencacah sinkron modulo-5

Tabel kebenaran pencacah sinkron modulo-5

INPUT CLOCK	OUTPUT			
	C	B	A	Desimal
 ke-1				
 ke-2				
 ke-3				
 ke-4				
 ke-5				
 ke-6				

**LANJUTAN LAMPIRAN 1**

LAPORAN SEMENTARA  
PERCOBAAN X: REGISTER

Nama Praktikan: ..... Tanggal : .....

Anggota Lain : 1. .... Pembimbing : .....

2. ....

\_\_\_\_\_  
Tanda tangan Pembimbing

1. Data untuk menentukan watak watak register PIPO 4-bit menggunakan FFD

Tabel kebenaran register PIPO 4-bit menggunakan FFD


























INPUT							OUTPUT			
D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	CLEAR	LOAD	CLOCK	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>
1	0	1	0	1	0					
1	0	1	0	1	1					
X	X	X	X		X	X				
1	1	0	1	1	0					
1	1	0	1	1	1					
X	X	X	X		X	X				
0	0	0	1	1	0					
0	0	0	1	1	1					
X	X	X	X		X	X				
0	0	1	1	1	0					
0	0	1	1	1	1					
X	X	X	X		X	X				
0	1	0	1	1	0					
0	1	0	1	1	1					
X	X	X	X		X	X				



- X: kondisi diabaikan
- CLEAR=1 dilakukan dengan meng-ON-kan saklar
- CLEAR= dilakukan dengan meng-OFF-kan saklar, sesaat kemudian meng-ON-kannya
- CLOCK= dilakukan dengan meng-ON-kan saklar, sesaat kemudian meng-OFF-kannya

## LANJUTAN LAMPIRAN 1

## 2. Data untuk menentukan watak register SIPO 4-bit menggunakan FFD

Tabel kebenaran register SIPO 4-bit menggunakan FFD


























INPUT			OUTPUT			
D	CLEAR	CLOCK	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>
1	1					
0	1					
1	1					
1	1					
X		X				
0	1					
1	1					
0	1					
1	1					
X		X				
0	1					
0	1					
1	1					
1	1					
X		X				
0	1					
1	1					
1	1					
1	1					
X		X				
1	1					
0	1					
0	1					
1	1					
X		X				



- X: kondisi diabaikan
- CLEAR=1 dilakukan dengan meng-ON-kan saklar
- CLEAR= dilakukan dengan meng-OFF-kan saklar, sesaat kemudian meng-ON-kannya
- CLOCK= dilakukan dengan meng-ON-kan saklar, sesaat kemudian meng-OFF-kannya

## LANJUTAN LAMPIRAN 1

## 3. Data untuk menentukan watak register SIPO 4-bit menggunakan IC 74164

Tabel kebenaran register SIPO 4-bit menggunakan IC 74164

INPUT			OUTPUT			
D	CLEAR	CLOCK	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>
1	1					
0	1					
1	1					
1	1					
X		X				
0	1					
1	1					
0	1					
1	1					
X		X				
0	1					
0	1					
1	1					
1	1					
X		X				
0	1					
1	1					
1	1					
1	1					
X		X				
1	1					
0	1					
0	1					
1	1					
X		X				

- X: kondisi diabaikan
- CLEAR=1 dilakukan dengan meng-ON-kan saklar
- CLEAR= dilakukan dengan meng-OFF-kan saklar, sesaat kemudian meng-ON-kannya
- CLOCK= dilakukan dengan meng-ON-kan saklar, sesaat kemudian meng-OFF-kannya

## LAMPIRAN 2

## DAFTAR NAMA FILE PERCOBAAN

Percobaan	Topik dan Sub Topik	Nama File
I	Watak gerbang logika dasar dan universal	
	1. Watak gerbang logika dasar	percobaan-1-BASIC.cir
	2. Gerbang OR menggunakan gerbang NAND atau NOR	percobaan-1-OR.cir
	3. Gerbang AND menggunakan gerbang NAND atau NOR	percobaan-1-AND.cir
	4. Gerbang NOT menggunakan gerbang NAND atau NOR	percobaan-1-NOT.cir
II	Minimalisasi Rangkaian Logika	
	1. Rangkaian dengan jumlah gerbang minimum	percobaan-2-SOP-MIN.cir
	2. Rangkaian logika dengan gerbang NAND saja	percobaan-2-NAND-MIN.cir
	3. Rangkaian logika dengan gerbang NOR saja	percobaan-2-NOR-MIN.cir
III	Komparator dan Penjumlah Biner	
	1. Komparator:	
	a. Non-Equality Comparator	percobaan-3-NEC.cir
	b. Equality Comparator	percobaan-3-EC.cir
	2. Adder 1-bit:	
	a. Half-adder	percobaan-3-HADD.cir
	a. Full-adder	percobaan-3-FADD.cir
	3. Full-adder paralel 4-bit	percobaan-3-PADD-4bit.cir
	4. Full-adder paralel 8-bit	percobaan-3-PADD-8bit.cir
IV	Multiplexer	
	1. MUX 1-bit 4 ke 1	percobaan-4-MUX.cir
	2. MUX 3-bit 2 ke 1	percobaan-4-MUX3-BIT-2KE1.cir
V	Demultiplexer	
	1. DEMUX 1 ke 4	percobaan-5-DEMUX-1KE4.cir
	2. DEMUX 1 ke 4 dengan IC 74139	percobaan-5-DEMUX-74139.cir
VI	Enkoder	
	Enkoder desimal ke BCD	percobaan-5-ENKODER-BCD.cir
VII	Dekoder	
	Dekoder BCD ke desimal	percobaan-5-DEKODER-BCD.cir
VIII	Flip-Flop	
	1. Flip-flop SR (FFSR)	percobaan-6-FFSR.cir
	2. FFSR Clocked	percobaan-6-FFSR-preset-clear.cir
	3. FFJK	percobaan-6-FFJK-clear.cir
	4. FFD	percobaan-6-FFD-preset-clear.cir
	5. FFT dengan clock saklar	percobaan-6-FFT-clear.cir
	6. FFT dengan clock kontinyu	percobaan-6-FFT-Generator-Clock.cir
	7. Toggle dengan clock saklar	percobaan-6-toggle-saklar.cir
	8. Toggle dengan clock kontinyu	percobaan-6-toggle-clock-gen.cir

## LANJUTAN LAMPIRAN 2

## LANJUTAN DAFTAR NAMA FILE PERCOBAAN

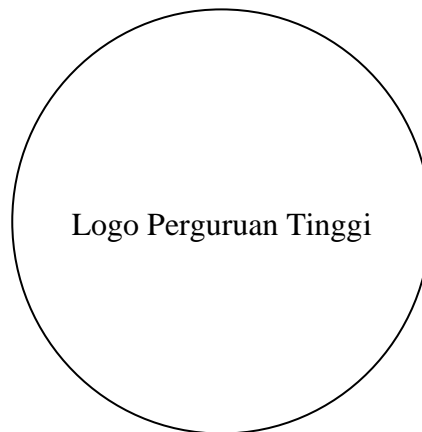
Percobaan	Topik dan Sub Topik	Nama File
IX	Pencacah	
	1. Pencacah modulo-16 dengan FF: a. Menggunakan FFJK	percobaan-7-CounterFFJK-16.cir
	b. Menggunakan FFD	percobaan-7-FFDmodulo-16.cir
	2. Pencacah dengan IC 74393: a. Modulo-16	percobaan-7-counter74393-modulo-16.cir
	b. Modulo-6	percobaan-7-counter74393-modulo-6.cir
	c. Modulo-12	percobaan-7-counter74393-modulo-12.cir
	3. Pencacah sinkron modulo-5	percobaan-7-Pencacah-Sinkron-5.cir
X	Register	
	1. PIPO 4-bit FFD	percobaan-8-PIPO-FFD-LOAD.cir
	2. SIPO 4-bit FFD	percobaan-8-SIPO-FFD.cir
	3. SIPO 4-bit dengan IC 74164	percobaan-8-SIPO-74164-4bit.cir

**LAMPIRAN 3**

CONTOH HALAMAN SAMPUL LAPORAN PRAKTIKUM

LAPORAN PRAKTIK  
TEKNIK DIGITAL

PERCOBAAN VIII  
REGISTER



oleh:  
Ahmad Sholeh  
12324007

<NAMA PROGRAM STUDI>  
<NAMA FAKULTAS>  
<NAMA UNIVERSITAS>  
<NAMA KOTA>  
<TAHUN>

**LAMPIRAN 4****A. Rujukan Tugas Pendahuluan I**

Nomor Tugas	Bab Rujukan Pada Buku Ajar	Halaman Rujukan
1	Bab III: Gerbang Logika Dasar dan Aljabar Boole	25 s.d.28 dan 32 s.d. 34
2	Bab III: Gerbang Logika Dasar dan Aljabar Boole	25, 26, 28, 32 dan 33
3	Bab III: Gerbang Logika Dasar dan Aljabar Boole	26 s.d. 28 dan 32, 33
4	Bab III: Gerbang Logika Dasar dan Aljabar Boole	25, 27, 28, 33 dan 34
5	Bab III: Gerbang Logika Dasar dan Aljabar Boole	36 dan 37
6	Bab III: Gerbang Logika Dasar dan Aljabar Boole	26 s.d.28 dan 33 s.d. 34
7.a.	Bab III: Gerbang Logika Dasar dan Aljabar Boole	26 s.d.28 dan 33 s.d. 34
7.b.	Bab III: Gerbang Logika Dasar dan Aljabar Boole	36 s.d. 37

**B. Rujukan Tugas Pendahuluan II**

Nomor Tugas	Bab Rujukan Pada Buku Ajar	Halaman Rujukan
1	Bab III: Gerbang Logika Dasar dan Aljabar Boole	35 dan 36
2	Bab IV: Rangkaian Logika Kombinasi	42 s.d. 44
3.a. 3.c.	Bab IV: Rangkaian Logika Kombinasi	29 s.d. 31
3.b.	Bab IV: Rangkaian Logika Kombinasi	46 s.d. 55
3.d.	Bab IV: Rangkaian Logika Kombinasi	57 s.d. 59
3.e.	Bab IV: Rangkaian Logika Kombinasi	58
4.a.	Bab IV: Rangkaian Logika Kombinasi	49
4.b.	Bab IV: Rangkaian Logika Kombinasi	57 s.d. 59



**LANJUTAN LAMPIRAN 4****C. Rujukan Tugas Pendahuluan III****Komparator:**

Nomor Tugas	Bab Rujukan Pada Buku Ajar	Halaman Rujukan
1	Bab V: Logika Kombinasi Dalam Kemasan IC	65 dan 68
2	Bab V: Logika Kombinasi Dalam Kemasan IC	65, 66, 68, dan 70
3	Bab V: Logika Kombinasi Dalam Kemasan IC	70
4.a.	Bab V: Logika Kombinasi Dalam Kemasan IC	65, 67
4.b.	Bab V: Logika Kombinasi Dalam Kemasan IC	68 s.d. 70

**Adder:**

Nomor Tugas	Bab Rujukan Pada Buku Ajar	Halaman Rujukan
1	Bab V: Logika Kombinasi Dalam Kemasan IC	71, 72
2	Bab V: Logika Kombinasi Dalam Kemasan IC	73
3	Bab V: Logika Kombinasi Dalam Kemasan IC	74, 75
4.a.	Bab V: Logika Kombinasi Dalam Kemasan IC	71 dan 72
4.b.	Bab V: Logika Kombinasi Dalam Kemasan IC	73
4.c.	Bab V: Logika Kombinasi Dalam Kemasan IC	75

**LANJUTAN LAMPIRAN 4****D. Rujukan Tugas Pendahuluan IV dan Tugas Pendahuluan V**

Nomor Tugas	Bab Rujukan Pada Buku Ajar	Halaman Rujukan
1 dan 2	Bab V: Logika Kombinasi Dalam Kemasan IC	76 dan 80
3	Bab V: Logika Kombinasi Dalam Kemasan IC	78
4	Bab V: Logika Kombinasi Dalam Kemasan IC	100
5	Bab V: Logika Kombinasi Dalam Kemasan IC	82, 83
6.a.	Bab V: Logika Kombinasi Dalam Kemasan IC	77, 78
6.b.	Bab V: Logika Kombinasi Dalam Kemasan IC	100
6.c.	Bab V: Logika Kombinasi Dalam Kemasan IC	81, 82

**E. Rujukan Tugas Pendahuluan VI dan Tugas Pendahuluan VII**

Nomor Tugas	Bab Rujukan Pada Buku Ajar	Halaman Rujukan
1	Bab V: Logika Kombinasi Dalam Kemasan IC	84
2	Bab II: Sistem Bilangan dan Sistem Kode	17 dan 18
3,4,5	Bab V: Logika Kombinasi Dalam Kemasan IC	85 s.d. 89
6	Bab V: Logika Kombinasi Dalam Kemasan IC	89 dan 90
7	Bab II: Sistem Bilangan dan Sistem Kode	17 dan 18
8, 9, 10	Bab V: Logika Kombinasi Dalam Kemasan IC	93 dan 94
11.a.	Bab V: Logika Kombinasi Dalam Kemasan IC	89
11.b.	Bab V: Logika Kombinasi Dalam Kemasan IC	94

**LANJUTAN LAMPIRAN 4****F. Rujukan Tugas Pendahuluan VIII**

:Nomor Tugas	Bab Rujukan Pada Buku Ajar	Halaman Rujukan
1	Bab VI: Rangkaian Logika Sekuensi	104, 107, 109, 110, 113, 115, dan 116
2	Bab VI: Rangkaian Logika Sekuensi	105
3	Bab VI: Rangkaian Logika Sekuensi	110
4	Bab VI: Rangkaian Logika Sekuensi	113
5	Bab VI: Rangkaian Logika Sekuensi	116
6	Bab VI: Rangkaian Logika Sekuensi	117 dan 118
7.a.	Bab VI: Rangkaian Logika Sekuensi	105, 110, 113, dan 116
7.b.	Bab VI: Rangkaian Logika Sekuensi	117, dan 118

**G. Rujukan Tugas Pendahuluan IX**

Nomor Tugas	Bab Rujukan Pada Buku Ajar	Halaman Rujukan
1	Bab VII: Pencacah dan Register	136
2	Bab VII: Pencacah dan Register	136
3	Bab VII: Pencacah dan Register	139, 140, 145
4	Bab VII: Pencacah dan Register	137
5	Bab VII: Pencacah dan Register	139 s.d. 141
6, 7	Bab VII: Pencacah dan Register	175
8.a.	Bab VII: Pencacah dan Register	137
8.b.	Bab VII: Pencacah dan Register	175

**LANJUTAN LAMPIRAN 4****H. Rujukan Tugas Pendahuluan X**

Nomor Tugas	Bab Rujukan Pada Buku Ajar	Halaman Rujukan
1	Bab VII: Pencacah dan Register	151
2	Bab VII: Pencacah dan Register	151 s.d. 155
3	Bab VII: Pencacah dan Register	151
4	Bab VII: Pencacah dan Register	152
5	Bab VII: Pencacah dan Register	175
6	Bab VII: Pencacah dan Register	153
7	Bab VII: Pencacah dan Register	151, 153
8.a.	Bab VII: Pencacah dan Register	151
8.b.	Bab VII: Pencacah dan Register	152