

TEKNIK DIGITAL

Teori dan Aplikasi

Dilengkapi dengan Contoh Simulasi Rangkaian

Buku ini membahas tentang materi teknik digital baik teori maupun aplikasinya dalam dunia kerja. Secara garis besar, buku ini membahas tentang konsep dasar teknik digital, sistem bilangan, gerbang logika dasar, Aljabar Boolean, Teori De Morgan, Penyederhanaan Rangkaian Digital, Rangkaian Flip-flop, Rangkaian Counter, Rangkaian Register, Rangkaian Analog to Digital Converter (ADC) dan Rangkaian Digital to Analog Converter (DAC). Pembahasan dilengkapi dengan contoh aplikasi baik di rumah tangga maupun industri sehingga akan memudahkan pembaca dalam memahaminya.



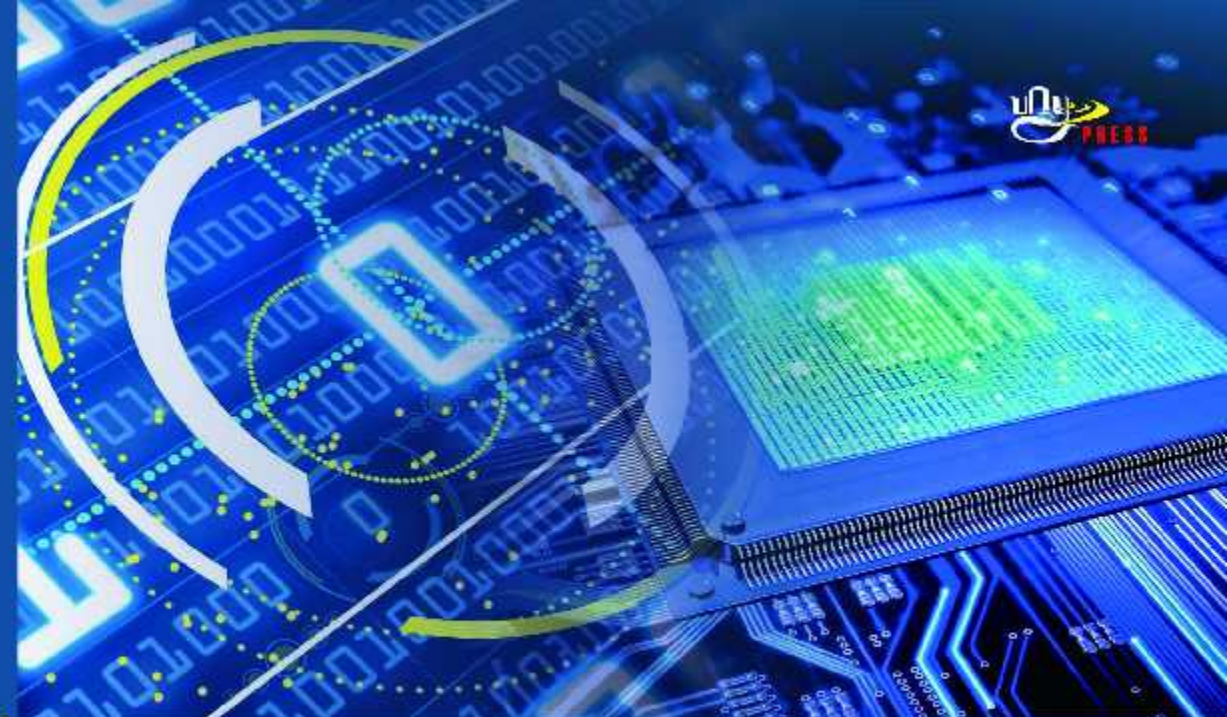
UNY Press
Jl. Gejayan, Gg. Alamanda, Komplek Fakultas Teknik UNY
Kampus UNY Karangmalang Yogyakarta 55281
Telp: 0274 - 589346
E-Mail: unypress.yogyakarta@gmail.com

Anggota Ikatan Penerbit Indonesia (IKAPI)
Anggota Asosiasi Penerbit Perguruan Tinggi Indonesia (APPTI)

Muhamad Ali
Ariadie Chandra Nugraha

TEKNIK DIGITAL
Teori dan Aplikasi

UNY PRESS

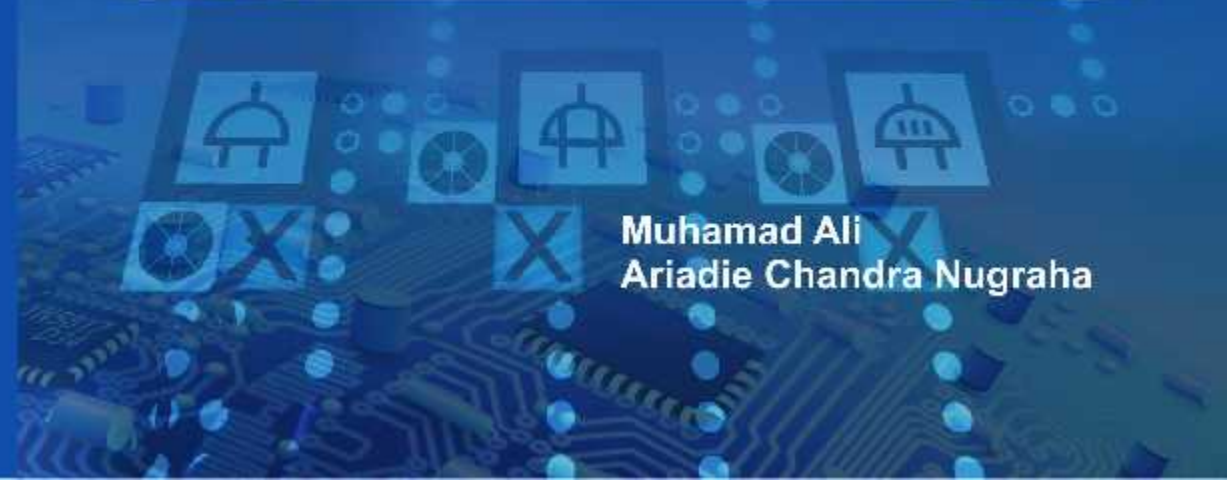


TEKNIK DIGITAL

Teori dan Aplikasi

Dilengkapi dengan Contoh Simulasi Rangkaian

Muhamad Ali
Ariadie Chandra Nugraha



Teknik Digital
Teori dan Aplikasi
Dilengkapi dengan Contoh Simulasi Rangkaian

**UNDANG-UNDANG REPUBLIK INDONESIA
NOMOR 28 TAHUN 2014
TENTANG HAK CIPTA**

Pasal 2

Undang-Undang ini berlaku terhadap:

- a. semua Ciptaan dan produk Hak Terkait warga negara, penduduk, dan badan hukum Indonesia;
- b. semua Ciptaan dan produk Hak Terkait bukan warga negara Indonesia, bukan penduduk Indonesia, dan bukan badan hukum Indonesia yang untuk pertama kali dilakukan Pengumuman di Indonesia;
- c. semua Ciptaan dan/atau produk Hak Terkait dan pengguna Ciptaan dan/atau produk Hak Terkait bukan warga negara Indonesia, bukan penduduk Indonesia, dan bukan badan hukum Indonesia dengan ketentuan:
 1. negaranya mempunyai perjanjian bilateral dengan negara Republik Indonesia mengenai perlindungan Hak Cipta dan Hak Terkait; atau
 2. negaranya dan negara Republik Indonesia merupakan pihak atau peserta dalam perjanjian multilateral yang sama mengenai perlindungan Hak Cipta dan Hak Terkait.

**BAB XVII
KETENTUAN PIDANA**

Pasal 112

Setiap Orang yang dengan tanpa hak melakukan perbuatan sebagaimana dimaksud dalam Pasal 7 ayat (3) dan/atau Pasal 52 untuk Penggunaan Secara Komersial, dipidana dengan pidana penjara paling lama 2 (dua) tahun dan/atau pidana denda paling banyak Rp300.000.000,00 (tiga ratus juta rupiah).

- (1) Setiap Orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp100.000.000 (seratus juta rupiah).
- (2) Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp500.000.000,00 (lima ratus juta rupiah).
- (3) Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf a, huruf b, huruf e, dan/atau huruf g untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 4 (empat) tahun dan/atau pidana denda paling banyak Rp1.000.000.000,00 (satu miliar rupiah).
- (4) Setiap Orang yang memenuhi unsur sebagaimana dimaksud pada ayat (3) yang dilakukan dalam bentuk pembajakan, dipidana dengan pidana penjara paling lama 10 (sepuluh) tahun dan/atau pidana denda paling banyak Rp4.000.000.000,00 (empat miliar rupiah).

TEKNIK DIGITAL

Teori dan Aplikasi

Dilengkapi dengan Contoh Simulasi Rangkaian

**Muhamad Ali
Ariadie Chandra Nugraha**

Teknik Digital Teori dan Aplikasi Dilengkapi dengan Contoh Simulasi Rangkaian

Oleh: Muhamad Ali dan Ariadie Chandra Nugraha

ISBN: 978-602-5566-80-6

Edisi Pertama

Diterbitkan dan dicetak oleh:

UNY Press

Jl. Gejayan, Gg. Alamanda, Komplek Fakultas Teknik UNY

Kampus UNY Karangmalang Yogyakarta 55281

Telp: 0274 - 589346

Mail: unypress.yogyakarta@gmail.com

© 2018 Muhamad Ali, Ariadie Chandra Nugraha

Anggota Ikatan Penerbit Indonesia (IKAPI)

Anggota Asosiasi Penerbit Perguruan Tinggi Indonesia (APPTI)

Desain Sampul: Muh Wildanul Firdaus

Tata Letak: Muh Wildanul Firdaus

Muhamad Ali, Ariadie Chandra Nugraha

Teknik Digital Teori dan Aplikasi Dilengkapi dengan Contoh Simulasi Rangkaian

-Ed. 1, Cet.1.- Yogyakarta: UNY Press 2018

viii + 142 hlm.; 17,6 x 25 cm

ISBN: 978-602-5566-80-6

1. Teknik Digital Teori dan Aplikasi Dilengkapi dengan Contoh Simulasi Rangkaian

Isi di luar tanggung jawab percetakan

Kata Pengantar

Alhamdulillah Robbil ‘alamin, puji dan syukur Kami panjatkan kehadiran Allah SWT, karena berkat rahmat dan hidayah-Nya penulis dapat menyelesaikan penulisan Buku dengan judul “Teknik Digital: Teori Dan Aplikasi Dilengkapi Dengan Contoh Simulasi Rangkaian”. Buku ini disusun guna melengkapi pustaka tentang Teknik Digital yang masih kurang di Indonesia. Dengan hadirnya buku ini, mahasiswa dapat lebih mudah belajar mata kuliah Teknik Digital sehingga diharapkan mampu membekali bidang ilmu keteknikelektroan.

Buku ini membahas tentang materi teknik digital baik teori maupun aplikasinya dalam dunia kerja. Secara garis besar, buku ini membahas tentang konsep dasar teknik digital, sistem bilangan, gerbang logika dasar, Aljabar Boole dan Teori De Morgan, Penyederhanaan Rangkaian Digital, Rangkaian Flip-flop, Rangkaian Counter, Rangkaian Register, Rangkaian Analog to Digital Converter (ADC) dan Rangkaian Digital to Analog Converter (DAC). Pembahasan dilengkapi dengan contoh aplikasi baik di rumah tangga maupun industri sehingga akan memudahkan pembaca dalam memahaminya.

Pada kesempatan ini, penulis mengucapkan banyak terima kasih kepada pihak-pihak yang telah membantu dalam penyusunan dan penyempurnaan buku ini. Tidak ada gading yang tidak retak, demikian pula dengan buku ini yang masih jauh dari kata sempurna. Kritik, saran dan masukan akan sangat diharapkan dan semoga buku ini dapat memberikan manfaat. Amin.

Yogyakarta, 1 November 2018

Penulis,

Daftar Isi

Kata Pengantar	v
Daftar Isi	vi
BAB 1. Pengantar Elektronika Daya	1
1.1. Pengantar	1
1.2. Sinyal dan Sistem	2
1.3. Contoh Sistem Digital	9
BAB 2. Komponen Semikonduktor Daya	13
2.1. Pengantar	13
2.2. Konversi Bilangan Biner ke Desimal dan Sebaliknya	15
2.3. Konversi Bilangan Hexadesimal dan Sebaliknya.....	17
2.4. Konversi Bilangan Biner Oktal dan Sebaliknya.....	20
BAB 3. Gerbang Logika dan Aljabar Boole.....	22
3.1. Operasi Logika NOT.....	22
3.2. Operasi Logika AND.....	23
3.3. Operasi Logika OR.....	23
3.4. Operasi Logika NAND.....	24
3.5. Operasi Logika NOR	24
3.6. Operasi Logika X-OR	25
3.7. Operasi Logika X-NOR	25
3.8. Simulasi Rangkaian dengan Proteus	26
3.9. Latihan Soal	37
BAB 4. Rangkaian Logika Kombinasi	40
4.1. Hukum-Hukum Aljabar Boolean.....	40
4.2. Simulasi Rangkaian dengan Proteus	44
4.3. Latihan Soal	48
BAB 5. Rangkain Flip Flop	51
5.1. Rangkaian Kombinatoral Sekuensial.....	51

5.2.	Flip-Flop RS	52
5.3.	Flip-Flop D	54
5.4.	Jenis-Jenis Flip-Flop Berdasar Clocknya	57
5.5.	Flip-Flop JK	58
5.6.	Simulasi Rangkaian Flip-Flop dengan Proteus	64
5.7.	Latihan Soal	73
BAB 6.	Rangkaian Register dan Counter.....	75
6.1.	Register	75
6.2.	Register Buffer	75
6.3.	Register Geser	77
6.4.	Counter	83
6.5.	Simulasi Rangkaian	92
6.6.	Latihan Soal	100
BAB 7.	Rangkaian Aritmatika dan Logika	103
7.1.	Rangkaian Half Adder	103
7.2.	Rangkaian Full Adder.....	103
7.3.	Rangkaian Half Subtractor.....	104
7.4.	Rangkaian Full Subtractor.....	104
7.5.	Simulasi Rangkaian dengan Proteus	105
7.6.	Latihan Soal	109
BAB 8.	Rangkaian Enkoder dan Dekoder	112
8.1.	Enkoder	112
8.2.	Dekoder.....	113
8.3.	Simulasi Rangkaian dengan Proteus.....	117
8.4.	Latihan Soal	120
BAB 9.	Rangkaian ADC dan DAC	123
9.1.	ADC	123
9.2.	DAC.....	128
9.3.	Simulasi dengan Proteus.....	133
9.4.	Latihan Soal.....	138
Daftar Pustaka	141

BAB 1

SISTEM DIGITAL

1.1. Pengantar

Dalam kehidupan sehari-hari, umat manusia selalu mempunyai kaitan dengan nilai atau kuantitas untuk menyatakan sesuatu ukuran seperti waktu, panjang, temperatur, kecepatan, kondisi peralatan dan lain sebagainya. Kuantitas-kuantitas ini biasanya diukur, dicatat dan dinyatakan dengan angka yang mempunyai satuan. Waktu dapat dinyatakan dalam satuan detik, menit, jam, hari, minggu dan seterusnya. Temperatur dapat dinyatakan dengan satuan celcius, fahrenheit, reamur, kelvin dan lain sebagainya. Panjang bisa dinyatakan dengan satuan meter, centimeter, kilo meter dan lain sebagainya.

Jika satuan dari kuantitas dinyatakan sesuai dengan besarnya, maka nilai dari kuantitas pada umumnya dinyatakan dengan angka. Contoh representasi suatu kuantitas dalam kehidupan sehari-hari yaitu waktu yang dibutuhkan untuk menyelesaikan pekerjaan 30 menit, kecepatan rata-rata kendaraan di jalan raya adalah 60 km/jam, zakat fitrah berupa beras 2,5 kg, suhu udara 25 0C. Contoh kuantitas lainnya adalah, jumlah pekerja di satu instansi didominasi oleh laki-laki.

Secara umum, besarnya kuantitas suatu bilangan dapat direpresentasikan dengan 2 cara yaitu secara analog dan secara digital. Representasi suatu kuantitas secara analog banyak digunakan dalam kehidupan sehari-hari untuk menyatakan suatu nilai yang mempunyai range data tak berhingga. Jumlah penduduk suatu negara dari tahun ke tahun dapat direpresentasikan dalam bentuk analog karena setiap tahun jumlah penduduk selalu mempunyai nilai baik naik, tetap atau turun. Beberapa kuantitas tidak dapat direpresentasikan secara analog karena hanya mempunyai dua kondisi. Contoh kuantitas yang hanya mempunyai dua kondisi adalah kondisi pintu apakah terbuka atau tertutup, jenis kelamin apakah pria atau wanita, lampu apakah menyala atau mati, televisi apakah hidup atau mati dan seterusnya.

Kuantitas-kuantitas seperti pada contoh ini tidak tepat kalau direpresentasikan secara analog, akan tetapi lebih cocok direpresentasikan secara digital.

Perbedaan representasi secara analog dan digital terletak pada kontinuitas datanya. Kuantitas yang direpresentasikan secara analog memberikan output yang kontinu sedangkan kuantitas yang direpresentasikan secara digital menghasilkan output yang diskrit atau hanya dalam 2 kondisi saja. Sistem analog memiliki piranti yang memproses atau bekerja pada berbagai besaran fisik yang direpresentasikan dalam bentuk analog, sedangkan sistem digital memiliki piranti yang memproses berbagai besaran fisik yang direpresentasikan dalam bentuk digital.

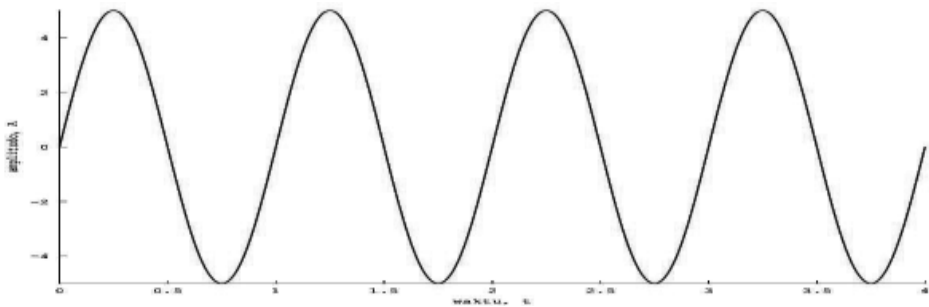
1.2. Sinyal dan Sistem

Dalam bidang teknik, istilah sinyal dan sistem mempunyai peran yang sangat penting. Banyak pembahasan memerlukan pemahaman yang baik tentang sinyal dan sistem. Secara umum sistem didefinisikan sebagai satu kesatuan dari komponen-komponen penyusun yang saling bekerjasama untuk mencapai tujuan yang direncanakan. Sedangkan pada bidang ilmu teknik digital, sistem diartikan sebagai satu kesatuan komponen yang mampu menghasilkan sinyal keluaran yang dikehendaki dari masukan sinyal yang diberikan kepadanya. Dalam bidang ilmu keteknikan, sistem dapat digunakan untuk menyatakan sistem perangkat lunak, sistem elektronik, sistem komputer atau sistem mekanik.

Sistem elektronika merupakan satu kesatuan yang terdiri dari komponen-komponen elektronika yang saling bekerjasama untuk menghasilkan keluaran yang diinginkan berdasarkan masukan tertentu. Contoh sistem elektronika adalah komputer yang terdiri dari banyak komponen diantaranya adalah prosesor, memori, *hardisk*, monitor, *keyboard*, *mouse*, *power supply* dan komponen pendukung lainnya yang saling berkolaborasi untuk menjalankan fungsi komputasi. Sebuah sistem selalu mempunyai masukan, pemrosesan dan keluaran. Pada sistem komputer masukan dapat dilakukan melalui *keyboard*, *mouse*, atau piranti masukan lainnya. Pemrosesan dilakukan oleh *Central Processing Unit* (CPU) yang melakukan

pemrosesan masukan secara digital sesuai dengan instruksi yang diberikan. Sedangkan keluaran dari sistem adalah tampilan di layar monitor atau hasil cetakan sesuai dengan hasil pemrosesan yang dilakukan.

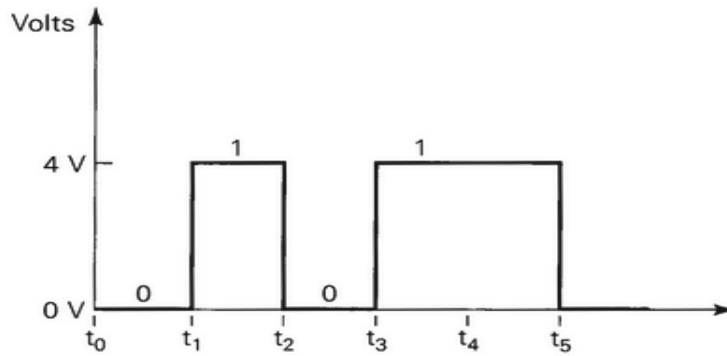
Sistem selalu berkaitan dengan sinyal yang didefinisikan sebagai suatu fungsi yang merepresentasikan kuantitas fisik (*variabel*) yang mengandung informasi tentang suatu fenomena. Pada umumnya sinyal dinyatakan sebagai grafik yang menunjukkan suatu nilai dari suatu parameter yang merupakan fungsi dari waktu. Sebagai contoh sinyal tegangan terhadap waktu, sinyal temperatur terhadap waktu dan lain sebagainya.



Gambar 1.1. Contoh sinyal tegangan (v) terhadap waktu (t)

Gambar 1.1. merupakan contoh sinyal tegangan terhadap waktu dimana nilai tegangan berubah-ubah terhadap waktu. Nilai minimal tegangan pada gambar di atas adalah $-4,5$ dan nilai maksimumnya adalah $4,5$ dengan nilai tengah adalah 0 . Sinyal tegangan di atas selalu mempunyai nilai sepanjang waktu t .

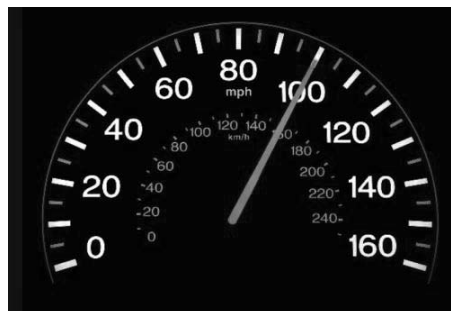
Contoh lain suatu sinyal adalah yang ditunjukkan pada gambar 1.2. Sinyal ini mempunyai karakteristik yang berbeda dengan sinyal pada gambar 1.1. Sinyal pada gambar 1.2 hanya mempunyai dua kondisi yaitu 0 dan 4 volt. Pada interval waktu t_0 sampai dengan t_1 sinyal mempunyai nilai nol, pada periode t_1 sampai t_2 sinyal mempunyai nilai 4 volt, pada periode t_2 sampai t_3 kembali bernilai nol dan pada periode t_3 sampai t_5 kembali bernilai 4 volt. Sinyal yang mempunyai karakteristik seperti ini disebut dengan sinyal diskrit atau sinyal digital.



Gambar 1.2. Contoh sinyal tegangan (v) terhadap waktu (t) dalam bentuk lain

1. Sistem dan Sinyal Analog

Representasi suatu kuantitas secara analog dilakukan dengan cara membandingkan dengan kuantitas lain yang nilainya berbanding lurus dengan kuantitas awal. Sebagai contoh representasi secara analog adalah kecepatan sepeda motor yang diukur dengan speedo meter, dimana jarum penunjuk kecepatan akan bergerak sesuai dengan kecepatan sepeda motor. Perubahan kecepatan sepeda motor akan ditunjukkan oleh perubahan penunjukan jarum pada speedo meter. Jika motor bergerak lebih cepat maka jarum speedo meter akan bergerak ke kanan, demikian sebaliknya jika sepeda motor berjalan lebih lambat, maka jarum speedo meter akan bergerak ke kiri.



Gambar 1.3. Tampilan kecepatan kendaraan dengan speedo meter

Contoh lain dari representasi secara analog adalah pada besaran tegangan listrik yang diukur dengan volt meter analog. Besarnya nilai hasil pengukuran tegangan ditunjukkan oleh jarum penunjuk pada volt meter. Jarum akan bergerak ke kanan jika nilai tegangan.



Gambar 1.4. Tampilan hasil pengukuran tegangan dengan voltmeter

Contoh-contoh di atas menjelaskan bahwa kuantitas analog mempunyai suatu ciri bahwa kuantitas akan berubah secara bertingkat pada suatu rentang harga secara kontinyu. pada abidang teknik elektro, sinyal analog seringkali dikaitkan dengan bentuk komunikasi elektromagnetik yang merupakan proses pengiriman sinyal pada gelombang elektromagnetik dan bersifat variable yang berurutan atau kontinyu. Sinyal analog mempunyai dua parameter penting yaitu amplitudo dan frekuensi. Sinyal analog biasanya dinyatakan dengan bentuk gelombang kontinyu baik dengan sinus, logaritmik maupun bentuk-bentuk gelombang kontinyu lainnya.

Sistem analog sampai sekarang masih dipakai dan dikembangkan oleh teknisi elektro. Beberapa keunggulan sistem analog diantaranya

1. Pada dasarnya sinyal yang terjadi di alam semesta kebanyakan berbentuk analog sehingga lebih mudah diolah dengan sinyal analog juga. Sebagai contoh sinyal suara yang dihasilkan oleh fenomena alam seperti angin, burung dan suara-suara alam lainnya. Selain sinyal yang dihasilkan oleh alam, beberapa sinyal buatan juga banyak menggunakan sinyal analog diantaranya suara dari mikrofon, sinyal alam yang ditangkap oleh seismograph, sinyal suara yang kirim dan diterima oleh sensor ultrasonic dan lain sebagainya.
2. Pada sistem komunikasi, untuk mengirim data pada sistem transmisi yang panjang biasanya lebih mudah digunakan sinyal analog, walaupun pada aplikasi sekarang banyak menggunakan sistem digital, tetapi perlu diubah menjadi sinyal analog. Untuk itu dibutuhkan perancangan konversi dari sinyal analog ke digital dan sebaliknya ADC dan DAC.

3. Berbagai peralatan penyimpanan data yang berkapasitas besar masih menggunakan bergantung pada peralatan yang menggunakan sinyal analog seperti pada Hard Disk komputer.
4. Sistem komunikasi data tanpa kabel (*wireless*) masih banyak yang menggunakan sistem dengan pengiriman dan penerima menggunakan sistem analog seperti pada antenna penerima RF, sinyal komunikasi Radio, TV dan peralatan komunikasi lainnya, walaupun sekarang ini sistem digital terus mengalami perkembangan yang sangat pesat.
5. Penerima optis pada komunikasi data dengan kecepatan tinggi melalui jalur fiber optik yang panjang, data harus diubah menjadi bentuk cahaya (*light*) yang merupakan sinyal analog. Sampai sekarang teknologi serat optik masih menjadi tulang punggung dalam komunikasi data dengan kecepatan tinggi.
6. Peralatan-peralatan bidang fotografi seperti pada sensor video camera masih banyak yang menggunakan sistem digital. Citra diubah menjadi arus menggunakan larik fotodioda sistem ultrasonik menggunakan sensor akustik untuk menghasilkan tegangan yang proporsional dengan amplitudo.

Selain kelebihan di atas, tentunya sistem analog mempunyai keterbatasan-keterbatasan, diantaranya adalah sebagai berikut:

1. Sinyal analog dalam implementasinya harus mempertimbangkan *speed*, *power dissipation*, *gain*, *precision*, *supply voltage* dan lain sebagainya. Secara umum sistem analog masih mengalami keterbatasan dalam hal kualitas, efisiensi dan gangguan atau noise.
2. Sistem analog lebih sensitif terhadap derau/noise, *crosstalk* dan interferensi sehingga rawan terjadi gangguan.
3. Implementasi sistem analog pada sistem otomasi memerlukan perancangan yang rumit karena harus menyesuaikan dengan kondisi yang ada dibandingkan dengan sistem digital yang lebih mudah.
4. Pemodelan dan simulasi sistem analog dibutuhkan perancangan yang kompleks karena banyak efek dan perilaku yang kadang tidak sesuai dengan teori sehingga dibutuhkan pengalaman

5. Perkembangan teknologi sekarang lebih mengarah ke sistem digital, sehingga banyak digunakan dan dirancang untuk memproduksi produk digital, karena sulit kalau mau memproduksi yang analog.

2. Sistem dan Sinyal Digital

Kata Digital berasal dari Bahasa Yunani yaitu Digitus yang berarti jari jemari. Manusia mempunyai 2 tangan yang masing-masing mempunyai jari 5 buah, sehingga jumlah jari kedua tangan adalah sepuluh yang ditulis “10”. Angka 10 terdiri dari dua 2 radix, yaitu 1 dan 0. Istilah digital digunakan untuk menggambarkan suatu keadaan bilangan yang terdiri dari angka 0 dan 1 atau keadaan on dan off. Bilangan digital juga disebut dengan dengan bilangan biner dan disebut juga dengan istilah Bit (Binary Digit).

Konsep digital memberikan gambaran pemahaman suatu keadaan yang saling berlawanan. Pada contoh lampu, bila saklar ditekan pada pada tombol on, lampu akan menyala dan ruangan akan tampak terang. Sebaliknya bila saklar ditekan pada tombol off, maka lampu akan mati dan ruangan menjadi gelap. Di dunia ini banyak kondisi yang hanya mempunyai dua kondisi seperti ON dan OFF seperti jenis kelamin yaitu Laki-laki dan Perempuan, kondisi katub terbuka dan tertutup, waktu siang dan malam, status pegawai aktif dan tidak aktif, suasana hati sedih dan bahagia serta kondisi-kondisi lain yang sejenis. Entitas-entitas ini hanya memiliki dua kondisi sepanjang waktu sehingga sering direpresentasikan dengan sistem digital.

Pada representasi digital, kuantitas-kuantitas tidak dinyatakan dengan kuantitas-kuantitas sebanding tetapi dengan simbol-simbol yang disebut digit. Sebagai contoh, perhatikanlah jam digital, yang menunjukkan waktu dalam bentuk digit-digit desimal yang menyatakan jam-menit-dan detik. Seperti diketahui, waktu berubah secara kontinyu, tetapi yang terbaca dalam jam digital tidak berubah secara kontinyu, ia berubah satu step demi satu step per detik. Dengan kata lain, representasi digital dari waktu berubah dalam step-step diskrit. Dibandingkan

dengan representasi analog dari waktu yang ditunjukkan oleh jarum jam, dimana pembacaan skala berubah secara kontinyu.



Gambar 1.5. Tampilan hasil pengukuran arus, tegangan dan frekuensi secara digital

Sistem digital mempunyai kelebihan jika dibanding dengan sistem analog diantaranya yaitu:

1. Sistem digital lebih mudah dikembangkan dan didesain.
2. Data dan informasi lebih mudah disimpan
3. Mempunyai tingkat ketelitiannya lebih baik
4. Dapat diprogram dengan mudah.
5. Noise atau gangguan lebih kecil
6. Rangkaian digital dapat di fabrikasi dalam bentuk IC chip

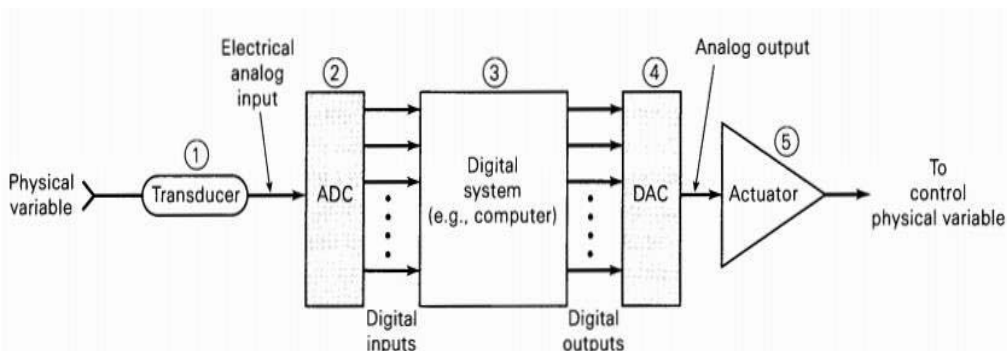
Selain kelebihan-kelebihan yang disebutkan di atas, sistem digital juga mempunyai keterbatasan diantaranya yaitu:

1. Dalam kenyataannya ada satu masalah utama dalam menggunakan sistem digital yaitu : Dalam kondisi riilnya semua kuantitas adalah bersifat analog, dan kuantitas-kuantitas inilah yang sering diukur, dimonitor atau dikontrol. Jadi apabila akan menggunakan teknik digital diperlukan langkah-langkah untuk mengubah menjadi sinyal digital dengan rangkaian Analog to Digital Converter atau ADC

2. Selanjutnya proses dilakukan secara digital
3. Ubah output digital menjadi besaran analog (Digital to Analog Converter atau DAC)
4. Diagram berikut menunjukkan sistem kontrol temperatur

1.3. Contoh Sistem Digital

Kebanyakan variable fisik pada dasarnya bersifat analog, yaitu dapat mempunyai nilai berapapun dalam sebuah jangkauan kontinyu dari nilai-nilai. Misalnya tegangan listrik, temperatur, tekanan, intensitas cahaya, dan lain-lain. Ketika sebuah sistem digital seperti komputer digunakan untuk memantau dan/atau mengendalikan sebuah proses fisik, kita harus menangani perbedaan antara sifat digital dari komputer dan sifat analog dari variabel proses. Gambar 1 memperlihatkan lima komponen yang terlibat ketika sebuah komputer digunakan untuk memantau dan mengendalikan sebuah variabel fisik analog.



Gambar 1.6 Penggunaan Sistem Kontrol Digital

Komponen-komponen pada sistem kontrol digital dapat dijelaskan sebagai berikut:

1. Sensor atau Transduser.

Variabel fisik pada umumnya berupa besaran atau kuantitas non-elektrik seperti suhu, tekanan, level, kecepatan, PH, kelembaban, intensitas cahaya, dan besaran-besaran lainnya. Besaran-besaran ini dapat diukur oleh alat ukur yang sesuai. Suhu dapat diukur dengan termometer, kecepatan dapat diukur dengan speedometer, PH dapat diukur dengan kertas lakmus, level dapat diukur dengan

alat ukur ketinggian dan lain sebagainya. Pada sistem digital, agar besaran ini dapat dibaca pada sistem digital harus diukur dengan peralatan ukur yang menghasilkan sinyal listrik yang berupa arus maupun tegangan. Untuk itu dibutuhkan alat yang disebut sensor atau transduser. Sensor atau transduser adalah adalah sebuah piranti yang dapat mengubah variabel fisik menjadi variabel listrik. Contoh transduser misalnya thermistor, tachometer, transduser tekanan, photodiode, dan lain-lain. Keluaran transduser adalah arus atau tegangan yang besarnya proporsional dengan variabel fisik yang dipantau.

Sebagai contoh, variabel fisik yang dipantau adalah suhu air dalam sebuah tangki besar yang mempunyai masukan dari pipa air dingin dan pipa air panas. Suhu air pada sebuah tangki berkisar antara 80 sampai 150° C, dan sebuah rangkaian thermistor akan menghasilkan keluaran tegangan yang berkisar antara 800 mV hingga 1500 mV.

2. Analog-to-digital Converter (ADC).

Pada suatu sistem elektronika digital, rangkaian hanya dapat membaca sinyal-sinyal digital. Keluaran sensor atau transduser ada yang analog dan ada juga yang digital. Untuk sensor yang mempunyai keluaran berupa sinyal analog, perlu ditambahkan rangkaian pengubah dari sinyal analog menjadi sinyal digital. Rangkaian ini sering disebut sebagai Analog to Digital Converter (ADC). Rangkaian ADC berfungsi untuk mengonversi masukan analog menjadi sebuah keluaran digital. Keluaran digital terdiri dari sejumlah bit yang mewakili nilai masukan analog.

Sebagai contoh, sebuah sensor suhu LM35 berfungsi untuk mengubah suhu menjadi tegangan listrik. Range tegangan keluaran LM35 berkisar antara 0 – 5 Volt. Agar keluaran sensor LM35 dapat dibaca pada sistem digital seperti mikrocontroller atau mikroprosesor diperlukan rangkaian pengubah dari analog ke digital. Misalnya, ADC akan mengubah nilai analog keluaran transduser yang berkisar dari 800-1500 mV menjadi nilai biner yang berkisar dari 0101000 (80_{10}) hingga 10010110 (150_{10}). Perhatikan bahwa keluaran biner dari ADC **proporsional** dengan tegangan analog masukan, sehingga setiap unit dari keluaran digital mewakili 10 mV.

3. **Kontroler.**

Kontroler digital bekerja dengan prinsip-prinsip digital yang hanya mampu membaca input digital dan akan menghasilkan output sinyal digital. Contoh kontroler digital yaitu mikrokontroler dan mikroprosesor. Mikrokontroler dan mikroprosesor banyak digunakan pada berbagai peralatan kontrol elektronik seperti pada mesin cuci, oven listrik, rice cooker, peralatan medis, Air Conditioner, Komputer, Smartphone dan peralatan-peralatan lainnya.

Kontroler berfungsi untuk mengatur keluaran sesuai dengan kondisi masukan dan program yang dimasukkan. Pada sistem Air Conditioner, kontroler berfungsi untuk mengatur aliran udara dingin sesuai dengan set point atau masukan yang diberikan oleh pengguna. Jika pengguna mengatur suhu yang diinginkan 22°C dan suhu ruangan lebih besar dari 26°C , maka kontroler akan memerintahkan kompresor untuk bekerja dan mengalirkan suhu dingin agar suhu ruangan bisa mencapai 22°C .

4. **Digital-to-analog Converter (DAC).**

Sistem digital akan mengeluarkan output berupa sinyal digital. Sementara aktuator yang bekerja untuk memanipulasi kondisi sesuai set point ada yang berupa analog atau digital. Jika aktuator bekerja secara analog, maka sistem kontrol digital memerlukan rangkaian pengubah sinyal digital menjadi sinyal analog. Contoh aktuator yang bekerja menggunakan sinyal analog adalah motor DC, motor servo, solenoide, valve, dan lain sebagainya. Aktuator analog membutuhkan sinyal analog agar dapat bekerja dengan baik, oleh karena itu, pada sistem digital diperlukan rangkaian pengubah sinyal digital menjadi sinyal analog dengan DAC. Keluaran dari DAC inilah yang nantinya digunakan untuk menggerakkan aktuator untuk memanipulasi kondisi yang dikontrol agar memenuhi nilai set point yang diharapkan.

5. **Aktuator.**

Aktuator merupakan komponen pada sistem kontrol yang berfungsi untuk memanipulasi parameter agar sesuai dengan nilai set point. Pada contoh sistem kontrol suhu air, aktuator dapat berupa katup yang dikendalikan secara elektrik, yang mengatur aliran air panas ke dalam tangki sesuai dengan tegangan analog dari DAC. Kecepatan aliran akan bervariasi sebanding dengan tegangan

analog, misalnya 0 V akan membuat katup tertutup dan 10 V akan menghasilkan aliran maksimal.

Pada sistem kontrol suhu Air Conditioner, aktuator yang digunakan adalah motor listrik untuk mengatur tekanan udara pada kompresor dan kecepatan kipas untuk menghasilkan suhu dingin agar dapat memanipulasi suhu ruangan menjadi suhu yang diseting.

BAB 2

SISTEM DAN KODE BILANGAN

2.1. Pengantar

Dalam kehidupan sehari-hari, kita mengenal sistem bilangan yang terdiri dari digit atau angka mulai dari nol sampai sembilan yaitu : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Bilangan ini digunakan untuk menyatakan nilai suatu besaran baik jumlah uang, nilai temperatur, jumlah anak, jumlah penduduk, berat barang, jarak satu tempat ke tempat lainnya dan banyak lagi parameter-parameter yang ada. Bilangan yang terdiri dari angka 0 sampai 9 yang biasa dipakai dalam kehidupan sehari-hari disebut dengan bilangan desimal. Pada ilmu matematika, Bilangan desimal mempunyai bilangan dasar sepuluh atau disebut dengan radiks 10 (sepuluh) yaitu terdiri dari digit sebanyak sepuluh.

Contoh pemakaian bilangan dalam kehidupan sehari dapat dilihat pada di bawah ini.

- Suhu udara hari ini berkisar pada 28° C
- Berat badan bayi yang baru lahir rata-rata 3 kg
- Syarat untuk dapat memperoleh beasiswa, mahasiswa harus mempunyai IPK minimal 3
- Kcecepatan putar motor listrik pompa air adalah 1450 RPM
- Efisiensi transformator adalah 90%
- Jumlah mahasiswa yang mengikuti kuliah Teknik Digital adalah 20 orang

Pada contoh di atas, angka-angka 28, 3, 3, 1450, 90, 20 merupakan angka yang menyatakan nilai suatu besaran. Nilai-nilai ini menggunakan sistem bilangan desimal.

2.1.1. Bilangan Desimal

Bilangan desimal berasal dari bahasa Latin “decum” yang berarti sepuluh atau ditulis dengan simbol 10. Sistem bilangan desimal merupakan sistem bilangan yang mempunyai basis 10 yang terdiri dari 10 angka yaitu dari angka 0 hingga 9. Bilangan desimal dapat terdiri dari 1,2,3,... digit yang

masing-masing menyatakan nilai dari bilangan itu. Penulisan bilangan terbagi dalam beberapa tempat dan banyaknya tempat tergantung dari besarnya bilangan. Setiap tempat mempunyai besaran tertentu yang harga masing-masing tempat secara urut dimulai dari kanan.

Ribuan	Ratusan	Puluhan	Satuan
10^3	10^2	10^1	10^0

antara lain : (125)₁₀, (468)₁₀, (8)₁₀, (1239)₁₀. Nilai dari setiap posisi angka bilangan desimal biasanya dimulai dari ujung kanan sebagai berikut: ... 10⁶, 10⁵, 10⁴, 10³, 10², 10¹, 10⁰, 10⁻¹, 10⁻², 10⁻³, 10⁻⁴, ... Artinya setiap angka atau digit, berbeda nilainya yang tergantung pada tempatnya.

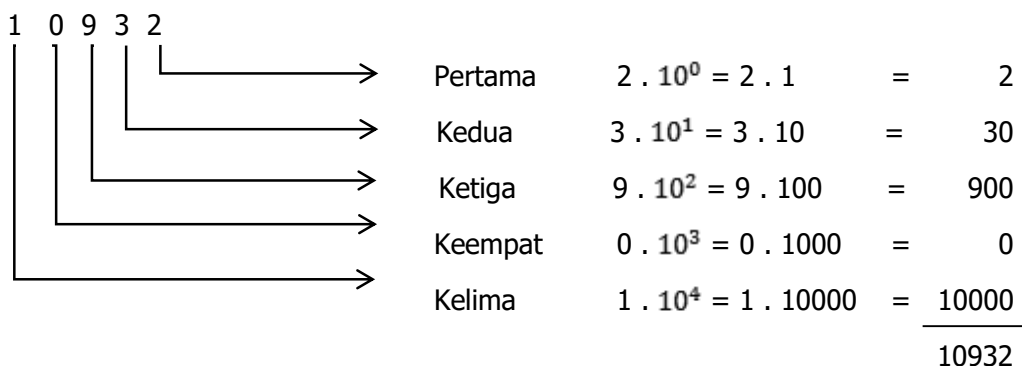
Sebagai contoh: Bilangan 123 dan 70 pada bilangan desimal dapat ditulis dalam bentuk ratusan, puluhan dan satuan atau secara matematik sebagai berikut:

1. (123)₁₀ = (1 x 10²) + (2 x 10¹) + (3 x 10⁰)
2. (78, 5)₁₀ = (7 x 10¹) + (8 x 10⁰) + (5 x 10⁻¹).

Setiap bilangan mempunyai nilai terbesar dan nilai terkecil. Pada angka 123, nilai terbesar adalah ratusan yaitu angka 1 yang berarti 100, selanjutnya puluhan yang bernilai 2 dan yang terkecil adalah satuan yang bernilai 3. Angka terbesar dari suatu bilangan pada sistem bilangan desimal disebut dengan Most Significant Digit (MSD). Sedangkan angka yang mempunyai harga terkecil disebut dengan Least Significant Digit (LSD).

Contoh:

Angka desimal 10932 (10932₍₁₀₎)



Kebiasaan sehari-hari harga suatu bilangan desimal dituliskan dalam bentuk yang mudah sbb:

$$\begin{aligned} 10932 &= 1 \cdot 10000 + 0 \cdot 1000 + 9 \cdot 100 + 3 \cdot 10 + 2 \cdot 1 \\ &= 1 \cdot 10^4 + 0 \cdot 10^3 + 9 \cdot 10^2 + 3 \cdot 10^1 + 2 \cdot 10^0 \end{aligned}$$

Konversi Bilangan

Sistem bilangan yang digunakan pada bidang teknik digital yaitu sistem bilangan desimal, biner, oktal dan hexadesimal. Masing-masing sistem bilangan digunakan sesuai dengan kebutuhan penggunaannya. Sistem bilangan desimal banyak digunakan untuk merepresentasikan bilangan yang dipakai dalam kehidupan sehari-hari. Sistem bilangan biner dipakai untuk memberikan masukan kepada sistem digital karena sistem ini hanya dapat menerima sinyal dalam bentuk digital. Selain sebagai input, bilangan biner juga dijadikan sebagai prinsip dalam pengolahan sinyal digital. Dengan bilangan biner maka sistem dapat dirancang dan dikembangkan dengan baik. Selain bilangan desimal dan biner, dalam aplikasi teknik digital juga diperlukan sistem bilangan hexadesimal yang banyak dipakai untuk mempermudah pembacaan dan penyusunan program dalam bilangan biner. Bilangan biner yang hanya terdiri dari angka 1 dan 0 akan sulit dibaca oleh manusia yang sudah terbiasa dengan sistem bilangan desimal, sehingga dengan bantuan sistem hexadesimal, bilangan biner yang panjang dapat dikelompokkan menjadi 4 bilangan biner yang kemudian dikonversi menjadi 1 digit bilangan hexadesimal.

Untuk memudahkan dalam mempelajari teknik digital dibutuhkan konversi bilangan dari desimal ke biner atau sebaliknya, dari biner ke hexadesimal dan sebaliknya, dari bilangan desimal ke hexadesimal dan sebaliknya.

2.2. Konversi Bilangan Biner ke Desimal dan Sebaliknya

Jika dalam kehidupan sehari-hari kita menggunakan sistem bilangan desimal, maka pada sistem digital menggunakan sistem bilangan biner sebagai basis operasinya. Sistem bilangan biner adalah sistem bilangan yang hanya menggunakan angka atau simbol 0 dan 1 untuk menuliskan suatu nilai.

Dengan demikian nilai dua (2_{10}) akan dituliskan sebagai 10_2 pada sistem bilangan biner. Konsep yang digunakan pada sistem bilangan biner sama dengan sistem bilangan desimal, yaitu setiap angka atau digit pada posisi yang berbeda akan mewakili nilai yang berbeda. Untuk lebih jelasnya dapat dilihat pada bahasan berikutnya.

2.2.1. Konversi Bilangan Biner ke Desimal

Untuk melakukan konversi dari sistem bilangan biner ke desimal, dilakukan dengan menghitung nilai dari masing-masing digit sesuai posisinya. Sebagai contoh, berapa nilai 1111_2 pada sistem bilangan desimal dapat dihitung sebagai contoh berikut.

$$\begin{aligned} 1111_2 &= 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ &= 8 + 4 + 2 + 1 \\ &= 15_{10} \end{aligned}$$

Jadi 1111_2 setara dengan 15_{10} (lima belas). Contoh lain 1010_{10} dapat dihitung sebagai berikut.

$$\begin{aligned} 1010_{10} &= 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 \\ &= 32 + 0 + 8 + 0 + 2 + 0 \\ &= 42_{10} \end{aligned}$$

Untuk bilangan pecahan, prinsip yang sama pada sistem bilangan desimal juga berlaku.

$$\begin{aligned} 11,11_2 &= 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} \\ &= 1 \times 2 + 1 \times 1 + 1 \times 0,5 + 1 \times 0,25 \\ &= 2 + 1 + 0,5 + 0,25 \\ &= 3,75_{10} \end{aligned}$$

$$\begin{aligned} 10,101_2 &= 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\ &= 1 \times 2 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\ &= 2 + 0 + 0,5 + 0 + 0,125 \\ &= 2,625_{10} \end{aligned}$$

2.2.2. Konversi Bilangan Desimal ke Biner

Secara sederhana kita bisa mencari bilangan biner yang ekuivalen dengan sebuah bilangan desimal adalah dengan menjabarkan bilangan

desimal tersebut sebagai penjumlahan ..., 2^4 , 2^3 , 2^2 , 2^1 , 2^0 , 2^{-1} , 2^{-2} , 2^{-3} , 2^{-4} , dan seterusnya. Sebagai contoh bilangan 11_{10} dapat dijabarkan sebagai penjumlahan 8 atau 2^3 , 2 atau 2^1 , serta 1 atau 2^0 . Dengan demikian 11_{10} setara dengan 1011_2 . Cara yang sistematis adalah dengan cara membagi bilangan desimal tersebut dengan 2 secara berurutan, sembari tetap mencatat sisa baginya, hingga diperoleh hasil bagi 0. Sebagai contoh adalah sebagai berikut.

$$\begin{array}{r} 11 \\ \hline 5 \quad \text{sisa 1} \\ \hline 2 \quad \text{sisa 1} \\ \hline 1 \quad \text{sisa 0} \\ \hline 0 \quad \text{sisa 1} \end{array}$$

Selanjutnya sisa bagi ditulis dari arah bawah, jadi 11_{10} (sebelas) setara dengan 1101_2 . Contoh lain 37_{10} dihitung sebagai berikut.

$$\begin{array}{r} 37 \\ \hline 18 \quad \text{sisa 1} \\ \hline 9 \quad \text{sisa 0} \\ \hline 4 \quad \text{sisa 1} \\ \hline 2 \quad \text{sisa 0} \\ \hline 1 \quad \text{sisa 0} \\ \hline 0 \quad \text{sisa 1} \end{array}$$

Jadi 37_{10} (tiga puluh tujuh) setara dengan 100101_2 .

2.3. Konversi Bilangan Desimal ke Hexadesimal dan Sebaliknya

Bilangan desimal mempunyai basis angka dari 0 sampai 9, sedangkan bilangan hexadesimal mempunyai basis angka dari 0 sampai dengan F. Bilangan 0 sampai dengan 9 bilangan hexadesimal mempunyai kesamaan dengan bilangan desimal. Bilangan 10 pada bilangan hexadesimal diubah menjadi A, 11 menjadi B, 12 menjadi C, 13 menjadi D, 14 menjadi E dan 15 menjadi F. Sehingga bilangan hexadesimal mempunyai nilai dari 0 – F. Selanjutnya adalah 10 sampai dengan FF dan seterusnya.

. Contoh cara konversi bilangan desimal ke hexadesimal dapat dilihat pada soal berikut ini.

Hitunglah berapa bilangan heksadesimal dari $(20)_{10}$?

Jawab :

$$20 : 16 = 1 \text{ Sisa } 4 \text{ LSD}$$

$$4 : 16 = 0 \text{ Sisa } 4 \text{ MSD}$$

$$\text{Jadi } (20)_{10} = (14)_{16}$$

Hitunglah berapa bilangan heksadesimal dari $(170)_{10}$?

Jawab :

$$170 : 16 = 10 \text{ Sisa } A \text{ LSD}$$

$$10 : 16 = 0 \text{ Sisa } A \text{ MSD}$$

$$\text{Jadi } (170)_{10} = (AA)_{16}$$

Hitunglah berapa bilangan heksadesimal dari $(170)_{10}$?

Jawab :

$$170 : 16 = 10 \text{ Sisa } A \text{ LSD}$$

$$10 : 16 = 0 \text{ Sisa } A \text{ MSD}$$

$$\text{Jadi } (170)_{10} = (AA)_{16}$$

Hitunglah berapa bilangan heksadesimal dari $(8659)_{10}$?

Jawab :

$$8659 : 16 = 541 \text{ Sisa } 3 \quad \text{LSD}$$

$$541 : 16 = 33 \text{ Sisa } D$$

$$33 : 16 = 2 \text{ Sisa } 1$$

$$2 : 16 = 0 \text{ Sisa } 2 \quad \text{MSD}$$

$$\text{Jadi } (8659)_{10} = (21D3)_{16}$$

Bagaimana konversi bilangan pecahan dalam desimal menjadi bilangan hexadesimal? Untuk bilangan pecahan maka dapat dilakukan dengan cara mengalikan dengan bilangan 16. Untuk lebih jelasnya dapat diperhatikan contoh-contoh berikut ini:

Hitunglah berapa bilangan pecahan heksadesimal dari $(0,6)_{10}$?

Jawab :

Representasi Heksadesimal

$$0,6 \times 16 = 0 \qquad 0,96$$

$$\text{Jadi } (0,6)_{10} = (0,96)_{16}$$

Hitunglah berapa bilangan pecahan heksadesimal dari $(0,75)_{10}$?

Jawab :

Representasi Heksadesimal

$$0,75 \times 16 = 0 \qquad 0,C$$

$$\text{Jadi } (0,75)_{10} = (0,C)_{16}$$

Konversi bilangan dari hexadesimal ke desimal dapat dilakukan dengan cara yang sama dengan konversi dari desimal ke hexadesimal. Cuma pembagiannya adalah angka 10. Untuk lebih jelasnya dapat dilihat beberapa contoh berikut ini

Hitunglah hasil konversi bilangan hexadesimal 80 ke dalam bilangan desimal

Jawab :

$$\begin{aligned} (80)_{16} &= 0 \times 16^0 + 8 \times 16^1 \\ &= 0 + 128 \\ &= 128 \end{aligned}$$

$$\text{Jadi } (80)_{16} = (128)_{10}$$

Hitunglah nilai bilangan desimal hasil konversi dari bilangan hexadesimal

2A7

Jawab :

$$\begin{aligned} (2A7)_{16} &= 7 \times 16^0 + A \times 16^1 + 2 \times 16^2 \\ &= 7 + 10 \times 16 + 2 \times 256 \\ &= 7 + 160 + 512 \\ &= 679 \end{aligned}$$

Hitunglah nilai bilangan desimal hasil konversi dari bilangan hexadesimal 2EF5

Jawab :

$$\begin{aligned}(2EF5)_{16} &= 5 \times 16^0 + F \times 16^1 + E \times 16^2 + 2 \times 16^3 \\ &= 5 + 15 \times 16 + 14 \times 256 + 2 \times 4096 \\ &= 5 + 240 + 3584 + 8192 \\ &= 12021\end{aligned}$$

Jadi $(2EF5)_{16} = (12021)_{10}$

Untuk konversi bilangan pecahan dapat dilakukan dengan mengalikan dengan pangkat minus. Contoh perhitungannya adalah sebagai berikut:

Hitunglah Berapa bilangan pecahan desimal dari bilangan pecahan hexadesimal berikut ini $(0,9)_{16}$

Jawab :

$$\begin{aligned}(0,4)_{16} &= 4 \times 16^{-1} \\ &= 1/16 \\ &= 0,25\end{aligned}$$

Jadi $(0,4)_{16} = (0,25)_{10}$

2.4. Konversi Bilangan Oktal dan Sebaliknya

Dalam rangkaian logika selain bilangan desimal dan bilangan biner, kita mengenal pula bilangan oktal. Bilangan oktal mempunyai 8 buah digit yaitu 0, 1, 2, 3, 4, 5, 6, 7. Dalam bilangan oktal tidak ada angka 8 dan 9, angka selanjutnya setelah angka 7 adalah angka 10, 11, 12 dan seterusnya. Agar lebih jelas perhatikan bilangan oktal dibawah ini.

0, 1, 2, 3, 4, 5, 6, 7 selanjutnya 10, 11, 12, 13, 14, 15, 16, 17, selanjutnya 20, 21, 22, 23, 24, 25, 26, 27 selanjutnya 30, 31, 32, 33, 34, 35, 36, 37 dan seterusnya.

Sama halnya dengan bilangan biner dan bilangan desimal, bilangan oktal mempunyai harga tempat seperti dibawah ini:

Oktal	8^4	8^3	8^2	8^1	8^0
Desimal	4096	512	64	8	1

2.4.1. Merubah Bilangan Desimal ke Oktal

Untuk merubah bilangan oktal menjadi bilangan desimal dapat dilakukan dengan harga tempat. Caranya adalah dengan menggunakan langkah-langkah sebagai berikut:

- Letakkan bilangan oktal dibawah harga tempatnya
- Kalikan masing-masing *digit* dari bilangan oktal sesuai dengan harga tempatnya
- Jumlahkan hasil perkalian masing-masing digit bilangan oktal

Contoh: Rubahlah bilangan oktal $(232)_8$ menjadi bilangan desimal

Jawab :

Oktal	8^2	8^1	8^0
Desimal	64	8	1

$$(2 \times 8^2) + (3 \times 8^1) + (2 \times 8^0) = 128 + 24 + 2 = (154)_{10}$$

2.4.2. Merubah Bilangan Oktal Ke desimal

Merubah bilangan desimal menjadi bilangan oktal dapat dilakukan dengan menggunakan harga tempat dan membagi 8 bilangan desimal terus menerus dan hasilnya dibaca dari bawah keatas.

Contoh: Rubahlah bilangan desimal $(678)_{10}$ menjadi bilangan oktal.

Soal diatas dapat diselesaikan dengan mudah dan sederhana dengan cara membagi 8 bilangan desimal secara terus menerus.

$$678/8 = 84 \text{ sisa } 6$$

$$84/8 = 10 \text{ sisa } 4$$

$$10/8 = 1 \text{ sisa } 2$$

$$1/8 = 0 \text{ sisa } 1 \text{ Dibaca dari bawah keatas} = (1246)_8$$

BAB 3

GERBANG LOGIKA DAN ALJABAR BOOLE

Suatu fungsi logika atau operasi logika adalah hubungan antara variabel biner pada masukan dan variabel biner pada keluaran dari suatu rangkaian digital yang mengikuti hukum aljabar Boolean. Di dalam aljabar Boolean semua hubungan logika antara variabel-variabel biner dapat dijelaskan oleh tiga operasi logika dasar yaitu :

- Operasi NOT (*negation*)
- Operasi AND (*conjunction*)
- Operasi OR (*disconjunction*)

Operasi operasi tersebut dijelaskan dalam tiga bentuk yaitu :

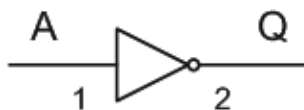
- Tabel fungsi (tabel kebenaran) yang menunjukkan keadaan semua variabel masukan dan keluaran untuk setiap kemungkinan.
- Simbol rangkaian untuk menjelaskan rangkaian digital.
- Persamaan fungsi.

3.1. Operasi logika NOT

Fungsi NOT adalah fungsi logika yang membalik sebuah variabel biner, misalnya jika masukannya adalah 0 maka keluarannya adalah 1. Gambar 1 memperlihatkan 3 macam bentuk penggambaran fungsi operasi NOT.

A	Q
0	1
1	0

Tabel Kebenaran



Simbol Rangkaian

$$Q = \overline{A}$$

Persamaan Fungsi

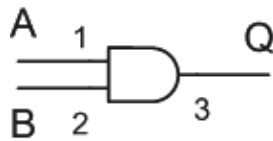
Gambar 1. Operasi NOT

3.2. Operasi logika AND

Operasi AND menghubungkan dua atau lebih variabel masukan mulai A, B, dan satu variabel keluaran Q. Variabel keluaran akan berlogika 1 hanya jika semua masukannya dalam keadaan 1. Gambar 2. menggambarkan 3 macam penggambaran fungsi operasi logika AND.

A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

Tabel Kebenaran



Simbol Rangkaian

$$Q = A \cdot B$$

Persamaan Fungsi

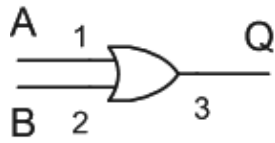
Gambar 2. Operasi AND

3.3. Operasi logika OR

Operasi OR juga menghubungkan dua atau lebih variabel masukan mulai A, B, ... dan satu variabel keluaran Q. Variabel keluaran akan berlogika 0 hanya jika semua masukannya dalam keadaan 0. Gambar 3. Menggambarkan 3 macam penggambaran fungsi operasi logika OR.

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

Tabel Kebenaran



Simbol Rangkaian

$$Q = A + B$$

Persamaan Fungsi

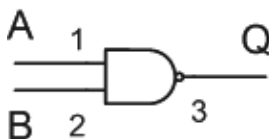
Gambar 3. Operasi OR

3.4. Operasi logika NAND

Operasi NAND merupakan kombinasi dua buah operasi logika dasar AND dan NOT. Masukan terdiri dari dua atau lebih variabel mulai dari A, B, ... dan satu variabel keluaran Q. Variabel keluaran akan berlogika 0 hanya jika semua masukannya dalam keadaan 1. Gambar 4 menggambarkan 3 macam penggambaran fungsi operasi logika NAND.

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

Tabel Kebenaran



Simbol Rangkaian

$$Q = \overline{A \cdot B}$$

Persamaan Fungsi

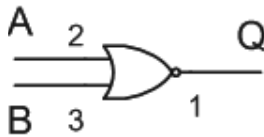
Gambar 4. Operasi NAND

3.5. Operasi logika NOR

Operasi NOR merupakan kombinasi dua buah operasi logika dasar OR dan NOT. Masukan terdiri dari dua atau lebih variabel mulai dari A, B, ... dan satu variabel keluaran Q. Variabel keluaran akan berlogika 1 hanya jika semua masukannya dalam keadaan 0. Gambar 5 menggambarkan 3 macam penggambaran fungsi operasi logika NOR.

A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

4. Tabel Kebenaran



Simbol Rangkaian

$$Q = \overline{A + B}$$

Persamaan Fungsi

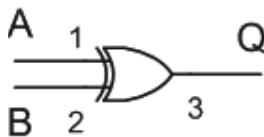
Gambar 5. Operasi NOR

4.1. Operasi logika X-OR

Operasi Exclusive OR biasanya disebut dengan XOR menghubungkan dua masukan variabel A dan B serta memiliki satu variabel keluaran Q. Gambar 6 menggambarkan 3 macam penggambaran fungsi operasi logika Exclusive OR .

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

Tabel Kebenaran



Simbol Rangkaian

$$Q = A \oplus B$$

Persamaan Fungsi

Gambar 6. Operasi XOR

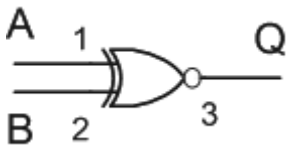
4.2. Operasi logika X-NOR

Operasi Exclusive Not OR biasanya disebut dengan X-NOR menghubungkan dua masukan variabel A dan B serta memiliki satu variabel keluaran Q. Gambar 7

menggambarkan 3 macam penggambaran fungsi operasi logika Exclusive X-NOR

A	B	Q
0	0	1
0	1	0
1	0	0
1	1	1

Tabel Kebenaran



Simbol Rangkaian

$$Q = \overline{A \oplus B}$$

Persamaan Fungsi

Gambar 6. Operasi X-NOR

3.9. Simulasi Rangkaian dengan Proteus

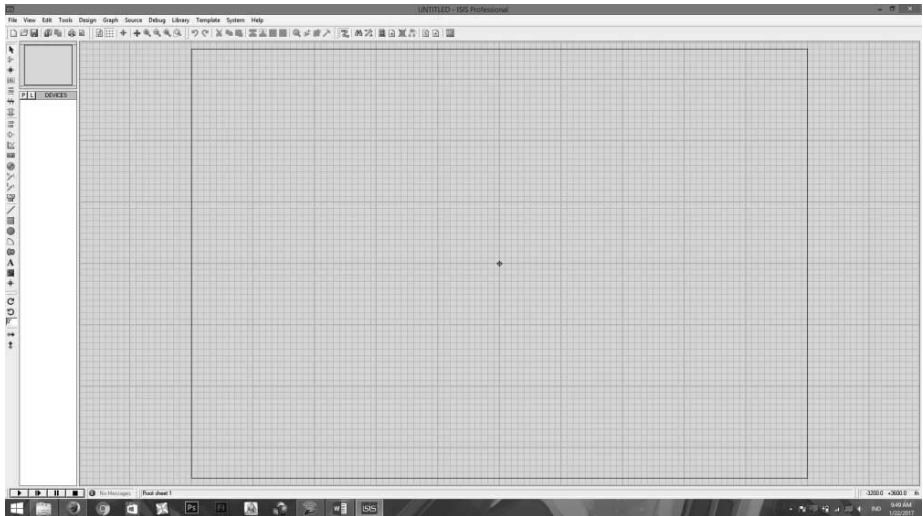
Simulasi AND

1. Buka program Proteus dengan mengklik dua kali *icon program ISIS 7 Professional*.



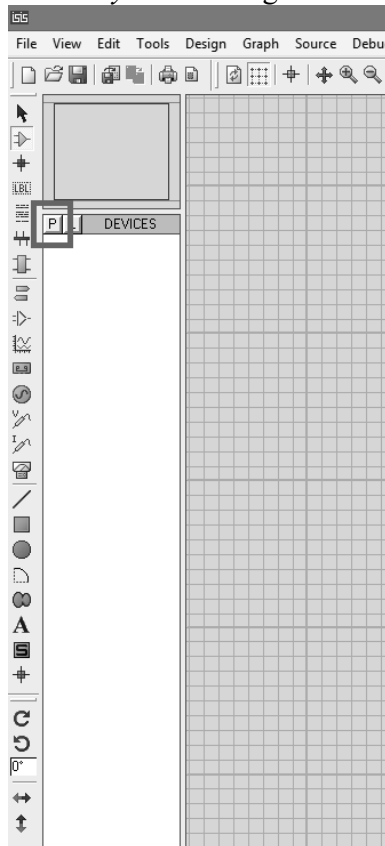
Gaambar 7. *Icon* Program ISIS 7 Professional

Lalu akan muncul tampilan seperti Gambar 8.



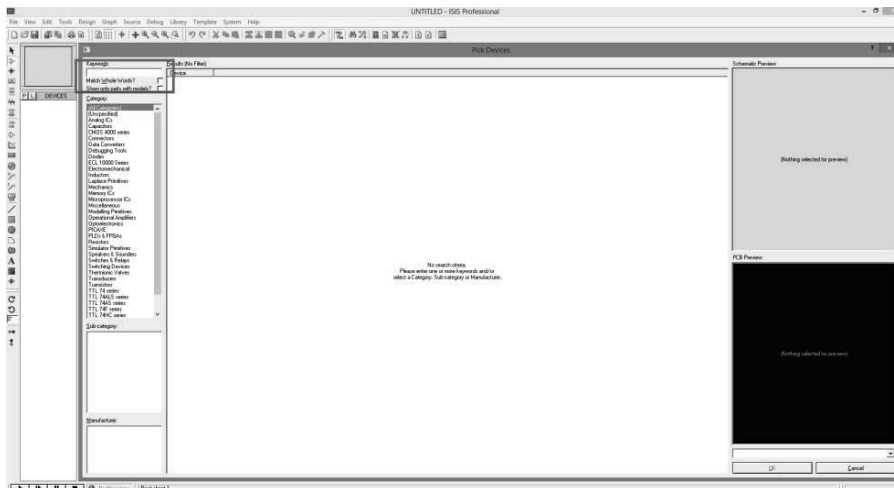
Gambar 8. Tampilan Utama Program ISIS 7 Proteus

2. Tekan tombol *pick from library* berlambang P berada kiri atas layar



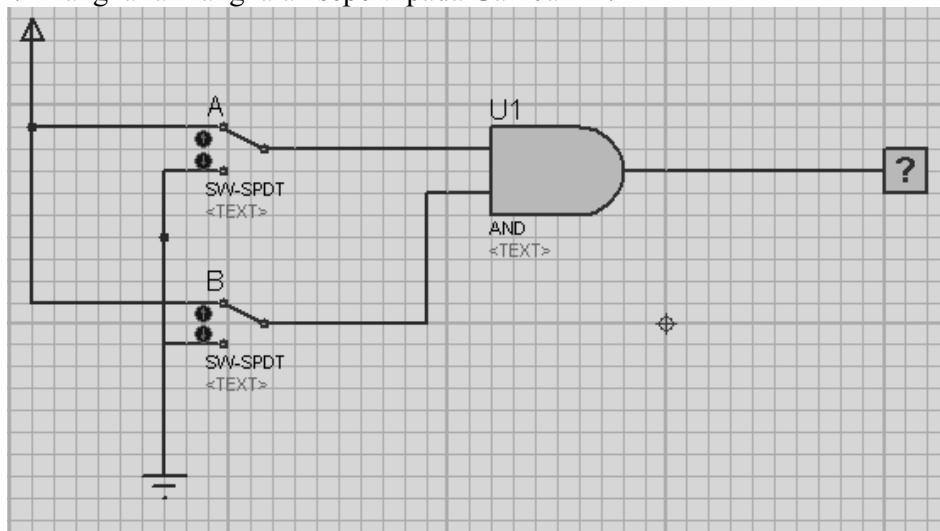
Gambar 9. Tekan Tombol *Pick from Library*

3. Carilah gerbang AND pada kotak pencari lalu OK, ulangi langkah ini untuk mencari komponen komponen yang diperlukan.



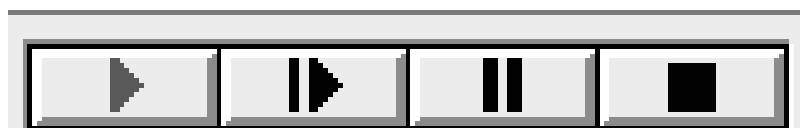
Gambar 10. Tempat penunjuk kotak pencari

4. Rangkailah rangkaian seperti pada Gambar 11.

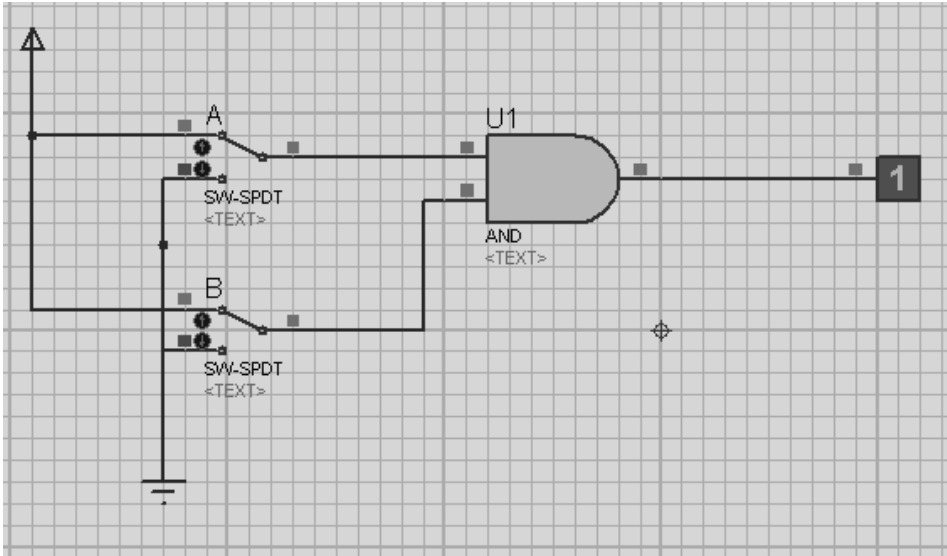


Gambar 11. Rangkaian Gerbang AND

5. Setelah selesai merangkai lalu tekan tombol *play* yang ada pada pojok kiri bawah panel proteus



Gambar 12. Tombol *Play* Proteus



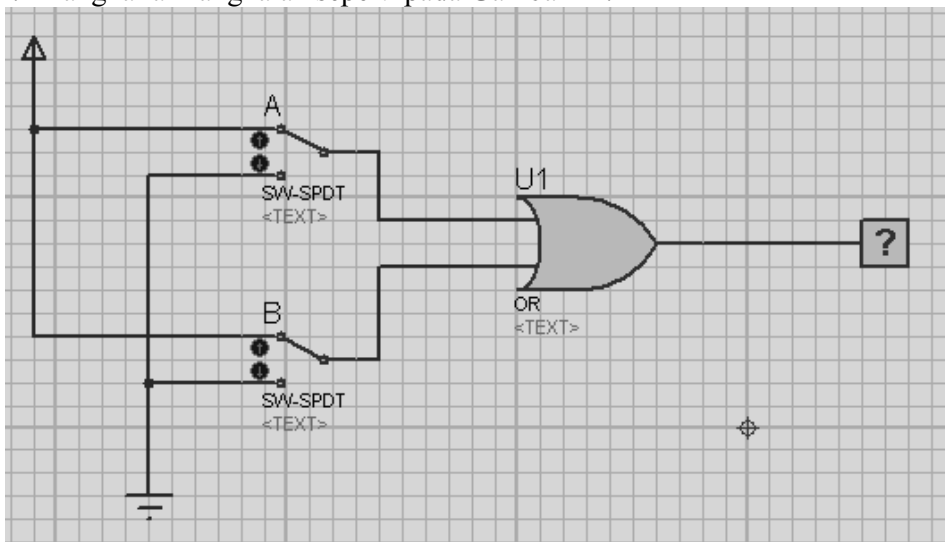
Gambar 13. Simulasi Rangkaian AND

A	B	X
0	0	0
0	1	
1	0	
1	1	

Tabel Kebenaran AND

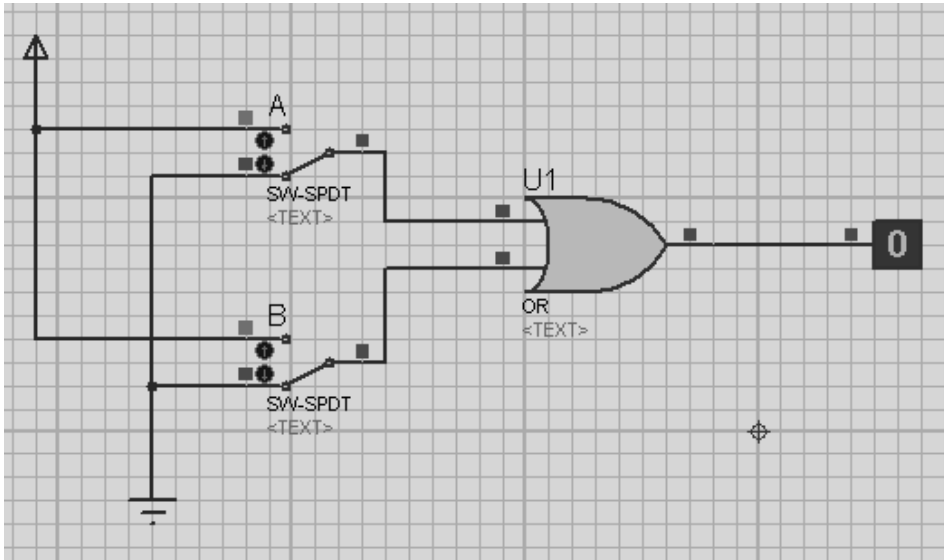
Simulasi OR

1. Ulangi langkah 1 sampai 3 pada simulasi AND dengan mengganti gerbang menjadi gerbang OR
2. Rangkailah rangkaian seperti pada Gambar 14.



Gambar 14. Rangkaian Gerbang OR

3. Setelah selesai merangkai rangkaian sesuai pada Gambar 14 simulasikan rangkaian dengan menggunakan tombol *Play*



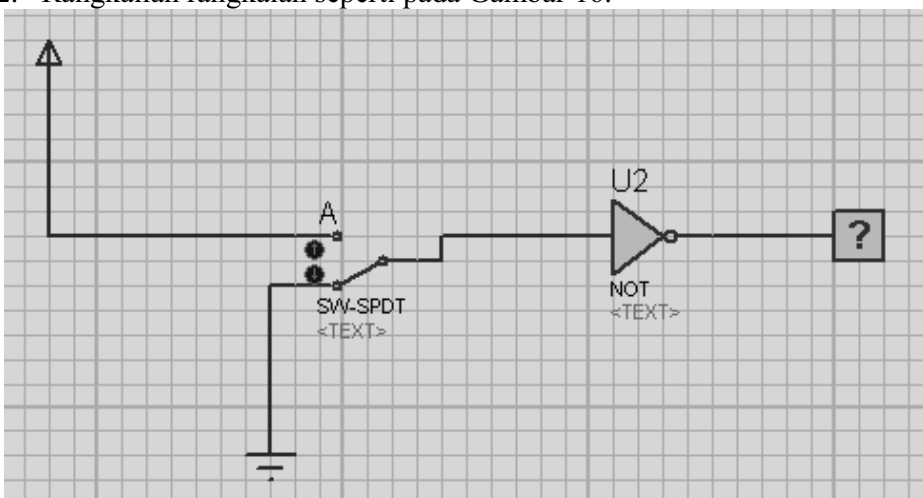
Gambar 15. Simulasi Rangkaian OR

A	B	X
0	0	0
0	1	
1	0	
1	1	

Tabel kebenaran Gerbang OR

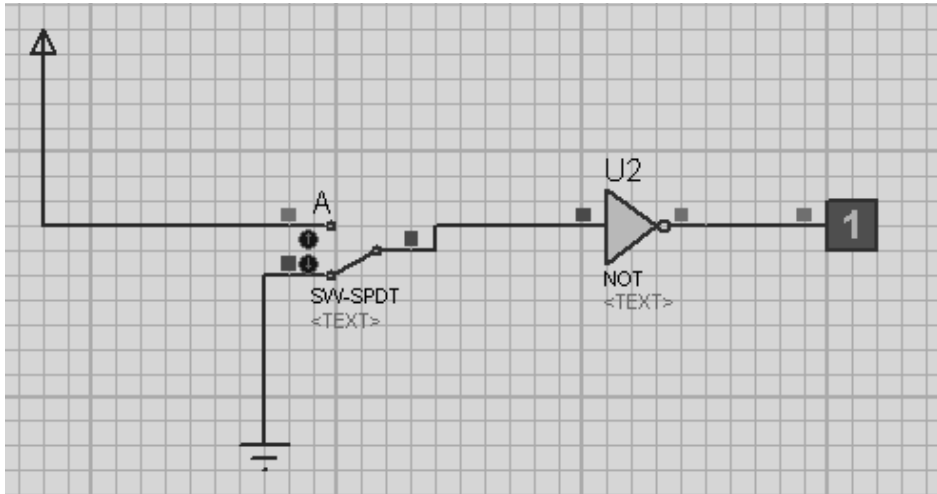
Simulasi NOT

1. Ulangi langkah 1 sampai 3 pada simulasi AND dengan mengganti gerbang menjadi gerbang NOT
2. Rangkailah rangkaian seperti pada Gambar 16.



Gambar 16. Gambar Rangkaian NOT

3. Setelah selesai merangkai rangkaian sesuai pada Gambar 16. simulasikan rangkaian dengan menggunakan tombol *Play*



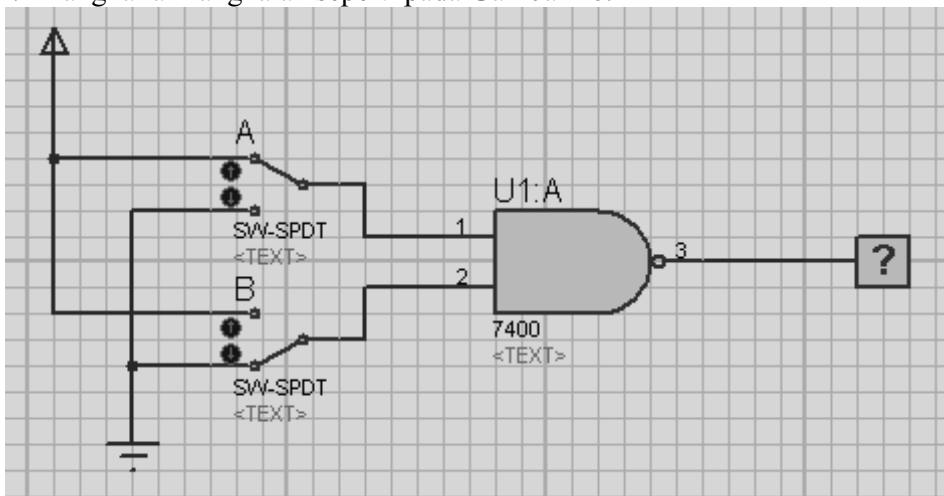
Gambar 17. Simulasi Rangkaian NOT

A	X
0	
1	

Tabel kebenaran Gerbang NOT

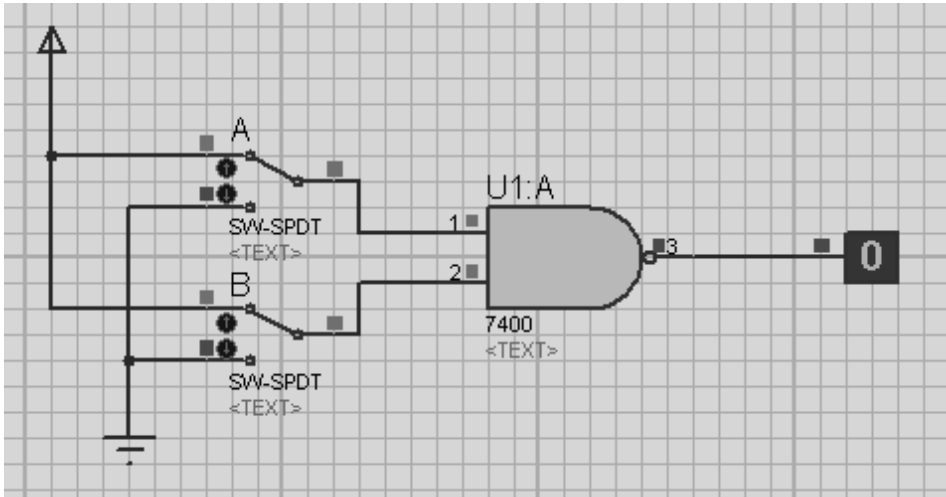
Simulasi Rangkaian NAND

1. Ulangi langkah 1 sampai 3 pada simulasi AND dengan mengganti gerbang menjadi gerbang NAND
2. Rangkailah rangkaian seperti pada Gambar 18.



Gambar 18. Rangkaian Gerbang NAND

3. Setelah selesai merangkai rangkaian sesuai pada Gambar 18 simulasikan rangkaian dengan menggunakan tombol *Play*



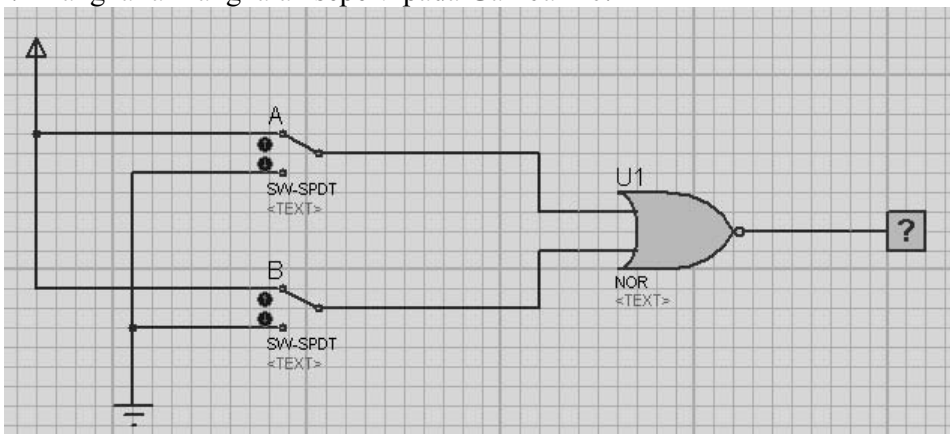
Gambar 19. Simulasi Rangkaian Gerbang NAND

A	B	X
0	0	
0	1	
1	0	
1	1	

Tabel Kebenaran NAND

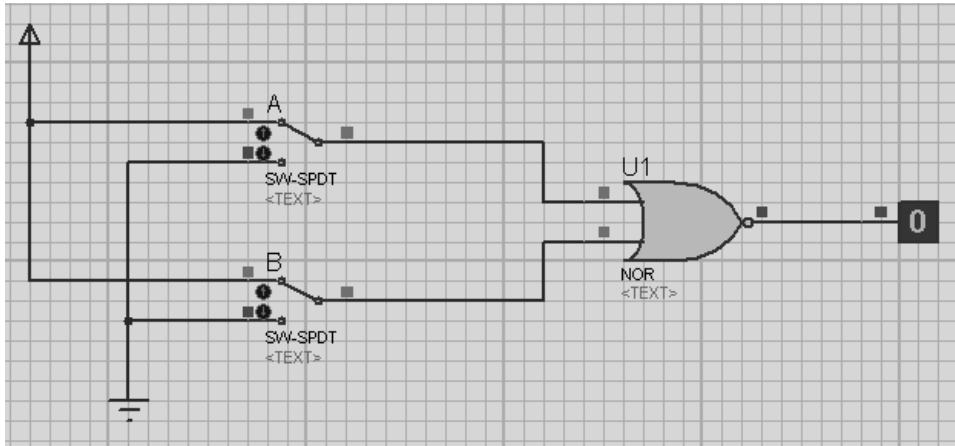
Simulasi Rangkaian NOR

1. Ulangi langkah 1 sampai 3 pada simulasi AND dengan mengganti gerbang menjadi gerbang NOR
2. Rangkailah rangkaian seperti pada Gambar 20.



Gambar 20. Rangkaian Gerbang NOR

3. Setelah selesai merangkai rangkaian sesuai pada Gambar 20 simulasikan rangkaian dengan menggunakan tombol *Play*



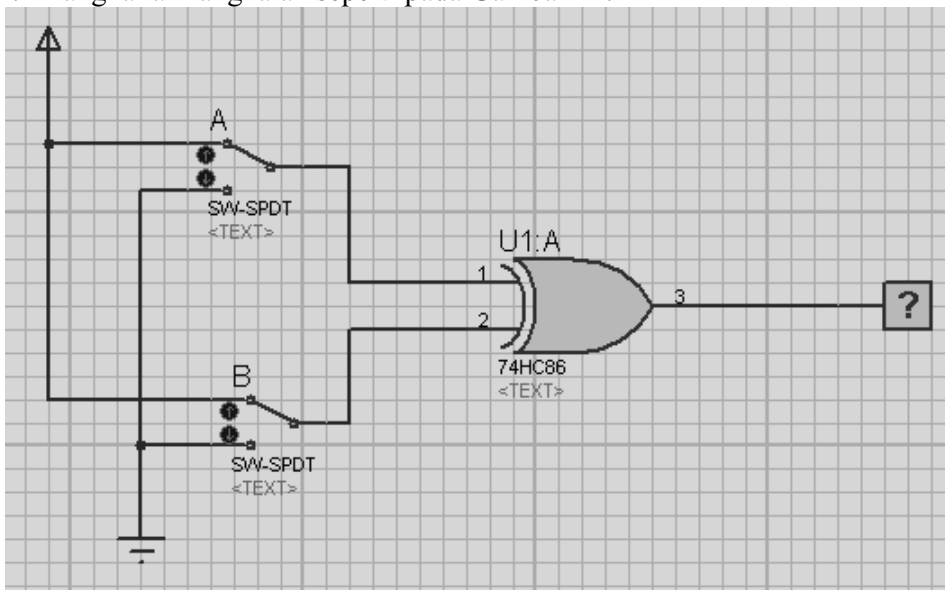
Gambar 21. Simulasi Rangkaian Gerbang NOR

A	B	X
0	0	
0	1	
1	0	
1	1	

Tabel Kebenaran NOR

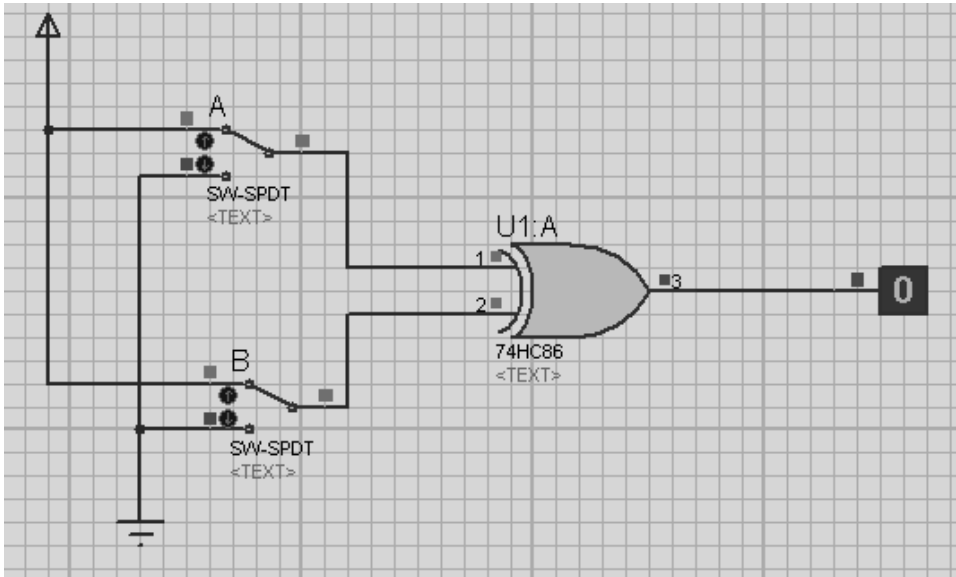
Simulasi Rangkaian EX-OR

1. Ulangi langkah 1 sampai 3 pada simulasi AND dengan mengganti gerbang menjadi gerbang EX-OR
2. Rangkailah rangkaian seperti pada Gambar 22.



Gambar 22. Rangkaian Gerbang EX-OR

3. Setelah selesai merangkai rangkaian sesuai pada Gambar 22 simulasikan rangkaian dengan menggunakan tombol *Play*



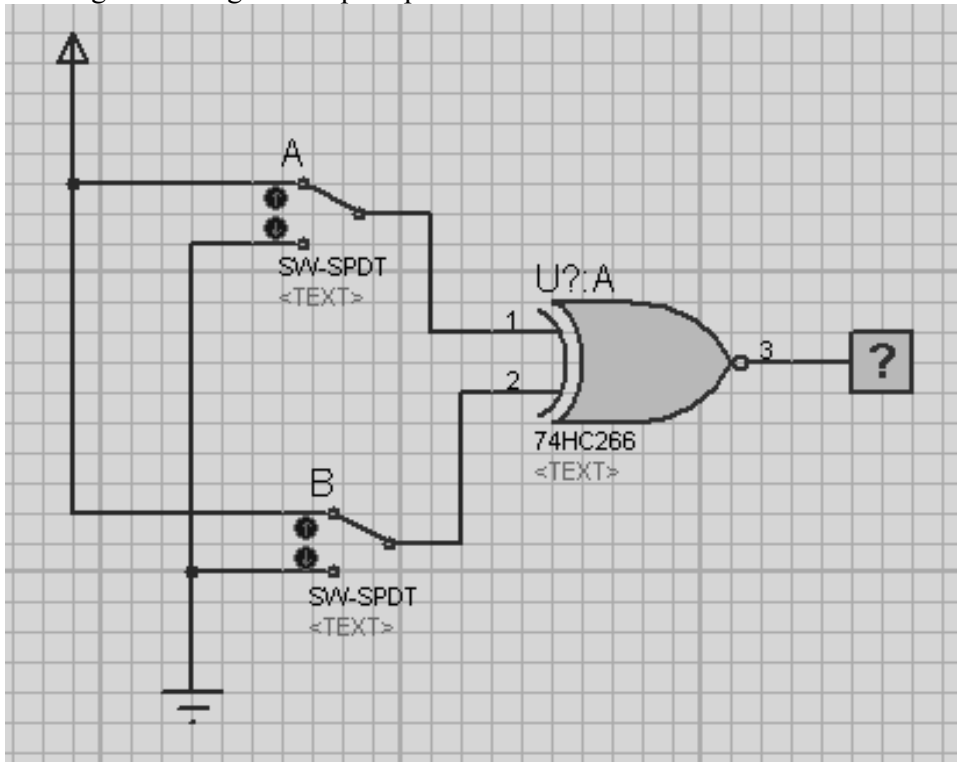
Gambar 23. Simulasi Rangkaian Gerbang EX-OR

A	B	X
0	0	
0	1	
1	0	
1	1	

Tabel Kebenaran EX-OR

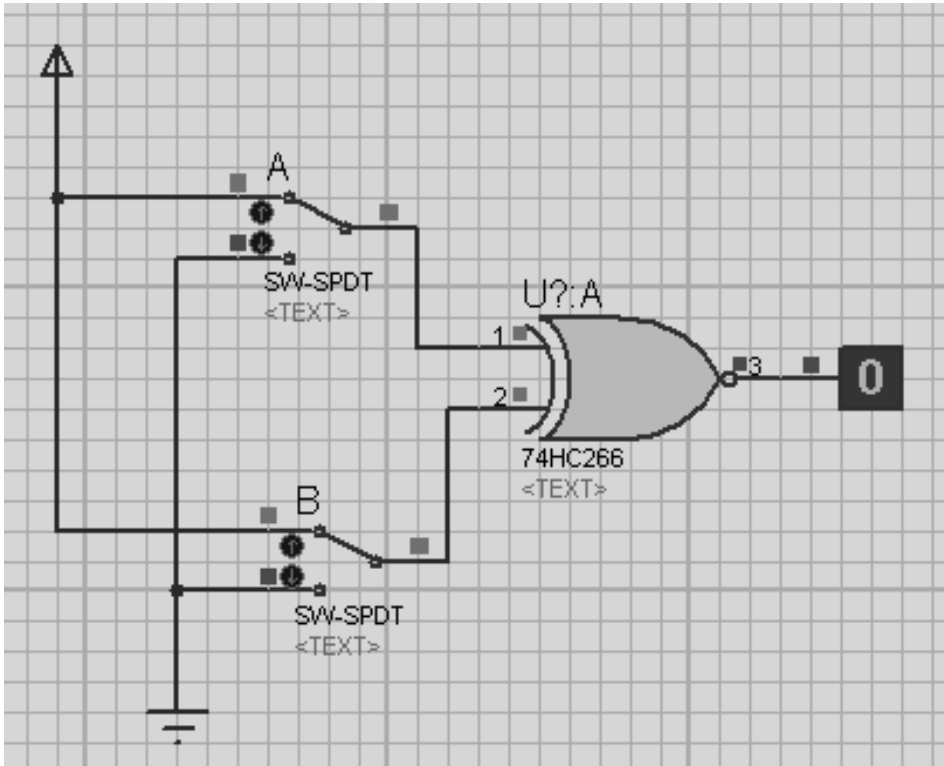
Simulasi Rangkaian EX-NOR

1. Ulangi langkah 1 sampai 3 pada simulasi AND dengan mengganti gerbang menjadi gerbang EX-NOR
2. Rangkailah rangkaian seperti pada Gambar 24.



Gambar 24. Rangkaian Gerbang EX-NOR

3. Setelah selesai merangkai rangkaian sesuai pada Gambar 24 simulasikan rangkaian dengan menggunakan tombol *Play*



Gambar 25. Simulasi Rangkaian Gerbang EX-NOR

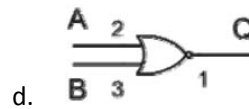
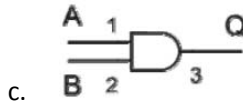
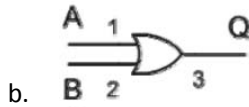
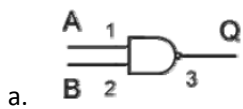
A	B	X
0	0	
0	1	
1	0	
1	1	

Tabel Kebenaran EX-NOR

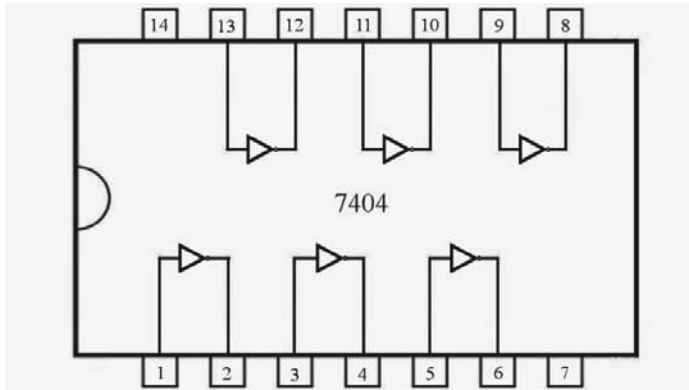
3.9. Latihan Soal

- Dalam suatu operasi logika hubungan variabel biner pada masukan dan keluaran dalam suatu rangkaian digital mengikuti hukum ...
 - Kirchoff I
 - Kirchoff II
 - Aljabar Boolean
 - Aljabar kalkulus
 - Operasi gerbang logika dasar
- Operasi logika dasar yang bersifat membalik sebuah variabel biner (negasi) yaitu ...
 - Operasi OR
 - Operasi AND
 - Operasi NOT
 - Operasi NEGATION
 - Operasi DISJUNCTION

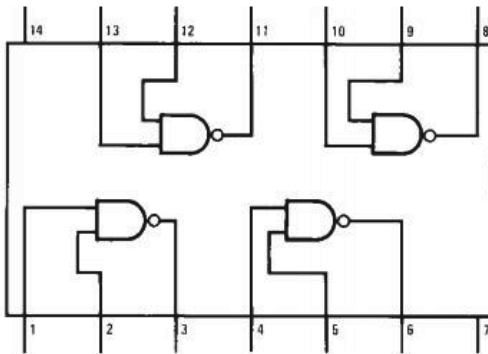
- Operasi logika NAND memiliki simbol rangkaian ...



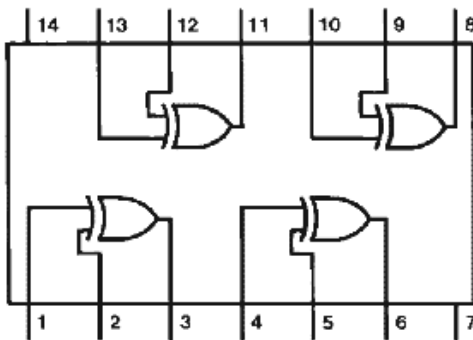
- Operasi logika AND memiliki persamaan fungsi ...
 - $Q = A \cdot B$
 - $Q = A + B$
 - $Q = \overline{A \cdot B}$
 - $Q = \overline{A + B}$
 - $Q = A \oplus B$
- Operasi logika NAND memiliki persamaan fungsi ...
 - $Q = A \cdot B$
 - $Q = A + B$
 - $Q = \overline{A \cdot B}$
 - $Q = \overline{A + B}$
 - $Q = A \oplus B$
- Operasi logika NOT menggunakan IC dengan nomor ...
 - 7400
 - 7402
 - 7404
 - 7408
 - 7432
- IC dengan nomor 74266 memiliki operasi logika ...
 - OR
 - AND
 - NOR
 - XOR
 - XNOR



8. IC 7404 seperti gambar diatas memiliki ground pada kaki nomor ...
- a. 1
 - b. 14
 - c. 2
 - d. 4
 - e. 7



9. IC diatas merupakan IC dengan nomor ...
- a. 7400
 - b. 7404
 - c. 7408
 - d. 7432
 - e. 7402



10. IC diatas memiliki masukan (input) pada kaki nomor ...
- a. 1 dan 2, 4 dan 5, 8 dan 9, 11 dan 12
 - b. 1 dan 2, 4 dan 5, 10 dan 9, 13 dan 12
 - c. 3, 6, 8, dan 11

- d. 7 dan 14
- e. 1, 7, 8 dan 14



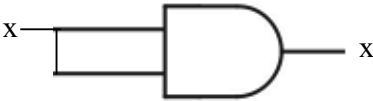
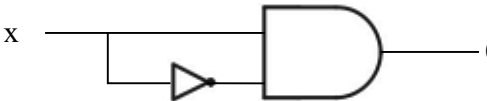
BAB 4.

RANGKAIAN LOGIKA KOMBINASI

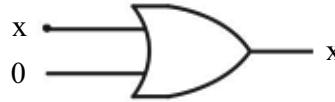
Untuk menjelaskan rangkaian digital digunakan persamaan fungsi yang disebut dengan aljabar Boolean. Fungsi-fungsi dari persamaan aljabar Boolean digambarkan dengan persamaan $M = f(X)$, dimana M merupakan keluaran dan X adalah masukan. Jumlah masukan bisa bervariasi, 1 atau lebih. Persamaan aljabar Boolean misalnya adalah $M=f(A,B,C) = \overline{(A + B)} \cdot C$. Persamaan tersebut adalah persamaan rangkaian digital dengan 3 masukan sehingga mempunyai 8 kemungkinan keadaan masukan.

Pada aljabar Boolean terdapat hukum-hukum aljabar Boolean yang memungkinkan kita menyederhanakan sebuah persamaan aljabar Boolean ataupun mencari bentuk persamaan aljabar Boolean yang setara namun lebih mudah diimplementasikan dengan gerbang-gerbang yang tersedia.

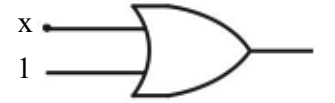
4.1. Hukum-hukum Aljabar Boolean

1. $X * 0 = 0$

2. $X * 1 = x$

3. $X * X = X$

4. $X * \bar{X} = 0$


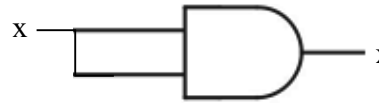
5. $X + 0 = x$



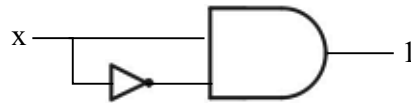
6. $X + 1 = 1$



7. $X + X = X$



8. $X + \bar{X} = 1$



9. $X + Y = Y + X$ (hukum komutatif)

10. $X * Z = Z * X$

11. $X + (Y + Z) = (X + Y) + Z = X + Y + Z$ (hukum asosiatif)

12. $X(YZ) = (XY)Z = XYZ$

13. $X(Y + Z) = (XY) + (XZ)$ (hukum distributif)
 $(W + X)(Y + Z) = WY + XY + WZ + XZ$

14. $X + XY = X$

15. $X + \overline{XY} = X + Y$
 $\overline{X} + XY = \overline{X} + Y$

16. $(\overline{X + Y}) = \overline{X} * \overline{Y}$ (hukum De Morgan)

17. $(\overline{X * Y}) = \overline{X} + \overline{Y}$

Contoh penggunaan hukum Aljabar Boolean untuk menyederhanakan sebuah persamaan adalah sebagai berikut.

Soal

Sederhanakan $z = (\overline{A + B})(A + B)$

Penyelesaian

Persamaan di atas dapat dikembangkan dengan menggunakan teorema (13)

$$z = \overline{A} \cdot A + \overline{A} \cdot B + B \cdot A + B \cdot B$$

Dari teorema (4), diketahui $\overline{A} \cdot A = 0$, serta $B \cdot B = B$ [teorema (3)]:

$$z = 0 + \overline{A} \cdot B + B \cdot A + B = \overline{A} \cdot B + B \cdot A + B$$

Dengan mengeluarkan variabel B [teorema (13)], diperoleh:

$$z = B(\bar{A} + A + 1)$$

Terakhir, dengan teorema (2) dan (6), diperoleh:

$$z = B$$

Sebenarnya selain menggunakan aljabar Boolean, juga terdapat metode peta Karnaugh yang dapat digunakan untuk menyederhanakan sebuah persamaan aljabar Boolean. Untuk metode ini Anda dapat mempelajari sendiri di buku-buk teks Teknik Digital

Teori De Morgan I

Teori ini menyatakan bahwa komplemen dari hasil penjumlahan akan sama dengan hasil perkalian dari masing-masing komplemen. Teori ini melibatkan gerbang OR dan AND. Penulisan dalam bentuk fungsi matematisnya sebagai berikut.

$$\overline{A + B} = \bar{A} \cdot \bar{B}$$

Teori De Morgan II

Teori ini menyatakan bahwa komplemen dari hasil kali akan sama dengan hasil penjumlahan dari masing-masing komplemen. Teori ini melibatkan gerbang AND dan OR. Penulisan dalam bentuk fungsi matematisnya sebagai berikut.

$$\overline{A \cdot B} = \bar{A} + \bar{B}$$

Prinsip Sum of Product

Prinsip **Sum of Product** adalah prinsip yang dapat digunakan untuk merancang sebuah rangkaian digital. Langkah-langkah dalam merealisasikan rangkaian digital adalah sebagai berikut:

- a. Tentukan tabel kebenaran dari kasus yang akan dibuat rangkaiannya.
- b. Tentukan persamaan aljabar dari tabel kebenaran dengan prinsip SUM of PRODUCT, setiap kombinasi masukan yang menghasilkan keluaran 1, akan bersesuaian dengan sebuah suku pada persamaan yang disusun.
- c. Buat rangkaian yang mengimplementasikan persamaan tersebut.

Contoh penggunaan prinsip ini adalah sebagai berikut.

- Tabel kebenaran :

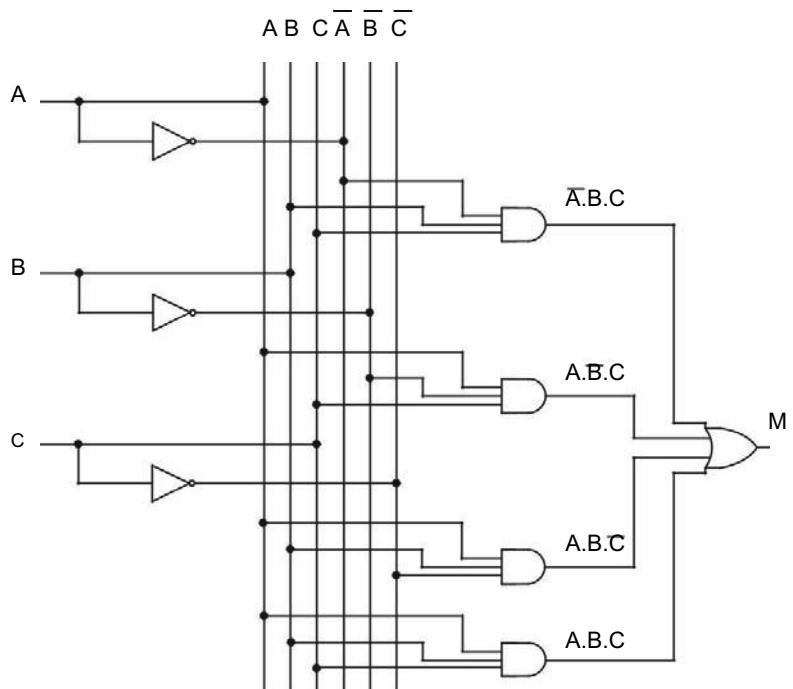
INPUT			OUTPUT
A	B	C	
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

- Dengan menggunakan prinsip Sum of Product, kita dapat melihat untuk tabel kebenaran di atas, persamaan M akan mempunyai 4 suku, masing-masing suku bersesuaian dengan kombinasi di mana hasil M bernilai 1, dengan demikian

kombinasi A=0, B=1, dan C=1 akan menghasilkan suku pertama, yaitu $\bar{A}BC$. Demikian untuk suku ke-2, suku ke-3, dan suku ke-4, maka diperoleh persamaan sebagai berikut.

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

- Rangkaian digital :

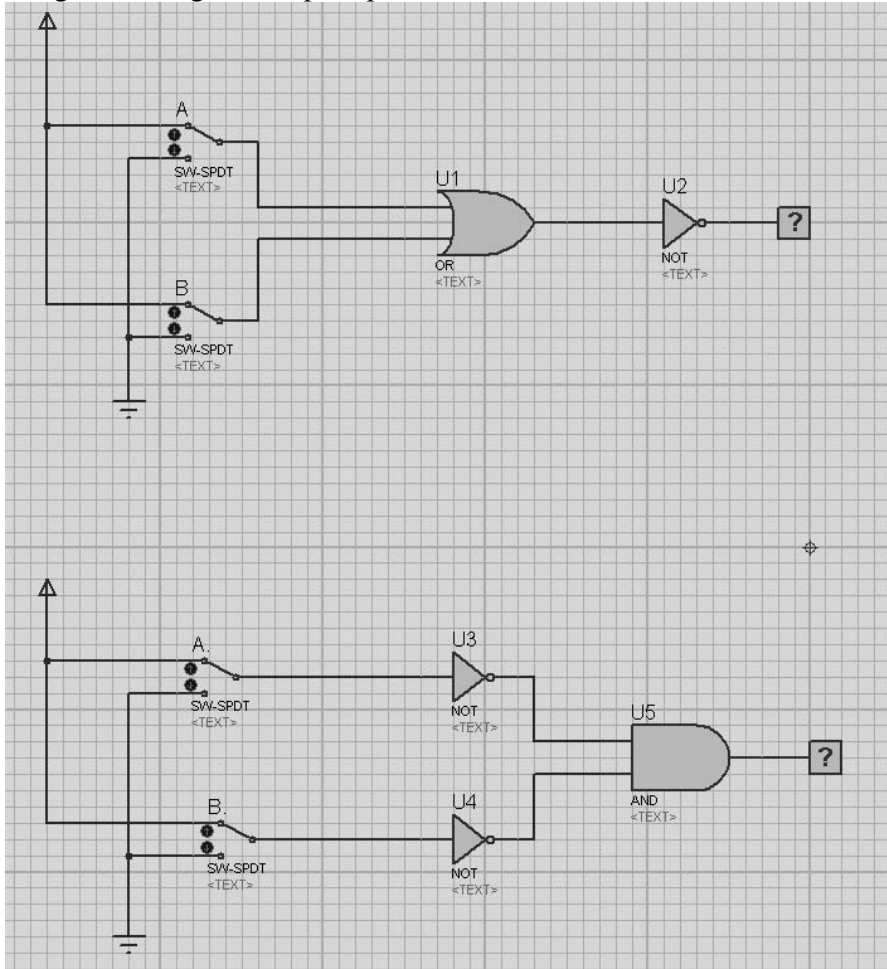


Gambar 1

4.2. Simulasi Rangkaian dengan Proteus

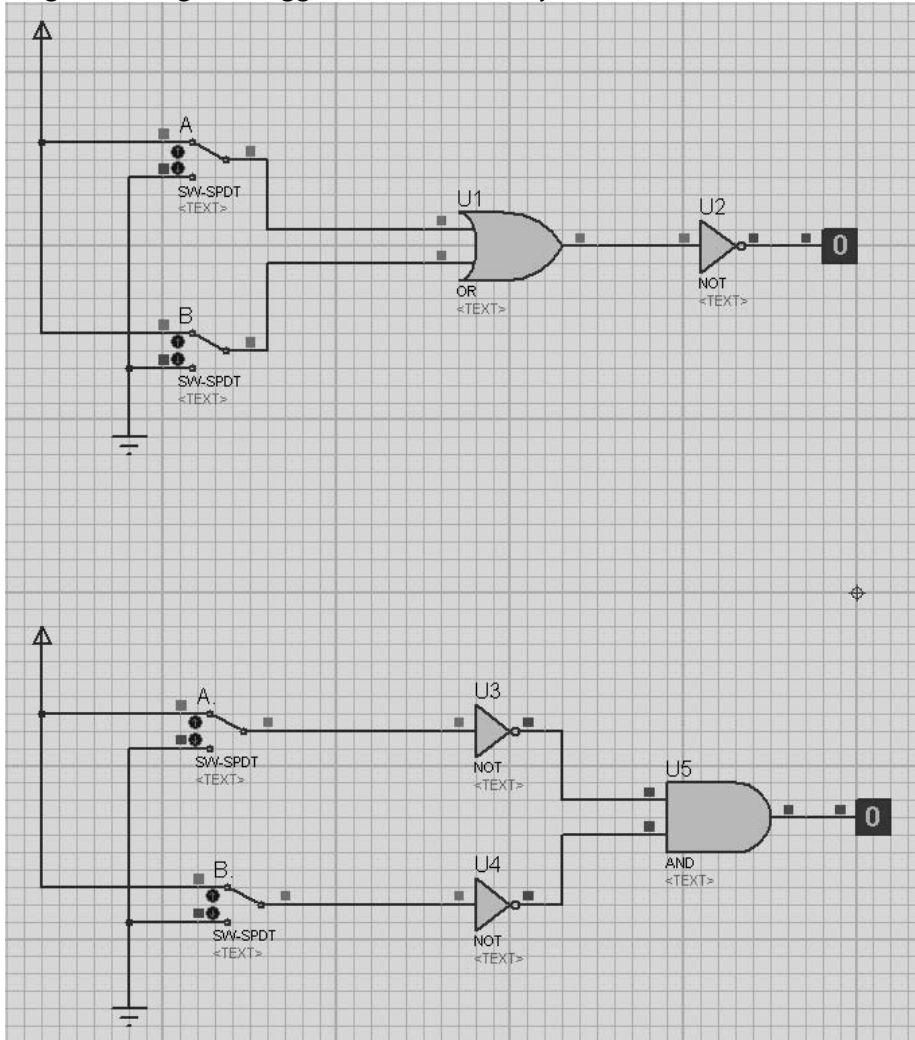
Simulasi Rangkaian Percobaan 1

1. Ulangi langkah 1 sampai 3 pada simulasi AND di BAB sebelumnya dengan mengganti gerbang menjadi gerbang NAND
2. Rangkailah rangkaian seperti pada Gambar 2.



Gambar 2. Rangkaian Percobaan 1

3. Setelah selesai merangkai rangkaian sesuai pada Gambar 2 simulasikan rangkaian dengan menggunakan tombol *Play*



Gambar 3. Simulasi Percobaan 1

Tabel 1a. Percobaan fungsi rangkaian $\overline{A + B} = \overline{A} \cdot \overline{B}$

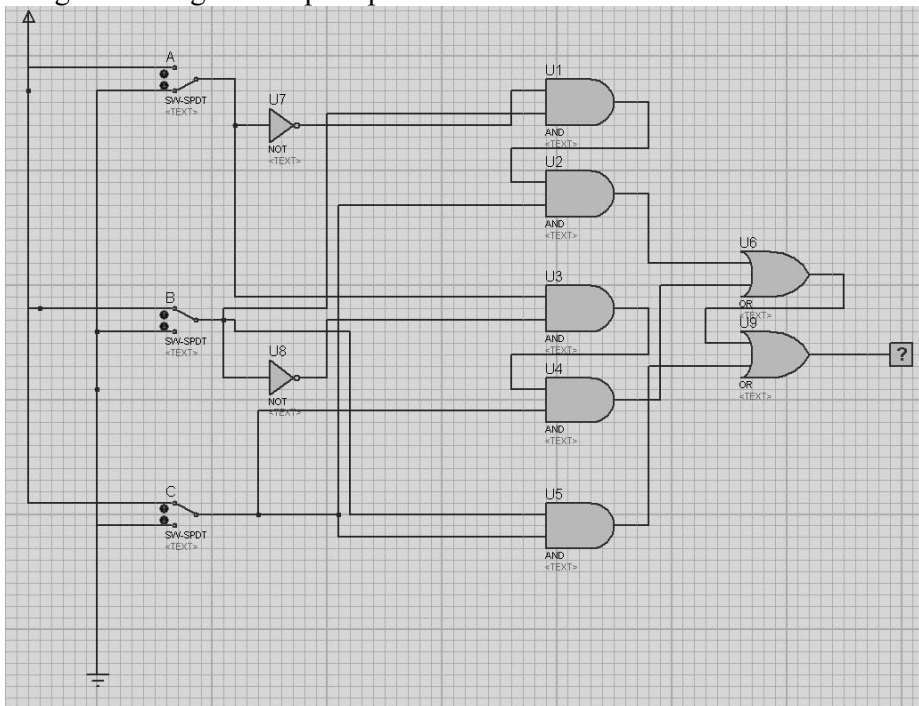
INPUT		OUTPUT	
A	B	$A + B$	$\overline{A} \cdot \overline{B}$
0	0		
0	1		
1	0		
1	1		

Tabel 1b. Percobaan fungsi rangkaian $\overline{A \cdot B} = \overline{A} + \overline{B}$

INPUT		OUTPUT	
A	B	$A \cdot B$	$\overline{A \cdot B}$
0	0		
0	1		
1	0		
1	1		

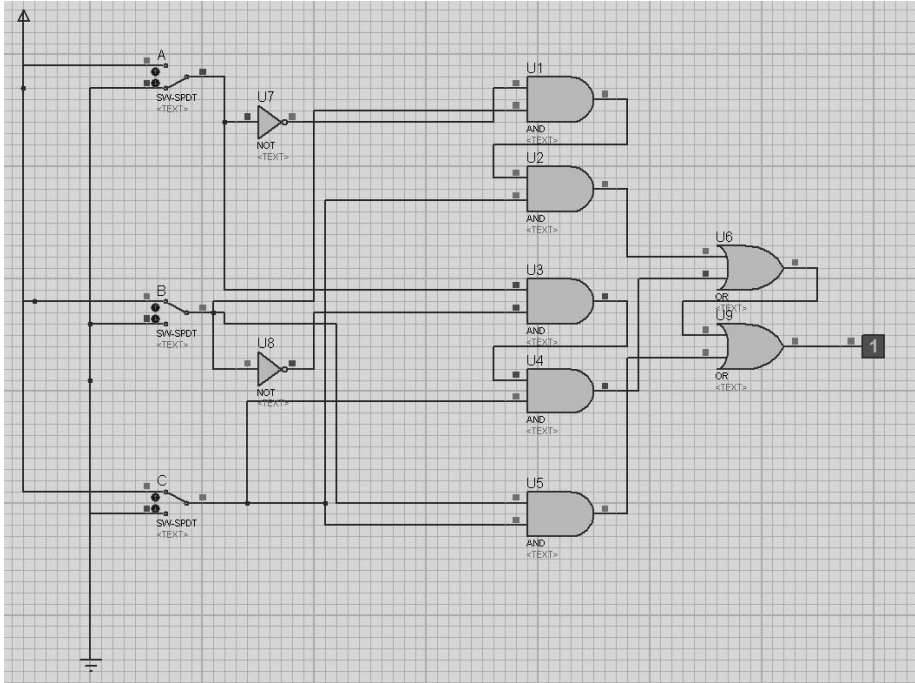
Simulasi Rangkaian Percobaan 2

1. Ulangi langkah 1 sampai 3 pada simulasi AND di BAB sebelumnya dengan mengganti gerbang menjadi gerbang NAND
2. Rangkailah rangkaian seperti pada Gambar 2.



Gambar 4. Rangkaian Percobaan 2

3. Setelah selesai merangkai rangkaian sesuai pada Gambar 2 simulasikan rangkaian dengan menggunakan tombol *Play*

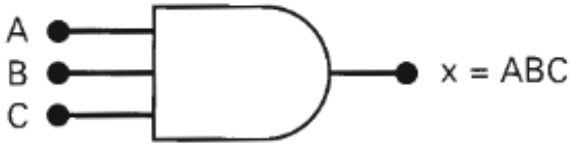


Gambar 5. Simulasi Percobaan 2

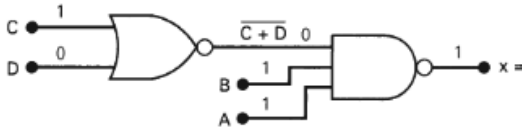
Tabel 2. Percobaan fungsi rangkaian $M = \bar{A} \cdot B \cdot C + A \cdot \bar{B} \cdot C + B \cdot \bar{C}$

INPUT			OUTPUT			
A	B	C	$A \cdot B \cdot C$	$A \cdot \bar{B} \cdot C$	$B \cdot \bar{C}$	M
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

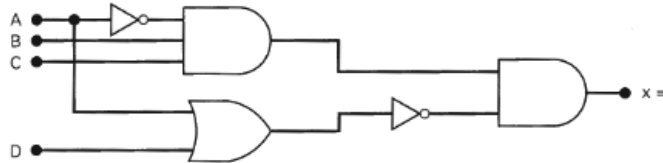
4.3. Latihan Soal



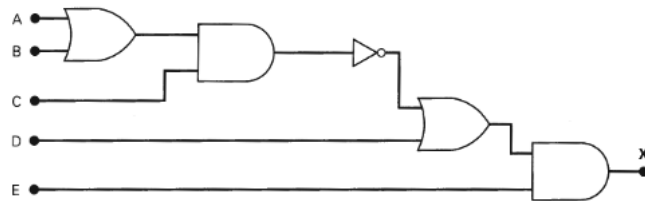
1. persamaan diatas menghasilkan output ABC bernilai high dengan input ...
- $A = 1, B = 1, C = 1$
 - $A = 1, B = 0, C = 0$
 - $A = 0, B = 1, C = 0$
 - $A = 0, B = 0, C = 1$
 - Semua jawaban benar



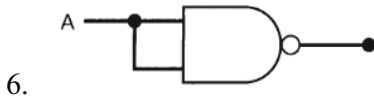
2. pada gambar rangkaian diatas nilai logika x ...
- $\overline{AB(C+D)}$
 - $\overline{A+B(CD)}$
 - $\overline{AB(CD)}$
 - $\overline{AB(\overline{CD})}$
 - $\overline{AB(CD)}$



3. pada gambar diatas nilai logika x ...
- $ABC(\overline{A+D})$
 - $\overline{ABC(A+D)}$
 - $\overline{ABC(A+D)}$
 - $A+B+C(\overline{A.D})$
 - $\overline{A+B+C(A.D)}$

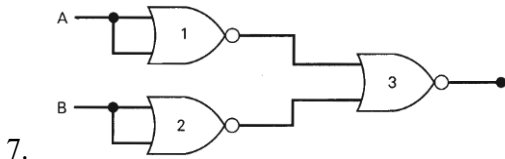


4. pada gambar rangkaian diatas nilai x ...
- $[D + (\overline{A+B})C].E$
 - $[D + (\overline{A+B})\overline{C}].E$
 - $[D + (\overline{A+B+C})].E$
 - $[D(\overline{A+B+C})].E$
 - $[D(\overline{A+B+C})].\overline{E}$
5. Pada soal diatas total IC yang dibutuhkan adalah ...
- 1
 - 2
 - 3
 - 4
 - 5



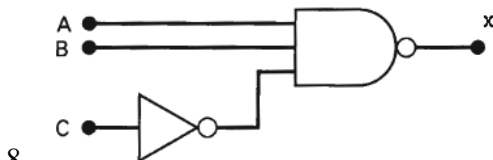
gambar diatas merupakan NAND gate yang dapat difungsikan sebagai logika ...

- a. NOT
- b. AND
- c. OR
- d. XOR
- e. XNOR



gambar diatas merupakan NAND gate yang dapat difungsikan sebagai logika ...

- a. NOT
- b. AND
- c. OR
- d. NAND
- e. XNOR



nilai x pada gambar diatas adalah ...

- a. $A + B + C$
- b. $\bar{A} + B + C$
- c. $\bar{A} + \bar{B} + C$
- d. $\bar{A} + \bar{B} + \bar{C}$
- e. $\overline{A + B + C}$

9. Dengan menggunakan prinsip Sum of Product tabel berikut memiliki persamaan ...

Input			Output M
A	B	C	
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

- a. $M = \bar{A}BC + A\bar{B}C + \bar{A}\bar{B}C + ABC$
- b. $M = \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C + ABC$
- c. $M = \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C + \bar{A}\bar{B}C$
- d. $M = \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C + ABC$
- e. $M = \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C + ABC$

10. Sama seperti soal nomor 14 persamaan yang didapat dari tabel berikut adalah ...

Input			Output M
A	B	C	
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

- $M = \overline{A}BC + A\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\overline{C}$
- $M = \overline{A}\overline{B}C + \overline{A}B\overline{C} + A\overline{B}C + A\overline{B}\overline{C}$
- $M = \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + A\overline{B}\overline{C} + A\overline{B}C$
- $M = \overline{A}\overline{B}C + \overline{A}B\overline{C} + A\overline{B}C + \overline{A}\overline{B}\overline{C}$
- $M = \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + A\overline{B}C + A\overline{B}\overline{C}$

BAB 5.

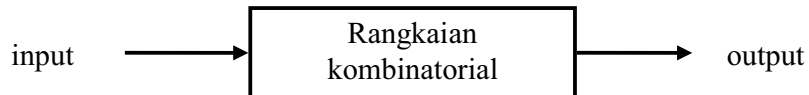
RANGKAIAN FLIP-FLOP

5.1. Rangkaian Kombinatorial dan Rangkaian Sekuensial

Rangkaian digital dapat dibagi menjadi 2 jenis berdasarkan hubungan *input* dan *output*, yaitu:

1. **Rangkaian kombinatorial**, yaitu rangkaian yang *output*-nya suatu saat hanya tergantung pada input pada saat itu. Kombinasi input yang sama akan selalu menghasilkan kombinasi *output* yang sama pula.

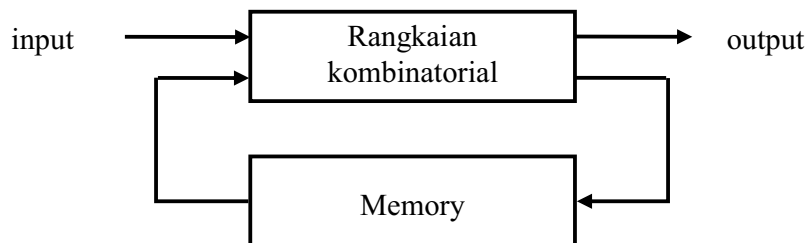
Contoh rangkaian kombinatorial adalah rangkaian yang dibahas pada Bab 4



Gambar 5.1. Rangkaian kombinatorial

2. **Rangkaian sekuensial**, yaitu rangkaian yang *output*-nya tidak hanya tergantung pada *input* saat itu, tetapi juga tergantung pada keadaan (*state*) dari rangkaian tersebut, atau pada ingatan (*memory*) dari rangkaian tersebut. Pada dasarnya jenis rangkaian ini dapat dilihat sebagai rangkaian kombinatorial yang dilengkapi *memory*, seperti ilustrasi pada Gambar 2.

Contoh rangkaian sekuensial adalah rangkaian flip-flop yang akan dibahas di Bab ini dan juga rangkaian pencacah (*counter*) dan register yang akan dibahas pada Bab 6.



Gambar 5.2. Rangkaian sekuensial

5.2. Flip-Flop RS

Flip-flop adalah sirkuit elektronik yang memiliki dua arus stabil dan dapat digunakan untuk menyimpan informasi. Sebuah flip flop merupakan multivibrator bistabil. Dikatakan multivibrator bistabil karena kedua tingkat tegangan keluaran pada multivibrator tersebut adalah stabil dan hanya mengubah situasi tingkat tegangan keluaran saat dipicu (trigger). Flip-flop mempunyai dua *output* (Keluaran) yang salah satu outputnya merupakan komplemen *output* yang lain. Keluaran flip-flop berupa tegangan rendah (0) atau tinggi (1). Untuk mengubah suatu tegangan dari tinggi ke rendah atau sebaliknya, rangkaian yang bersangkutan harus di-*drive* oleh suatu masukan yang disebut *trigger*. Sampai datang pemicu, tegangan keluaran tetap rendah atau tinggi untuk selang waktu yang tak terbatas. Ada beberapa jenis Flip-flop yang dibedakan berdasarkan perilakunya, salah satunya adalah Flip-flop RS. Flip-flop RS kadang disebut juga Flip-flop SC (Set-Clear).

Tabel 5.1. Tabel Kebenaran Flip-flop RS

R	S	Q	\bar{Q}
0	0	Nilai terakhir	
0	1	1	0
1	0	0	1
1	1	Terlarang	

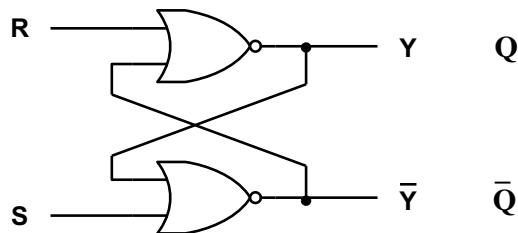
Tabel 1 meringkas kemungkinan-kemungkinan masukan/keluaran bagi flip-flop RS (Reset-Set) :

- Kondisi masukan yang pertama adalah $RS = 0-0$, Ini berarti tidak diterapkan pemicu. Dalam hal ini keluaran Q mempertahankan nilai terakhir yang dimilikinya ($Q_n = Q_{n-1}$)
- Kondisi masukan yang kedua adalah $RS = 0-1$ berarti bahwa suatu pemicu diterapkan pada masukan S (**Set**). Seperti kita ketahui, hal ini mengeset flip-flop dan menghasilkan keluaran Q bernilai 1.

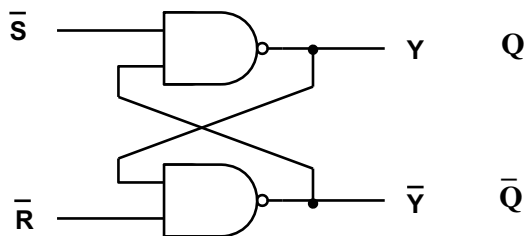
- Kondisi masukan yang ketiga adalah $RS = 1-0$ ini menyatakan bahwa suatu pemacu diterapkan pada masukan R (**Reset**). Keluaran Q yang dihasilkan adalah 0.
- Kondisi masukan $RS = 1-1$ merupakan masukan terlarang. Kondisi ini berarti menerapkan suatu pemacu pada kedua masukan S dan R pada saat yang sama. Hal ini merupakan suatu pertentangan karena mengandung pengertian bahwa kita berupaya untuk memperoleh keluaran Q yang secara serentak sama dengan 1 dan sama dengan 0.

5.2.1. Rangkaian Flip-flop RS

Flip-flop RS atau SR (Set-Reset) merupakan dasar dari flip-flop jenis lain. Flip-flop ini mempunyai 2 masukan: satu disebut S (SET) yang dipakai untuk menyetel (membuat keluaran flip-flop berkeadaan 1) dan yang lain disebut R (RESET) yang dipakai untuk me-reset (membuat keluaran berkeadaan 0). Flipflop RS dapat dibentuk dari dua gerbang NOR atau dua gerbang NAND seperti yang ditunjukkan pada gambar 1 dan gambar 2.



Gambar 5.3. Rangkaian flip-flop RS dengan gerbang NOR



Gambar 5.4. Rangkaian flip-flop RS dengan gerbang NAND

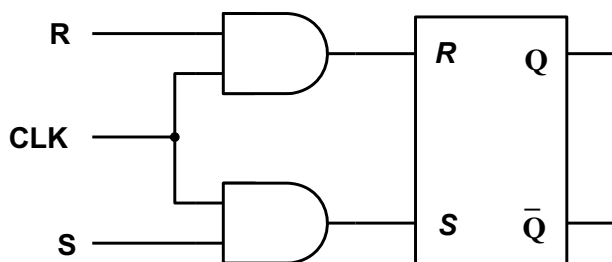
Konsep Flip-flop RS yang harus diingat adalah sebagai berikut.

- R dan S keduanya rendah berarti keluaran Q tetap berada pada keadaan terakhirnya secara tak terbatas akibat adanya aksi penggrendelan internal.

- Masukan S yang tinggi mengeset keluaran Q ke 1, kecuali jika keluaran ini memang telah berada pada keadaan tinggi. Dalam hal ini keluaran tidak berubah, walaupun masukan S kembali ke keadaan rendah.
- Masukan R yang tinggi mereset keluaran Q ke 0, kecuali jika keluaran ini memang telah rendah. Keluaran Q selanjutnya tetap pada keadaan rendah, walaupun masukan R kembali ke keadaan rendah.
- Masukan R dan S keduanya tinggi pada saat yang sama adalah terlarang karena merupakan pertentangan (Kondisi ini mengakibatkan masalah pacu, yang akan dibahas kemudian).

5.2.2. Flip-flop RS Berdetak

Pengembangan lebih lanjut dari Flip-flop RS adalah *Clocked RS FF*. Masukan R dan S pada rangkaian RS flip-flop dapat disinkronisasi dengan menambahkan masukan klok (detak) yang dapat dilihat pada gambar 3. Perbedaan cara kerja dari *Clocked RS FF* adalah bahwa flip-flop akan mengalami perubahan seperti pada RS FF menunggu sinyal clock aktif (logika tinggi) atau keluaran Q pada RS flip-flop tidak dapat segera merespo masukan S dan R sebelum ada masukan klok.



Gambar 5.5. Rangkaian flip-flop RS Berdetak

5.3. Flip-flop D

Flip-flop D berasal dari kata Data. Flip-flop ini hanya mempunyai satu masukan yaitu D. Jenis flip-flop ini sangat banyak dipakai sebagai sel memori dalam komputer. Data flip-flop merupakan pengembangan dari RS flip-flop, pada D flip-flop kondisi *output* terlarang (tidak tentu) tidak lagi terjadi. Data flip-flop sering juga disebut dengan istilah D-FF. *Input* atau masukan pada RS flip-flop

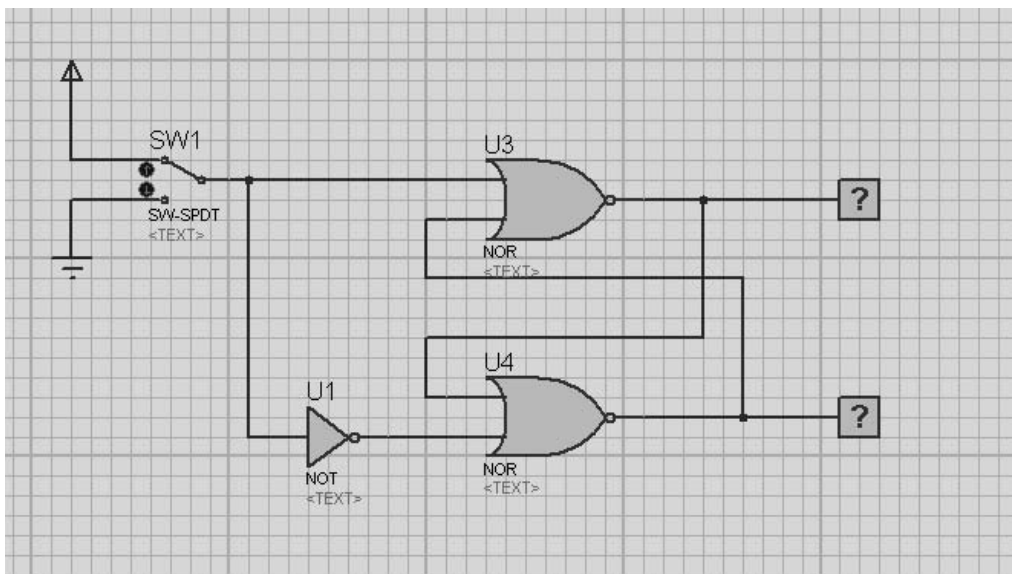
adalah 2 buah yaitu R (reset) dan S (set), kedua input tersebut dimodifikasi sehingga pada Data flip-flop menjadi 1 buah input saja yaitu input atau masukan D (data) saja. Model modifikasi RS flip-flop menjadi D flip-flop adalah dengan penambahan gerbang NOT (Inverter) dari input S ke input R pada RS flip-flop seperti terlihat pada gambar 3.

Tabel 5.2. Flip-flop D

Input	Output	
D	Q	\bar{Q}
0	0	1
1	1	0

5.3.1. Rangkaian Flip-flop D

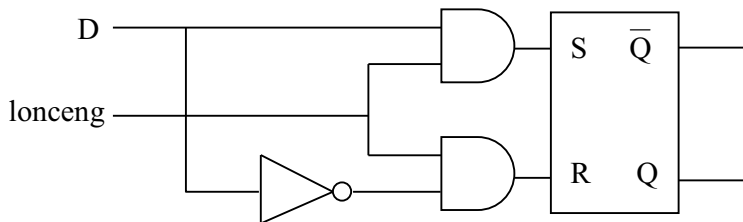
Terdapat berbagai cara untuk merancang flip-flop D. Flip-flop D dapat dibangun dengan mengembangkan flip-flop R-S seperti Gambar 3. Sedangkan Gambar 4 memperlihatkan suatu cara untuk membangun sebuah flip-flop D dengan lonceng (*clock*). Jenis flip-flop ini mencegah nilai D mencapai keluaran Q sampai berlangsungnya sinyal lonceng aktif.



Gambar 5.6. Rangkaian flip-flop D dikembangkan dari FF R-S

Cara kerja rangkaian yang bersangkutan adalah sebagai berikut:

1. Bila lonceng bernilai 0 (rendah), kedua gerbang AND tertutup, oleh karenanya perubahan nilai D tidak mempengaruhi nilai output Q.
2. Sebaliknya, bila lonceng bernilai 1 (tinggi), kedua gerbang AND terbuka. Dalam hal ini, Q terdorong untuk menyamai nilai D. Bila lonceng turun kembali, Q tak berubah dan menyimpan nilai D yang terakhir.

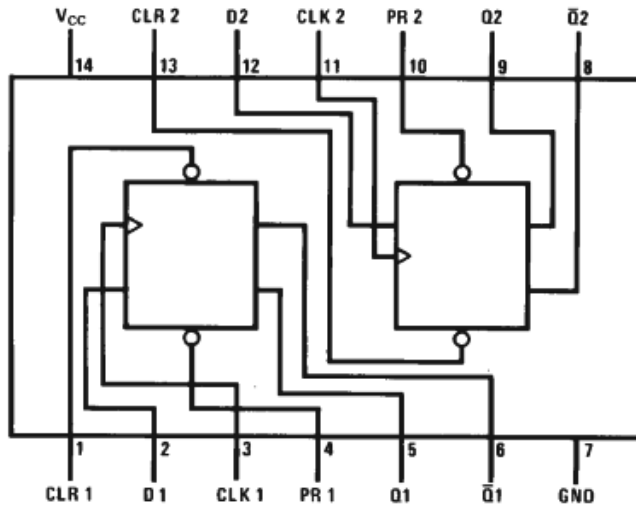


Gambar 5.7. Rangkaian flip-flop D dengan lonceng

5.3.2. IC 7474 (Dual Positive-Edge-Triggered D Flip-Flops)

Untuk penggunaan praktis, kita dapat menggunakan IC 7474 yang berisi 2 buah *Positive-Edge-Triggered D Flip-Flop*. *Positive-Edge-Triggered* artinya nilai pada masukan kaki D akan diterima oleh Flip-Flop saat terjadi perubahan sinyal lonceng (*clock*) dari 0 ke 1 atau sering juga disebut *rising edge*. Perubahan masukan pada kaki D tidak akan berpengaruh pada keluaran Q bila tidak terjadi transisi pada lonceng dari 0 ke 1, walaupun misalnya lonceng bernilai 1. Diagram hubungan kaki-kaki IC ini dapat dilihat pada gambar 5.

Pada IC 7474, juga terdapat kaki **Preset** dan kaki **Clear**. Kaki **Preset** berfungsi untuk memaksa *output* menjadi $Q=1$ dan $\bar{Q}=0$, tanpa mempedulikan input D, sedangkan kaki **Clear** berfungsi untuk memaksa *output* menjadi $Q=0$ dan $\bar{Q}=1$, tanpa mempedulikan input D. Yang perlu diperhatikan adalah pada IC ini, kaki **Preset** dan **Clear** bersifat **active-low** (lawan **active-high**), artinya kaki tersebut aktif justru ketika mendapat masukan 0 (**low**). Sifat tersebut dapat diketahui dengan adanya tanda \circ pada kaki yang bersangkutan di diagram seperti gambar 5.



Gambar 5.8. Diagram hubungan IC 7474

5.4. Jenis-jenis rangkaian flip-flop berdasar Clock-nya.

Berdasarkan perilakunya atau hubungan *input* dan *output*, flip-flop dibedakan menjadi beberapa jenis, antara lain **Flip-flop RS**, **Flip-flop D**, **Flip-flop JK**, dan **Flip-flop T**. Berdasarkan ada atau tidaknya lonceng (*clock*), rangkaian Flip-flop dapat dibagi menjadi 3, yaitu sebagai berikut.

1. **Latch**, yaitu rangkaian flip-flop yang tidak mempunyai *input* sinyal *clock*. Pada untai ini *output* akan bereaksi seiring dengan perubahan *input*. Contohnya adalah rangkaian pada Gambar 5.3. Rangkaian flip-flop RS dengan gerbang NOR Gambar 5.3 dan Gambar 5.4rangkaian 2 pada Labsheet 3 .
2. **Level-sensitive flip-flop**, yaitu rangkaian flip-flop yang mempunyai input sinyal *clock*, dan output akan bereaksi terhadap perubahan input saat sinyal *clock* aktif (bisa saat bernilai 1 atau 0). Contoh dari untai ini adalah rangkaian 3 Gambar 5.5 dan Gambar 5.6dari Labsheet 3 dan rangkaian 1 dari Labsheet 4.
3. **Edge-triggered flip-flop**, yaitu rangkaian flip-flop yang mempunyai input sinyal *clock*, dan *output* akan bereaksi terhadap perubahan input saat sinyal *clock* berubah dari 0 ke 1 (untuk tipe *Positive-Edge-Triggered FF*) atau

saat *clock* berubah dari 1 ke 0 (untuk tipe *Negative-Edge-Triggered FF*).

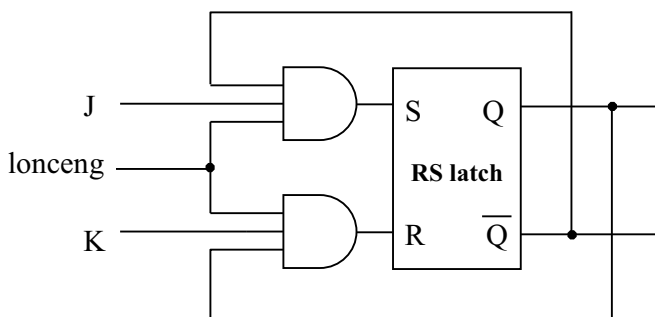
Beberapa literatur menyebut tipe inilah yang benar-benar disebut **flip-flop**.

Contoh dari rangkaian ini adalah rangkaian flip-flop D pada IC 7474.

Jenis ke-2 dan ke-3 juga biasa disebut sebagai *clocked flip-flop*.

5.5. Flip-Flop JK

Nama flip-flop JK berasal dari nama masukannya, yaitu J dan K. Masukan J dan K disebut masukan pengendali karena menentukan apa yang dilakukan oleh flip-flop pada saat suatu pinggiran pulsa positif tiba. Flip-flop ini juga digunakan untuk mengatasi kelemahan dari flip-flop RS, yang tidak mengijinkan pemberian masukan $R=S=1$. Rangkaian flip-flop JK ditunjukkan pada gambar berikut.



Gambar 5.9. Rangkaian flip-flop JK

Cara kerja rangkaian di atas dapat dijelaskan sebagai berikut.

1. Pada saat J dan K keduanya 0, R dan S pasti bernilai 0-0, sehingga Q tetap pada nilai terakhirnya .
2. Pada saat J rendah dan K tinggi, gerbang atas tertutup (S bernilai 0), maka tidak terdapat kemungkinan untuk mengeset flip-flop. Bila Q tinggi ($Q = 1$) dan lonceng = 1, gerbang bawah (lonceng AND K AND Q) akan melewati pemicu reset ($R = 1$) yang akan menyebabkan Q menjadi rendah. Jadi $J = 0$ dan $K = 1$ berarti lonceng = 1 akan mereset flip-flopnya ($Q = 0$), bila Q sebelumnya tinggi.
3. Pada saat J tinggi dan K rendah, maka tidak terdapat kemungkinan untuk mereset flip-flop ((karena R pasti bernilai 0). Bila Q rendah ($Q = 0$ dan $\bar{Q} = 1$) dan lonceng = 1, gerbang atas (lonceng AND J AND \bar{Q}) akan

melewatkan pemicu set ($S = 1$) yang akan menyebabkan Q menjadi tinggi. Jadi $J = 1$ dan $K = 0$ berarti lonceng = 1 akan mengeset flip-flopnya ($Q = 1$), bila Q sebelumnya rendah.

4. Pada saat J dan K keduanya tinggi, dapat mengeset atau mereset flip-flopnya, tergantung kondisi Q sebelumnya. Bila Q tinggi ($Q = 1$) dan lonceng = 1, gerbang bawah akan melewati pemicu reset ($R = 1$) yang akan menyebabkan Q menjadi rendah.

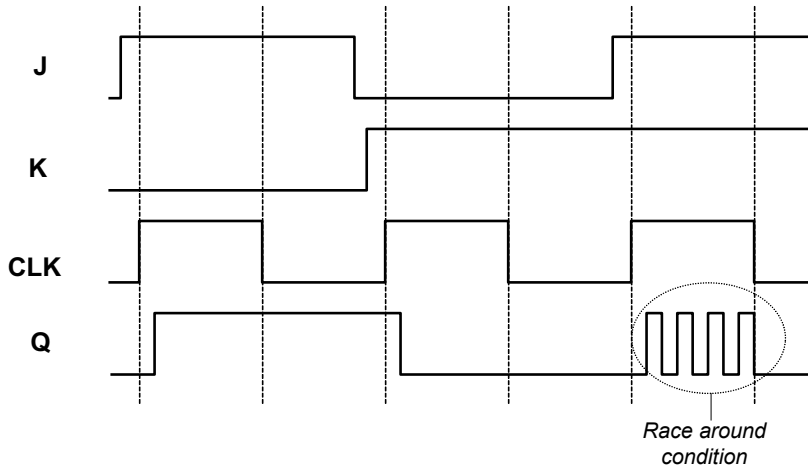
Bila Q rendah ($Q = 0$) dan lonceng = 1, maka $\bar{Q} = 1$, gerbang atas akan melewati pemicu set ($S = 1$) yang akan menyebabkan Q menjadi tinggi. Jadi $J = 1$ dan $K = 1$ berarti bahwa pinggiran pulsa lonceng positif berikutnya akan membuat nilai Q yang baru adalah kebalikan dari nilai Q sebelumnya ($Q_{t+1} = \bar{Q}_t$).

Tabel 5.3. Tabel Kebenaran JK dengan clock

CLK	J	K	Q	\bar{Q}
0	0	0	Keadaan terakhir	
↑	0	0	0	1
↑	0	1	1	0
↑	1	1	$\overline{\text{Keadaan terakhir}}$	

5.5.1. Kondisi Osilasi (*Race-around Condition*)

Rangkaian FF J-K di atas mempunyai satu kelemahan, yaitu memungkinkan terjadinya kondisi osilasi atau *race-around condition*. Hal ini terjadi jika lebar pulsa lonceng (*clock*) lebih besar dari waktu pensaklaran FF (waktu yang dibutuhkan keluaran untuk bereaksi terhadap keluaran). Dalam keadaan ini keluaran yang diumpanbalikkan ke masukan akan mengubah masukan, sehingga menyebabkan perubahan pada keluaran dan seterusnya, hingga akhir pulsa lonceng, kemudian menyebabkan keluaran FF tidak jelas. Hal ini sering terjadi ketika rangkaian di atas mendapat masukan $J = 1$ dan $K = 1$. Hal tersebut dapat dilihat pada diagram pewaktuan (*timing diagram*) berikut.

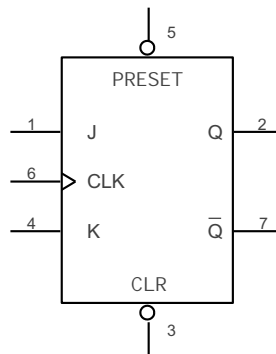


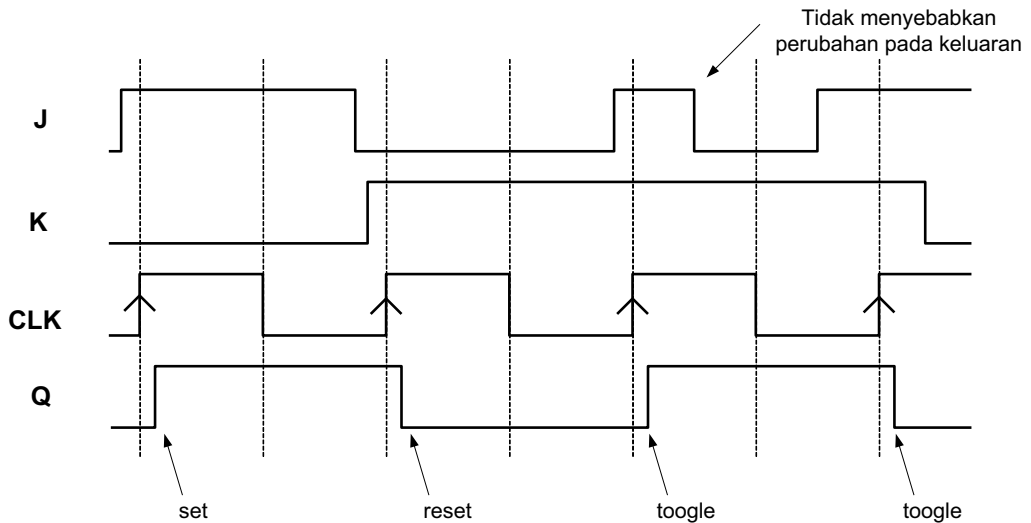
Gambar 5.10. Kondisi Osilasi

Untuk mengatasi kondisi osilasi, dikembangkan flip-flop dengan pemuncian sisi (*edge triggering*) dan flip-flop utama/pembantu (*master/slave flip-flop*).

5.5.2. Edge-triggered Flip-Flop

Seperti telah disebutkan di atas, salah satu cara untuk mengatasi *race-around condition* adalah dengan mengembangkan *edge-triggered flip-flop*. Berikut adalah simbol dari *Positive-edge-triggered J-K FF* yang berada dalam IC 74109. Tabel 1 di atas sebenarnya menunjukkan perilaku F-F J-K jenis ini.



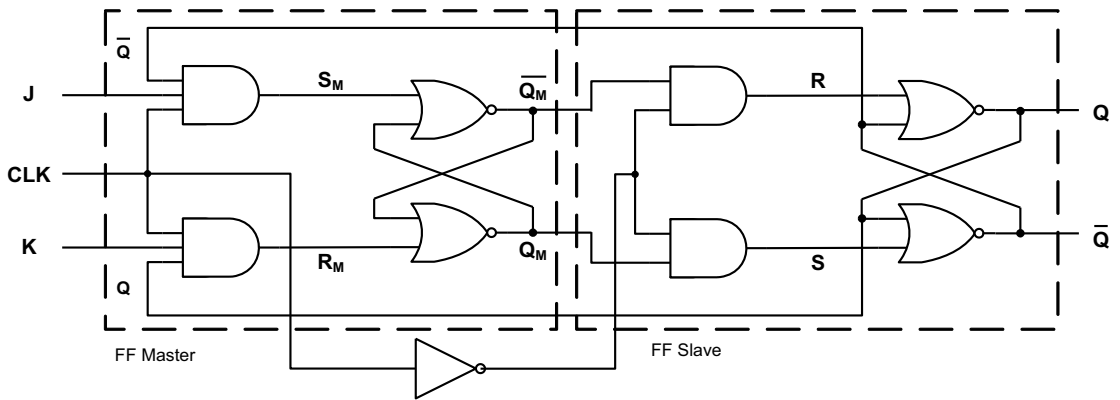


**Gambar 5.11. Simbol dan bentuk pulsa masukan-keluaran
Positive-edge-triggered JK-FF.**

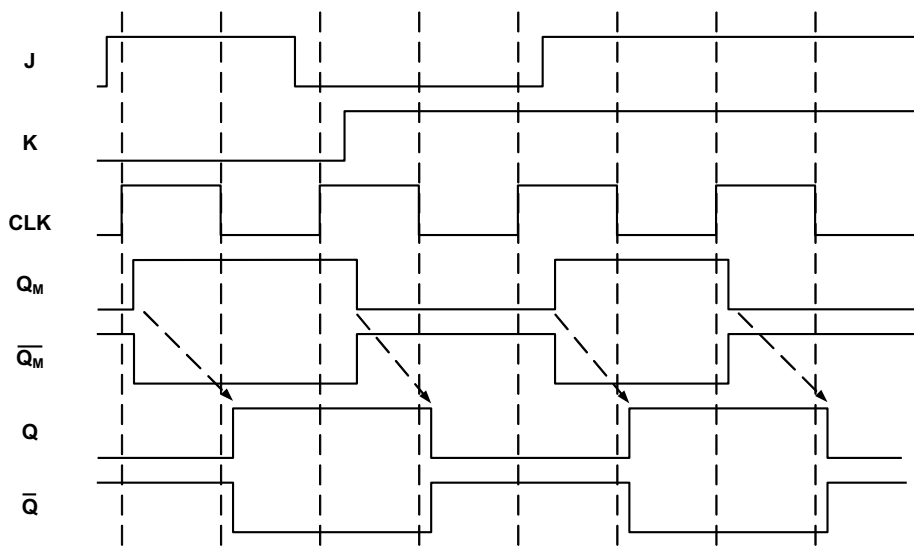
5.5.3. Flip-flop JK Utama/Pembantu (JK Master/Slave Flip-Flops)

Gambar 5.12. memperlihatkan cara membangun sebuah flip-flop JK utama/pembantu (*JK master/slave flip-flop*) untuk menghindari **kondisi osilasi**. Cara kerjanya adalah sebagai berikut.

- Pada saat pulsa *clock* diterapkan ($CLK = 1$), F-F *master* akan *enable* dan keluaran akan berubah sesuai masukan J dan K. Tetapi saat itu pulsa *clock* yang masuk ke F-F *slave* adalah $\overline{CLK} = 0$, F-F *slave* akan *disable*, tidak ada perubahan pada keluaran.
- Pada akhir pulsa *clock*, $CLK = 0$ dan $\overline{CLK} = 1$, F-F *master* akan *disable* dan F-F *slave* akan *enable*. F-F *slave* akan merubah keluarannya sesuai dengan keluaran Q_M dan $\overline{Q_M}$. Jika $Q_M = 1$ dan $\overline{Q_M} = 0$, maka $Q = 1$ dan $\overline{Q} = 0$, atau sebaliknya.
- Jadi selama selang waktu detak, keluaran Q tidak akan berubah tetapi Q_M mengikuti logika J-K, pada akhir pulsa detak, nilai Q_M akan ditransfer ke Q.



Gambar 5.12. Rangkaian Master-Slave JK FF untuk menghindari osilasi



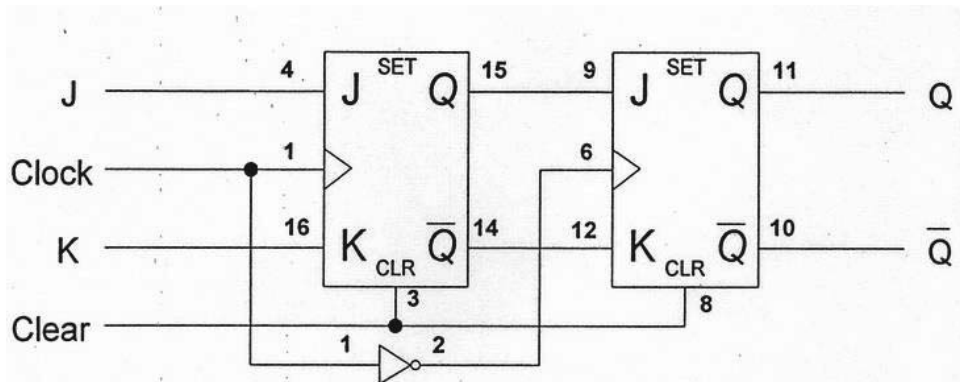
Gambar 5.13. Keluaran-masukan *Master/Slave JK* untuk menghindari osilasi.

5.5.4. Master/Slave JK Flip-Flops dengan *edge-triggered flip-flop*

Gambar 6 memperlihatkan cara lain membangun sebuah flip-flop JK utama/pembantu (*JK master/slave flip-flop*). Cara kerjanya adalah sebagai berikut.

- Jika $J=1$ dan $K=0$, flip-flop utama diset pada saat pinggir-an-positif pulsa lonceng (*clock*) tiba. Keluaran Q_M yang tinggi dari flip-flop utama mendrive masukan J pada flip-flop pembantu, maka pada saat pinggir-an-negatif pulsa lonceng tiba, flip-flop pembantu diset, menyamai kerja flip-flop utama.

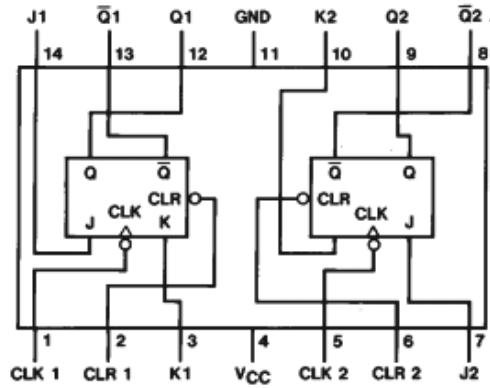
- Jika $J=0$ dan $K=1$, flip-flop utama direset pada saat pinggiran-positif pulsa lonceng tiba. Keluaran \overline{Q}_M yang tinggi dari flip-flop utama menuju ke masukan K pada flip-flop pembantu. Oleh karenanya, kedatangan pinggiran-negatif pulsa lonceng mendorong flip-flop pembantu untuk reset. Sekali lagi, flip-flop pembantu menyamai kerja flip-flop utama.
- Jika masukan J dan K pada flip-flop utama adalah tinggi, maka flip-flop ini *toggle* pada saat pinggiran-positif pulsa lonceng tiba sedang flip-flop pembantu *toggle* pada saat pinggiran-negatif pulsa lonceng tiba. Dengan demikian, apapun yang dilakukan oleh flip-flop utama, akan dilakukan pula oleh flip-flop pembantu: jika flip-flop utama diset, flip-flop pembantu diset; jika flip-flop utama direset, flip-flop pembantu direset pula, namun waktunya selisih satu pulsa detak.



Gambar 5.14. Master-Slave JK FF dengan *Positive-edge-triggered* JK-FF.

5.5.5. Flip-flop J-K dalam IC 7473

Flip-flop JK yang terdapat di dalam IC 7473 disebut *Negative-Edge-Triggered Master/Slave J-K Flip-Flop* (*datasheet* lain menyebutnya *Positive Pulse-Triggered*, yang pada dasarnya hampir sama maknanya). Dalam satu IC 7473 terdapat dua Flip-Flop J-K dengan konfigurasi kaki-kaki sebagai berikut.



Gambar 5.15. Diagram koneksi IC 7473.

5.6. Simulasi Rangkaian Flip-Flop dengan Proteus

Langkah-langkah untuk melakukan simulasi dengan Proteus dapat dilihat pada Bab sebelumnya

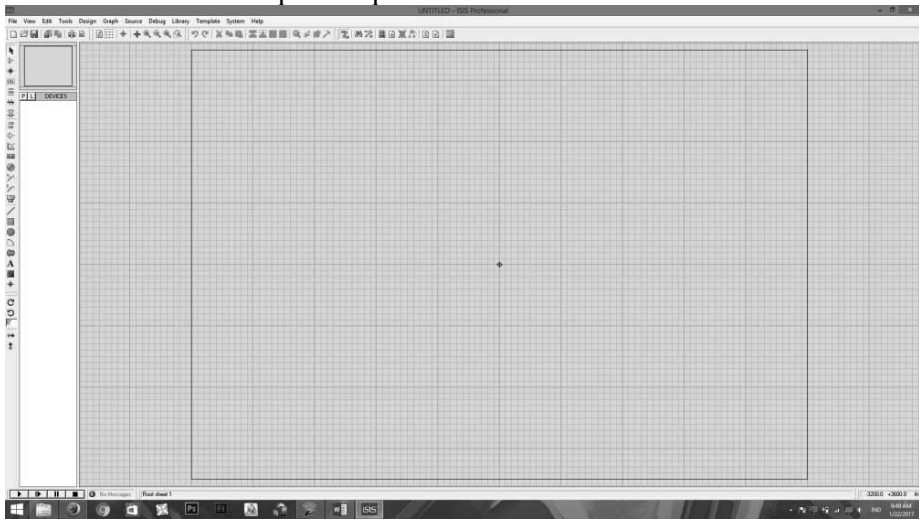
5.6.1. Simulasi Flip-Flop RS

1. Buka program Proteus dengan mengklik dua kali *icon program ISIS 7 Professional*.



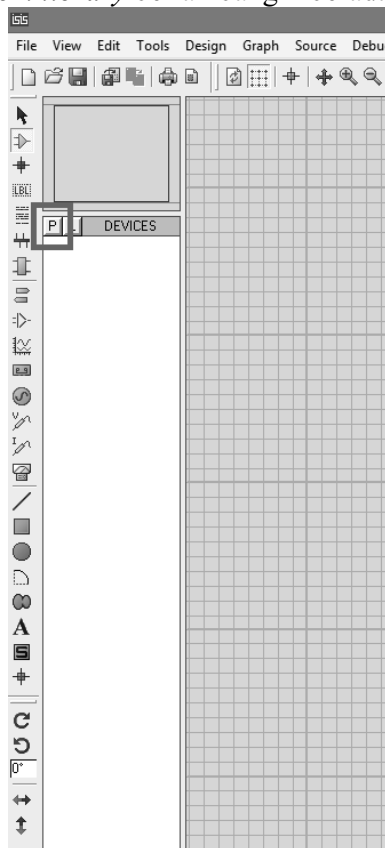
Gaambar 5.16. *Icon Program ISIS 7 Professional*

Lalu akan muncul tampilan seperti Gambar 5.17



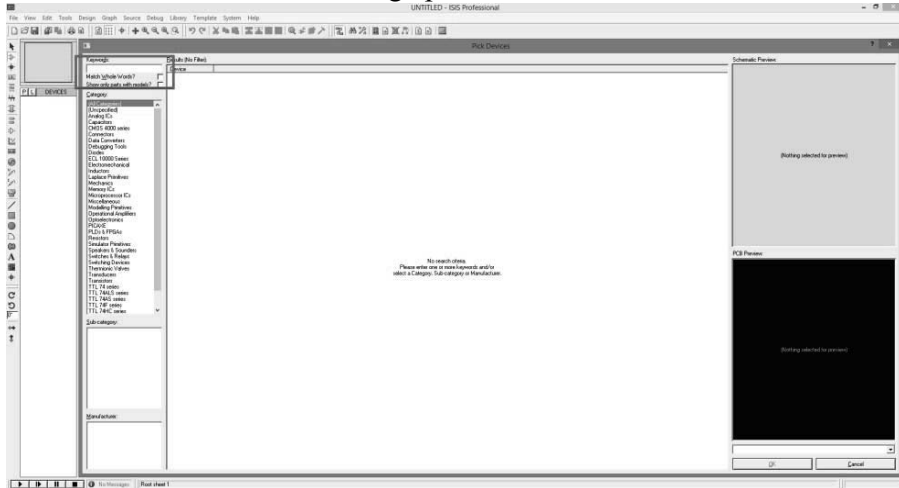
Gambar 5.17. Tampilan Utama Program ISIS 7 Proteus

2. Tekan tombol *pick from library* berlabel P berada kiri atas layar



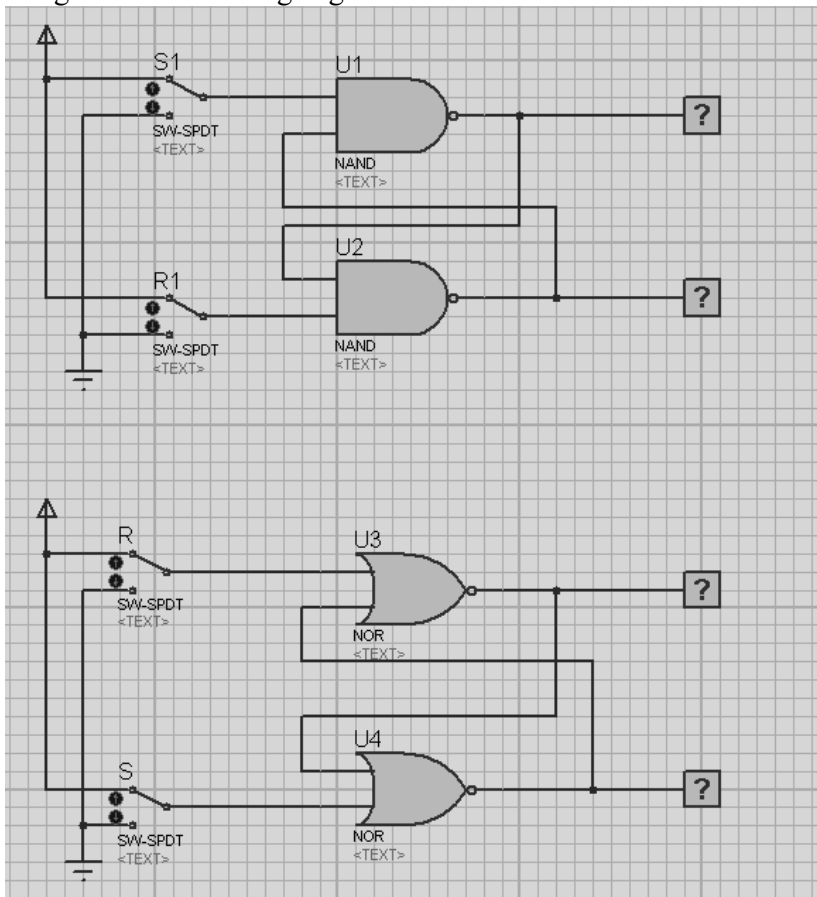
Gambar 5.18. Tekan Tombol *Pick from Library*

- Carilah gerbang NOR atau gerbang NAND pada kotak pencari, setelah menemukan gerbang yang dimaksud lalu tekan ok, ulangi langkah ini untuk mencari switch dan logicprobe.



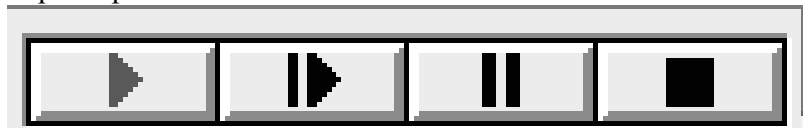
Gambar 5.19. Tempat penunjuk kotak pencari

- Rangkailah sesuai dengan gambar berikut

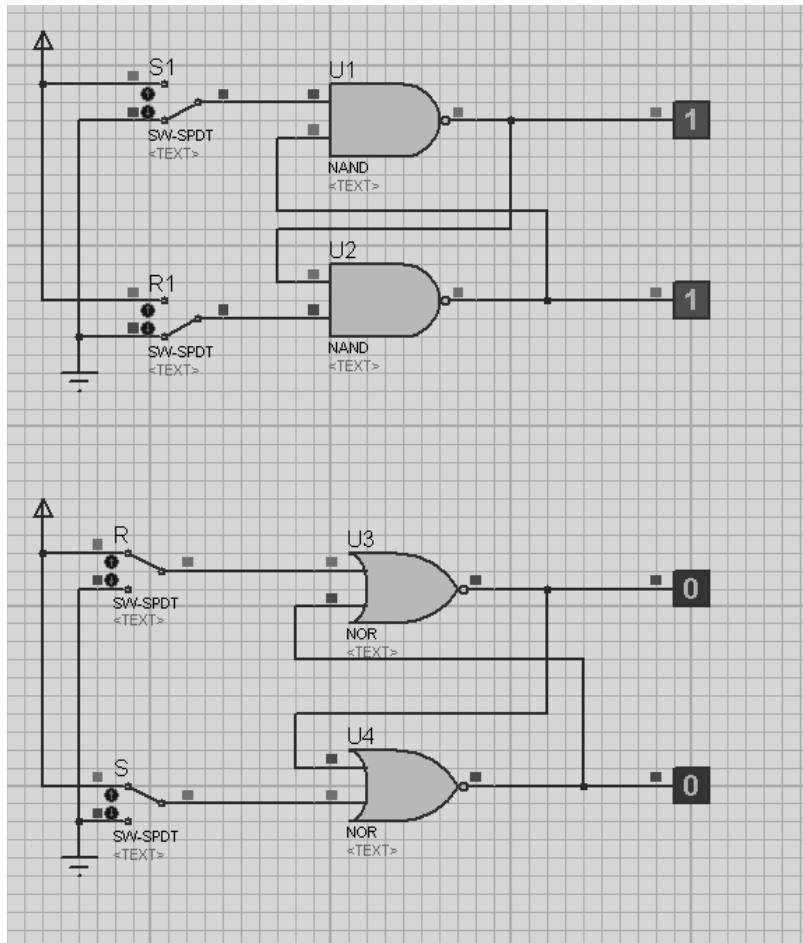


Gambar 5.20. Rangkaian Flip – Flop RS

- Setelah selesai merangkai lalu tekan tombol *play* yang ada pada pojok kiri bawah panel proteus



Gambar 5.21. Tombol *Play* Proteus



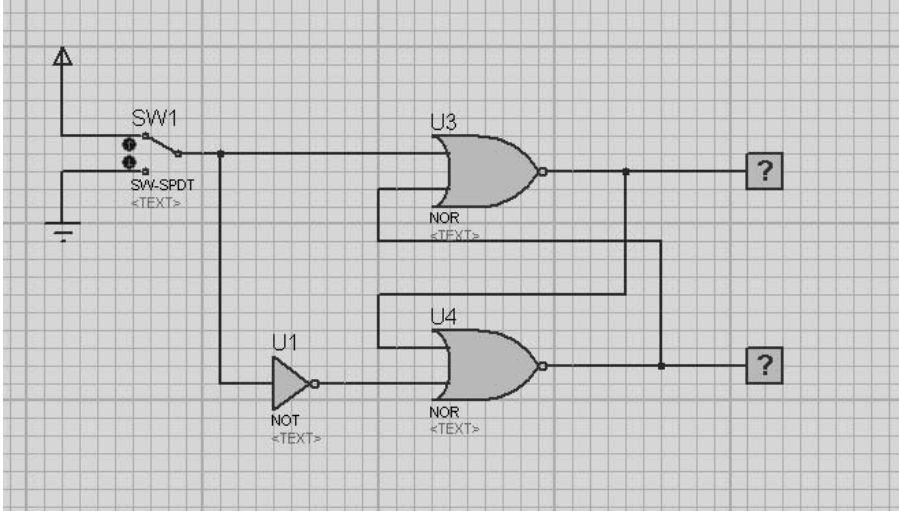
Gambar 5.22. Simulasi Rangkaian Flip – Flop RS

Tabel Percobaan Flip-Flop RS dengan gerbang NAND

R	S	\bar{R}	\bar{S}	Q	\bar{Q}
0	0	1	1	NT	NT
0	1	1	0	0	1
1	0	0	1	1	0
1	1	0	0	-	-

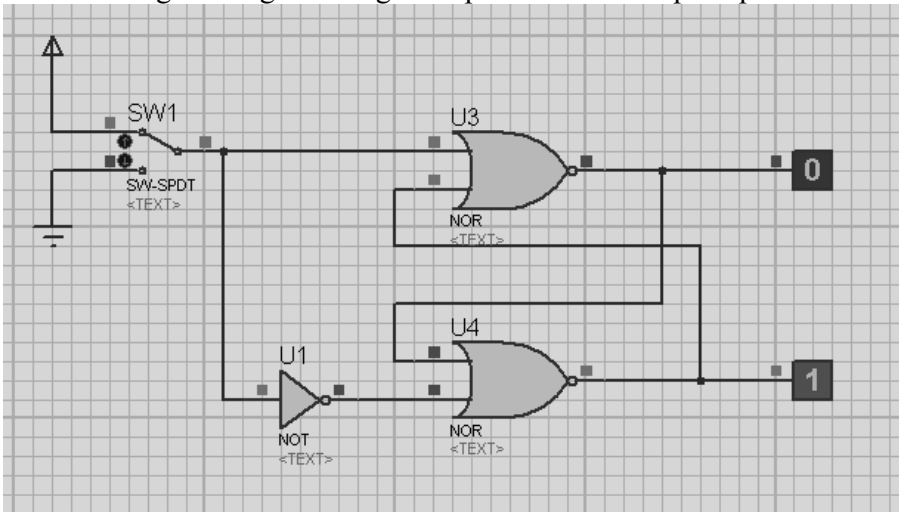
5.6.2. Simulasi Flip-Flop D

1. Lakukan langkah 1-3 pada simulasi Flip-Flop RS dengan menambahkan gerbang NOT pada langkah ketiga dan rangkai rangkaian seperti pada Gambar 5.23 berikut.



Gambar 5.23. Rangkaian Flip-Flop D

2. Setelah merangkai rangkaian sesuai pada Gambar 5.23 lakukanlah simulasi dengan mengikuti langkah 5 pada simulasi Flip-Flop RS



Gambar 5.24 Simulasi Rangkaian Flip-Flop D

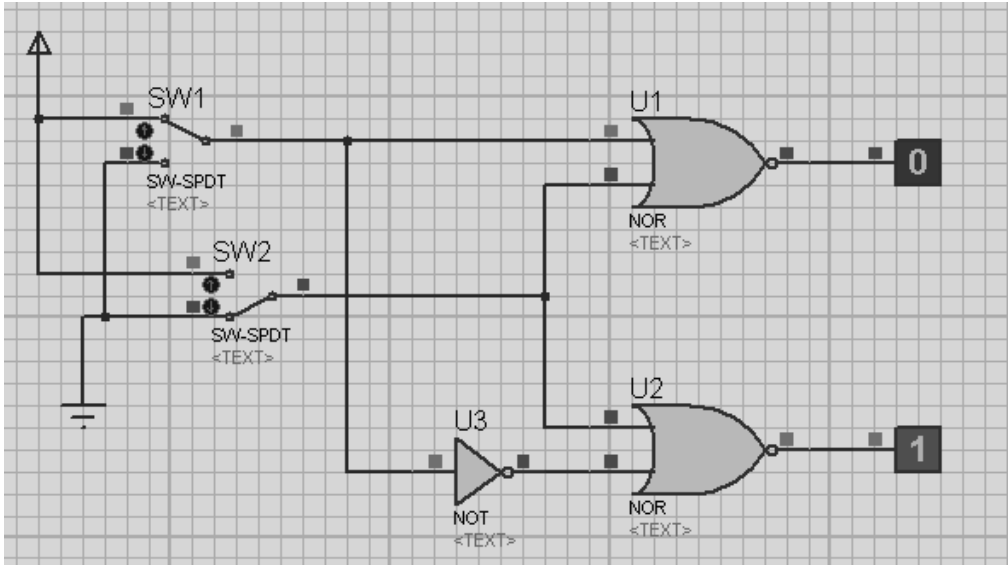
Tabel 1.percobaan Flip-flop dengan gerbang NOR

INPUT		OUTPUT	
Clock	D	Q	Q'
0	0		
1	0		
0	1		
1	1		
0	0		
1	0		
0	1		
1	1		

Tabel 2. Percobaan Flip-Flop D dengan fasilitas preset dan clear menggunakan IC

7474

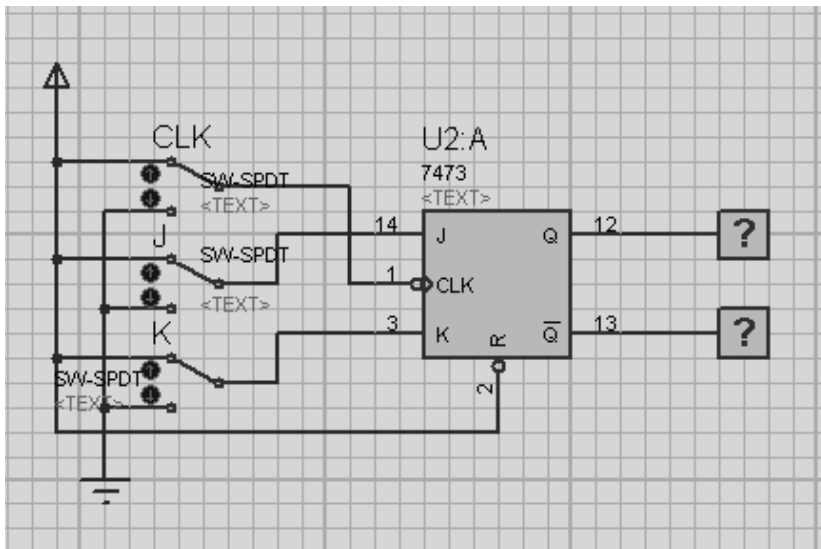
Preset	Clear	Input		Output	
		clock	D	Q	Q'
0	0				
0	0				
1	0				
1	0				
0	1				
0	1				
1	1				
1	1				
0	1				
0	1				
1	1				
1	1				



Gambar 5.25 Simulasi Rangkaian Flip-Flop D dengan Lonceng

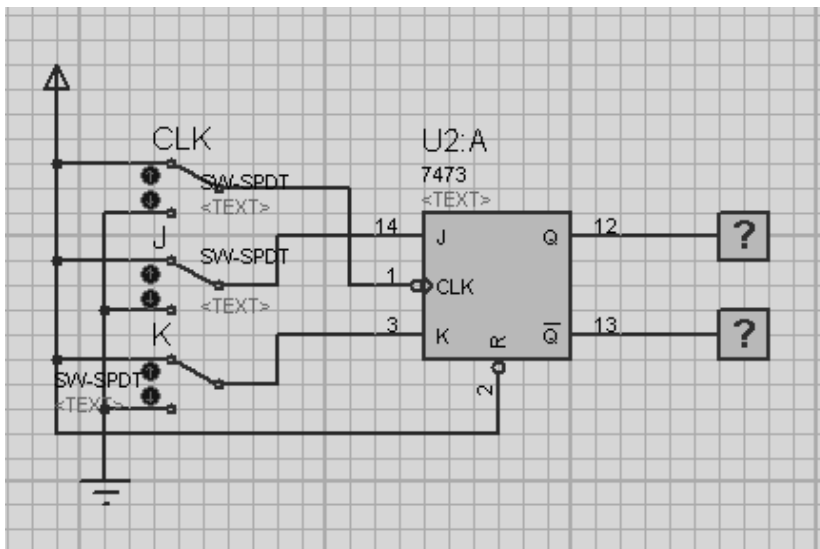
5.6.3. Simulasi Flip-Flop JK

1. Lakukan langkah 1-3 pada simulasi Flip-Flop RS dengan menambahkan IC 7473 pada langkah ketiga dan rangkai rangkaian seperti pada Gambar 5.26 berikut

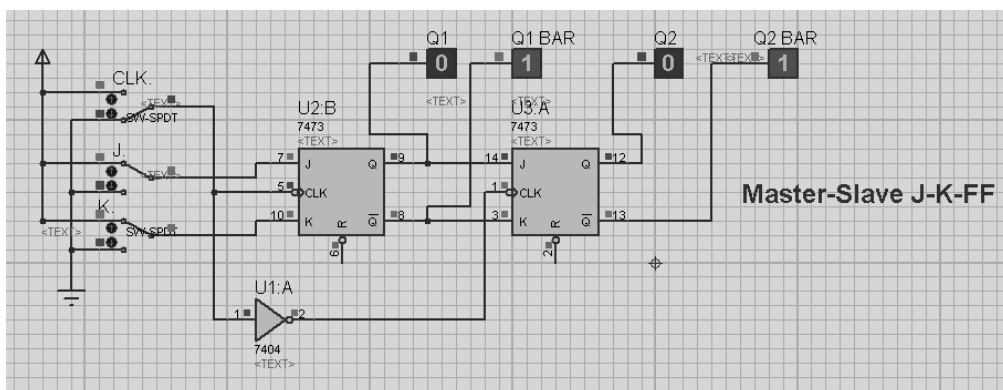


Gambar 5.26. Rangkaian Flip-Flop JK

2. Setelah merangkai rangkaian sesuai pada Gambar 5.26 lakukanlah simulasi dengan mengikuti langkah 5 pada simulasi Flip-Flop RS.



Gambar 5.27. Simulasi Rangkaian Flip-Flop JK



Gambar 5.28. Simulasi Rangkaian Master-Slave JK

Tabel 1. Percobaan JK F-F Flip- Filp dengan gerbang NOR

INPUT			OUTPUT	
Clock	J	K	Q	Q'
0	0	0		
1	0	0		
0	0	1		
1	0	1		
0	1	0		
1	1	0		
0	1	1		
1	1	1		

Tabel 2. Percobaan J-K F-F denagn IC 7473

INPUT		CLOCK	OUTPUT	
J	K		Q	Q'
0	0	↓		
0	1	↓		
1	0	↓		
1	1	↓		
1	1	↓		
0	1	1		
1	0	1		

Tabel 3. Percobaan Master-Slave j-K F-F

Clock	J	K	Q ₁	Q ₁ '	Q ₂	Q ₂ '
0	0	0				
1	0	0				
0	0	1				
1	0	1				
0	1	0				
1	1	0				
0	1	1				
1	1	1				
0	1	1				

5.7. Latihan Soal

- Berikut ini yang merupakan karakteristik dari flip-flop adalah ...
 - Memiliki 2 bentuk keluaran yang sama
 - Tidak memiliki masukan bersifat feedback
 - Memiliki 2 bentuk keluaran yang sama**
 - Memiliki 2 bentuk masukan yang berlawanan
 - Memiliki 2 bentuk masukan yang sama
- Nama lain dari rangkaian flip-flop adalah ...
 - Multivibrator astabil
 - Multivibrator bistabil**
 - Schmitt bistabil
 - Comparator
 - Decoder
- Perbedaan RS flip-flop dapat dibuat dari kombinasi gerbang ...
 - AND dan OR
 - NOT dan AND
 - NAND dan NOR**
 - NAND dan OR
 - AND dan NAND
- Apabila kita menginginkan suatu kondisi dimana RS flip-flop menghasilkan output Y seperti keadaan terakhirnya, maka kita membutuhkan input ...
 - R dan S tinggi
 - R tinggi, S rendah
 - R rendah, S tinggi
 - R dan S rendah**
 - Semua jawaban benar
- Dari pernyataan berikut yang benar tentang operasi negative-edge-triggered flip-flop D adalah ...
 - Logika pada input D ditransfer ke Q di NGT dari CLK**
 - Output Q selalu sama dengan input CLK jika input D high
 - Output Q selalu sama dengan input CLK jika input D low
 - Output Q selalu sama dengan input D ketika CLK = PGT
 - Output Q selalu sama dengan input D
- Dari pernyataan berikut yang benar tentang gated D flip-flop adalah ...
 - Toggle dari output akan aktif jika salah satu dari input high
 - Hanya satu dari input yang dapat berlogika high
 - Output complement sama dengan input jika dalam kondisi aktif
 - Output Q sama dengan input D ketika diaktifkan dalam kondisi high**
 - Output Q sama dengan input D ketika diaktifkan dalam kondisi low
- Pernyataan berikut yang benar terkait output Q ketika D latch adalah ...
 - Output Q sama dengan input D ketika EN dalam kondisi low
 - Output Q sama dengan input D ketika EN dalam kondisi high**
 - Output Q berlawanan dengan input D ketika EN dalam kondisi high

- d. Output Q berlawanan dengan input D ketika EN dalam kondisi low
 - e. Output Q dalam kondisi high tidak terpengaruh kondisi input EN
8. Sebuah output yang benar dicapai master-slave JK flip-flop hanya jika input yang stabil ketika ...
- f. Clock dalam kondisi low
 - g. Slave dalam kondisi transferring
 - h. Flip-flop reset
 - i. Flip-flop set
 - j. Clock dalam kondisi high**
9. Yang dimaksud lambang segitiga pada input clock JK flip-flop adalah ...
- f. Level enabled
 - g. **Edge-triggered**
 - h. Positive-triggered
 - i. Negative-triggered
 - j.
10. Sebuah JK flip-flop dengan $J = 1$ dan $K = 1$ mempunyai 20 kHz input clock. Output Q bernilai ...
- a. Low konstan
 - b. High konstan
 - c. 20 kHz square-wave
 - d. 10 kHz square-wave**
 - e. 0 kHz square-wave

BAB 6.

RANGKAIAN REGISTER DAN COUNTER

6.1. Register

Register merupakan blok logika yang sangat penting dalam kebanyakan dalam sistem digital. Register sering digunakan untuk menyimpan (sementara) informasi biner yang muncul pada keluaran sebuah matrik pengkodean. Disamping itu register sering digunakan menyimpan (sementara) data biner yang sedang dikode. Maka register membentuk suatu kaitan yang sangat penting antara sistem digital utama dan kanal-kanal masukan atau keluaran.

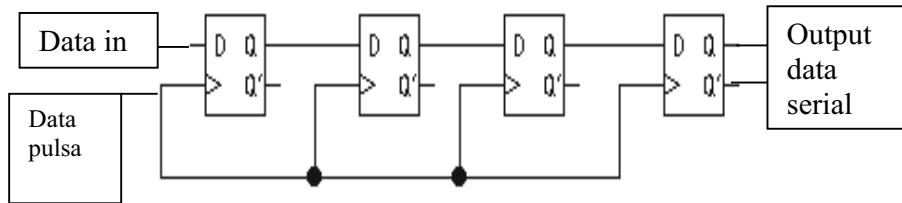
Register biner juga membentuk basis bagi beberapa operasi aritmatika yang sangat penting. Sebagai contoh, operasi-operasi komplementasi, perkalian, dan pembagian seringkali diwujudkan dengan menggunakan register.

Register tidak lebih dari pada sekelompok flipflop yang dapat digunakan untuk menyimpan sebuah bilangan biner. Harus terdapat flip flop bagi masing-masing bit dalam bilangan biner tersebut. Tentunya flip flop harus dihubungkan sedemikian hingga bilangan bilangan biner dapat dimasukkan kedalam register dan dikeluarkan dari register.

Register adalah kumpulan elemen-elemen memori yang bekerja bersama sebagai satu kesatuan unit. Register dapat dibentuk dari logika sekuensial yang terbentuk dari flip-flop. Berdasarkan fungsi register terdiri atas, register *buffer* dan register geser.

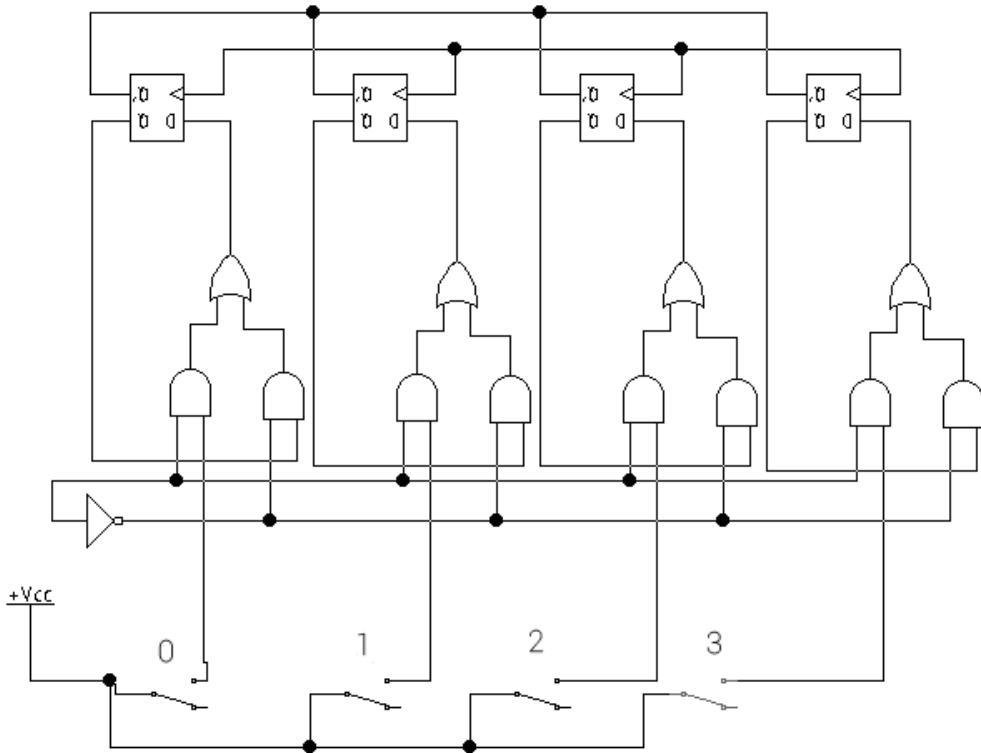
6.2. Register Buffer

Register *buffer* berfungsi untuk menyimpan kata digital. Setiap data pulsa clock, data dari input D dari masing-masing FF akan di transfer kepada Q output. Pada awalnya, isi dari register diset 0 dengan mengirimkan clock pada clear. Jika 1 merupakan input dari FF yang pertama, maka pada pulsa berikutnya 1 akan di transfer ke output FF1 dan sekaligus menjadi input FF2.



Gambar 1 Rangkaian buffer D-FF yang tersambung dalam sebuah rangkaian serial in, serial out shift register.

Register Buffer terkendali, register buffer yang ditambahkan dengan beberapa gerbang logika dasar AND, OR, dan NOT. Gambar rangkaian menunjukkan sebuah Register buffer terkendali dengan CLR aktif tinggi. Apabila $CLR = 1$, maka akan terjadi reset pada flip-flop dan data yang tersimpan (Q) menjadi 0000. Dan ketika $CLR = 0$, register siap beroperasi kembali. Sinyal kendali $LOAD = 0$, semua input data tidak diizinkan masuk, artinya flip-flop mengisolasi input data atau menahan semua data yang ada di dalamnya. Dengan kata lain, register tidak berubah selama $LOAD = 0$. Ketika $LOAD = 1$, semua input data akan diterima oleh register. Ketika $LOAD$ kembali = 0, maka input data yang diterima register tadi akan tersimpan dengan aman tanpa gangguan perubahan input.

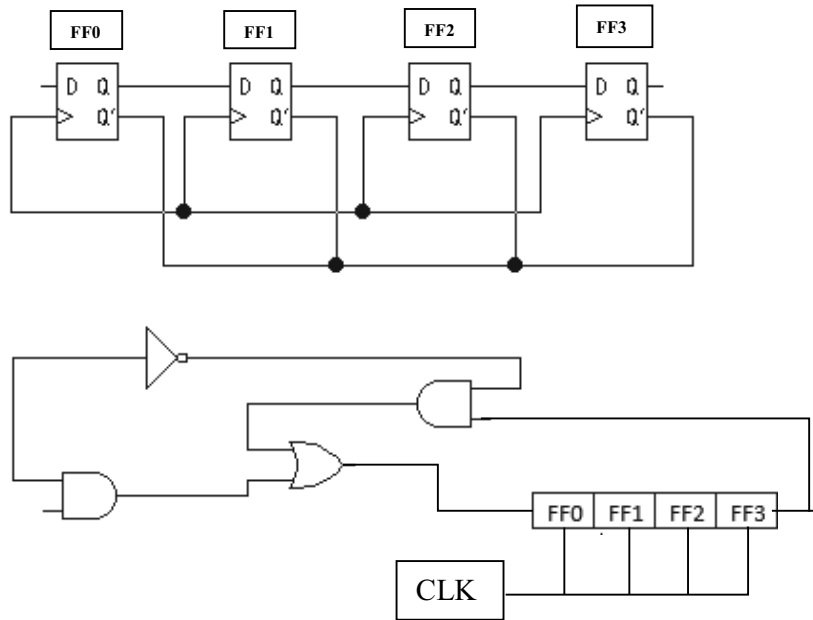


Gambar 2. Rangkaian register buffer terkendali

6.3. Register Geser

Register geser *serial in – serial out* (SISO)

Dasar register geser empat-bit dapat dirangkai dengan menggunakan empat D *flip-flop*, seperti yang diperlihatkan di bawah. Selama pulsa *clock*, satu bit ditransmisikan dari kiri ke kanan. Menerima suatu kata data menjadi 1001.



Gambar 3. Rangkaian register geser SISO

Data diangkut pada register, saat garis kontrol tinggi (HIGH dengan kata lain WRITE). Data dapat di geser keluar dari register saat garis kontrol rendah (LOW dengan kata lain READ). Aliran data pada register SISO ini diibaratkan pada gambar 4.



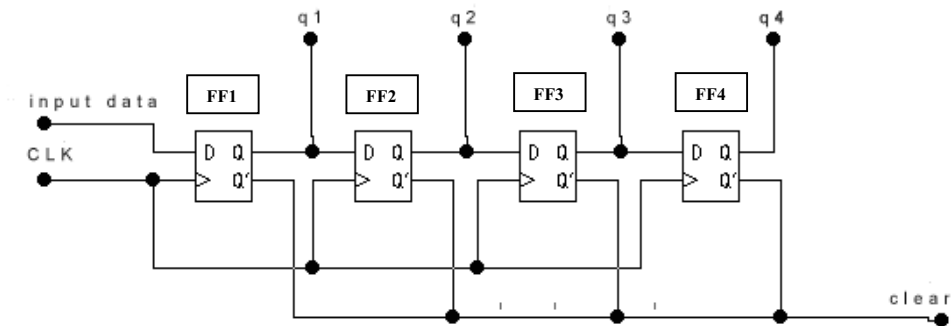
Gambar 4. Ilustrasi data masuk dan keluar register SISO

Tabel kebenaran register geser SISO (misal inputan 1101)

clock	in	Q1	Q2	Q3	Q4
0	0	0	0	0	0
1	1	1	0	0	0
2	0	0	1	0	0
3	1	1	0	1	0
4	1	1	1	0	1

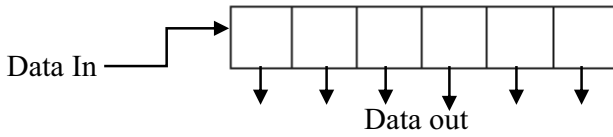
Register geser serial in - parallel out (SIPO)

Dari jenis register ini, bit-bit data dimasukan secara serial sama artinya sama dengan SISO. Perbedaanya adalah cara dimana bit-bit data dipindahkan dari register. Sekali data disimpan, setiap bit muncul pada masing-masing baris keluarannya, dan semua bit-bitnya mampu secara simultan. Sebuah susunan empat-bit register SIPO diperlihatkan di bawah ini.



Gambar 5. Register SIPO dalam susunan 4 bit

Pada jenis SIPO ini, hasil keluaran dari register geser dapat diilustrasikan seperti pada gambar 6.

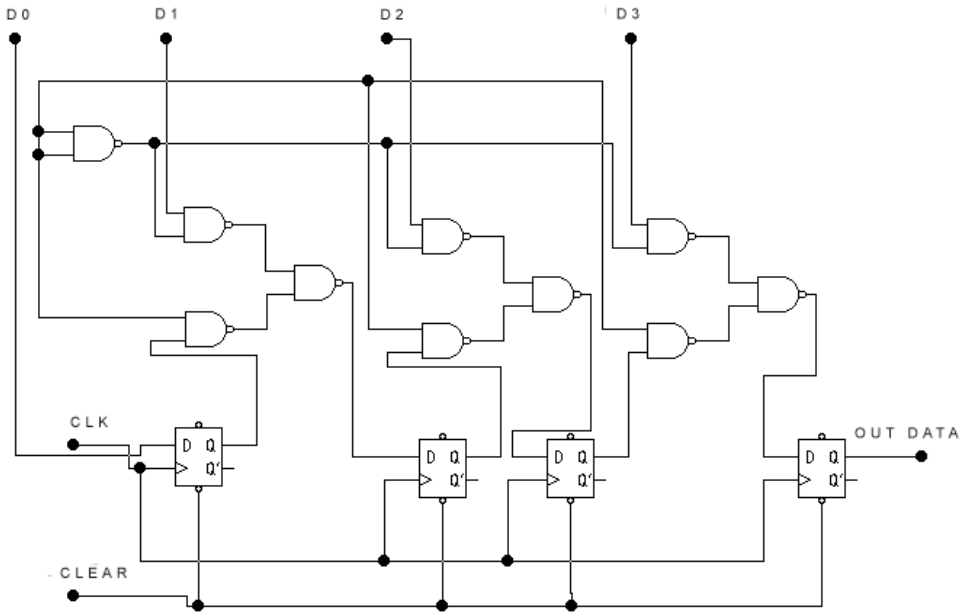


Gambar 6. Aliran data keluar dari SIPO

Tabel kebenaran register SIPO

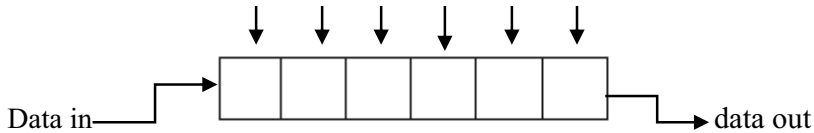
Clock	Input	Q1	Q2	Q3	Q4	A	B	C	D
0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0
2	1	1	1	0	0	0	0	0	0
3	0	0	1	1	0	0	0	0	0
4	1	1	0	1	1	0	0	0	0
		1	0	1	1	1	0	1	1

Register geser *parallel in – serial out* (PISO)



Gambar 7. Register PISO

D0, D1, D2 dan D3 adalah paralel input, dimana D0 adalah *most significant bit* (MSB) dan D3 adalah least significant bit (LSB). Untuk menulis data masuk, baris pengontrolan mode diambil pada rendah dan data di-*clock* masuk. Data dapat digeser saat baris kontrol mode tinggi bersamaan SHIFT aktif tinggi. Register menampilkan operasi geser kanan pada aplikasi satu pulsa *clock*. Ilustrasi keluaran dari register PISO terlihat pada Gambar 8.



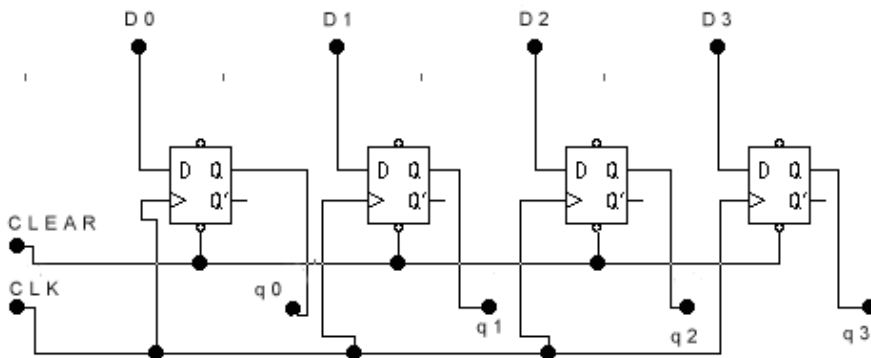
Gambar 8. Aliran data pada PISO

Tabel kebenaran register geser PISO

Clock	D0	D1	D2	D3
0	1	1	0	1
1	1	1	0	1
2	1	0	0	1
3	0	0	0	1

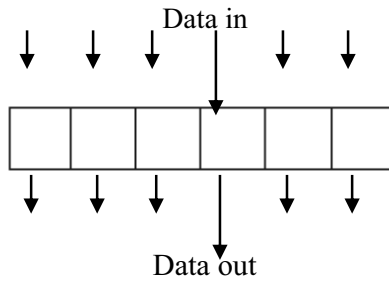
Register geser *parallel In - parallel out* (PIPO)

Untuk register *Parallel In – Parallel Out*, semua bit-bit data muncuk pada keluaran-keluaran paralel secara mendadak mengikuti masukan yang simultan dari bit-bit data.



Gambar 9. Register PIPO

Masukan-masukan D dan keluaran Q adalah paralel. Sekali register di-*clock*, semua data di *D input* muncul pada keluaran Q yang berhubungan secara simultan. Pada register geser ini, keluarannya diilustrasikan oleh Gambar 10.

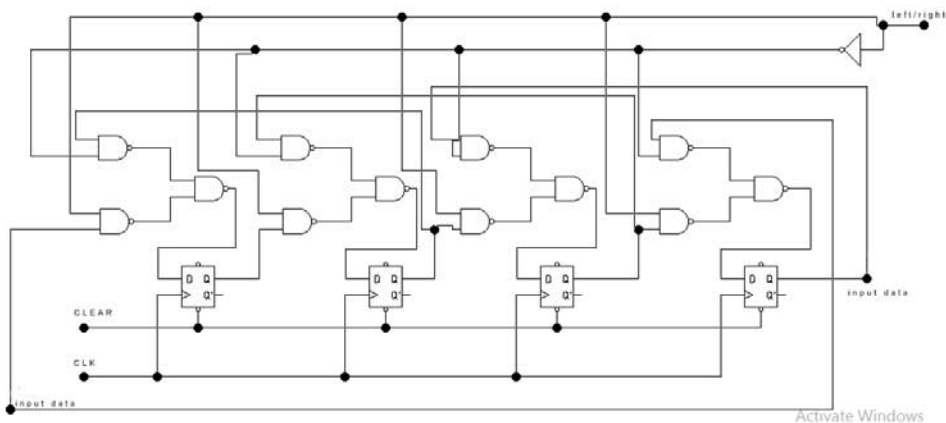


Tabel kebenaran register geser PIPO

Clockl	D1	D2	D3	D4	QD/QA	QC	QB
0	1	1	0	1	0	0	0
1	1	1	0	1	1	1	1
2	1	0	0	1	1	0	1
3	0	0	0	1	0	0	1

Bidirectional Shift Registers

Suatu register dua arah dalam data dijadikan *geser kanan atau kiri*. *Bidirectional Shift Registers* menggunakan *D flip-flop* ditunjukkan dibawah.



Gambar 10. *Bidirectional Shift Registers*

Dalam hal ini kumpulan gerbang NAND dikonfigurasi sebagai gerbang OR untuk memilih data masukan dari dua keadaan stabil yang berdekatan kanan atau kiri (the right or left adjacent bistables), seperti yang dipilih dengan LEFT/RIGHT baris pengontrolan.

Tabel kebenaran *Bidirectional Shift Registers*

S0	S1	CLOCK	S
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

6.4. Counter

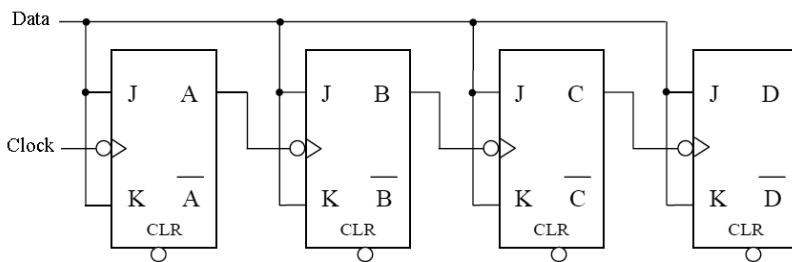
Counter atau disebut dengan pencacah/ penghitung merupakan rangkaian logika sekuensial yang digunakan untuk menghitung jumlah pulsa yang diberikan pada bagian masukan. Counter digunakan di berbagai operasi aritmatika, pembagi frekuensi, penghitung jarak (odometer), penghitung kecepatan, dan lain-lain. Pengembangan dari counter sering digunakan pada instrumen ilmiah, kontrol industri, komputer, alat komunikasi, dan sebagainya. Pada umumnya counter diciptakan dari sederetan flip-flop yang dimanipulasi dengan menggunakan peta karnough, sehingga masukan yang masuk dapat dihitung sesuai dengan rancangan.

Secara umum counter dibagi menjadi dua jenis, yaitu Asynchronous counter (ripple counter) dan synchronous counter. Perbedaan dari kedua jenis counter tersebut adalah pada pemicuannya. Asynchronous counter pemicuannya minimal ada salah satu flip-flop yang clock-nya dipicu oleh keluaran flip-flop lain atau dari sumber clock lain, dan flip-flop pada jenis counter ini tersusun seri. Sedangkan synchronous counter pemicuannya dilakukan secara serentak/ hanya dipicu oleh satu sumber clock, dan flip-flop pada jenis counter ini tersusun paralel.

Selanjutnya berdasarkan arah cacahannya counter dibagi menjadi dua jenis, yaitu counter up (naik) dan counter down (turun). Counter up adalah pencacah yang melakukan cacahan dari kecil ke arah yang besar yang selanjutnya akan m=kembali ke cacahan awal secara otomatis. Sedangkan counter down adalah pencacah yang mealakukan cacahan dari besar ke arah yang kecil hingga cacahan terakhir kemudian kembali ke cacahn awal. Terdapat tiga faktor yang perlu diperhatikan dalam pembuatan counter up maupun counter down, yaitu proses transisi flip-flop yang aktif, yaitu transisi pulsa dari positif ke negatif atau sebaliknya, lalu output flip-flop yang diumpankan ke flip-flop berikutnya, dan indikator hasil cacahan dinyatakan sebagai output.

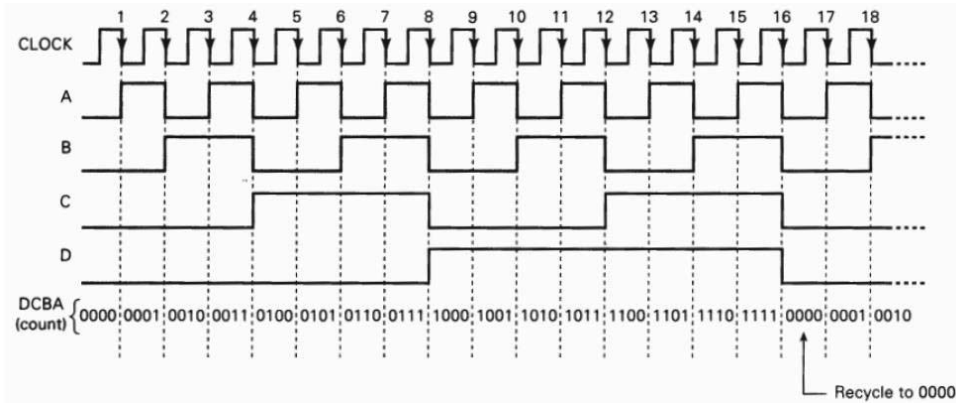
Asynchronous Counter

Asynchronous counter adalah counter yang masukan clock pemicunya tidak terhubung ke setiap flip-fopnya secara langsung. pemicunya harus merambat melalui setiap flip-flop untuk mencapai masukan flip-flop yang berikutnya. Secara garis besar pencacah asinkron terjadi jika keluaran suatu flip-flop memicu flip-flop yang lain.



Gambar 11. Diagram Asynchronous Counter Biner 4-Bit

Prinsip kerja dari counter jenis ini adalah flip-flop A harus berubah keadaannya sebelum dapat memicu flip-flop B, lalu flip-flop B harus berubah keadaannya sebelum dapat memicu flip-flop C, dan flip-flop C harus berubah keadaannya sebelum dapat memicu flip-flop D. Clock/ pemicu bergerak melalui flip-flop bagaikan riak gelombang dalam air. Oleh sebab itu, waktu rambatan keseluruhan merupakan jumlah masing-masing waktu tunda. IC 74LS90 merupakan IC yang memiliki fungsi pencacah asinkron biner 4-bit.



Gambar 12. Bentuk Gelombang dari Rangkaian Asynchronous Counter Biner 4-Bit

Menurut gambar bentuk gelombang di atas, maka proses kerja dari counter jenis ini adalah sebagai berikut.

- 1) Ketika clock pertama tiba, flip-flop A berubah keadaan pada titik perpindahan menuju pulsa negatif. Sehingga hasil masukan pertama menghasilkan kondisi DCBA = 0001. Keluaran A berpindah dari 0 (negatif) menuju 1 (positif). Perubahan positif yang dialami oleh A tidak akan berpengaruh terhadap B, karena perubahan B akan terjadi jika terjadi perubahan negatif dari A.
- 2) Ketika clock kedua tiba, flip-flop A kembali berubah keadaan pada kondisi negatif (0). Perubahan A dari 1 menuju 0, mengakibatkan picuan terhadap B. Sehingga B mengalami perubahan dari 0 menuju 1. Perubahan positif pada B tidak memberikan dampak terhadap C, sehingga kondisi akhir dari keluaran DCBA = 0010.
- 3) Selanjutnya ketika clock ketiga tiba, A akan berubah kembali menjadi 1. Perubahan positif ini tidak akan berpengaruh atau berdampak terhadap flip-flop lainnya, sehingga kondisi akhir dari keluaran DCBA = 0011.
- 4) Pada picuan clock keempat, maka A akan berubah dari 1 menuju 0. Perubahan negatif ini akan berpengaruh terhadap B, sehingga B akan berubah dari 1 menuju 0. Perubahan negatif B juga berpengaruh terhadap C, sehingga C akan berubah dari 0 menuju 1. Dengan demikian kondisi akhir dari keluaran DCBA = 0100.

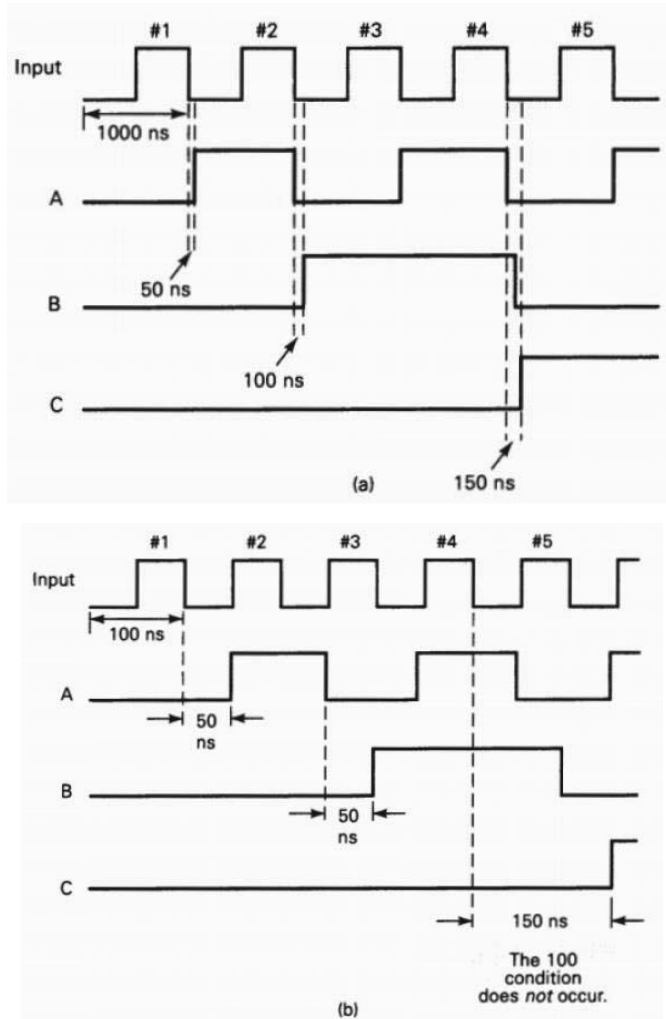
- 5) Proses di atas terjadi secara terus menerus sampai kondisi keluaran DCBA = 1111 dan akan kembali pada kondisi keluaran DCBA = 0000. Selanjutnya akan kembali pada proses pertama.

Dari proses kerja yang telah disampaikan di atas, maka akan menghasilkan tabel kebenaran sebagai berikut.

Tabel 1. Tabel Kebenaran Asynchronous Counter Biner 4-Bit

Pulsa Clock ke-	D	C	B	A
1	0	0	0	0
2	0	0	0	1
3	0	0	1	0
4	0	0	1	1
5	0	1	0	0
6	0	1	0	1
7	0	1	1	0
8	0	1	1	1
9	1	0	0	0
10	1	0	0	1
11	1	0	1	0
12	1	0	1	1
13	1	1	0	0
14	1	1	0	1
15	1	1	1	0
16	1	1	1	1

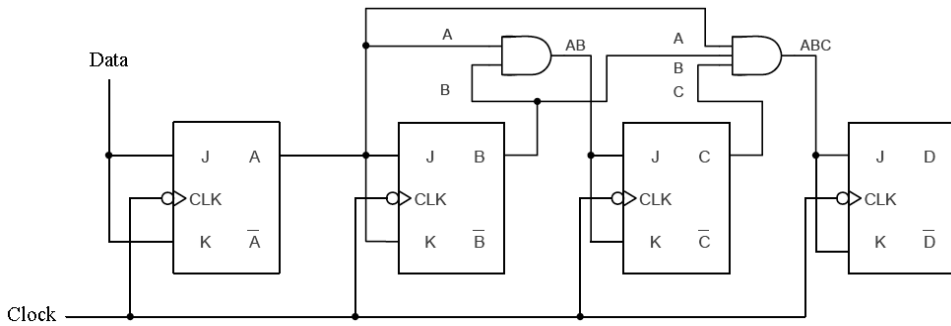
Pencacah asinkron memiliki salah satu hambatan dalam proses kerjanya, yaitu waktu tunda propagasi flip-flop yang diakumulasi. Masalah tersebut terjadi karena flip-flop yang terlibat tidak mengubah kondisinya secara bersamaan dengan pulsa masukan clock. Sehingga jika frekuensi clock yang tinggi dan banyak flip-flop yang dirangkai, maka akan menimbulkan kesalahan dalam proses pencacahan.



Gambar 13. Gelombang Keluaran dari Pencacah Asinkron Menggambarkan Waktu Tunda Propagasi

Synchronous Counter

Synchronous counter (pencacah sinkron) merupakan pencacah yang masukan clock pemicunya dihubungkan ke setiap flip-flop, sehingga flip-flop akan beroperasi dengan masukan transisi clock yang sama. pencacah sinkron harus dipastikan bahwa setiap flip-flop menahan transisi luaranny sampai tiba gilirannya. Pencacah sinkron digunakan untuk menutupi keterbatasan yang terdapat pada pencacah asinkron, yaitu waktu tunda propagasi yang disebabkan oleh tidak samanya perubahan flip-flop dengan clock. IC 74LS190 adalah IC yang memiliki fungsi pencacah sinkron biner 4-bit.



Gambar 14. Diagram Synchronous Counter Biner 4-Bit

Secara garis besar prinsip kerja dari pencacah sinkron adalah sama dengan pencacah asinkron, sehingga keluaran yang dihasilkan oleh pencacah sinkron akan sama dengan pencacah asinkron.

Tabel 2. Tabel Kebenaran Synchronous Counter Biner 4-Bit

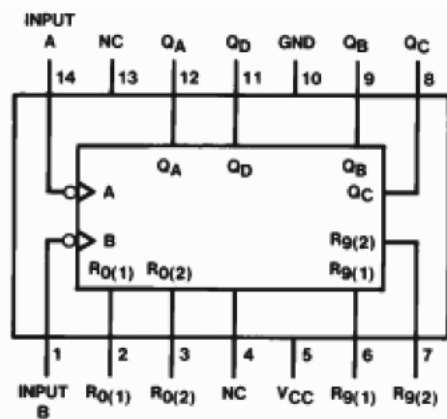
Pulsa Clock ke-	D	C	B	A
1	0	0	0	0
2	0	0	0	1
3	0	0	1	0
4	0	0	1	1
5	0	1	0	0
6	0	1	0	1
7	0	1	1	0
8	0	1	1	1
9	1	0	0	0
10	1	0	0	1
11	1	0	1	0
12	1	0	1	1
13	1	1	0	0
14	1	1	0	1
15	1	1	1	0
16	1	1	1	1

Pencacah sinkron dibagi menjadi dua jenis, yaitu pencacah sinkron yang menggunakan series carry (ripple-through carry) dan yang menggunakan parallel carry. Pencacah sinkron dengan series carry terdapat gerbang AND antara flip-flop C dan flip-flop D yang harus menunggu keluaran dari AND sebelumnya, sehingga waktu propagasinya akan lebih buruk dibandingkan dengan pencacah sinkron yang menggunakan parallel carry. Pencacah sinkron dengan parallel carry memiliki waktu propagasi yang lebih baik daripada yang menggunakan series carry. Namun parallel carry juga memiliki kelemahan, yaitu mensyaratkan fan-in

yang tinggi pada gerbang AND (dua masukan untuk gerbang AND pertama, tiga masukan untuk bergang AND kedua, dan seterusnya) dan beban yang berat untuk flip-flop di bagian awal deret (pada pencacah 4-bit, fan-put untuk flip-flop A adalah $4-1=3$, dan seterusnya).

Decimal Counter

Decimal counter (pencacah desimal) adalah pencacah yang digunakan untuk melakukan perhitungan dalam bentuk bilangan desimal. Pencacah jenis ini paling sering digunakan di industri dan keperluan aplikatif lainnya. Sistem digit desimal tersebut dalam diwakili satu sekuens digit biner dengan mekanisme encoding (penyandian) yang disebut Binary Coded Decimal (BCD). BCD dapat mewakili 4 bit pada biner menjadi sebuah digit desimal yang mempresentasikan digit 0-9. Pada umumnya pencacah dekade yang digunakan adalah BCD 8421, meskipun demikian kita tetap dapat menggunakan sandi-sandi BCD lainnya. Secara umum industri-industri menggunakan IC 7490 sebagai pencacah dekade. IC 7490 dapat mencacah dalam sandi BCD 8421 dari 0000-1001, kemudian pencacah akan direset kembali ke 0. IC 7490 menggunakan seven segment sebagai keluaran dari bilangan desimal yang dihasilkan.



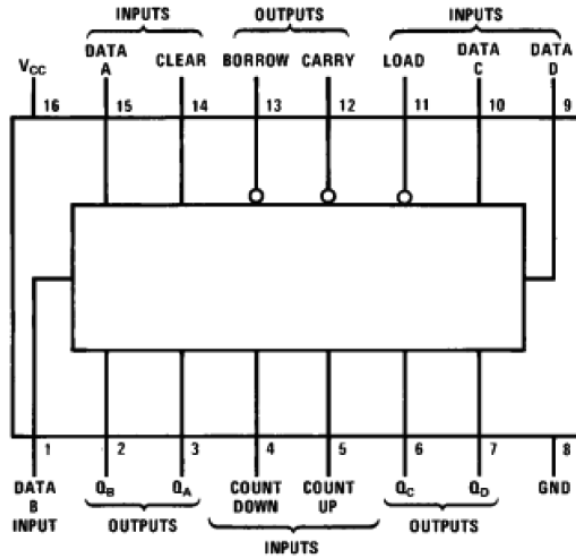
Gambar 15. Diagram IC 7490 Pencacah Desimal

Tabel 3. Tabel Kebenaran Pencacah Desimal

Input Clock	Output Seven Segment
1	0
2	1
3	2
4	3
5	4
6	5
7	6
8	7
9	8
10	9
11	0
12	1

Rangkaian Counter Up/ Down

Counter up (pencacah naik) merupakan pencacah yang digunakan untuk menghitung dari bilangan kecil menuju bilangan besar. Sedangkan counter down (pencacah turun) merupakan pencacah yang digunakan untuk menghitung dari bilangan besar menuju bilangan kecil. Rangkaian pencacah naik maupun turun dapat dibangun dari pencacah jenis asinkron maupun sinkron seperti yang telah dicontohkan pada pembahasan sebelumnya. Namun secara praktis pencacah naik dan turun ini dapat langsung dibangun tanpa menggunakan IC flip-flop yang dirangkai sedemikian rupa. Pencacah naik dan turun dapat dibangun menggunakan IC 74192 dan IC 74193. IC 74192 merupakan IC dekade pencacah naik/ turun yang dapat mencacah dari nilai 0000-1001 biner atau 0-9 desimal. Sedangkan IC 74193 merupakan IC dekade pencacah naik/ turun yang dapat mencacah dari nilai 0000-1111 biner atau 0-15 desimal.



Gambar 16. Diagram IC 74193

Proses pemilihan operasi antara pencacah naik dan pencacah turun dilakukan dengan suatu rangkaian kendali yang memanfaatkan gerbang-gerbang logika. Gerbang logika yang digunakan adalah gerbang NAND, yaitu apabila salah satu input berlogika 0 maka output akan selalu berlogika 1, sehingga kondisi ini dapat menginci output pada satu kondisi seskipun kondisi input kaki yang lain berubah-ubah.

Tabel 4. Tabel Kebenaran Counter Up

Input Clock	Output Seven Segment
1	0
2	1
3	2
4	3
5	4
6	5
7	6
8	7
9	8
10	9
11	0
12	1

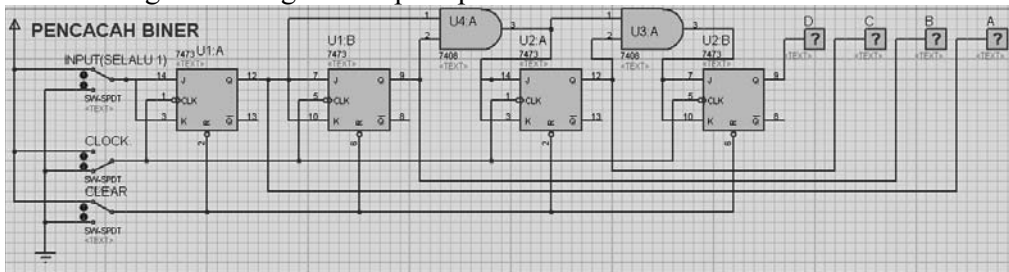
Tabel 5. Tabel Kebenaran Counter Down

Input Clock	Output Seven Segment
1	0
2	9
3	8
4	7
5	6
6	5
7	4
8	3
9	2
10	1
11	0
12	9

6.5. Simulasi Rangkaian

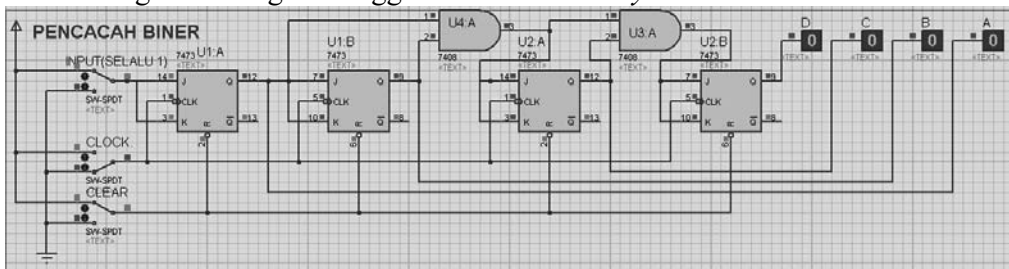
Simulasi Rangkaian Pencacah Biner

1. Masukkan Komponen 7segmen, logicprobe, sw-spdt, IC 7408 7447, 7473, 7490 pada Proteus.
2. Rangkailah rangkaian seperti pada Gambar 17.



Gambar 17. Rangkaian Pencacah Biner

3. Setelah selesai merangkai rangkaian sesuai pada Gambar 17 simulasikan rangkaian dengan menggunakan tombol *Play*



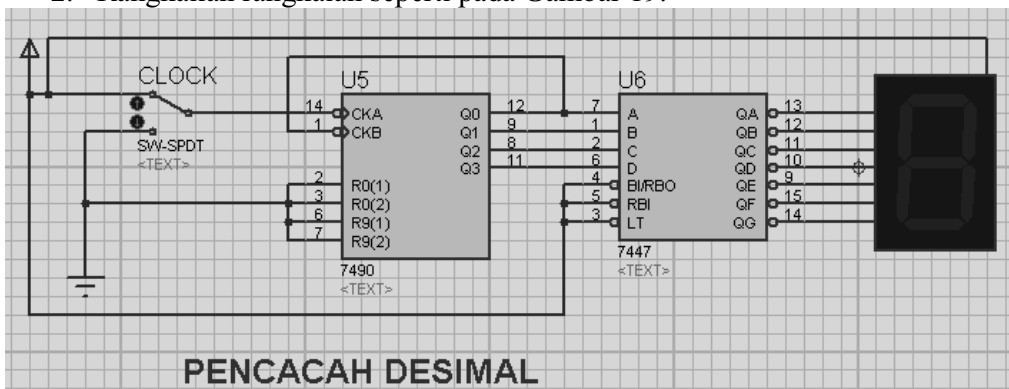
Gambar 18. Simulasi Rangkaian Pencacah Biner

Tabel Percobaan Pencacah Biner

Input Clock	Output			
	D	C	B	A
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				

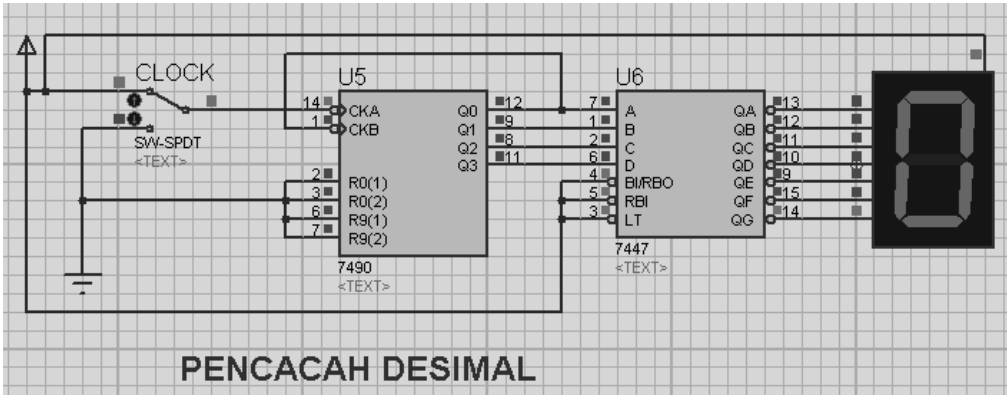
Simulasi Rangkaian Pencacah Desimal

1. Masukkan Komponen 7segmen, logicprobe, sw-spdt, IC 7408 7447, 7473, 7490 pada Proteus.
2. Rangkailah rangkaian seperti pada Gambar 19.



Gambar 19. Gambar Rangkaian Pencacah Desimal

3. Setelah selesai merangkai rangkaian sesuai pada Gambar 19 simulasikan rangkaian dengan menggunakan tombol *Play*



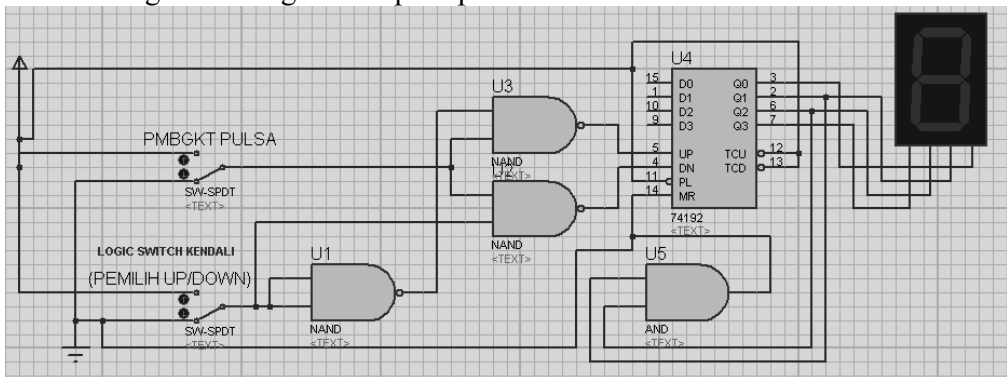
Gambar 20. Gambar Simulasi Rangkaian Pencacah Desimal

Tabel Percobaan Pencacah Desimal

Input Clock	Output 7 Segment
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	

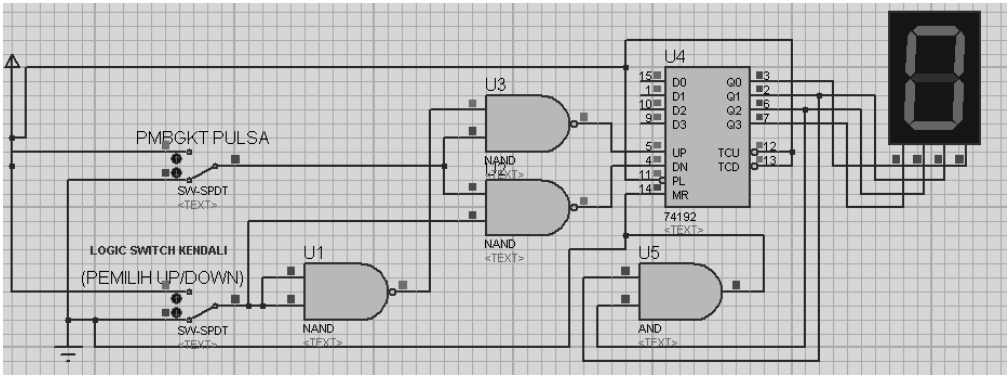
Simulasi Pencacah MOD-6

1. Masukkan Komponen 7segmen, sw-spd, Gerbang NAND dan AND, IC 74192
2. Rangkailah rangkaian seperti pada Gambar 21.



Gambar 21. Rangkaian Pencacah MOD-6

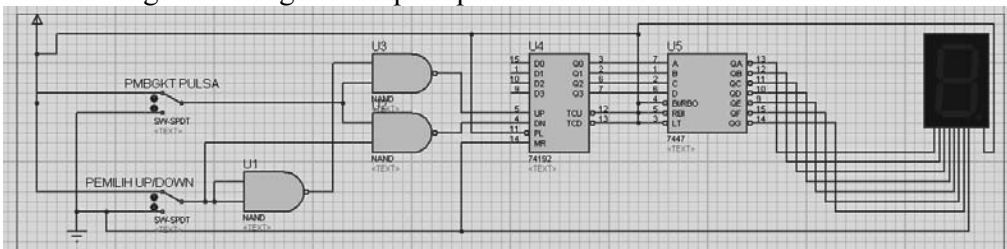
3. Setelah selesai merangkai rangkaian sesuai pada Gambar 21 simulasikan rangkaian dengan menggunakan tombol *Play*



Gambar 22. Simulasi Rangkaian Pencacah MOD-6

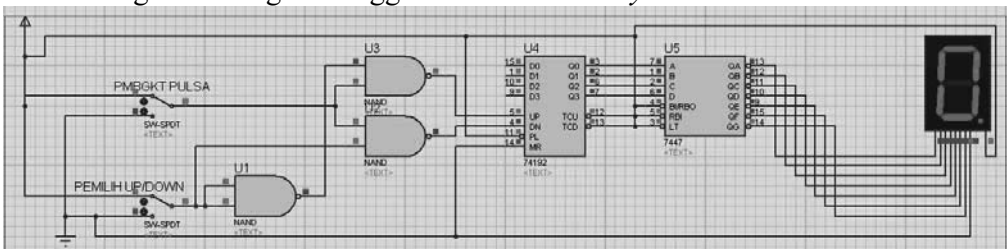
Simulasi Counter Up/Down dengan IC 7447

1. Masukkan Komponen 7segmen, sw-spdt, Gerbang NAND, IC 74192, IC 7447
2. Rangkailah rangkaian seperti pada Gambar 23.



Gambar 22. Rangkaian Counter Up/Down dengan IC 7447

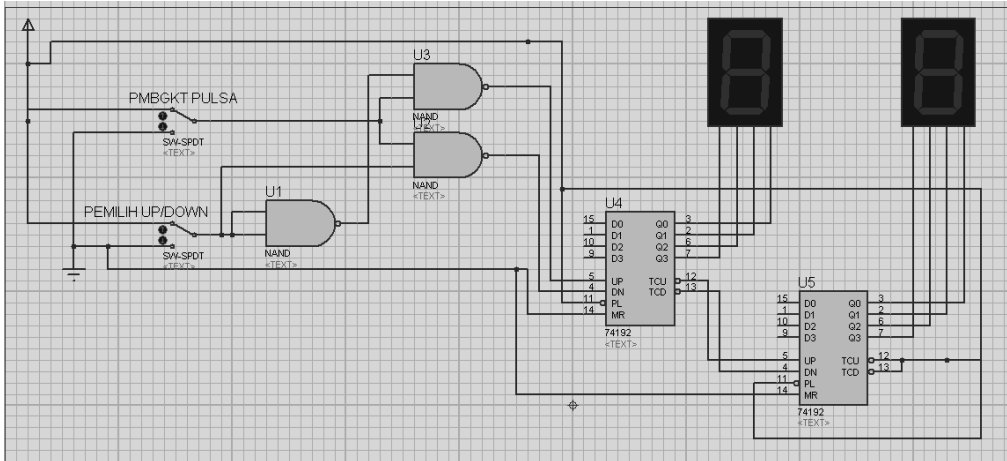
3. Setelah selesai merangkai rangkaian sesuai pada Gambar 22 simulasikan rangkaian dengan menggunakan tombol Play



Gambar 23. Simulasi Rangkaian Counter Up/Down dengan IC 7447

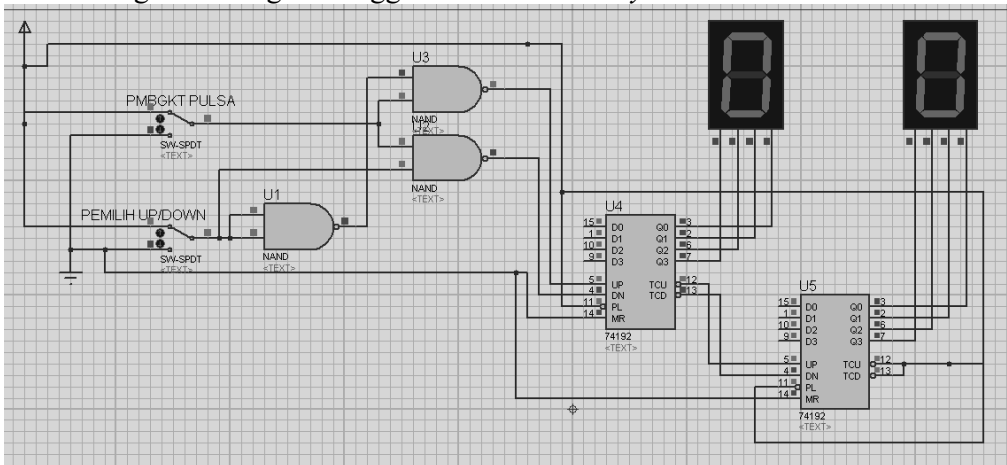
Simulasi Counter Up/Down Dua Digit

1. Masukkan Komponen 7segmen, sw-spdt, Gerbang NAND, IC 74192, IC 7447
2. Rangkailah rangkaian seperti pada Gambar 24.



Gambar 24. Rangkaian Counter Up/Down Dua Digit

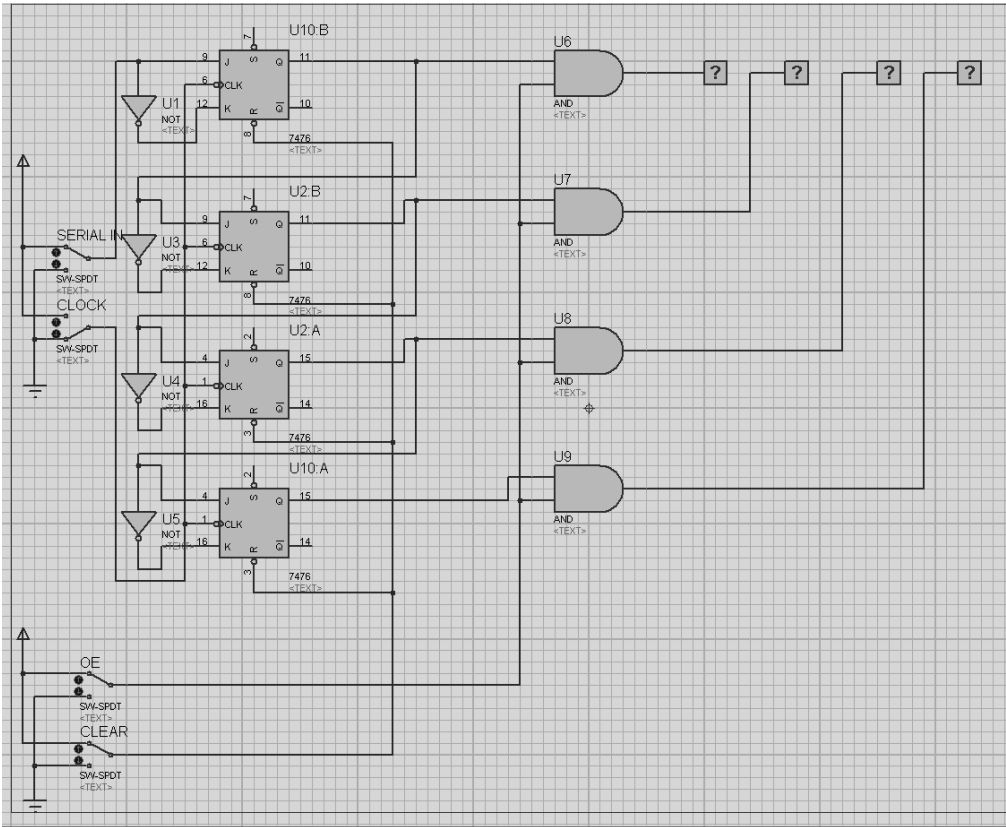
3. Setelah selesai merangkai rangkaian sesuai pada Gambar 24 simulasikan rangkaian dengan menggunakan tombol *Play*



Gambar 25. Simulasi Rangkaian Counter Up/Down Dua Digit

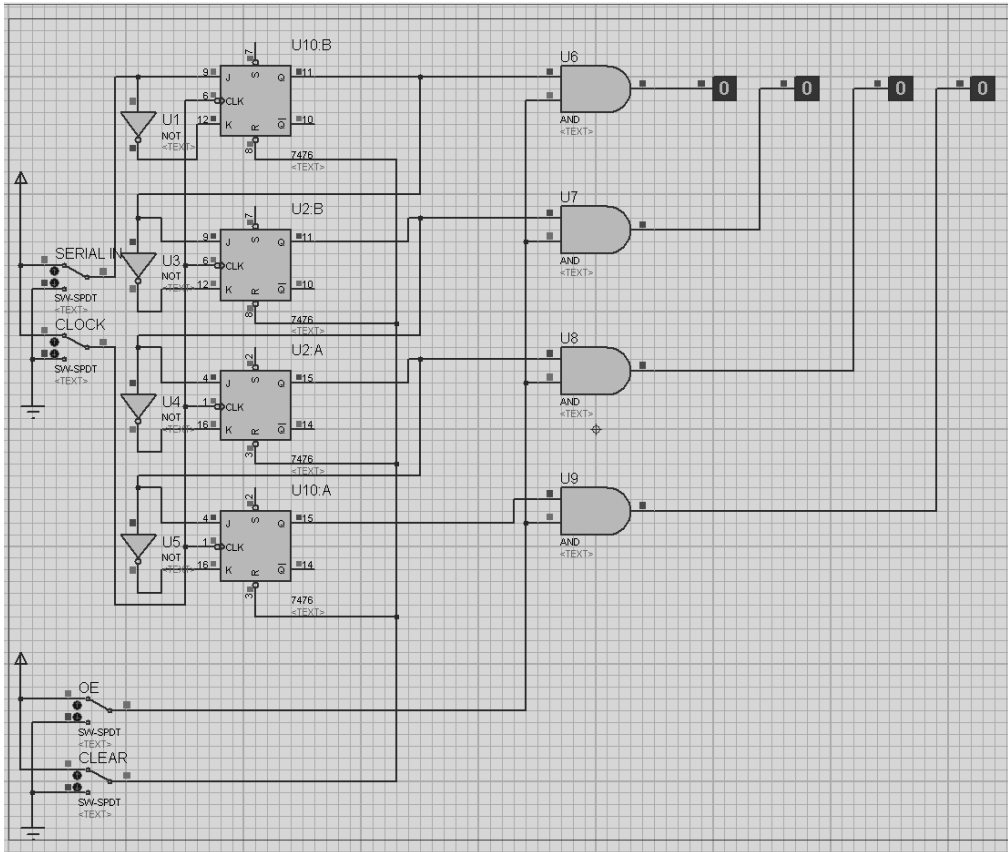
Simulasi Rangkaian Register Geser Serial In-Paralel Out

1. Masukkan Komponen Gerbang AND dan NOT, logicprobe, sw-spd, IC 7476
2. Rangkailah rangkaian seperti pada Gambar 26.



Gambar 26. Rangkaian Register geser Serial In-Paralel Out

3. Setelah selesai merangkai rangkaian sesuai pada Gambar 26 simulasikan rangkaian dengan menggunakan tombol *Play*



Gambar 27. Simulasi Register geser Serial In-Paralel Out

Tabel Percobaan Register Geser Serial In-Paralel Out

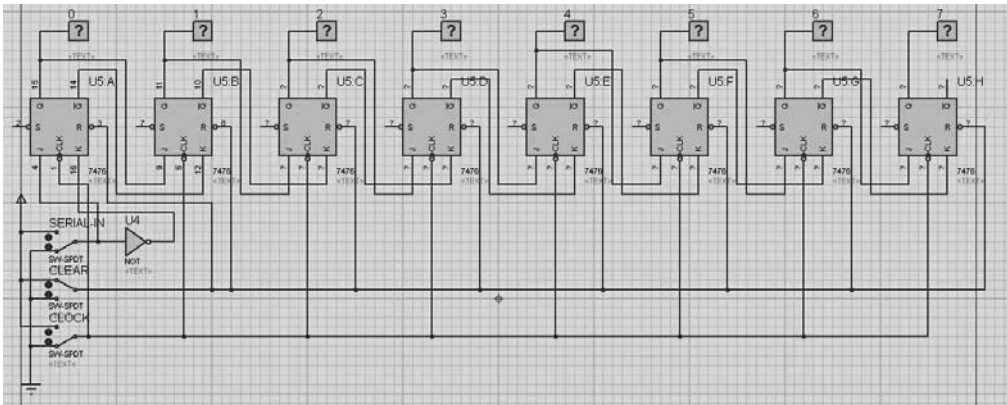
No	Serial –In pada Clock ke-				Output			
	1	2	3	4	A	B	C	D
1	1	1	1	0				
2	1	1	1	1				
3	1	1	0	0				
4	1	1	0	1				
5	1	0	1	0				
6	1	0	1	1				
7	1	0	0	0				
8	1	0	0	1				

Tabel Percobaan Register Geser Serial In-Paralel Out

No	Serial In pada clock ke-							Output D(0) pada clock ke-						
	1	2	3	4	5	6	7	1	2	3	4	5	6	7
1	1	1	1	0	0	0	0							
2	1	1	1	1	0	0	0							
3	1	1	0	0	0	0	0							
4	1	1	0	1	0	0	0							
5	1	0	1	0	0	0	0							
6	1	0	1	1	0	0	0							
7	1	0	0	0	0	0	0							
8	1	0	0	1	0	0	0							

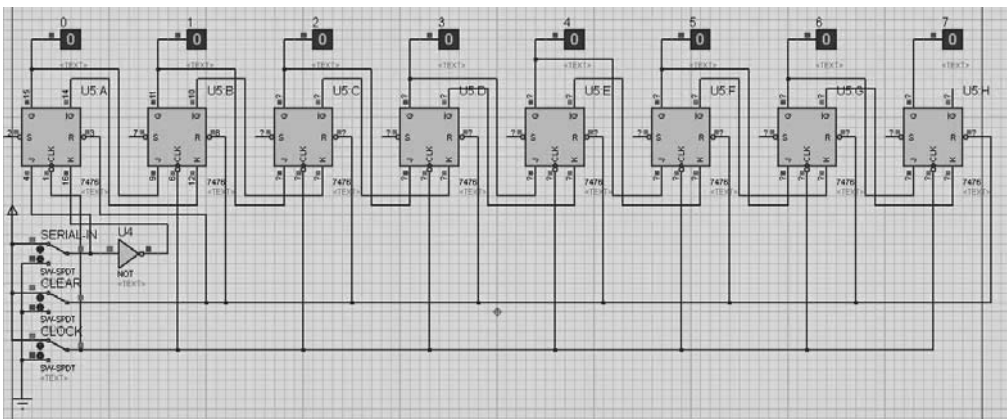
Simulasi Rangkaian Konverter Serial-Paralel 8 bit

1. Masukkan Komponen Gerbang NOT, logicprobe, sw-spd, IC 7476, IC 7473
2. Rangkailah rangkaian seperti pada Gambar 28



Gambar 28. Rangkaian Konverter Serial-Paralel 8 bit

3. Setelah selesai merangkai rangkaian sesuai pada Gambar 28 simulasikan rangkaian dengan menggunakan tombol *Play*



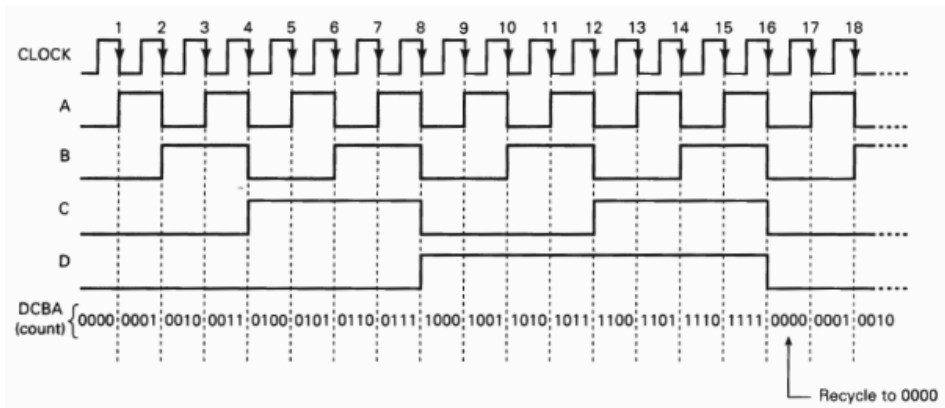
Gambar 29. Simulasi Rangkaian Konverter Serial-Paralel 8 bit

Tabel Percobaan Konverter Serial-Paralel 8 bit

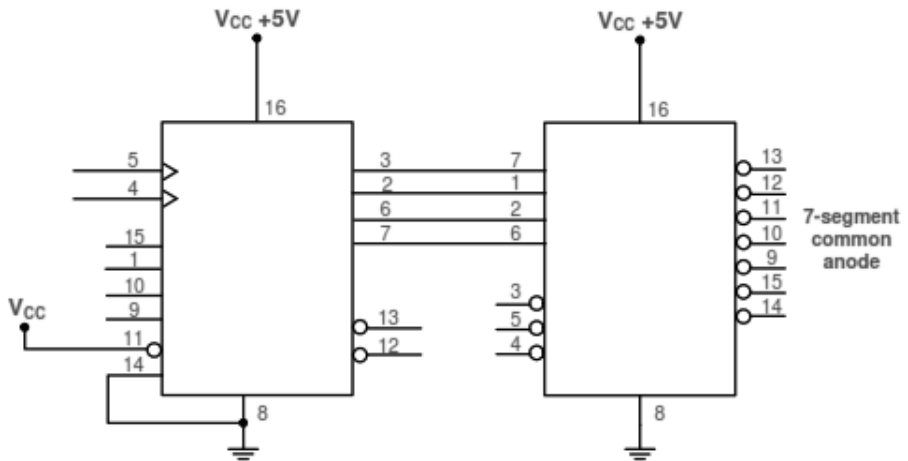
Serial In	Clock ke-	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
1	1								
1	2								
0	3								
1	4								
1	5								
0	6								
1	7								
0	8								

6.6. Latihan Soal

- Suatu unit yang mencacah banyaknya picu masukan disebut ...
 - Flip-flop
 - D flip-flop
 - RS flip-flop
 - Counter**
 - Register
- Pencacah yang menggunakan keluaran suatu flip-flop untuk memicu flip-flop lain disebut ...
 - Pencacah Biner
 - Pencacah Sinkron
 - Pencacah Asinkron
 - Pencacah Seri
 - Pencacah Paralel
- Banyak flip-flop yang dibutuhkan untuk membuat pencacah biner 32 bitadalah ...
 - 3
 - 4
 - 5**
 - 6
 - 7



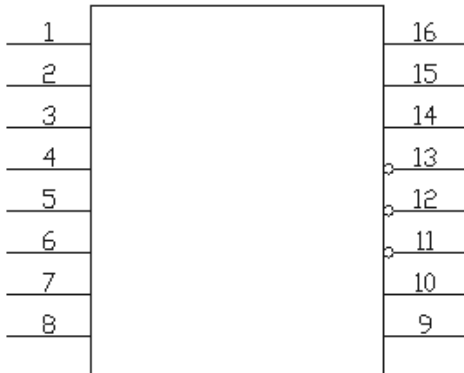
- gambar diatas merupakan bentuk gelombang dari ...
 - Pencacah biner 2-bit
 - Pencacah biner 4-bit
 - Pencacah biner 8-bit
 - Pencacah biner 16-bit**
 - Pencacah biner 32-bit



5.

IC yang dibutuhkan untuk membuat sebuah up/down counter seperti gambar diatas adalah ...

- a. 7473 dan 7400
- b. 7447 dan 7400
- c. 74192 dan 7473
- d. 7473 dan 7447
- e. **74192 dan 7447**

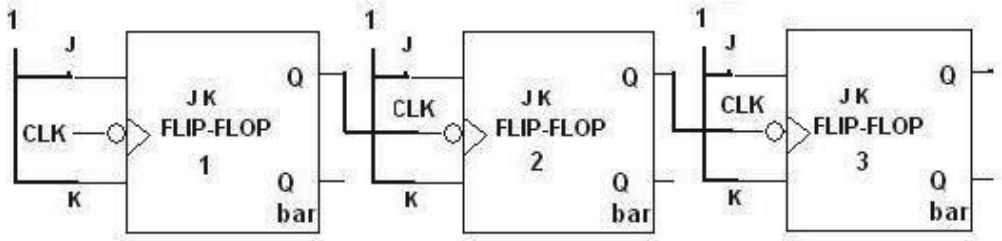


74192

6.

IC 74192 seperti gambar diatas memiliki kaki count up dan count down pada pin nomor ...

- a. 3 dan 2
- b. 7 dan 6
- c. **5 dan 4**
- d. 10 dan 9
- e. 13 dan 12



7. Counter pada gambar diatas mencacah bilangan antara ...
- 0 - 3
 - 0 - 4
 - 0 - 5
 - 0 - 6
 - 0 - 7**
8. Register yang memiliki semua bilangan biner yang dimuat dalam satu bit pada suatu waktu dan memiliki semua bit tersimpan yang dimuat dalam satu waktu adalah ...
- Paralel-in, paralel-out
 - Paralel-in, serial-out
 - Serial-in, paralel-out**
 - Serial-in, serial out
 - Semua jawaban salah
9. Sebuah register geser serial in/serial out yang digunakan dengan waktu delay 20 ns memiliki frekuensi clock sebesar ...
- 20 kHz
 - 40 kHz
 - 50 kHz
 - 400 kHz**
 - 500 kHz
10. Ketika sebuah output dari register geser tristate dimatikan, level output ditempatkan pada ...
- Kondisi float
 - Kondisi LOW
 - Kondisi high impedance
 - Kondisi float atau high impedance**
 - Semua jawaban benar

BAB 7.

RANGKAIAN ARITMATIKA DAN LOGIKA

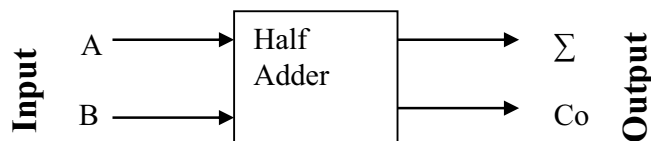
Dalam menjumlahkan suatu bilangan pada umumnya dimulai dengan menjumlahkan digit disebelah kanan, yaitu digit yang mempunyai bobot yang paling kecil (LSB). Dilanjutkan dengan menjumlahkan kolom berikutnya dengan memperhatikan apakah ada nilai pindahan (*Carry*) yang harus dijumlahkan. Dalam rangkaian logika cara penjumlahan ini sering disebut *ADDER* (penjumlahan). Adapun fungsinya adalah untuk menjumlahkan, mengurangi, mengali, dan membagi angka-angka biner dimana dalam pelaksanaannya dapat dianggap sebagai cara penjumlahan. Berdasarkan cara penggunaannya *ADDER* dapat dibagi menjadi: *Half Adder*, *Full Adder*.

Seperti halnya dengan rangkaian *ADDER*, maka perhitungan yang dilakukan oleh rangkaian dasar pengurang (*Subtractor*) dapat dilakukan secara langsung. Artinya dilakukan dimulai dari digit yang disebelah kanan dilanjutkan dengan pengurangan dari kolom berikutnya dengan memperhatikan apakah ada nilai pinjaman (*Borrow*) yang harus dikurangkan dan kalau mungkin ada selisihnya (*Difference*). Dalam rangkaian logika cara pengurangan seperti ini disebut *SUBTRACTOR* (pengurangan). Berdasarkan penggunaannya *SUBTRACTOR* dapat dibagi menjadi: *Half Subtractor*, *Full Subtractor*

Untuk membandingkan 2 buah bilangan digunakan *COMPARATOR*. *Comparator* tersebut digunakan untuk membandingkan 2 buah bilangan apakah lebih besar ($>$), lebih kecil ($<$), atau sama dengan ($=$).

7.1. Rangkaian Half Adder

Sebuah rangkaian *Adder* terdiri dari *Half Adder* dan *Full Adder*. *Half Adder* menjumlahkan dua buah bit input, dan menghasilkan nilai jumlahan (sum) dan nilai lebihnya (*carry-out*). *Half Adder* diletakkan sebagai penjumlah dari bit-bit terendah (*Least Significant Bit*). Blok Diagram dari sebuah rangkaian *Half Adder* ditunjukkan pada Gambar 7.1.

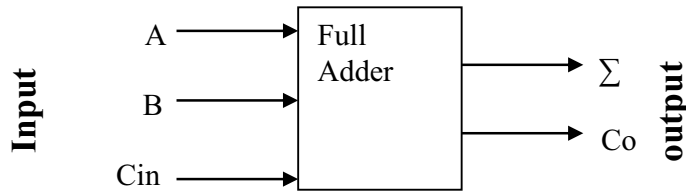


Gambar 1. Blok Diagram *Half Adder*

7.2. Rangkaian Full Adder

Sebuah *Full Adder* menjumlahkan dua bilangan yang telah dikonversikan menjadi bilangan-bilangan biner. Masing-masing bit pada posisi yang sama saling dijumlahkan. *Full Adder* sebagai penjumlah pada bit-bit selain yang terendah. *Full Adder* menjumlahkan dua bit input ditambah dengan nilai *Carry-Out* dari penjumlahan bit sebelumnya. *Output* dari *Full Adder* adalah hasil penjumlahan

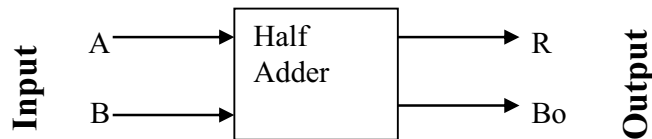
(Sum) dan bit kelebihannya (*carry-out*). Blok diagram dari sebuah *full adder* diberikan pada Gambar 7.2.1.



Gambar 2. Blok Diagram *Half Adder*

7.3. Simulasi Rangkaian *Half Subtractor*

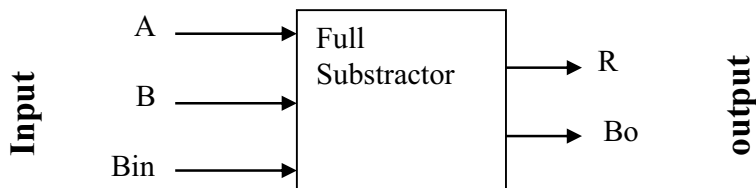
Sebuah rangkaian *Subtractor* terdiri dari *Half Subtractor* dan *Full Subtractor*. *Half Subtractor* mengurangi dua buah bit input, dan menghasilkan nilai hasil pengurangan (*Remain*) dan nilai yang dipinjam (*Borrow-out*). *Half Subtractor* diletakkan sebagai pengurang dari bit-bit terendah (*Least Significant Bit*). Blok Diagram dari sebuah rangkaian *Half Subtractor* ditunjukkan pada gambar 7.3.1.



Gambar 3. Blok Diagram *Half Subtractor*

7.4. Simulasi Rangkaian *Full Subtractor*

Sebuah *Full Subtractor* mengurangi dua bilangan yang telah dikonversikan menjadi bilangan-bilangan biner. Masing-masing bit pada posisi yang sama saling dikurangkan. *Full Subtractor* mengurangi dua bit input dan nilai *Borrow-Out* dari pengurangan bit sebelumnya. Output dari *Full Subtractor* adalah hasil pengurangan (*Remain*) dan bit pinjamannya (*borrow-out*). Blok diagram dari sebuah *full subtractor* diberikan pada gambar 7.4.1.

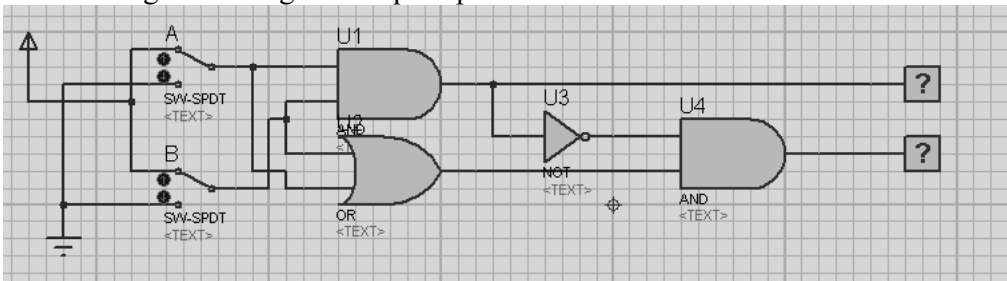


Gambar 4. Blok Diagram *Full Subtractor*

7.5. Simulasi Rangkaian dengan Proteus

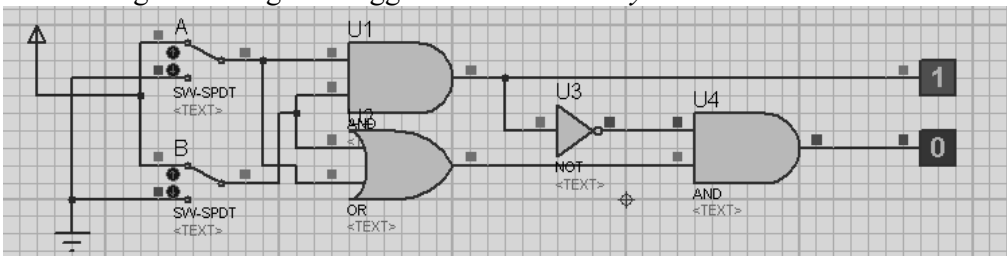
Simulasi Rangkaian Half Adder

1. Masukkan Komponen Gerbang AND, NOT, OR, Logicprobe, sw-spdt
2. Rangkailah rangkaian seperti pada Gambar 5.



Gambar 5. Rangkaian Half Adder

3. Setelah selesai merangkai rangkaian sesuai pada Gambar 5 simulasikan rangkaian dengan menggunakan tombol *Play*



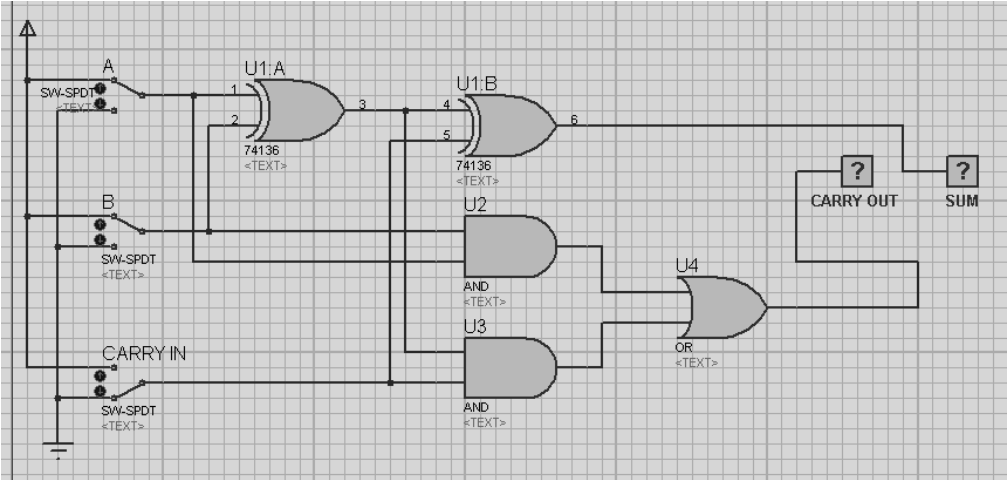
Gambar 6. Simulasi Half Adder

Tabel Percobaan Half Adder

Input		Output	
A	B	Carry	Sum
0	0		
0	1		
1	0		
1	1		

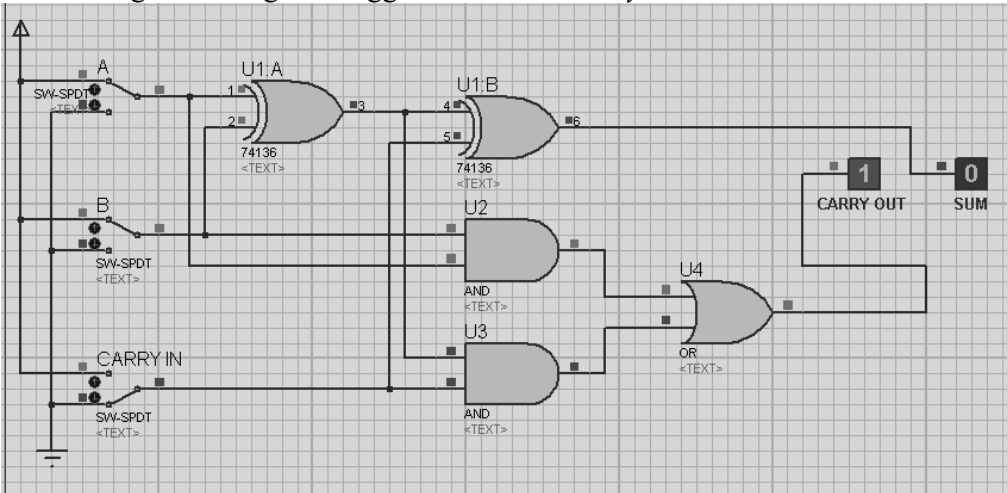
Simulasi Rangkaian Full Adder

1. Masukkan Komponen Gerbang AND, OR, Logicprobe, sw-spdt, IC 74136
2. Rangkailah rangkaian seperti pada Gambar 7.



Gambar 7 Rangkaian Full Adder

3. Setelah selesai merangkai rangkaian sesuai pada Gambar 7 simulasikan rangkaian dengan menggunakan tombol *Play*



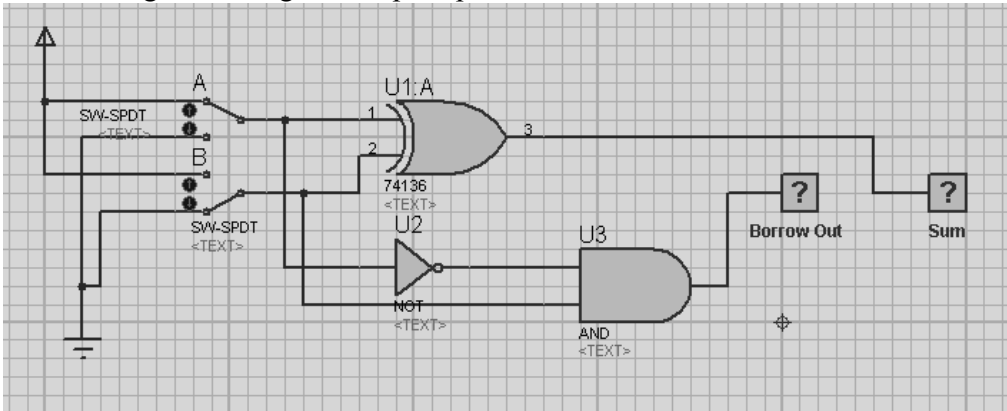
Gambar 8 Simulasi Rangkaian Full Adder

Tabel Percobaan Full Adder

No	Input			Output	
	A	B	Carry In	Carry Out	Sum
1	0	0	0		
2	0	0	1		
3	0	1	0		
4	0	1	1		
5	1	0	0		
6	1	0	1		
7	1	1	0		
8	1	1	1		

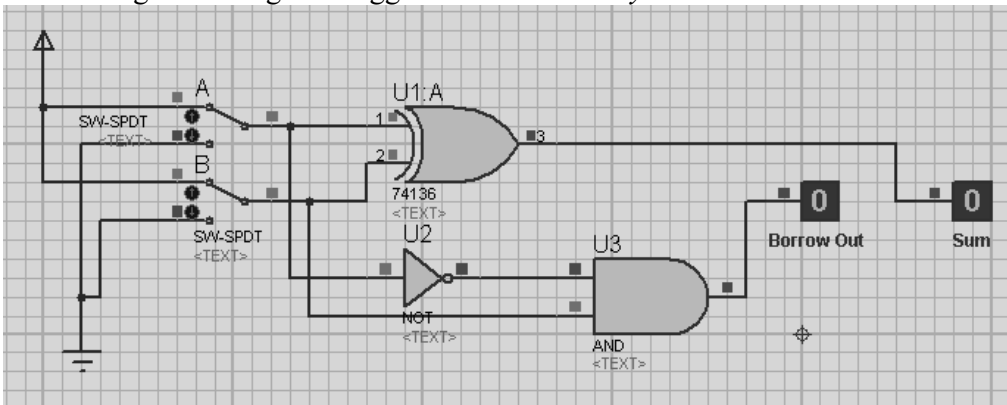
Simulasi Rangkaian Half Subtractor

1. Masukkan Komponen Gerbang AND, NOT, Logicprobe, sw-spdt, IC 74136
2. Rangkailah rangkaian seperti pada Gambar 9.



Gambar 7.9 Rangkaian Half Subtractor

3. Setelah selesai merangkai rangkaian sesuai pada Gambar 9 simulasikan rangkaian dengan menggunakan tombol *Play*



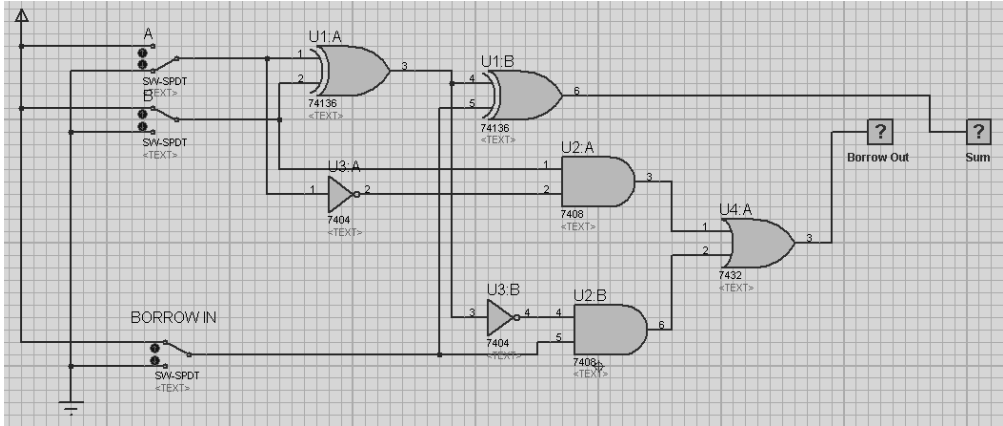
Gambar 10. Simulasi Half Subtractor

Tabel Percobaan Half Adder

Input		Output	
A	B	Borrow	Sum
0	0		
0	1		
1	0		
1	1		

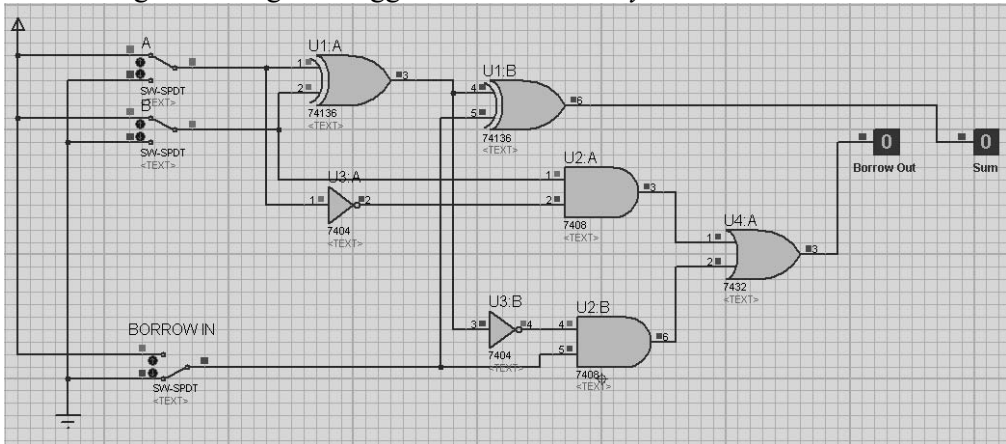
Simulasi Rangkaian Full Subtractor

1. Masukkan Komponen IC 7404, 7408, 7432, 74136, Logicprobe, sw-spdt
2. Rangkailah rangkaian seperti pada Gambar 11.



Gambar 11. Rangkaian Full Subtractor

3. Setelah selesai merangkai rangkaian sesuai pada Gambar 11. simulasikan rangkaian dengan menggunakan tombol *Play*



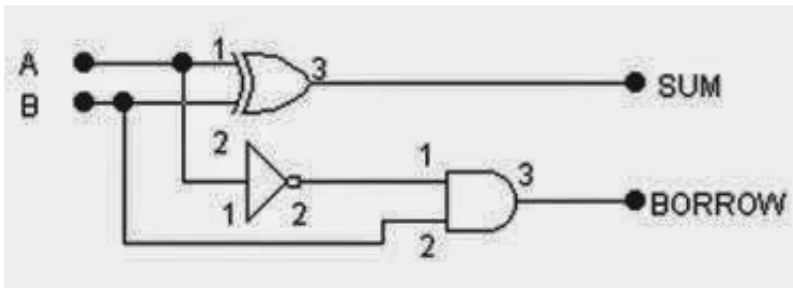
Gambar 12. Simulasi Full Suttractor

Tabel Percobaan Full Adder

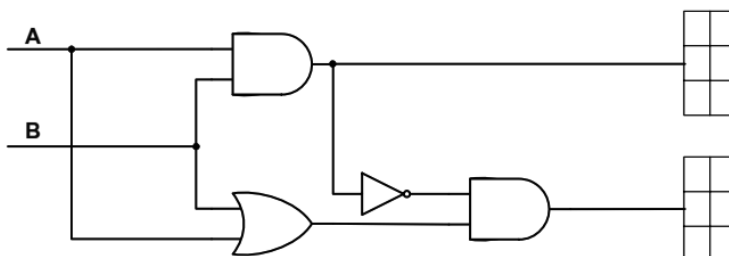
No	Input			Output	
	A	B	Borrow In	Borrow Out	Sum
1	0	0	0		
2	0	0	1		
3	0	1	0		
4	0	1	1		
5	1	0	0		
6	1	0	1		
7	1	1	0		
8	1	1	1		

7.6. Latihan Soal

- Sisa desimal dari pembagian 1100010 dengan 0101 adalah ...
 - 2
 - 3
 - 4
 - 6
 - 8
- Dua tipe dari sirkuit dasar adder adalah ...
 - Half adder dan full adder
 - Half adder dan paralel adder
 - Asinkron dan sinkron
 - Komplimen satu dan komplimen dua
 - Multivibrator stabil dan bistabil
- Dua terminal output dari rangkaian full subtractor adalah ...
 - Sum dan carry
 - Sum dan borrow
 - Borrow dan carry
 - Difference dan borrow
 - Semua jawaban salah
- Kelebihan dari jumlah penambahan dalam operasi half dan full adder disebut ...
 - Sum
 - Borrow
 - Carry out
 - Difference
 - Compliment

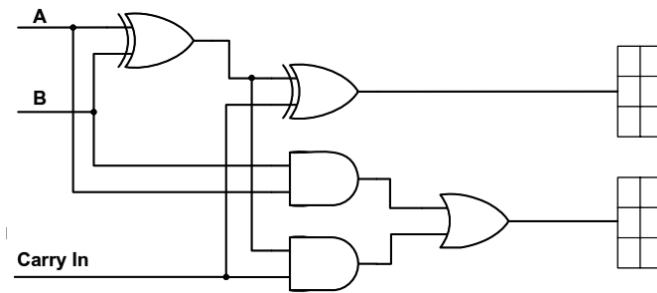


- gambar diatas disebut rangkaian ...
 - Encoder
 - Decoder
 - Half adder
 - Full subtractor
 - Half subtractor



- gambar diatas merupakan rangkaian dari ...
 - Half adder
 - Full adder
 - Full subtractor
 - Encoder

c. Half subtractor



7.

gambar diatas merupakan rangkaian dari ...

- a. Half adder
- b. Full adder
- c. Half subtractor
- d. Full subtractor
- e. Encoder

8. Tabel kebenaran dibawah ini merupakan hasil dari percobaan half adder.

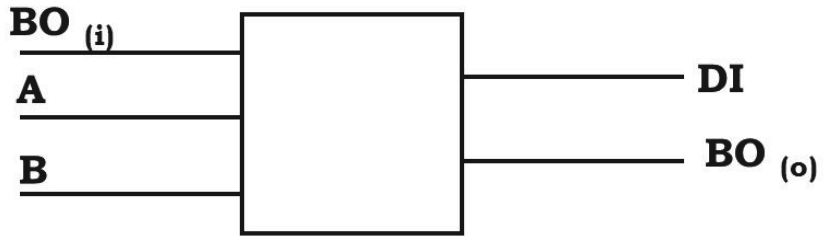
Angka yang tepat untuk mengganti huruf a, b, c dan d adalah ...

INPUT		OUTPUT	
A	B	C	D
0	0	0	a
0	1	0	b
1	0	0	c
1	1	d	0

- a. $a = 1, b = 1, c = 1, d = 0$
- b. $a = 1, b = 1, c = 0, d = 1$
- c. $a = 1, b = 0, c = 1, d = 1$
- d. $a = 0, b = 1, c = 1, d = 1$
- e. $a = 0, b = 0, c = 1, d = 1$

9. Persamaan Boolean untuk rangkaian Full Subtractor adalah ...

- a. $Sum = A \oplus B \oplus Bin$
 $Bin = ((A \oplus B). Bin) + (A.B)$
- b. $Sum = A \oplus B \oplus Bin$
 $Bin = ((A.B) \oplus Bin) + (A.B)$
- c. $Sum = A \oplus B \oplus Bin$
 $Bin = ((A \oplus B). Bin) + (A.B)$
- d. $Sum = A \oplus B \oplus Bin$
 $Bin = (A.B) + ((A \oplus B). Bin)$
- e. $Sum = A \oplus B \oplus Bin$
 $Bin = (\bar{A}.B) + ((A \oplus B). Bin)$



10.

Gambar diatas merupakan simbol blok dari rangkaian ...

- | | |
|--------------------|--------------------|
| a. Half adder | d. Full subtractor |
| b. Full adder | e. Encoder |
| c. Half subtractor | |

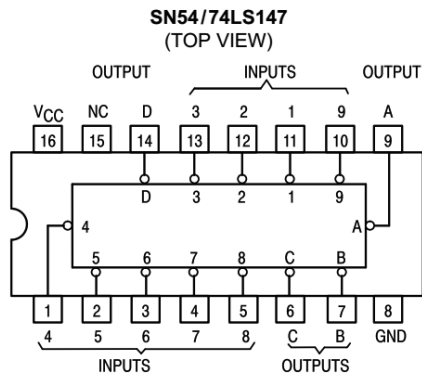
BAB 8.

RANGKAIAN ENKODER DAN DEKODER

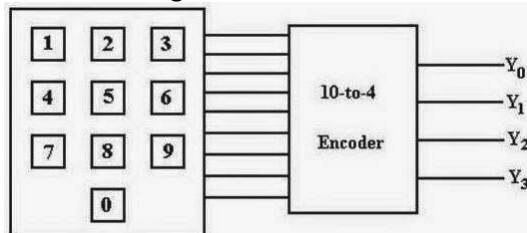
8.1. Enkoder

Pengertian Encoder adalah suatu piranti yang dapat mengubah suatu sistem (bilangan desimal) yang terdapat pada bagian masukan, menjadi system bilangan biner yang terdapat ada bagian keluarannya. Pengertian Encoder juga dapat diartikan sebagai rangkaian yang akan menerjemahkan data pada maksimal 2^n masukan menjadi data pada n keluarannya. Encoder dalam rangkaian digital adalah rangkaian kombinasi gerbang digital yang memiliki input banyak dalam bentuk line input dan memiliki output sedikit dalam format bilangan biner.

Fungsi Encoder adalah untuk rangkaian untuk mengkodekan data input menjadi data bilangan dengan format tertentu. Encoder akan mengkodekan setiap jalur input yang aktif menjadi kode bilangan biner. Dalam teori digital banyak ditemukan istilah encoder seperti “Desimal to BCD Encoder” yang berarti rangkaian digital yang berfungsi untuk mengkodekan line input dengan jumlah line input desimal (0-9) menjadi kode bilangan biner 4 bit BCD (Binary Coded Decimal). Atau “8 line to 3 line encoder” yang berarti rangkaian encoder dengan input 8 line dan output 3 line (3 bit BCD).



Gambar 1. Konfigurasi Kaki IC Encoder 74LS147

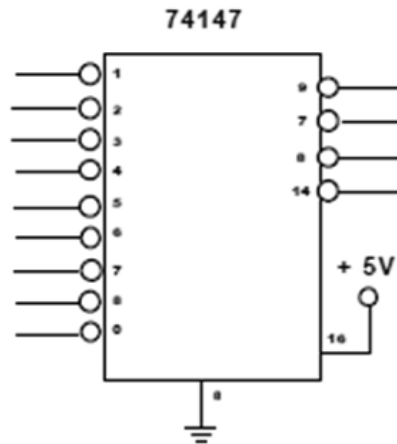


Gambar 2. Ilustrasi Digital Encoder

Binary Encoding

1. Mengkonversi 2^n input dan dikeluarkan kedalam bentuk n bit output
2. Banyak digunakan untuk kompresi data
3. Dapat dibangun menggunakan AND atau OR Gate
4. Jumlah Masukan (Input) > Jumlah Keluaran (Output).

Sirkuit Enkoder



Gambar 3. Sirkuit Encoder

1. Fungsi Sirkuit Encoder.

Fungsi sirkuit encoder adalah mengubah kode desimal menjadi kode - kode biner BCD.

2. Struktur Sirkuit Encoder

Struktur dari sirkuit ini terdiri dari IC Encoder famili TTL 74147 yang memiliki 10 jalur input (desimal) dan mempunyai 4 jalur output (biner BCD). Baik jalur - jalur input maupun jalur - jalur output , semuanya berkondisi active low.

3. Cara Kerja Sirkuit Encoder.

Bila salah satu input berlogika 0 (misalnya input desimal 0 berlogika 0), maka output BCD nya adalah 0000. Informasi lebih lengkap operasi encoder 74147 adalah seperti tabel berikut ini.

Input Desimal										Output BCD			
0	1	2	3	4	5	6	7	8	9	D	C	B	A
0	1	1	1	1	1	1	1	1	1	0	0	0	0
1	0	1	1	1	1	1	1	1	1	0	0	0	1
1	1	0	1	1	1	1	1	1	1	0	0	1	0
1	1	1	0	1	1	1	1	1	1	0	0	1	0
1	1	1	1	0	1	1	1	1	1	0	1	0	0
1	1	1	1	1	0	1	1	1	1	0	1	0	0
1	1	1	1	1	1	0	1	1	1	0	1	1	0
1	1	1	1	1	1	1	0	1	1	0	1	1	1
1	1	1	1	1	1	1	1	0	1	1	0	0	0
1	1	1	1	1	1	1	1	1	0	1	0	0	1

Tabel 1. Operasi Encoder TTL 74147

8.2. Dekoder

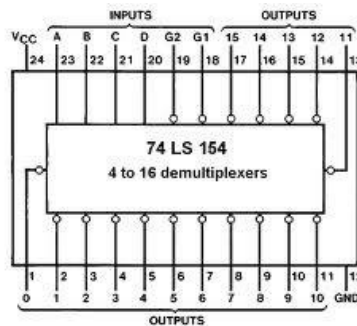
Pengertian *Decoder* adalah alat yang digunakan untuk dapat mengembalikan proses *encoding* sehingga kita dapat melihat atau menerima informasi aslinya. Pengertian *Decoder* juga dapat diartikan sebagai rangkaian

logika yang ditugaskan untuk menerima input - input biner dan mengaktifkan salah satu outputnya sesuai dengan urutan biner tersebut.

Fungsi *Decoder* adalah untuk memudahkan kita dalam menyalakan seven segmen. Itu lah sebabnya kita menggunakan *decoder* agar dapat dengan cepat menyalakan seven segmen. Output dari decoder maksimum adalah $2n$. Jadi dapat kita bentuk n to $2n$ *decoder*. Jika kita ingin merangkaian *decoder* dapat kita buat dengan 3 to 8 *decoder* menggunakan 2 to 4 *decoder*. Sehingga kita dapat membuat 4 to 16 decoder dengan menggunakan dua buah 3 to 8 *decoder*.

Beberapa rangkaian *decoder* yang sering kita jumpai saat ini adalah *decoder* jenis 3 x 8 (3 bit input dan 8 output line), *decoder* jenis 4 x 16, *decoder* jenis BCD to *Decimal* (4 bit input dan 10 output line) dan *decoder* jenis BCD to 7 segmen (4 bit input dan 8 output line). Khusus untuk pengertian *decoder* jenis BCD to 7 segmen mempunyai prinsip kerja yang berbeda dengan *decoder decoder* lainnya, dimana kombinasi setiap inputnya dapat mengaktifkan beberapa output linanya.

Pengertian *decoder* dapat dibentuk dari susunan gerbang logika dasar atau menggunakan IC yang banyak jual di pasaran, seperti decoder IC 74LS48, 74LS155, 74LS139 (decoder 2 masukkan ke 4 keluaran), 74LS138 (decoder 3 masukkan ke 8 keluaran), 74LS154 (decoder 4 masukkan ke 16 keluaran).



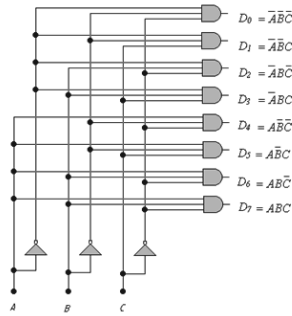
Gambar 4. Konfigurasi kaki IC Dekoder 74LS154

Binary Dekoding

1. Mengkonversi sebuah n bit code kedalam sebuah 1 (satu) output yang aktif (LOW/HIGH).
2. Rangkaiannya dapat dibentuk menggunakan AND atau OR gate.
3. Jumlah Masukan (Input) < Jumlah Keluaran (Output).
4. n input dan $2n$ output.
5. Hanya satu output yang aktif (LOW/HIGH) dari banyak input yang diberikan.

Dekoder Biner ke Oktal

Pada decoder dari biner ke oktal ini terdapat tiga input yaitu A, B dan C yang mewakili suatu bilangan biner tiga bit dan delapan output yang yaitu D0 sampai dengan D7 yang mewakili angka oktal dari 0 sampai dengan 7.

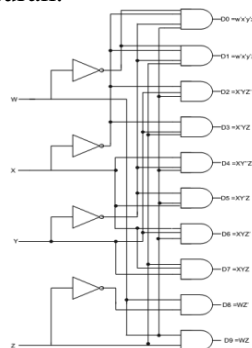


Gambar 5. Delapan Angka Oktal.

Sandi untuk informasi diskrit ini terdiri dari bilangan biner yang diwakili oleh tiga bit. Kerja dekoder ini dapat lebih jelas tampak dari hubungan input dan output yang ditunjukkan pada tabel kebenaran dibawah ini. Tampak bahwa variabel outputnya itu hanya dapat mempunyai sebuah logika 1 untuk setiap kombinasi inputnya. Saluran output yang nilainya sama dengan 1 mewakili angka oktal yang setara dengan bilangan biner pada saluran inputnya.

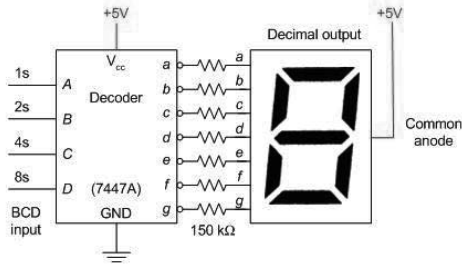
Dekoder BCD ke Desimal

Unsur informasi dalam hal ini adalah sepuluh angka desimal yang diwakili oleh sandi BCD. Masing - masing keluarannya sama dengan 1 hanya bila variabel masukannya membentuk suatu kondisi bit yang sesuai dengan angka desimal yang diwakili oleh sandi BCD itu. Misalnya suatu angka desimal yang disandikan dalam BCD dan tersimpan dalam register empat sel dapat diperagakan dengan pertolongan rangkaian dekoder BCD ke desimal dimana keluaran keempat sel biner tersebut diubah sehingga menyalakan 10 lampu penunjuk. Lampu penunjuk itu dapat berupa angka peraga (display digit), sehingga suatu angka desimal akan menyala bila keluaran dekoder yang sesuai adalah logika 1. Rangkaian dekoder juga berguna untuk menentukan isi register dalam proses pengambilan keputusan. Pemakaiannya yang lain adalah untuk membangkitkan sinyal waktu dan sinyal urutan untuk keperluan pengaturan.

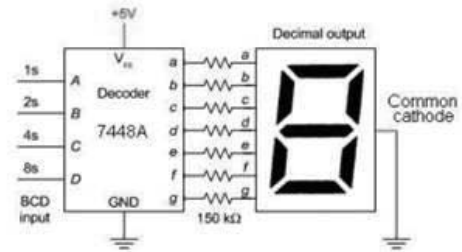


Gambar 6. Dekoder BCD ke decimal

Decoder BCD ini ada 2 macam yaitu outputnya aktif level tinggi dan outputnya aktif rendah, sehingga membutuhkan 7 segmen yang berbeda. Untuk aktif level tinggi menggunakan 7 Segmen Common Katoda, sedangkan untuk aktif level rendah menggunakan 7 Segmen Common Anoda.

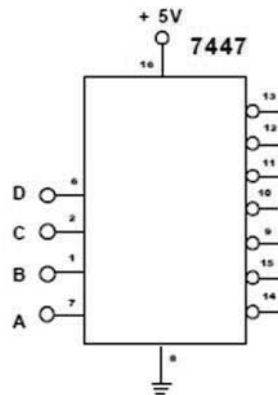


Gambar 7. Rangkaian Decoder BCD to 7 Segmen Common Anoda



Gambar 8. Rangkaian Decoder BCD to 7 Segmen Common Katoda

Sirkuit Dekoder



Gambar 9. Sirkuit Decoder

1) Fungsi Sirkuit Decoder.

Fungsi sirkuit decoder atau BCD to 7 segment decoder adalah sebagai decoder/driver segment. Input BCD akan didecode menjadi output pulsa-pulsa yang akan mendrive 7 segment display.

2) Struktur Sirkuit Decoder

Struktur dari sirkuit ini terdiri dari IC decoder/driver dari keluarga TTL 7447 yang bekerja sebagai driver 7 segment display.

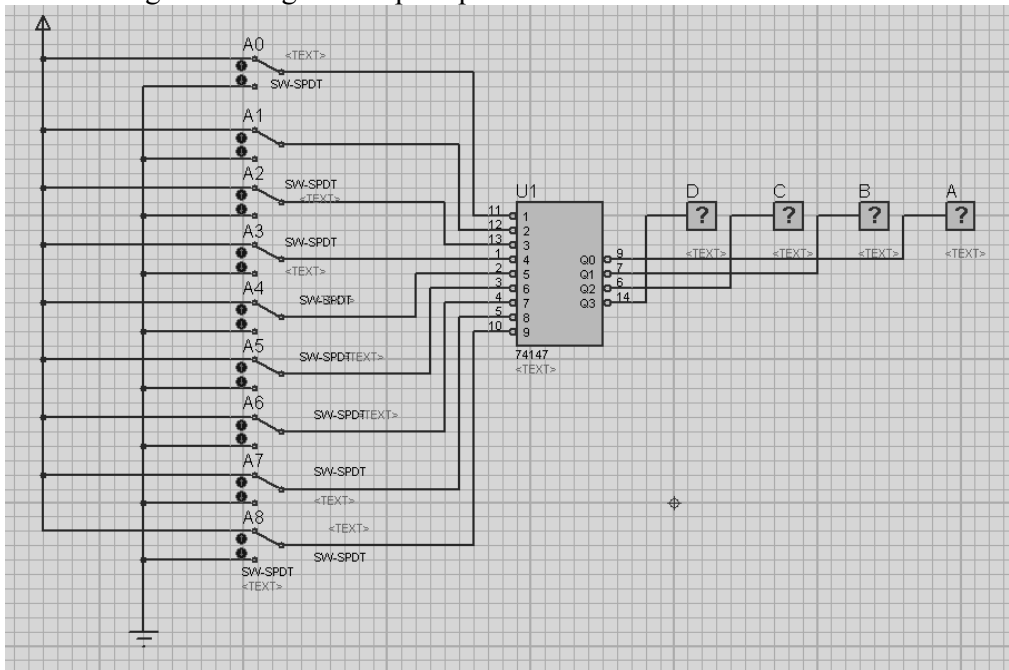
3) Cara Kerja Sirkuit Decoder

Sebagai sirkuit BCD to 7 segment decoder, maka pada setiap perubahan kondisi input DCBA akan didecode menjadi pulsa-pulsa untuk mendrive 7 segment common anoda.

8.3. Simulasi Rangkaian dengan Proteus

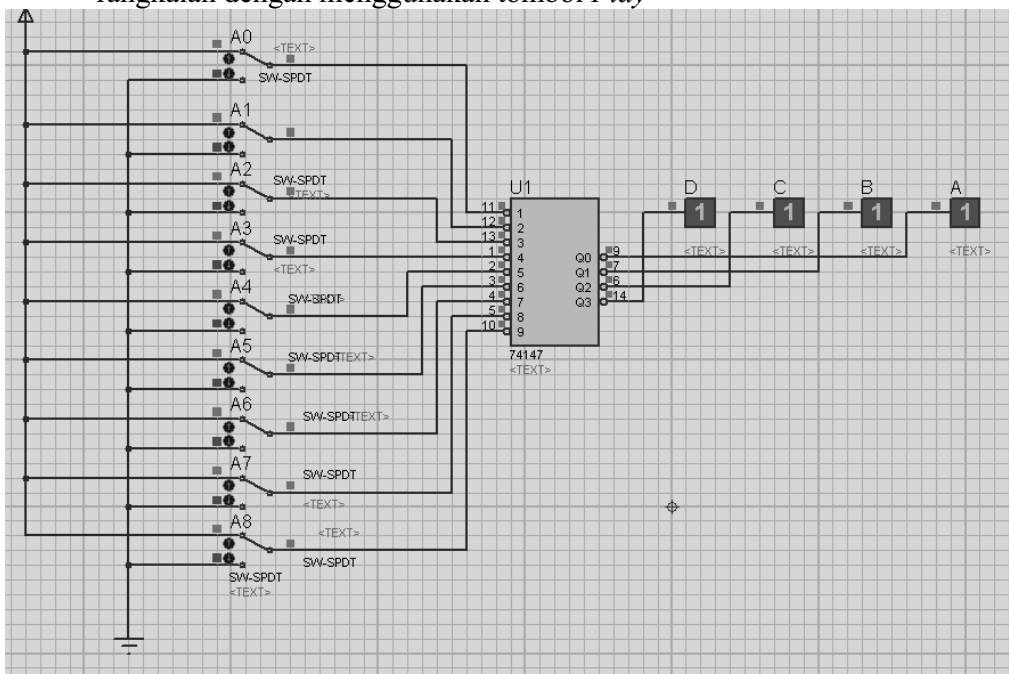
Simulasi Rangkaian Enkoder

1. Masukkan Komponen IC 74147, Logicprobe, sw-spdt
2. Rangkailah rangkaian seperti pada Gambar 8.10



Gambar 10. Rangkaian Enkoder

3. Setelah selesai merangkai rangkaian sesuai pada Gambar 8.10 simulasikan rangkaian dengan menggunakan tombol *Play*



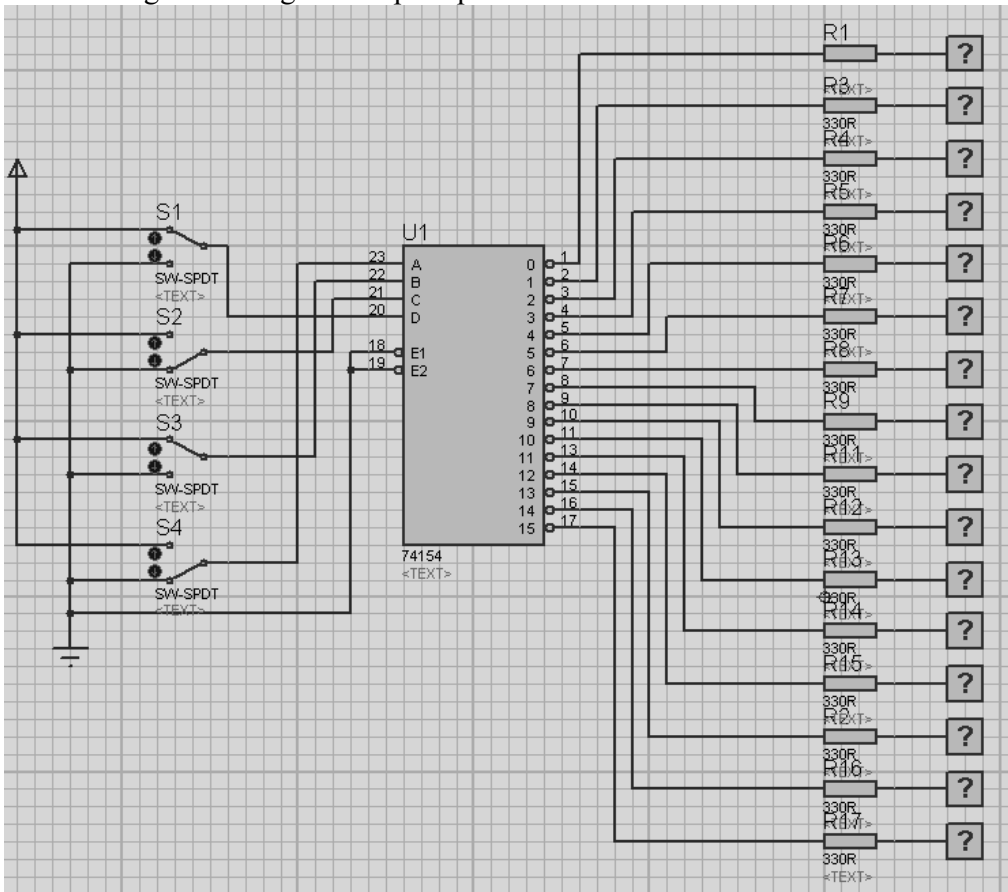
Gambar 11. Simulasi Enkoder

Tabel 8.1.1 Tabel Percobaan Enkoder

Input									Output			
A0	A1	A2	A3	A4	A5	A6	A7	A8	D	C	B	A
1	1	1	1	1	1	1	1	1				
0	1	1	1	1	1	1	1	1				
x	0	1	1	1	1	1	1	1				
x	x	0	1	1	1	1	1	1				
x	x	x	0	1	1	1	1	1				
x	x	x	x	0	1	1	1	1				
x	x	x	x	x	0	1	1	1				
x	x	x	x	x	x	0	1	1				
x	x	x	x	x	x	x	0	1				
x	x	x	x	x	x	x	x	0				

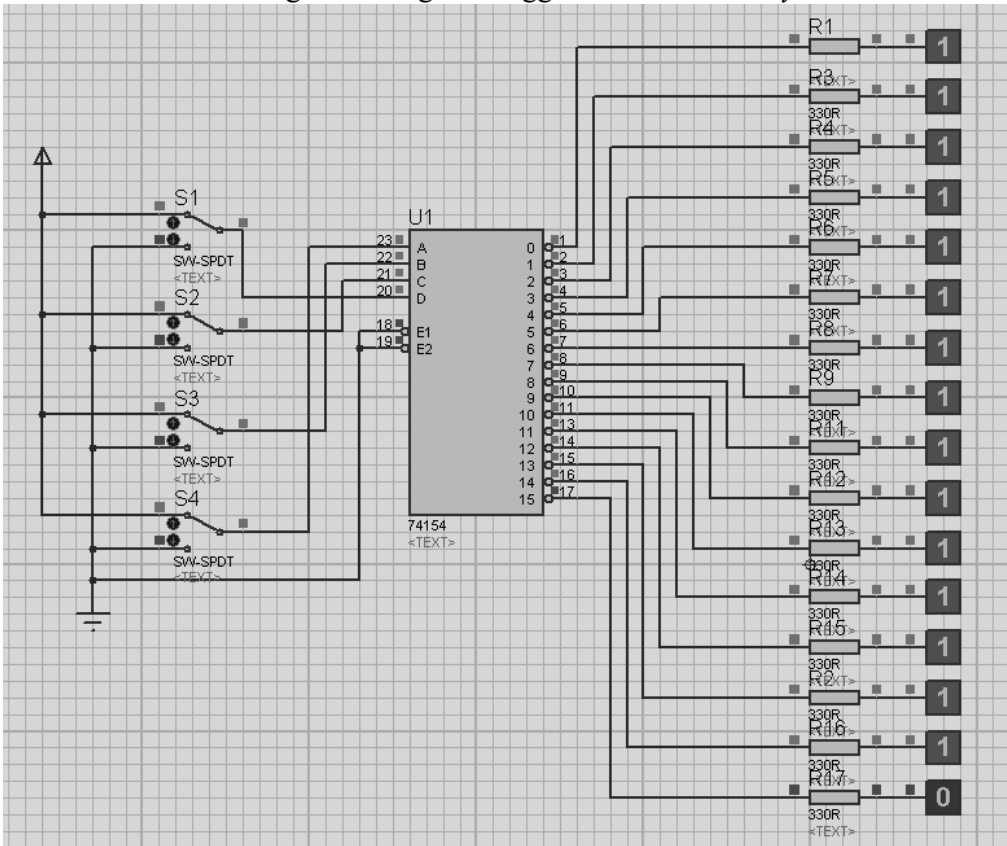
Simulasi Rangkaian Dekoder

1. Masukkan Komponen IC 74154, R 3w330ohm, Logicprobe, sw-spdt
2. Rangkailah rangkaian seperti pada Gambar 8.12.



Gambar 12. Rangkaian Dekoder

- Setelah selesai merangkai rangkaian sesuai pada Gambar 8.2.1 simulasikan rangkaian dengan menggunakan tombol *Play*



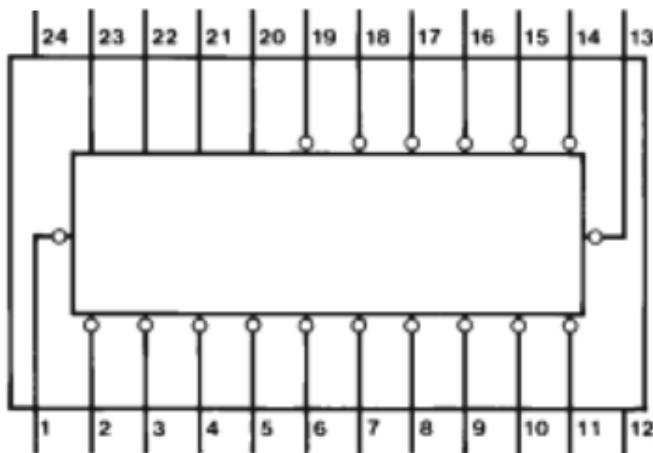
Gambar 13. Simulasi Dekoder

Tabel 8.2.1 Tabel Percobaan Dekoder

Logic Switch				Data Output															
S1	S2	S3	S4	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0	0	0																
0	0	0	1																
0	0	1	0																
0	0	1	1																
0	1	0	0																

8.4. Latihan Soal

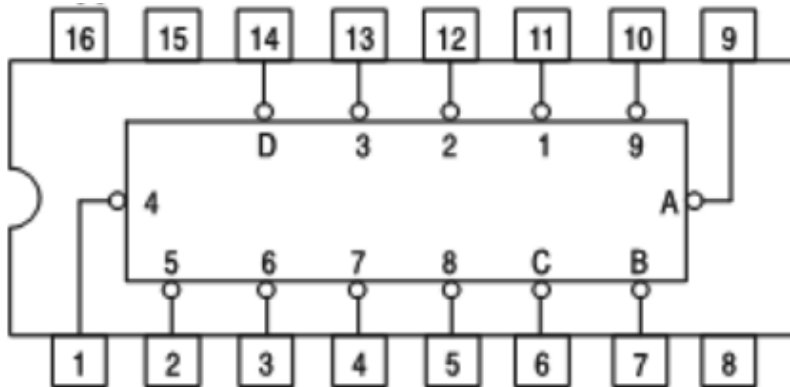
- Rangkaian yang menerjemahkan data pada maksimal 2^n masukan menjadi data pada n keluaran adalah ...
 - Decoder
 - Encoder
 - Multiplexer
 - Demultiplexer
 - Adder
- Rangkaian yang menerjemahkan data pada n masukan menjadi data pada maksimal 2^n keluaran adalah ...
 - Decoder
 - Encoder
 - Multiplexer
 - Demultiplexer
 - Adder
- Jenis IC dekoder yang sering digunakan untuk dekoder 2 masukan dan 4 keluaran adalah ...
 - 74LS138
 - 74LS139
 - 74LS154
 - 74LS164
 - 74LS178
- Jumlah dekoder 3 masukan 8 keluaran yang dibutuhkan untuk membuat dekoder 1 masukan – 32 keluaran adalah ...
 - 1
 - 2
 - 4
 - 8
 - 16
- Jumlah dekoder 1 masukan 16 keluaran yang dibutuhkan untuk decoding bilangan biner 7-bit adalah ...
 - 1
 - 2
 - 4
 - 8
 - 16



6.

Gambar diatas merupakan IC decoder 74154. Yang termasuk kaki input pada IC diatas adalah ...

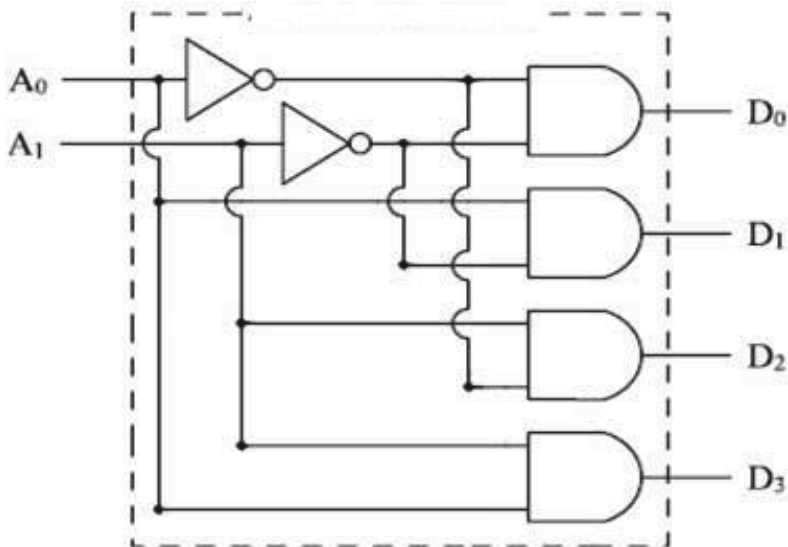
- a. 18 - 23
- b. 18 - 20
- c. 1 - 11
- d. 11 - 12
- e. 1 - 12



7.

Gambar diatas merupakan IC 74147 Yang termasuk kaki input pada IC diatas adalah ...

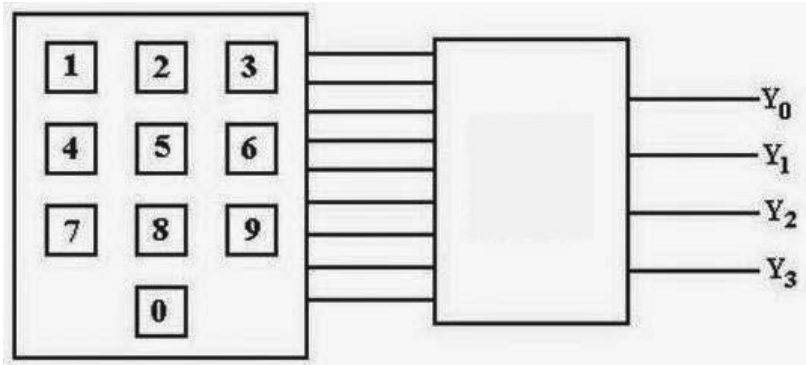
- a. 1 - 4
- b. 1 - 7
- c. 1 - 9
- d. 11 - 12
- e. 1 - 12



8.

Gambar diatas merupakan rangkaian dari ...

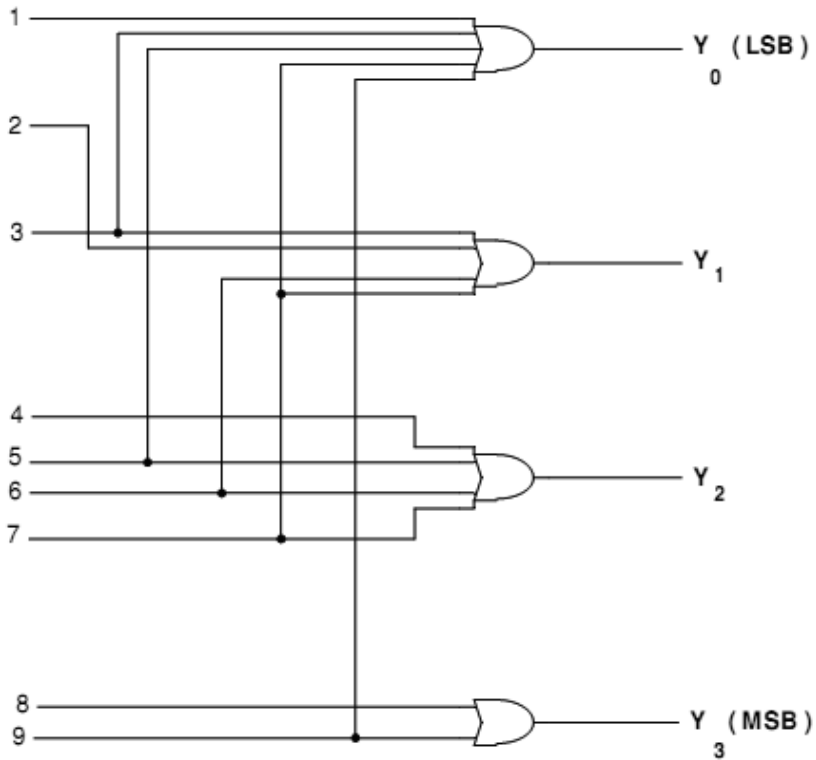
- a. decoder
- b. encoder
- c. multiplexer
- d. demultiplexer
- e. adder



9.

Gambar diatas merupakan ilustrasi digital dari ...

- a. decoder
- b. encoder
- c. multiplexer
- d. demultiplexer
- e. adder



10.

Gambar diatas merupakan logic diagram dari ...

- a. BCD Encoder
- b. Decimal to BCD Encoder
- c. BCD Decoder
- d. Decimal to BCD Decoder
- e. Shift Register

BAB 9.

RANGKAIAN ADC DAN DAC

9.1. ADC

ADC (*Analog to Digital Converter*) adalah perangkat elektronika yang berfungsi untuk mengubah sinyal analog (sinyal kontinu) menjadi sinyal digital. Perangkat ADC (*Analog To Digital Conversion*) dapat berbentuk suatu modul atau rangkaian elektronika maupun suatu chip IC. ADC (*Analog To Digital Converter*) berfungsi untuk menjembatani pemrosesan sinyal analog oleh sistem digital.

Converter

Alat bantu digital yang paling penting untuk teknologi kontrol proses adalah yang menerjemahkan informasi digital ke bentuk analog dan juga sebaliknya. Sebagian besar pengukuran variabel-variabel dinamik dilakukan oleh piranti ini yang menerjemahkan informasi mengenai variabel ke bentuk sinyal listrik analog. Untuk menghubungkan sinyal ini dengan sebuah komputer atau rangkaian logika digital, sangat perlu untuk terlebih dahulu melakukan konversi analog ke digital (A/D). Hal-hal mengenai konversi ini harus diketahui sehingga ada keunikan, hubungan khusus antara sinyal analog dan digital.

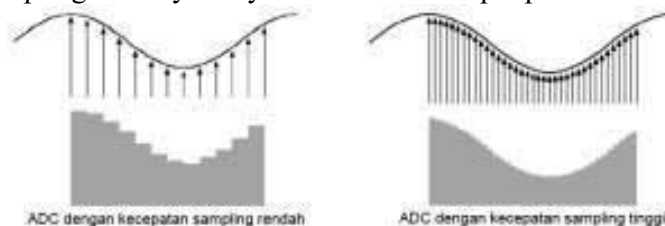
ADC (Analog to Digital Conversion)

Analog To Digital Converter (ADC) adalah pengubah input analog menjadi kode – kode digital. ADC banyak digunakan sebagai Pengatur proses industri, komunikasi digital dan rangkaian pengukuran/ pengujian. Umumnya ADC digunakan sebagai perantara antara sensor yang kebanyakan analog dengan sistim komputer seperti sensor suhu, cahaya, tekanan/ berat, aliran dan sebagainya kemudian diukur dengan menggunakan sistim digital (komputer).

ADC (*Analog to Digital Converter*) memiliki 2 karakter prinsip, yaitu kecepatan sampling dan resolusi.

Kecepatan Sampling ADC

Kecepatan sampling suatu ADC menyatakan “seberapa sering sinyal analog dikonversikan ke bentuk sinyal digital pada selang waktu tertentu”. Kecepatan sampling biasanya dinyatakan dalam sample per second (SPS).



Gambar 1. Ilustrasi Kecepatan Sampling ADC

Resolusi ADC

Resolusi ADC menentukan “ketelitian nilai hasil konversi ADC”. Sebagai contoh: ADC 8 bit akan memiliki output 8 bit data digital, ini berarti sinyal input dapat dinyatakan dalam 255 ($2^n - 1$) nilai diskrit. ADC 12 bit memiliki 12 bit output data digital, ini berarti sinyal input dapat dinyatakan dalam 4096 nilai diskrit. Dari contoh diatas ADC 12 bit akan memberikan ketelitian nilai hasil konversi yang jauh lebih baik daripada ADC 8 bit.

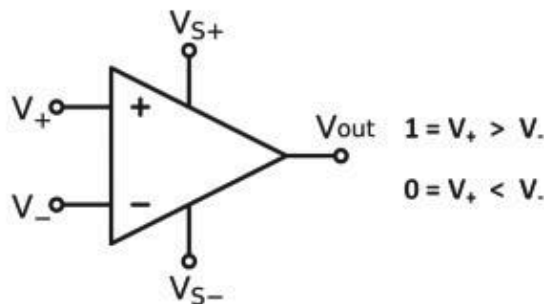
Prinsip Kerja ADC

Prinsip kerja ADC adalah mengkonversi sinyal analog ke dalam bentuk besaran yang merupakan rasio perbandingan sinyal input dan tegangan referensi. Sebagai contoh, bila tegangan referensi 5 volt, tegangan input 3 volt, rasio input terhadap referensi adalah 60%. Jadi, jika menggunakan ADC 8 bit dengan skala maksimum 255, akan didapatkan sinyal digital sebesar $60\% \times 255 = 153$ (bentuk decimal) atau 10011001 (bentuk biner).

$$\begin{aligned} \text{signal} &= (\text{sample}/\text{max_value}) * \text{reference_voltage} \\ &= (153/255) * 5 \\ &= 3 \text{ Volts} \end{aligned}$$

Komparator ADC

Bentuk komunikasi yang paling mendasar antara wujud digital dan analog adalah piranti (biasanya berupa IC) disebut komparator. Piranti ini, yang diperlihatkan secara skematik pada gambar dibawah, secara sederhana membandingkan dua tegangan pada kedua terminal inputnya. Bergantung pada tegangan mana yang lebih besar, outputnya akan berupa sinyal digital 1 (high) atau 0 (low). Komparator ini digunakan secara luas untuk sinyal alarm ke komputer atau sistem pemroses digital. Elemen ini juga merupakan satu bagian dengan konverter analog ke digital dan digital ke analog yang akan didiskusikan nanti.



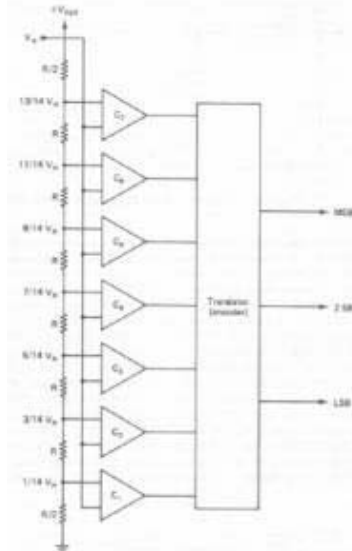
Gambar 2. Konsep Komparator Pada ADC (*Analog to Digital Converter*)

Gambar diatas memperlihatkan sebuah komparator merubah keadaan logika output sesuai fungsi tegangan input analog. Sebuah komparator dapat tersusun dari sebuah opamp yang memberikan output terpotong untuk menghasilkan level yang diinginkan untuk kondisi logika (+5 dan 0 untuk TTL 1 dan 0). Komparator komersil didesain untuk memiliki level logika yang diperlukan pada bagian outputnya.

Jenis-Jenis ADC (Analog to Digital Converter)

ADC Simultan

ADC Simultan atau biasa disebut *flash converter* atau *parallel converter*. Input analog V_i yang akan diubah ke bentuk digital diberikan secara simultan pada sisi + pada komparator tersebut, dan input pada sisi - tergantung pada ukuran bit *converter*. Ketika V_i melebihi tegangan input - dari suatu komparator, maka output komparator adalah high, sebaliknya akan memberikan output low.



Gambar 3. ADC Simultan

Bila V_{ref} diset pada nilai 5 Volt, maka dari gambar 3 dapat didapatkan :

$V(-)$ untuk C_7 =	$V_{ref} * (13/14) =$	4,64
$V(-)$ untuk C_6 =	$V_{ref} * (11/14) =$	3,93
$V(-)$ untuk C_5 =	$V_{ref} * (9/14) =$	3,21
$V(-)$ untuk C_4 =	$V_{ref} * (7/14) =$	2,5
$V(-)$ untuk C_3 =	$V_{ref} * (5/14) =$	1,78
$V(-)$ untuk C_2 =	$V_{ref} * (3/14) =$	1,07
$V(-)$ untuk C_1 =	$V_{ref} * (1/14) =$	0,36

Misal :

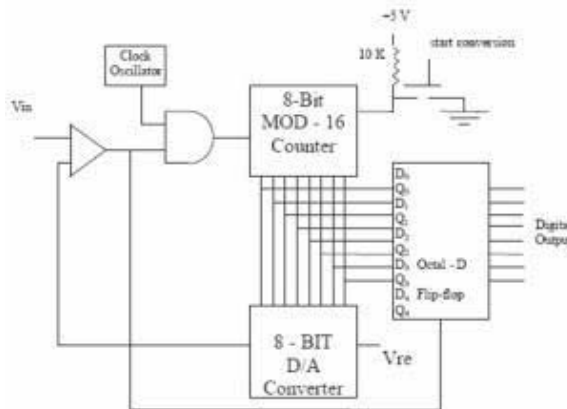
V_{in} diberi sinyal analog 3 Volt, maka output dari $C_7=0$, $C_6=0$, $C_5=0$, $C_4=1$, $C_3=1$, $C_2=1$, $C_1=1$, sehingga didapatkan output ADC yaitu 100 biner

Output Comparator							Output Translator		
C7	C6	C5	C4	C3	C2	C1	2 ²	2 ¹	2 ⁰
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	1
0	0	0	0	0	1	1	0	1	0
0	0	0	0	1	1	1	0	1	1
0	0	0	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	0	0
1	1	1	1	1	1	1	1	0	1

Tabel Output ADC Simultan

Ada beberapa konsep dasar dari ADC adalah dengan cara Counter Ramp ADC, Successive Aproximation ADC dan lain sebagainya.

Counter Ramp ADC



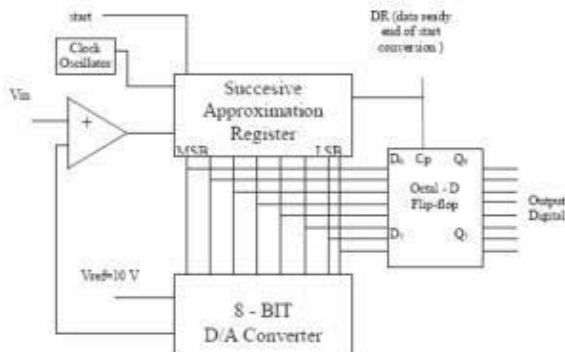
Gambar 4. Blok Diagram Counter Ramp ADC

Pada gambar diatas, ditunjukkan blok diagram Counter Ramp ADC didalamnya terdapat DAC yang diberi masukan dari counter, masukan counter dari sumber Clock dimana sumber Clock dikontrol dengan cara meng AND kan dengan keluaran Comparator. Comparator membandingkan antara tegangan masukan analog dengan tegangan keluaran DAC, apabila tegangan masukan yang akan dikonversi belum sama dengan tegangan keluaran dari DAC maka keluaran comparator = 1 sehingga Clock dapat memberi masukan counter dan hitungan counter naik.

Misal akan dikonversi tegangan analog 2 volt, dengan mengasumsikan counter reset, sehingga keluaran pada DAC juga 0 volt. Apabila konversi dimulai maka counter akan naik dari 0000 ke 0001 karena mendapatkan pulsa masuk dari Clock oscillator dimana saat itu keluaran Comparator = 1, karena mendapatkan kombinasi biner dari counter 0001 maka tegangan keluaran DAC naik dan dibandingkan lagi dengan tegangan masukan demikian seterusnya nilai counter naik dan keluaran tegangan DAC juga naik hingga suatu saat tegangan masukan dan tegangan keluaran DAC sama yang mengakibatkan keluaran komparator = 0 dan Clock tidak dapat masuk. Nilai counter saat itulah yang merupakan hasil konversi dari analog yang dimasukkan.

Kelemahan dari counter tersebut adalah lama, karena harus melakukan trace mulai dari 0000 hingga mencapai tegangan yang sama sehingga butuh waktu.

SAR (*Successive Aproximation Register*) ADC

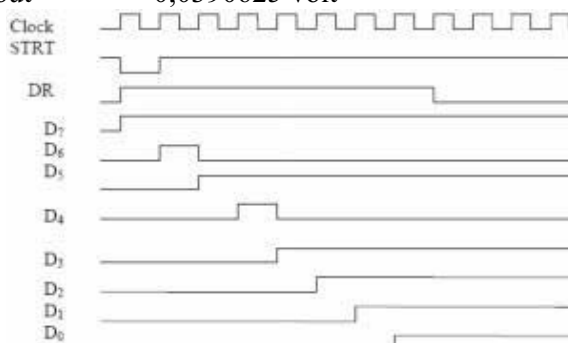


Gambar 5. Blok Diagram SAR ADC

Pada gambar diatas ditunjukkan diagram ADC jenis SAR, Yaitu dengan memakai konvfigurasi yang hampir sama dengan counter ramp tetapi dalam melakukan trace dengan cara tracking dengan mengeluarkan kombinasi bit MSB = 1 =====> 1000 0000. Apabila belum sama (kurang dari tegangan analog input maka bit MSB berikutnya = 1 =====>1100 0000) dan apabila tegangan analog input ternyata lebih kecil dari tegangan yang dihasilkan DAC maka langkah berikutnya menurunkan kombinasi bit =====> 10100000.

Untuk mempermudah pengertian dari metode ini diberikan contoh seperti pada timing diagram gambar 6 Misal diberi tegangan analog input sebesar 6,84 volt dan tegangan referensi ADC 10 volt sehingga apabila keluaran tegangan sbb :

- Jika D7 = 1 Vout = 5 volt
- Jika D6 = 1 Vout = 2,5 volt
- Jika D5 = 1 Vout = 1,25 volt
- Jika D4 = 1 Vout = 0,625 volt
- Jika D3 = 1 Vout = 0,3125 volt
- Jika D2 = 1 Vout = 0,1625 volt
- Jika D1 = 1 Vout = 0,078125 volt
- Jika D0 = 1 Vout = 0,0390625 volt



Gambar 6. Timing diagram urutan Trace SAR ADC

Setelah diberikan sinyal start maka konversi dimulai dengan memberikan kombinasi 1000 0000 ternyata menghasilkan tegangan 5 volt dimana masih kurang dari tegangan input 6,84 volt, kombinasi berubah menjadi 1100 0000 sehingga $V_{out} = 7,5$ volt dan ternyata lebih besar dari 6,84 sehingga kombinasi menjadi 1010 0000 tegangan $V_{out} = 6,25$ volt kombinasi naik lagi 1011 0000 demikian seterusnya hingga mencapai tegangan 6,8359 volt dan membutuhkan hanya 8 clock.

9.2. DAC

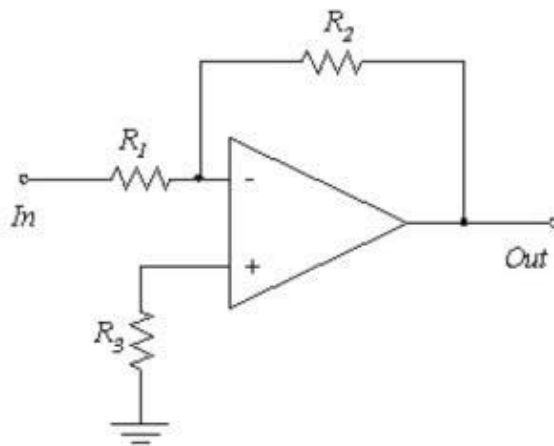
DAC (*Digital To Analog Converter*) adalah perangkat elektronika yang berfungsi untuk mengubah sinyal digital (diskrit) menjadi sinyal analog (kontinyu). Aplikasi DAC (*Digital To Analog Converter*) adalah sebagai antarmuka (interface) antara perangkat yang bekerja dengan sistem digital dan perangkat pemroses sinyal analog. Perangkat DAC (*Digital To Analog Converter*) dapat berupa rangkaian elektronika dan chip IC DAC.

Konsep Dasar DAC (*Digital To Analog Converter*)

Pada dasarnya rangkaian penjumlah op-amp (summing amplifier) dapat digunakan untuk menyusun suatu konverter D/A (DAC “*Digital To Analog Converter*”) dengan memakai sejumlah hambatan masukan yang diberi bobot dalam deret biner.

Penguat Inverting

Rangkaian untuk penguat inverting adalah seperti yang ditunjukkan gambar dibawah. Penguat ini memiliki ciri khusus yaitu sinyal keluaran memiliki beda fasa sebesar 180° .

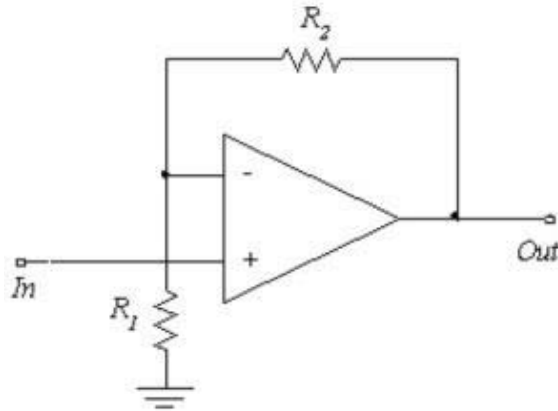


Gambar 7. Penguatan rangkaian penguat inverting adalah berdasar pada persamaan berikut :

$$V_{out} = -V_{in}(R_2/R_1)$$

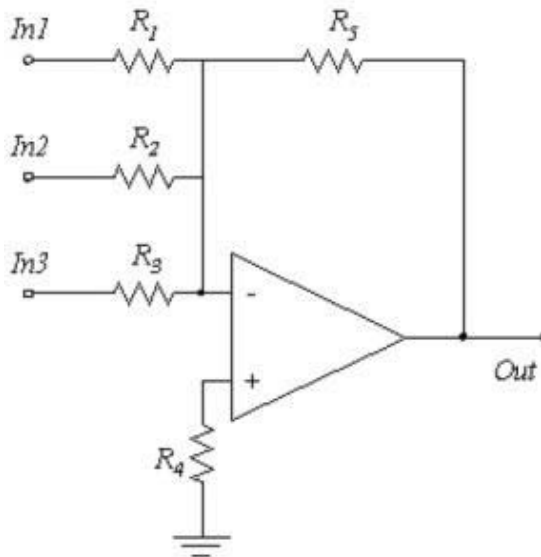
Penguat Non-Inverting

Penguat non-inverting memiliki ciri khusus yaitu sinyal output adalah sefasa dengan sinyal masukan. Rangkaian ini ditunjukkan oleh gambar berikut.



Gambar 8. Penguatan dari rangkaian penguat jenis ini adalah berdasar pada persamaan berikut :

$$V_{out} = V_{in} \left(\frac{R_1 + R_2}{R_1} \right)$$



Gambar 9. Penguat Penjumlah (Dasar DAC)

Penguat penjumlah memiliki ciri khusus yaitu sinyal keluaran merupakan hasil penguatan dari penjumlahan sinyal masukannya. Pada bagian ini dicontohkan penguat penjumlah berdasarkan rangkaian penguat inverting. Sehingga sinyal keluaran adalah berbeda fasa sebesar 180o. Rangkaian penguat penjumlah merupakan konsep dasar dari rangkaian DAC (Digital To Analog Converter).

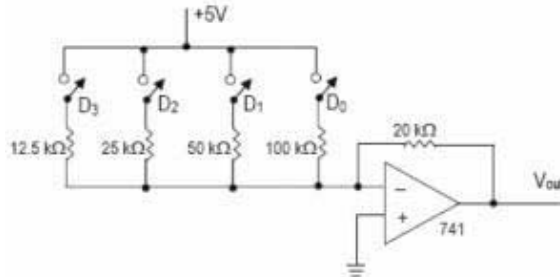
Penguatan dari rangkaian ini dihitung menggunakan persamaan berikut :

$$V_{out} = (-V_{in1} \frac{R_5}{R_1}) + (-V_{in2} \frac{R_5}{R_2}) + (-V_{in3} \frac{R_5}{R_3})$$

Jenis-Jenis DAC (*Digital To Analog Converter*)

Binary-Weighted DAC (*Digital To Analog Converter*)

Suatu rangkaian Binary-weighted DAC dapat disusun dari beberapa Resistor dan Operational Amplifier (Op-Amp) seperti gambar berikut.



Gambar 10. Rangkaian Binary Weighted DAC

Secara prinsip rangkaian DAC diatas dapat dijelaskan sebagai berikut. Resistor 20 kΩ menjumlahkan arus yang dihasilkan dari penutupan switch-switch D0 sampai D3. Resistor-resistor ini diberi skala nilai sedemikian rupa sehingga memenuhi bobot biner (binary-weighted) dari arus yang selanjutnya akan dijumlahkan oleh resistor 20 kΩ. Dengan menutup D0 menyebabkan arus 50 μ A mengalir melalui resistor 20 kΩ, menghasilkan tegangan -1 V pada Vout. Penutupan masing-masing switch menyebabkan penggandaan nilai arus yang dihasilkan dari switch sebelumnya. Nilai konversi dari kombinasi penutupan switch ditunjukkan pada tabel berikut.

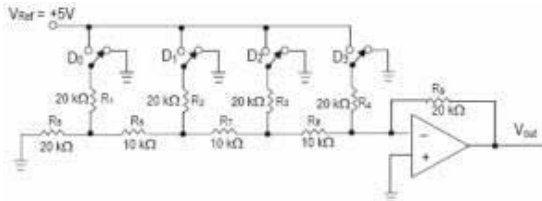
Tabel Output Binary-weighted DAC

Konversi dari nilai digital ke nilai analog berdasarkan rangkaian Binary Weighted DAC diatas

D3	D2	D1	D0	Vout
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

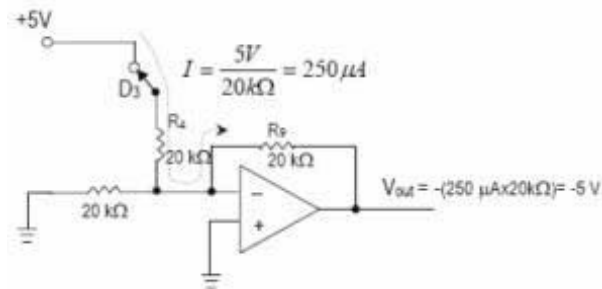
R/2R Ladder DAC (Digital To Analog Converter)

Metode lain dari konversi *Digital to Analog* adalah R/2R Ladder. Metode ini banyak digunakan dalam IC-IC DAC. Pada rangkaian R/2R Ladder, hanya dua nilai resistor yang diperlukan, yang dapat diaplikasikan untuk IC DAC dengan resolusi 8,10 atau 12 bit. Rangkaian R/2R Ladder ditunjukkan pada gambar berikut.



Gambar 11. Rangkaian R/2R Ladder DAC

Prinsip kerja dari rangkaian R/2R Ladder DAC adalah sebagai berikut : informasi digital 4 bit masuk ke switch D0 sampai D3. Switch ini mempunyai kondisi “1” (sekitar 5 V) atau “0” (sekitar 0 V). Dengan pengaturan switch akan menyebabkan perubahan arus yang mengalir melalui R9 sesuai dengan nilai ekivalen biner-nya Sebagai contoh, jika D0 = 0, D1 = 0, D2 = 0 dan D3 = 1, maka R1 akan paralel dengan R5 menghasilkan 10 k . Selanjutnya 10 k ini seri dengan R6 = 10 k menghasilkan 20 k . 20 k ini paralel dengan R2 menghasilkan 10 k , dan seterusnya sampai R7, R3 dan R8. Rangkaian ekivalennya ditunjukkan pada gambar 6. Vout yang dihasilkan dari kombinasi switch ini adalah -5V.



Gambar 12. Rangkaian Ekivalen R/2R Ladder DAC

Untuk mendapatkan Vout analog dari rangkaian R/2R Ladder DAC diatas dapat dihitung dengan menggunakan persamaan :

$$V_{out} = (-V_{ref}(R_9/R)) * ((D_0/16) + (D_1/8) + (D_2/4) + (D_3/2))$$

Tabel Output Rangkaian R/2R Ladder DAC

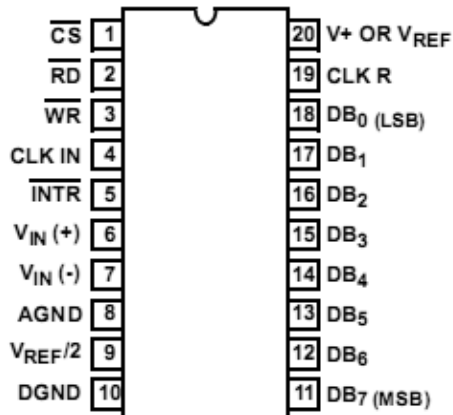
Nilai kombinasi dan hasil konversi rangkaian R/2R Ladder DAC ditunjukkan pada tabel dibawah.

D3	D2	D1	D0	Vout
0	0	0	0	0.00
0	0	0	1	0.625
0	0	1	0	1.250
0	0	1	1	1.875
0	1	0	0	2.500
0	1	0	1	3.125
0	1	1	0	3.750
0	1	1	1	4.375
1	0	0	0	5.000
1	0	0	1	5.625
1	0	1	0	6.250
1	0	1	1	6.875
1	1	0	0	7.500
1	1	0	1	8.125
1	1	1	0	8.750
1	1	1	1	9.375

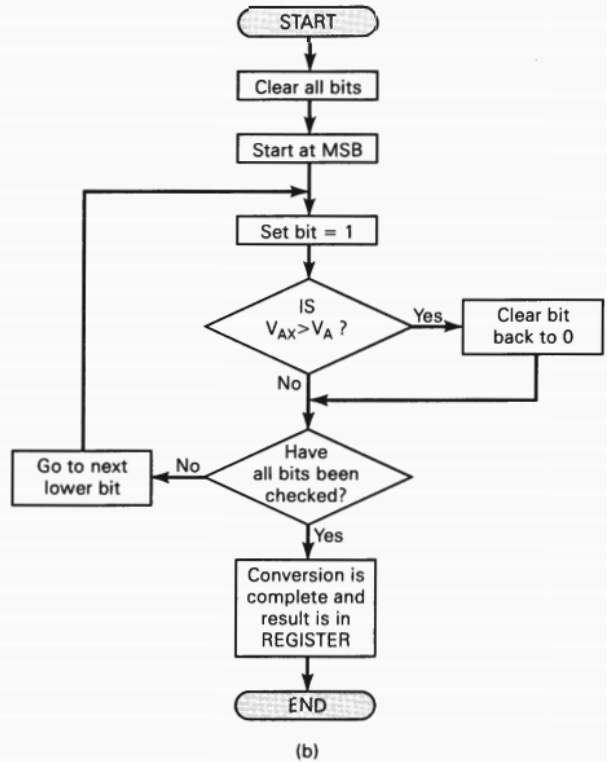
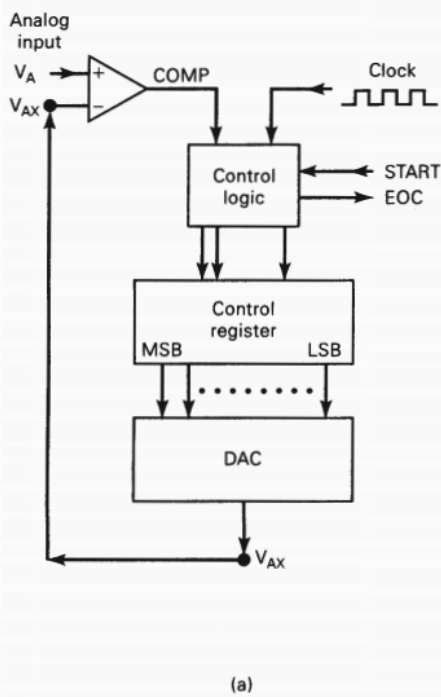
Tabel diatas merupakan hasil konversi dari nilai digital ke nilai analog berdasarkan rangkaian R/2R Ladder DAC (*Digital To Analog Converter*).

Rangkaian ADC ada yang sudah dikemas dalam satu chip IC, salah satu contohnya adalah ADC0804 yang sudah tersedia di trainer digital INEX. ADC0804 adalah ADC jenis CMOS 8 bit succesive approximation. Diagram koneksi kaki-kaki IC ADC0804

DAC Succesive-approxination (a) Diagram blok yang disederhanakan (b)



Gambar 13. Diagram alir cara kerja

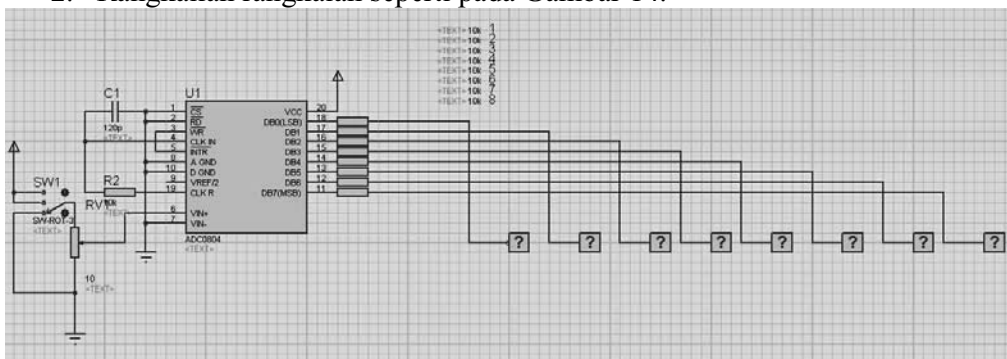


Contoh operasi DAC Successive-approximation dengan step size 1 V, $V_A=10,4$ V, dan keluaran digital 10112 = 1110

9.3. Simulasi dengan Proteus

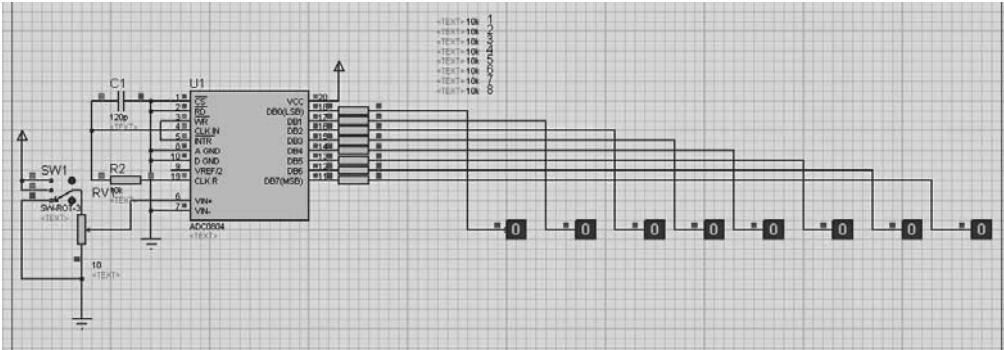
Simulasi Rangkaian ADC

1. Masukkan Komponen IC ADC0804, R 10k, Logicprobe dan komponen lain yang diperlukan pada Gambar 9.1.1
2. Rangkailah rangkaian seperti pada Gambar 14.



Gambar 14. Rangkaian ADC

3. Setelah selesai merangkai rangkaian sesuai pada Gambar 14 simulasikan rangkaian dengan menggunakan tombol *Play*



Gambar 15. Simulasi Rangkaian ADC

Tabel Percobaan ADC

No	Vin	Logic Monitor								Nilai Desimal	Step Size
		D7	D6	D5	D4	D3	D2	D1	D0		
1	0,5										
2	1										
3	1,5										
4	2										
5	2,5										
6	3										
7	3,5										
8	4										
9	4,5										
10	5										

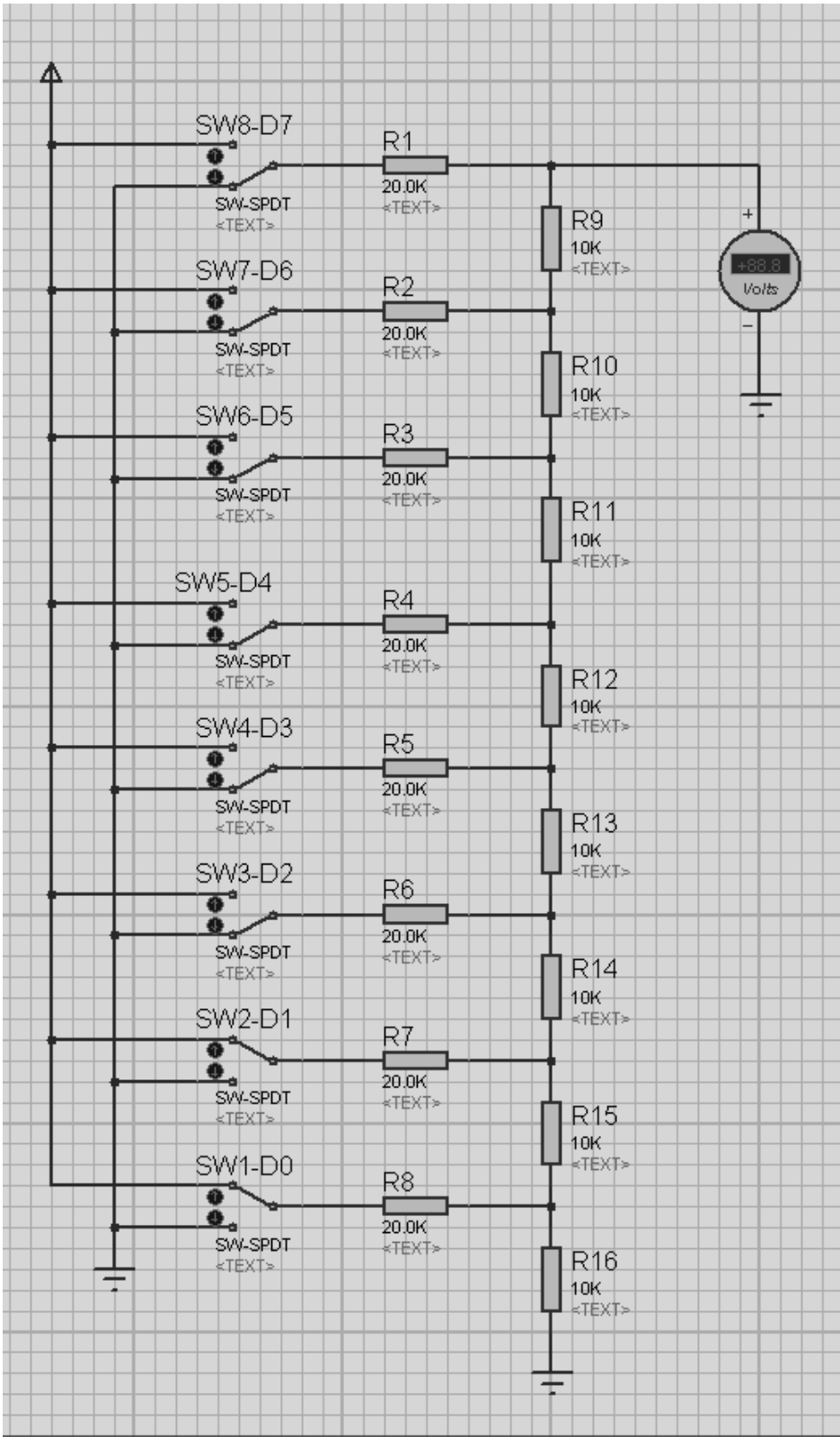
Catatan

Step Size dihitung dengan rumus sebagai berikut

Step size = Nilai Analog / Nilai Digital

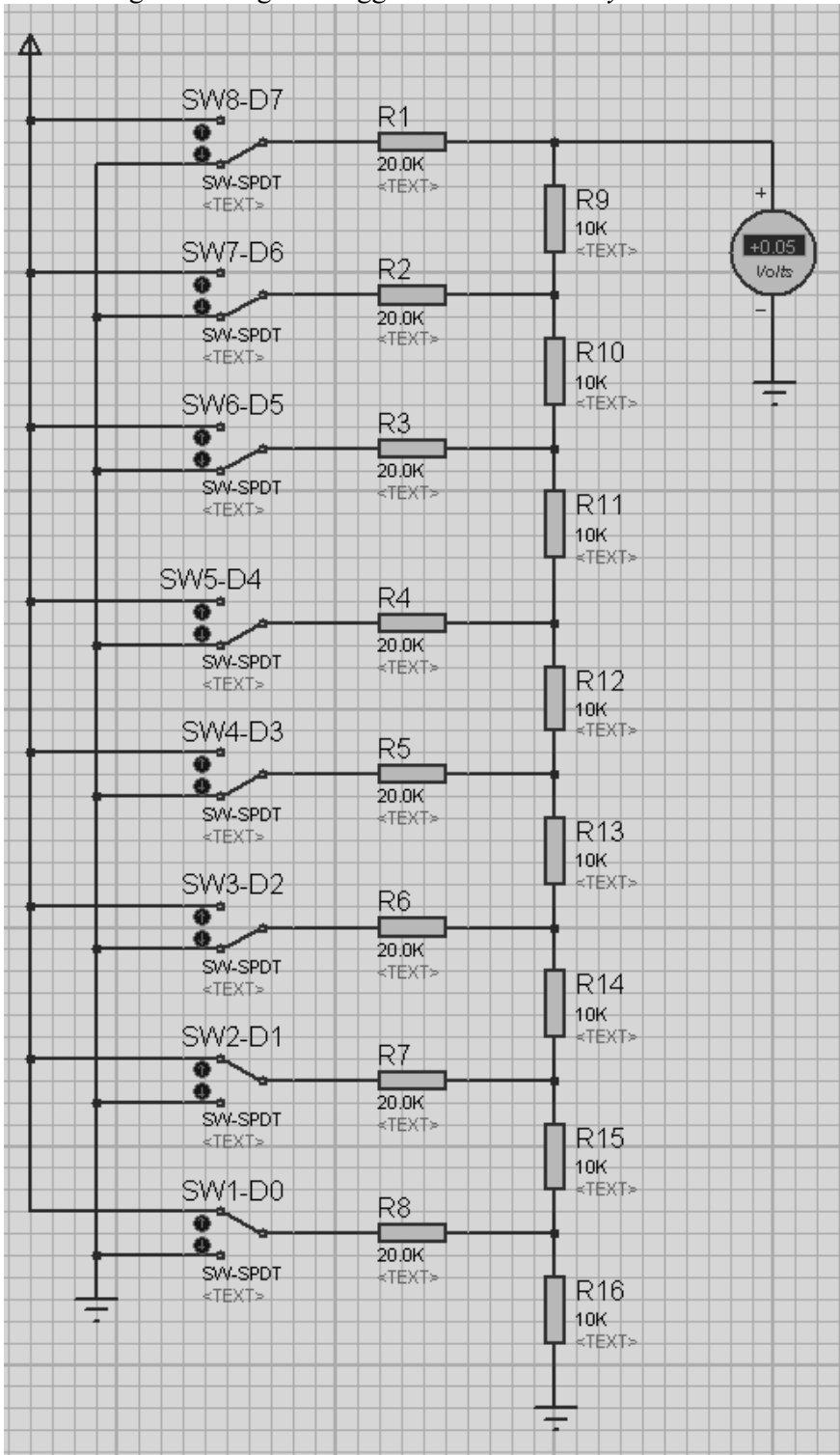
Simulasi Rangkaian DAC

1. Masukkan Komponen yang diperlukan sesuai pada Gambar 16.
2. Rangkailah rangkaian seperti pada Gambar 16.



Gambar 16. Rangkaian DAC

3. Setelah selesai merangkai rangkaian sesuai pada Gambar 16 simulasikan rangkaian dengan menggunakan tombol *Play*



Gambar 17. Simulasi DAC

Tabel Percobaan DAC

No	Nilai Desimal									Vout	Step Size
		D7	D6	D5	D4	D3	D2	D1	D0		
1	0	0	0	0	0	0	0	0	0		
2	1	0	0	0	0	0	0	0	1		
3	2	0	0	0	0	0	0	1	0		
4	3	0	0	0	0	0	0	1	0		
5	4	0	0	0	0	0	1	0	0		
6		0	0	0	0	0	1	0	1		
7		0	0	0	0	1	0	0	0		
8		0	0	0	1	0	0	0	0		
9		0	0	1	0	0	0	0	0		
10		0	0	1	1	0	0	0	0		
11		0	1	0	0	0	0	0	0		
12		1	0	0	0	0	0	0	0		
13		1	0	0	1	0	0	0	0		
14		1	0	1	0	0	0	0	0		
15		1	0	1	1	0	0	0	0		
16		1	1	1	1	1	1	1	1		

9.4. Latihan Soal

1. ASCII merupakan singkatan dari ...
 - a. American Serial Communication Interface
 - b. Additive Signal Coupling Interface
 - c. American Standard Code for Information Interchange
 - d. American Signal Coupling Interface
 - e. Semua jawaban salah
2. Sebuah sistem konversi data dapat digunakan pada interface sistem komputer digital pada ...
 - a. Output analog device
 - b. Digital output device
 - c. Analog input device
 - d. Digital input device
 - e. Printer device
3. Kelemahan utama flash ADC adalah ...
 - a. Tegangan input dibutuhkan secara terus menerus
 - b. Dibutuhkan waktu yang lama untuk konversi
 - c. Membutuhkan nilai komparator yang besar untuk menghasilkan bilangan biner yang wajar (masuk akal)
 - d. Membutuhkan nilai input line untuk men-decode tegangan input
 - e. Membutuhkan nilai output line untuk men-decode tegangan input
4. Keuntungan utama dari R/2R ladder DAC dibandingkan binary-weighted DAC adalah ...
 - a. Hanya membutuhkan dua nilai resistor yang berbeda
 - b. Memiliki jumlah input yang lebih sedikit
 - c. Operasi lebih mudah untuk dianalisis
 - d. Virtual ground dihilangkan
 - e. Sirkuit lebih mudah untuk dipahami
5. Pada flash ADC output dari setiap komparator dihubungkan pada ...
 - a. Decoder
 - b. Priority encoder
 - c. Multiplexer
 - d. Demultiplexer
 - e. Encoder
6. Dibawah ini yang bukan merupakan error pada ADC adalah ...
 - a. Differential nonlinearity
 - b. Missing code
 - c. Incorrect code
 - d. Offset
 - e. Semua jawaban benar
7. Sebuah DAC 4-bit memiliki tegangan K bernilai 1, dan digital input sebesar 12 volt, dan nilai max flow sebesar 10v. Maka besar nilai output analog adalah ...
 - a. 12
 - b. 20
 - c. 30
 - d. 48
 - e. 120

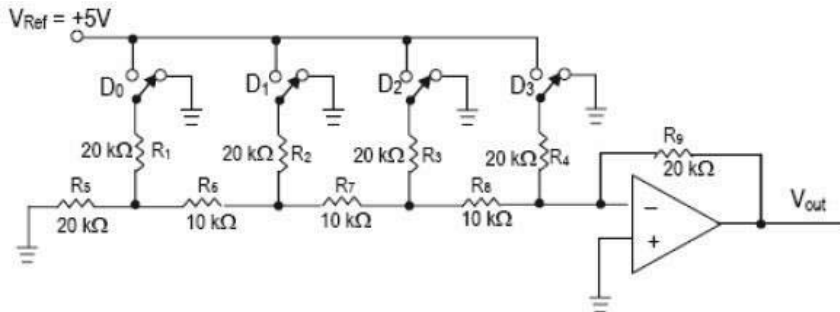
c. 24



8.

Gambar diatas merupakan IC ADC0804. Pin yang menunjukkan kaki clock in adalah nomor ...

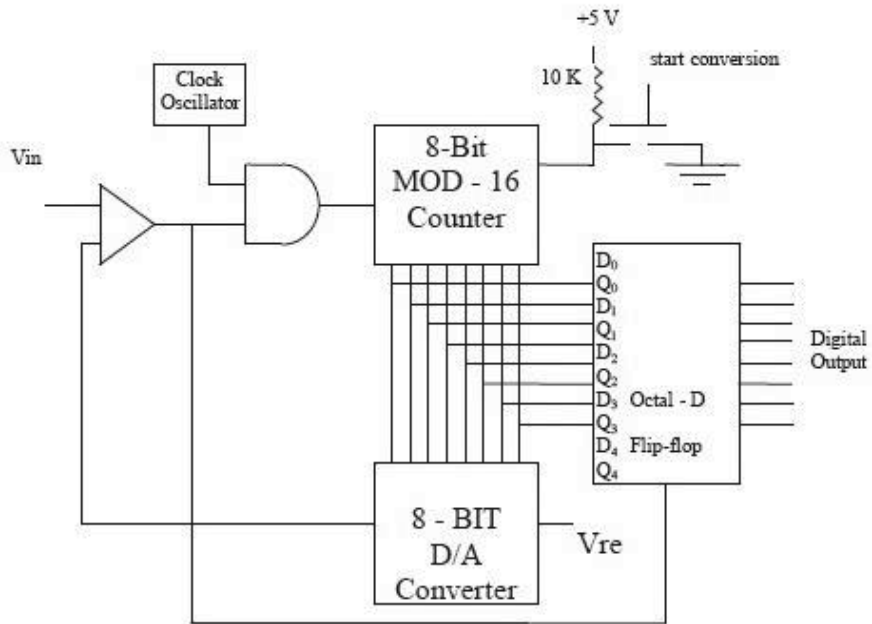
- a. 1
- b. 2
- c. 3
- d. 4
- e. 5



9.

Gambar diatas merupakan rangkaian ...

- a. DAC Ladder
- b. Binary Weighted DAC
- c. R/2R Ladder DAC
- d. Counter Ramp ADC
- e. Successive Approximation ADC



10.

Gambar diatas merupakan rangkaian dari ...

- | | |
|------------------------|---------------------------------|
| a. DAC Ladder | d. Counter Ramp ADC |
| b. Binary Weighted DAC | e. Successive Approximation ADC |
| c. R/2R Ladder DAC | |

Daftar Pustaka

1. Ronald J Tocci, Neal S Widmer, 2010, Digital Systems Principle and Applications, Eight Edition Prentice Hall, Colombus, Ohio
2. Malvino, Leach, 1975, Digital Principles and Applications, Mc. Graw Hill, Singapore.
3. Eko Didik Widiyanto, Sistem Digital: Analisis, Desain dan Implementasi, Edisi Pertama, Graha Ilmu, 2014
4. Peter J. Ashenden, Digital Design: An Embedded Systems Approach Using Verilog/VHDL, Morgan Kaufmann, 2008
5. Stephen Brown and Zvonko Vranesic, Fundamentals of Digital Logic with Verilog/VHDL, 2nd Edition, McGraw-Hill, 2005
6. KF Ibrahim, Teknik Digital, Penerbit Andi Yogyakarta, 1996
7. Wijaya Wijanarko, Teknik Digital, Penerbit Erlangga Surabaya, 2006