**MODUL II**

**ELEKTRONIKA DIGITAL**

**(GERBANG LOGIKA,RANGKAIAN SEQUENSIAL DAN RANGKAIAN KOMBINASI)**

1. **TUJUAN**

1. Mengetahui dan memahami prinsip gerbang logika.

2. Mampu menurunkan gerbang logika dalam rangkaian digital.

3. Memahami karakteristik dan cara kerja rangkaian sequensial yaitu flip-flop dan counter.

4. Mampu mendesain rangkaian counter dari rangkaian flip-flop.

5. Memahami karakteristik dan cara kerja rangkaian kombinasi yaitu multiplexer.

1. **DASAR TEORI**

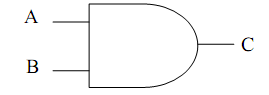
Saat ini sistem analog sudah banyak digantikan dengan sistem digital. Selain karena lebih akurat, sistem digital banyak dipilih karena lebih stabil apabila dikenai noise. Contoh sistem digital ini misalnya pada kalkulator dan komputer. Dalam membentuk suatu sistem pada komputer diperlukan gerbang-gerbang logika yangterbuat dari jutaan transistor pada mikroprosesor. Gerbang logika memiliki satuatau lebih masukan (input) dan sebuah keluaran (output), dengan masukan ataupun keluarannya bernilai tinggi (high) biasa dilambangkan dengan angka 1, danmasukan atau keluaran yang bernilai rendah (low) dilambangkan dengan 0. Dalam praktikum ini kita akan mempelajari 7 gerbang logika yaitu AND, NAND, OR, NOR, Inverter, XOR, XNOR. Selain itu kita juga akan mempelajari rangkaian sequensial dan rangkaian kombinasi.

1. **GERBANG LOGIKA**

Banyak IC gerbang logika di pasaran, namun yang sering digunakan berasal dari jenis 74xx atau 40xx. Oleh karena itu kita perlu mengetahui spesifikasi tiap-tiap IC tersebut sesuai dengan kebutuhan gerbang logika yang kita perlukan.

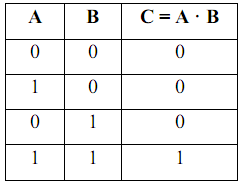
**1.1 Gerbang AND**

Gerbang AND memiliki 2 ataulebih input dan 1 output dengan nilai keluaran akan bernilai 1 jika nilai seluruh masukan bernilai 1. Dan akan bernilai 0 jika salah satu masukannya bernilai rendah atau 0. Lambang gerbang AND seperti pada gambar 1.1 dengan 2 input (A & B) dan 1 output (C).



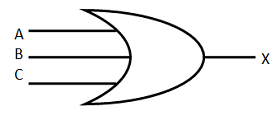
Gambar 1.1. Simbol Gerbang AND

Tabel 1.1 Tabel kebenaran gerbang AND

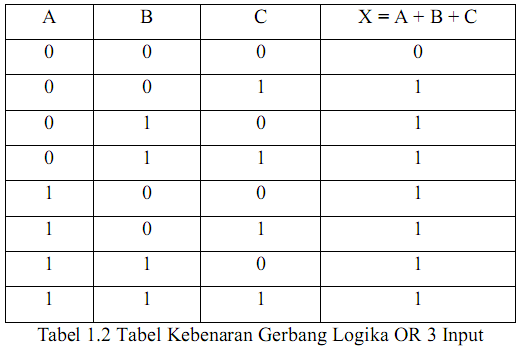


* 1. **Gerbang OR**

Gerbang OR memiliki 2 atau lebih masukan dan 1 keluaran, dengan nilai keluaran akan bernilai 1 bila salah satu saja input bernilai 1. Untuk memudahkan pemahaman maka seperti dalam persamaan Boolean, untuk gerbang OR semua masukan dijumlahkan,sedangkan untuk gerbang AND semua masukan dikalikan. Lambanggerbang OR untuk 3 input (A, B, C) dan 1 output X tampak dalam gambar 1.2 di bawah ini:



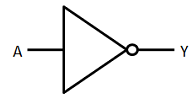
Gambar 1.2 Simbol Gerbang OR 3 input



* 1. **Gerbang Inverter (NOT)**

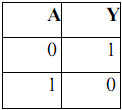
Gerbang inverter menghasilkan nilai output yang berkebalikan dengan nilai inputnya (dalam bilangan biner). Misal nilai masukan tinggi (1) maka output dari gerbang ini adalah rendah (0). Begitu juga sebaliknya bila input low maka ouputnya high. Untuk menandakan bahwa gerbang logika tersebut adalah inverter maka simbolnya berupa gerbang logika dengan lingkaran kecil di ujung outputnya. Dan dalam penulisan persamaan Boolean adalah dengan cara memberi garis di atas variabel.

Contoh X =  artinya X = tidak A. Terdapat 3 gerbang logika inverter yaitu Inverter, NAND (NOT-AND), dan NOR (NOT-OR). Ketiga gerbang inverter memiliki fungsi yang sama, namun dalam digunakan pada gerbang logika yang berbeda.



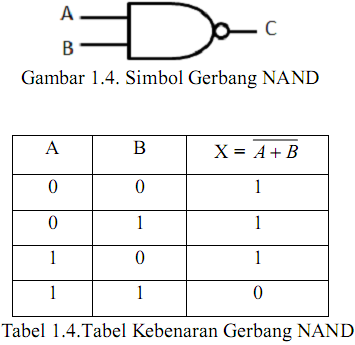
Gambar 1.3. Simbol Gerbang Inverter (NOT)

Tabel 1.3 Tabel Kebenaran Gerbang Inverter (NOT)



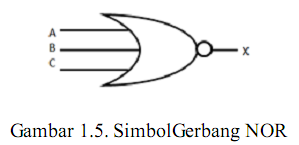
* 1. **Gerbang NAND**

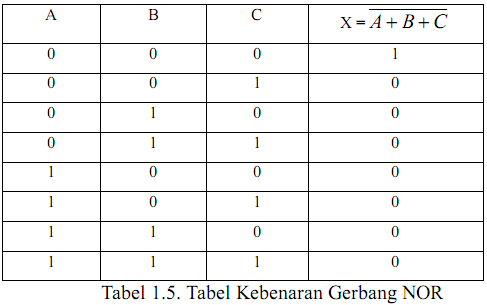
Gerbang NAND adalah gerbang AND yang keluarannya diinventer. Output C berasal dari perkalian antara A dan B yang kemudian di inverter, yang dalam persamaan Boolean X = . Simbol dari gerbang NAND adalah gerbang AND yang diberi lingkara kecil di jalur keluarannya, sepeti dalam gambar 1.4.



* 1. **Gerbang NOR**

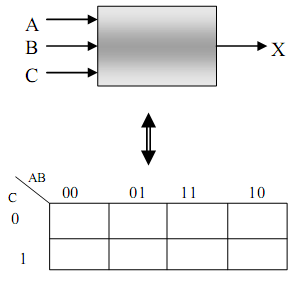
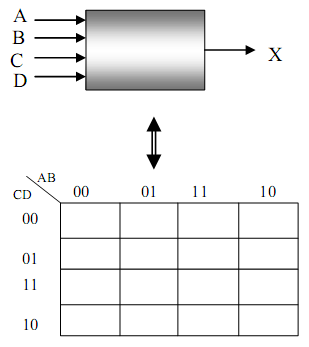
Seperti halnya gerbang NAND, gerbang NOR adalah operasi gerbang OR yang jalur keluarannya diinverter. Persamaan Boolean dari operasi gerbang NOR adalah dengan menginverterkan jumlahan input yang ada. Contoh pada gambar 1.4 di bawah ini:





1. **PETA KATNAUGH**

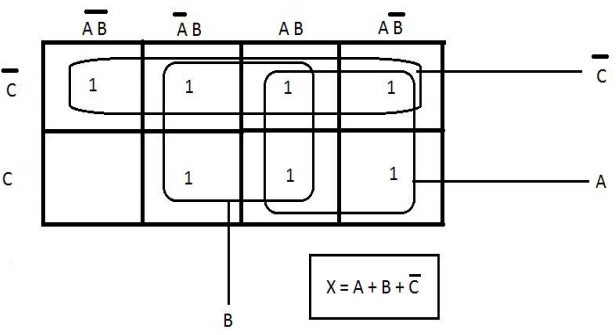
Dalam suatu rangkaian logika yang tersusun dari ratusan atau bahkan jutaan gerbang logika dapat menimbulkan masalah dalam analisa logika ataupun hasil yang dicapai. Untuk mengatasinya terdapat suatu metode yang bernama Peta Karnaugh, atau biasa disebut K-map. Metodenya menjumlahkan seluruh masukan dengan cara memasukkan nilai input pada tabel. Dalam analisa variabelnya metode K-map tergantung pada jumlah masukan di awal.

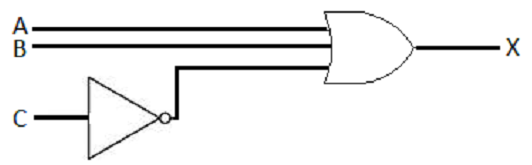
Gambar 2.1 K-map 3 Input

Contoh penyederhanaan persamaan dengan K-map adalah :





Sehingga Bentuk Rangkaian inplementasinya adalah sebagai berikut :



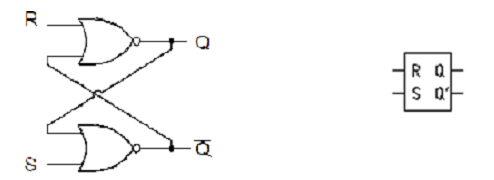
1. **RANGKAIAN SEQUENSIAL**

Salah satu rangkaian logika yang sangat bermanfaat yaitu rangkaian logika sekuensial yang di interkoneksikan untuk penyimpanan, pewaktu, perhitungan dan pengurutan. Bentuk dasar dari rangkaian logika sekuensial adalah rangkaian flip-flop yang dirangkai dari gerbang logika seperti NAND dan AND. Nama lain dari flip-flop adalah multivibrator bistabil, dimana keluarannya adalah suatu tegangan rendah (0) atau tinggi (1). Keluaran ini tetap rendah atau tinggi selama belum ada masukkan yang merubah keadaan tersebut. Rangkaian yang bersangkutan harus di-drive (dikendalikan) oleh satu masukkan yang disebut pemicu (trigger). Keadaan tersebut akan berubah kembali bila ada masukkan pemicu lagi. Ada tiga jenis multivibrator,yaitu : astabil, monostabil, dan bistabil. Pada bagian ini hanya membahas multivibrator bistabil (flip-flop). Berdasarkan cara penyimpanannya flip-flop dapat digolongkan atas :

* RS Flip-flop
* JK Flip-flop
* D Flip-flop
* T Flip-flop

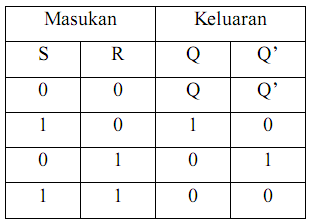
3.1 **RS Flip-Flop**

Kebanyakan flip-flop dasar disebut flip-flop RS. RS flip-flop tersusun dari rangkaian NAND atau NOR. Gambar dibawah adalah symbol dari RS flip-flop dan rangkaian RS flip-flop dengan logika NOR serta table kebenarannya.



Gambar 3.1 Simbol Logika RS Flip-Flop

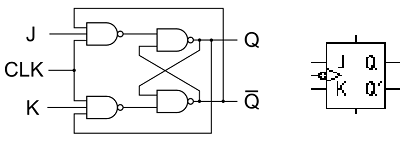
Tabel 3.1. Tabel Kebenaran Nor RS Flip-Flop.



Baris pertama dari tabel kebenaran disebut Hold karena tidak adanya perubahan, dalam arti bahwa sama sekali tidak diberi perlakuan. Baris kedua dari tabel kebenaran menunjukkan kondisi set dari flip-flop. Disini masukan low (0) mengaktifkan masukan set (S). Logika 0 ini mengeset keluaran Q menjadi tinggi atau 1. Kondisi set ini akan terlihat bila kita menganalisa rangkaian NAND diatas. Logika 0 pada gerbang 1 membangkitkan 1 pada keluaran. Logika 1 ini dimasukkan lagi ke gerbang 2. Sekarang gerbang 2 mempunyai dua logika 1 yang dimasukkan pada masukannya, sehingga mendorong keluaran menjadi 0. Baris ketiga dari tabel kebenaran merupakan kondisi reset. Logika 0 mengaktifkan masukan reset tersebut. Hal ini akan mereset keluaran Q menjadi 0.

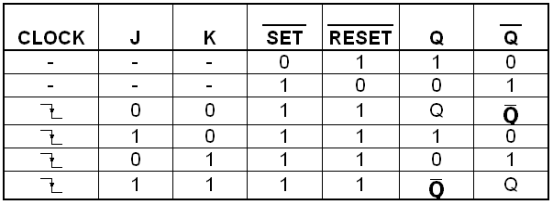
* 1. **JK Flip-Flop**

Rangkaian ini dapat dianggap sebagai flip-flop universal. Flip-flop jenis lain dapat dibuat dari flip-flop ini. Simbol logika unutk ini ditunjukkan oleh gambar dibawah ini.



Gambar 3.2 JK Flip-Flop

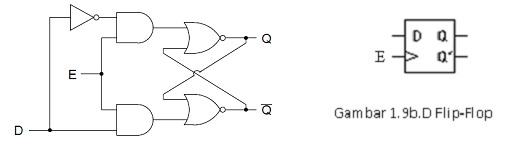
Tabel 3.2. Tabel Kebenaran JK Flip-Flop



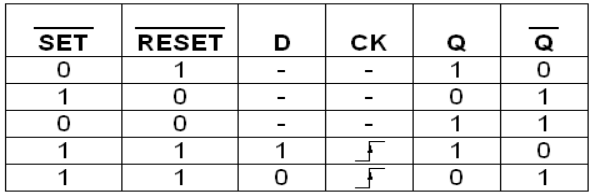
Tabel kebenaran untuk flip-flop ini diberikan oleh gambar di atas. Mode operasi diberikan di sebelah kiri dan tabel kebenaran di sebelah kanan. Baris 1 dari tabel kebenaran tersebut menunjukkan kondisi tetap atau kondisi terbuka. Perhatikan bahwa semua data masuk (J dan K) adalah LOW. Kondisi reset atau clear dari flip-flop tersebut ditunjukkan pada baris 2 dari tabel kebenaran. Bila J=0 dan K=1 serta pulsa clock dating pada masukan CK, maka flip-flop tersebut direset (Q=0). Baris 3 menunjukkan kondisi set dari flip-flop JK. Bila J=1 dan K=0 serta terdapat pulsa clock , maka keluaran Q akan menjadi 1. Baris 4 mengilustrasikan kondisi yang sangat berguna dari flip-flop JK. Kondisi ini disebut posisi toggle. Bila masukan J dan K kedua – duanya HIGH, maka keluaran Q akan berlawanan dengan keadaan pada waktu pulsa clock tiba pada masukan CK.

* 1. **D Flip-Flop**

Flip-flop D sering sekali disebut flip-flop tunda (delay). Nama ini menggambarkan secara tepat cara kerja dari flip-flop ini. Apapun bentuk masukkan data (D), masukan tersebut akan tertunda selama satu pulsa clock untuk mencapai keluaran. Gambar dibawah menunjukka symbol dari D flip-flop dan tabel kebenarannya.



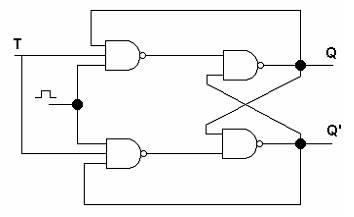
Tabel 3.3. Rangkaian dan Tabel Kebenaran D Flip-Flop.



Dapat terlihat hasil keluaran dari D flip-flop tersebut sesuai dengan tabel kebenaran diatas. Dimana nilai keluaran (Q) akan selalu sama dengan nilai inputan D jika nilai masukan enable adalah HIGH. Tetapi sebaliknya jika nilai masukan enable adalah LOW maka apapun nilai masukan pada D, nilai keluaran (Q) akan bernilai sama dengan nilai sebelumnya.

* 1. **T Flip-Flop**

Merupakan modifikasi dari JK flip-flop, dengan menggabungkan kedua masukan J danK sehinga keluaran akan berubah-ubah sesuai perubahan pada clocknya.



Gambar 3.4 Rangkaian T Flip-Flop

1. **COUNTER**

Counter (pencacah) adalah alat/rangkaian digital yang berfungsi menghitung/mencacah banyaknya pulsa cIock atau juga berfungsi sebagai pembagi frekuensi, pembangkit kode biner, Gray. Ada 2 jenis pencacah yaitu : Pencacah sinkron *(syncronuous*counter*s)* dan Pencacah tak sinkron *(asyncronuous* counter*s)* yang disebut juga pencacah deret *(series* counter*s)*.

Sebagaimana dengan rangkaian sekuensial yang lain, untuk menyusun counterdigunakan *flip-flop*. Counterdapat digunakan untuk menghitung banyaknya *clock*-pulsa dalam waktu yang tersedia (pengukuran frekuensi), counterdapat juga digunakan untuk membagi frekuensi dan menyimpan data. Jenis flip-flop yang diperlukan/digunakan (D-FF, JK FF atau RS-FF). Prasyarat perubahan logikanya (dari flip-flop yang digunakan).

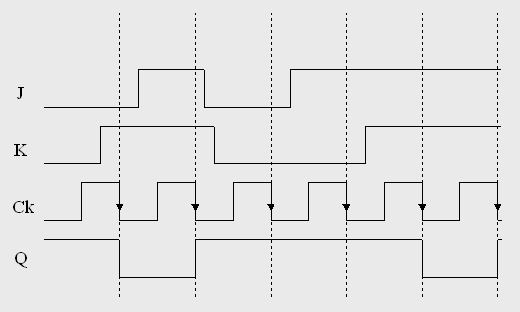
****

Gambar 4.1 Simbol flip-flop J-K dengan *preset* dan *reset.*

Tabel 4.1. Tabel kebenaran dari flip-flop J-K

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Input** | | | | | **Output** | |
| **Preset** | **Clear** | **Clock** | **J** | **K** | **Q** |  |
| L | H | X | X | X | H | L |
| H | L | X | X | X | L | H |
| L | L | X | X | X | Race | race |
| H | H |  | L | L | Tetap | tetap |
| H | H | ↓ | H | L | H | L |
| H | H | ↓ | L | H | L | H |
| H | H | ↓ | H | H | toggle | |
| H | H | H | X | X | Q0 | Q0 |

Flip-flop ini akan bekerja sebagai counter jika pin J dan K diberi logika tinggi (seperti terlihat dalam tabel, output Q akan mengalami toggle jika ada transisi clock tinggi ke rendah). Sinyal yang dicacah tidak lain adalah sinyal clock itu sendiri. Sebelum sampai ke counter, Gambar berikut memberikan penjelasan bagaimana flip-flop J-K ini bekerja.



Gambar 4.2. Diagram pewaktuan flip-flop J-K

Dari Gambar tersebut dapat dilihat bahwa, ketika J adalah *low*, dan K adalah *high* serta sinyal clocknya adalah transisi dari tinggi ke rendah, maka output (Q) akan bernilai *low*. Ketika J dan K adalah berlogika *high* dan sinyal clocknya adalah transisi dari tinggi ke rendah, maka *output* (Q) akan berubah ke komplemen sebelumnya (di sini berubah dari rendah ke tinggi) yang disebut mengalami kondisi *toggle.*

Rangkaian counter menggunakan flip-flop J-K dapat diwujudkan menggunakan rangkaian. Counter pada rangkaian gambar memiliki lebar data sebesar 3 bit, serta jenis counternya adalah jenis asinkron. Jika menginginkan lebar data yang yang lebih besar maka tinggal merangkaian seri sebanyak yang diinginkan. Diagram pewaktuannya dapat dilihat pada gambar diatas.



Gambar 4.3.Rangkaian counter 3 bit

**4.1 Rangkaiancounter sinkron**

* *Up* Counter

Pada countersinkron, sumber *clock* diberikan pada masing-masing *input clock* dari flip-flop penyusunnya, sehingga apabila ada perubahan pulsa dari sumber, maka perubahan tersebut akan men-t*rigger* seluruh *flip-flop* secara bersama-sama.



Gambar 4.4.Up counter sinkron 3 bit

Sebuah counter disebut sebagai *up* counter jika dapat menghitung secara berurutan mulai dari bilangan terkecil sampai bilangan terbesar. Contoh : 0-1-2-3-4-5-6-7-0-1-2-…. Sedangkan *down* counter adalah counter yang dapat menghitung secara berurutan dari bilangan terbesar ke bilangan terkecil

Tabel 4.2. Tabel PS/NS untuk *up* dan *down* counter 3 bit



Untuk membuat sebuah rangkaian *up* counter*,* lakukan langkah-langkah sintesa rangkaian yang telah dijelaskan sebelumnya. Dari hasil persamaan logika berdasarkan Tabel PS NS di atas didapatkan rangkaian seperti di bawah ini :



Gambar 4.5. Rangkaian *up* counter sinkron 3 bit

* *Down* Counter

Berdasarkan tabel PS/NS, dapat dilihat bahwa *down counting* merupakan kebalikan dari *up counting*, sehingga rangkaiannya masih tetap menggunakan rangkaian *up* counter, hanya *output*nya diambilkan dari *Q* masing-masing flip-flop. Bentuk rangkaian *down* counter yaitu :



Gambar 4.6 Rangkaian *down* counter sinkron 3 bit

* *Up-Down* Counter

Rangkaian *up/down c*counter merupakan gabungan dari *up* counter dan *down* counter. Rangkaian ini dapat menghitung bergantian antara *up* dan d*own* karena adanya *input* eksternal sebagai *control* yang menentukan saat menghitung *up* atau *down*. Pada gambar dibawah ini ditunjukkan rangkaian *up/down* counter sinkron 3 bit. Jika *input* CNTRL bernilai ‘1’ maka counterakan menghitung naik (*up),* sedangkan jika *input* CNTRL bernilai ‘0’, c*ounter* akan menghitung turun *(down)*.



Gambar 4.7.Rangkaian *up/down* counter sinkron 3 bit

**4.2 Rangkaian CounterAsinkron**

Pada counterasinkron, sumber *clock* hanya diletakkan pada input Clock di Flip-flop terdepan (bagian *Least Significant Bit* / *LSB*), sedangkan input-input clock Flip-flop yang lain mendapatkan catu dari output flip-flop sebelumnya. Konfigurasi ini didapatkan dari gambar timing diagram counter 3-bit . Dengan konfigurasi ini, masing-masing flip-flop di-*trigger* tidak dalam waktu yang bersamaan (*ripple* counter).

Tabel Kebenaran 4.3.Tabel kebenaran *up* counter asinkron 3-bit



Berdasarkan bentuk timing diagram di atas, *output* dari flip-flop C menjadi clock dari flip-flop B, sedangkan *output* dari flip-flop B menjadi *clock* dari flip-flop A. Perubahan pada negatif di masing-masing *clock* flip-flop sebelumnya menyebabkan flip-flop sesudahnya berganti kondisi *(toggle),* sehingga *input-input* J dan K di masing-masing flip-flop diberi nilai ”1” (sifat *toggle* dari JK flip-flop).



Gambar 4.8.Up counter asinkron 3 bit

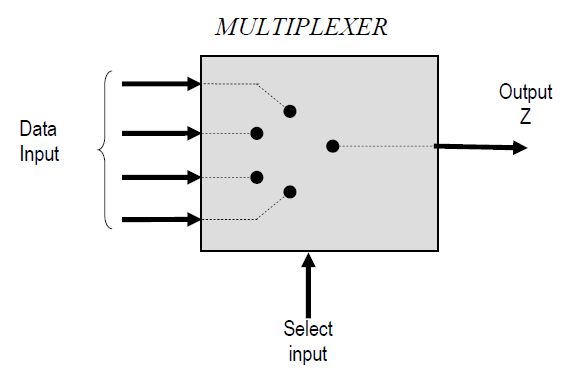
1. **RANGKAIAN KOMBINASIONAL**

Rangkaian kombinasional mempunyai nilai keluaran di suatu waktu hanya ditentukan oleh nilai dari masukannya di waktu tersebut. Dalam rangkaian ini tidak ada penyimpanan informasi atau ketergantungan terhadap keadaan rangkaian (keluaran) sebelumnya. Contoh rangkaian kombinasional adalah multiplekser, enkoder, dekoder, demultiplekser dan ALU.

**5.1 Multiplexer**

Sebuah *Multiplexer* adalah rangkaian logika yang menerima beberapa input data digital dan menyeleksi salah satu dari input tersebut pada saat tertentu, untuk dikeluarkan pada sisi output.

Seleksi data-data input dilakukan oleh *selector line*, yang juga merupakan input dari *multiplexer* tersebut. Blok diagram sebuah *multiplexer* ditunjukkan pada gambar.

****

Gambar 5.1.Blok Diagram Multiplexer

Jumlah data input maksimum pada multiplexer adalah 2 jumlah Select line.

Tabel Kebenaran sebuah Multiplexer ditunjukkan pada Tabel dibawah ini.

Tabel 5.1. Kebenaran Multiplexer dengan 2 Select line

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | | | | | | **Output** | |
| **S0** | **S1** | **D0** | **D1** | **D2** | **D3** | **X** | **Ket** |
| **0** | **0** | **0** | **X** | **X** | **X** | **0** | **D0** |
| **0** | **0** | **1** | **X** | **X** | **X** | **1** |
| **0** | **1** | **X** | **0** | **X** | **X** | **0** | **D1** |
| **0** | **1** | **X** | **1** | **X** | **X** | **1** |
| **1** | **0** | **X** | **X** | **0** | **X** | **0** | **D2** |
| **1** | **0** | **X** | **X** | **1** | **X** | **1** |
| **1** | **1** | **X** | **X** | **X** | **0** | **0** | **D3** |
| **1** | **1** | **X** | **X** | **X** | **1** | **1** |

* 1. **De*multiplexer***

De*multiplexer* atau DEMUX (demultiplexer) adalah kebalikan dari MUX. Diagram blok untuk DMUX 1-ke-4 dengan kendali masukan A dan B serta tabel kebenaran yang sesuai ditunjukkan oleh gambar di bawah ini. DEMUX mengirim data masukan D ke salah satu jalur keluaran Fi yang ditentukan oleh kendali masukan. Rangkaian DEMUX 1-ke-4 ditunjukkan pada gambar. Aplikasi DEMUX digunakan untuk mengirim data dari satu sumber ke salah satu dari sejumlah tujuan, seperti tombol pada elevator kepada wahana elevator terdekat. DEMUX tidak biasa digunakan pada implementasi fungsi Boolean umumnya, walaupun cara ini juga bisa dilakukan.

****

Gambar 5.2.Diagram blok dan tabel kebenaran untuk DEMUX 1-ke-4

****

Gambar 5.3.Rangkaian DEMUX 1-ke-4

**C. PERALATAN**

1. Breadboard

2. Multimeter

3. IC 7404, 7432, 7408 .

4. LED

5. Power Supply 1 buah.

6. Kabel jumper.

1. **PROSEDUR PERCOBAAN**

**Gerbang Logika**

* 1. Buat rangkaian logika sesuai dengan :



* 1. Sederhanakan rangkaian tersebut dengan K-map dan buat rangkaiannya
  2. Beri perlakuan pada rangkaian dengan memberikan kondisi High & Low sesuai dengan nilai bit yang yang anda tentukan.
  3. Amati nyala led (posisi led akan ditentukan oleh asisten). Selidiki apakah sesuai dengan logika dan kondisi yang diberikan.
  4. Apabila tidak sesuai gunakan multimeter untuk mengukur tegangan pada led yang tidak sesuai tersebut.
  5. Buat tabel percobaan untuk percobaan tersebut

**Flip-Flop**

* + 1. **-** Rangkaialah RS Flip-flop menggunakan gerbang NOR.
       - Rangkaian RS Flip-flop menggunakan gerbang NAND.
    2. Buatlah tabel kebenaran hasil percobaan tersebut dan bandingkan dengan tabel kebenarannya.

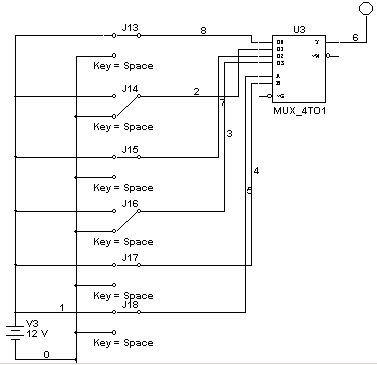
**Counter**

1. Buat rangkaian seperti gambar dibawah ini



1. Buatlah rangkaian Up Counter sinkron 3 bit dengan menggunakan metode K-Map
2. Bandingkan rangkaianmu dengan gambar diatas
3. Analisis apakah ada perbedaan hasil rangkaianmu dengan gambar diatas
4. Aktifkan rangkaian pada gambar diatas dan catat data dengan membuat tabel hubungan antara masukan dan keluaran pada rangkaian.
5. Analisis tabel kebenaran rangkaian pada gambar diatas.

**Multiplexer**



1. Susun Rangkaian Multiplexer 2 select line seperti gambar diatas
2. Catat sinyal keluaran yang terjadi untuk beberapa kombinasi data selektor
3. Buat Tabel kebenaran dari hasil keluaran
4. Bandingkan Hasil keluaran rangkaian anda dengan Tabel kebenaran
5. Analisis hasil keluaran, masing-masing praktikan harus berbeda