**BAB I**

**PENDAHULUAN**

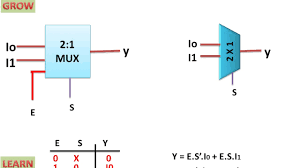
* 1. **Tujuan**

1. Memahami Prinsip kerja dari Mux, Demux, Encoder dan Decoder.
2. Dapat membuat rangkaian full adder dengan decoder 3 ke 8 yang menggunakan 2 buah gerbang OR.
3. Dapat menganalisis sebuah rangkaian yang terdiri dari multiplexer, demultiplexer, encoder, decoder.
   1. Permasalahan
4. Bagaimana cara merancang rangkaian full adder dengan decoder 3 ke 8 yang menggunakan buah gerbang OR.
5. Bagaimana cara menganalisa sebuah rangkaian yang terdiri dari multiplexer, demultiplexer, encoder, decoder.

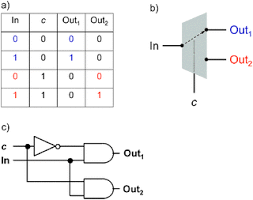
**BAB II**

**DASAR TEORI**

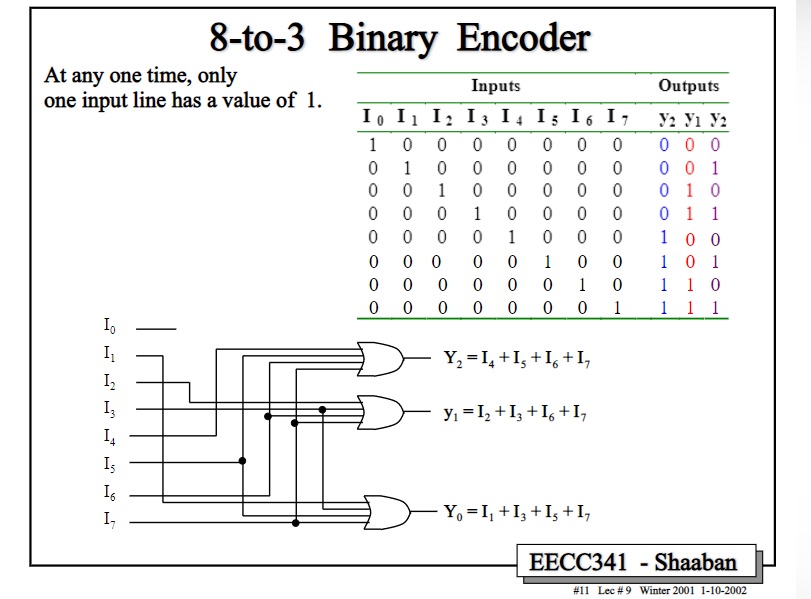
**DASAR TEORI**

***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. Multiplexer disebut juga pemilih data. Jumlah input multiplexer adalah 2n (n=1,2,3,…) dengan n merupakan jumlah bit sinyal pemilih, sehingga terdapat MUX 2 ke 1 dengan 1 bit sinyal pemilih, MUX 4 ke 1 dengan 2 bit sinyal pemilih, MUX 8 ke 1 dengan 3 bit sinyal pemilih, dan seterusnya. Contoh dari Mux 2 to 1.

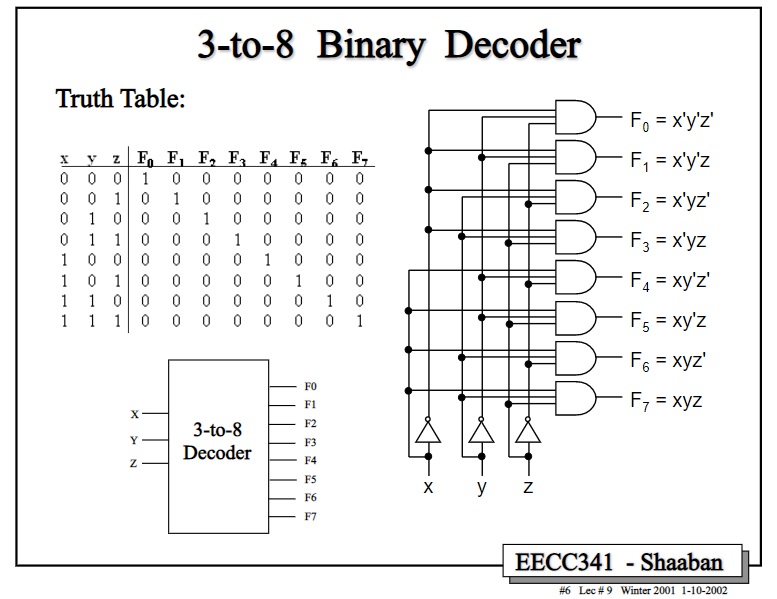
***Demultiplexer*** merupakan rangkaian logika yang berfungsi menyalurkan data yang ada pada inputnya ke salah satu dari beberapa outputnya dengan bantuan sinyal pemilih atau sinyal control. Demultiplexer sering disebut DEMUX. Demultiplexer disebut juga sebagai penyalur data(data distributor), dan fungsinya merupakan kebalikan dari fungsi multiplexer. Jumlah output demultiplexer adalah 2n (n=1,2,3,…), dalam hal ini n merupakan jumlah sinyal bit pemilih, sehingga terdapat DEMUX 1 ke 2 dengan 1 bit sinyal pemilih, DEMUX 1 ke 4 dengan 2 bit sinyal pemilih, DEMUX 1 ke 8 dengan 3 bit sinyal pemilih, dan seterusnya



***Encoder*** adalah rangkaian yang memiliki fungsi berkebalikan dengan dekoder. Encoder berfungsi sebagai rangakain untuk mengkodekan data input mejadi data bilangan dengan format tertentu. 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. 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).



***Decoder*** adalah alat yang digunakan untuk dapat mengembalikan proses encoding sehingga kita dapat melihat atau menerima informasi aslinya. Decoder juga dapat di artikan sebagai rangkaian logika yang di tugaskan untuk menerima input input biner dan mengaktifkan salah satu outputnya sesuai dengan urutan biner tersebut. Kebalikan dari decoder adalah encoder. 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, di mana kombinasi setiap inputnya dapat mengaktifkan beberapa output linenya.

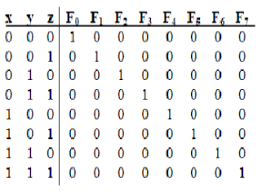


**BAB III**

**PEMBAHASAN**

* 1. **Membuat Rangkaian Decoder**

Rangkaian full adder dengan decoder 3 ke 8 dengan menggunakan 2 buah gerbang OR dapat dibuat dengan menggabungkan rangkaian decoder 3 ke 8 d selanjutnya diimplementasi dari full adder. Untuk membuat sebuah 3 ke 8 decoder, kita perlu membuat table kebenarannya terlebih dahulu dan menentukan keluarannya agar mudah dalam membuat rangkaiannya

****

= ‘X’Y’Z

= ‘X’YZ

= ‘XY’Z

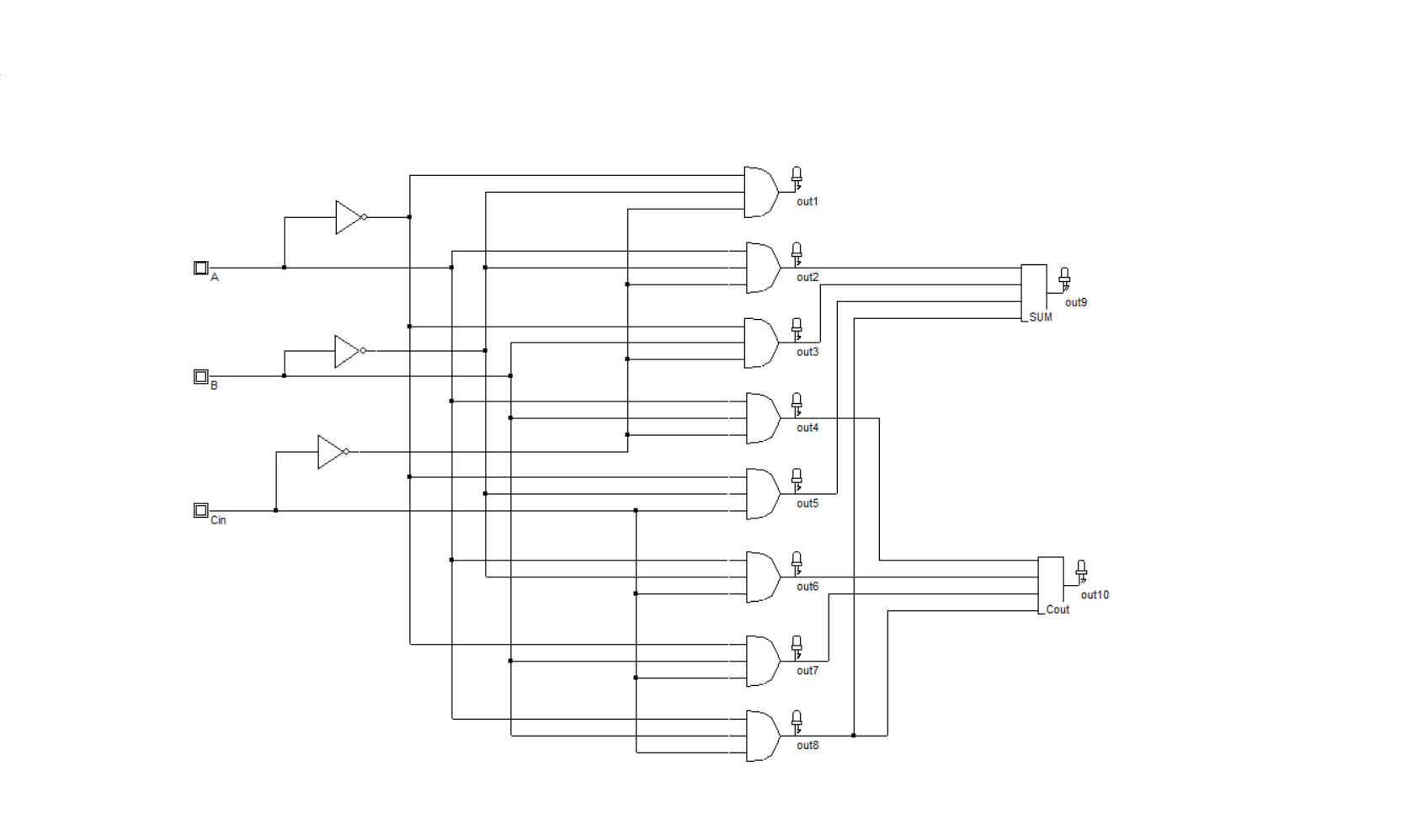
= X’Y’Z

= ‘XYZ

= X’YZ

= XY’Z

= XYZ

****

\*Catatan: Kotak 4 input = Gerbang OR dari (Mux4ke1.sch)\

Setelah kita membuat rangkaian decoder 3 ke 8, lalu kita membuat table kebenaran untuk full adder, supaya kita tahu mana yang SUM dan mana yang CarryOut.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | Cin | SUM | Cout |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

m0

m1

m2

m3

m4

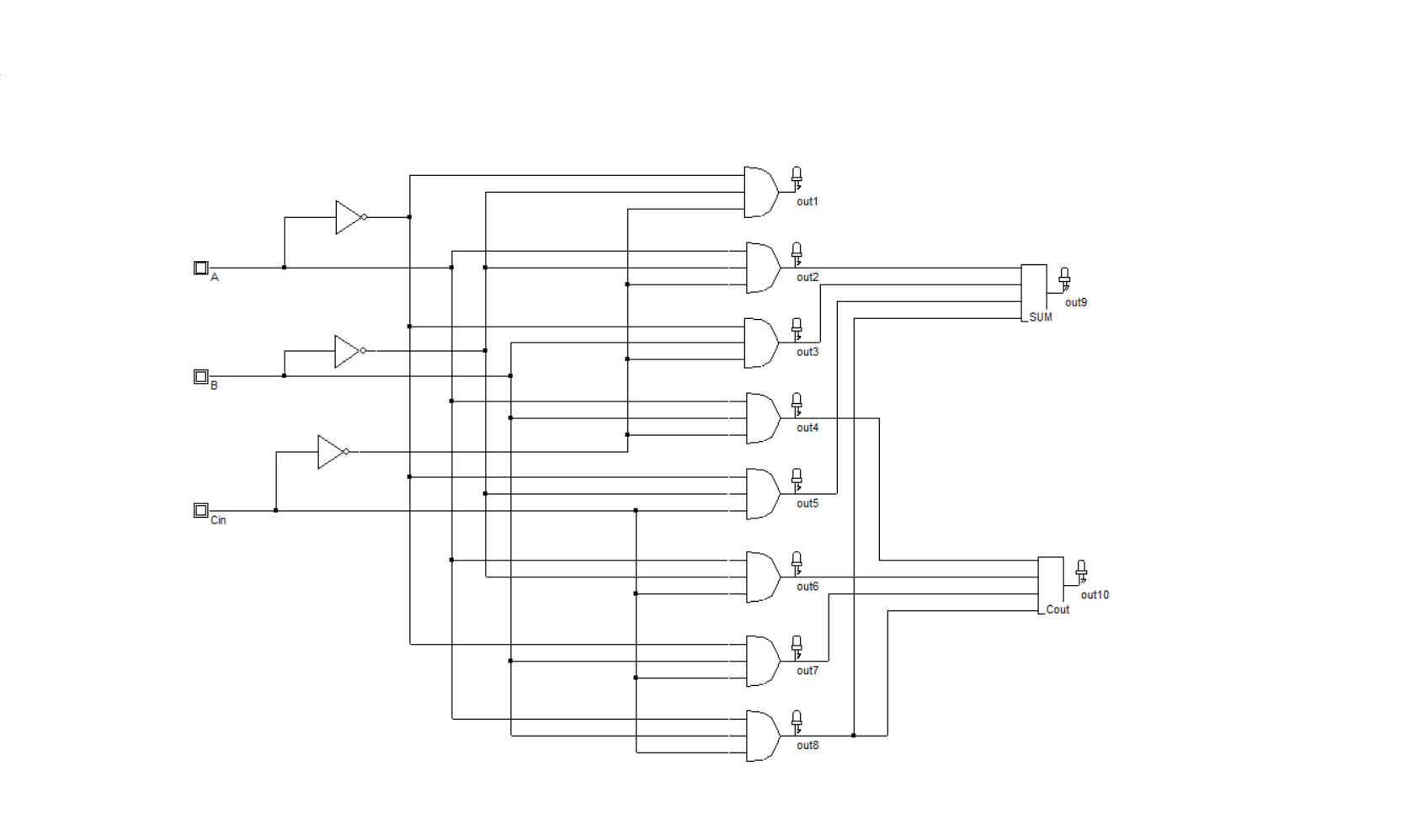
m5

m6

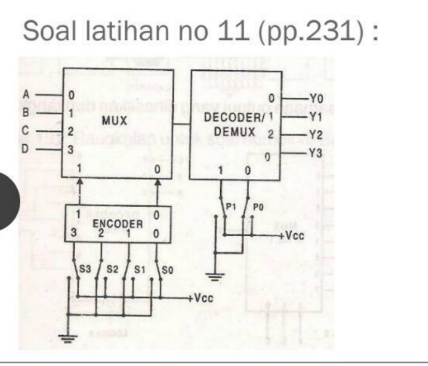
m7

untuk menentukan SUM dan CarryOut kita juga harus menentukan minterm nya terlebih dahulu. Setelah itu baru kita membuat rangkaian dari full adder

Fs = ∑(m1,m2,m4,m7)

****Fcout = ∑(m3,m5,m6,m7)

* 1. **Analisis Rangkaian**



Dari rangkaian diatas, kita dapat membuat sebuah tabel. A menunjukan saklar yang mengarah kekanan atau dalam posisi aktif. Sedangkan TA menunjukan saklar yang mengarah ke kiri atau tidak aktif.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SAKLAR ENCODER | | | | SAKLAR DECODER | | OUTPUT DECODER/DEMUX | | | |
| S3 | S2 | S1 | S0 | P1 | P0 | Y0 | Y1 | Y2 | Y3 |
| TA | TA | TA | A | A | TA | 0 | 0 | A | 0 |
| TA | A | A | TA | A | A | 0 | 0 | 0 | C |
| TA | TA | A | TA | TA | TA | B | 0 | 0 | 0 |
| A | A | TA | TA | TA | TA | D | 0 | 0 | 0 |
| TA | TA | A | TA | A | TA | 0 | 0 | A | 0 |
| TA | A | TA | TA | TA | TA | C | 0 | 0 | 0 |

\*Catatan = TA : Tidak Aktif dan A : Aktif

Pada rangkaian diatas digunakan prinsip encoder prioritas, artinya untuk membangkitkan kode biner pada outputnya, saklar pada input tertinggi saja yang diperhatikan atau diprioritaskan. Keadaan saklar-saklar selain saklar pada input tertinggi diabaikan. Kita ambil contoh pada baris keempat tabel tersebut. Terlihat saklar encoder yang diaktifkan adalah saklar 2 dan saklar 3. Berdasarkan prinsip prioritas, maka yang perlu diperhatikan hanyalah saklar 2. Sehingga encoder tersebut menghasilkan kode biner dari 2 yaitu 10. Kemudian nilai tersebut masuk ke dalam multiplexer sebagai selector input. Hasil dari selector dengan input 00 menyebabkan data pada input 4 terpilih yaitu huruf D. Lalu huruf D tersebut masuk ke decoder/demux. Pada decoder/demux ini kembali terdapat selector untuk output. Fungsi dari selector ini adalah untuk memilih output berdasarkan input pada selector. Dalam hal ini P0 dan P1 bernilai TA atau 0, sehingga menghasilkan biner 00. Biner 00 sama dengan 0 desimal yang kemudian menentukan output yaitu pada Y0. Sehingga input huruf D tadi akan keluar pada output Y0.

**BAB IV**

**PENUTUP**

* 1. **Kesimpulan**
* Dari pembahasan diatas dapat disimpulkan bahwa rangkaian full adder dalam decoder 3 ke 8 yang menggunakan 2 buah gerbang OR dapat dibuat dengan membuat decoder 3 ke 8 terlebih dahulu dengan menggunakan tabel kebenaran yang dilihat dari tiap keluarannya. Lalu membuat tabel kebenaran dan menentukan minterm dari setiap outputnya berupa SUM dan Cout. Kemudian menambahkan 2 gerbang OR yang menampung minterm dari tiap output untuk membuat rangkaian full adder.
* Analisa dari rangkaian yang terdapat pada soal menunjukkan penggunaan dari encoder prioritas untuk menentukan output dari encoder tersebut. Kemudian output tersebut dijadikan selector pada multiplexer untuk menentukan input yang dipilih. Lalu output dari multiplexer masuk ke demux sebagai input. Terdapat juga selector untuk menentukan output mana yang akan dipilih untuk mengeluarkan nilai.