

Mata Kuliah
Pengantar Sistem Digital

Multiplexer dan Demultiplexer

Dosen Pengampu:
Arif Rifai Dwiyanto ST., MTI



Disusun Oleh:
Khalila indana (202410715127)

F3A7

PROGRAM STUDI INFORMATKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BHAYANGKARA JAKARTA
RAYA
2025

DAFTAR ISI

BAB I PENDAHULUAN	3
1.1 Tujuan Praktikum.....	3
BAB II TEORI DASAR.....	4
2.1 Multiplexer (MUX)	4
2.2 Demultiplexer (DEMUX).....	5
BAB III PERANCANGAN	8
3.1 Langkah-Langkah	8
3.1.1 Rangkaian Multiplexer	8
3.1.2 Rangkaian Demux	8
3.1.3 Rangkaian gabungan MUX dan DEMUX.....	8
BAB IV HASIL DAN PEMBAHASAN	9
4.1 Hasil Pengujian Rangkaian Multiplexer (MUX)	9
4.2 Hasil Pengujian Rangkaian Demultiplexer(DEMUX).....	10
4.3 Hasil Pengujian Rangkaian Gabungan MUX & DEMUX.....	12
BAB V KESIMPULAN.....	13

BAB I

PENDAHULUAN

1.1 Tujuan Praktikum

- 1.1 Mahasiswa memahami cara kerja Multiplexer dan Demultiplexer
- 1.2 Mahasiswa bisa membuat dan mensimulasikan MUX dan DEMUX menggunakan Logisim.
- 1.3 Mahasiswa dapat mengecek hasil simulasi untuk memastikan rangkaian berjalan sesuai dengan tabel kebenaran.

BAB II

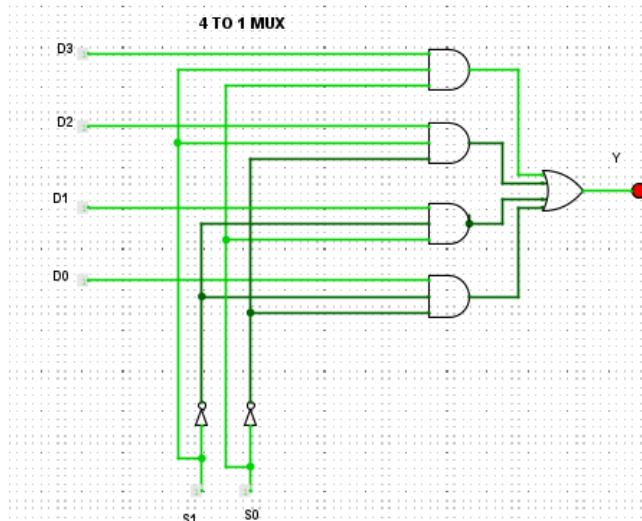
TEORI DASAR

2.1 Multiplexer (MUX)

Di dalam multiplexer terdapat beberapa input dan hanya satu output. Rangkaian logika digunakan untuk memilih satu dari beberapa sumber input yang akan diteruskan ke output tunggal, berdasarkan kombinasi sinyal kontrol. Semakin banyak input yang ada, semakin banyak pula kombinasi sinyal kontrol yang dibutuhkan. Multiplexer melakukan proses yang kebalikan dari demultiplexer. Berikut beberapa contoh multiplexer:

- Multiplexer 4x1 atau 4 to 1 multiplexer
- Multiplexer 8x1 atau 8 to 1 multiplexer
- Multiplexer 16x1 atau 16 to 1 multiplexer

Berikut adalah salah satu contoh 4 to 1 multiplexer:



Gambar 2.1 Multiplexer 4-to-1

Rangkaian tersebut menggambarkan sebuah fungsi untuk memilih satu dari empat input data (D0, D1, D2, D3) untuk diteruskan ke satu output tunggal. Pemilihan input data ke output akan ditentukan dari kombinasi dua sinyal selektor, yaitu A dan B. Maka MUX akan berfungsi sebagai saklar

elektronik yang mengarahkan salah satu dari beberapa sumber data yang menuju ke output.

Ekspresi logika dari rangkaian tersebut dapat dituliskan sebagai:

$$Y = (\bar{S1} \cdot \bar{S0} \cdot I0) + (\bar{S1} \cdot S0 \cdot I1) + (S1 \cdot \bar{S0} \cdot I2) + (S1 \cdot S0 \cdot I3)$$

Dari persamaan tersebut dapat disimpulkan hanya satu gerbang AND yang aktif ketika satu waktu, tergantung pada kombinasi S1 dan S0, maka hanya satu input yang diteruskan menuju output. Oleh karena itu, MUX bekerja sebagai saklar digital yang memilih dan meneruskan satu dari banyak sinyal input secara efisiensi melalui satu jalur keluaran.

Tabel 2.1 Truth Tabel MUX

S₁	S₀	Output Y
0	0	I ₀
0	1	I ₁
1	0	I ₂
1	1	I ₃

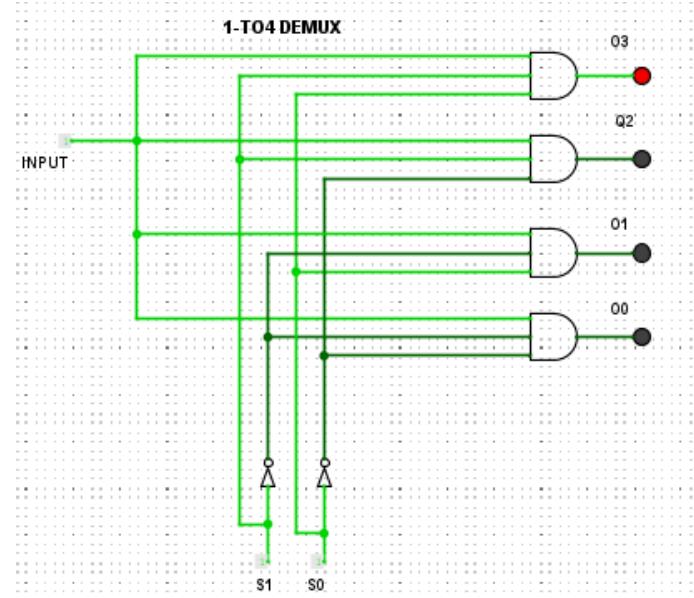
Pada tabel kebenaran terlihat bahwa output Y dihasilkan karena inputan dari kombinasi sinyal selektor, seperti S1=0 dan S0=0 maka inputan I0 diteruskan ke output. Ketika S1S0 = 01 maka input I1 diteruskan ke output, sedangkan S1S0 = 10 akan mengaktifkan I2 dan S1S0 = 11 mengaktifkan I3. Oleh karena itu, perubahan nilai pada jalur selektor menentukan input mana yang menjadi keluaran multiplexer.

2.2 Demultiplexer (DEMUX)

Rangkaian logika yang mengarahkan satu sinyal biner (data serial) pada input ke salah satu dari beberapa saluran output yang tersedia berdasarkan kombinasi sinyal kontrol. Dapat dikatakan bahwa demultiplexer menentukan jalur keluaran mana yang aktif tergantung pada nilai dari select line.

Proses

pemilihan output akan dilakukan melalui sinyal penyeleksi yang menjadi input tambahan bagi rangkaian. Select line yang berjumlah n akan menentukan satu dari 2^n output yang dapat aktif pada suatu waktu. Ketika kombinasi select line berubah, DEMUX akan memindahkan data input ke saluran output lainnya selaras dengan kombinasi tersebut. Mekanisme ini menjadikan demultiplexer berfungsi seperti saklar elektronik yang mendistribusikan satu sumber data ke beberapa tujuan secara terkontrol. Contoh demultiplexer seperti dibawah ini:



Gambar 1.1 Demultiplexer 1-to-4

Rangkaian 1-to-4 DEMUX tersebut bekerja untuk mengarahkan satu input data (DATA D) menuju ke salah satu dari empat output O0, O1, O2, O3. Pada kombinasi sinyal kontrol A dan B. menunjukan terdapat dua select line yang menentukan jalur output aktif. Kedua select line (A dan B) melewati gerbang NOT, maka sinyal A, A', B, B' tersedia dan membentuk kombinasi logika pada setiap gerbang AND.

Pada rangkaian DEMUX ini, sinyal masukan D kan diarahkan ke salah satu dari ke empat keluaran, yaitu (O0, O1, O2, 3) berdasarkan kombinasi sinyal

selektor S1 dan S0. Hubungan antara selektor dan output dapat dituliskan seperti ekspresi logika berikut:

$$O_0 = \bar{S1} \cdot \bar{S0} \cdot I$$

$$O_1 = \bar{S1} \cdot S0 \cdot I$$

$$O_2 = S1 \cdot \bar{S0} \cdot I$$

$$O_3 = S1 \cdot S0 \cdot I$$

Dengan demikian, hanya ada satu output yang akan aktif karena sesuai kombinasi sinyal selektor, sedangkan output lain bernilai 0.

Table 2.1 Truth Table Demux

S1	S0	Output O ₀	Output O ₁	Output O ₂	Output O ₃
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Secara garis besar, tabel ini menjelaskan bahwa output yang aktif hanya satu sesuai kombinasi selektor, sehingga fungsi utama DEMUX sebagai pembagi satu input ke salah satu dari beberapa jalur output dapat diselesaikan.

BAB III

PERANCANGAN

3.1 Langkah-Langkah

3.1.1 Rangkaian Multiplexer

- Buka program Logisim Evolution.
- Tambahkan 8 input data (I0 sampai I7), 3 input selektor (S0, S1, S2), dan 1 output LED (Y).
- Tambahkan 8 buah gerbang AND, lalu hubungkan setiap input data ke masing-masing gerbang AND.
- Tambahkan 3 buah gerbang NOT pada masing-masing input selektor (S0, S1, S2) untuk menghasilkan sinyal pembalik dari selektor.
- Setiap input data (I0 hingga I7) dihubungkan dengan selektor yang sesuai berdasarkan kebutuhan, lalu kirimkan ke gerbang AND yang sesuai.
- Hubungkan semua output dari 8 gerbang AND ke satu gerbang OR, sehingga hanya input yang sesuai dengan kombinasi selektor yang akan diteruskan ke output LED (Y).

3.1.2 Rangkaian Demux

1. Buka software leogisim evolution
2. Tambahkan komponen 8 gerbang AND (D0 sampai D7), 3 input selektor (S0,S1,S2), 1 input data (E), dan 8 output LED
3. Setiap gerbang AND dihubungkan ke input data
4. Tambahkan 3 gerbang NOT pada input selektor (S0, S1, S2) untuk menghasilkan sinyal pembalik selektor.
5. Setiap gerbang AND dikombinasikan dengan inputan data dan inputan selektor yang sesuai dengan kebutuhan.
6. Hubungkan output dari 8 gerbang AND ke 8 LED sebagai keluaran D0 sampai D7.
7. Membuat pengujian pada input selektor dengan mengubah nilainya untuk memastikan hanya satu output yang menyala sesuai kombinasi selektor.

3.1.3 Rangkaian gabungan MUX dan DEMUX

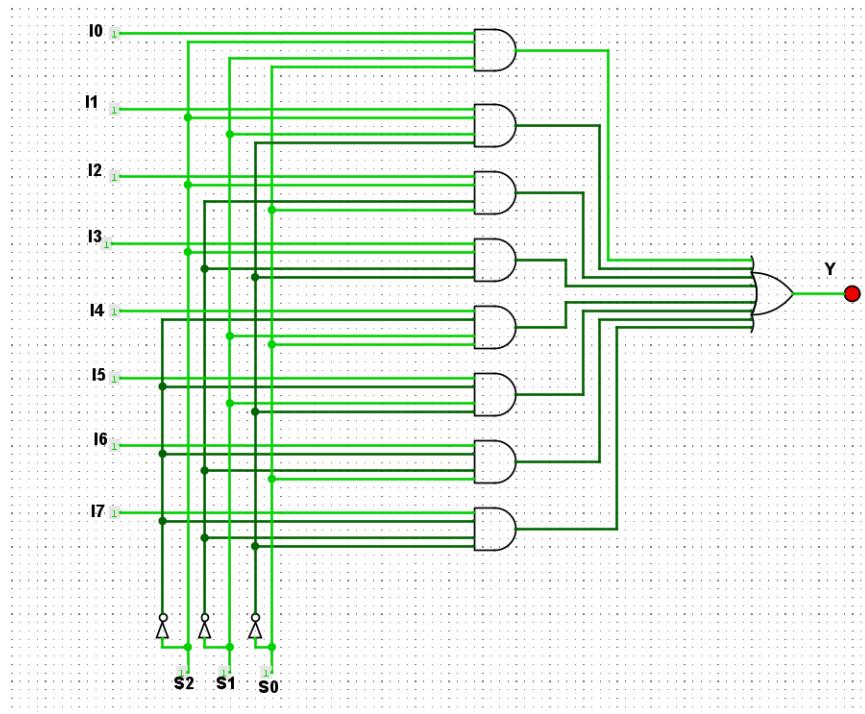
1. Tambahkan rangkaian Multiplexer (8-to-1 MUX) dan rangkaian Demultiplexer (1-to-8 DEMUX) pada area kerja.
2. Tarik output Y menjadi satu jalur dan hubungkan ke input data DEMUX (E) sebagai input rangkaian DEMUX.
3. Lakukan pengujian dengan mengubah kombinasi selektor di MUX dan DEMUX, dan lihatlah hasil outputnya pada LED.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Pengujian Rangkaian Multiplexer (MUX)

Setelah melakukan langkah-langkah diatas, berikut adalah hasil 8-to-1 MUX



Gambar 4.1. Gerbang Logika 8-to-1 Multiplexer

Tabel 4.1 Truth Table 8-to-1 MUX

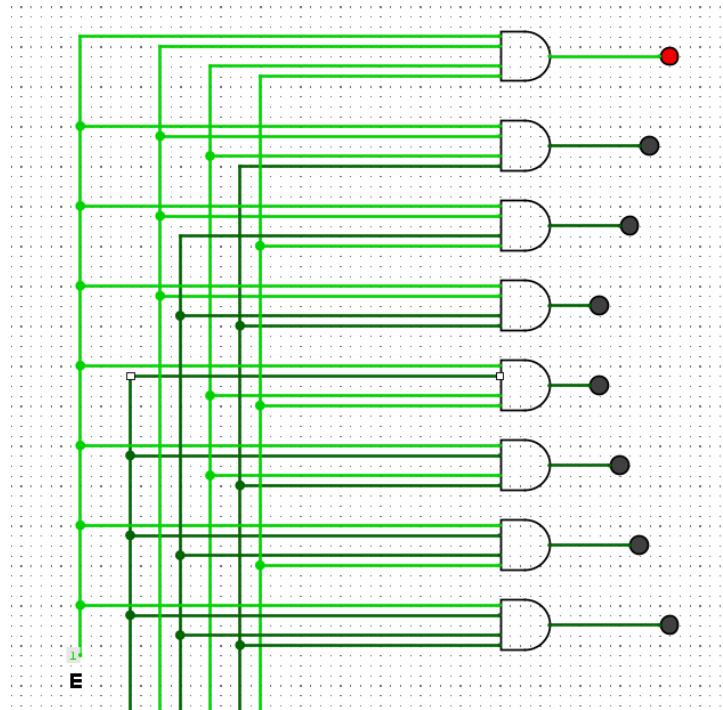
S2	S1	S0	Output (Y)
0	0	0	I0
0	0	1	I1
0	1	0	I2
0	1	1	I3
1	0	0	I4
1	0	1	I5
1	1	0	I6
1	1	1	I7

berfungsi sebagai penghubung salah satu dari 8 input data (D0 hingga D7) menuju ke satu output tunggal (Y). Tahapan pemilihan ini dilaksanakan melalui tiga sinyal selektor (S0, S1, S2), pada ketiga sinyal tersebut akan membentuk kombinasi biner yang dapat menentukan input mana yang akan aktif. Setiap input (D0-D7) terhubung ke kombinasi gerbang logika bersama selektor dan keluaran dari NOT. Kombinasi ini memperlihatkan bahwa satu jalur data yang aktif pada waktu tetentu. Ketika selektor diubah, jalur yang aktif akan berpindah ke input lain sesuai nilai binernya.

Seperti, selektor yang memiliki nilai sebesar $S2 = 0, S1 = 1, S0 = 0$, maka input yang akan diteruskan ke output adalah D2. Jika selektor berubah menjadi 1 0 1, maka yang diteruskan adalah D5. Menggunakan cara ini, MUX dapat memilih data tertentu tanpa perlu banyak jalur.

Dengan ini, temuan dari simulasi menunjukkan bahwa rangkaian multiplexer berhasil menjalankan fungsinya sebagai pemilih satu input dari delapan input secara akurat. Output selalu sesuai dengan kombinasi selektor dan tidak terjadi konflik sinyal selama pengujian

4.2 Hasil Pengujian Rangkaian Demultiplexer(DEMUX)



Gambar 4.2 Gerbang Logika 1-to-8 Demultiplexer

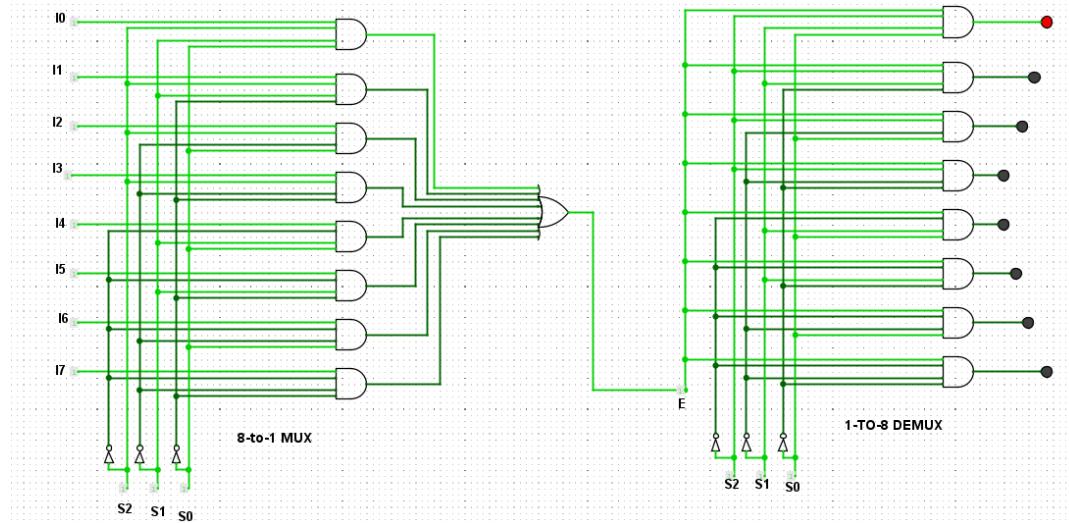
Tabel 4.2 Truth Table Demultiplexer

D	S2	S1	S0	Output aktif
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

ada 1 input data, 3 selektor (S2, S1, S0), dan 8 output AND. Data input akan dikirim ke salah satu output sesuai kombinasi selektor. Ketika S2 = 0, S1 = 0, S0 = 0, sinyal dari input hanya muncul pada Y0, sedangkan output lainnya tidak aktif. Ketika kombinasi selektor diubah menjadi 001, sinyal berpindah ke Y1, dan Y7 akan mengaktifkan jika nilai 111.

Hal tersebut memperlihatkan hanya satu gerbang AND yang aktif untuk setiap kombinasi selektor, sehingga jalur data tidak pernah bercabang ke lebih dari satu output secara bersamaan. Hasil yang ditunjukkan terlihat bahwa selektor bekerja dengan benar sebagai penentu arah data. NOT gate juga berperan penting untuk membentuk pola kombinasi selektor agar masing-masing output hanya aktif pada kondisi tertentu. Secara menyeluruh, hasil simulasi menunjukkan bahwa DEMUX menjalankan fungsi distribusi data dengan tepat, yaitu satu input dapat diarahkan ke delapan output berbeda secara selektif tanpa adanya tabrakan sinyal.

4.3 Hasil Pengujian Rangkaian Gabungan MUX & DEMUX



Tabel 4.3 Gerbang Logika Gabungan MUX & DEMUX

Tabel 4.3 Truth Table MUX & DEMUX

S2	S1	S0	Input aktif MUX	Output akhir DEMUX
0	0	0	D0	Y0
0	0	1	D1	Y1
0	1	0	D2	Y2
0	1	1	D3	Y3
1	0	0	D4	Y4
1	0	1	D5	Y5
1	1	0	D6	Y6
1	1	1	D7	Y7

MUX berfungsi untuk memilih salah satu dari beberapa input berdasarkan sinyal kontrol (select) dan melanjutkan ke output tunggal. Sebaliknya, DEMUX berfungsi untuk menerima satu input dan mengarahkan sinyal tersebut ke salah satu dari beberapa output, juga mengacu pada sinyal kontrol. Ketika kedua komponen ini digabungkan dalam satu rangkaian, proses pengiriman dan distribusi data dapat dilakukan secara efisien tanpa perlu banyak jalur kabel.

Penggabungan MUX dan DEMUX ini sangat bermanfaat saat dibutuhkan komunikasi satu ke banyak atau banyak ke satu tanpa penggunaan banyak kabel fisik. Dengan proses ini, sistem lebih hemat biaya dan ringkas, serta mengurangi kompleksitas rangkaian. Selain itu, konfigurasi ini banyak digunakan dalam sistem transmisi data, jaringan komunikasi, dan perangkat kontrol digital untuk mengirimkan data dari beberapa sensor ke beberapa aktuator melalui satu jalur komunikasi.

KESIMPULAN

Berdasarkan proses perancangan yang telah dilakukan, dapat disimpulkan bahwa :

- Rangkaian MUX mampu memilih 1 sinyal masukan dari 8 sumber berdasarkan kombinasi nilai selektor (S2, S1, S0). Hasil pengujian menunjukkan bahwa sinyal keluaran selalu sesuai dengan input yang dipilih, sehingga membuktikan fungsi pemilihan data berjalan dengan benar.
- Rangkaian DEMUX mampu mengirim 1 sinyal masukan ke salah satu dari 8 keluaran sesuai dengan kombinasi nilai selektor. LED yang menyala selalu berada pada posisi yang tepat sesuai sinyal kontrol, menunjukkan bahwa distribusi data bekerja sesuai dengan teori.
- Pada rangkaian gabungan MUX-DEMUX, proses pengiriman dan penerusan data melalui satu jalur berjalan dengan baik. MUX terlebih dahulu memilih input yang aktif, kemudian DEMUX mengarahkan sinyal tersebut ke output yang sesuai. Pengujian menunjukkan output yang menyala pada DEMUX selalu sesuai dengan input yang terpilih pada MUX, sehingga aliran data tidak mengalami gangguan.
- Penggabungan MUX dan DEMUX membuat jalur transmisi lebih efisien karena hanya membutuhkan satu kabel utama untuk mengirimkan data dari banyak input ke banyak output. Sistem ini lebih hemat jalur, terstruktur, dan mudah dianalisis dibandingkan dengan koneksi langsung antara setiap input dan output.