

LAPORAN PRAKTIKUM PENGANTAR SISTEM DIGITAL

Dosen: Arif Rifai Dwiyanto, ST., MTI

MULTIPLEXER (MUX) & DEMULTIPLEXER (DEMUX)



Disusun Oleh:

Khoirunnisa

(202410715047)

F3A6

PROGRAM STUDI INFORMATIKA FAKULTAS ILMU KOMPUTER

UNIVERSITAS BHAYANGKARA JARAKARTA RAYA

2025

BAB I

PENDAHULUAN

1.1 Tujuan

1. Mahasiswa memahami konsep kerja Multiplexer (MUX) dan Demultiplexer (DEMUX)
2. Mahasiswa mampu merancang dan mensimulasikan MUX dan DEMUX menggunakan Logism.
3. Menerapkan gerbang logika dasar (AND, OR, NOT) untuk merancang MUX-DEMUX.
4. Mensimulasikan rangkaian gabungan MUX dan DEMux pada Logism

1.2 Dasar Teori

1.2.1 Merancang 8-to-1 Multiplexer

- Multiplexer (MUX) Adalah perangkat yang memungkinkan informasi digital dari beberapa input dapat dialihkan kedalam satu jalur output untuk menuju tujuan bersama.
- N-to-1 Multiplexer: Rangkaian yang memiliki N input, $\log_2(N)$ select lines, dan 1 output.
- Contoh: 8-to-1 Multiplexer
 - Input : $I_0, I_1, I_2, I_3, I_4, I_5, I_6, I_7$.
 - Select Lines : S_0, S_1 , dan S_2 .
 - Output : Y
- Ekspresi logika output:
$$Y = (\overline{S_2} \cdot \overline{S_1} \cdot \overline{S_0} \cdot I_0) + (\overline{S_2} \cdot \overline{S_1} \cdot S_0 \cdot I_1) + (\overline{S_2} \cdot S_1 \cdot \overline{S_0} \cdot I_2) + (\overline{S_2} \cdot S_1 \cdot S_0 \cdot I_3) + (S_2 \cdot \overline{S_1} \cdot \overline{S_0} \cdot I_4) + (S_2 \cdot \overline{S_1} \cdot S_0 \cdot I_5) + (S_2 \cdot S_1 \cdot \overline{S_0} \cdot I_6) + (S_2 \cdot S_1 \cdot S_0 \cdot I_7)$$
- Tabel Kebenaran

INPUT			OUTPUT
S2	S1	S0	Y
0	0	0	I_0
0	0	1	I_1

0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

1.2.2 Merancang 1-to-8 Demultiplexer:

- Demultiplexer (DEMUX) Adalah rangkaian logika kombinasional kabel maupun yang menggunakan media udara seperti wireless atau radio.
- N-to-1 Multiplexer: Rangkaian yang memiliki 1 input, $\log_2 N$ select lines, dan N output.
- Contoh: 1-to-8 Demultiplexer
 - Input : 1
 - Select Lines : S2, S1, dan S0.
 - Output : O0,O1,O2,O3.

- Ekspresi logika output:

$$Y_0 = D \cdot \overline{S_0} \cdot \overline{S_1} \cdot \overline{S_2}$$

$$Y_1 = D \cdot \overline{S_0} \cdot \overline{S_1} \cdot S_2$$

$$Y_2 = D \cdot \overline{S_0} \cdot S_1 \cdot \overline{S_2}$$

$$Y_3 = D \cdot \overline{S_0} \cdot S_1 \cdot S_2$$

$$Y_4 = D \cdot S_0 \cdot \overline{S_1} \cdot \overline{S_2}$$

$$Y_5 = D \cdot S_0 \cdot \overline{S_1} \cdot S_2$$

$$Y_6 = D \cdot S_0 \cdot S_1 \cdot \overline{S_2}$$

$$Y_7 = D \cdot S_0 \cdot S_1 \cdot S_2$$

- Tabel Kebenaran

SELECT LINES			DATA INPUT (D)	OUTPUT							
S2	S1	S0		Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
0	0	0	1	1	0	0	0	0	0	0	0
0	0	1	1	0	1	0	0	0	0	0	0
0	1	0	1	0	0	1	0	0	0	0	0
0	1	1	1	0	0	0	1	0	0	0	0

1	0	0	1	0	0	0	0	1	0	0	0
1	0	1	1	0	0	0	0	0	1	0	0
1	1	0	1	0	0	0	0	0	0	1	0
1	1	1	1	0	0	0	0	0	0	0	1

1.3 Alat dan Bahan:

1. Laptop/PC dengan aplikasi Logisim Evolution terinstal
2. Modul panduan praktikum
3. Buku catatan

1.4 Langkah-langkah Kerja:

1.4.1 Merancang 8-to-1 Multiplexer

1. Tambahkan 8 Constant untuk input $I_0, I_1, I_2, I_3, I_4, I_5, I_6, I_7$
2. Tambahkan 3 Constant untuk select lines S_2, S_1 dan S_0
3. Gunakan kombinasi gerbang AND, OR, dan NOT sesuai ekspresi logika.
4. Tambahkan LED untuk melihat hasil output Y.
5. Simulasikan rangkaian dengan berbagai kombinasi input dan select lines

1.4.2 Merancang 1-to-8 Demultiplexer:

1. Tambahkan 1 Constant untuk input 1
2. Tambahkan 3 Constant untuk select lines S_2, S_1 dan S_0
3. Gunakan kombinasi gerbang AND, OR, dan NOT sesuai ekspresi logika.
4. Tambahkan LED untuk melihat hasil output Y.
5. Simulasikan rangkaian dengan berbagai kombinasi lines dan input I.

1.4.3 Implementasi Gabungan Multiplexer-Demultiplexer

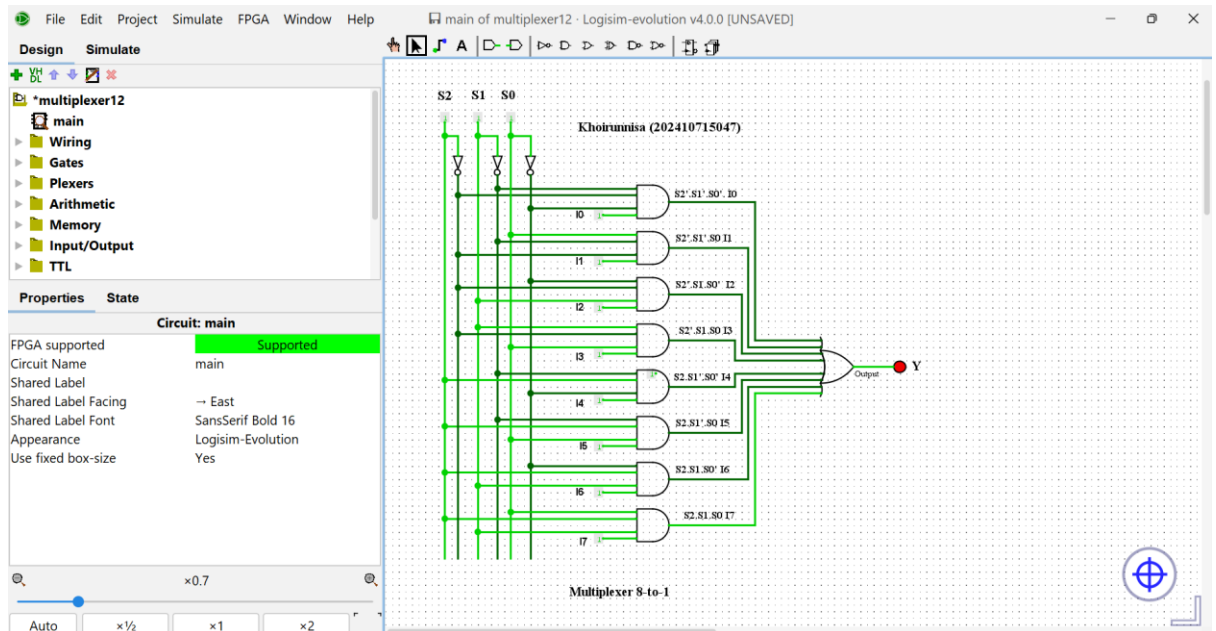
1. Buat MUX 8 x 1 sesuai Langkah sebelumnya.
2. Buat DEMUX 1 x 8.
3. Hubungkan output Y (MUX) ke input D (DEMUX)
4. Gunakan select line yang sama untuk MUX dan DEMUX.
5. Tambahkan switch sebagai input.
6. Simulasikan untuk melihat apakah output O_0-O_7 sesuai select line

BAB II

HASIL SIMULASI

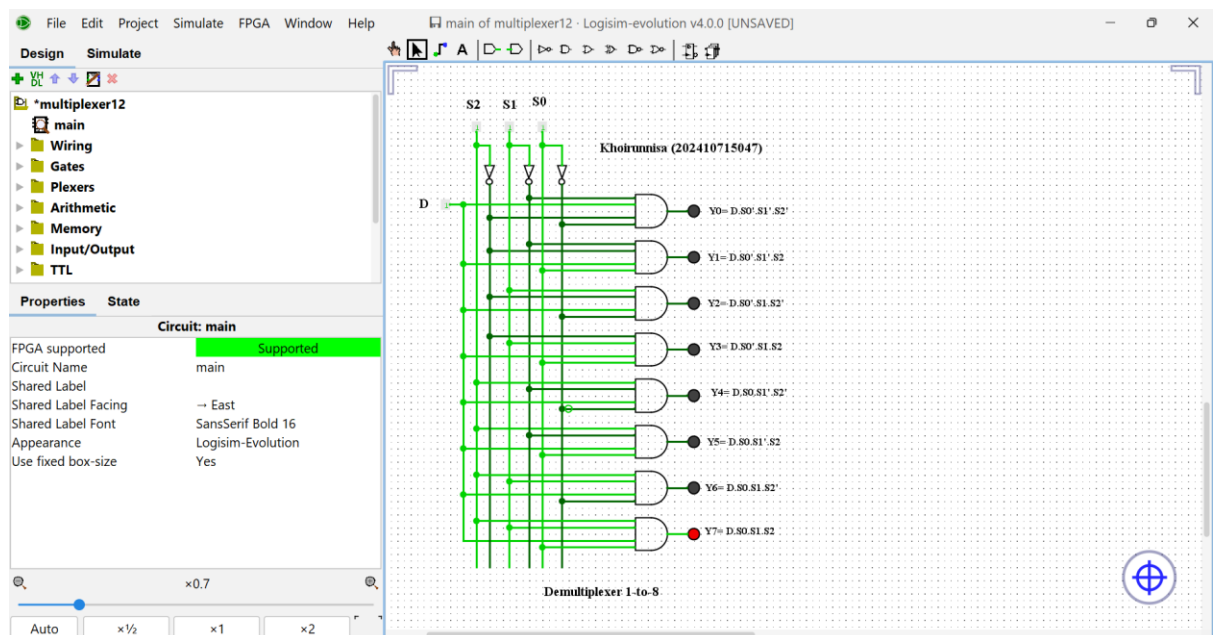
3.1 Hasil Implementasi Multiplexer

Hasil implementasi Multiplexer 8-to-1 menggunakan aplikasi Logism



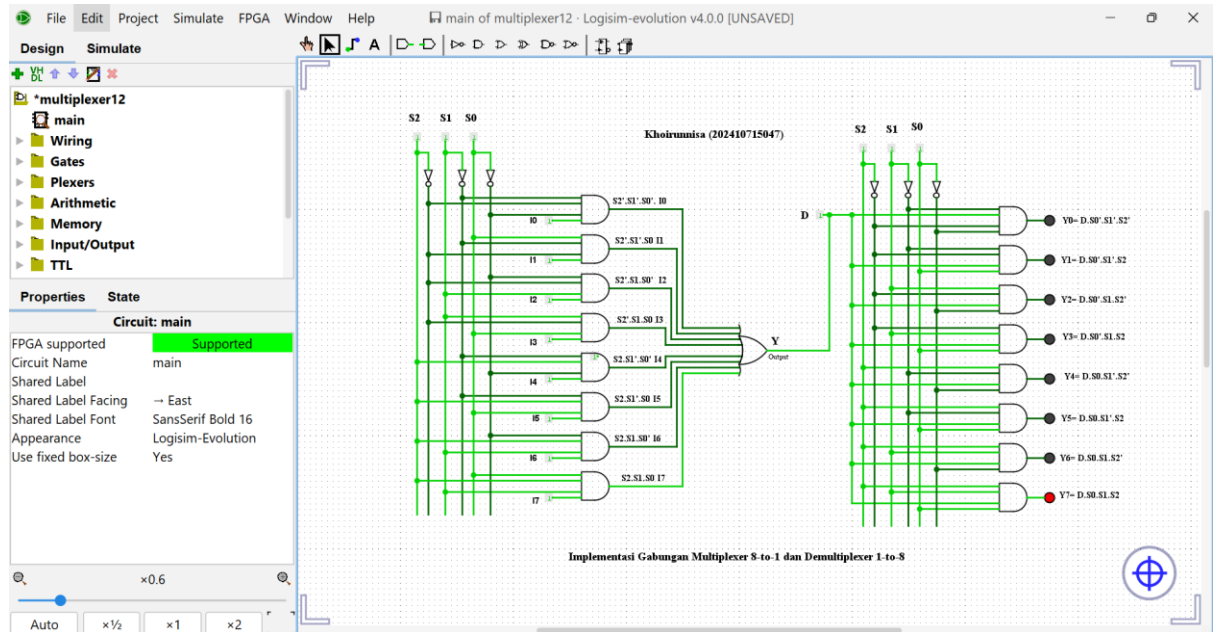
3.2 Hasil Implementasi Demultiplexer

Hasil implementasi Demultiplexer 1-to-8 menggunakan aplikasi Logism



3.3 Hasil Implementasi Gabungan Multiplexer-Demultiplexer

Hasil implementasi gabungan Multiplexer 8-to-1 dan Demultiplexer 1-to-8 menggunakan aplikasi Logism



BAB III

KESIMPULAN

Berdasarkan hasil praktikum, dapat disimpulkan bahwa Multiplexer (MUX) dan Demultiplexer (DEMUX) merupakan rangkaian logika yang memiliki fungsi saling berkebalikan dalam pengolahan sinyal digital.

Multiplexer (MUX) berfungsi untuk memilih satu dari beberapa input dan meneruskannya ke satu output tunggal. Pemilihan input dilakukan oleh kombinasi sinyal kontrol (*select lines*). Dengan demikian, MUX bekerja sebagai rangkaian many-to-one, yang menggabungkan beberapa sinyal menjadi satu jalur keluaran.

Sebaliknya, Demultiplexer (DEMUX) berfungsi untuk mengarahkan satu input ke salah satu dari beberapa output berdasarkan kombinasi *select lines*. DEMUX bekerja sebagai rangkaian one-to-many, yang memecah satu sinyal masuk menjadi jalur keluaran sesuai pilihan.

Selain itu, ketika MUX dan DEMUX digabungkan, keduanya dapat bekerja sebagai sistem lengkap untuk pengiriman sinyal digital. MUX digunakan untuk menggabungkan banyak sinyal menjadi satu sinyal tunggal, kemudian DEMUX digunakan untuk memecah kembali sinyal tunggal tersebut menjadi sinyal-sinyal aslinya. Hal ini menunjukkan bahwa pasangan MUX–DEMUX mampu mengirimkan beberapa data melalui satu jalur komunikasi dengan tetap menjaga struktur data yang diterima.

Melalui praktikum ini, mahasiswa memahami hubungan kerja MUX dan DEMUX, mekanisme *select lines*, serta aplikasi keduanya dalam sistem digital seperti komunikasi data, pengendalian jalur sinyal, dan multiplexing pada perangkat elektronik.