

PRAKTIKUM PENGANTAR SISTEM DIGITAL

Dosen: Arif Rifa'i Dwiyanto, ST., MTI

IMPLEMENTASI K-MAP DAN DECODER & ENCODER



Disusun oleh:

Pingky Nanda Rahmadhani

202410715036

F3A6

**LAPORAN PRAKTIKUM PENGANTAR SISTEM
DIGITAL
PROGRAM STUDI INFORMATIKA – FAKULTAS ILMU
KOMPUTER
UNIVERSITAS BHAYANGKARA JAKARTA RAYA
2025**

BAB I

PENDAHULUAN

1.1 Tujuan

- Mahasiswa memahami konsep tabel kebenaran
- Mahasiswa mampu menyederhanakan ekspresi logika menggunakan Karnaugh Map (K-Map)
- Mahasiswa dapat mengimplementasikan hasil penyederhanaan pada Logisim
- Mahasiswa memahami konsep encoder dan decoder
- Mahasiswa mampu merancang dan mensimulasikan rangkaian encoder 8-to-3 dan decoder 3-to-8 menggunakan aplikasi Logisim.

1.2 Alat dan Bahan

- Laptop/PC dengan aplikasi Logisim Evolution terinstal
- Modul panduan praktikum
- Buku catatan

1.3 Langkah-Langkah Kerja

A. Implementasi K-Map

Diketahui peta Karnaugh berikut:

AB\CD	00	01	11	10
00	0	0	0	0
01	1	1	1	0
11	1	1	1	0
10	0	0	0	0

a. Langkah 1: Tentukan Minterm (kombinasi bernilai 1)

Nilai 1 muncul pada posisi:

- AB = 01, CD = 00, 01, 11
- AB = 11, CD = 00, 01, 11

Minterm: m(4), m(5), m(7), m(12), m(13), m(15)

Fungsi logika awalnya yaitu:

$$Y = (A'BC'D') + (A'BC'D) + (A'BCD) + (ABC'D') + (ABC'D) + (ABCD)$$

Berikut tabel kebenaran pada fungsi logika tersebut

[illegible]

b. Langkah 2: Penyederhanaan dengan K-Map

Jika dikelompokkan pada peta Karnaugh diperoleh:

AB\CD	00	01	11	10
00	0	0	0	0
01	1	1	1	0
11	1	1	1	0
10	0	0	0	0

- Kelompok 4 pertama (AB=01 dan 11, CD = 00 dan 01) → BC'

- Kelompok 2 kedua (AB = 01 dan 11, CD = 11) → BCD

Sehingga diperoleh :

$$Y = BC' + BCD$$

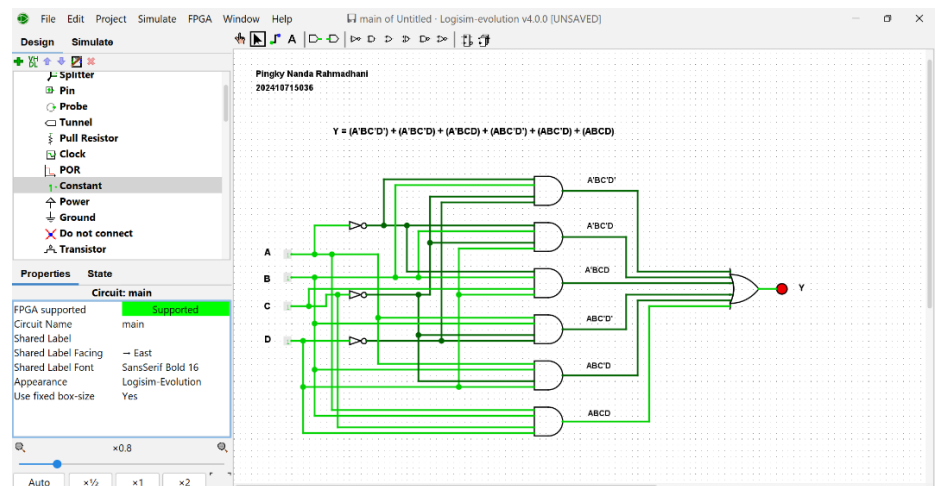
$$Y = B(C' + CD)$$

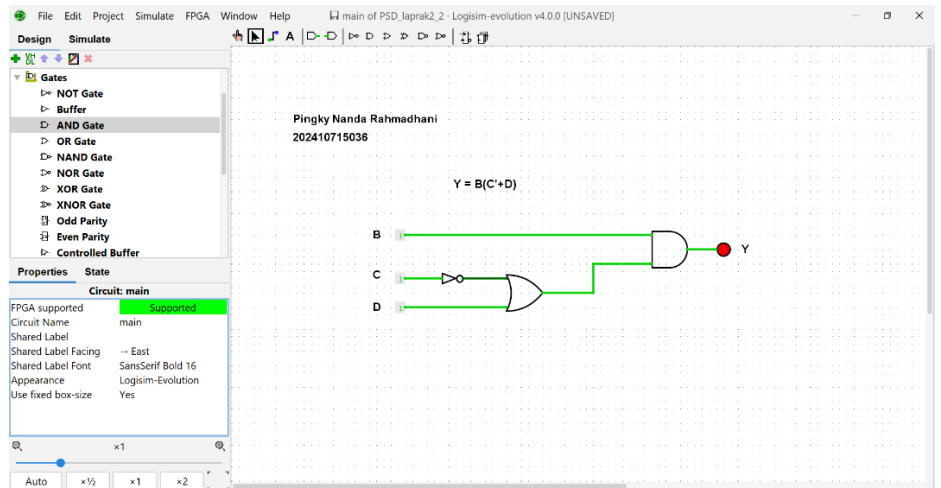
$$Y = B(C' + D)$$

Maka hasil penyederhanaanya yaitu $Y = B(C' + D)$

c. Langkah 3: Membuat rangkaian simulasi pada Logisim Evolution

Berikut rangkaian gerbang logika sebelum disederhanakan dan sesudah disederhanakan





d. Langkah 4: Bandingkan rangkaian sebelum dan sesudah disederhanakan

Aspek	Rangkaian awal	Rangkaian setelah disederhanakan
Ekspresi logika	$Y = (A'BC'D') + (A'BC'D) + (A'BCD) + (ABC'D') + (ABC'D) + (ABCD)$	$Y = B(C' + D)$
Jumlah Gerbang AND	6 buah	1 buah
Jumlah Gerbang OR	1 buah	1 buah
Jumlah Gerbang NOT	3 buah	1 buah
Total Komponen	10 gerbang	3 gerbang
Output Simulasi	sama	sama

B. Encoder dan Decoder

1) Merancang 8-to-3 Encoder

Langkah 1: Buat Tabel Kebenaran untuk Encoder 8-to-3

Bilangan Oktal	A	B	C
	2^2	2^1	2^0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

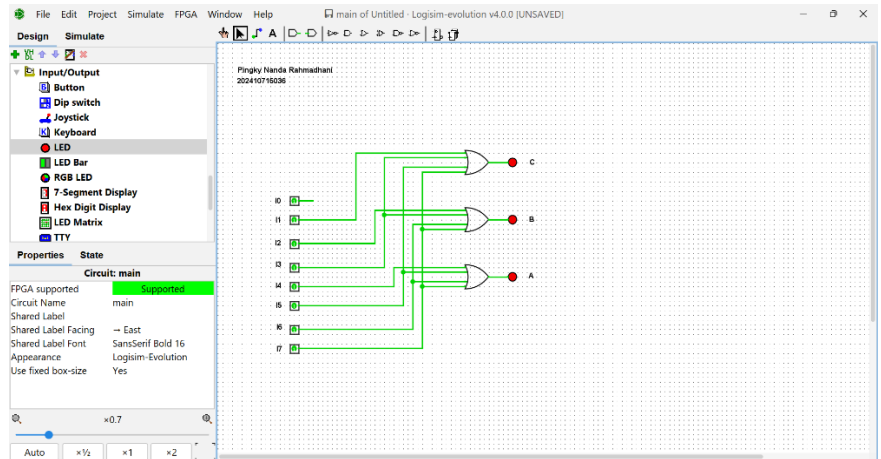
Input								Output		
I ₀	I ₁	I ₂	I ₃	I ₄	I ₅	I ₆	I ₇	A	B	C
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

$$A = I_4 + I_5 + I_6 + I_7$$

$$B = I_2 + I_3 + I_6 + I_7$$

$$C = I_1 + I_3 + I_5 + I_7$$

Langkah 2: Implementasikan rangkaian menggunakan Logisim

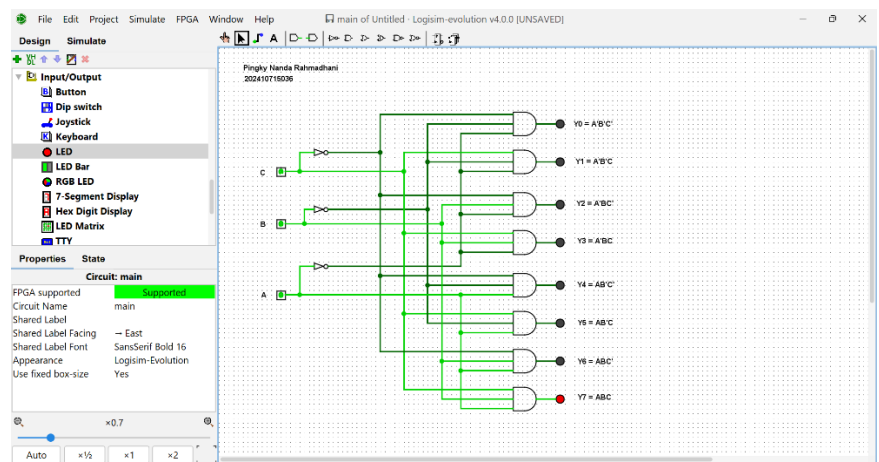


2) Merancang 3-to-8 Decoder

Langkah 1: Buat Tabel Kebenaran untuk Decoder 3-to-8

Input			Output							
A	B	C	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

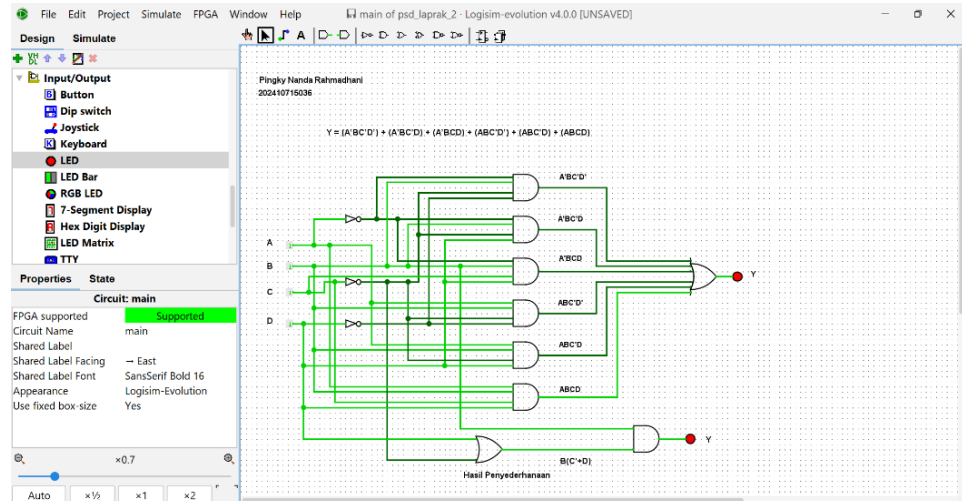
Langkah 2: Implementasikan rangkaian menggunakan Logisim



BAB II

HASIL SIMULASI DAN PEMBAHASAN

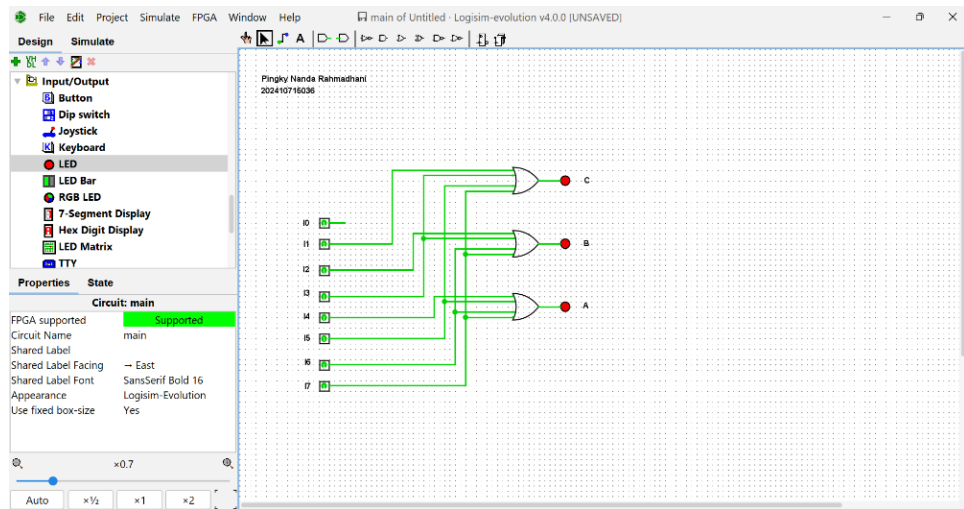
a) K-Map



Pada percobaan ini, dilakukan perancangan dan simulasi rangkaian logika menggunakan empat variabel input yaitu A, B, C dan D. Diperoleh fungsi logika awal yang disusun dalam bentuk Sum Of Product (SOP) yang cukup panjang, karena setiap minterm dengan output 1 direpresentasikan oleh kombinasi gerbang AND yang berbeda. Fungsi logika awal yang diimplementasikan adalah $Y = (A'BC'D') + (A'BC'D) + (A'BCD) + (ABC'D') + (ABC'D) + (ABCD)$. Rangkaian awal tersebut terlihat cukup kompleks, memiliki banyak sambungan dan gerbang logika.

Karena rangkaian tersebut memerlukan banyak komponen dan cukup rumit, dilakukan proses penyederhanaan menggunakan peta Karnaugh. Melalui penyederhanaan ini diperoleh bentuk fungsi logika yang lebih efisien tanpa mengubah hasil keluaran. Setelah disederhanakan, jumlah gerbang logika yang digunakan menjadi lebih sedikit sehingga rangkaian menjadi lebih sederhana. Fungsi logika hasil penyederhanaannya yaitu $Y = B(C'+D)$. Terlihat pada gambar rangkaian hasil penyederhanaan menunjukkan bahwa output Y tetap sama yaitu bernilai 1 (menyala) pada kondisi yang sama seperti rangkaian awal. Perbedaannya hanya terletak pada jumlah dan susunan gerbang logika.

b) Encoder



Pada percobaan encoder, dilakukan perancangan rangkaian encoder 8-to-3 menggunakan aplikasi logisim. Encoder merupakan rangkaian logika kombinasi yang berfungsi untuk mengubah sejumlah besar input menjadi kombinasi biner output yang lebih sedikit. Pada gambar menunjukkan hasil implementasi encoder 8-to-3 dengan 8 input (I_0 - I_7) dan 3 output (A,B dan C). Setiap input merepresentasikan nilai biner yang dikonversi menjadi bentuk 3 bit biner pada bagian output.

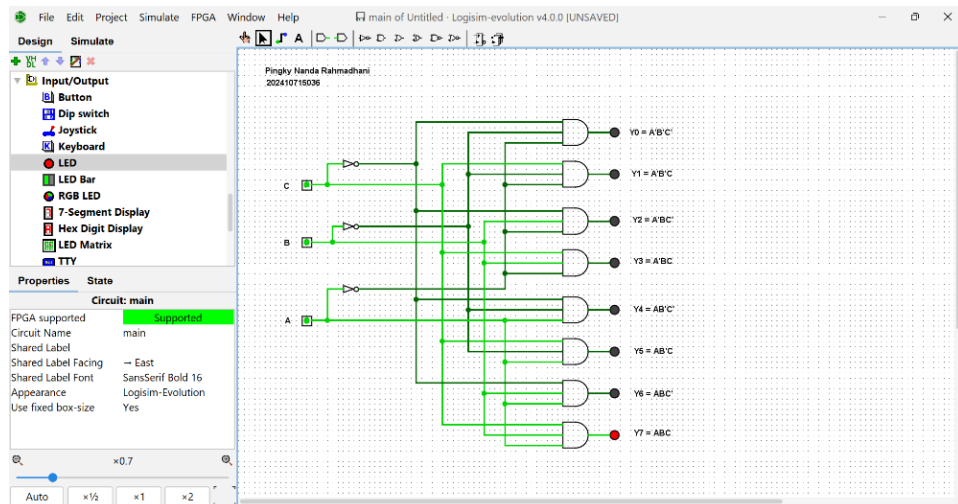
- Setiap input dihubungkan ke gerbang OR sesuai dengan persamaan logika yang terlihat pada gambar.
- Tiga output dihasilkan dari kombinasi logika OR yang merepresentasikan kode biner dari input yang aktif.
- Hasil pengujian menunjukkan bahwa ketika salah satu input diberi nilai logika 1, output menghasilkan kombinasi biner yang sesuai dengan posisi input tersebut.

Contoh: jika $I_5=1$, maka output menjadi ABC=101 (nilai biner 5)

Berdasarkan tabel kebenaran, diperoleh persamaan logika untuk masing-masing output:

- Output A akan bernilai 1 jika salah satu input dari I_4 hingga I_7 aktif
- Output B akan bernilai 1 jika salah satu input dari I_2, I_3, I_6, I_7 aktif
- Output C akan bernilai 1 jika salah satu input dari I_1, I_3, I_5, I_7 aktif

c) Decoder



Pada percobaan decoder, dilakukan perancangan rangkaian decoder 3-to-8. Decoder merupakan rangkaian logika kombinasi yang berfungsi untuk menerjemahkan kode biner pada input menjadi satu output yang aktif dari beberapa keluaran. Dengan kata lain, decoder adalah kebalikan dari encoder. Rangkaian decoder 3-to-8 memiliki 3 input yaitu A, B dan C yang masing-masing terhubung ke gerbang NOT untuk menghasilkan sinyal komplemen (A' , B' , C'). Delapan output yaitu Y_0 - Y_7 , diperoleh melalui gerbang AND yang mengkombinasikan sinyal input dan sinyal komplemennya sesuai persamaan logika. Setiap output dihubungkan ke LED untuk menampilkan hasil bahwa hanya satu output yang aktif setiap kali kombinasi input berubah.

Contoh berdasarkan hasil simulasi:

- Jika input = 111 ($A=1$, $B=1$, $C=1$) maka output yang aktif adalah Y_7
- Jika input = 100 ($A=1$, $B=0$, $C=0$) maka output yang aktif adalah Y_4
- Jika input = 000 ($A=0$, $B=0$, $C=0$) maka output yang aktif adalah Y_0

Hal ini sesuai dengan tabel kebenaran yang telah dibuat.

BAB III

KESIMPULAN

Berdasarkan praktikum yang telah dilakukan, yaitu implementasi Peta Karnaugh (K-Map) serta perancangan encoder dan decoder, dapat disimpulkan bahwa penggunaan Peta Karnaugh sangat efektif dalam menyederhanakan fungsi logika menjadi bentuk yang lebih efisien. Dari proses penyederhanaan diperoleh fungsi $Y = B(C' + D)$ yang menghasilkan pengurangan jumlah gerbang logika dari 10 komponen menjadi 3 komponen.

Selain itu, perancangan encoder 8-to-3 dan decoder 3-to-8 menunjukkan penerapan rangkaian kombinasi. Encoder berfungsi mengubah sinyal input tunggal menjadi kode biner 3 bit, sedangkan decoder melakukan proses kebalikannya. Percobaan ini dilakukan dengan menggunakan aplikasi Logisim, membuktikan bahwa seluruh rangkaian bekerja sesuai teori dan tabel kebenarannya.