

# **LAPORAN PRAKTIKUM**

## **RANGKAIAN ADDER DAN COMPARATOR**

**Mata Kuliah**

**Pengantar Sistem Digital**

**Dosen: Arif R. Dwiyanto ST., MTL.**



**Disusun oleh:**

**Hanif Fauziyyah (202410715172)**

**F3A6**

**PROGRAM STUDI INFORMATIKA FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS BHAYANGKARA JAKARTA RAYA**

**2025**

# BAB I

## PENDAHULUAN

### 1.1 Tujuan Praktikum

1. Mahasiswa memahami konsep rangkaian kombinasi.
2. Mahasiswa mampu membangun rangkaian 8-bit Adder menggunakan Logisim.
3. Mahasiswa mampu membangun rangkaian Comparator menggunakan logisim.
4. Menguji kebenaran fungsi dengan membandingkan hasil penjumlahan input terhadap output yang dihasilkan.

### 1.2 Langkah-langkah kerja

1. Membuat 8 bit adder

Rangkaian adder ini memiliki total 16 bit input, menghasilkan kombinasi masukan sebesar  $2^{16}$  atau 65.536 baris pada tabel kebenaran.

Tabel kebenaran:

Ai	Bi	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

**Bangun Rangkaian:** Gunakan ekspresi logika berikut<sup>21</sup>:

- **Rumus Umum:**

$$Si = Ai \oplus Bi \oplus Ci$$

$$Ci + 1 = (Ai \cdot Bi) + (Ci \cdot (Ai \oplus Bi))$$

**Ekspresi logika 8-bit adder:**

$$S0 = A0 \oplus B0 \oplus C0$$

$$C1 = (A0 \cdot B0) + (C0 \cdot (A0 \oplus B0))$$

$$S1 = A1 \oplus B1 \oplus C1$$

$$C2 = (A1 \cdot B1) + (C1 \cdot (A1 \oplus B1))$$

$$S2 = A2 \oplus B2 \oplus C2$$

$$C3 = (A2 \cdot B2) + (C2 \cdot (A2 \oplus B2))$$

$$S3 = A3 \oplus B3 \oplus C3$$

$$\begin{aligned}
C4 &= (A3 \cdot B3) + (C3 \cdot (A3 \oplus B3)) \\
S4 &= A4 \oplus B4 \oplus C4 \\
C5 &= (A4 \cdot B4) + (C4 \cdot (A4 \oplus B4)) \\
S5 &= A5 \oplus B5 \oplus C5 \\
C6 &= (A5 \cdot B5) + (C5 \cdot (A5 \oplus B5)) \\
S6 &= A6 \oplus B6 \oplus C6 \\
C7 &= (A6 \cdot B6) + (C6 \cdot (A6 \oplus B6)) \\
S7 &= A7 \oplus B7 \oplus C7 \\
C8 &= (A7 \cdot B7) + (C7 \cdot (A7 \oplus B7)) \rightarrow \text{Carry akhir (Cout)}
\end{aligned}$$

#### Implementasi:

1. Ekspresi logika menggunakan operasi XOR untuk \$Sum\$ dan kombinasi AND-OR untuk Carry.
2. Perhitungan dilakukan berantai dari bit paling rendah hingga bit paling tinggi, di mana hasil Carry sebuah bit menjadi Carry-in untuk bit berikutnya.
3. Menyusun rangkaian menggunakan delapan gerbang XOR dan delapan gerbang AND pada Logisim.
4. Hubungkan input A dan B menggunakan *Constant*
5. Tambahkan LED pada output Sum (S) dan Carry (C).

#### Comparator:

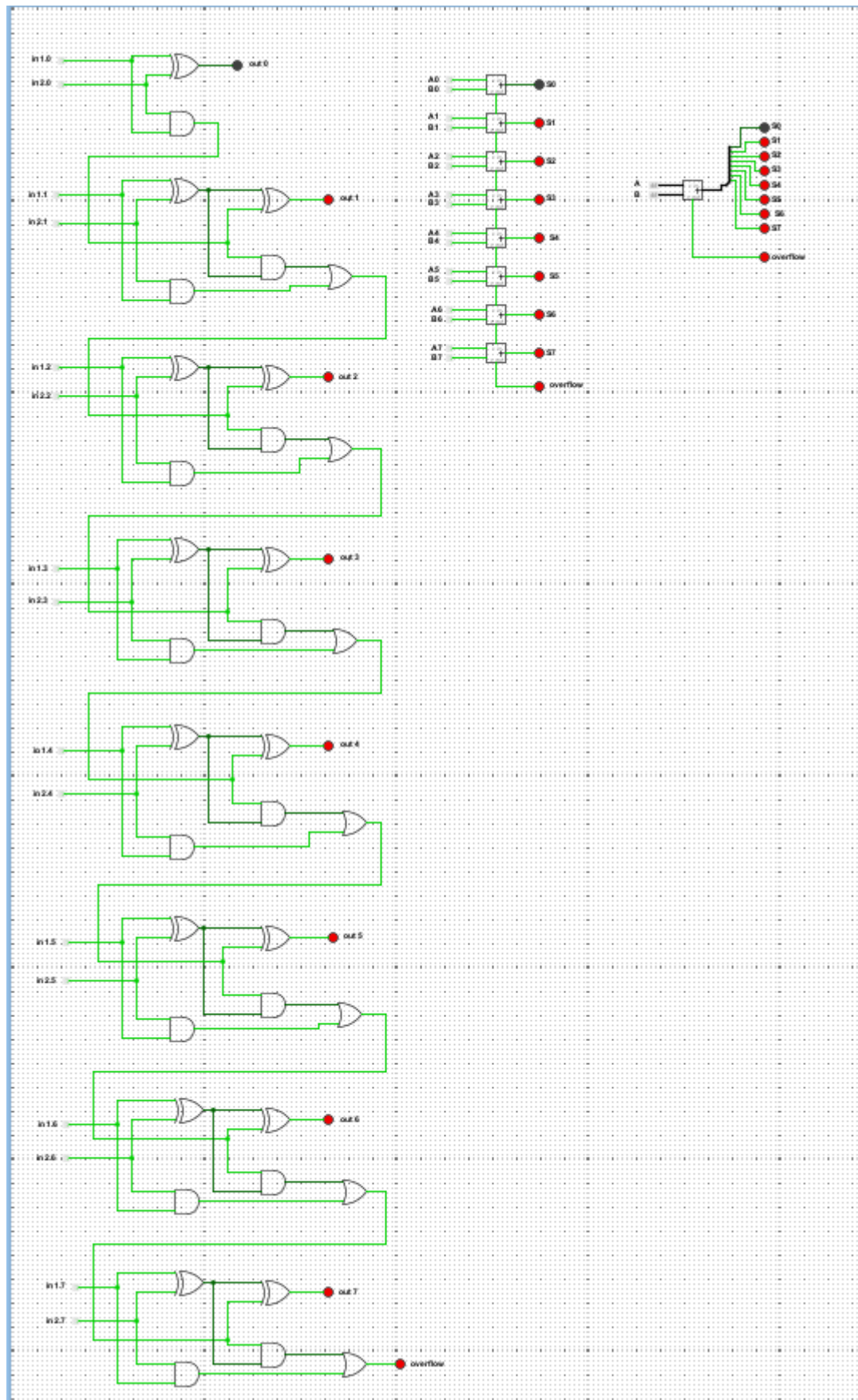
A	B	Output A > B	Output A = B	Output A < B
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

Pembuatan tabel kebenaran untuk rangkaian 8-bit comparator secara praktis tidak mungkin dilakukan karena kompleksitasnya yang sangat tinggi. Detail masukan kombinasi : Rangkaian comparator ini memiliki total **16 bit masukan**, yang terdiri dari 8 bit untuk input A dan 8 bit untuk input B. Jumlah masukan yang besar ini menghasilkan kombinasi masukan sebesar  $2^{16}$ , atau setara dengan **65.536 baris** pada tabel kebenaran.

- Menentukan kondisi perbandingan bit menggunakan tiga sinyal logika
  - $(A = B): Ei = (Ai \oplus Bi)'$
  - $(A > B): Gi = Ai \cdot Bi'$
  - $(A < B): Li = Ai' \cdot Bi$

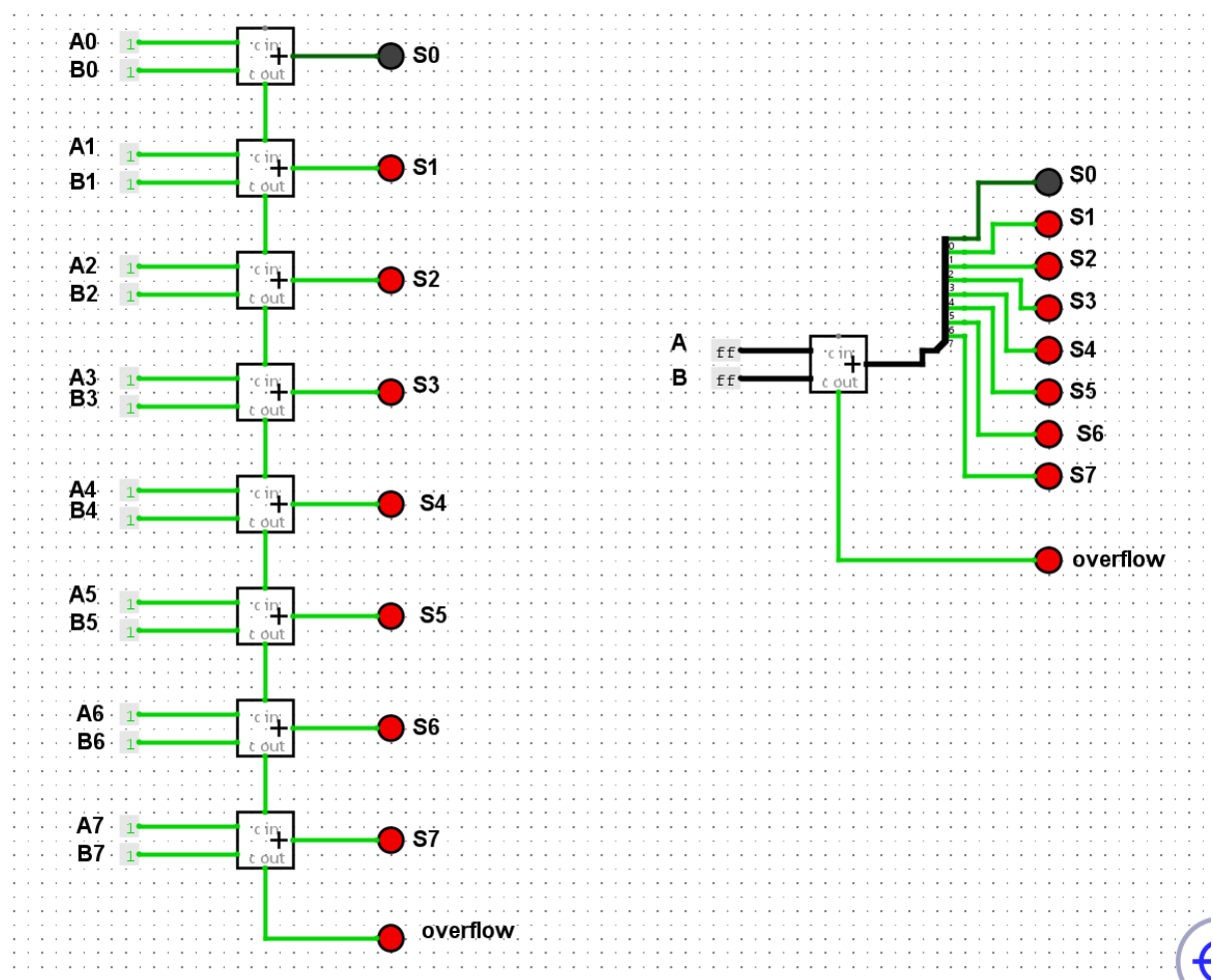
- Menyusun ekspresi logika
  - $A = B$ , akan bernilai benar jika semua bit bernilai sama  
 $(A = B) = E_7 \cdot E_6 \cdot E_5 \cdot E_4 \cdot E_3 \cdot E_2 \cdot E_1 \cdot E_0$
  - $A > B$ , akan bernilai benar jika pada suatu bit ke- $i$ ,  $A > B$  dan semua bit yang lebih signifikan bernilai sama  
 $A > B = G_7 + (E_7G_6) + (E_7E_6G_5) + (E_7E_6E_5G_4) + \dots + (E_7E_6E_5E_4E_3E_2E_1G_0)$
  - $A < B$ , akan bernilai benar jika pada suatu bit ke- $i$ ,  $A < B$  dan semua bit yang lebih signifikan bernilai sama  
 $A < B = L_7 + (E_7L_6) + (E_7E_6L_5) + (E_7E_6E_5L_4) + \dots + (E_7E_6E_5E_4E_3E_2E_1L_0)$
- Menggabungkan semua sinyal kesetaraan  
 Semua output kesetaraan ( $E_7$ - $E_0$ ) dikombinasikan secara berantai menggunakan gerbang AND
- Menentukan prioritas perbandingan. Jika terdapat bit ke- $i$  (dimulai dari MSB) dengan  $G_i = 1$ , maka hasil akhir adalah  $A > B$ , dan rangkaian berhenti mengevaluasi bit yang lebih rendah. Jika terdapat bit ke- $i$  dengan  $L_i = 1$ , maka hasil akhir adalah  $A < B$ .
- Menggabungkan logika hasil ( $G$ ,  $L$ ,  $E$ )
  - Sinyal  $A > B$  diperoleh dari kondisi  $G_7 + (E_7 \cdot G_6) + (E_7 \cdot E_6 \cdot G_5) + \dots + (E_7 \cdot E_6 \cdot \dots \cdot E_1 \cdot G_0)$
  - Sinyal  $A < B$  diperoleh dari kondisi sebaliknya
  - Sinyal  $A = B$  bernilai 1 jika seluruh  $E_i = 1$
- Meneruskan sinyal ke output akhir
  - EQ (Equal /  $A == B$ )
  - GT (Greater /  $A > B$ )
  - LT (Less /  $A < B$ )

### 1.3 Hasil Simulasi



Rangkaian penjumlahan biner 8-bit dibangun menggunakan gerbang logika dasar atau full-adder. Proses penjumlahan dilakukan mulai dari bit paling rendah (A0 dan B0), di mana setiap bit menghitung nilai sum dan menghasilkan carry-out. Nilai sum diperoleh melalui dua tahap operasi XOR: pertama  $A \oplus B$ , kemudian hasilnya di-XOR kembali dengan carry-in (Cin), sehingga diperoleh  $\text{Sum} = A0 \oplus B0 \oplus \text{Cin}$ . Sementara itu, carry-out (Cout) dihasilkan dari gabungan tiga gerbang AND yang kemudian di-OR-kan, yaitu  $\text{Cout} = (A_i \cdot B_i) + (C_i \cdot (A_i \oplus B_i))$ . Carry-out dari setiap full adder selanjutnya menjadi carry-in bagi full adder berikutnya, sehingga membentuk rangkaian ripple carry adder 8-bit. Hasil akhir penjumlahan ditampilkan menggunakan delapan LED, dan carry dari bit paling tinggi menjadi indikator terjadinya overflow.

### 1. Aritmatika



Pada rangkaian ini terdapat dua input utama, yaitu A dan B, yang masing-masing memiliki lebar 8 bit. Kedua masukan tersebut langsung dihubungkan ke modul adder sehingga proses penjumlahan biner dapat dilakukan otomatis oleh blok aritmatikanya. Adder kemudian menjumlahkan setiap bit mulai dari bit paling rendah hingga bit paling tinggi dengan tetap memperhitungkan carry-in pada setiap tahap. Hasil penjumlahan tersebut ditampilkan melalui delapan output S0 hingga S7 yang

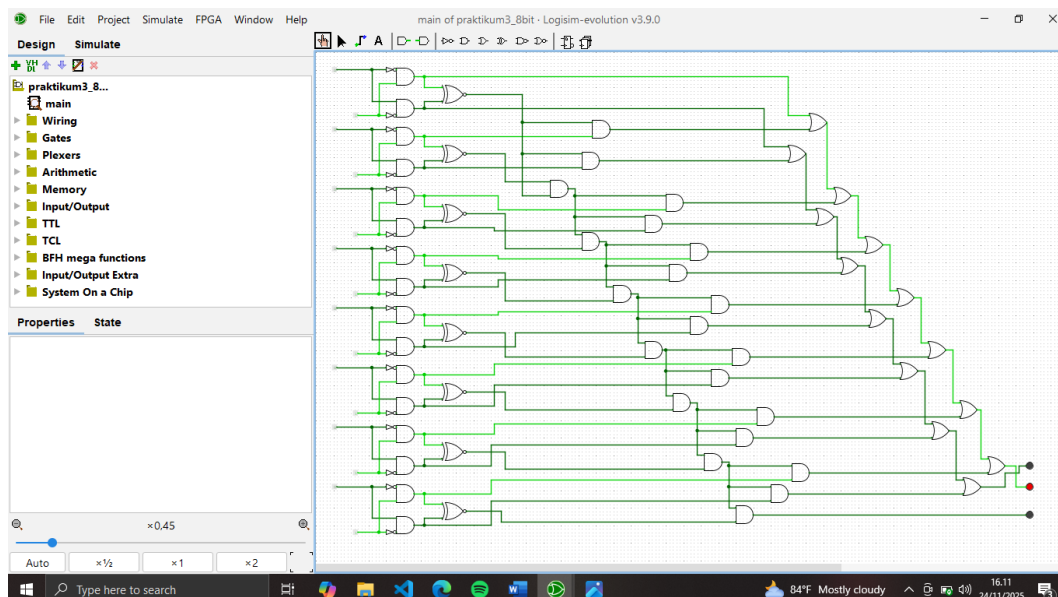
merepresentasikan nilai sum dari bit 0 sampai bit 7. Sementara itu, carry yang dihasilkan pada bit paling signifikan ditampilkan melalui LED overflow sebagai penanda bahwa hasil penjumlahan telah melampaui kapasitas 8 bit.

#### Tabel pengujian:

A	B	C-in	Sum	C-out
00000000	00000000	0	00000000	0
00001101	00000111	0	00010100	0
10101010	01010101	0	11111111	0
11111111	11111111	0	11111110	1
10011001	11001100	1	01100110	1

Pada pengujian pertama, baik input A maupun B bernilai 00000000 dan menghasilkan C-out = 0, yang menunjukkan tidak ada nilai yang dijumlahkan maupun carry yang muncul. Pada pengujian kedua, kombinasi acak 00001101 dan 00000111 dengan C-in bernilai 0 menghasilkan sum 00010100 dan C-out tetap 0, sesuai dengan operasi penjumlahan biner tanpa overflow. Pengujian ketiga memakai pasangan komplementer 10101010 dan 01010101 yang menghasilkan keluaran 11111111 dengan C-out 0, menandakan rangkaian mampu mengolah pola bit berlawanan dan tetap memberikan hasil yang benar. Pada pengujian keempat, kedua input diatur pada nilai maksimum 11111111 dan menghasilkan sum 11111110 dengan C-out bernilai 1, yang mengindikasikan bahwa hasil penjumlahan telah melampaui batas 8 bit sehingga memicu overflow. Sedangkan pada pengujian kelima, nilai 10011001 dan 11001100 dengan C-in 1 menghasilkan sum 01100110 serta C-out 1, membuktikan bahwa rangkaian dapat menangani carry-in eksternal secara tepat.

#### comparator:



Hasil praktikum menunjukkan bahwa comparator 8-bit bekerja dengan membandingkan dua bilangan biner A dan B secara berurutan mulai dari bit paling signifikan hingga bit paling rendah. Setiap pasangan bit diperiksa apakah nilainya sama, lebih besar, atau lebih kecil, dengan XNOR digunakan untuk mendeteksi kesetaraan. Sinyal kesetaraan ini diteruskan ke tahap berikutnya hingga ditemukan bit pertama yang berbeda. Kondisi  $A = B$  bernilai 1 jika seluruh bit identik, sedangkan  $A > B$  atau  $A < B$  ditentukan oleh bit pertama dari sisi MSB yang tidak sama. Rangkaian menghasilkan dua output utama, yaitu  $A > B$  dan  $A < B$ , dan jika keduanya 0 maka dapat dipastikan A sama dengan B.