

LAPORAN PRAKTIKUM

Mata Kuliah

Pengantar Sistem Digital

RANGKAIAN ADDER DAN COMPARATOR

Dosen: Arif R. Dwiyanto ST., MTI.



Disusun oleh:

Khalila Indana (202410715127)

F3A6

PROGRAM STUDI INFORMATIKA FAKULTAS ILMU KOMPUTER

UNIVERSITAS BHAYANGKARA JAKARTA RAYA

2025

BAB I

PENDAHULUAN

1.1 Tujuan Praktikum

1. Memahami konsep dasar rangkaian penjumlah biner seperti Half Adder dan Full Adder, serta perannya dalam sistem logika digital. Mahasiswa juga diharapkan dapat merancang rangkaian Adder 8-bit menggunakan Logisim.
2. Mempelajari cara kerja rangkaian Comparator yang digunakan untuk membandingkan dua angka biner dan menentukan hubungan logis di antara keduanya, seperti apakah suatu nilai lebih besar, lebih kecil, atau sama.
3. Menguasai teknik merancang dan menganalisis rangkaian Adder dan Comparator berdasarkan tabel kebenaran serta persamaan logika yang relevan.
4. Mampu membuat, menguji, dan mensimulasikan rangkaian Adder dan Comparator dengan bantuan Logisim Evolution untuk memverifikasi bahwa rangkaian bekerja sesuai fungsinya.
5. Mengembangkan kemampuan menghubungkan konsep dasar logika digital dengan penerapannya secara nyata dalam sistem komputasi maupun pada rangkaian elektronik.

1.2 Langkah-langkah Kerja

1. Membuat 8-bit Adder
 - 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

Tabel kebenaran 8-bit adder direpresentasikan melalui tabel kebenaran full adder, karena 8-bit adder disusun dari delapan buah full adder yang beroperasi

secara berantai. Rangkaian Adder ini memiliki total 16 bit, sehingga menghasilkan kombinasi masukan sebesar 2^{16} atau 65.536 baris pada tabel kebenaran. Setiap full adder memiliki tiga input (A, B, dan C-in) serta dua output (Sum dan C-out).

- Bangun rangkaian 8-bit adder dengan ekspresi logika berikut:

Rumus umum:

$$S_i = A_i \oplus B_i \oplus C_i$$

$$C_{i+1} = (A_i \cdot B_i) + (C_i \cdot (A_i \oplus B_i))$$

Ekspresi logika 8-bit adder:

$$S_0 = A_0 \oplus B_0 \oplus C_0$$

$$C_1 = (A_0 \cdot B_0) + (C_0 \cdot (A_0 \oplus B_0))$$

$$S_1 = A_1 \oplus B_1 \oplus C_1$$

$$C_2 = (A_1 \cdot B_1) + (C_1 \cdot (A_1 \oplus B_1))$$

$$S_2 = A_2 \oplus B_2 \oplus C_2$$

$$C_3 = (A_2 \cdot B_2) + (C_2 \cdot (A_2 \oplus B_2))$$

$$S_3 = A_3 \oplus B_3 \oplus C_3$$

$$C_4 = (A_3 \cdot B_3) + (C_3 \cdot (A_3 \oplus B_3))$$

$$S_4 = A_4 \oplus B_4 \oplus C_4$$

$$C_5 = (A_4 \cdot B_4) + (C_4 \cdot (A_4 \oplus B_4))$$

$$S_5 = A_5 \oplus B_5 \oplus C_5$$

$$C_6 = (A_5 \cdot B_5) + (C_5 \cdot (A_5 \oplus B_5))$$

$$S_6 = A_6 \oplus B_6 \oplus C_6$$

$$C_7 = (A_6 \cdot B_6) + (C_6 \cdot (A_6 \oplus B_6))$$

$$S_7 = A_7 \oplus B_7 \oplus C_7$$

$$C_8 = (A_7 \cdot B_7) + (C_7 \cdot (A_7 \oplus B_7)) \rightarrow \text{Carry akhir (Cout)}$$

Ekspresi logika pada adder 8-bit menggunakan operasi XOR untuk menghasilkan keluaran Sum dan kombinasi AND-OR untuk menghasilkan Carry. Perhitungan dilakukan berantai dari bit paling rendah hingga bit paling tinggi, di mana hasil Carry sebuah bit menjadi Carry-in untuk bit berikutnya.

- Menyusun rangkaian menggunakan delapan gerbang XOR dan delapan gerbang AND pada Logisim.
- Hubungkan input A dan B menggunakan Constant.

- Tambahkan LED pada output Sum (S) dan Carry (C).

2. Comparator

- Tabel kebenaran

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

Penyusunan tabel kebenaran untuk comparator 8-bit tidak memungkinkan dilakukan secara manual karena tingkat kompleksitasnya sangat tinggi. Rangkaian ini memiliki 16 input total (masing-masing 8 bit untuk A dan 8 bit untuk B), sehingga menghasilkan 2^{16} atau 65.536 kemungkinan kombinasi pada tabel kebenarannya. Pada comparator 8-bit terdapat tiga output, yaitu A>B, A=B, dan A<B, karena hanya tiga kondisi inilah yang mungkin muncul saat dua bilangan biner dibandingkan. Jika nilai A lebih besar daripada B, maka output A>B akan bernilai 1 sementara dua output lainnya bernilai 0. Jika A sama dengan B, maka output A=B bernilai 1. Sebaliknya, apabila A lebih kecil daripada B, maka output A<B akan bernilai 1.

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

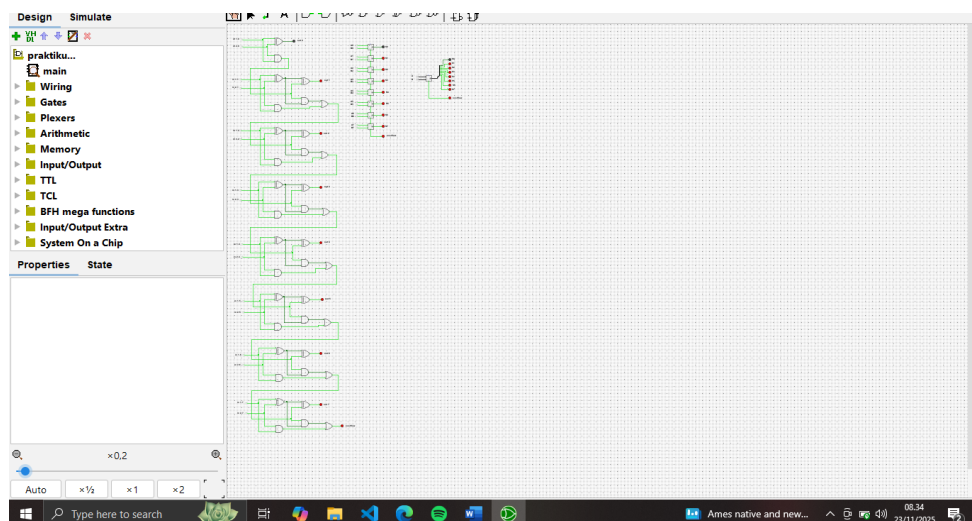
- A = B, akan bernilai benar jika semua bit bernilai sama
 $(A = B) = E7 \cdot E6 \cdot E5 \cdot E4 \cdot E3 \cdot E2 \cdot E1 \cdot E0$
- A > B, akan bernilai benar jika pada suatu bit ke-i, A > B dan semua bit yang lebih signifikan bernilai sama
 $A > B = G7 + (E7G6) + (E7E6G5) + (E7E6E5G4) + \dots + (E7E6E5E4E3E2E1G0)$
- A < B, akan bernilai benar jika pada suatu bit ke-i, A < B dan semua bit yang lebih signifikan bernilai sama
 $A < B = L7 + (E7L6) + (E7E6L5) + (E7E6E5L4) + \dots + (E7E6E5E4E3E2E1L0)$

- Menggabungkan semua sinyal kesetaraan
Semua output kesetaraan (E7-E0) 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 $G_7 + (E_7 \cdot G_6) + (E_7 \cdot E_6 \cdot G_5) + \dots + (E_7 \cdot E_6 \dots E_1 \cdot G_0)$
 - 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

1. 8-bit Adder

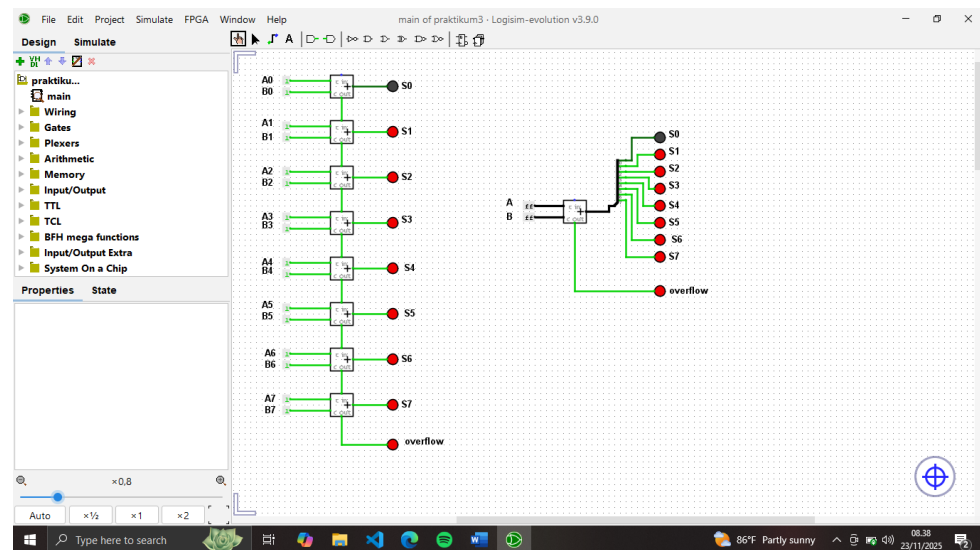
- Gerbang Logika



Rangkaian penjumlahan biner 8-bit dibuat menggunakan gerbang logika dasar atau full-adder. Proses penjumlahan dimulai dari bit terkecil (A_0, B_0), di mana tiap bit melakukan operasi penjumlahan yang menghasilkan nilai sum dan carry-out. Sinyal sum diperoleh melalui dua tahap gerbang XOR, pertama untuk $A \oplus B$, kemudian hasilnya di-XOR-kan lagi dengan carry-in

(Cin) untuk menghasilkan Sum ($A0 \oplus B0 \oplus Cin$). Sedangkan carry-out (Cout) dibentuk dari kombinasi tiga gerbang AND yang hasilnya di-OR-kan ($Cout = (Ai \cdot Bi) + (Ci \cdot (Ai \oplus Bi))$). Carry-out dari setiap full adder diteruskan menjadi carry-in untuk full adder berikutnya, membentuk susunan ripple carry adder 8-bit. Hasil penjumlahan (sum) ditampilkan melalui delapan LED, dan carry terakhir dari bit ke-7 berfungsi sebagai sinyal indikator overflow.

- Aritmatika



Pada rangkaian ini terdapat dua input utama, yaitu A dan B, yang masing-masing terdiri dari 8 bit. Kedua masukan tersebut langsung dihubungkan ke modul adder sehingga proses penjumlahan biner dapat berlangsung secara otomatis di dalam blok aritmetika. Adder bekerja dengan menjumlahkan setiap bit mulai dari bit paling rendah hingga bit paling tinggi sambil memperhitungkan carry-in internal. Hasil penjumlahan kemudian ditampilkan melalui keluaran S0–S7 yang merepresentasikan nilai sum untuk bit 0 sampai 7. Sementara itu, carry dari bit paling signifikan ditampilkan melalui LED overflow sebagai tanda bahwa hasil penjumlahan melebihi batas kapasitas.

- Tabel pengujian

A	B	C-in	Sum	C-out
00000000	00000000	0	00000000	0
00001101	00000111	0	00010100	0
10101010	01010101	0	11111111	0

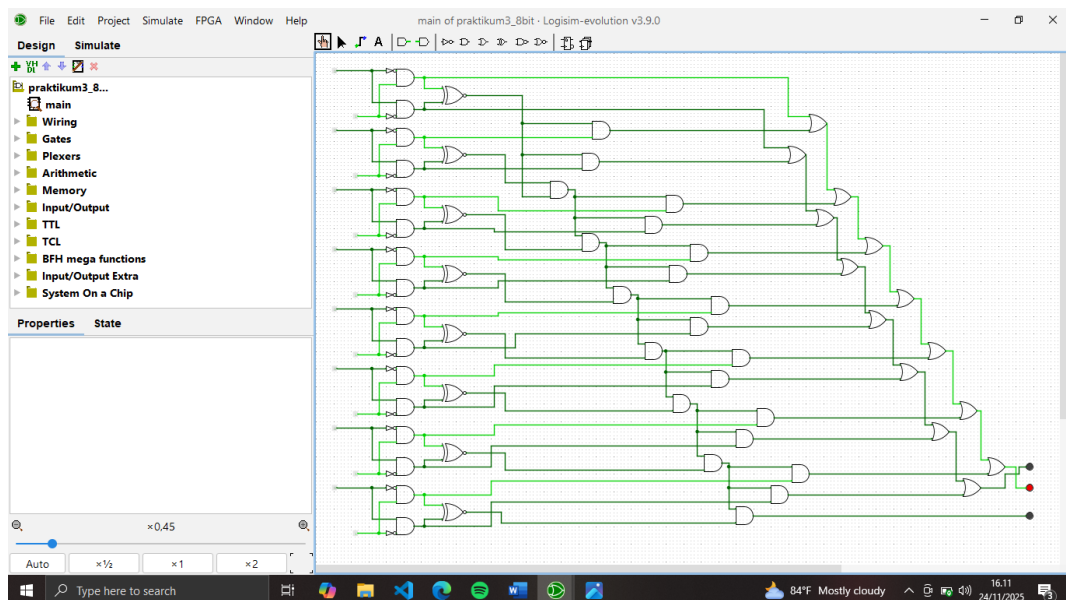
11111111	11111111	0	11111110	1
10011001	11001100	1	01100110	1

Pada pengujian pertama, input A dan B bernilai 00000000 dengan C-out bernilai 0, menunjukkan bahwa tidak terjadi proses penjumlahan tambahan maupun overflow. Pada pengujian kedua, nilai acak 00001101 dan 00000111 dengan C-in 0 menghasilkan sum 00010100 dan C-out tetap 0, sesuai dengan hasil penjumlahan biner tanpa kelebihan bit.

Pada pengujian ketiga, kombinasi komplemen 10101010 dan 01010101 menghasilkan sum 11111111 dengan C-out 0, membuktikan bahwa rangkaian mampu menangani berbagai pola bit dan tetap memberikan output yang benar. Pada pengujian keempat, kedua input bernilai maksimum yaitu 11111111, dan hasil penjumlahannya adalah 11111110 dengan C-out 1, menandakan bahwa jumlah bit telah melampaui kapasitas 8bit sehingga terjadi overflow.

Pada pengujian kelima, input 10011001 dan 11001100 dengan C-in 1 menghasilkan sum 01100110 dan C-out 1. Hal ini menunjukkan bahwa rangkaian dapat memproses nilai C-in dari rangkaian eksternal secara tepat.

2. Comparator



Pada praktikum 8-bit comparator, rangkaian digunakan untuk membandingkan dua bilangan biner 8-bit (A dan B). Proses pembandingan dilakukan secara berurutan mulai dari bit yang paling signifikan (A7 dan B7) hingga bit paling kecil (A0 dan B0). Setiap pasangan bit A_i dan B_i dianalisis untuk menentukan apakah nilai A_i

lebih besar, lebih kecil, atau sama dengan B_i . Untuk mengecek apakah kedua bit sama, digunakan gerbang XNOR, dan hasil kesamaan ini diteruskan sebagai sinyal acuan untuk tahapan bit berikutnya.

Keputusan akhir dari perbandingan didasarkan pada tiga kondisi yang saling tidak mungkin terjadi bersamaan: $A > B$, $A = B$, atau $A < B$. Kondisi $A > B$ akan muncul apabila terdapat bit A_i yang bernilai lebih tinggi daripada B_i dan seluruh bit yang lebih signifikan sebelumnya bernilai sama. Kondisi $A = B$ diperoleh jika semua bit pada A dan B identik, sementara kondisi $A < B$ terjadi bila terdapat bit A_i yang lebih rendah dari B_i dengan bit-bit yang lebih signifikan tetap sama. Berdasarkan kondisi ini, keluaran $A > B$, $A = B$, dan $A < B$ diaktifkan sesuai untuk menunjukkan hubungan kedua bilangan tersebut.

1.4 Kesimpulan

Dari hasil praktikum, dapat disimpulkan bahwa rangkaian Adder dan Comparator memainkan peranan yang sangat penting dalam logika digital serta pengolahan data biner. Pada rangkaian Adder 8-bit, proses penjumlahan berlangsung secara berurutan menggunakan prinsip full adder, di mana setiap bit menghasilkan nilai sum dan carry yang diteruskan ke tahap berikutnya. Simulasi pada Logisim membuktikan bahwa rangkaian bekerja sesuai teori untuk berbagai variasi input, baik tanpa carry, dengan carry-in, hingga kondisi overflow. Hal ini menunjukkan bahwa desain logika yang dibuat telah berjalan dengan benar.

Untuk rangkaian Comparator 8-bit, perbandingan antara dua data biner dilakukan mulai dari bit paling signifikan hingga bit paling rendah, memanfaatkan sinyal kesetaraan, lebih besar, dan lebih kecil. Hasil simulasi menunjukkan bahwa rangkaian mampu menentukan kondisi $A > B$, $A = B$, atau $A < B$ secara tepat dan tidak saling tumpang tindih pada setiap kombinasi input. Penyebaran sinyal kesetaraan dari satu bit ke bit berikutnya bekerja efektif untuk memastikan hasil perbandingan akurat.

Secara keseluruhan, praktikum ini memperkuat pemahaman mengenai keterkaitan antara tabel kebenaran, persamaan logika, serta implementasinya dalam rangkaian digital. Dengan bantuan simulasi Logisim, mahasiswa dapat melihat secara langsung bagaimana teori diterapkan pada rangkaian Adder dan Comparator, sehingga menambah pemahaman dalam merancang sistem elektronik dan komputasi digital.