Mata Kuliah

Pengantar Sistem Digital

Rangkaian Kombinasi - Half Adder dan Full Adder

Dosen Pengampu:

Arif Rifai Dwiyanto ST., MTI

Disusun Oleh:

Inez Agatha (202410715023)

F3A7

PROGRAM STUDI INFORMATKA FAKULTAS

ILMU KOMPUTER

UNIVERSITAS BHAYANGKARA JAKARTA RAYA

2025

DAFTAR ISI

BAB I

PENDAHULUAN

1.1 TUJUAN

1. Mahasiswa memahami konsep rangkaian kombinasi.
2. Mahasiswa mampu membangun rangkaian Half Adder dan Full Adder menggunakan Logisim.
3. Menganalisis prinsip kerja Half Adder, yaitu rangkaian logika yang mampu melakukan penjumlahan dua bit biner dan menghasilkan output *Sum* serta *Carry*.
4. Memahami fungsi Full Adder, yang dapat menjumlahkan tiga bit (dua bit input dan satu *carry-in*) sehingga lebih sesuai untuk operasi aritmetika yang lebih kompleks.

1.2 LANGKAH-LANGKAH

a. Membuat Half Adder

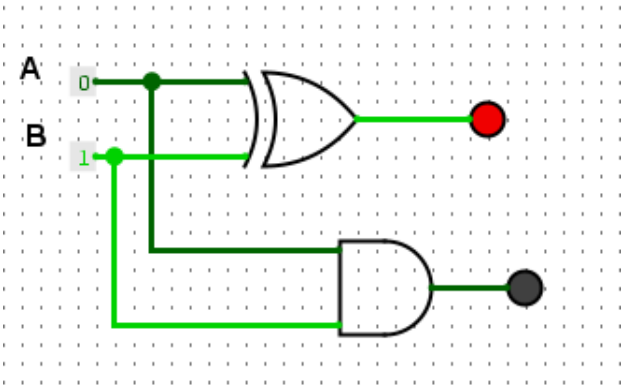
1. Buka Logisim-evolutin
2. Membuat rangkaian Half Adder dengan ekspresi logika berikut:

   $S=A \oplus B$ *(XOR)*

   $C=A \cdot B$ *(AND)*

A	B	C - Carry	S - Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0
		AND	**XOR**

3. Tambahkan Gerbang logika XOR untuk input A dan B dan gerbang logika AND untuk menghubungkan output XOR dan C_{in}



Half Adder 1

b. Membuat Full Adder

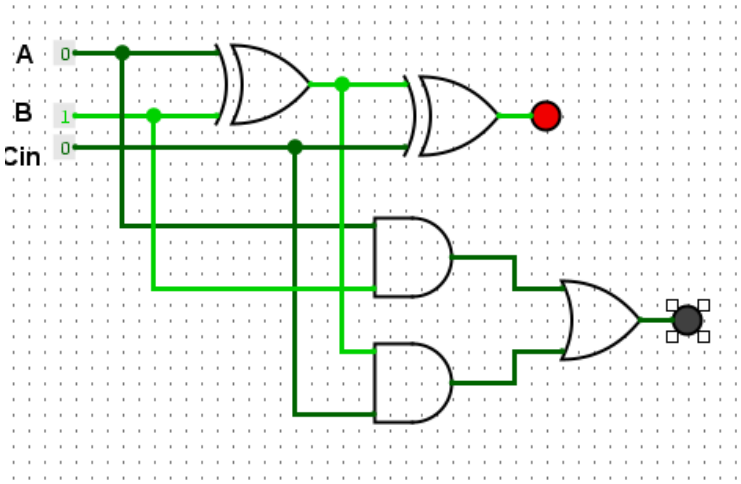
1. Membuat rangkaian Full Adder dengan ekspresi logika berikut:

$S = (A \oplus B) \oplus C_{in}$

$C_{out} = (A \cdot B) + ((A \oplus B) \cdot C_{in})$

A	B	C_{in}	SUM	C_{out}
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

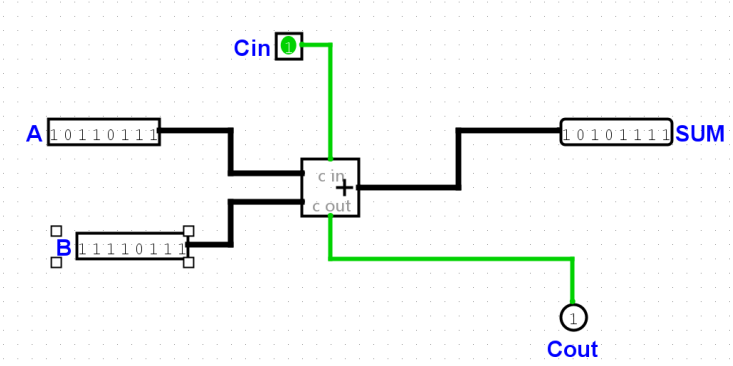
2. Hubungkan input A dan B ke gerbang XOR $(A \oplus B)$
3. Hasil output XOR AB hubungkan ke gerbang logika XOR dengan inputan C_{in} untuk menghasilkan output SUM
4. Tambahkan 2 gerbang AND untuk menghubungkan output XOR A dan input C_{in}, lalu input A dan B
5. Kedua gerbang AND akan di hubungkan dengan gerbang OR untuk menghasilkan Carry-out



Full Adder 1

c. Adder Multi-bit

1. Tambahkan 5 pin untuk A, B, C_{in}, C_{out}, dan SUM
2. Lalu tambahkan Adder dan hubungkan kelima input tersebut ke adder
3. Ubah bit menjadi 8 pada Adder A,B, dan SUM
4. Ubah tipe SUM dan C_{out} menjadi output

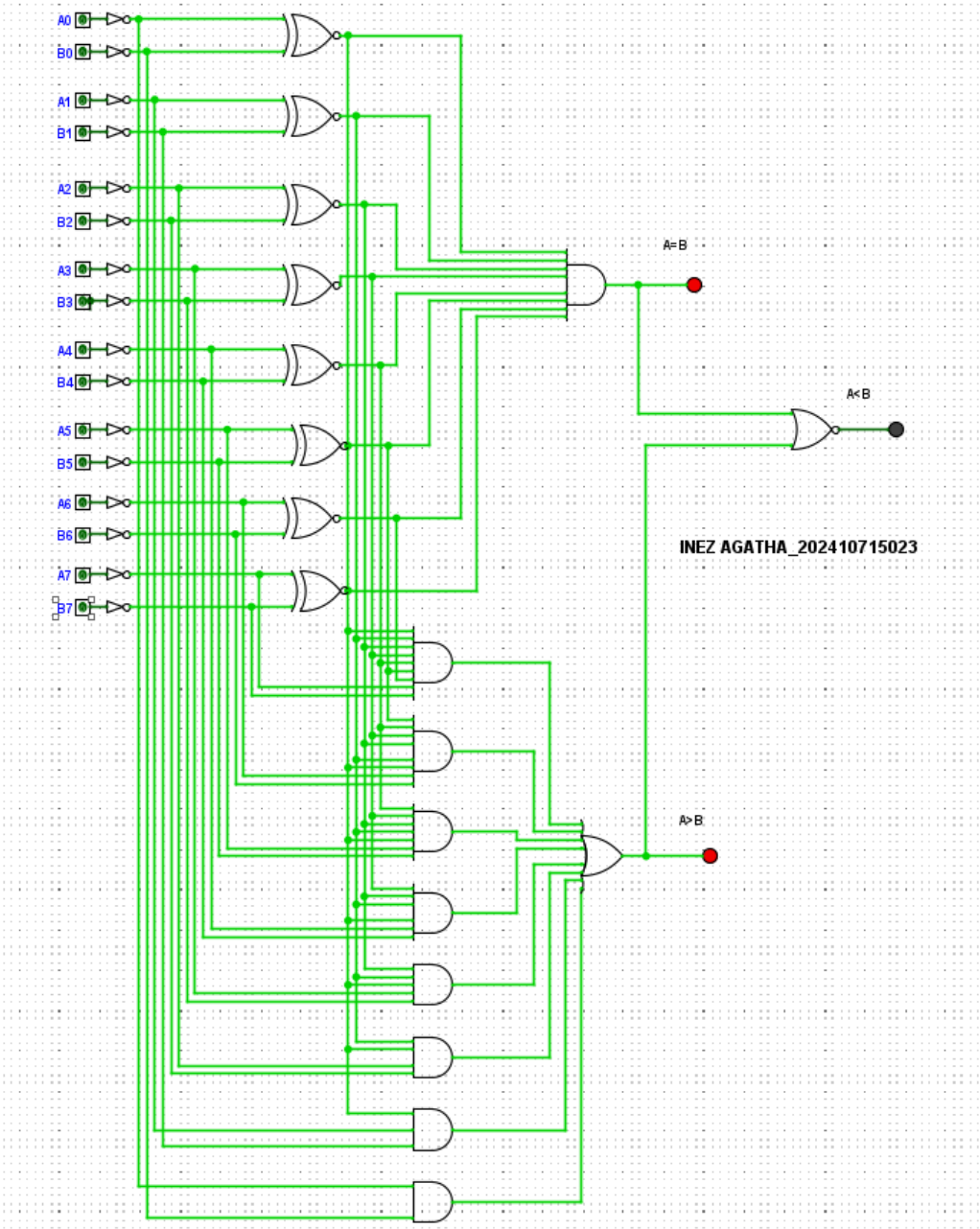


Adder Multi-bit 8

BAB II

2.1 HASIL DAN PEMBAHASAAN

a. Hasil 8bit adder dan komperator



8bit Adder Komparator

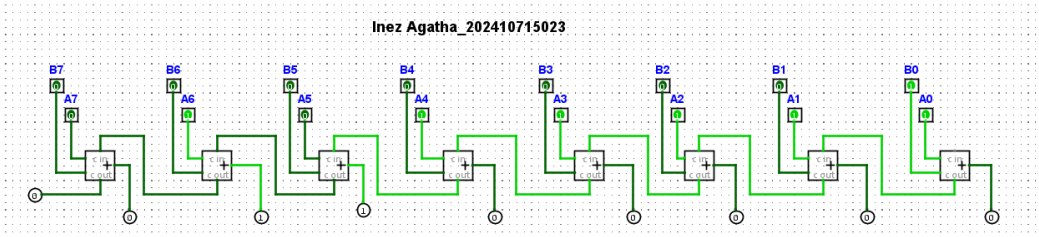
Rangkaian pada gambar adalah sebuah komparator 8-bit yang dipakai untuk membandingkan dua data biner, yaitu A dan B. Masing-masing data terdiri dari delapan bit, sehingga rangkaian harus mengecek setiap pasangan bit dari A0–A7 dan B0–B7. Tujuan utama rangkaian ini adalah menghasilkan tiga keluaran, yaitu kondisi A = B**,** A > B**,** dan A < B**.** Pada gambar terlihat bahwa bagian untuk mengecek A = B dan A > B sudah dibuat lengkap, sedangkan keluaran A < B bisa ditambahkan menggunakan logika yang sama.

Pada sisi paling kiri, setiap bit Ai dan Bi masuk ke gerbang XOR. Fungsi XOR di sini adalah mengecek apakah kedua bit tersebut sama atau beda. Jika output XOR bernilai 0, berarti Ai dan Bi sama; kalau 1 berarti berbeda. Karena kondisi A = B hanya terjadi jika semua bit dari A dan B sama, maka semua output XOR harus bernilai 0. Setelah itu, semua hasil XOR digabungkan lewat gerbang NOR, sehingga rangkaian akan mengaktifkan output A = B ketika tidak ada bit yang berbeda.
Untuk kondisi A > B**,** rangkaian memulai pengecekan dari bit yang paling signifikan (MSB). Setiap bit dibandingkan menggunakan logika Ai AND NOT Bi untuk mendeteksi apakah A lebih besar pada posisi bit tertentu. Tetapi pengecekan ini hanya valid jika bit-bit di atasnya (yang lebih signifikan) nilainya sama. Karena itu, rangkaian memakai beberapa gerbang AND berlapis untuk memastikan bahwa bit yang lebih besar sudah diperiksa dan nilainya memang sama. Setelah itu, seluruh hasil pembandingan tiap bit digabungkan melalui gerbang OR untuk menghasilkan sinyal A > B.
Sementara itu, keluaran A < B sebenarnya bisa disusun dengan cara yang sama seperti A > B, hanya saja logikanya dibalik menjadi NOT Ai AND Bi. Hasil dari tiap bit juga harus dicek dengan syarat bit-bit yang lebih tinggi sama, lalu digabungkan dengan OR agar bisa menghasilkan satu output A < B. Jika bagian ini belum ada di rangkaian, tinggal meniru struktur A > B dengan kondisi pembanding yang kebalikan.
Secara umum, rangkaian ini sudah memenuhi konsep dasar komparator 8-bit, karena sudah memiliki bagian pengecekan kesetaraan dan pembandingan per bit. Dengan menambahkan logika untuk kondisi A < B, rangkaian ini akan lengkap dan bisa digunakan sebagai komparator digital yang bekerja secara penuh.

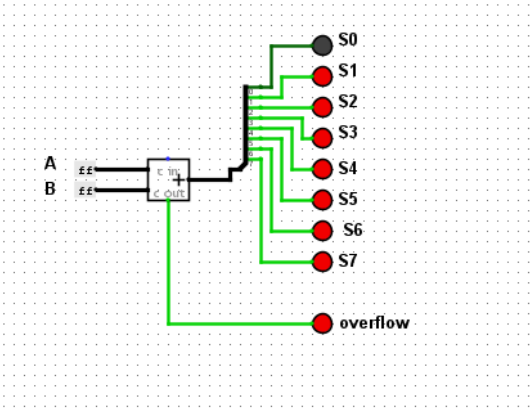
Komparator yang memiliki 8bit mempunyai 256x256 (65.536) baris, karena terlalu banyak maka akan menggunakan tabel kondisi output, seperti :

A	B	A > B	A = B	A < B
00000000	00000000	0	1	0
00000001	00000000	1	0	0
00000100	00000110	0	0	1
10000000	01111111	1	0	0
11111111	11111110	1	0	0
01010101	01010101	0	1	0
00110000	01000000	0	0	1

b. 8bit adder artimatika



8bit Adder Aritmatika1



8bit Adder Aritmatika 2

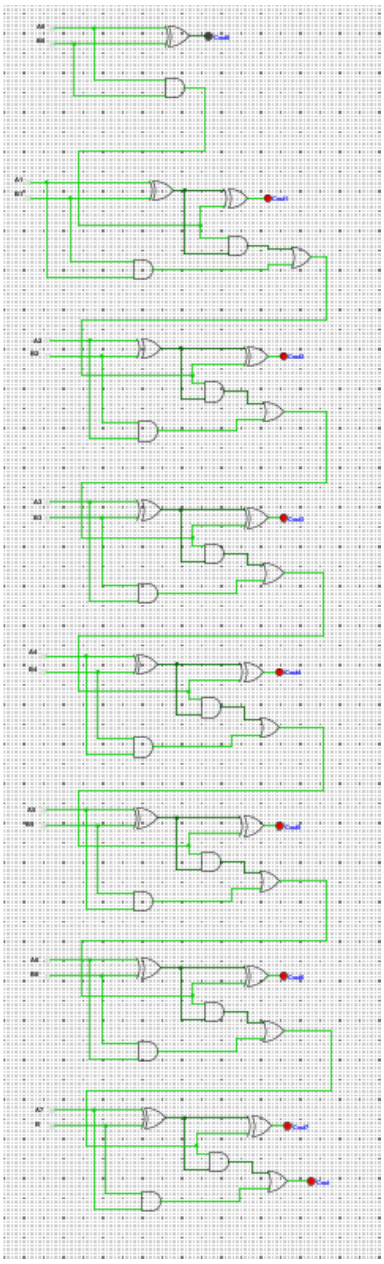
Rangkaian pada gambar merupakan implementasi dari penjumlah biner 8-bit yang disusun menggunakan beberapa blok Full Adder yang dihubungkan secara berantai. Setiap full adder menerima dua input bit, yaitu Ai dan Bi, lalu menghasilkan output berupa sum dan carry. Susunan ini memungkinkan operasi penjumlahan dilakukan secara bertahap mulai dari bit paling rendah (LSB) hingga bit paling tinggi (MSB).

Setiap pasangan bit, seperti A0 dengan B0, A1 dengan B1, dan seterusnya hingga A7 dengan B7, masuk ke full adder masing-masing. Pada bit pertama (LSB), carry-in bernilai 0 karena belum ada penjumlahan sebelumnya. Hasil carry dari full adder pertama kemudian diteruskan ke full adder berikutnya sebagai carry-in. Mekanisme ini disebut ripple carry, karena nilai carry "mengalir" dari satu blok ke blok berikutnya sampai mencapai bit paling tinggi.

Output sum dari masing-masing full adder menunjukkan hasil penjumlahan per bit, sementara carry-out yang terakhir (dari bit A7 + B7) menunjukkan adanya overflow jika hasil penjumlahan melebihi kapasitas 8-bit. Dengan demikian, rangkaian ini dapat menjumlahkan dua bilangan biner 8-bit secara lengkap dan akurat.

Secara keseluruhan, rangkaian ini sudah mencerminkan prinsip dasar aritmatika digital, di mana beberapa full adder digabungkan untuk membentuk penjumlah yang lebih besar. Desain ripple carry seperti ini banyak digunakan karena sederhana dan mudah diimplementasikan, meskipun memiliki kelemahan berupa delay yang bergantung pada banyaknya bit.

c. 8bit Adder



- Sum = Ai XOR Bi XOR Cin

- Cout = (Ai AND Bi) OR (Cin AND (Ai XOR Bi))

  Pada rangkaian ini, setiap full adder dihubungkan secara bertingkat dari atas ke bawah. Carry-out dari satu blok diteruskan menjadi carry-in bagi blok berikutnya. Cara ini dikenal sebagai ripple-carry, di mana carry "mengalir" ke adder berikutnya sampai mencapai bit terakhir. Karena itu, rangkaian dapat melakukan penjumlahan bit dari posisi paling rendah hingga paling tinggi secara berurutan. Hasil sum pada setiap blok (yang ditunjukkan dengan indikator warna merah pada gambar) menunjukkan hasil penjumlahan per bit, sedangkan carry yang bergerak ke bawah memastikan bahwa operasi penjumlahan multi-bit dapat dilakukan secara lengkap. Dengan menggunakan metode ini, rangkaian dapat menjumlahkan dua bilangan biner namun tetap mempertahankan desain yang sederhana menggunakan gerbang logika dasar.

2.2 KESIMPULAN

Pada praktikum 8-bit Adder dan 8-bit Comparator, dapat disimpulkan bahwa kedua rangkaian ini merupakan blok dasar yang sangat penting dalam sistem digital, khususnya pada unit aritmatika dan logika (ALU).

Pertama, pada 8-bit Adder, rangkaian berhasil melakukan proses penjumlahan dua data biner berukuran 8-bit menggunakan konsep *full adder* yang disusun secara berantai (*ripple carry*). Hasil pengujian menunjukkan bahwa output penjumlahan sudah sesuai, termasuk bit *carry out* ketika terjadi overflow. Hal ini membuktikan bahwa rangkaian bekerja akurat dalam melakukan operasi aritmatika dasar.

Sementara itu, 8-bit Comparator berfungsi untuk membandingkan dua data biner 8-bit dan menghasilkan tiga kondisi:

- A = B
- A > B
- A < B

Pengujian menunjukkan bahwa sinyal output yang aktif selalu sesuai dengan hasil perbandingan yang seharusnya. Ini menandakan bahwa comparator mampu membedakan nilai biner dengan benar berdasarkan logika pembanding yang digunakan.

Secara keseluruhan, praktikum ini memperkuat pemahaman tentang bagaimana data biner diproses dalam rangkaian digital, terutama terkait operasi aritmatika dan logika. Selain itu, praktikum juga membantu mahasiswa memahami pentingnya akurasi dalam perancangan rangkaian digital serta bagaimana setiap bit memiliki peran pada hasil akhir.