**MODUL 12**

**ADDER DAN KOMPARATOR**

Pertemuan ke : 12

Hari/Tanggal :

Total Alokasi Waktu : 100 menit

**12.1.** **Kemampuan Akhir yang Direncanakan**

Mahasiswa mempelajari konsep rangkaian adder, Komparator dan membuat rangkaian pada simulator

**12.2.** **Alat dan bahan**

1. Digital trainer set atau Digital Work
2. IC 7400 (Nand), 7402 (Nor), 7408 (And), 7432 (Or), 7486 (Xor), 7404 (Not), 7483 (4-bit full adder), 7485 (Comparator).

**12.3.** **Teori Dasar**

**12.3.1 Adder**

Adder adalah sebuah rangkaian logika yang digunakan untuk melakukan operasi penjumlahan pada angka biner. Digunakan untuk menambahkan dua bilangan biner tunggal atau digabungkan secara bertingkat untuk melakukan penjumlahan bilangan biner dengan beberapa bit. Rangkaian adder sangat penting dalam bidang elektronika dan komputasi digital karena berbagai alasan yang mendasar. Berikut adalah beberapa alasan mengapa kita membutuhkan rangkaian adder:

* **Operasi Aritmatika Dasar:**

Adder digunakan untuk melakukan penjumlahan, yang merupakan salah satu operasi aritmatika dasar dalam matematika. Penjumlahan adalah operasi fundamental yang diperlukan untuk hampir semua jenis perhitungan.

* **Komponen Kunci dalam ALU (Arithmetic Logic Unit):**

ALU adalah bagian dari unit pemrosesan pusat (CPU) yang melakukan semua operasi aritmatika dan logika. Rangkaian adder adalah komponen inti dalam ALU yang memungkinkan CPU untuk melakukan operasi penjumlahan, pengurangan (dengan menggunakan adder yang dimodifikasi), dan operasi aritmatika lainnya.

* **Perhitungan dalam Sistem Digital:**

Dalam sistem digital, hampir semua bentuk perhitungan dan pengolahan data melibatkan operasi penjumlahan. Misalnya, penghitungan alamat memori, penambahan waktu dalam sistem pengatur waktu (timer), dan operasi pada register dalam mikroprosesor.

* **Desain Memori:**

Adder digunakan dalam desain memori untuk pengalamatan dan pengelolaan data. Mereka membantu dalam menghitung alamat memori berikutnya atau mengindeks data secara efisien.

* **Pengolahan Sinyal Digital:**

Dalam pengolahan sinyal digital (DSP), adder digunakan untuk operasi seperti penggabungan sinyal, filtering, dan transformasi Fourier. Penjumlahan adalah operasi dasar dalam banyak algoritma DSP.

* **Algoritma dan Struktur Data:**

Banyak algoritma dan struktur data memerlukan operasi penjumlahan untuk penghitungan indeks, pengelolaan data, dan manipulasi struktur seperti stack, queue, dan array.

* **Sistem Kendali dan Robotika:**

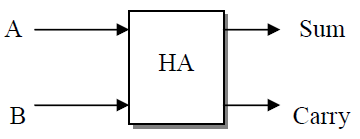
Dalam sistem kendali dan robotika, adder digunakan untuk menggabungkan sinyal sensor, menghitung posisi atau kecepatan, dan mengimplementasikan algoritma kendali.

* **Desain dan Pengembangan Komputer:**

Dalam desain komputer, adder adalah blok bangunan dasar untuk pengembangan unit komputasi. Mereka digunakan dalam berbagai macam operasi mulai dari penjumlahan sederhana hingga pengolahan data yang kompleks.

Rangkaian adder ini bisa berupa half-adder atau full-adder, tergantung pada kompleksitas operasi yang diperlukan. Half-adder digunakan untuk menjumlahkan dua bit tunggal, sementara full-adder dapat menjumlahkan tiga bit (dua bit data dan satu bit carry dari penjumlahan sebelumnya). Implementasi adder yang lebih kompleks, seperti ripple-carry adder, carry-lookahead adder, dan adder paralel lainnya, digunakan untuk meningkatkan kecepatan dan efisiensi operasi penjumlahan dalam sistem yang lebih besar dan lebih cepat.

1. **Half Adder**

****

Rangkaian ini hanya dapat digunakan untuk operasi penjumlahan data bilangan biner sampai 1 bit saja. Rangkaian half adder mempunyai 2 masukan dan 2 keluaran yaitu Summary out (Sum) dan Carry out (Carry). Aturan half adder :

1. Jika A=0 dan B=0 dijumlahkan, hasilnya S (Sum) = 0.
2. Jika A=0 dan B=1 dijumlahkan, hasilnya S (Sum) = 1.
3. Jika A=1 dan B=1 dijumlahkan, hasilnya S (Sum) = 0. Dengan nilai pindahan Cy (Carry Out) = 1.

**Struktur Half Adder:**

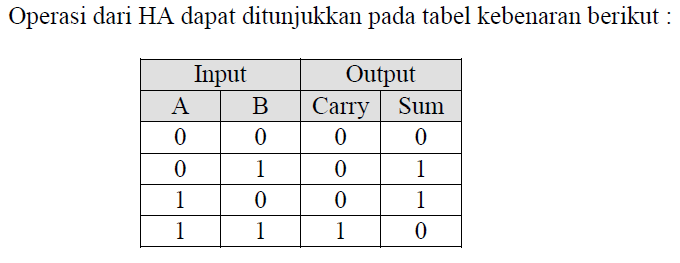
**Input:**

* + A: Bit pertama yang akan dijumlahkan.
  + B: Bit kedua yang akan dijumlahkan.

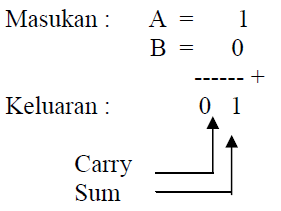
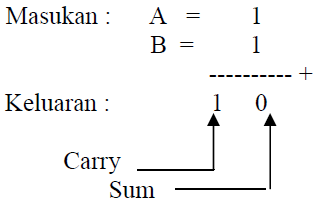
**Output:**

* + Sum (S): Hasil penjumlahan bit A dan B tanpa memperhitungkan carry (sisa penjumlahan).
  + Carry (C): Bit carry yang merupakan hasil dari penjumlahan yang membawa nilai lebih dari 1.

**Tabel Kebenaran Half Adder:**



**Perhitungan Half Adder:**

**Persamaan Logika Half Adder:**

* **Sum (S):** Sum merupakan hasil dari operasi XOR pada input A dan B. Persamaan logikanya adalah: S=A⊕B
* **Carry (C):** Carry merupakan hasil dari operasi AND pada input A dan B. Persamaan logikanya adalah: C=A⋅B

**Implementasi Rangkaian Logika Half Adder**

Half adder dapat diimplementasikan menggunakan gerbang logika dasar seperti AND dan XOR. Berikut adalah diagram rangkaian logika untuk half adder:

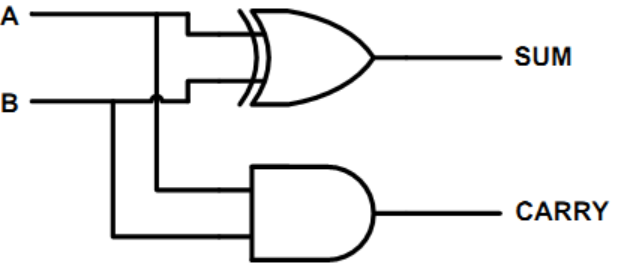
**Gerbang XOR untuk Sum:**

* + Menghasilkan output 1 jika salah satu dari input adalah 1, tetapi bukan keduanya.

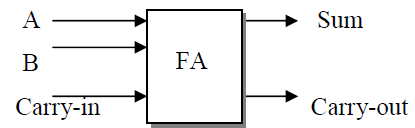
**Gerbang AND untuk Carry:**

* + Menghasilkan output 1 hanya jika kedua input adalah 1.

**Rangkaian logika sederhana Half Adder:**

****

1. **Full Adder**

****

Digunakan untuk menjumlahkan bilangan biner yang lebih dari 1 bit. Rangkaian half adder mempunyai 3 masukan dan 2 keluaran yaitu Summary out (Sum) dan Carry out (Carry). Satu masukan tambahan adalah Carry-in.

**Struktur Full Adder:**

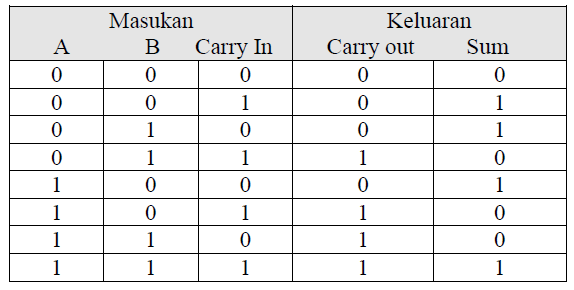
**Input:**

* + A: Bit pertama yang akan dijumlahkan.
  + B: Bit kedua yang akan dijumlahkan.
  + Cin​: Carry input, yaitu bit carry dari penjumlahan sebelumnya.

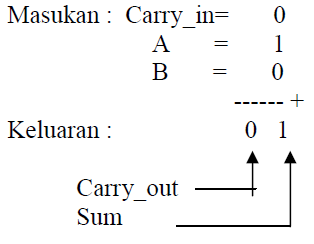
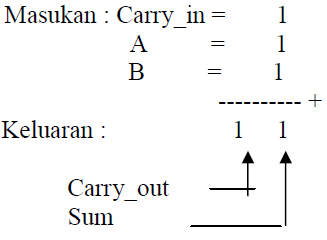
**Output:**

* + Sum (S): Hasil penjumlahan bit A, B, dan Cin.
  + Carry (C\_{out}): Bit carry yang dihasilkan dari penjumlahan, yang akan dibawa ke penjumlahan berikutnya.

**Tabel Kebenaran Full Adder:**



**Perhitungan Full Adder:**

**Persamaan Logika Full Adder:**

* **Sum (S)**: Sum merupakan hasil dari operasi XOR pada input A, B, dan Cin​. Persamaan logikanya adalah: S=A⊕B⊕Cin​
* **Carry (C\_{out}):** Carry merupakan hasil dari operasi logika yang melibatkan input A, B, dan Cin​. Persamaan logikanya adalah: Cout=(A⋅B)+(Cin⋅(A⊕B))

**Implementasi Rangkaian Logika Full Adder**

Full adder dapat diimplementasikan dengan menggabungkan dua half adder dan satu gerbang OR. Berikut adalah diagram rangkaian logika untuk full adder:

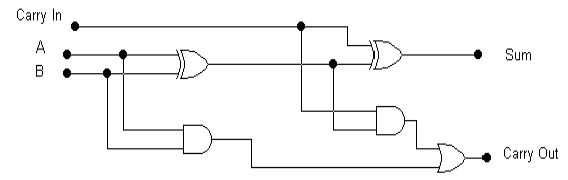
**Gerbang XOR untuk Sum:**

Dua gerbang XOR digunakan untuk menjumlahkan A, B, dan Cin​.

**Gerbang AND dan OR untuk Carry:**

Dua gerbang AND dan satu gerbang OR digunakan untuk menghasilkan output carry.

**Rangkaian logika sederhana Full Adder:**

****

**12.3.2 Komparator**

Komparator adalah rangkaian logika digital yang membandingkan dua bilangan biner dan menentukan hubungan di antara mereka, seperti apakah satu bilangan lebih besar, lebih kecil, atau sama dengan bilangan lainnya. Komparator digunakan dalam berbagai aplikasi digital, termasuk pengendalian proses, pengolahan data, dan sistem komputer.

**Fungsi Komparator**

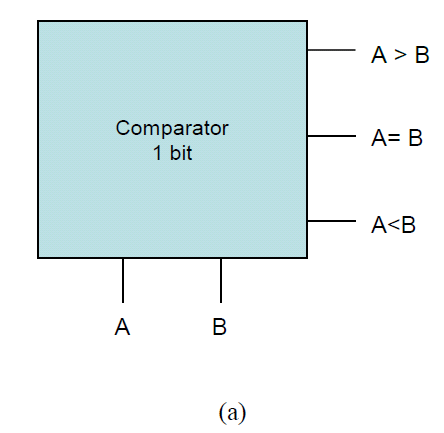
Komparator membandingkan dua bilangan biner, biasanya dilambangkan sebagai A dan B, dan menghasilkan tiga output utama:

* **A > B:** Output ini bernilai 1 jika A lebih besar dari B.
* **A = B:** Output ini bernilai 1 jika A sama dengan B.
* **A < B:** Output ini bernilai 1 jika A lebih kecil dari B.

**Jenis Komparator**

* **Komparator 1-bit:** Membandingkan dua bit tunggal.
* **Komparator Multi-bit (N-bit):** Membandingkan dua bilangan biner yang terdiri dari N bit.

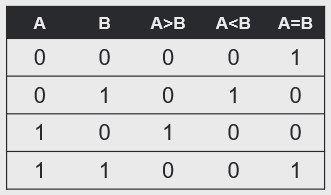
1. **Komparator 1-bit** (Membandingkan 2 bit biner (A dan B))

****

**Output :**

* A > B: Output = 1
* A < B: Output = 0
* A = B: Output = 1 jika A sama dengan B, selain itu Output = 0

**Tabel kebenaran:**

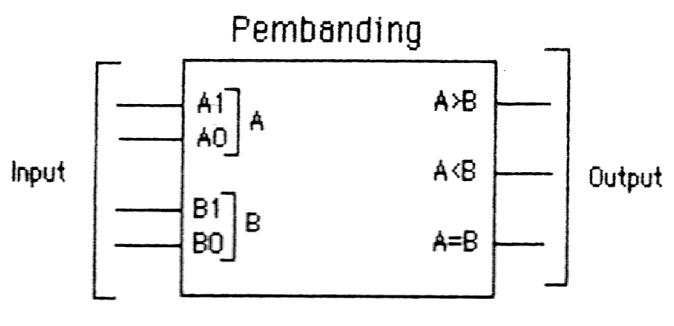


**Rangkaian logika:**

Rangkaian logika komparator 1-bit bisa diimplementasikan menggunakan gerbang AND, OR, dan NOT. Berikut adalah persamaan logikanya:

* **A > B:** A⋅B’
* **A = B:** (A⋅B)+(A’⋅B’)
* **A < B:** A’⋅B

1. **Komparator 2-bit**

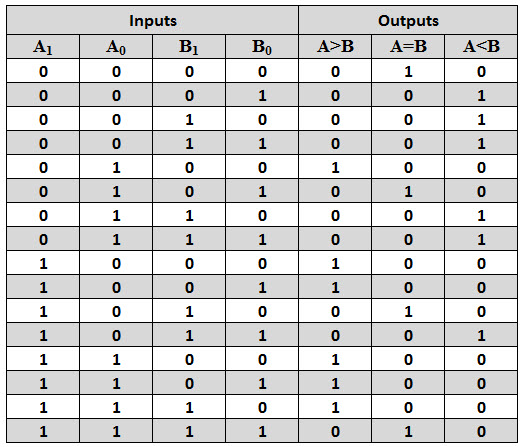
****

Rangkaian digital yang membandingkan dua bilangan biner masing-masing terdiri dari dua bit. Rangkaian ini menentukan apakah satu bilangan lebih besar, lebih kecil, atau sama dengan bilangan lainnya.

**Output :**

* A > B: Output = 1, jika A lebih besar dari B
* A < B: Output = 1, jika A lebih kecil dari B
* A = B: Output = 1 jika A sama dengan B, selain itu Output = 0

**Tabel kebenaran:**

****

**Rangkaian logika:**

Rangkaian komparator 2-bit dapat diimplementasikan dengan menggabungkan beberapa gerbang logika dasar (AND, OR, NOT) berdasarkan persamaan logika dari tabel kebenaran.

**Persamaan Logika:**

* A > B: (A1⋅B1’)+(A1⋅A0⋅B0’)+(B1’⋅A0⋅B0’)
* A = B: (A1⋅B1⋅A0⋅B0)+(A1’⋅B1’⋅A0’⋅B0’)+(A1⋅B1⋅A0’⋅B0’)+(A1’⋅B1’⋅A0⋅B0)
* A < B: (A1’⋅B1)+(A1’⋅A0’⋅B0)+(B1⋅A0’⋅B0)

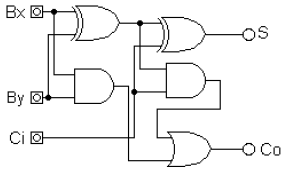
**12.4. Kesehatan dan Keselamatan Kerja**

* 1. Periksalah terlebih dahulu semua komponen aktif maupun pasif sebelum digunakan!
  2. Bacalah dan pahami petunjuk pratikum pada lembar kegiatan belajar!
  3. Hati-hati dalam penggunaan peralatan pratikum!

**12.5. Prosedur percobaan**

**Adder**

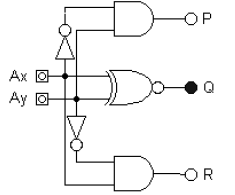
1. Silahkan buka lembar kerja DigitalWork!
2. Rangkailah gerbang logika adder berikut ini:



1. Sambungkan terminal input dengan switch input dan terminal output dengan lampu LED.
2. Sambungkan terminal supply dengan +5 Volt dan ground.
3. Amati dan catat output terhadap kombinasi keadaan input.
4. Rangkailah adder dengan menggunakan IC 7483.
5. Sambungkan terminal input dengan switch input dan terminal output dengan lampu LED.
6. Sambungkan terminal supply dengan +5 Volt dan ground
7. Amati dan catat output terhadap kombinasi keadaan input.

**Komparator**

1. Rangkailah gerbang logika komparator berikut ini:



1. Sambungkan terminal input dengan switch input dan terminal output dengan lampu LED.
2. Sambungkan terminal supply dengan +5 Volt dan ground.
3. Amati dan catat output terhadap kombinasi keadaan input.
4. Rangkailah comparator dengan menggunakan IC 7485.
5. Sambungkan terminal input dengan switch input dan terminal output dengan lampu LED.
6. Sambungkan terminal supply dengan +5 Volt dan ground.
7. Amati dan catat output terhadap kombinasi keadaan input.

**12.6. Tugas**

1. Buatlah tabel kebenaran rangkaian gerbang logika adder yang dirangkai dari gerbang logika dasar dan adder dalam bentuk IC!
2. Buatlah tabel kebenaran rangkaian gerbang logika comparator yang dirangkai dari gerbang logika dasar dan adder dalam bentuk IC!
3. Buatlah Laporan pembahasan dan kesimpulan percobaan diatas