

MODUL PRAKTIKUM SISTEM DIGITAL

MODUL 1: INTRODUKSI LOGISIM EVOLUTION & VERIFIKASI SISTEM BILANGAN BINER

MATA KULIAH: SISTEM DIGITAL (TIF-1106) — S1 TEKNIK INFORMATIKA

Alokasi Waktu: 150 Menit — Software Utama: Logisim Evolution v3.8.0+

1. TUJUAN PRAKTIKUM & ALAT

Tujuan Pembelajaran Praktikum:

1. Mahasiswa mampu mengoperasikan antarmuka grafis (*GUI*) software **Logisim Evolution**.
2. Mahasiswa mampu membangun sirkuit logika 8-bit menggunakan komponen *Pin Input*, *Probe Output*, *Adder*, dan *Hex Display*.
3. Mahasiswa mampu memverifikasi proses pembentukan **1's Complement** dan **2's Complement** pada data biner 8-bit secara visual.
4. Mahasiswa mampu mengidentifikasi dan mendeteksi kondisi **Arithmetic Overflow** pada operasi aritmatika biner bertanda.

Alat & Perangkat Keras/Lunak:

- Komputer/Laptop dengan OS Windows, macOS, atau Linux.
- Perangkat Lunak: **Logisim Evolution** (dapat diunduh gratis dari repository resmi GitHub Logisim-Evolution).

2. DASAR TEORI SINGKAT

Representasi Bilangan Biner Bertanda (2's Complement): Dalam arsitektur komputer 8-bit, bit paling kiri (*Most Significant Bit* — MSB) bertindak sebagai bit tanda (0 = positif, 1 = negatif). Representasi **2's Complement** dari sebuah bilangan biner X diperoleh melalui rumus:

$$X_{2's} = \overline{X} + 1$$

di mana $\overline{X}$ adalah komplement satu (*1's Complement*) yang dihasilkan dengan membalikkan setiap bit (0 → 1, 1 → 0) menggunakan gerbang NOT.

Deteksi Arithmetic Overflow: *Overflow* terjadi ketika hasil operasi matematika melampaui rentang nilai yang dapat ditampung oleh format 8-bit bertanda [−128 s.d. +127]. Secara hardware, overflow terdeteksi apabila terdapat perbedaan nilai antara *Carry-In* (C_{in}) dan *Carry-Out* (C_{out}) pada bit MSB:

$$\text{Overflow} = C_{in} \oplus C_{out}$$

1. 3. LANGKAH-LANGKAH PERCOBAAN

1.1. Percobaan 1: Pengenalan Interface & Inverter 8-Bit (1's Complement)

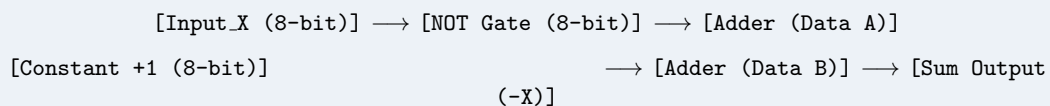
1. Buka aplikasi **Logisim Evolution**.
2. Pada panel **Explorer** (kiri), buka folder **Wiring** → pilih komponen **Pin** dan letakkan pada canvas kerja.
3. Atur parameter pada panel **Attribute Table** (kiri bawah):
 - **Data Bits:** 8
 - **Label:** Input_X
 - **Facing:** East
4. Pada folder **Gates**, pilih gerbang **NOT Gate** dan atur **Data Bits:** 8.
5. Hubungkan pin **Input_X** ke masukan gerbang NOT menggunakan kabel (*wire*).

- Ambil komponen **Probe** dari folder **Wiring**, atur **Data Bits**: 8 dan **Radix**: 2's Complement. Hubungkan ke luaran gerbang NOT.
- Pilih **Poke Tool** (ikon tangan Ctrl+1), ubah bit pada **Input_X** menjadi 0000 0101₂ (+5₁₀) dan catat luaran komplemen satu pada Probe.

1.2. Percobaan 2: Perancangan Sirkuit 2's Complement Generator 8-Bit

- Lanjutkan sirkuit dari Percobaan 1.
- Buka folder **Arithmetic** → pilih komponen **Adder** (8-bit).
- Hubungkan luaran gerbang NOT (1's Complement) ke terminal input **Data A** pada Adder.
- Ambil komponen **Constant** dari folder **Wiring**, atur **Data Bits**: 8 dan **Value**: 0x1 (+1).
- Hubungkan komponen **Constant** (+1) ke terminal input **Data B** pada Adder.
- Hubungkan luaran **Sum** dari Adder ke ****Probe Output**** (8-bit) dan ****Hex Display****.
- **Pengujian****: Set **Input_X** = 0000 0101₂ (+5). Amati bahwa keluaran Adder menghasilkan 1111 1011₂ (HEX: FB₁₆), yang merupakan nilai -5₁₀ dalam representasi 2's Complement!

Skema Blok Rangkaian Percobaan 2



1.3. Percobaan 3: Penjumlahan Biner Bertanda & Deteksi Overflow

- Buat dua buah Pin Input 8-bit: **Num_A** dan **Num_B**.
- Hubungkan kedua pin ke komponen ****Adder 8-Bit****.
- Hubungkan pin ****Carry-Out (C_{out})**** dan ****Carry-In MSB**** dari Adder ke gerbang ****XOR 1-bit**** untuk membuat indikator LED **Overflow**.
- Uji Kasus Normal**: Masukkan **Num_A** = +50 (0011 0010₂) dan **Num_B** = +30 (0001 1110₂). Catat hasil Sum (+80) dan status Overflow (0).
- Uji Kasus Overflow**: Masukkan **Num_A** = +100 (0110 0100₂) dan **Num_B** = +50 (0011 0010₂). Catat hasil Sum (terjadi pembacaan salah -106) dan LED Overflow menyala (1).

2. 4. TABEL HASIL PERCOBAAN & EVALUASI

Lengkapilah tabel pengamatan berikut dalam laporan resmi praktikum Anda:

Input Desimal	Biner 8-bit (X)	1's Comp ($\bar{X}$)	2's Comp (Logisim)	Hex Display	Hasil Desimal
+5	0000 0101	1111 1010	1111 1011	FB	-5
+12	0000 1100				-12
+45	0010 1101				-45
+127	0111 1111				-127

5. TUGAS LAPORAN RESMI PRAKTIKUM

Ketentuan Pengumpulan Laporan:

1. Lampirkan *tangkap layar (screenshot)* sirkuit yang Anda buat untuk Percobaan 1, Percobaan 2, dan Percobaan 3 pada Logisim Evolution.
2. Berikan analisis mengapa pada Percobaan 3 kasus uji $+100 + (+50)$ menghasilkan nilai desimal negatif (-106) dan jelaskan mekanisme gerbang XOR dalam mendeteksi overflow bit tanda.
3. Simpan berkas simulasi Logisim dengan format nama: `Modul1_NIM>Nama.circ`.
4. Unggah laporan resmi (format PDF) beserta berkas `.circ` ke sistem LMS sebelum pertemuan praktikum minggu berikutnya.