

# Percobaan 2

## Membangun Logika Kombinasi dengan Transistor CMOS

### 2.1. Tujuan

- Memberikan pengenalan terhadap VLSI Design CAD Tool: **Electric**<sup>TM</sup>
- Memperkenalkan pendekatan desain hirarki (*Hierarchical Design Approach*) dalam proses perancangan
- Membangun CMOS Non-Inverting Gate: AND dan OR
- Membangun MUX, Decoder
- Mensimulasi dan menganalisa logika kombinasi pada level transistor

### 2.2. Landasan Teori

#### • Pendekatan Desain-Hirarki

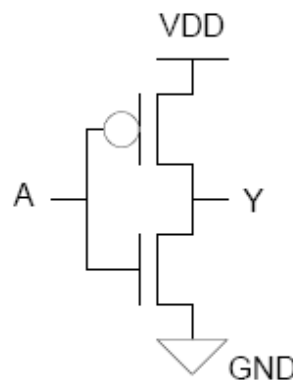
Suatu mekanisme yang sangat diperlukan dalam menangani perancangan sistem yang kompleks adalah hirarki (*hierarchy*). Sebuah sistem yang besar dapat dilakukan partisi kedalam banyak unit. Setiap unit disusun atas beberapa blok-fungsi (*functional block*). Sementara blok-blok fungsi ini dibangun dari beberapa sel (*cell*). Yang paling akhir sel ini dibangun dari transistor.

Fungsi dari sebuah sistem memang dapat dengan mudah jika dilihat dalam sisi pandang top-level dengan unit-unit sebagai kotak-hitam (black box) dengan interface-nya yang didefinisikan dengan baik dan fungsi-fungsi, dibandingkan dengan melihat sistem dari sudut pandang pada masing-masing transistor. Hirarki juga memberikan fasilitas pemanfaatan ulang atas desain (*design reuse*). Sebuah blok dapat dirancang dan diverifikasi sekali, kemudian dipakai berulang-ulang di banyak tempat. Logika (*logic*),

rangkaian (*circuit*), gambaran fisik (*physical view*) dari sebuah desain seharusnya berbagi dalam level-hirarki yang sama. Hal ini untuk kemudahan proses verifikasi. Sebuah desain hirarki dapat dilihat seperti sebuah struktur pohon dengan keseluruhan chip sebagai akar dan primitive sel sebagai daun.

- **CMOS Inverter**

CMOS Inverter dapat dibangun dari transistor NMOS dan transistor PMOS. Masukan **A** mengontrol kedua masukan transistor. Keluaran **Y** terhubung ke terminal *drain* dari kedua transistor. Terminal *source* dari transistor NMOS dihubungkan ke *ground* (GND) dan terminal *source* dari transistor PMOS dihubungkan ke power (VDD), misalkan 5 volt. Sebagai ketentuan bahwa “*sisi terdekat dengan rel (VDD/GND) adalah source, dan sisi terdekat dengan output adalah drain*”.



Gambar 1: CMOS Inverter

Ketika masukan **A** = 1, transistor NMOS ON dan transistor PMOS OFF. Oleh karena itu keluaran **Y** terhubung ke GND melalui transistor NMOS, tapi tidak terdapat koneksi ke VDD, sehingga bisa dikatakan *pull-down* ke 0. Ketika masukan **A** = 0, transistor PMOS ON dan transistor NMOS OFF. Oleh karena itu keluaran **Y** di-pull-up ke 1 melalui transistor PMOS. Jadi  $Y = \sim A$ .

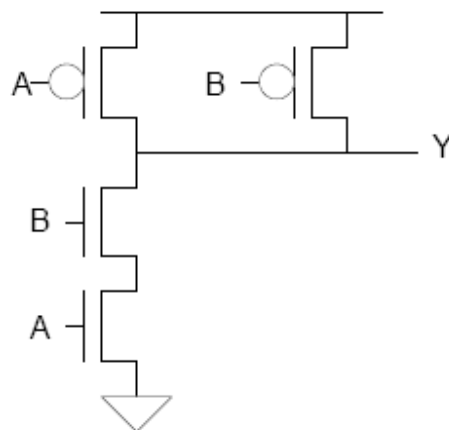
Tabel 1: CMOS Inverter

A	NMOS	PMOS	Y
0	OFF	ON	1
1	ON	OFF	0

- **CMOS NAND Gate**

CMOS NAND gate 2-input dibangun dari 2-transistor NMOS yang tersusun seri dan 2-transistor PMOS yang tersusun parallel, seperti tampak pada gambar-2. Terminal *source* dari salah satu transistor NMOS dihubungkan ke *ground* (GND) dan terminal *drain* dari transistor NMOS yang lain dihubungkan ke keluaran **Y**, sedangkan kedua terminal *source* transistor PMOS dihubungkan ke power (VDD), dan terminal *drain* kedua transistor PMOS dihubungkan ke keluaran **Y**.

Seperti tercantum pada tabel 2, ketika kedua masukan **A** dan **B** diberikan logika 1, maka kedua transistor NMOS ON dan keluaran **Y** akan *pull-low*, atau  $Y = 0$ , dan ketika salah satu masukan **A** atau **B** diberikan logika 0, maka salah satu transistor PMOS akan ON dan keluaran **Y** akan *pull-high*, atau  $Y = 1$ .



Gambar 2: CMOS NAND Gate 2-input

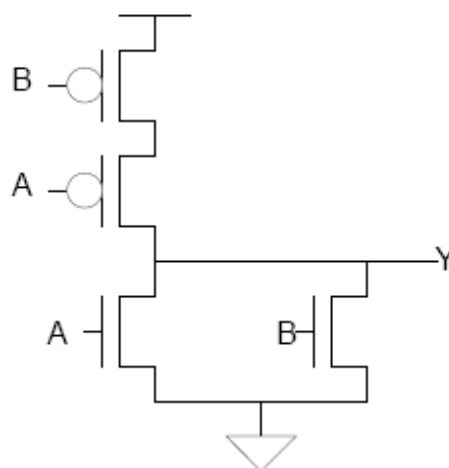
Tabel 2: Tabel kebenaran CMOS NAND Gate 2-input

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

- **CMOS NOR Gate**

NOR gate adalah complement dari NAND gate. CMOS NAND gate 2-input dibangun dari 2-transistor NMOS yang tersusun parallel dan 2-transistor PMOS yang tersusun seri. Terminal *source* dari salah satu transistor PMOS dihubungkan ke power (VDD) dan terminal *drain* dari transistor NMOS yang lain dihubungkan ke keluaran Y, sedangkan kedua terminal *source* dari transistor NMOS dihubungkan ke *ground* (GND) dan terminal *drain* dari kedua transistor NMOS dihubungkan ke keluaran Y. Rangkaian CMOS NAND gate 2-input ditunjukkan pada gambar-3.

Ketika kedua masukan A dan B diberikan logika 0, maka kedua transistor PMOS akan ON dan keluaran Y akan *pull-up*, atau  $Y = 1$ , dan ketika salah satu masukan A atau B diberikan logika 1, maka salah satu transistor NMOS akan ON dan keluaran Y akan *pull-low*, atau  $Y = 0$ . Perilaku ini bisa ditabelkan, seperti pada tabel 3.



Gambar 3: CMOS NOR Gate dengan 2-input

Tabel 3: Tabel kebenaran CMOS NOR Gate 2-input

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

### 2.3. Alat dan Bahan

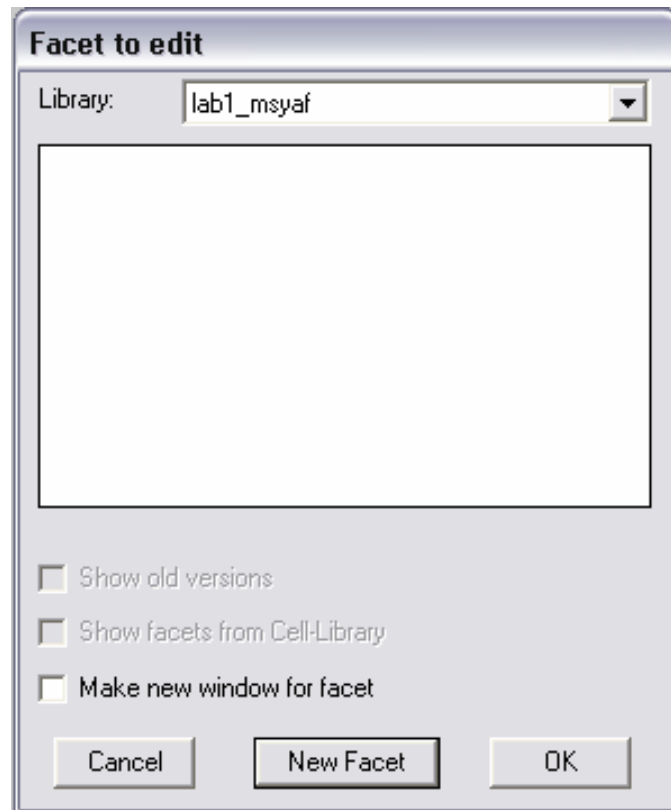
- VLSI Design CAD Tool: **Electric**<sup>TM</sup>

### 2.4. Prosedur Percobaan

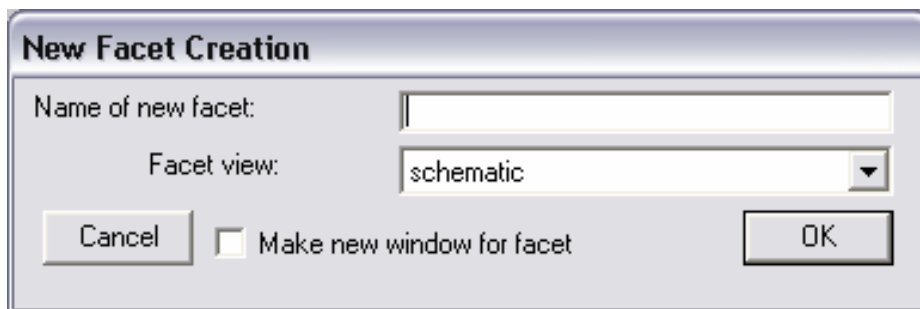
Pada percobaan-1, CMOS NAND gate 2-input sudah dibuat. Oleh karena itu bukanlah kembali file Library yang telah disimpan, kemudian bukalah *facet* baru, Misal: Inverter{sch} facet. Hal ini perlu dilakukan karena pada **Electric**<sup>TM</sup> tiap-tiap design disimpan dalam sebuah *facet*. Setelah *facet*, kemudian gambarlah CMOS Inverter dengan prosedur seperti pada percobaan-1.

Untuk membuka *facet* baru lakukan:

1. Pilih menu **Facet**→**Edit Facet** maka akan muncul **Facet to edit** dialog [gambar-4]. Klik New Facet. Pada **New Facet Creation** dialog box [gambar-5], masukkan nama **Inverter** sebagai nama facet dan pada **facet view** pilih **schematic**. Sebuah window editing baru akan muncul dengan title perc1\_xx:Inverter{sch}. Nama tersebut menyatakan *Library*, *Nama Facet*, dan *View*.
2. Mulailah menggambar rangkaian hingga tampak seperti gambar-1.

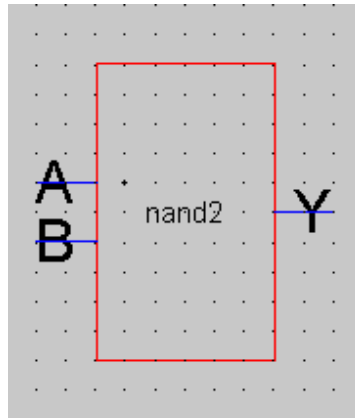


Gambar 4: Library Dialog



Gambar 5: New Facet Creation dialog

3. Buat *icon* pada masing-masing schematic NAND-2 dan Inverter. *Icon* ini sebenarnya adalah symbol dari sebuah sel jika dilihat pada level yang lebih tinggi. Bukalah rancangan `nand2{sch}` dan pilih **View**→**Make Icon**. **Electric**<sup>TM</sup> akan membuat generic *icon* berdasarkan pada *export-export*-nya seperti tampak pada gambar-6.

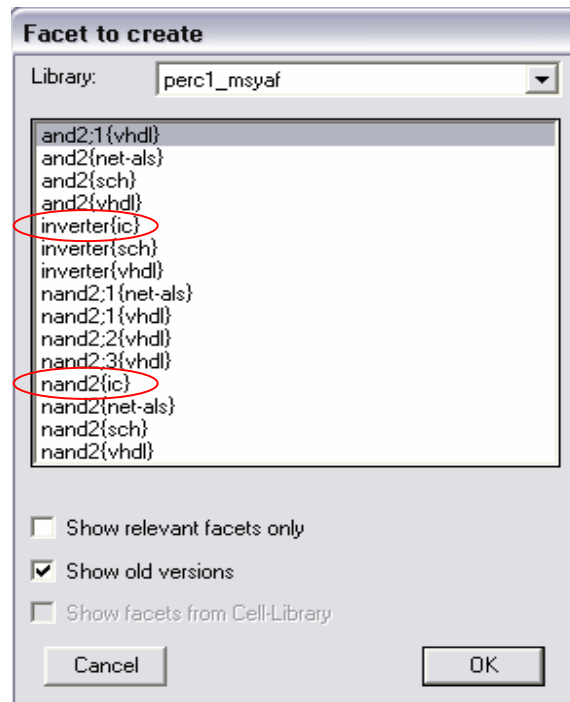


Gambar 6: NAND-2 Icon

4. Ubah bentuk icon NAND-2 yang berbentuk kotak ke bentuk yang lebih familier yaitu seperti simbol gerbang NAND 2-input. Klik pada object icon NAND-2, kemudian pilih **Facets**→**Down Hierarchy (Ctrl-D)** lakukan editing sehingga bentuknya sama dengan symbol gerbang NAND 2-input. Kemudian pilih **Facets**→**Up Hierarchy (Ctrl-U)** untuk kembali lagi ke facet schematic.
5. Bukalah *facet schematic* baru, dan beri nama **and2**.
6. Pilih **Edit**→**New Facet Instance**, pilih `nand2{ic}` klik button **OK**. Hal ini untuk mengambil icon `nand2` untuk dimasukkan dalam *facet* `and2`. Lakukan hal yang sama untuk `inverter{ic}`. Gambar-7, window yang muncul ketika **New Facet Instance** dipilih.
7. Gambarlah schematic hingga tampak seperti Gambar-8 (sisi kiri).
8. Lakukan pengecekan (*Design Rule Check*) sebelum melakukan simulasi. Pilih **Tools**→**DRC**→**Hierarchical Check (F1)**. Jika tidak terdapat kesalahan dalam desain, maka pada window **Electric Messages** akan muncul pesan:
 

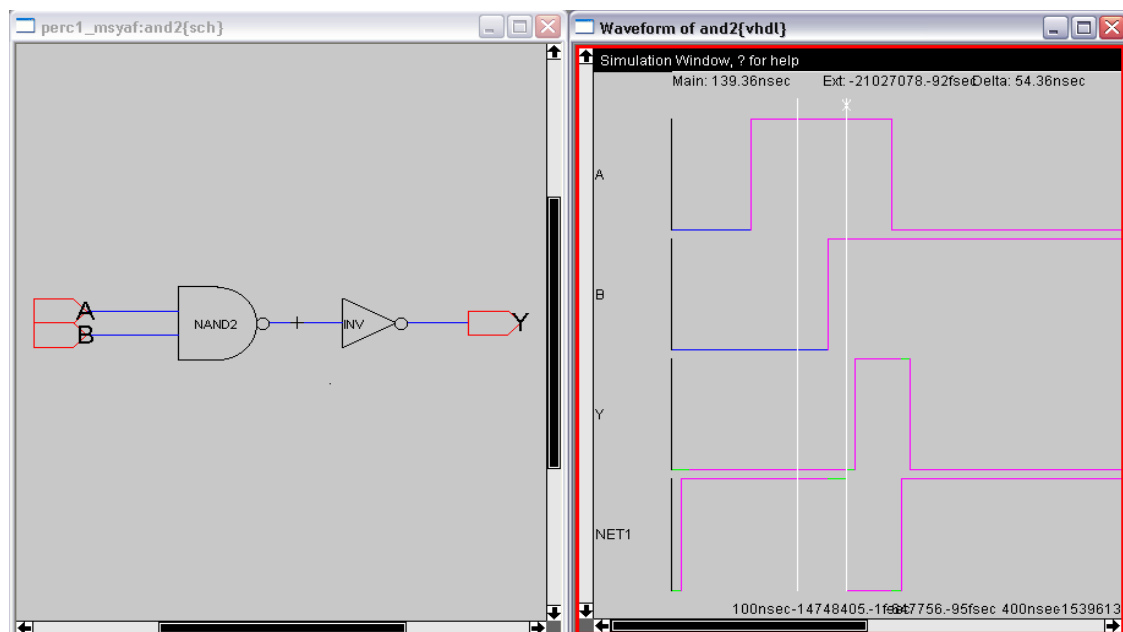
```

      "Found 9 networks, allocated 52 bytes for network numbering (took 0 seconds)
      0 errors found (took 0 seconds)
      
```
9. Lakukan simulai pada rangkaian `and2`. Jika proses simulai tidak berhasil dikarenakan VHDL yang belum digenerate, maka harus digenerate terlebih dahulu. Pilih menu **View**→**Make VHDL View**. List VHDL yang dihasilkan akan tampak seperti pada Lampiran Percobaan-2.



Gambar 7: List facet

10. Pada window **Waveform of and2{vhd}** ubahlah bentuk gelombang seperti tampak pada gambar-8 (sisi kanan).



Gambar 8: AND-2 dan bentuk gelombang hasil simulasi



## 2.5. Tugas

1. Gambarkan rangkaian CMOS untuk MUX dan Decoder
2. Lakukan simulasi pada kedua rangkaian tersebut.
3. Buat kesimpulan atas seluruh percobaan yang telah dilakukan.

### Lampiran Percobaan-2

**-- VHDL automatically generated from facet and2{sch}**

```
entity and2 is port(A, B: in BIT; Y: out BIT);
  end and2;
architecture and2_BODY of and2 is
  component nand2 port(A, B: in BIT; Y: out BIT);
    end component;
  component inverter port(A: in BIT; Y: out BIT);
    end component;
  signal net1: BIT;
begin
  nand21: nand2 port map(A, B, net1);
  inverter1: inverter port map(net1, Y);
end and2_BODY;
```

**-- VHDL automatically generated from facet nand2{sch}**

```
entity nand2 is port(A, B: in BIT; Y: out BIT);
```

```

end nand2;
architecture nand2_BODY of nand2 is
  component PMOStran port(g: in BIT; s, d: inout BIT);
    end component;
  component power port(pwr: out BIT);
    end component;
  component ground port(gnd: out BIT);
    end component;
  component nMOStran port(g: in BIT; s, d: inout BIT);
    end component;
  signal net3, net2, net1: BIT;
begin
  pmos1: PMOStran port map(A, Y, net1);
  pmos2: PMOStran port map(B, Y, net1);
  pwr1: power port map(net1);
  gnd1: ground port map(net2);
  nmos1: nMOStran port map(B, net2, net3);
  nmos2: nMOStran port map(A, net3, Y);
end nand2_BODY;

```

**-- VHDL automatically generated from facet inverter{sch}**

```

entity inverter is port(A: in BIT; Y: out BIT);
  end inverter;
architecture inverter_BODY of inverter is
  component ground port(gnd: out BIT);
    end component;
  component power port(pwr: out BIT);
    end component;
  component nMOStran port(g: in BIT; s, d: inout BIT);
    end component;
  component PMOStran port(g: in BIT; s, d: inout BIT);
    end component;
  signal net2, net1: BIT;
begin
  gnd1: ground port map(net1);
  pwr1: power port map(net2);
  nmos1: nMOStran port map(A, net1, Y);
  pmos1: PMOStran port map(A, Y, net2);
end inverter_BODY;

```

