



MODUL KURSUS DESAIN CHIP

“Desain Skematik Analog”



Di Susun Oleh :

TIM PENYUSUN

PUSAT STUDI MIKROELEKTRONIKA DAN PENGOLAHAN CITRA

UNIVERSITAS GUNADARMA

2017

Kata Pengantar

Dengan memuji dan mengucapkan syukur kepada Allah SWT, yang telah memberikan karunia kekuatan dan kesabaran kepada Penulis untuk menyelesaikan salah satu seri pembuatan modul yaitu Pelatihan Desain Chip “Desain Skematik Analog”. Modul ini merupakan panduan dalam pelaksanaan kursus, yang disusun sebagai materi persiapan bagi mahasiswa peserta kursus sebelum setiap pertemuan akan dimulai. Modul Pelatihan Desain Chip “Desain Skematik Analog” terdiri dari 6 bab dengan materi:

1. Pendahuluan
2. Pengenalan Software Mentor Graphics
3. Skematik Desain Analog
4. Simulasi Rangkaian Skematik Analog
5. Desain Skematik Analog Menggunakan Simbol
6. Latihan Skematik

Secara umum, materi pada setiap pertemuan akan menerangkan konsep atau teori mengenai topik yang akan dibahas, dan menjelaskan secara garis besar langkah yang diperlukan untuk mendesain sebuah chip untuk sirkuit analog menggunakan perangkat lunak mentor graphics yang mendukung pembahasan topik dalam materi tersebut.

Setiap peserta kursus sangat diharapkan untuk mempelajari dengan seksama modul ini, mengingat pemahaman yang baik atas materi ini akan sangat membantu pada waktu proses belajar selama kegiatan kursus berlangsung.

Modul ini merupakan pengembangan dari modul sebelumnya yang berjudul Desain Skematik, Layout dan Simulasi dengan Menggunakan Perangkat Lunak Mentor Graphics Edisi 1. Untuk itu, dalam mengikuti perkembangan teknologi dimasa mendatang, maka modul ini akan direvisi terus menerus apabila diperlukan. Modul ini disusun oleh Pusat Studi Mikroelektronika Dan Pengolahan Citra Universitas Gunadarma dengan bantuan tim yang bekerja secara penuh, yaitu: Robby Kurniawan Harahap, SKom.,MT, Dr. Eri Prasetyo Wibowo, Dr. Atit Pertiwi, dan Veronica Ernita Kristianti, ST.,MT.

Jakarta, 15 Desember 2017

Pusat Studi Mikroelektronika Dan Pengolahan Citra
Universitas Gunadarma

Daftar Isi

Kata pengantar	iii
Daftar Isi	iii
Bab 1. Pendahuluan	1
1.1 Konsep Kerja CMOS	1
1.1.1 Struktur MOS	1
1.1.2 Complementary	2
1.2. Proses Disain IC CMOS.....	3
1.3. Metode Desain IC.....	5
1.4. Perangkat Lunak Desain IC CMOS	6
1.5. Library Teknologi / Process Design Kits	7
1.6. Pabrikasi IC CMOS.....	8
Bab 2. Pengenalan Mentor Graphics Software.....	11
2.1. Sistem Operasi Linux	11
2.2. Menjalankan <i>software</i> Mentor Graphics	14
D. Kegiatan.....	14
D.1. Ekplorasi Opensuse	14
D.2. Menjalankan ICStudio	16
D.3. Menjalankan Design Architech-IC	17
D.4. Menjalankan ICStation	18
Bab 3. Skematik Desain Analog.....	20
3.1. Membuat Skematik Desain Analog Gerbang Logika	20
D. Kegiatan.....	21
D.1. Skematik Gerbang Not / Inverter.....	21
Bab 4. Simulasi Rangkaian Skematik Analog.....	25
4.1. Simulasi Skematik Desain Analog Gerbang Logika.....	25
D. Kegiatan.....	25
D.1. Simulasi Analisa DC.....	25
D.2. Simulasi Analisa DC Operating Point	33
D.3. Simulasi Analisa AC.....	34
D. 4. Simulasi Analisa Transient	36

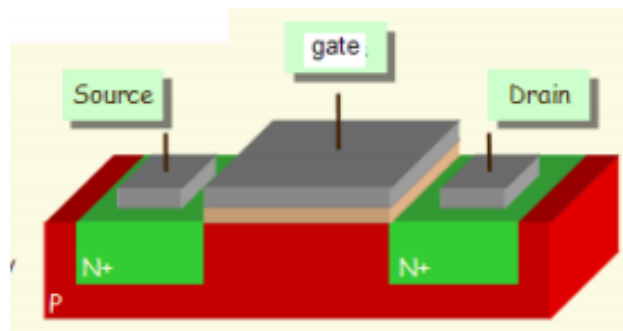
Bab 5. Desain Skematik Analog Menggunakan Simbol.....	39
5.1. Membuat Desain Skematik Analog Gerbang Logika Menggunakan Simbol.....	39
D.1. Pembuatan simbol.....	39
D.2. Simulasi Menggunakan Simbol.....	40
Bab 6. Latihan Skematik.....	42
6.1. Latihan membuat Skematik NAND, OR, NOR, XOR dan XNOR.....	42
D.1.Latihan Skematik Gerbang logika	42
6.2. Latihan membuat Skematik Adder.....	42
D.1. Latihan Skematik Adder	42
6.3. Latihan membuat Skematik Komparator	42
D.1. Latihan Skematik Komparator.....	43
6.4. Latihan membuat Skematik Encoder dan Decoder	43
D.1.Latihan Skematik Encoder dan Decoder	44
6.5. Latihan membuat Skematik Multiplexer dan Demultiplexer.....	44
D.1. Latihan Skematik Multiplexer	44
6.6. Latihan membuat Skematik Latch.....	44
D.1.Latihan Skematik Latch.....	45
6.7. Latihan membuat Skematik Flip-Flop.....	45
D.1. Latihan Skematik Flip-Flop.....	46
6.8. Latihan membuat Skematik Counter.....	46
D.1. Latihan Skematik Counter	47
6.9. Latihan membuat Skematik Register	47
D.1. Latihan Skematik Register.....	48

Bab 1. Pendahuluan

1.1 Konsep Kerja CMOS

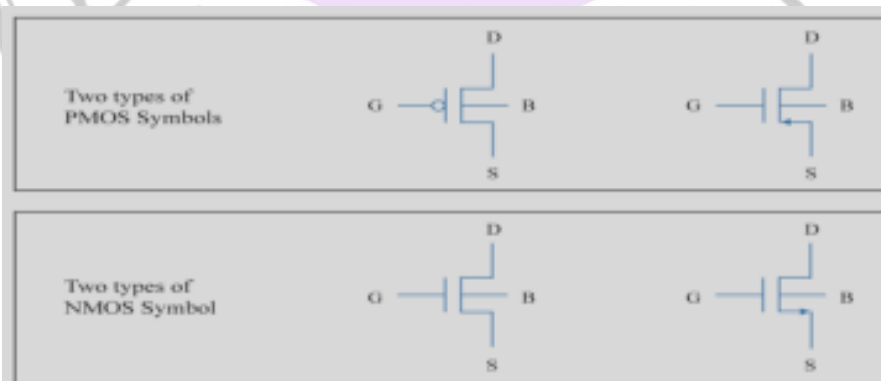
1.1.1 Struktur MOS

Awal tahun 1962 gate semikonduktor terbuat dari aluminium dimana dikenal dengan nama MOS (Metal Oxyde Semiconductor). Setiap MOS terdiri dari Source, Gate dan Drain. Struktur ini bisa dilihat pada gambar 1.1. Untuk MOS tipe-P, source akan terhubung dengan sumber tegangan positif atau Vcc dan drain akan terhubung dengan sumber tegangan 0 V atau ground. Sedangkan untuk MOS tipe-N, source akan terhubung dengan sumber tegangan 0 V dan drain terhubung dengan sumber tegangan positif atau Vcc [1] [2]. Gate berfungsi sebagai pengendali aliran tegangan atau dengan kata lain sebagai pengendali switch.



Gambar 1. 1 Struktur MOS[1] [2].

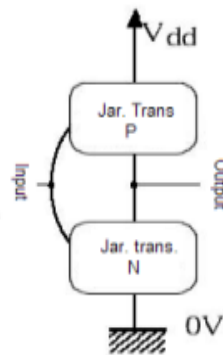
Transistor MOS terdiri dari 2 jenis, pertama transistor MOS dengan tipe kanal-N disebut dengan **NMOS** dan tipe kanal-P disebut dengan **PMOS**. Transistor MOS memiliki 4 pin atau terminal, diantaranya adalah D sebagai **drain**, G sebagai **gate**, S sebagai **source** dan B sebagai **bulk**. Simbol Transistor Mos dapat dilihat pada gambar 1.2.



Gambar 1. 2 Simbol Transistor MOS [1] [2].

1.1.2 Complementary

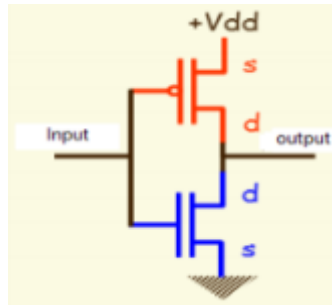
Transistor MOS atau CMOS dapat bekerja karena merupakan kombinasi dari transistor tipe-P (PMOS) dan tipe-N (NMOS) yang masing-masing memiliki fungsi kerja yang berlawanan. Oleh karena itu dapat dikatakan CMOS karena merupakan penggabungan atau kombinasi dari dua atau lebih jenis transistor MOS yang disebut juga dengan komplementer. Gambar 1.5 mengilustrasikan cara kerja Transistor MOS dengan gabungan dua jenis transistor MOS. Prinsip dasar dari Transistor adalah saklar dan. Untuk transistor MOS dengan jenis PMOS akan aktif atau kondisi ON ketika jaringan transistor PMOS tidak aktif, begitupula sebaliknya untuk NMOS. Maka gambar 1.5 merupakan gambaran untuk jaringan CMOS [1] [2].



Gambar 1. 3 Jaringan CMOS [1] [2].

Bentuk rangkaian sederhana yang dapat menjelaskan cara kerja CMOS adalah rangkaian Inverter. Untuk membangun Rangkaian inverter atau dalam simbol gerbang logika dasar disebut dengan gerbang NOT menjadi rangkaian CMOS, maka terdiri dari dua transistor MOS **PMOS** dan **NMOS**. Pola kerja dari rangkaian inverter dalam dua kondisi yaitu terbuka dan tertutup seperti pada gambar 1.4. Berikut ini penjelasan dari dua kondisi tersebut:

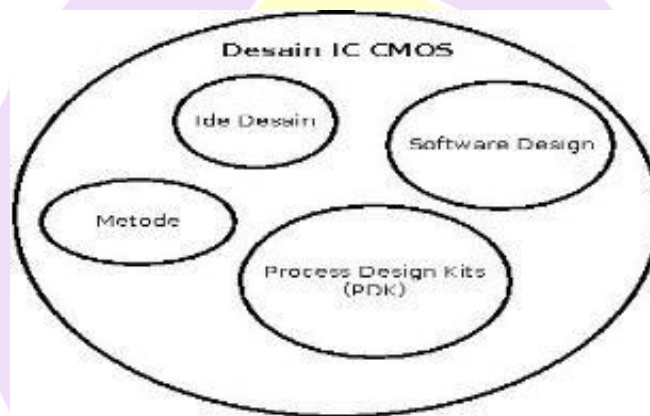
1. Kondisi Tertutup, kondisi ini akan terjadi apabila logika tegangan masukan (input) diberi nilai 1, maka transistor NMOS akan tertutup dan transistor PMOS akan terbuka. Sehingga nilai keluaran akan bernilai 0.
2. Kondisi Terbuka, kondisi ini akan terjadi apabila logika tegangan masukan (input) diberi nilai 0, maka transistor NMOS akan terbuka dan transistor PMOS akan tertutup. Sehingga nilai keluaran akan bernilai 1.



Gambar 1.4. Rangkaian inverter dari CMOS[1] [2].

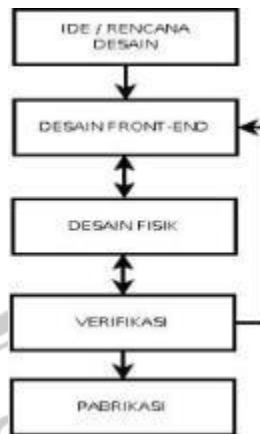
1.2. Proses Disain IC CMOS

Dalam mendesain IC CMOS terdapat 3 komponen utama pada proses desain. Ketiga komponen tersebut adalah (1) Desain, (2) Perangkat Lunak Desain (*Software EDA*), (3) Library Teknologi / *Process Design Kits* (PDK) dan (4) Metode. Gambar 1.5 akan memvisualisasikan ketiga komponen ini dalam desain.



Gambar 1. 5 Empat Komponen Desain IC [1]

Terdapat beberapa tahap dalam mendesain IC CMOS, secara umum terdiri dari 5 tahapan seperti pada gambar 1.6 merupakan hirarki dari desain VLSI untuk IC CMOS. Kelima tahap tersebut adalah dimulai dari (1) ide desain yang akan dibuat, kemudian proses (2) menuangkan kedalam *software* desain IC berupa skematik atau disebut desain *front-end*. Tahap selanjutnya dilanjutkan dengan (3) membuat layout dan melakukan (4) verifikasi terhadap layout tersebut. Terakhir merupakan tahap yang tidak dilakukan oleh pendesain atau user yaitu proses pabrikasi untuk mencetak IC.



Gambar 1.6 Hirarki Alur Konsep Desain CMOS [1] [2].

Pertama diawali dari sebuah ide atau rencana dapat dituangkan ke dalam bentuk gambar pada kertas atau pun gambar secara elektronik seperti menggunakan perangkat lunak untuk membuat blok diagram.

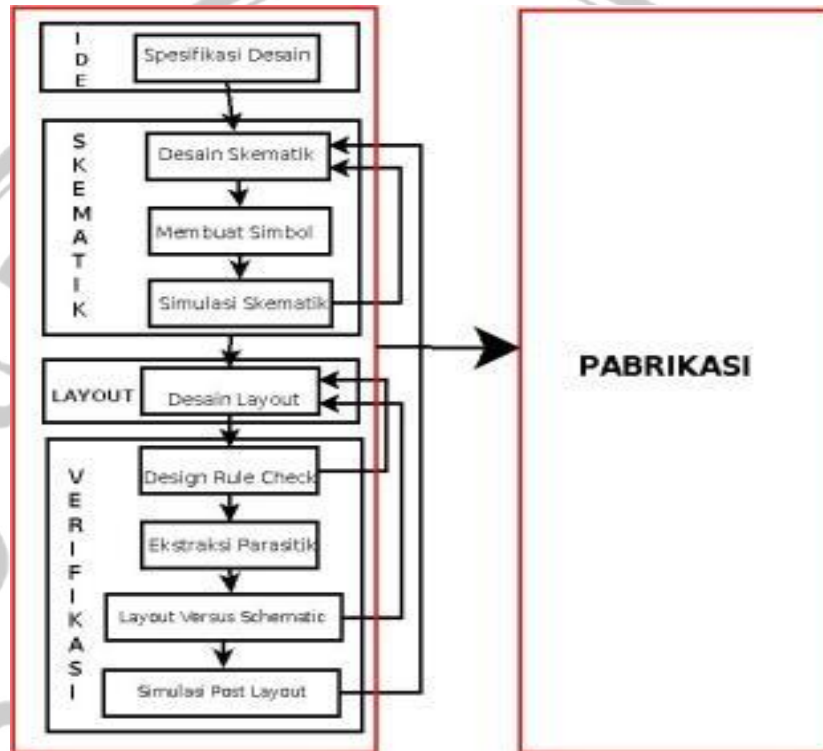
Kedua, tahap *Front-End Design*, tahap ini ide kemudian diimplementasikan kedalam perangkat lunak desain elektronik atau CAD / EDA, seperti Mentor Graphics, Cadence, Synopsis dan lainnya. Tahap ini terdiri dari pembuatan logika / skematik, simulasi dan uji generasi vector.

Ketiga, tahap *physical layout*, tahap ini membuat layout untuk bentuk fisik yang nantinya akan menjadi gambar berupa layout untuk ditanamkan pada sebuah IC. Pada tahap pembuatan layout berdasarkan material atau bahan yang disesuaikan dengan teknologi CMOS. Pada layout Implementasi logika berupa skematik ke bentuk fisik disesuaikan dengan ukuran teknologi CMOS. Ukuran teknologi CMOS sangatlah kecil mikrometer-nanometer (μm -nm). Pembahasan lebih lanjut mengenai Layout akan dibahas pada bab 3 buku ini.

Keempat, tahap verifikasi yang merupakan proses untuk menguji layout yang telah dibuat. Proses pengujian dilakukan dengan menguji layout dengan *design rule* teknologi CMOS, pengujian ini disebut dengan *Design Rules Check (DRC)*. Selain DRC pengujian layout dilakukan dengan membandingkan layout dengan skematik disebut *Layout Versus Schematic (LVS)*. Penjelasan lebih lanjut mengenai verifikasi IC dapat dilihat pada bab 4 buku ini

Kelima, tahap pabrikasi untuk memproses layout menjadi bentuk IC /CHIP. Tahap ini menjadi tahap terakhir dari desain IC CMOS. Layout yang telah diverifikasi akan dikonversi menjadi file *database* yang siap untuk dipabrikasi dengan format GDSII.

Secara Khusus tahap dalam Desain VLSI/ IC CMOS merupakan tahap yang terdiri dari pembuatan skematik dan layout. Gambar 1.7 akan menjelaskan secara detail tahap-tahap yang dilalui ketika mendesain IC CMOS. Dari penjelasan di atas telah dijelaskan konsep desain IC CMOS. Pada prakteknya terdapat beberapa perbedaan tahap dari konsep yang telah dijelaskan di atas. Perbedaan tersebut dikarenakan beberapa faktor atau kendala-kendala seperti keterbatasan layout dan penggunaan teknologi.



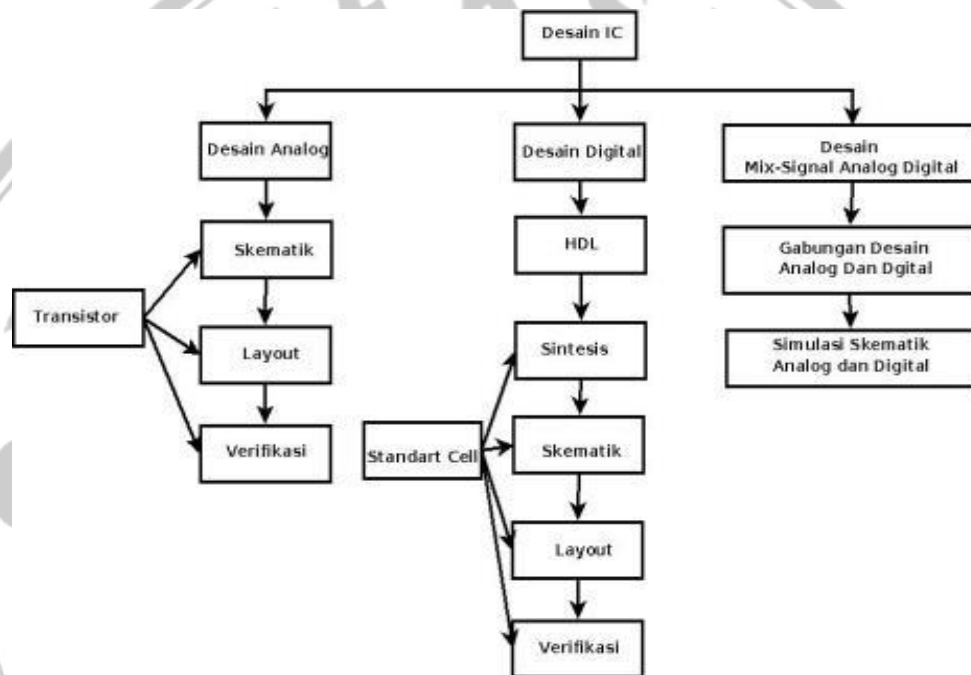
Gambar 1. 7 Alur Desain IC Lengkap[1] [4].

1.3. Metode Desain IC

Metode desain IC didasarkan dari dua jenis desain elektronika, yaitu desain analog dan desain digital. Desain analog IC pada implementasinya, desainer langsung menggunakan transistor untuk mendesain rangkaian. Istilah desain langsung menggunakan transistor MOS disebut dengan “*primitive Design*”. Sedangkan desain digital pada implementasinya, desainer tidak langsung menggunakan transistor MOS dalam mendesain, melainkan menggunakan “*standart Cell*” yang telah disediakan oleh vendor teknologi cmos. *Standart cell* merupakan file-file yang berisi gerbang-gerbang logika dan blok-blok rangkaian digital. Dalam *Standart cell* sudah berisi

skematik, simbol dan layout. Gambar 1.8 akan menjelaskan perbedaan dari desain IC analog dan desain Digital.

Selain desain analog dan digital, desain IC juga dapat dilakukan dengan metode penggabungan desain. Penggabungan desain analog dan digital merupakan desain penggabungan sinyal diantara kedua jenis desain tersebut. Metode penggabungan ini disebut dengan “*Analog Digital Mixed-Signal*” (ADMS). Pada modul ini hanya menjelaskan desain IC CMOS dengan metode desain analog.



Gambar 1.8 Alur Desain IC Lengkap[1]

1.4. Perangkat Lunak Desain IC CMOS

Software desain IC CMOS dalam beberapa sumber mengkategorikan ke dalam perangkat lunak *Computer Aided Design* (CAD) dan ada pula yang mengkategorikan kedalam perangkat luna *Electronic Design Automation* (EDA). Dalam dunia akademik *Software* Desain IC CMOS dikenal dengan Perancangan Elektronika Berbantuan Komputer.

Perkembangan saat ini *software* desain IC CMOS proses desain skematik dan layout dapat dilakukan oleh user. Sehingga user dapat dengan sendirinya membuat layout yang diinginkan dengan menggunakan *software* desain IC CMOS. Kemudian proses mencetak kebentuk fisik adalah proses yang dilakukan oleh pihak pabrik. Penggunaan *Software* desain IC dilakukan

dengan menggunakan library teknologi. Kedua unsur ini selalu berkaitan antara teknologi dan *software*. Suatu *software* desain IC dapat menggunakan beberapa library teknologi.

Software desain IC CMOS dapat dibedakan menjadi 2 kategori berdasarkan tujuan tertentu untuk mendesain yaitu komersil (*commercial*) dan non komersial. *Software* desain IC CMOS komersial digunakan untuk tujuan menghasilkan produk untuk dijual dalam jumlah besar. *Software* ini digunakan oleh perusahaan-perusahaan yang bisnisnya di perangkat elektronik, dari sisi harga *Software* desain IC CMOS komersial ini sangatlah mahal. Sedangkan *Software* desain IC CMOS non-komersial digunakan untuk tujuan edukasi, pengembangan dan penelitian.



Gambar 1.9 Logo Mentor Graphics

Mentor graphics adalah salah satu penyedia *Software* Desain CHIP. Vendor dengan Slogan “**The EDA Technology Leader**”. Perusahaan yang bergerak di bidang pengembangan perangkat elektronik dan penyedia *software* Desain VLSI. Beberapa produk *software* untuk desain diantaranya “**ICFlow & Pyxis**” untuk desain IC dan “**Calibre**” untuk Verifikasi IC. Dapat berjalan di sistem operasi windows dan linux. Untuk informasi lebih lengkap mengenai Mentor Graphics anda dapat mengunjungi situs <https://www.mentor.com/>.

1.5. Library Teknologi / Process Design Kits

Diperlukan *library* teknologi untuk menunjang proses desain IC. Teknologi pada *library* yang dimaksud adalah ukuran dari transistor yang digunakan untuk mendesain. Ukuran teknologi saat ini mencapai ukuran nanometer(nm) dan akan terus semakin mengecil. Selain ukuran teknologi CMOS juga diharapkan pada penggunaan atau konsumsi daya (*power*) yang semakin hemat.

Beberapa vendor atau perusahaan yang menyerdikan library teknologi berlomba-lomba dalam mengembangkan teknologi salah satu dari sisi ukuran dan konsumsi daya. Library teknologi yang dikeluarkan atau produk dari vendor disebut juga dengan *Process Design Kit* (PDK). Masing-masing vendor yang besar menyediakan beberapa versi untuk satu pdk. Sebagai contoh suatu vendor mengeluarkan pdk ukuran 0,35 μ m dan dapat digunakan di beberapa *software* desain IC. FoundryPDK yang akan digunakan pada modul ini adalah **Austria MicroSystem (AMS)**

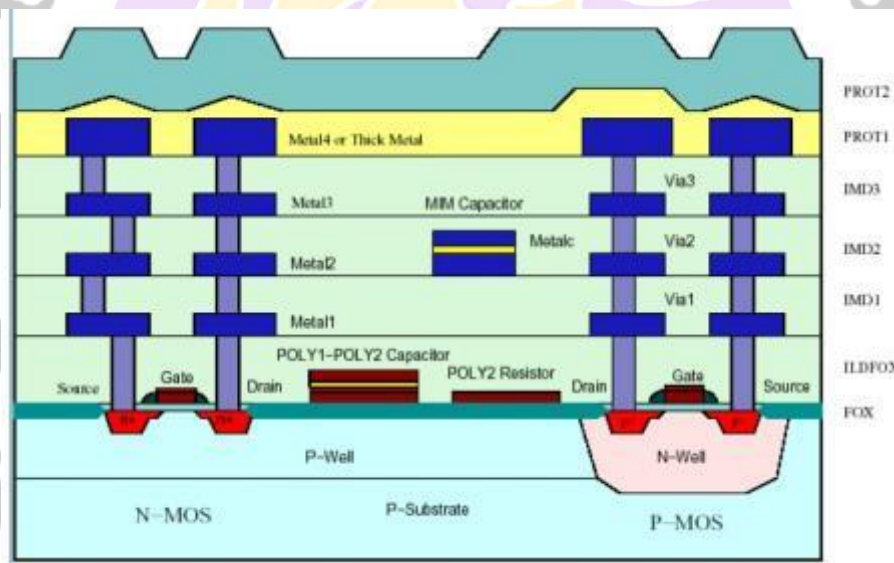


Gambar 1. 10 Logo AMS

Foundry yang menyediakan *proses design kits* (PDK) dengan nama HitKit (HK). AMS mendukung desain IC menggunakan *software* Mentor Graphics dan Cadence. HitKit AMS saat ini hadir dengan versi V.4 dengan beberapa variasi.

- **hitkit v4.14** untuk ukuran **0.18 μ m** dengan code proses (**aC18**).
- **hitkit v4.11** untuk ukuran **0.18 μ m** dengan code proses (**C18,H18**)
- **hitkit v4.10** untuk ukuran **0.35 μ m** dengan code proses (**C35,H35,S35**)

Untuk susunan MOS yang digunakan pada teknologi AMS 0.35 μ m adalah tampak seperti pada gambar 1.11. Terlihat bahwa susunan MOS mempunyai 4 metal, yaitu metal 1, metal 2, metal 3 dan metal 4 dan 2 polysilicon (poly), yaitu poly 1 dan poly 2. Untuk menghubungkan antara poly-metal, metal-metal dan poly-poly digunakan via dan contact.



Gambar 1.11 Susunan MOS pada teknologi AMS 0.35 μ m

1.6. Pabrikasi IC CMOS

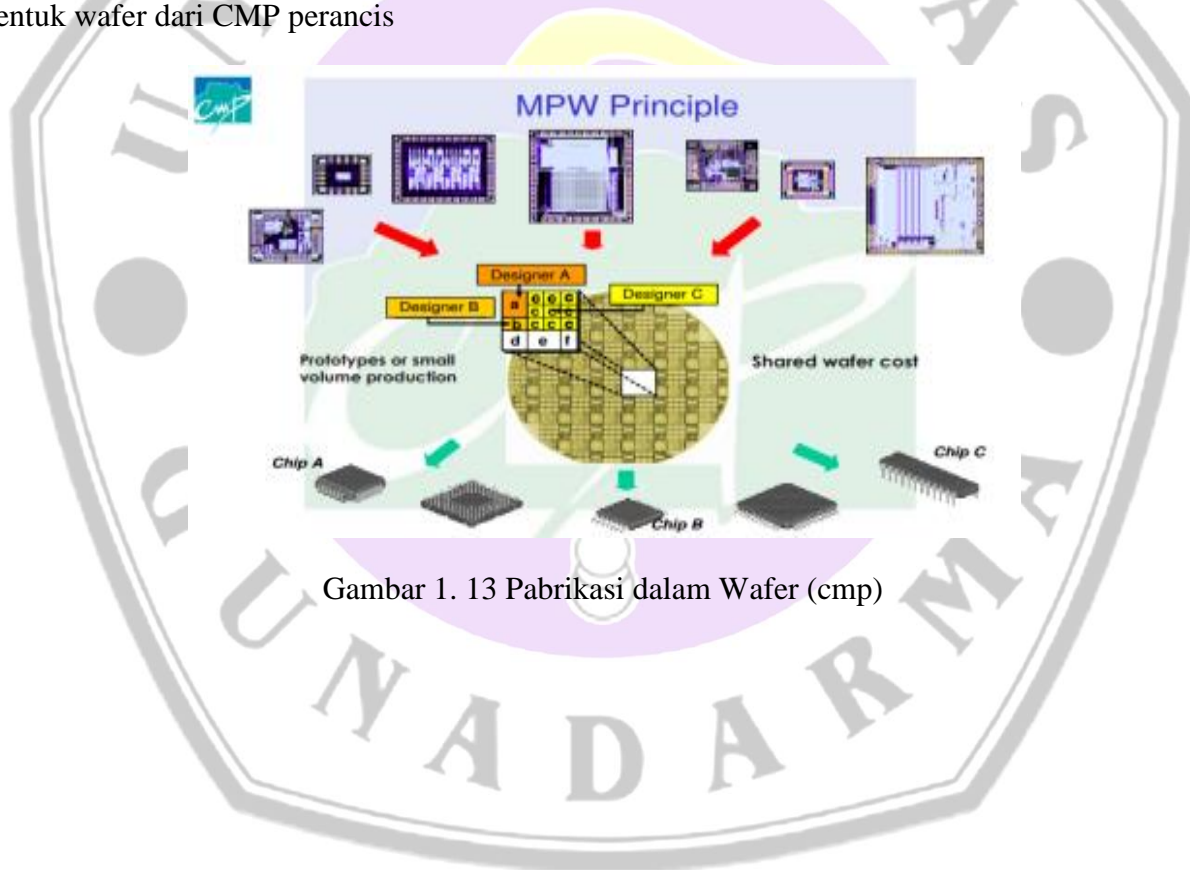
Terdapat beberapa pabrik yang memproduksi IC CMOS untuk produksi skala kecil maupun skala besar atau industry.



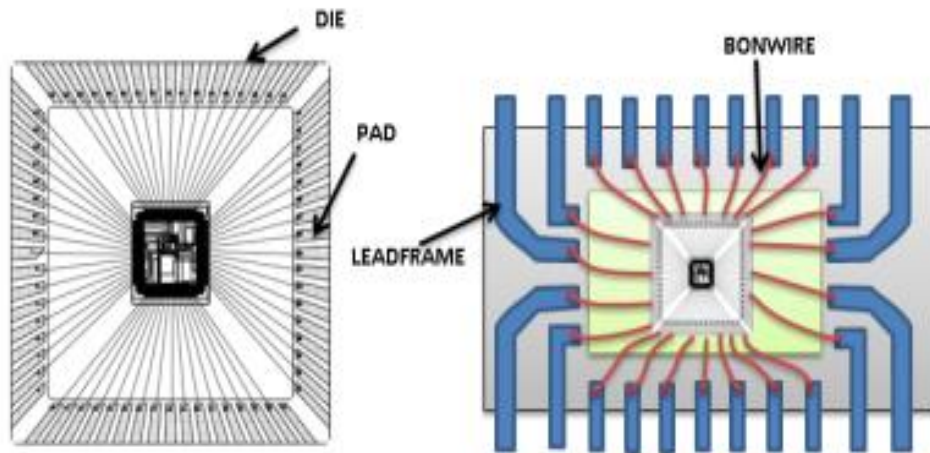
Gambar 1. 12 Logo CMP

Pabrik IC yang terletak di negara perancis, berdiri sejak 1981. Jenis yang dapat diproduksi antara lain IC, Photonic IC, dan MEMS. Teknologi yang dapat diproduksi antara lain CMOS, SiGe, BiCMOS, HV-CMOS, SOI, MEMS, dan 3D-IC. Selain itu cmp menyediakan layanan penghubung dengan vendor *library* teknologi bagi pengguna yang ingin menggunakan *library* teknologi (*design kits*). CMP bekerja sama dengan vendor-vendor teknologi atau *foundry* diantaranya AMS, STMicroelectronics, dan MEMSCAP. Untuk Teknologi CMOS dengan ukuran $0,35\mu\text{m}$ harga yang ditawarkan $650\text{€}/\text{m}^2$. Informasi lebih lengkap dapat mengunjungi situs <http://cmp.imag.fr/>.

Silicon merupakan bahan utama untuk membuat IC. Silicon akan dibentuk menjadi lapisan berbentuk “**wafer**”. Rangkaian IC CMOS yang diproduksi akan diletakkan pada lingkaran disebut dengan “**thin film**”. Satu wafer terdiri dari banyak desain yang akan diproduksi, metode ini disebut juga dengan “**Shared Wafer cost**”. *Shared Wafer cost* digunakan untuk produksi untuk prototipe atau produksi skala kecil bukan untuk produksi masal. Gambar 1.13 akan memvisualisasikan bentuk wafer dari CMP perancis



Gambar 1. 13 Pabrikasi dalam Wafer (cmp)



Gambar 1. 14 Packaging Layout Ke CHIP model DIL.



Bab 2. Pengenalan Mentor Graphics Software

2.1. Sistem Operasi Linux

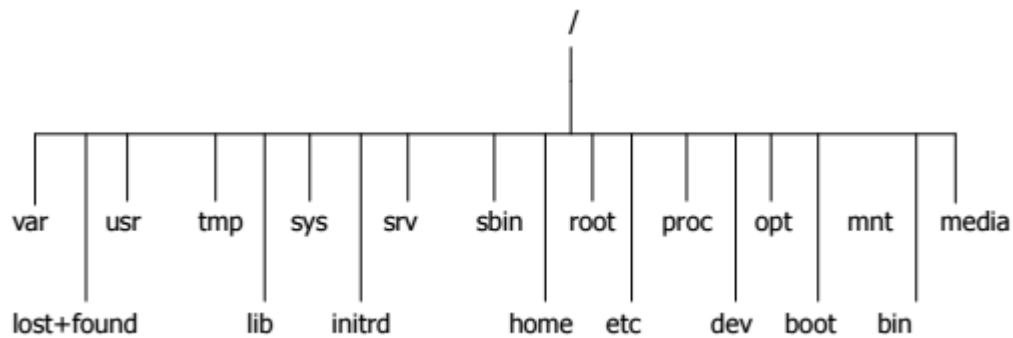
A. Tujuan : Untuk mempelajari, memahami dan mengeksplorasi sistem operasi linux Opensuse 11.4.

B. Kemampuan : - Mampu mengoperasikan sistem operasi Opensuse 11.4
- Mampu mengenal organisasi file di sistem operasi Opensuse 11.4

C. Dasar Teori :

Organisasi File

Sistem file pada OpenSuse sama dengan sistem file distribusi Linux lainnya. Sistem file dibentuk secara hirarki atau menyerupai pepohonan (tree). Dimulai dari direktori **root** yang biasanya disimbolkan dengan “/”, kemudian dilanjutkan dengan sub direktori lainnya seperti pada gambar 2.1.



Gambar 2.1 Hirarki Sistem File Linux

D. Kegiatan :

1. Masuk ke desktop dengan username dan password



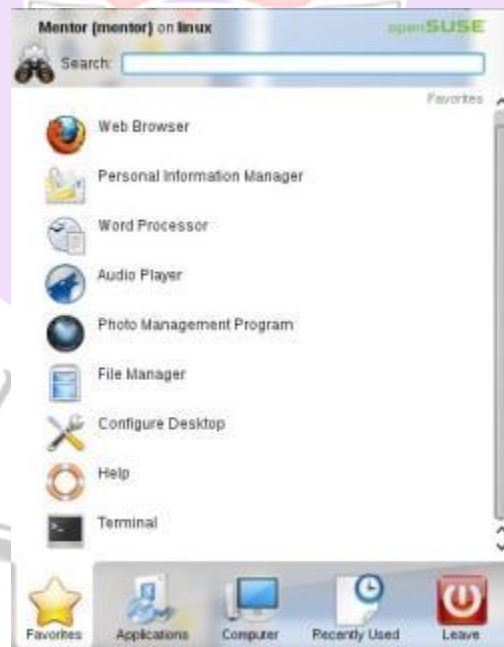
2. Tampilan desktop Opensuse 11.4 Versi KDE



3. Panel (Taskbar) desktop Opensuse 11.4 Versi KDE



4. StartMenu desktop Opensuse 11.4 Versi KDE



5. Terminal (Shell) desktop Opensuse 11.4 Versi KDE

2.2. Menjalankan *software* Mentor Graphics

- A. Tujuan** : - Mempelajari dan memahami Perintah dasar Linux (CLI).
- Mempelajari dan memahami langkah-langkah menjalankan Mentor Graphics.
- B. Kemampuan** : - Mampu Menjalankan perintah dasar CLI.
- Mampu menjalankan mentor graphics.
- C. Teori** :

Menjalankan *software* Mentor Graphics pada sistem operasi linux disesuaikan dengan dasar sistem sistem operasi linux yaitu *Command Line Interface* (CLI) atau menjalankan aktivitas sistem dengan perintah/instruksi seperti pada *Command Prompt* (CMD) pada sistem operasi windows namun berbeda sintak perintah. Beberapa perintah dasar linux diantaranya “ls”, “cd”, “exit”, “su”, dan “pwd”. Desain IC menggunakan *software* Mentor Graphics yang disebut ICFlow, dimana terdiri dari tiga bagian utama *software* yaitu **ICStudio** untuk manajemen desain, **Design Architect-IC** untuk mendesain Skematik dan ICStation untuk desain Layout.

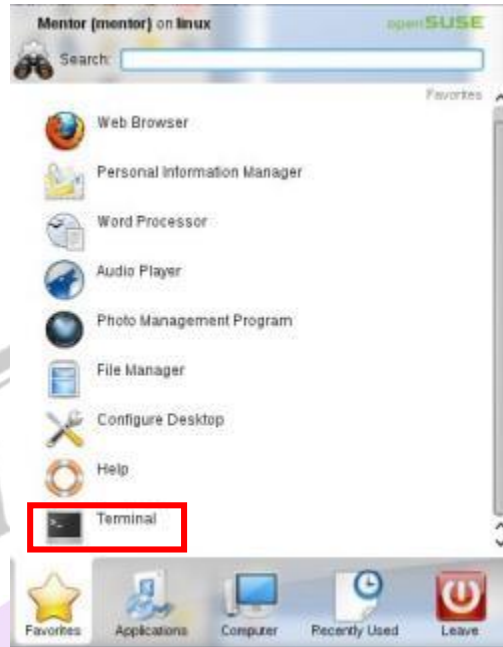
D. Kegiatan

D.1. Ekplorasi Opensuse

1. Akses OpenSuse menu.



2. Klik / buka icon Terminal.



3. Ketikkan perintah “ls” untuk menampilkan list file dan direktori.

```
File Edit View Bookmarks Settings Help
mentor@linux-9hdh:~> ls
ada.proj      datax.sh      desainku.proj
bin           datax.txt     Desktop
coba.proj     demo.proj     Documents
mentor@linux-9hdh:~> █
```

4. Ketikkan perintah “cd” untuk masuk ke dalam direktori dan ketikkan “cd..” untuk keluar direktori.

```
mentor@linux-9hdh:~> cd Documents
mentor@linux-9hdh:~/Documents> cd..
```

5. Ketikkan perintah “su” untuk masuk ke dalam direktori root dan ketikkan “exit” untuk keluar direktori root.

```
mentor@linux-9hdh:~> su
Password:
linux-9hdh:/home/mentor # exit
exit
mentor@linux-9hdh:~> █
```

6. Ketikkan perintah “pwd” untuk mengetahui posisi direktori yang sedang diakses

```
mentor@linux-9hdh:~> pwd
/home/mentor
mentor@linux-9hdh:~> █
```

D.2. Menjalankan ICStudio

1. Akses OpenSuse menu
2. Klik / buka icon Terminal
3. Ketikkan perintah “su -l” untuk mengakses root

```
File Edit View Bookmarks Settings Help
mentor@linux-9hdh:~> su -l
Password:
linux-9hdh:~ # █
```

4. Ketikkan perintah “./mg-source/ganti-mac-addr.sh”

```
linux-9hdh:~ # ./mg-source/ganti-mac-addr.sh
linux-9hdh:~ # █
```

5. Ketikkan perintah “exit”

```
linux-9hdh:~ # exit
logout
mentor@linux-9hdh:~> █
```

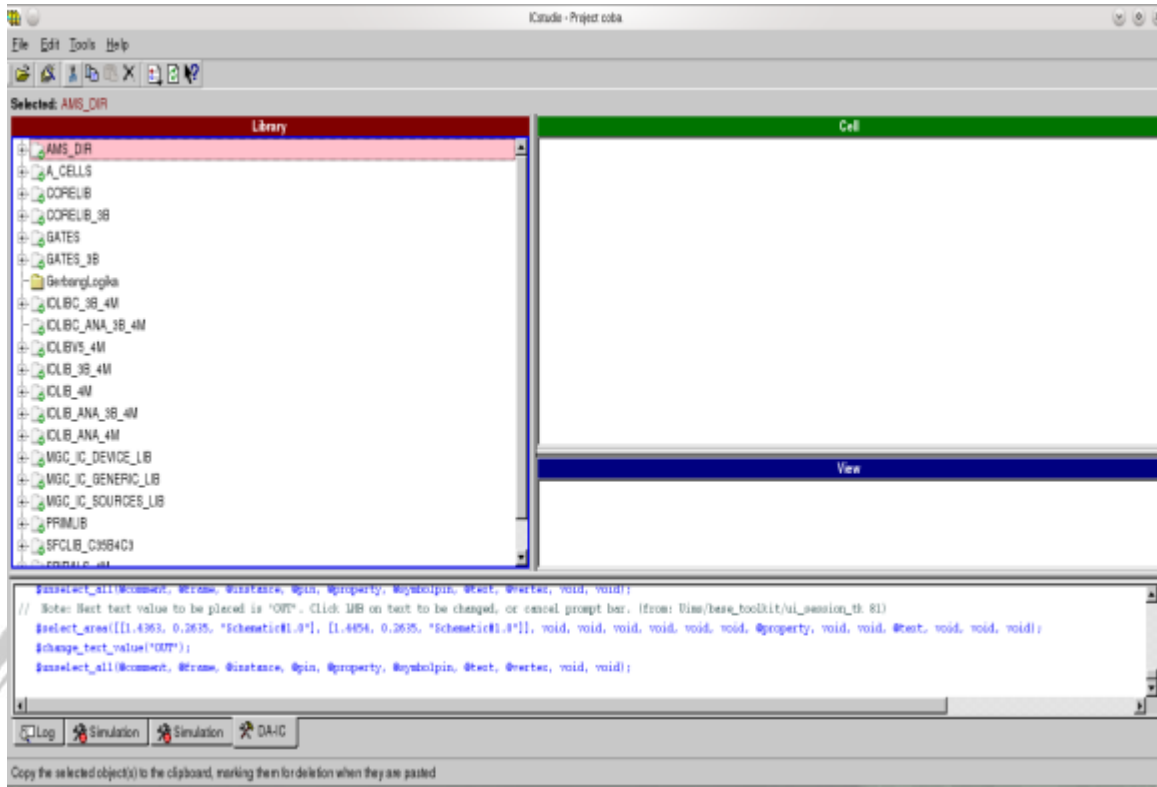
6. Ketikkan perintah “lmgrd” untuk mengaktifkan lisensi

```
mentor@linux-9hdh:~> lmgrd
10:39:32 (mgld) zncdcfv zncdcfvcomp zncdcfvmc
10:39:32 (mgld) zncmpccl zncmpcdc zncmpckl
10:39:32 (mgld) zncmpcsl zncconfirm zncw
10:39:32 (mgld) zncwcompiler zncwbase znformal
10:39:32 (mgld) znformalbase znformalseed znfx
10:39:32 (mgld) znfxsim zngui znhiercdc
10:39:32 (mgld) znhiercdcilm znpacdc znprove
10:39:32 (mgld) znpsl znramgen znreplaysim
10:39:32 (mgld) znsearch znseed znsim
10:39:32 (mgld) znsva znsynthnl znview
10:39:32 (mgld)
10:39:32 (mgld) Licenses are case sensitive for mgld
10:39:32 (mgld)
10:39:32 (mgld) EXTERNAL FILTERS are OFF
10:39:32 (mgld)
10:39:32 (lmgrd) mgld using TCP-port 55129
```

7. Memulai proyek dengan perintah “ams_icstudio -project nama_project -tech c35b4c3”

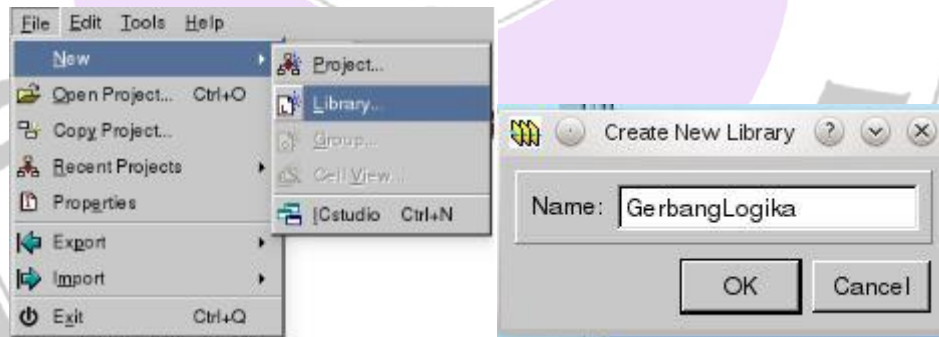
```
mentor@linux-9hdh:~> ams_icstudio -project coba -tech c35b4c3
```

8. Akan tampil layar ICStudio.

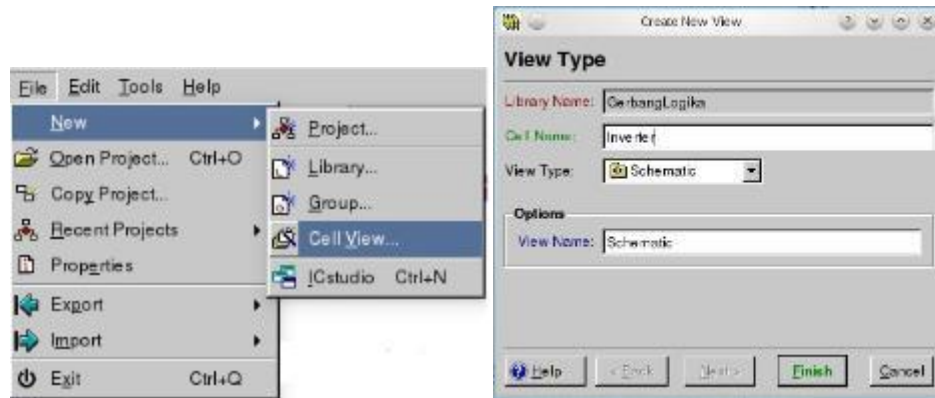


D.3. Menjalankan Design Architech-IC

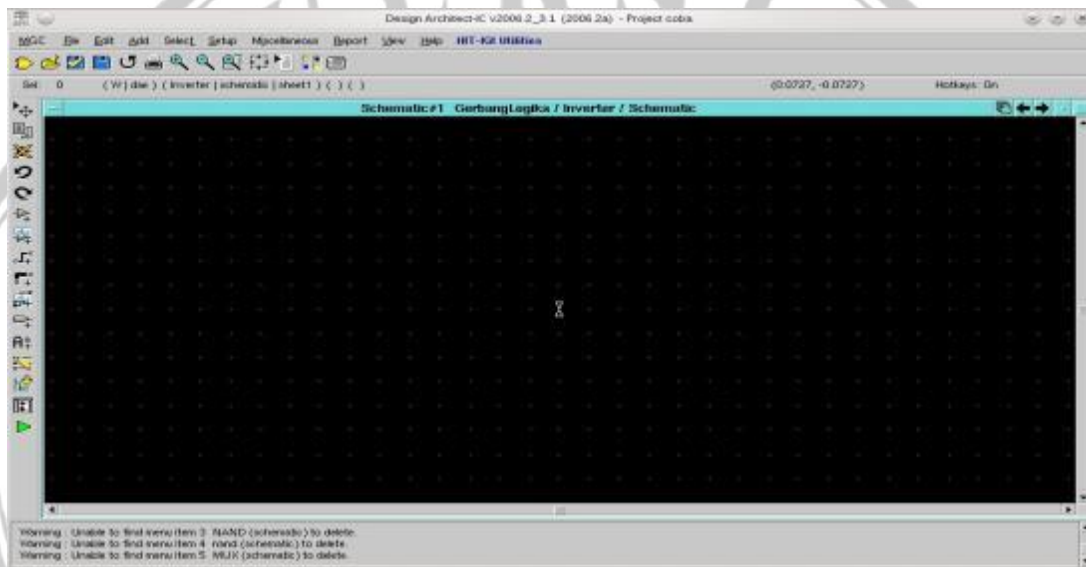
1. Membuat *library* dengan klik “file>New>Library” atau klik kanan pada layar. Isikan nama library “GerbangLogika”. Kemudian tekan tombol **OK**



2. Membuat Cell dengan klik “file>New>CellView”. Isikan **Cell Name = Inverter** dan **View Type = Schematic**. Kemudian tekan tombol **finish**

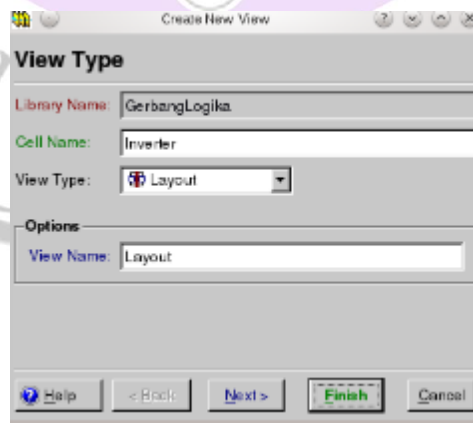


3. Tampilan Layar skematik

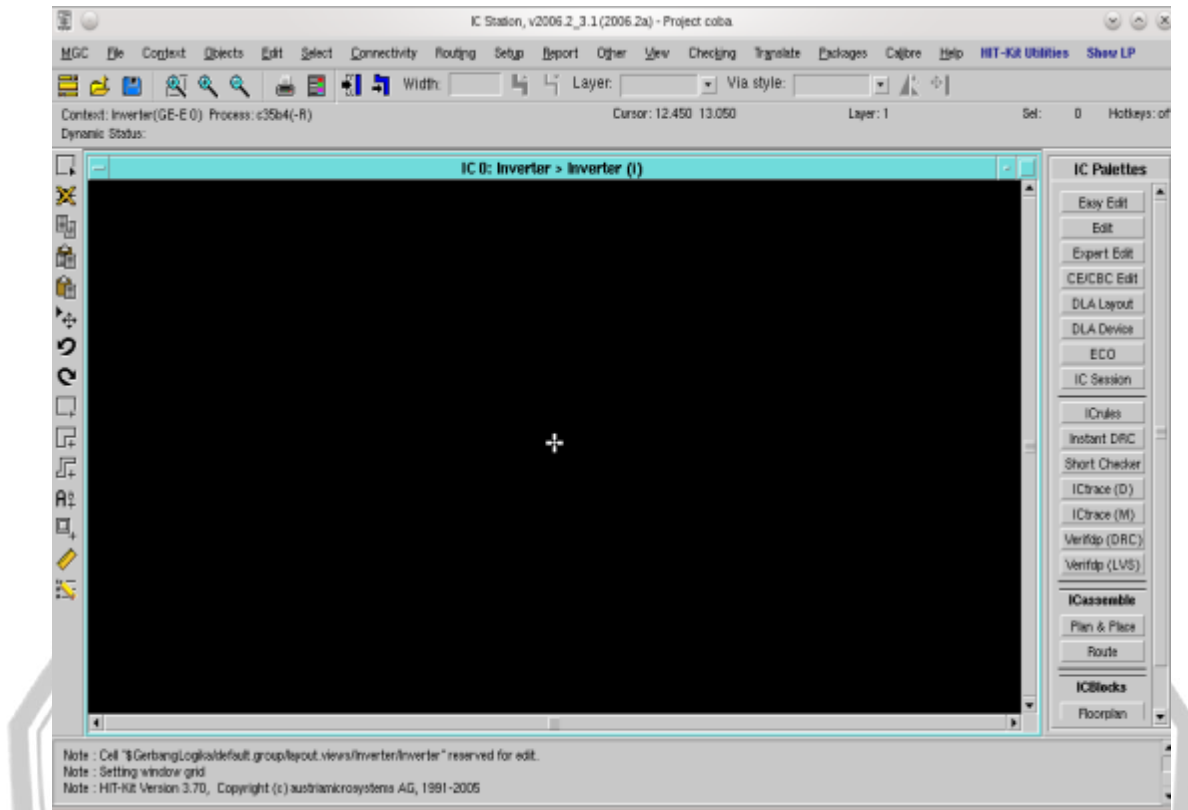


D.4. Menjalankan ICStation

1. Membuat Cell dengan klik “file>New>CellView”. Isikan **Cell Name = Inverter** dan **View Type = Layout**. Kemudian tekan tombol **finish**.



2. Akan tampil Layar layout



Bab 3. Skematik Desain Analog

3.1. Membuat Skematik Desain Analog Gerbang Logika

A. Tujuan : - Membuat rangkaian dasar gerbang logika (Gerbang dasar) dari transistor cmos menggunakan skematik

B. Kemampuan : - Mampu membuat dan mensimulasi rangkaian logika dasar.

C. Teori :

Pada prinsip rangkaian dasar CMOS, merupakan penggabungan atau kombinasi dari dua atau lebih jenis transistor MOS yang disebut juga dengan komplementer. Maka Percobaan pertama adalah membuat Gerbang Logika. Diawali membuat gerbang Inverter (Not) yang terdiri dari dua transistor MOS yaitu PMOS dan NMOS, kemudian dilanjutkan dengan membuat gerbanglogika lainnya.

NAMA	SIMBOL GRAFIK	FUNGSI ALJABAR	TABEL KEBENARAN															
AND		$X = A \cdot B$	<table border="1"> <tr><td>A</td><td>B</td><td>X</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	X	0	0	0	0	1	0	1	0	0	1	1	1
A	B	X																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR		$X = A + B$	<table border="1"> <tr><td>A</td><td>B</td><td>X</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	X	0	0	0	0	1	1	1	0	1	1	1	1
A	B	X																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
INVERTER		$X = A'$ $X = \overline{A}$	<table border="1"> <tr><td>A</td><td>X</td></tr> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table>	A	X	0	1	1	0									
A	X																	
0	1																	
1	0																	
BUFFER		$X = A$	<table border="1"> <tr><td>A</td><td>X</td></tr> <tr><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td></tr> </table>	A	X	0	0	1	1									
A	X																	
0	0																	
1	1																	
NAND		$X = \overline{A \cdot B}$	<table border="1"> <tr><td>A</td><td>B</td><td>X</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	X	0	0	1	0	1	1	1	0	1	1	1	0
A	B	X																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOR		$X = \overline{A + B}$	<table border="1"> <tr><td>A</td><td>B</td><td>X</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	X	0	0	1	0	1	0	1	0	0	1	1	0
A	B	X																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
EXCLUSIVE OR (XOR)		$X = A \oplus B$ $X = \overline{A} \cdot B + A \cdot \overline{B}$	<table border="1"> <tr><td>A</td><td>B</td><td>X</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	X	0	0	0	0	1	1	1	0	1	1	1	0
A	B	X																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
EXCLUSIVE NOR (XNOR)		$X = \overline{A \oplus B}$ $X = \overline{\overline{A} \cdot B + A \cdot \overline{B}}$	<table border="1"> <tr><td>A</td><td>B</td><td>X</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	X	0	0	1	0	1	0	1	0	0	1	1	1
A	B	X																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

Gambar 3.1. Gerbang Logika

D. Kegiatan

D.1. Skematik Gerbang Not / Inverter

A. Buatlah rangkaian inverter yang terdiri transistor Pmos dan 1 transistor Nmos menggunakan menu “**HIT-KIT Utilities > AMS Library > Devices > MOS > Nmos/Pmos**” dan Letakkan pada layar kerja. Ukuran yang digunakan untuk kedua transistor adalah

- Instance Name = m_1 / m_2
- Transistor Length = 0,35 μ
- Width = 0,4 μ ,
- Number of Gate = 1
- Strip Width = 0,4 μ

The image illustrates the process of creating an inverter schematic in a CAD tool. It shows the navigation path through the AMS Library to select a MOS transistor. The final schematic shows two transistors, pmos4 and nmos4, connected in a common-source configuration. Handwritten annotations specify the parameters for both transistors: $w_{tot} = 0,4$ and $l = 0,35$.

AMS Library

- MGC Library
- AMS Library
- AMS Utility
- C35b4 Libraries**
- Devices
- Devices-HP
- SPIRALS_4M
- CORELIB
- CORELIB_3B
- IOLIB_4M
- IOLIB_3B_4M
- IOLIBC_3B_4M
- IOLIBV5_4M
- IOLIB_ANA_4M
- IOLIB_ANA_3B_4M
- IOLIBC_ANA_3B_4M
- CELLS
- Other Libraries

Devices

- MOS
- BIPOLAR
- RESISTORS
- CAPACITORS
- DIODE
- MOS**
- nmos4
- nmos4
- nmosh
- nmosmh
- pmos4
- pmos4

austrainmicrosystems AG TRANSISTOR DEVICE
Technology: c35b4

AMS Transistor Type: nmos4

Instance Name: m_1

Transistor Width [μ]: 0,4

Strip Width [μ]: 0,4

Transistor Length [μ]: 0,35

Number of Gates : 1

Schematic#1 GerbangLogika / Inverter / Schematic

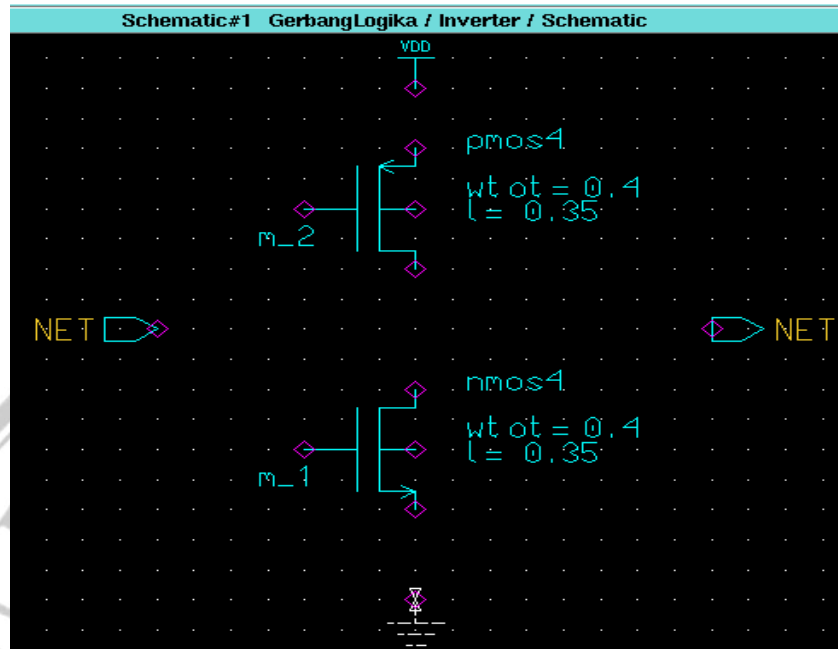
pmos4
 $w_{tot} = 0,4$
 $l = 0,35$

m_2

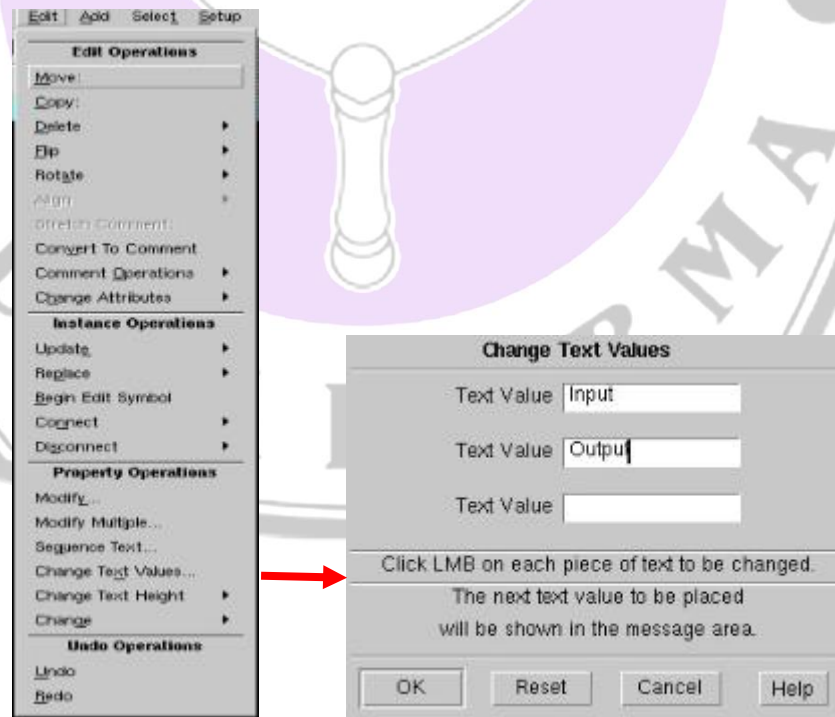
nmos4
 $w_{tot} = 0,4$
 $l = 0,35$

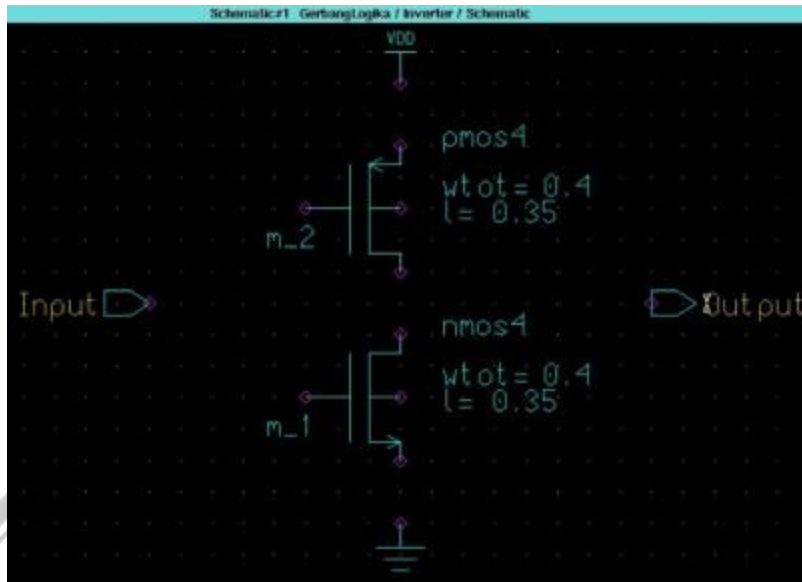
m_1

- B. meletakkan port-port masukan dan keluaran yaitu, Vdd, Ground, Input dan Output menggunakan “MGC Library > Generic Lib > Vdd / Ground / Portin / Portout”.

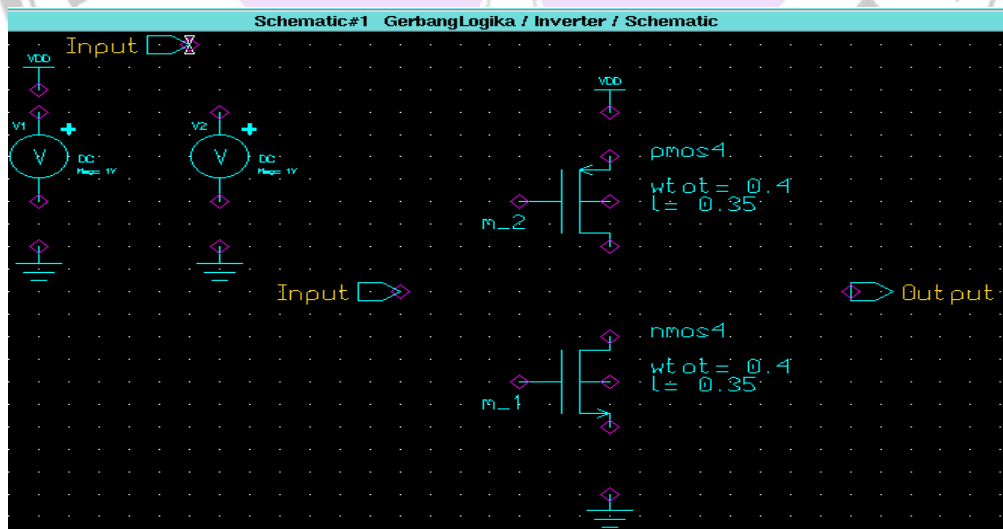
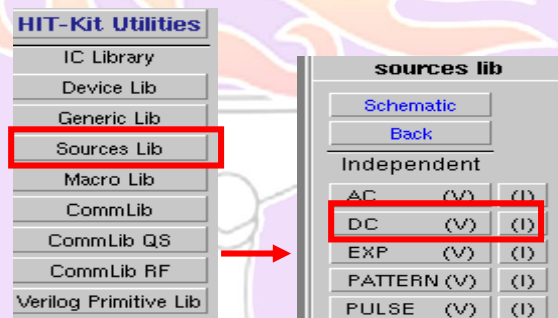


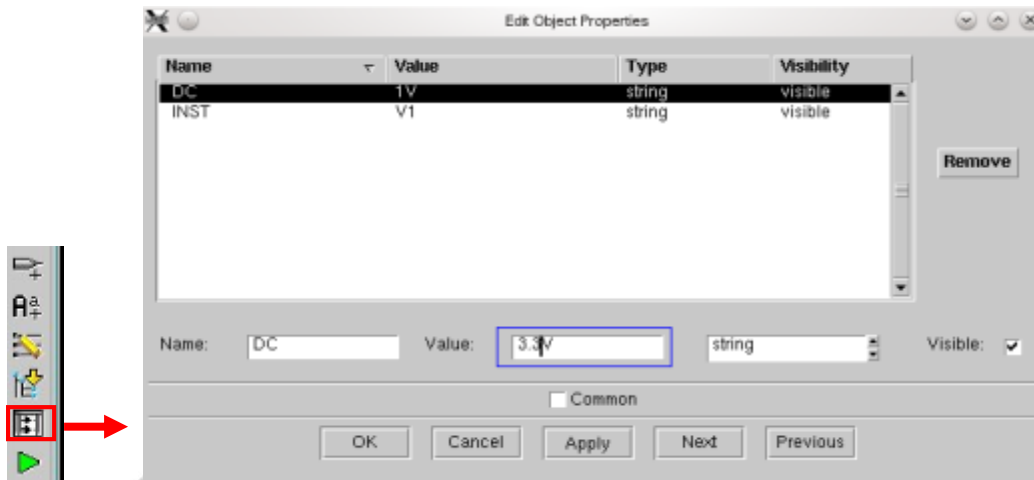
- C. Ubah label port “NET” masukan dan keluaran menjadi “Input” dan “Output” menggunakan menu “EDIT > Change Text Value”. Isikan “Input” dan “Output” bersamaan. Kemudian Klik di masing-masing Objek.



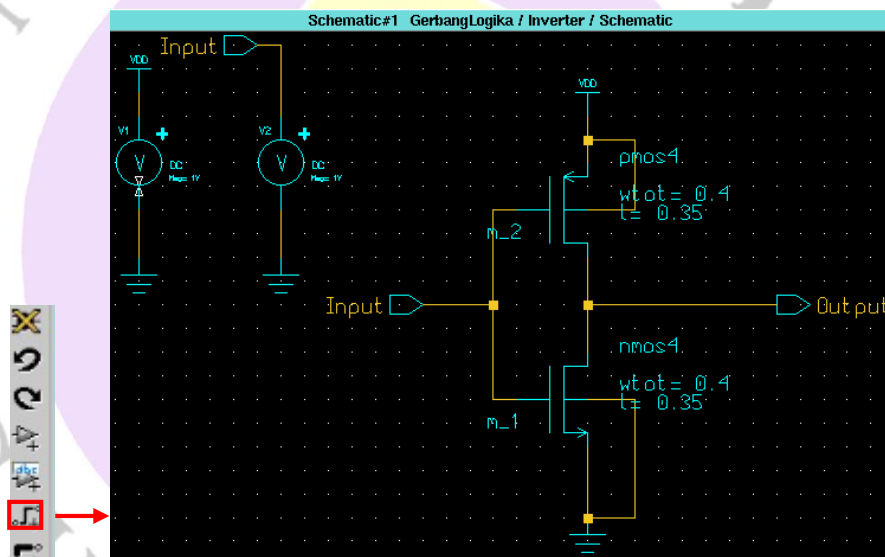


D. Meletakkan sumber tegangan menggunakan “Source Lib > DC” pada layar. Ubahlah tanggangan masukkan V1 menjadi 3,3 Volt menggunakan button “Edit Object Properties” pada toolbox. Name = DC, Value = 3,3 V. Ubahlah teganggn masukkan V2 menjadi Name = DC, Value = 3,3 V.





E. Sambungkan masing-masing objek/komponen menggunakan “add Wire” pada *toolbox*. Sampai disini rangkaian inverter telah dibuat kemudian dilanjutkan melakukan proses simulasi.



Bab 4. Simulasi Rangkaian Skematik Analog

4.1. Simulasi Skematik Desain Analog Gerbang Logika

A. Tujuan : Memahami dan mempelajari langkah-langkah Simulasi Rangkaian Dasar dari Gerbang Logika

B. Kemampuan : - Mampu melakukan simulasi menggunakan analisa DC dan Transient

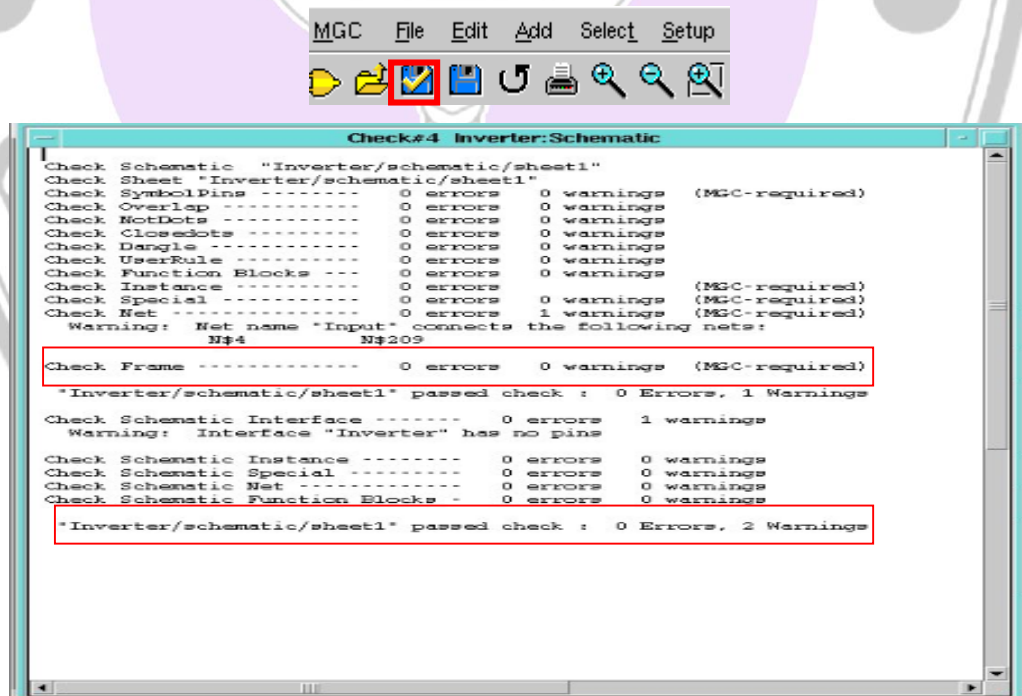
C. Teori :

Analisa DC menentukan titik operasi rangkaian dan kondisi stabil dari rangkaian analog sebelum analisa transient. Analisa transient menghitung keluaran transient sebagai fungsi waktu selama interval yang ditentukan oleh user.

D. Kegiatan

D.1. Simulasi Analisa DC

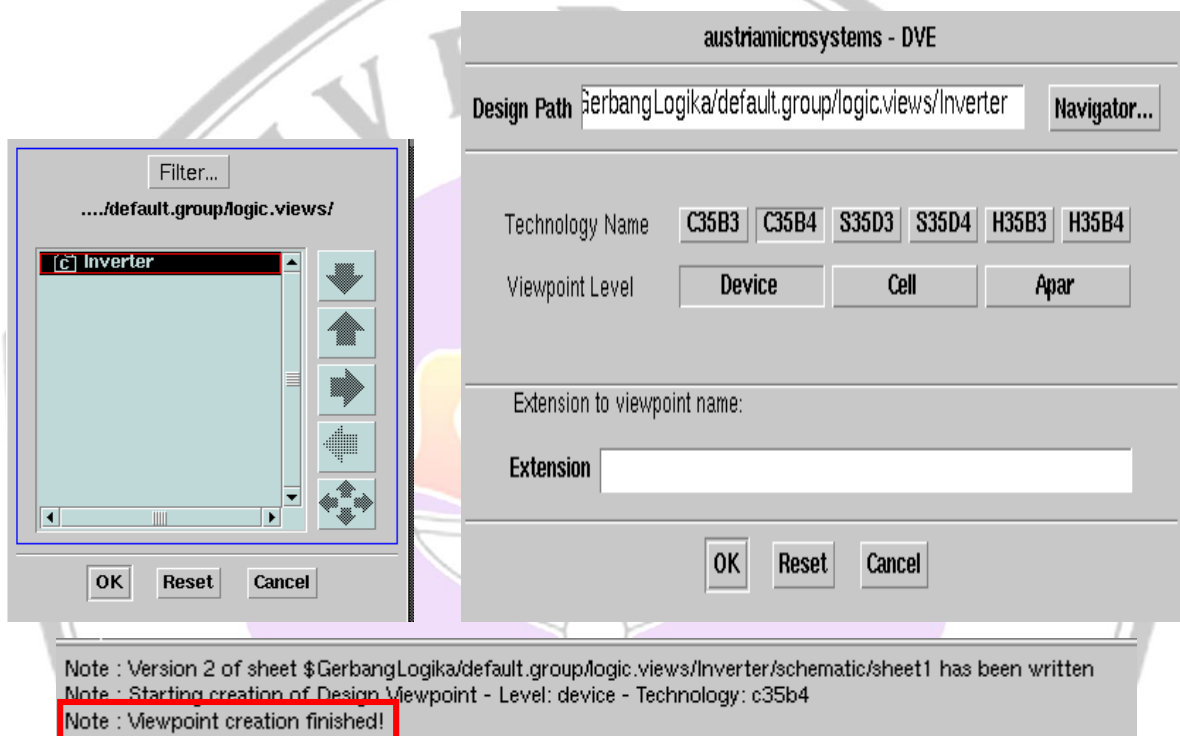
- Rangkaian skematik yang telah dibuat kemudian dilakukan pengecekan skematik menggunakan menu button **“Check and Save”** disebelah kiri button **save** atau menggunakan full check pada menu **“File > Check Schematic”**. Sehingga akan muncul hasil pengecekan. Jika terdapat **error** maka wajib diperbaiki, namun jika hanya terdapat **warning tanpa error** maka dapat diabaikan.



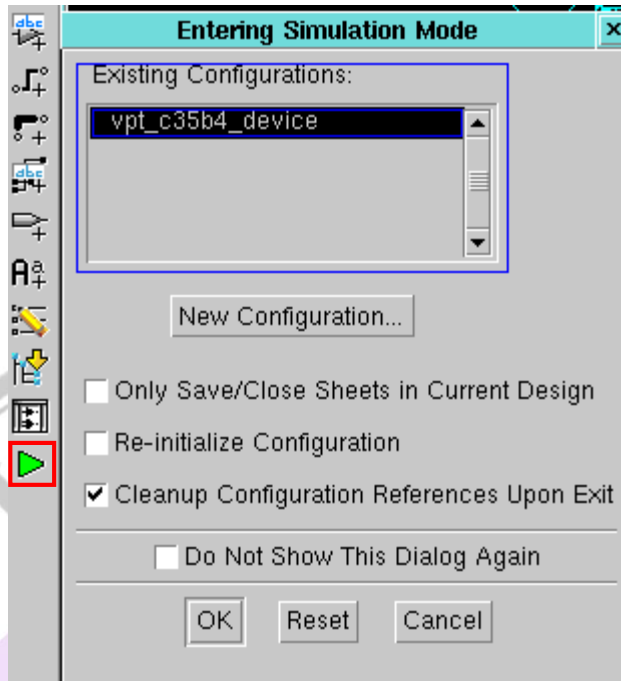
b. Membuat **Viewpoint** sebagai file konfigurasi simulasi menggunakan menu “**Hit-Kit Utilities > Create Viewpoint**”. Berikut konfigurasi **Viewpoint**

- **Design Path** = GerbangLogika.lib/default.grup/logic.view/Inverter
- **Technology Name** = C35B4
- **Viewpoint Level** = Device

Hasil dari konfigurasi ini adalah pesan yang ditunjukkan “**Viewpoint Creation finished**”.

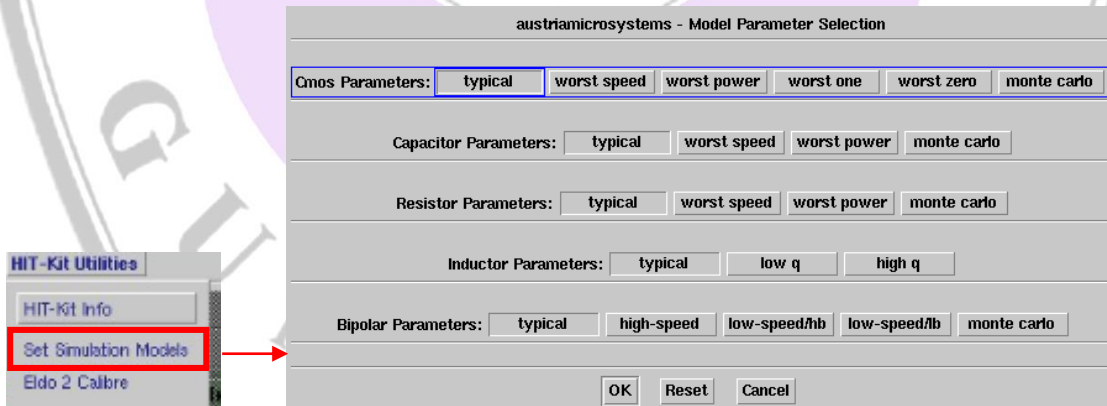


c. Membuka mode simulasi dengan mengklik button “**Enter Simulation Mode**” pada toolbox. Jika **Viewpoint** sebelumnya sukses maka akan muncul konfigurasi “**vpt_c35b4_device**”. Kemudian Klik OK, maka akan muncul layar simulasi.

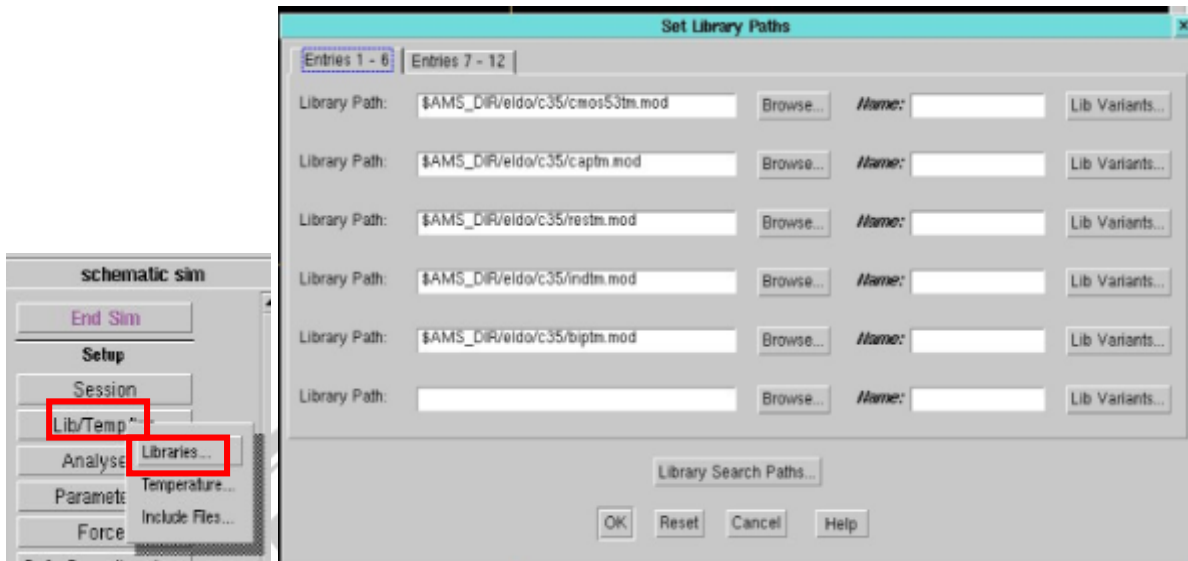


d. Menset model Simulasi menggunakan menu “**Hit-Kit Utilities > Set Simulation Models**”. Pada layar ini menggunakan parameter “**typical**”.

- **Cmos Parameters : typical**
- **Capacitor Parameters : typical**
- **Resistor Parameters : typical**
- **Inductor Parameters : typical**
- **Bipolar Parameters : typical**



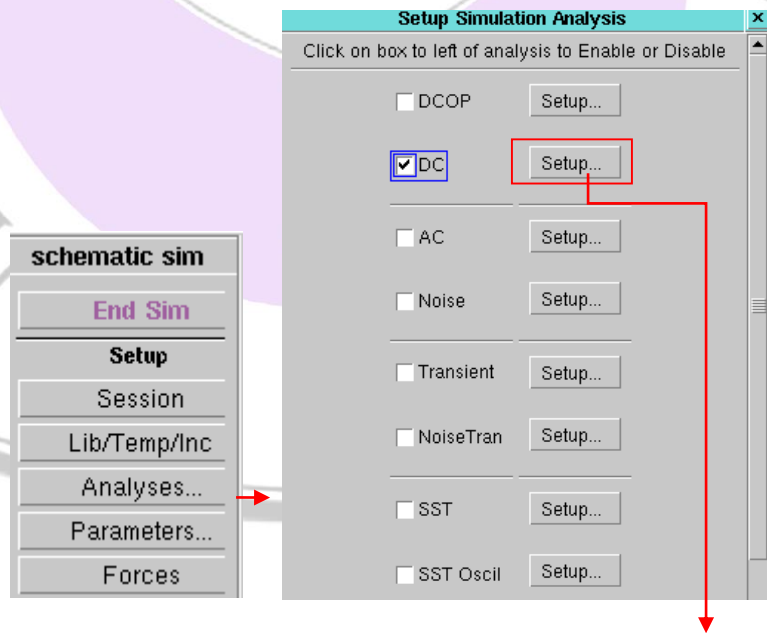
e. Jika set model simulasi berhasil maka kemudian memastikan **library c35** telah dimuat semua. Library **cmos53tm.mod**, **captm.mod**, **restm.mod**, **indtm.mod** dan **biptm.mod**.

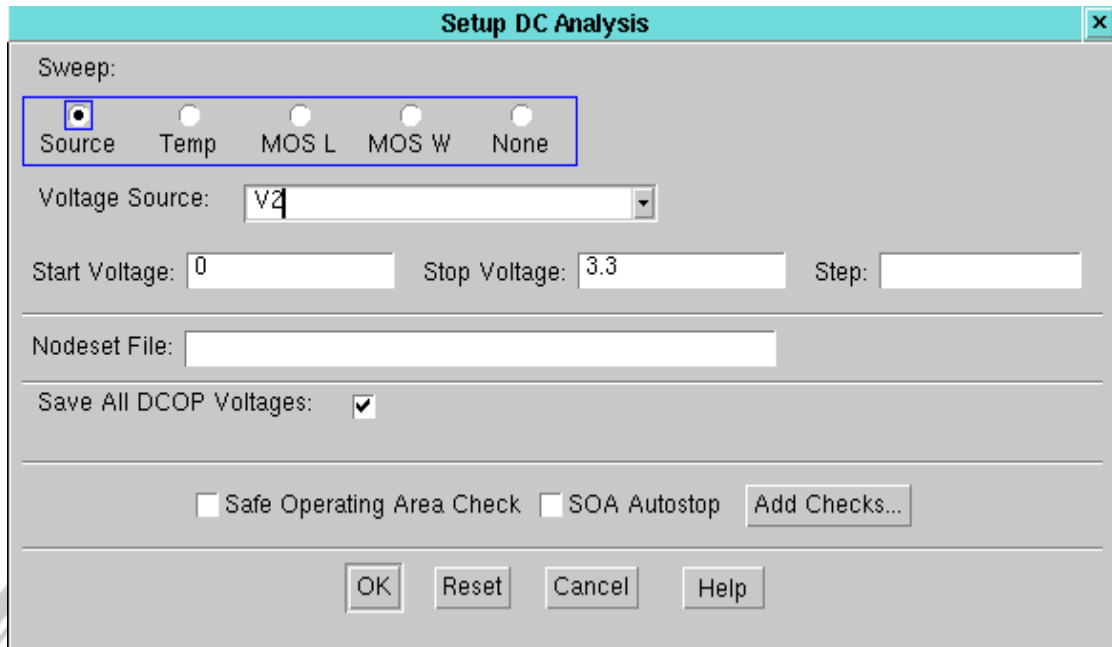


f. Membuat analisa menggunakan menu “**Analyses...**” pada menu Hit-Kit sebelah kanan layar. Berikan tanda ceklist pada DC dan klik button **Setup**. Pilih menu radiobutton **Source** dan konfigurasi sebagai berikut

- **Voltage Source** = V2 (Masukkan/Input)
- **Start Voltage** = 0
- **Stop Voltage** = 3.3

Klik Ok untuk memproses.

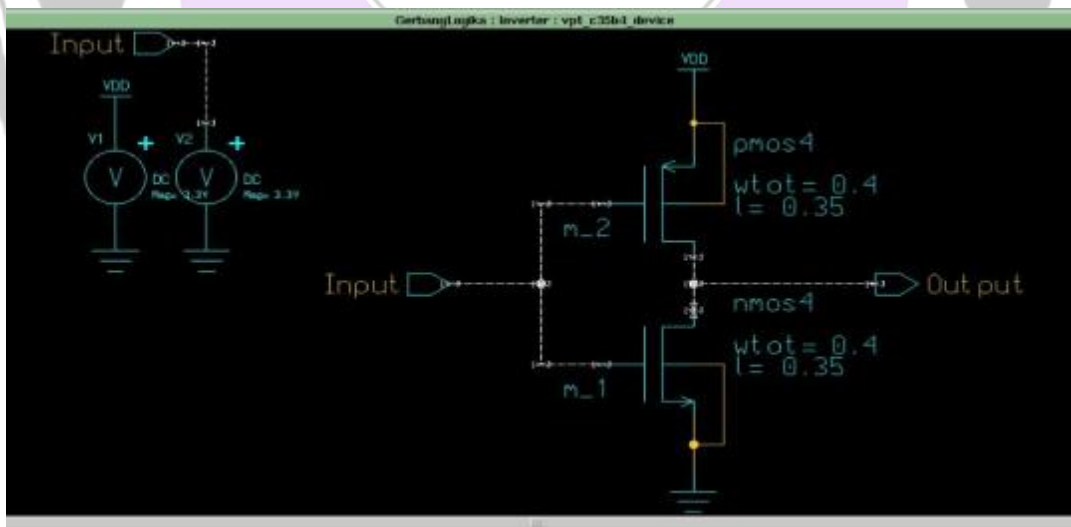


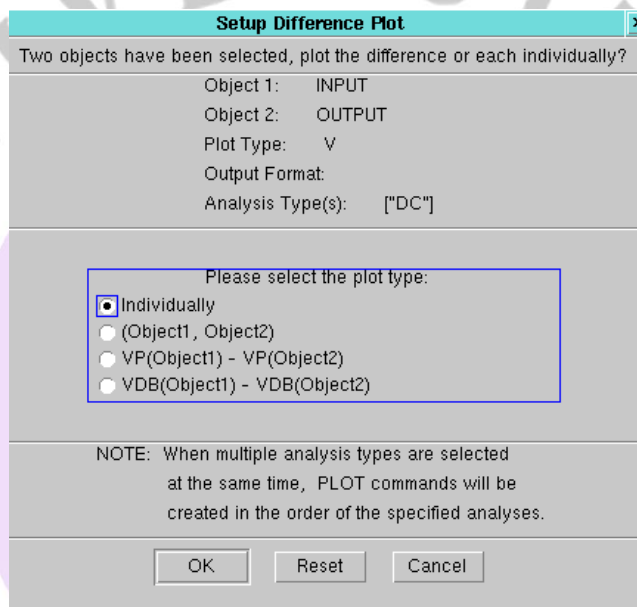
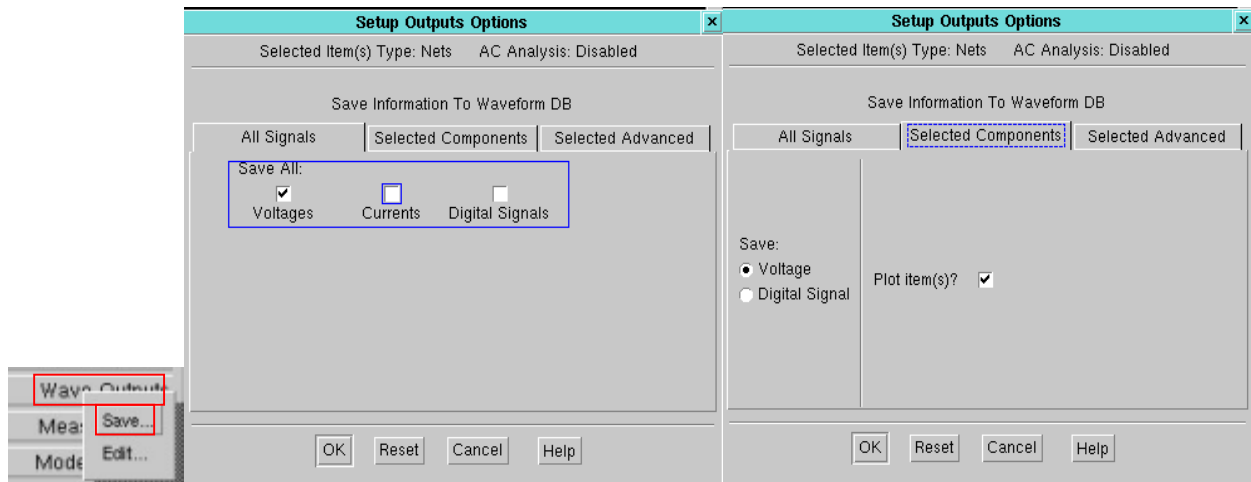


g. Menyet Tampilan grafik menggunakan menu “**Wave Outputs > save**” bersamaan dengan mengaktifkan port masukan dan keluaran.

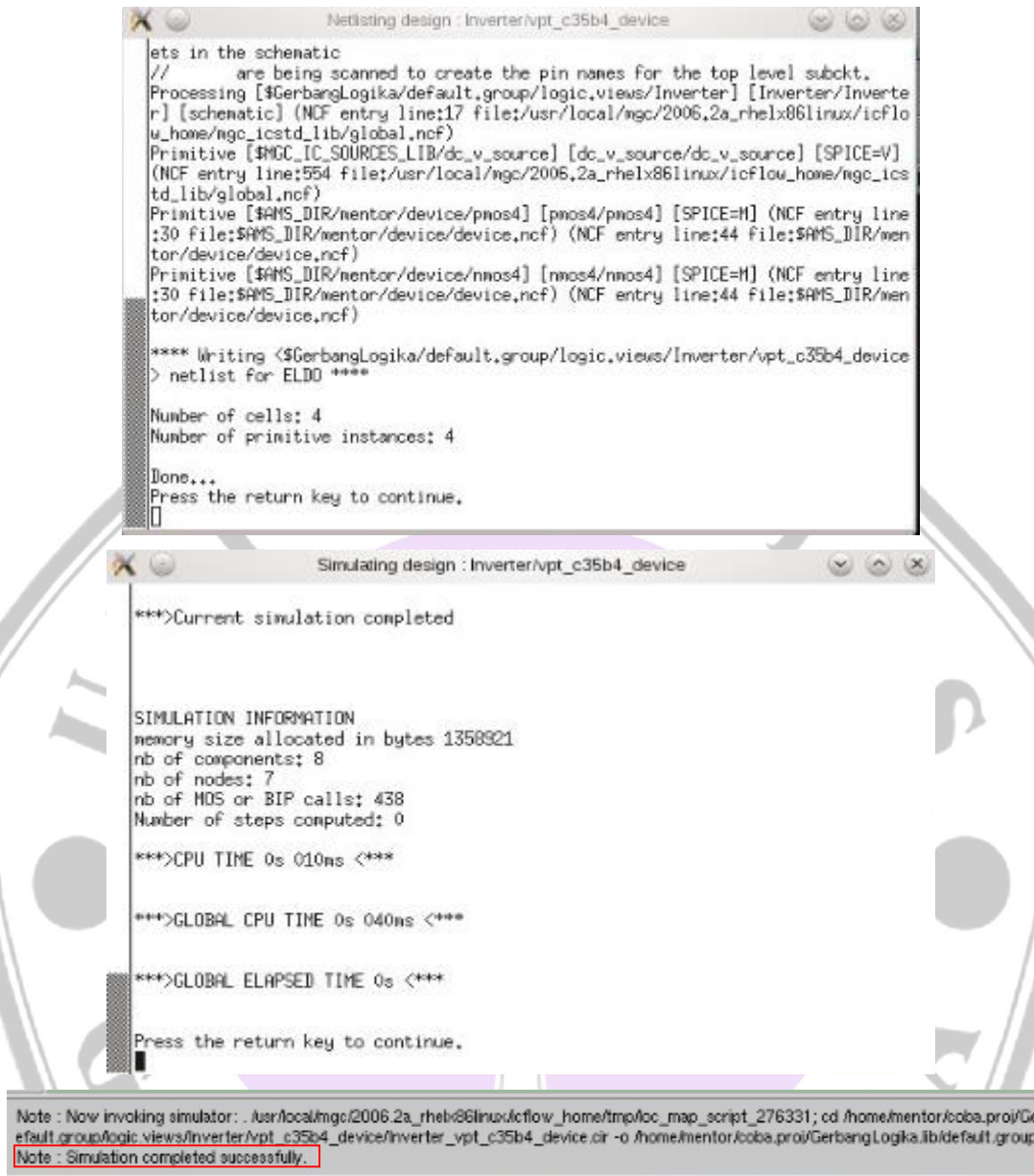
- **All Signal = Volgates**
- **Selected Components = Voltage dan Plot item(s).**

Pada layar **Setup Diffrence Plot** pilih **type plot = Individually** kemudian OK

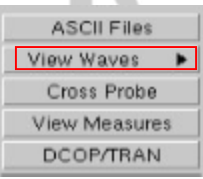


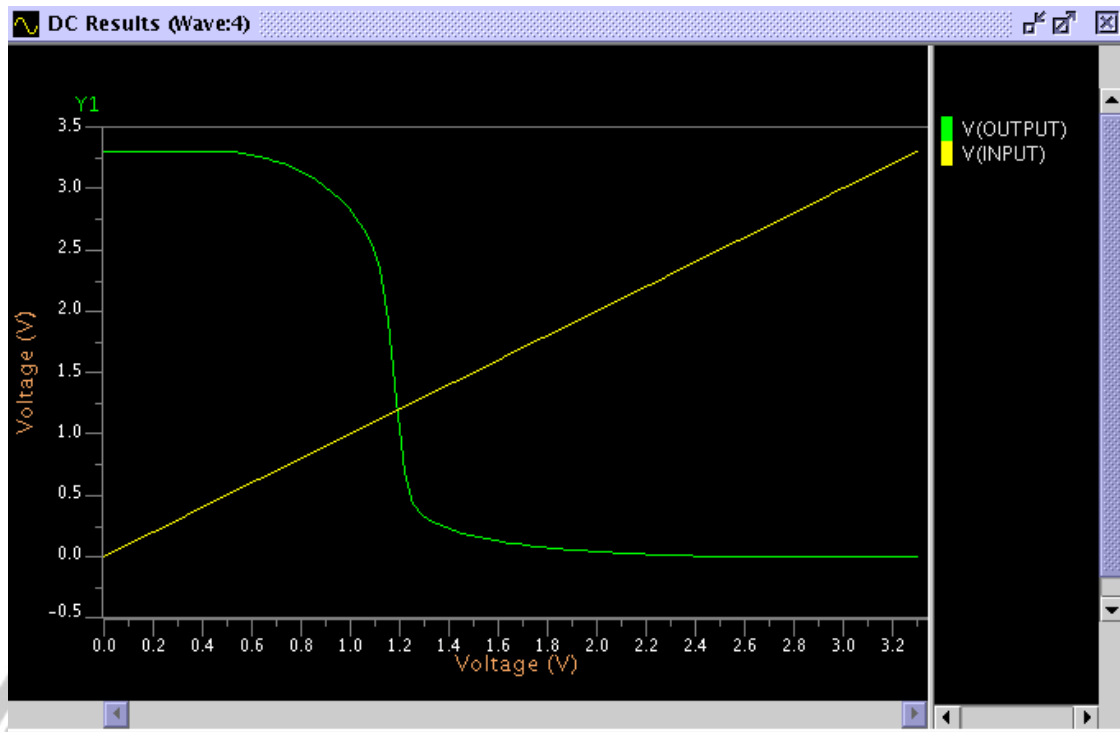


- h. Jalankan simulasi menggunakan Button **RUN ELDO**. Selama proses simulasi akan muncul 2 layar prompt sebagai indikator proses simulasi. Jika sukses maka akan muncul pesan **“Note : Simulation Completed Successfully”**.



i. Melihat hasil simulasi menggunakan button “View Waves”, akan muncul hasil grafik simulasi.





j. Melihat **Total Power Dissipation**, menggunakan menu ASCII Files > View Log.
Kemudian carilah **Total Power Dissipation**.

```

Notepad - /home/mentor/coba.proj/GerbangLogika.lib/c

```

NODE	VOLTAGE	NODE	VOLTAGE
INPUT	3.3000	OUTPUT	12.6366N
3000	OUTPUT	12.6366N	
VDD	3.3000		


```

VOLTAGE SOURCE CURRENT

```

NAME	CURRENT	VOLTAGE	POWER
V2	0.0000	3.3000	0.0000
V1	-3.3003P	3.3000	-10.8911P

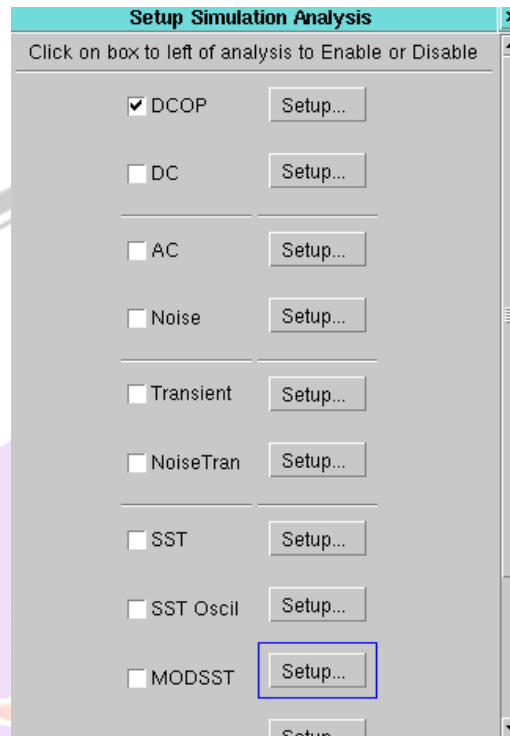
```

TOTAL POWER DISSIPATION: 10.8911P WATTS

```

D.2. Simulasi Analisa DC Operating Point

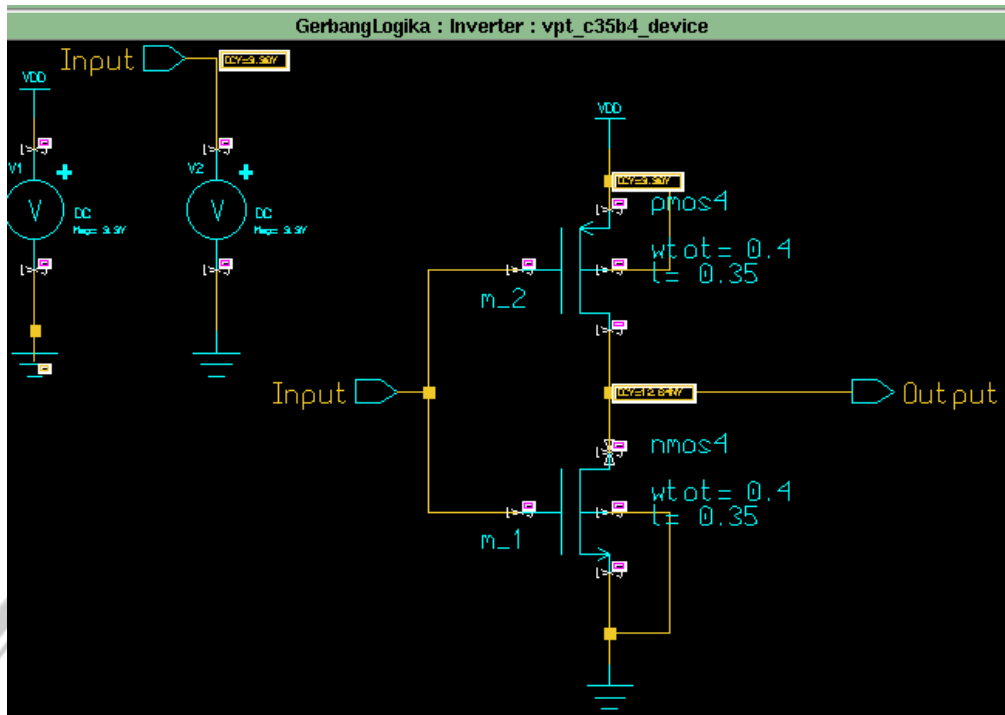
- a. Analisa yang digunakan adalah **Operating Point DC** pada menu **Analyses>DCOP**.
Centang **DCOP**. Kemudian Klik OK.



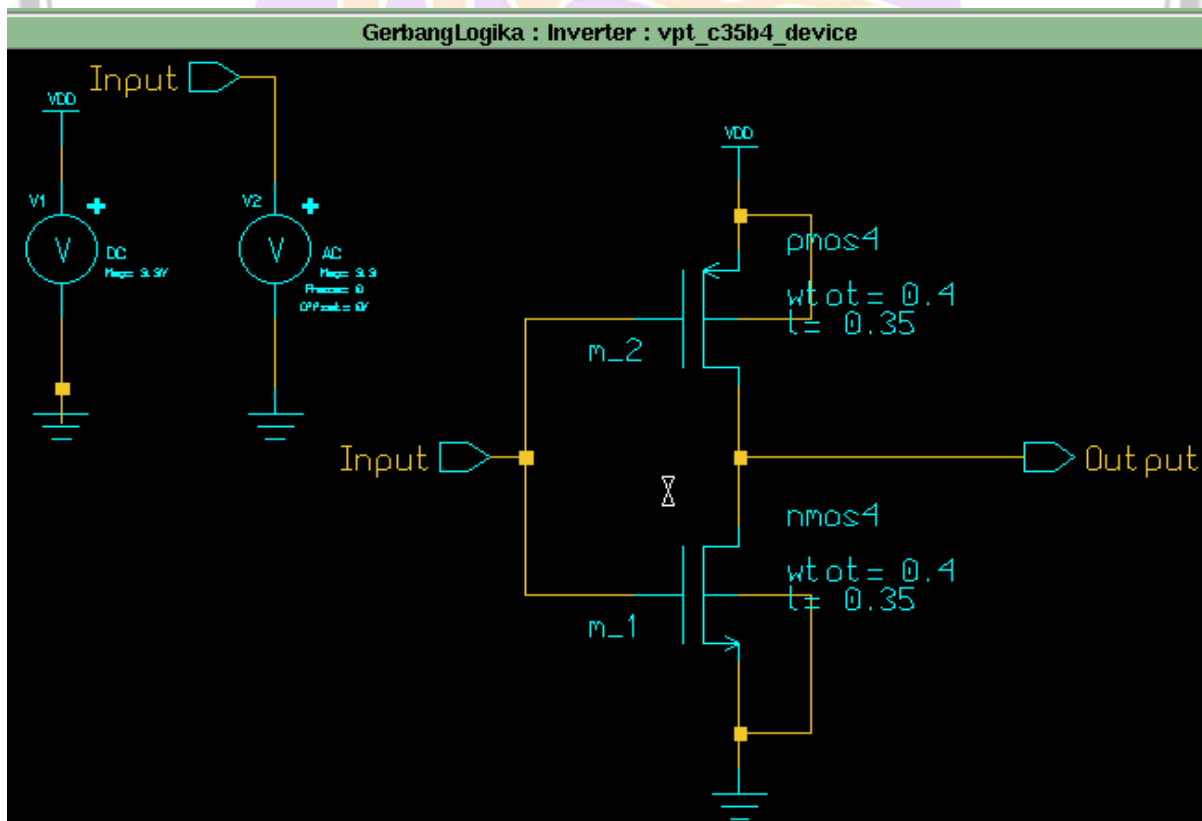
- b. Jalankan Simulasi Menggunakan **RUN ELDO**.

- c. Untuk melihat hasilnya

- klik **DCOP/TRAN>Add Monitor> All Nets – Voltages**
- klik **DCOP/TRAN>Add Monitor> All Pins – Voltages**

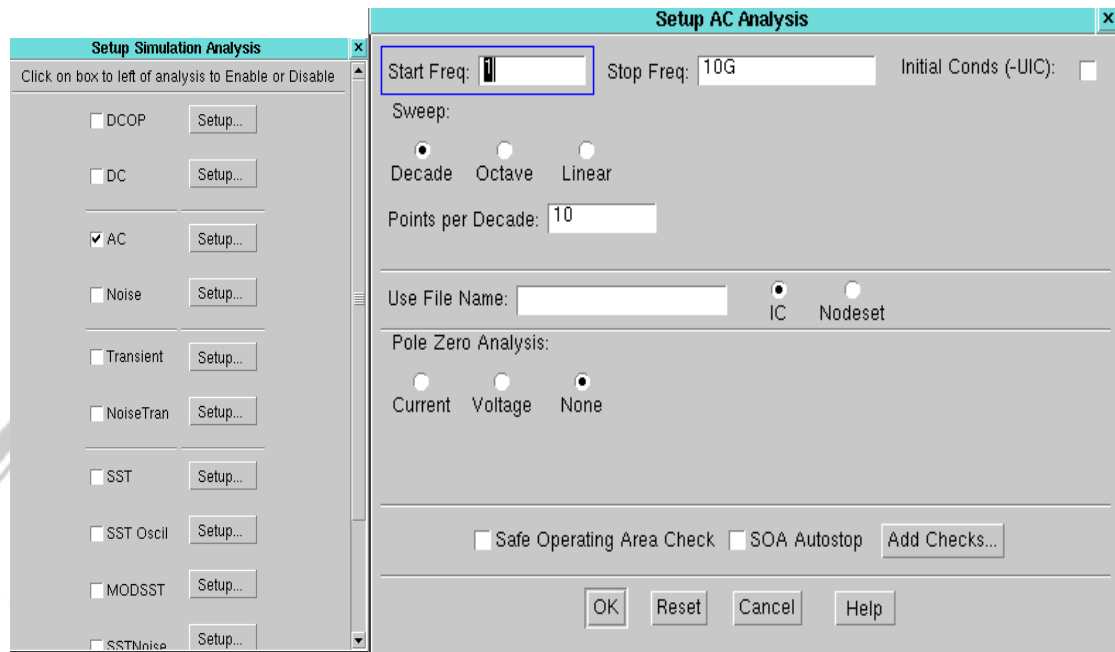


D.3. Simulasi Analisa AC

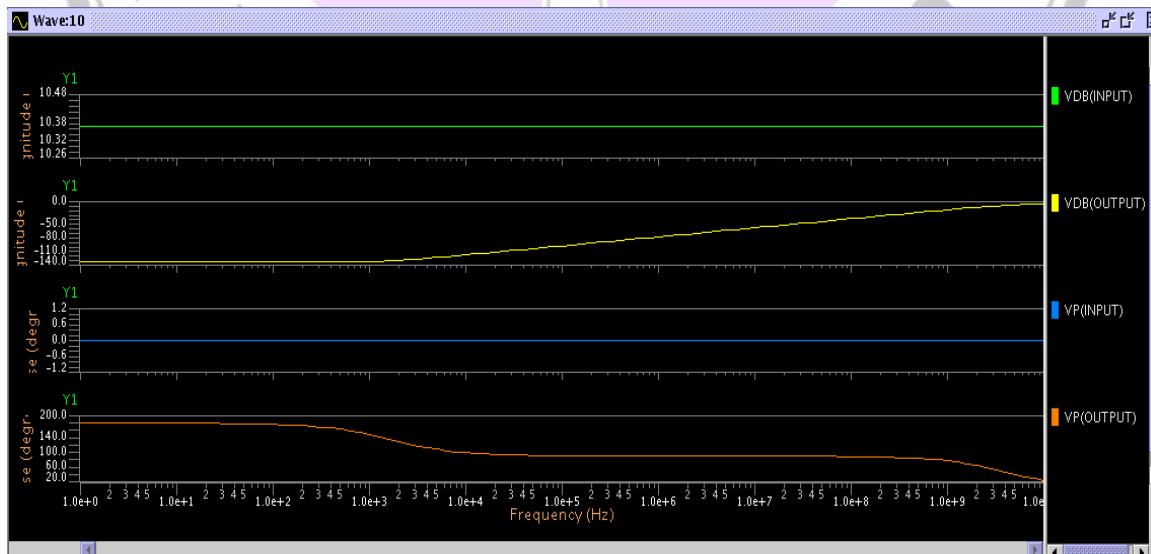


- a. Ubahlah V2 menjadi AC dengan nilai Magnitute = 3.3 V

- b. Jalankan Mode simulasi dan Analisa yang digunakan adalah AC pada menu **Analyses>AC**. Centang **AC**, kemudian Klik Setup. Isikan **Start Freq = 1** dan **Stop Freq = 10G**. Kemudian klik OK

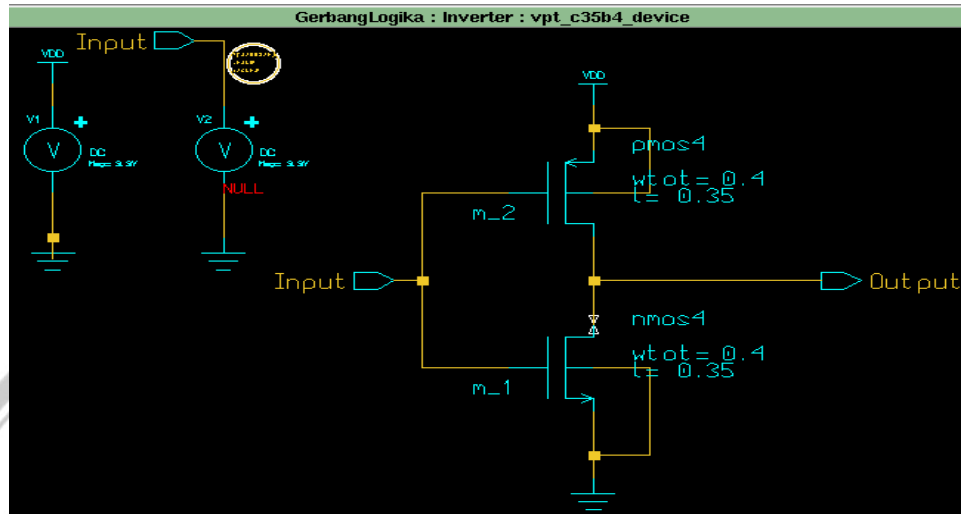


- c. Menyetor Tampilan grafik menggunakan menu **“Wave Outputs > save”** bersamaan dengan mengaktifkan port masukan dan keluaran.
- d. Jalankan Simulasi Menggunakan **RUN ELDO**.
- e. Melihat hasil simulasi menggunakan button **“View Waves”**, akan muncul hasil grafik simulasi.



- f. Melihat **Total Power Dissipation**, menggunakan menu ASCII Files > View Log.
Kemudian carilah **Total Power Dissipation**.

D. 4. Simulasi Analisa Transient

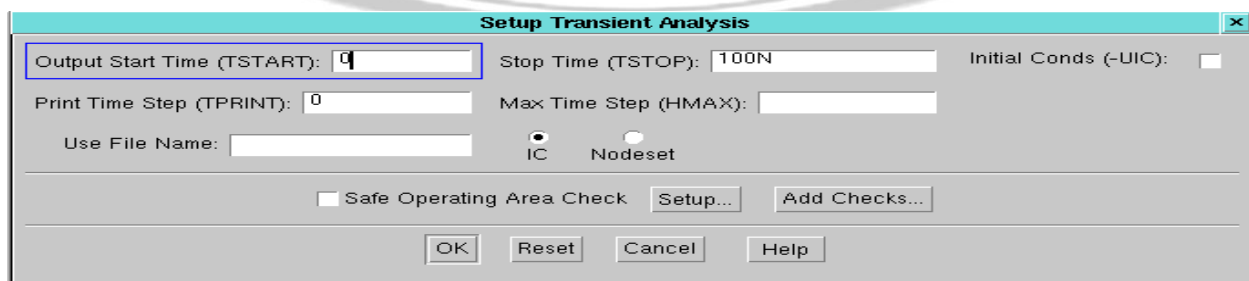
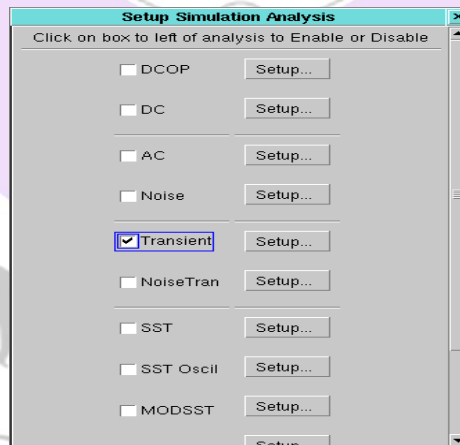


D.4.1. Menggunakan Pattern

- a. Analisa yang digunakan adalah transient pada menu **Analyses>Transient>setup**.

Isikan

- **TSTART = 0**
- **TSTOP = 100N**



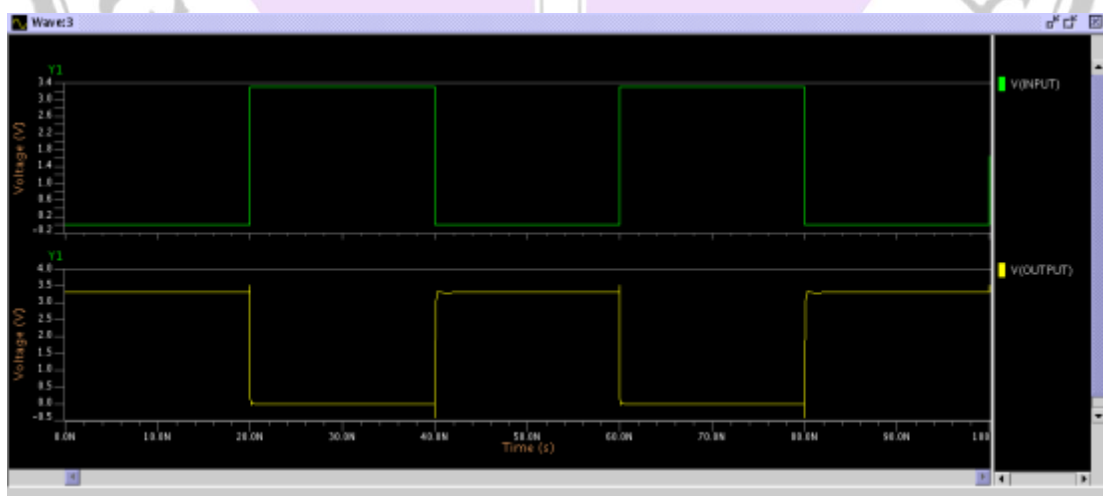
b. Masukkan Virtual menggunakan “**Forces**” dengan cara mengaktifkan setiap port dan diberikan nilai virtual.

- **PATTERN** : - **High Value (1) = 3,3**
 - **Low Value (0) = 0**
 - **Delay (s) = 0**
 - **Rise Time (s) = 20P**
 - **Fall Time (s) = 20P**
 - **Pattern Duration (s) = 20N**
 - **Pattern (Comma or Space delimited) = 0101**



c. Jalankan simulasi menggunakan **RUN Eldo**.

d. Lihatlah hasil simulasi menggunakan “**View Waves**”.

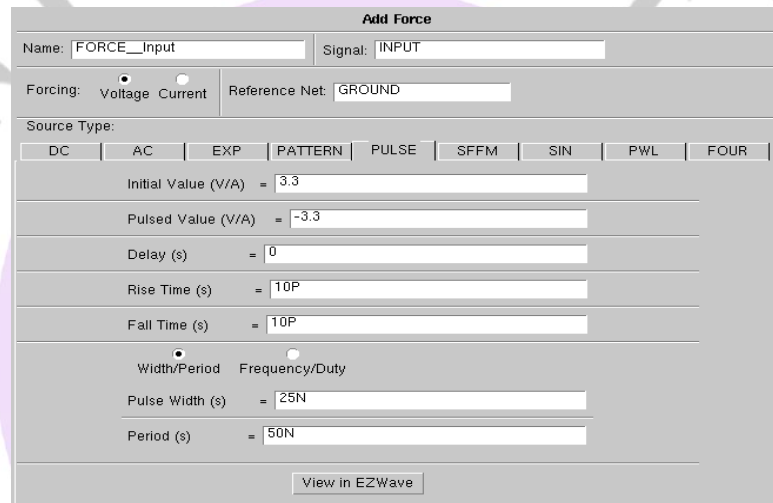


e. Melihat **Total Power Dissipation**, menggunakan menu ASCII Files > View Log. Kemudian carilah **Total Power Dissipation**.

D.4.2. Menggunakan Pulse

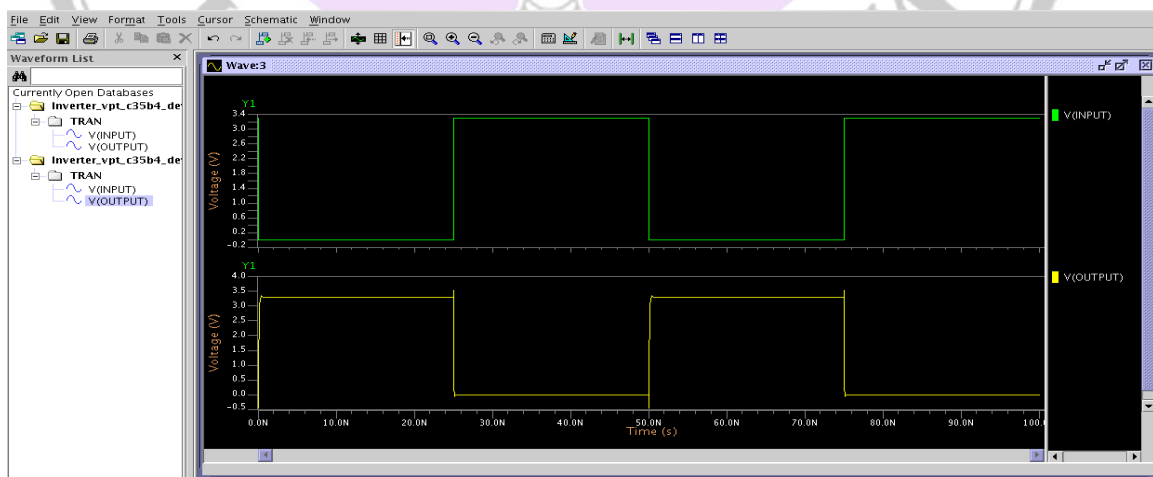
a. Masukkan Virtual menggunakan “Pulse” dengan cara mengaktifkan setiap port dan diberikan nilai virtual.

- **PATTERN** : - Initial Value (V/A) = 3,3
 - Pulsed Value (V/A) = 0
 - Delay (s) = 0
 - Rise Time (s) = 20P
 - Fall Time (s) = 20P
 - Pattern Duration (s) = 20N
 - Pattern (Comma or Space delimited) = 0101



b. Jalankan Simulasi menggunakan **RUN Eldo**.

c. Lihatlah hasil simulasi menggunakan “View Waves”.



Bab 5. Desain Skematik Analog Menggunakan Simbol

5.1. Membuat Desain Skematik Analog Gerbang Logika Menggunakan Simbol

A. Tujuan : Membuat rangkaian dasar gerbang logika (Gerbang dasar) menggunakan simbol

B. Kemampuan : - Mampu membuat dan melakukan simulasi menggunakan Simbol..

C. Teori :

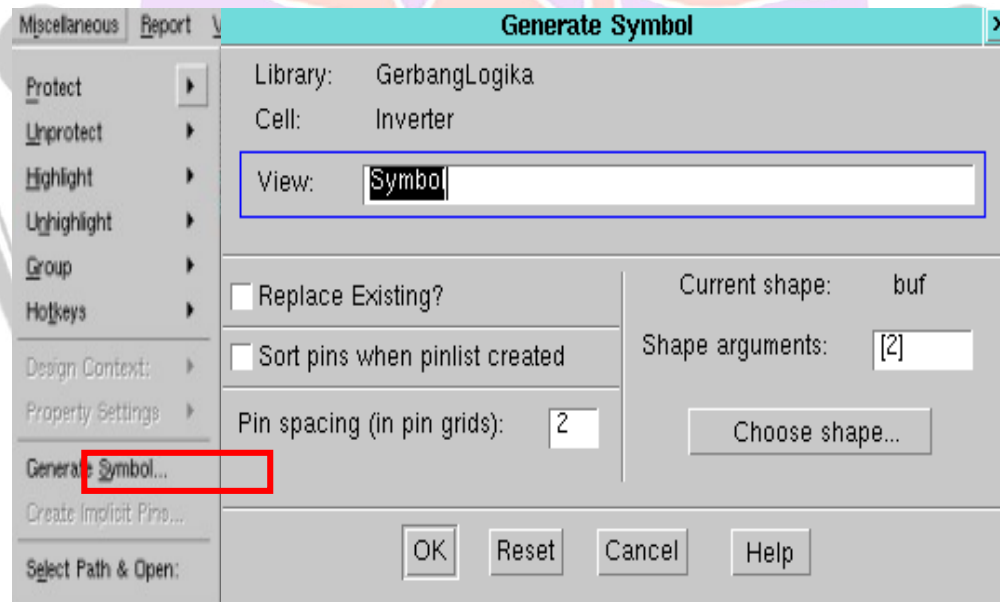
Prinsip penggunaan simbol sama dengan membuat rangkaian langsung dengan transistor. Fungsi dari penggunaan simbol adalah menyederhanakan suatu rangkaian menjadi satu blok simbol. Pada desain rangkaian yang kompleks, simbol sangatlah dianjurkan untuk digunakan.

D. Kegiatan :

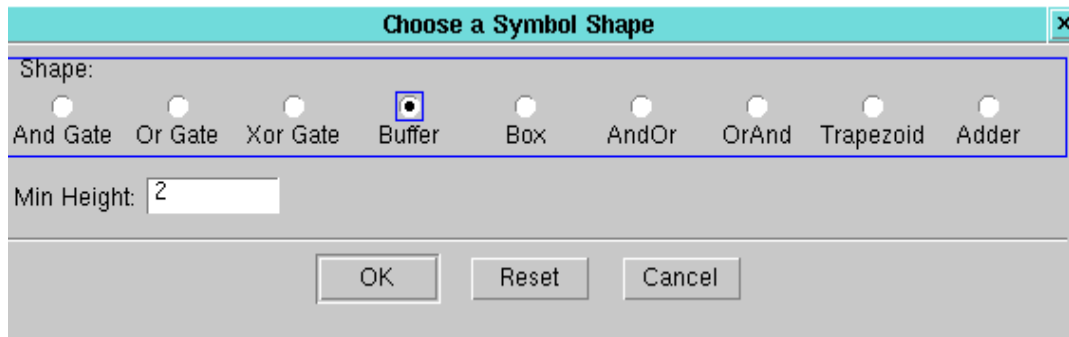
D.1. Pembuatan simbol

a. Rangkaian Inverter yang telah dibuat dapat langsung diubah menjadi simbol dengan menggunakan Menu **Miscellaneous>Generate Symbol**.

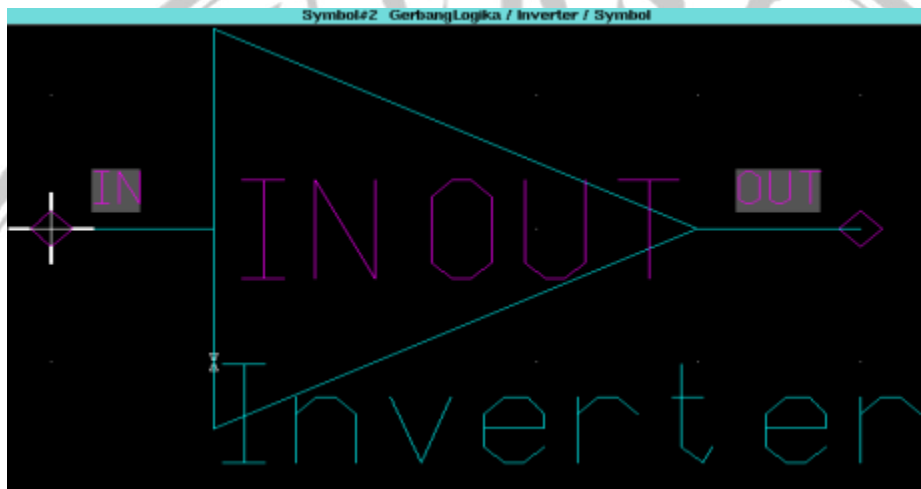
- Pada layar **Generate Symbol, View** diisikan dengan **Symbol**.



- Klik Button **Choose Shape**, pilih **Shape = Buffer** dan **Min Height = 2**, Klik **OK**.

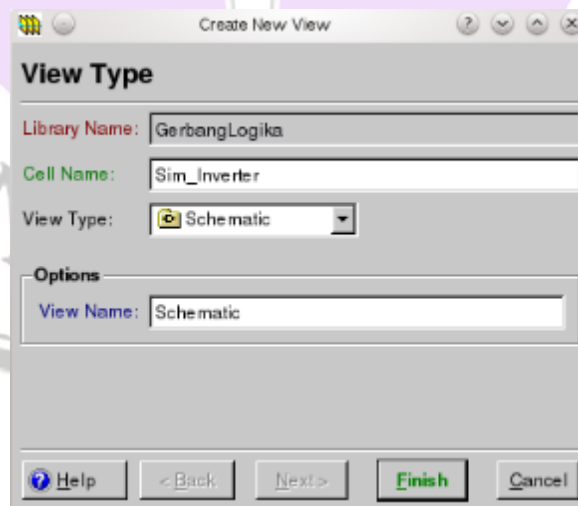


- b. Layar simbol akan muncul dan klik **Save**, sampai disini simbol telah berhasil dibuat

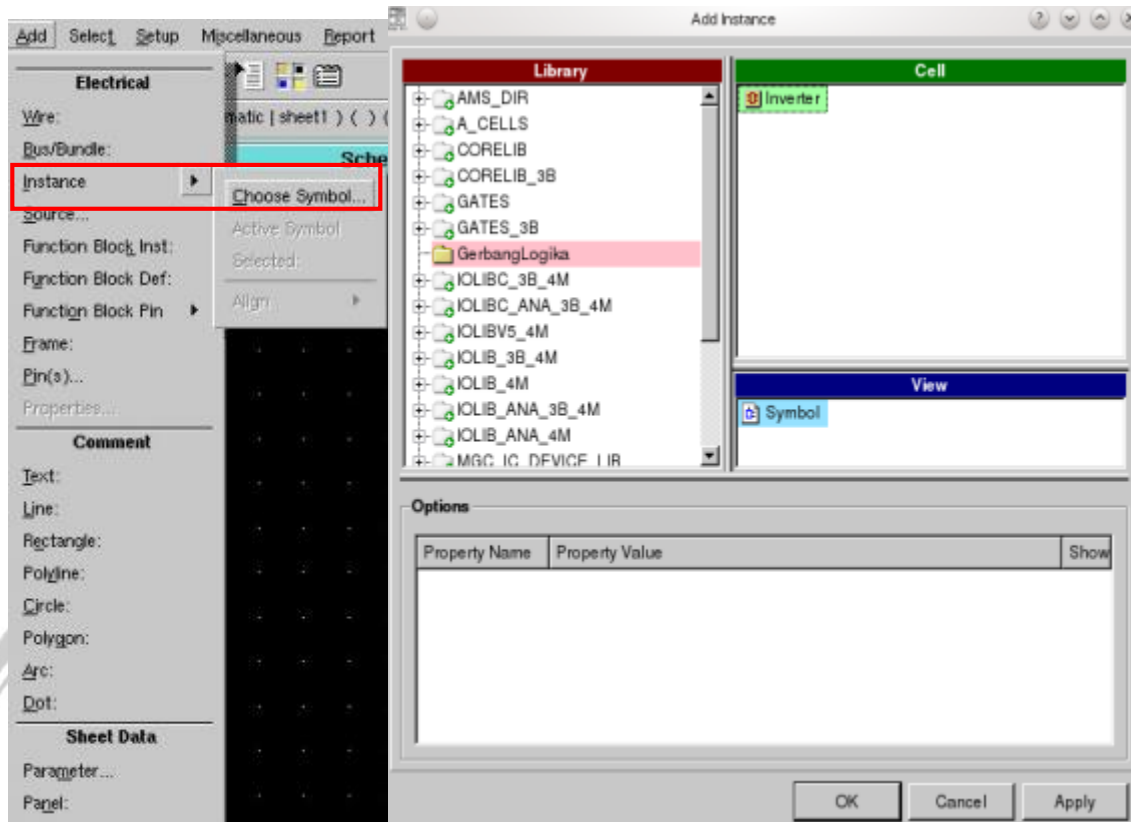


D.2. Simulasi Menggunakan Simbol.

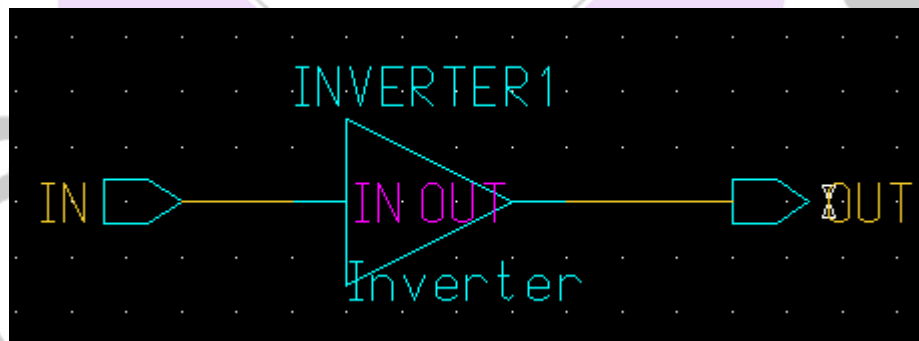
- a. Buatlah cell baru dengan **Cell Name : Sim_Inverter** dan **View Type : Schematic**



- b. Letakkan simbol yang telah dibuat menggunakan menu **Add > Instance > Choose Symbol**. Pilihlah cell **Inverter**.



c. Langkah selanjutnya simulasi, proses simulasi sama dengan bab pembuatan skematik inverter.



Bab 6. Latihan Skematik

6.1. Latihan membuat Skematik NAND, OR, NOR, XOR dan XNOR

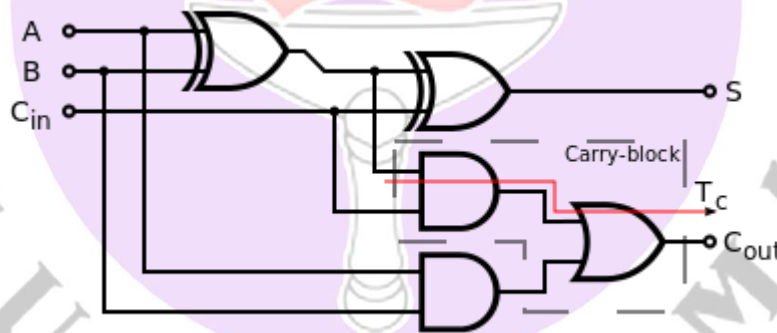
- A. Tujuan : Membuat rangkaian lanjutan **Skematik** NAND, OR, NOR, XOR dan XNOR
- B. Kemampuan : Mampu membuat **Skematik** NAND, OR, NOR, XOR dan XNOR
- C. Teori :
Gerbang lanjutan dapat dibuat menggunakan menggunakan rangkaian inverter yang telah ada.
- D. Kegiatan

D.1. Latihan Skematik Gerbang logika

Buatlah rangkaian NAND, OR, NOR, XOR dan XNOR kemudian simulasikan!

6.2. Latihan membuat Skematik Adder

- A. Tujuan : Membuat rangkaian lanjutan **Skematik Adder**
- B. Kemampuan : Mampu membuat **Skematik Adder**
- C. Teori :
Adder merupakan komponen elektronika digital yang dipakai untuk menjumlahkan dua buah angka dalam sistem bilangan biner



- D. Kegiatan

D.1. Latihan Skematik Adder

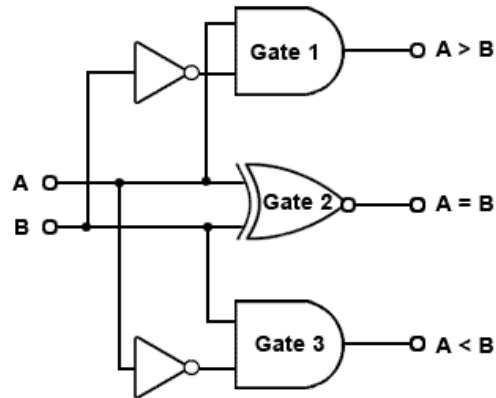
Buatlah rangkaian Adder menggunakan simbol-simbol gerbang logika yang telah dibuat dan simulasikan!

6.3. Latihan membuat Skematik Komparator

- A. Tujuan : Membuat rangkaian **Skematik Komparator**
- B. Kemampuan : Mampu membuat **Skematik Komparator**

C. Teori :

Komparator merupakan rangkaian elektronik yang berfungsi membandingkan besar tegangan atau nilai masukan.



D. Kegiatan

D.1. Latihan Skematik Komparator

Buatlah rangkaian Komparator menggunakan simbol-simbol gerbang logika yang telah dibuat dan simulasikan!

6.4. Latihan membuat Skematik Encoder dan Decoder

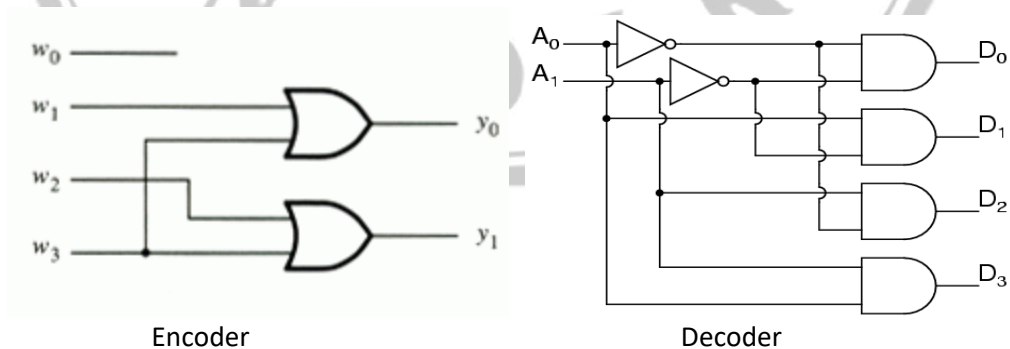
A. Tujuan : Membuat **Skematik Encoder dan Decoder**

B. Kemampuan : Mampu membuat **Skematik Encoder dan Decoder**

C. Teori :

Rangkaian encoder merupakan rangkaian elektronika yang berfungsi mengkodekan data masukan menjadi data dengan format tertentu atau merupakan rangkaian kombinasi gerbang logika yang memiliki masukan lebih banyak dan memiliki keluaran yang sedikit.

Rangkaian decoder merupakan kebalikan dari rangkaian encoder



D. Kegiatan

D.1. Latihan Skematik Encoder dan Decoder

Buatlah rangkaian encoder dan decoder menggunakan simbol-simbol gerbang logika yang telah dibuat dan simulasikan!

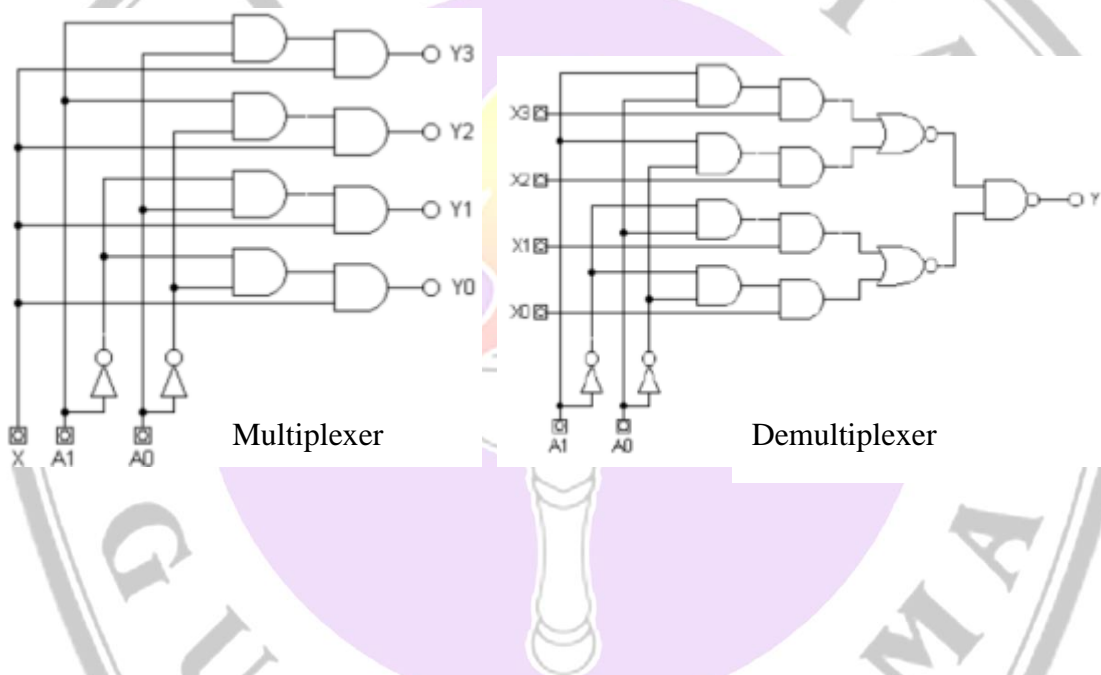
6.5. Latihan membuat Skematik Multiplexer dan Demultiplexer

A. **Tujuan** : Membuat **Skematik Multiplexer dan Demultiplexer**

B. **Kemampuan** : Mampu membuat **Skematik Multiplexer dan Demultiplexer**

C. **Teori** :

Rangkaian multiplexer merupakan rangkaian logika yang menerima beberapa input data dan menyeleksi salah satu dari input dan menghasilkan satu keluaran. Rangkaian demultiplexer merupakan kebalikand dari rangkain multiplexer.



D. Kegiatan

D.1. Latihan Skematik Multiplexer

Buatlah Skematik Multiplexer dan Demultiplexer menggunakan simbol-simbol gerbang logika yang telah dibuat dan simulasikan!

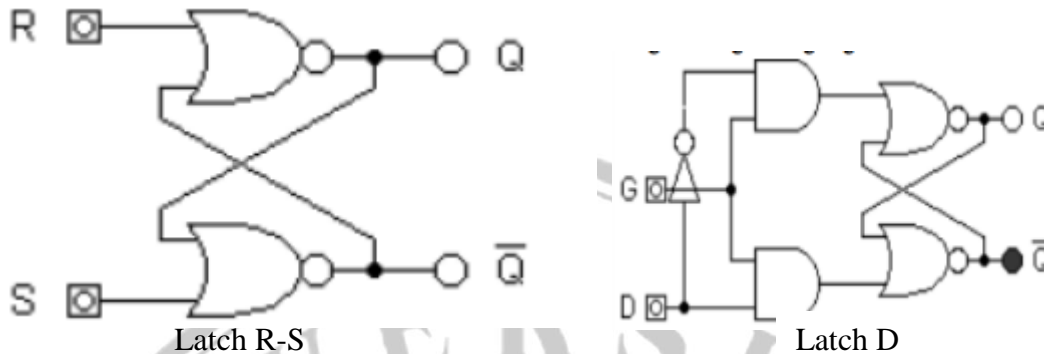
6.6. Latihan membuat Skematik Latch

A. **Tujuan** : Membuat **Skematik Latch**

B. **Kemampuan** : Mampu membuat **Skematik Latch**

C. **Teori** :

Rangkaian latch merupakan rangkaian elektronik yang mempunyai kondisi stabil dan dapat menyimpan satu bit informasi.



D. Kegiatan

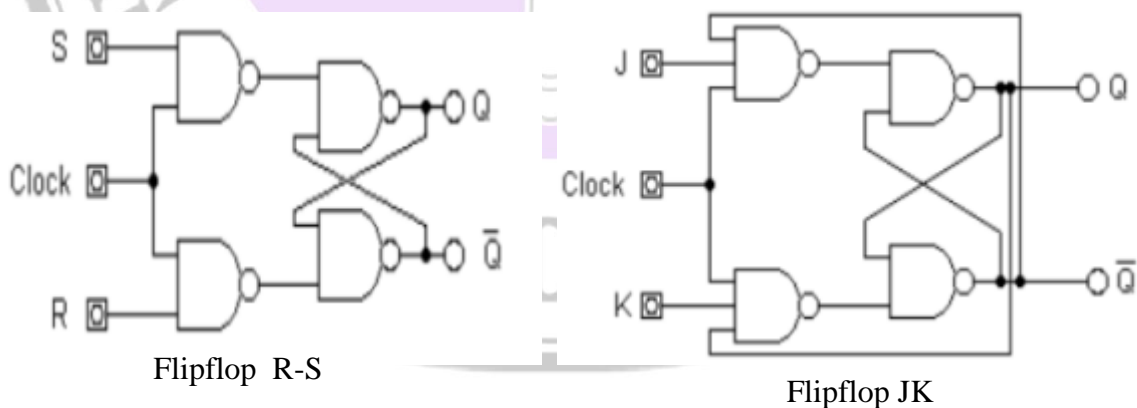
D.1. Latihan Skematik Latch

Buatlah Skematik Latch menggunakan simbol-simbol gerbang logika yang telah dibuat dan simulasikan!

6.7. Latihan membuat Skematik Flip-Flop

- A. **Tujuan** : Membuat **Skematik Flip-Flop**
- B. **Kemampuan** : Mampu membuat **Skematik Flip-Flop**
- C. **Teori** :

Rangkaian Flip-Flop merupakan rangkaian elektronik yang mempunyai kondisi stabil dan dapat menyimpan satu bit informasi.



D. Kegiatan

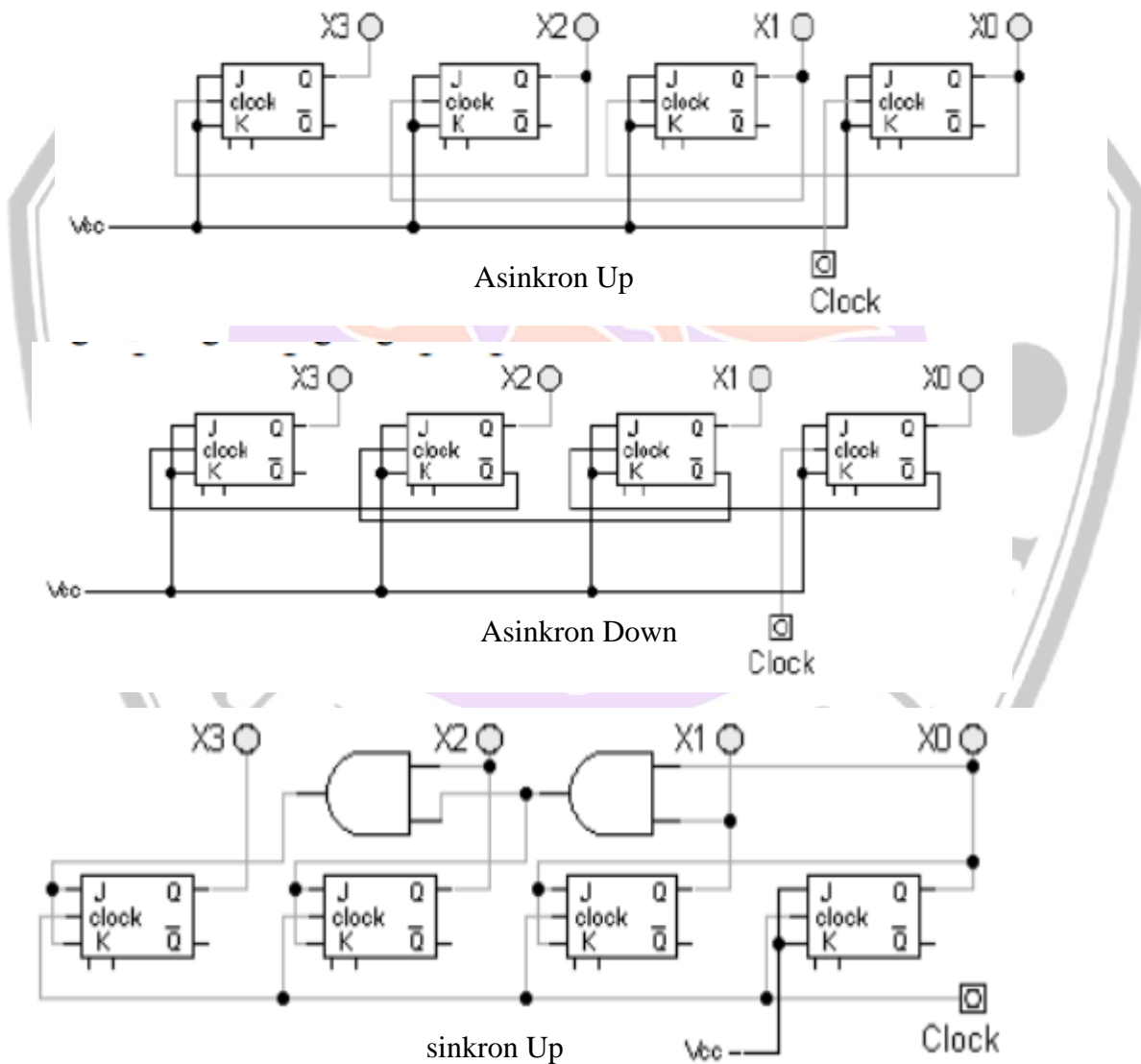
D.1. Latihan Skematik Flip-Flop

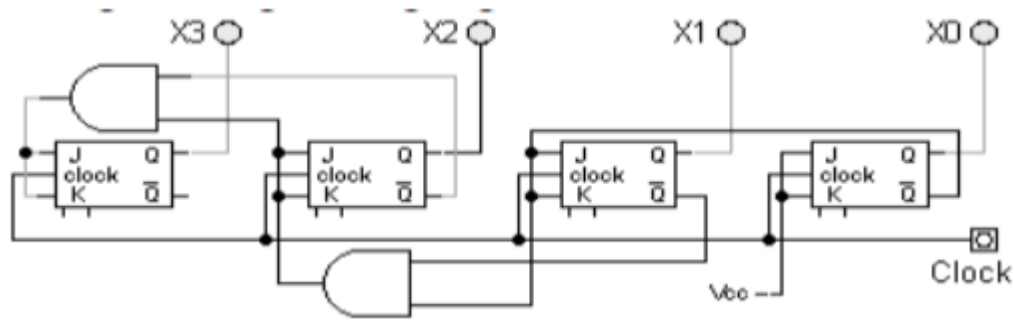
Buatlah Skematik Flip-Flop menggunakan simbol-simbol gerbang logika yang telah dibuat dan simulasikan!

6.8. Latihan membuat Skematik Counter

- A. Tujuan : Membuat **Skematik Counter**
- B. Kemampuan : Mampu membuat **Skematik Counter**
- C. Teori :

Rangkaian Counter merupakan rangkaian elektronik yang berfungsi melakukan perhitungan angka secara berurutan baik berupa perhitungan maju(Up) dan perhitungan mundur (Down)





sinkron Down

D. Kegiatan

D.1. Latihan Skematik Counter

Buatlah Skematik Counter menggunakan simbol-simbol gerbang logika yang telah dibuat dan simulasikan!

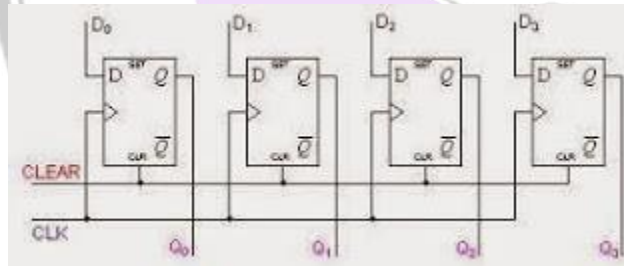
6.9. Latihan membuat Skematik Register

A. **Tujuan** : Membuat **Skematik Register**

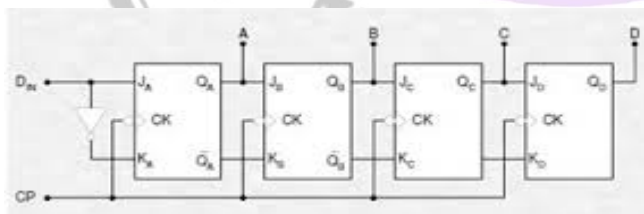
B. **Kemampuan** : Mampu membuat **Skematik Register**

C. **Teori** :

Rangkaian register merupakan rangkaian elektronik yang dibentuk dari beberapa flipflop yang mampu menyimpan dan memproses nilai dalam bentuk biner..



Storage register



Shift register

D. Kegiatan

D.1. Latihan Skematik Register

Buatlah Skematik Flip-Flop menggunakan simbol-simbol gerbang logika yang telah dibuat dan simulasikan!



DAFTAR PUSTAKA

- [1] Kurniawan, R. and Prasetyo, E., 2016. Konsep dan Metodologi Desain Analog CHIP-Berbasis Teknologi disertai Penggunaan Tool.
- [2] Wibowo, E.P., Disain Skematik, Layout dan Simulasi dengan Menggunakan Perangkat Lunak Mentor Graphics T M.
- [3] Harahap, R.K., Prasetyo, E., Heruseto, B. and Afandi, H., 2015, December. Design analog layout using schematic-driven eda tools. In *2015 International Conference on Electrical, Electronics and Mechatronics*. Atlantis Press.
- [4] Bhatti, M.K., Minhas, A.A., Najam-ul-Islam, M., Bhatti, M.A., Haque, Z.U. and Khan, S.A., 2012, August. Curriculum design using mentor graphics higher education program (hep) for asic designing from synthesizable hdl to gdsii. In *Proceedings of IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE) 2012* (pp. W1D-1). IEEE.
- [5] Baker, R.J., 2010. CMOS: Circuit Design, Layout, and Simulation.
- [6] Nelson, V.P., 2015. Computer-Aided Design of ASICs Concept to Silicon.
- [7] Wang, L.T., Chang, Y.W. and Cheng, K.T.T. eds., 2009. *Electronic design automation: synthesis, verification, and test*. Morgan Kaufmann.
- [8] Graphics, M., 2012. Modelsim-advanced simulation and debugging.
- [9] Station, I.C. and Station, D.S.P., 2006, Mentor Graphics. *San Jose, California*.
- [10] URL : <http://cmosedu.com/cm0s1/mentor/mentor.htm>., diakses 20 juni 2017