Terminal In you folder.

>csh >csh

>source /home/install/cshrc

>virtuoso &

Step 1:- CREATE A LIBRARY.

→ In virtuoso window

↳ tools → library Manager

→ New window will pop up there

↳ file → New → library.

→ In bottom taskbar there will be technology window blinking click that & Select

↳ Attach to an existing technology library.

↳ Select:- gpdk180 → Apply → ok.

Step 2:- CREATE A CELL VIEW

→ In ~~file~~ library manager

↳ File → New → cell view

→ Give cell Name & ok

↳ ok

Note:- In library Manager you should first click on your library that you have created then only the New cell view option will come.

STEP 3. ADD AN INSTANCE

→ create → Instance or click ('I') in keyboard.

→ Add Instance window will pop up

① library:- gpdk180

cell Name:- NMOS

view:- symbol

Total width:- 1.74

length:- 180n

→ click hide & click anywhere on Schematic window

→ Similarly create PMOS with give values
  ↳ click 'I'
  ↳ ~~cell~~ library:- gpdk180
  ↳ cellname:- PMOS
  ↳ view:- Symbol
  total width:- 1.875u
  length @:- 180n
→ Hide & click on the Schematic where you want to put it.
→ Create → Pin or (little pin option on top.)
  ↳ Create pin for Input & output type as required
→ After ~~creating~~ Adding all the pins use wire to make respective connections.
→ check & Save [If no warning] then continue.

Step 4:- Symbol Creation

→ In Schematic window
  ↳ create → cellview → From cellview [verify dummy]
  ↳ ok
→ Symbol Generation options will pop up
  ↳ put where you want your pins to be placed
  [Left:- vss, right:- VDD ...]
  ↳ ok

Step 5:- Test Circuit for Simulation

→ In virtuoso
  ↳ file → new → cell view → cell:- CMOS_tb

→ Instantiate your Symbol
  ↳ click 'I'
  ↳ Select your library
  ↳ & previous cell.

→ Add Instance
  ↳ click 'I'
  ↳ In 'analoglib' library
  ↳ Add Vc, ~~Vpulse~~ cap, gnd
  → Add A, B pins Input

→ make all the connection using wire

→ check & Save.

Step 6:- FUNCTIONAL SIMULATION WITH SPECTRE:

→ Launch
  ↳ ADE L

→ In ADE L window
  ↳ Setup
    ↳ Stimuli

→ Add pattern & value for A & B
  ↳ Stimulus type ☑
  ↳ Select A
  ↳ Enabled ☑ function:- bit
  ↳ one value :- 1.8
    zero value :- 0
    Rise time :- In
    fall time :- In
    Pattern :- 11001001 B:-pattern:- 00011011

→ Similarly for B with different pattern.

Step 7: SELECTING THE ANALYSIS

→ In ADE L window
  ↳ Analysis
    ↳ choose
      → tran
      → skip time loon
      → moderate

Skp 8:- SELECTING THE SIGNALS TO BE PLOTTED

→ In ADE L window
  ↳ output
    ↳ Setup
→ New window will pop up
  ↳ choose Input & output wires
    ↳ OK
→ Saving ADE L State Session: Save State (cellview)
  ↳ ok

Skp 9:- RUNNING THE SIMULATION

→ In ADE L window
  ↳ Simulation
    ↳ Netlist & Run
→ Grap
  ↳ Slipt all lines

Step 10:- CALCULATE THE DELAY

→ Select calculator
→ In Signal 1: Select Any one Input (A or B)
→ In Signal 2: Select O/P.
→ Threshold value 1 = value 2 = 0.75
→ once: take Edge type 1:- rising
  2: falling

Second one
: falling
:- rising.

→ get two value Add both & ÷ by 2.

Step 11: LAYOUT FOR CMOS

→ go to main file where we get transistor.

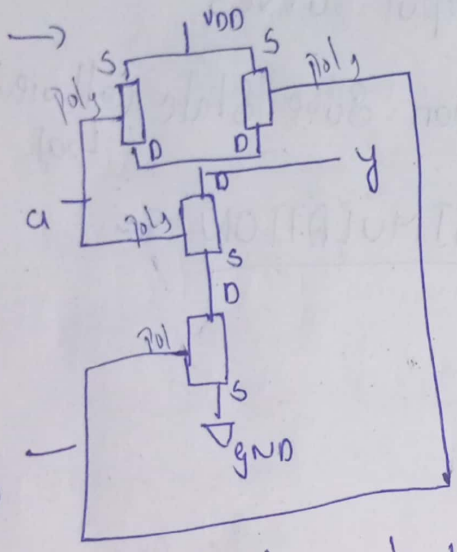
↳ Launch
↳ layout XL → create new → Automatic → pole
↳ pole

→ layout window will pop up
↳ connectivity
↳ generate
↳ All from Source (minim Sep:- 1)

→ fit all component in vechyle.

→ do all metal to metal connection
click P for metal

→



→ for connect b/w pins like A & B
we should like poly

↳ creat
↳ via
↳ via_difination
↳ M1_POLY1

→ save
→ select big boxes except middle
↳ parameter → body type → Integrated.
→ for middle " " → detached.

→ there will another one port at middle
→ take that connect it to vss
↳ bottom one

→ save

DRC :- Step 12

→ Assure
↳ technology
↳ ...
↳ ..
↳ ..
↳ . unhill

→ home
↳ jnjhill
↳ foundary
↳ analog
↳ 180nm
↳ Right side (assure.rcf)
↳ ok

Assura
↳ RUN DRC
↳ technology: gpdk 180
↳ ok
↳ ok

→ no errors.

LVS :- Step 13

Assura
↳ run LVS.
↳ technoly :- gdpk 180
↳ ok
↳ ok
↳ ok

no errors

→ task bar extract window open it.
↳ it should say match.

QRC

Assura
↳ Run Quantus QRC
→ extraction
↳ extraction type : RC
reference. Node :- VSS or GND
↳ ok

→ successfull.

Virtuoso
↳ tools
↳ library manager.
↳ CMOS 1 → library.
↳ CMOS → main file
↳ av_extracted
↳ double click

→ open file → tub will be there
↳ choose ~~cmtb~~ CMOS_tb.
view :- schematic
ok

→ again double click av. extracted.

→ Shift - f for va..y.

file
↳ New
↳ cell view
↳ type → config
↳ ok

cell :- main file

---

view :- schematic
cell :- CMOS_hb.
↳ use template
Name = specter
↳ ok
↳ ok

tree view
↳
☑ 10 (CMOS - ..)
↳ Right click
↳ set Instance value
↳ av_extracted.
→ Save

Launch
↳ ADE-L
↳ set model library
↳ first only tick & SS

↳ Analysis
↳ choose
↳ stp 1000
↳ ☑ modulation → ok.

outputs
↳ set
↳ from dajiyn
→ Select I/P & O/P.

→ netlist & run.