LABORATÓRIO 4 – PROJETO DE PORTAS LÓGICAS

Prof. Fernando Gehm Moraes – Revisão: 12/maio/2025

DOWNLOAD DOS ARQUIVOS E ÁRVORE DE DIRETÓRIOS

Fazer o download do lab4 e executar os seguintes comandos:

```
wget https://fgmoraes.github.io//microel/lab4/lab4.tar
tar -xvf lab4.tar
cd lab4
```

O conteúdo do diretório layout é dado abaixo (comando *tree –a*):

```
lab4
|-- .cdsinit                    → arquivo do design kit
|-- .cshrc_cmos065              → arquivo do design kit
|-- .ucdprod                    → arquivo do design kit
|-- cds.lib                     → indica as bibliotecas de trabalho
|-- characterization
|   `-- liberate.tcl            → script para caracterizar eletricamente a célula projetada
|-- gera_abstract.txt           → script para gerar o modelo físico da célula projetada
|-- sim
|   |-- anel.sp                 → spice para simular um anel de inversores
|   |-- inv.sp                  → spice para um  inversor
|   `-- st65.scs                → arquivos da tecnologia com modelos elétricos dos transistores
`-- synthesis
    |-- cmd_genus               → script para síntese lógica
    |-- cmd_innovus             → script para síntese física
    |-- load.tcl                → parâmetros para o script da síntese lógica
    `-- src
        `-- anel.vhd            → VHDL que instancia a célula projetada
```

Conteúdo do relatório a ser entregue

1. **Esquemático** (10%) - *print screen* da tela e *print screen* do console com mensagem se houve ou não erro no esquemático.

2. **Layout do inversor** (30%) [print screen da tela]

3. **Relatório do DRC** (5%) [tela que indica se houve ou não erros]

4. **Relatório do LVS** (5%) [tela que indica se houve ou não erros]

5. **Extração elétrica** (5%). Apresentar os arquivos:
   - inv.pex.spi
   - inv.pex.spi.inv.pxi
   - inv.src.net

6. **Simulação elétrica** (10%). Curvas da simulação e o atraso. O atraso é dado copiando-se os dado do arquivo *inv.measure*. Simulação do anel do inversor para com **21 inversores**. Curvas com a simulação e a indicação da frequência de operação (obtido do arquivo *anel.measure*)

7. **Layout da view abstract** (5%). Print screen da tela da view abstract e o arquivo texto LEF.

8. **Geração do arquivo de caracterização elétrica** (5%) – adicionar a homepage gerada pela ferramenta (*datasheet/index.html*) no relatório

9. **Síntese lógica** (10%) – apresentar o esquemático gerado pela ferramenta de síntese. **Modificar** o anel para 13 inversores (12 inversores mais uma NAND). No tutorial temos 11 inversores (10 inversores mais uma NAND).

10. **Síntese física** (15%) – apresentar o layout para 13 inversores, e o relatório de DRC.

CONFIGURAÇÃO DO AMBIENTE

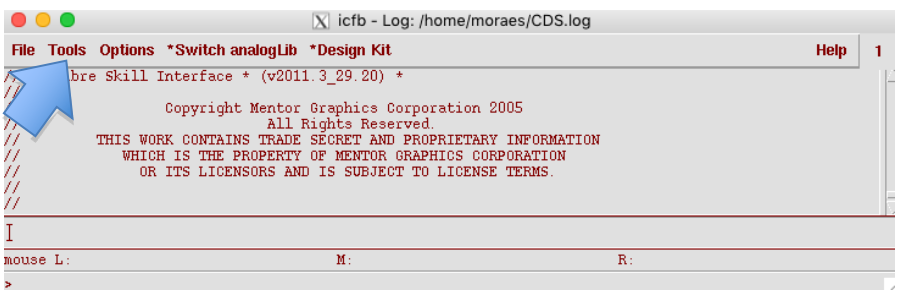
Executar os seguintes comandos para abrir a ambiente de projeto de células da CADENCE, **no diretório lab4**:

```
module load ic/5.1.41
csh
source .cshrc_cmos065
setenv CDS_AUTO_64BIT
icfb &
```

➔ opcional: para evitar incompatibilidade com 32 bits

➔ **se não abrir o icfb: icfb -nosplash &** (só se houver erro ao invocar o icfb)

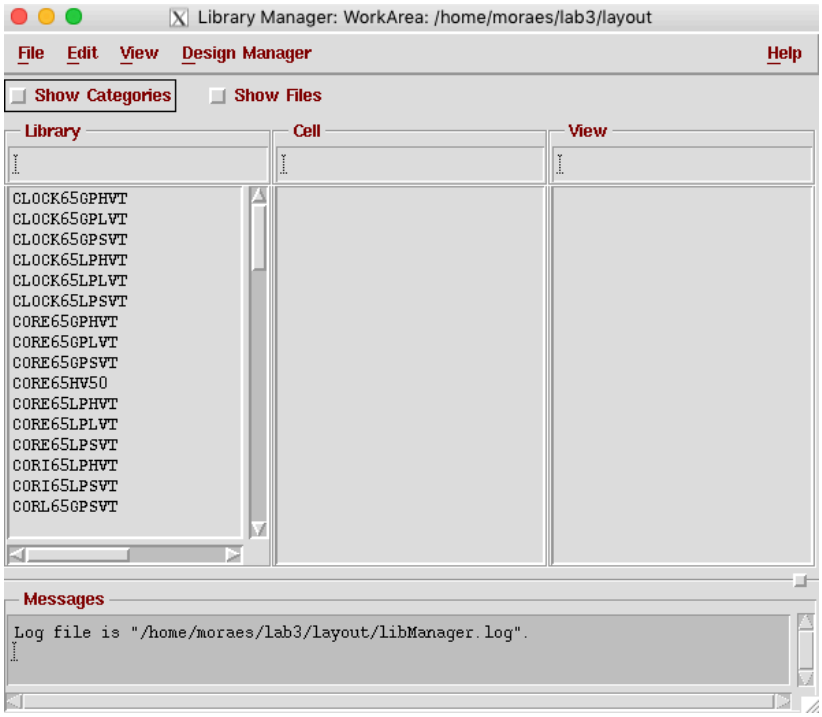
Abrirá a seguinte janela (demora um pouco para carregar todas as bibliotecas):



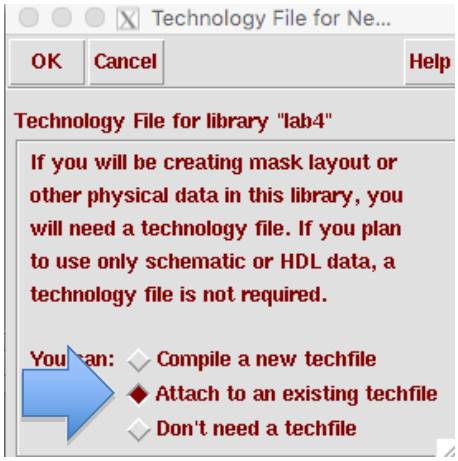
1 - CRIANDO O ESQUEMÁTICO

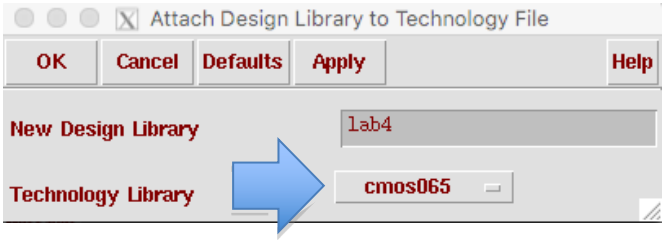
Utilizar a ferramenta *icfb* para os seguintes passos:

a) Primeiro abrir o Library Manager: **Tools→Library Manager...**

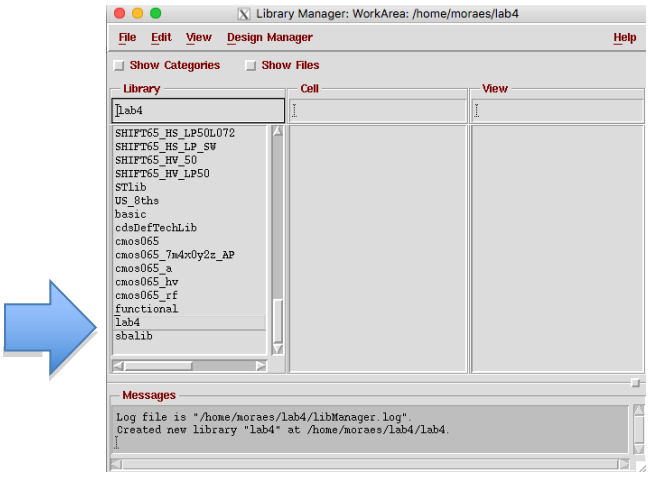


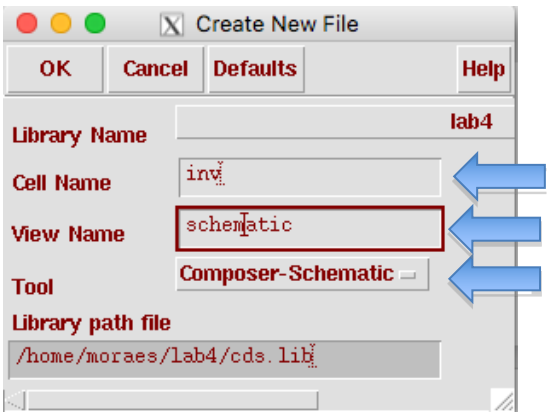
b) Criar a biblioteca onde os *layouts* serão gravados: **File→New Library**. Ao clicarmos em OK abre a janela da biblioteca de referência. Marcar a opção "*Attach to an existing techfile*" para que a biblioteca faça uso dos modelos disponíveis no design kit. Clicar em ok. Selecionar a tecnologia "cmos065".



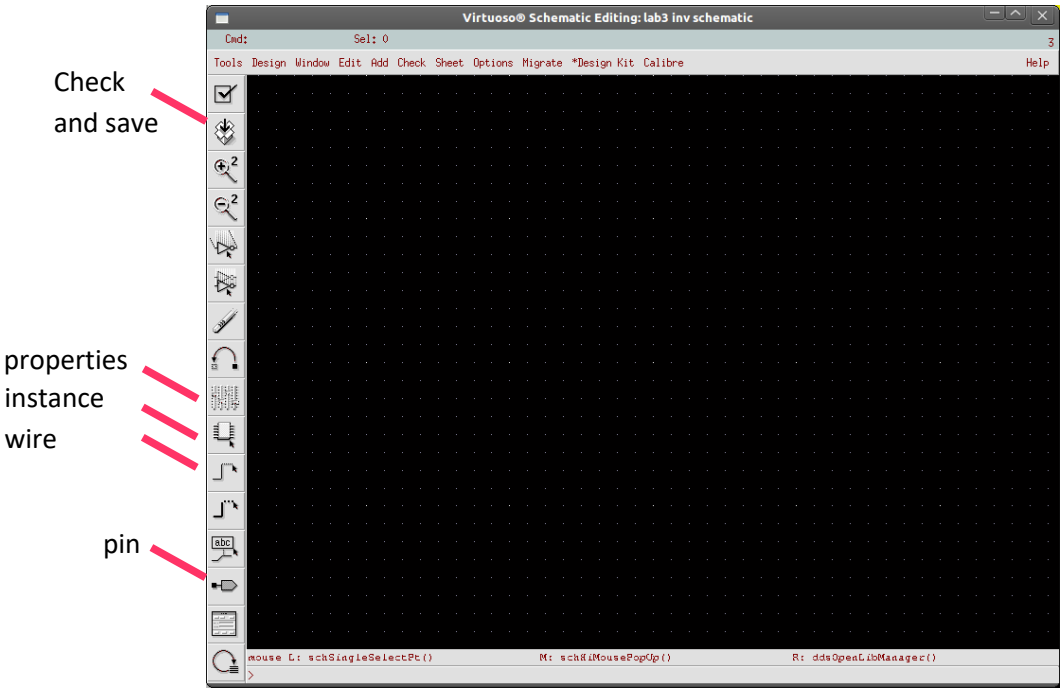


c) Agora, criar o esquemático do inversor: No Library Manager, selecionar a biblioteca criada e clicar em **File → New → Cell View**... Digitar o nome da célula que será projetada, selecionar a ferramenta "Composer-Schematic", dar um nome a visão ("schematic") e clicar em ok.

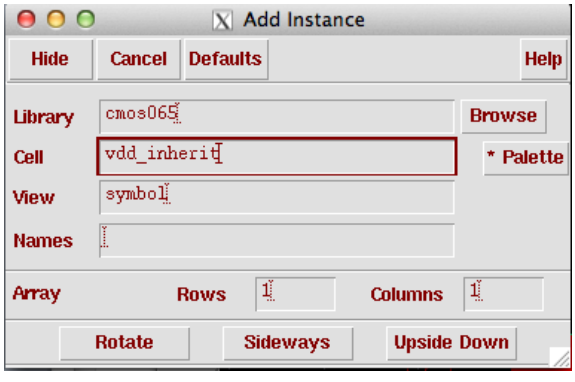




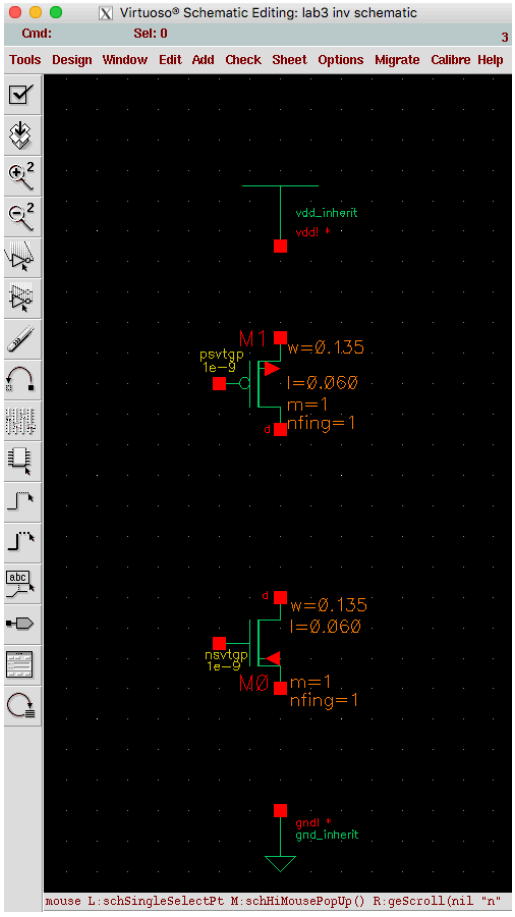
O resultado é a interface gráfica da ferramenta de edição de esquemáticos.



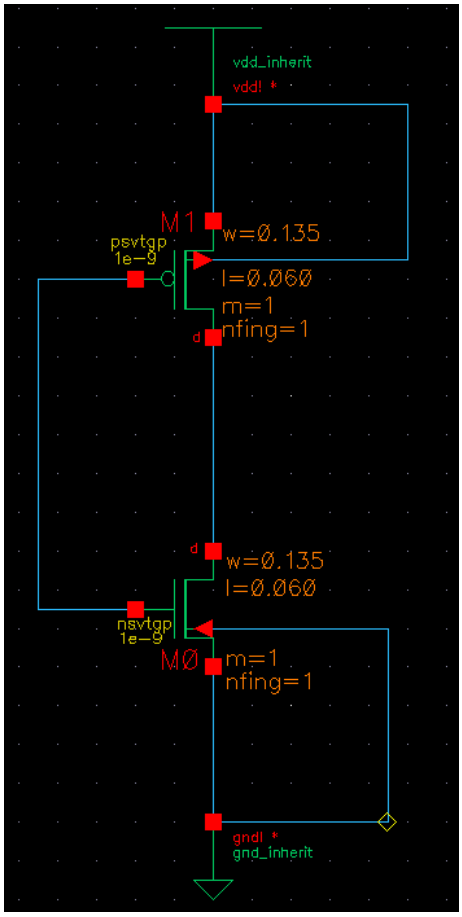
d) Instanciar a fonte de alimentação da célula e massa: clicar em *instance* (ou apertar a tecla "i"), clicar em browse, escolher a biblioteca cmos065, escolher a célula "vdd_inherit", view "symbol" e instanciá-la no esquemático, clicando em qualquer lugar da janela do *schematic editor* (para sair teclar ESC). Repetir o processo escolhendo a célula "gnd_inherit". **DICA: PODE-SE ESCREVER O NOME DA CÉLULA NO CAMPO "cell" do Library Browser**".



e) Agora deve-se instanciar dois transistores, um PMOS e um NMOS, para gerar o esquemático que implementa a lógica do inversor. Para tanto deve-se: clicar em *instance*, clicar em *browse*, selecionar a biblioteca cmos065 e instanciar o símbolo da célula "**nsvtgp**". Repetir o processo, instanciando a célula "**psvtgp**". Notar que as dimensões dos transistores são definidas para mínimas por default (L=0.060 µm e W=0.135 µm). O resultado deve se um esquemático similar à figura (a) abaixo.



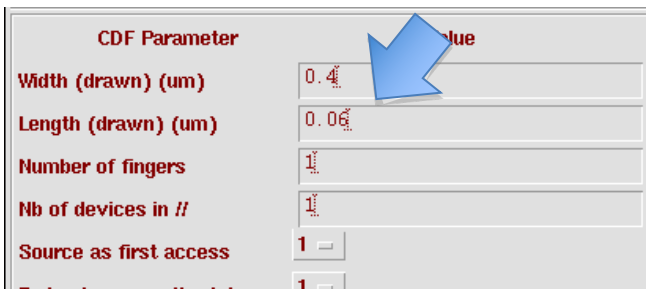
(a) Esquemático com as instâncias



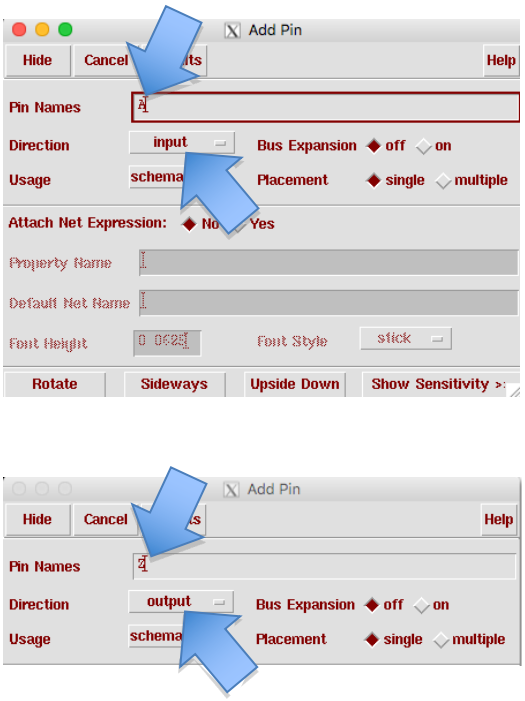
(b) Esquemático com adição dos fios conectando as instâncias

f) O próximo passo é clicar em *wire* (ou apertar a tecla "w") e conectar os dispositivos a fim de implementar a lógica do inversor. Deve-se obter um esquemático equivalente ao que está representado na figura (b) acima. Notar que o bulk do PMOS deve ser conectado a vdd! e o do NMOS a gnd!.

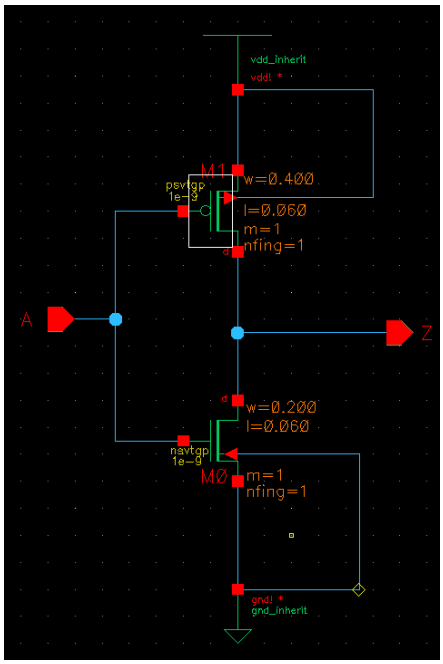
g) Dimensionar os transistores para os seguintes tamanhos: **transistor N para 0.2 µm e o transistor P para 0.4 µm**. Dessa forma obteremos uma célula com tempos de propagação balanceados. Para redimensionar os transistores: selecionar o dispositivo, clicando no mesmo, *nsvtlp* ou *psvtlp*, clicar em property (ou apertar a tecla "q"), procurar o parâmetro "Width" e modificá-lo para 0.2 ou 0.4, respectivamente. Clicar em ok.



h) Agora criar os pinos de entrada e saída. Para tanto, clicar em pin (ou apertar a tecla "p"). Criar um pino de entrada, definindo a direção (*Direction*) para "input" e nomeando-o. Para o pino aparecer clicar na janela do esquemático. Executar o mesmo procedimento para criar um pino de saída, porém dessa vez a direção deve ser definida para "output". **USAR os nomes A e Z. Depois conectar os pinos à entrada e saída do inversor.**

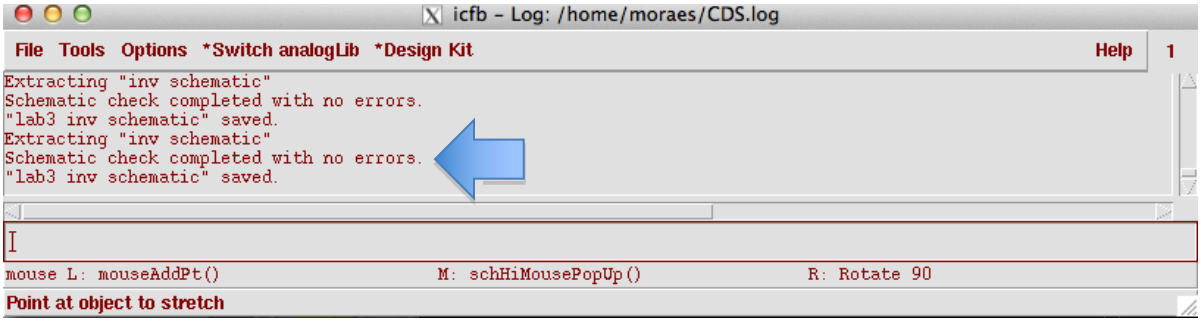


(a) Inserção dos pinos



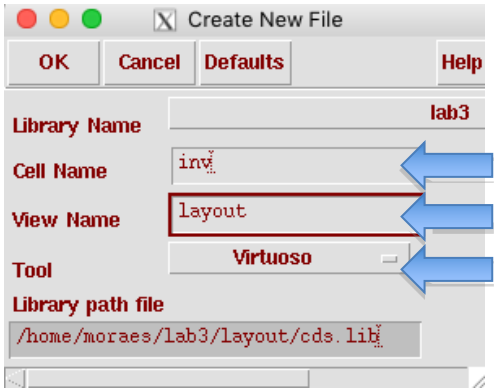
(b) Esquemático completo

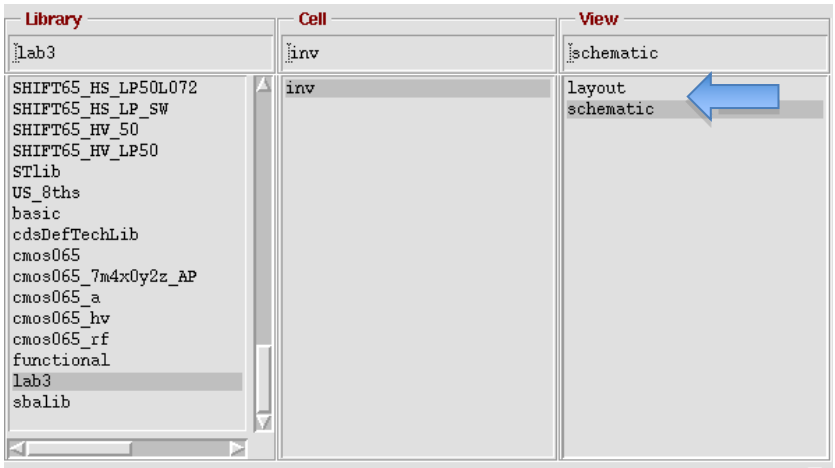
i) Clicar em **Check and Save**. Se nenhuma mensagem de alerta/erro for gerada (na janela *ifcb* – a menor), a célula está estruturalmente correta.



2 - DESENHANDO O LAYOUT

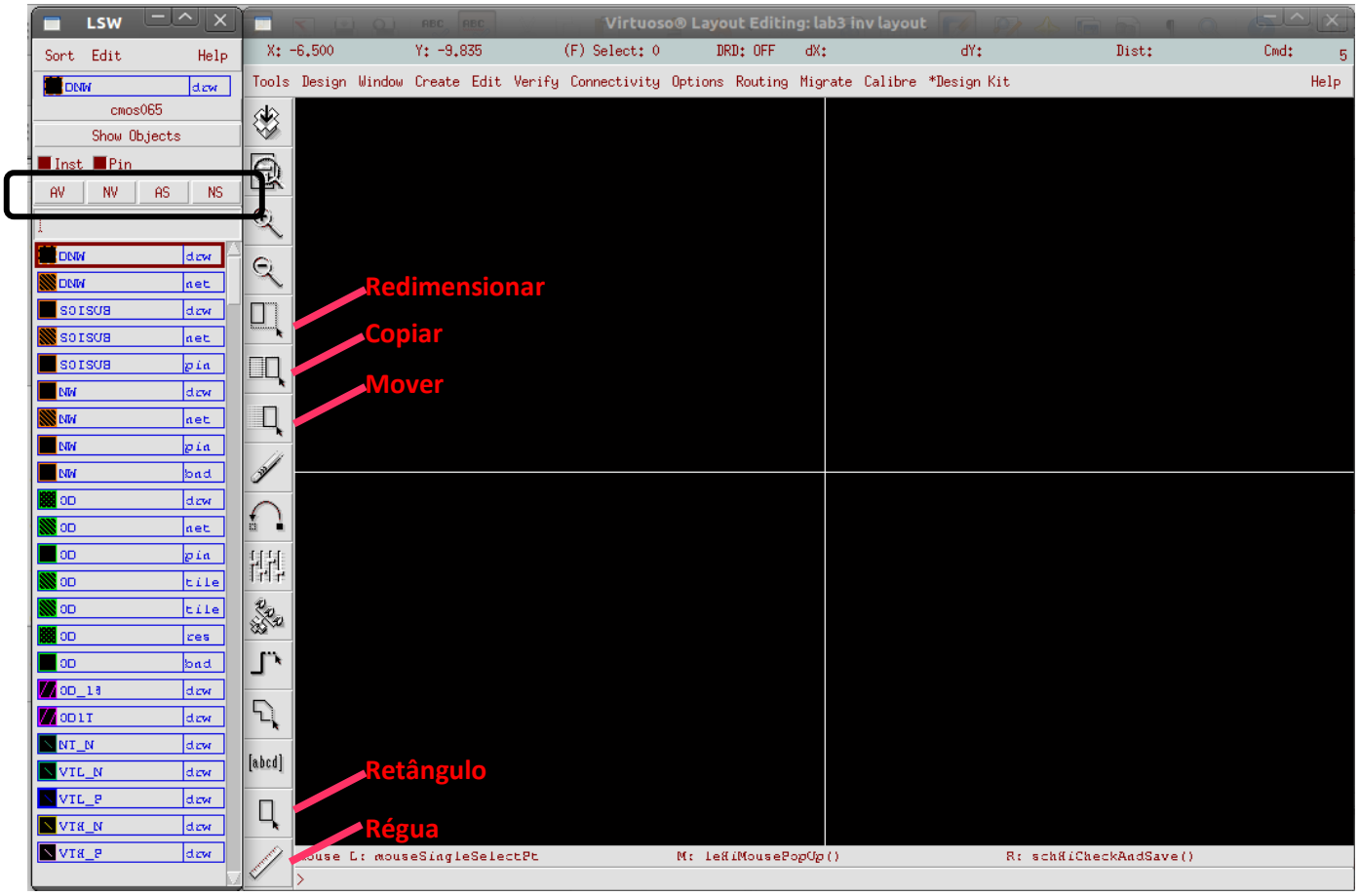
Agora criar a "visão" (*view*) layout: no Library Manager clicar em **FILE → New → Cell view**. Selecionar a biblioteca previamente criada (no caso *lab4*), digitar o nome da célula que iremos desenhar, a **visão layout** e selecionar a ferramenta **Virtuoso**.





Notar que a célula inv agora tem duas "views": *layout* e *schematic*.

O resultado é a interface gráfica da ferramenta, com a **lista de camadas à esquerda** (janela LSW).



- **NOTAR NA JANELA (*PALETTE*) DA ESQUERDA:**
  - **AV** - todos os níveis visíveis
  - **NV** – nenhum nível visível – pode-se **desabilitar** tudo e só habilitar para edição um nível desejado – **MUITO ÚTIL PARA VISUALIZAR UM SÓ NÍVEL (exemplo metal 1)**
  - **AS** - todos os níveis selecionáveis
  - **NS** – nenhum nível selecionável – pode-se desabilitar tudo e só habilitar para seleção um nível desejado – **MUITO ÚTIL PARA SELEÇÃO DE UM DADO NÍVEL (exemplo: só metal 1)**

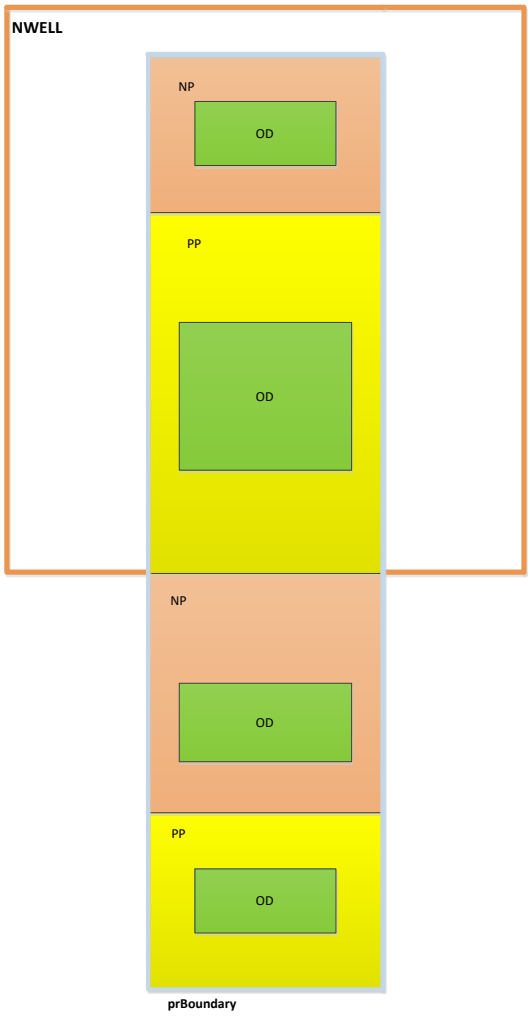
- **DICAS DA INTERFACE GRÁFICA:**
  - **A régua pode ser invocada pelo atalho "k" do teclado e todas réguas são apagadas ao utilizar-se o atalho "shift+k".**
  - O local onde o layout está armazenado é definido de forma absoluta no arquivo **cds.lib**. Por exemplo, no meu caso: "**DEFINE lab4 /home/moraes/lab4/layout/lab4".** Se o layout for copiado para outra máquina este arquivo deve ser editado manualmente.
  - Observar no topo da janela as coordenadas e as dimensões do retângulo que se está desenhando.
  - A interface gráfica trabalha com um conceito de "**empilhamento**" de comandos. Cuidar o canto inferior esquerdo, o qual indica o comando ativo – no caso o desenho de um retângulo. Pode-se por exemplo estar-se inserindo um retângulo e fazer-se zoom (F ou Z).
  - Os comandos de mover, zoom, apagar, salvar, esticar, undo, etc. concentram-se nos botões da esquerda da interface gráfica.
  - Para desfazer a última alteração no layout, pode-se usar o atalho "U" do teclado. Para refazê-la, o atalho "shift+U".
  - Para visualizar todo o layout deve-se usar o atalho "F" do teclado.
  - A ferramenta de zoom pode ser usada fazendo-se um retângulo na área a ser visualizada com o botão direito do mouse.

- **As regras de projeto serão dadas ao longo deste texto. Caso seja necessário, consultar o professor para regras mais detalhadas.**

- **Principais camadas utilizadas:**

Nível	Nome	Observação
NW	Poço N	
OD	Área Ativa (DIFUSÃO)	
NP	Região N+	Local para os transistores N
PP	Região P+	Local para os transistores P
PO	Polisilício	
CO	Contato	
M1	Metal 1	
VIA1	Via	Conecta metal 1 ao metal 2
M2	Metal 2	
prBoundary DCO	Margens	Delimitam as margens da célula

CAMADAS PARA OS TRANSISTORES



Exemplo de regras de projeto:

Nível	Largura Mínima	Espaçamento Mínimo	Overlap para CTM/V1M
PO (poli)	0,06µm	0,12µm	0,03µm
M1 (metal 1)	0,09µm	0,09µm	0,025µm

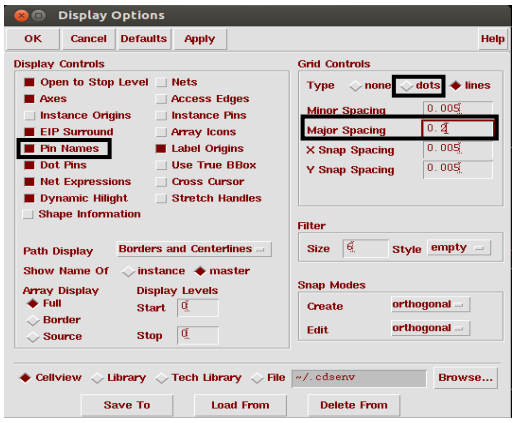
- Contatos (CO): 0,09 µm x 0,09 µm, com espaçamento entre contatos igual ou mair a 0,11 µm
- Vias (VIA1): 0,1 *x* 0,1 µm

O transistor **N** é formado por área ativa (**OD**), com **N+** ao redor (**NP**).
O transistor **P** é formado por área ativa (**OD**), com **P+** ao redor (**PP**), e poço **N** ao redor da área **P+** (**NWELL**).

- Antes de iniciarmos o projeto físico do inversor, vamos explorar as opções de exibição. Para isto, ir no menu "*options → display*".

- Selecionar a caixa "*pin names*". Explorar as opções de display, que auxiliam o projeto físico da célula.

- Reparar os controles de grid de manufatura, *X Snap* e *Y Snap Spacing*. Esses valores devem sempre estar coerentes com especificações feitas pelo fornecedor da tecnologia utilizada. Nesse caso, ambos estão definidos para 0.005. Isso significa que, para essa tecnologia, as dimensões de qualquer camada devem sempre ser múltiplas de 0.005 µm, tanto para o eixo X quanto para o eixo Y.



- Selecionar a opção "***dots***" e alterar o *major spacing* para **0.2**.

Para criar um retângulo, deve-se primeiro selecionar a camada desejada na janela LSW. O próximo passo é usar o botão "Retângulo" (atalho "R" no teclado), clicar em qualquer lugar do editor de layout, arrastar o mouse e clicar novamente, fazendo assim um retângulo da camada escolhida. Para acertar as dimensões, usar a ferramenta "Régua" (tecla "K" do teclado) para medir o tamanho desejado, e a ferramenta "Redimensionar" (atalho "S") para definir o novo tamanho. Para redimensionar, clicar na borda do retângulo e arrastar o mouse até obter-se o tamanho desejado, então clicar novamente. Cuidar para que as bordas dos retângulos respeitem o *grid*.

L1. Começar fazendo uma régua de 0,6 µm (horizontal) por 3 µm (vertical) (figura L1). Nota uma terceira régua de 0,56 µm.

L2. Selecionar a camada M1 (drawing) na janela LSW (figura L2).

L3. Desenhar dois retângulos de M1 (metal 1) 0,6 µm x 0,56 µm - serão os fios de alimentação (figura L3). **Respeitar a altura de 3 µm**. (o segundo retângulo pode ser desenhado copiando-se o primeiro).

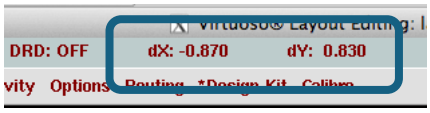
L4. Selecionar a camada OD (drawing) na janela LSW – difusão para implementar os transistores. Desenhar um retângulo de difusão com dimensão 0,48 µm x 0,2 µm – centrado no metal1 (0,06+**0,48**+0,06=**0,6**) com distância entre o M1 e OD igual a 0,255 µm (figura L4).

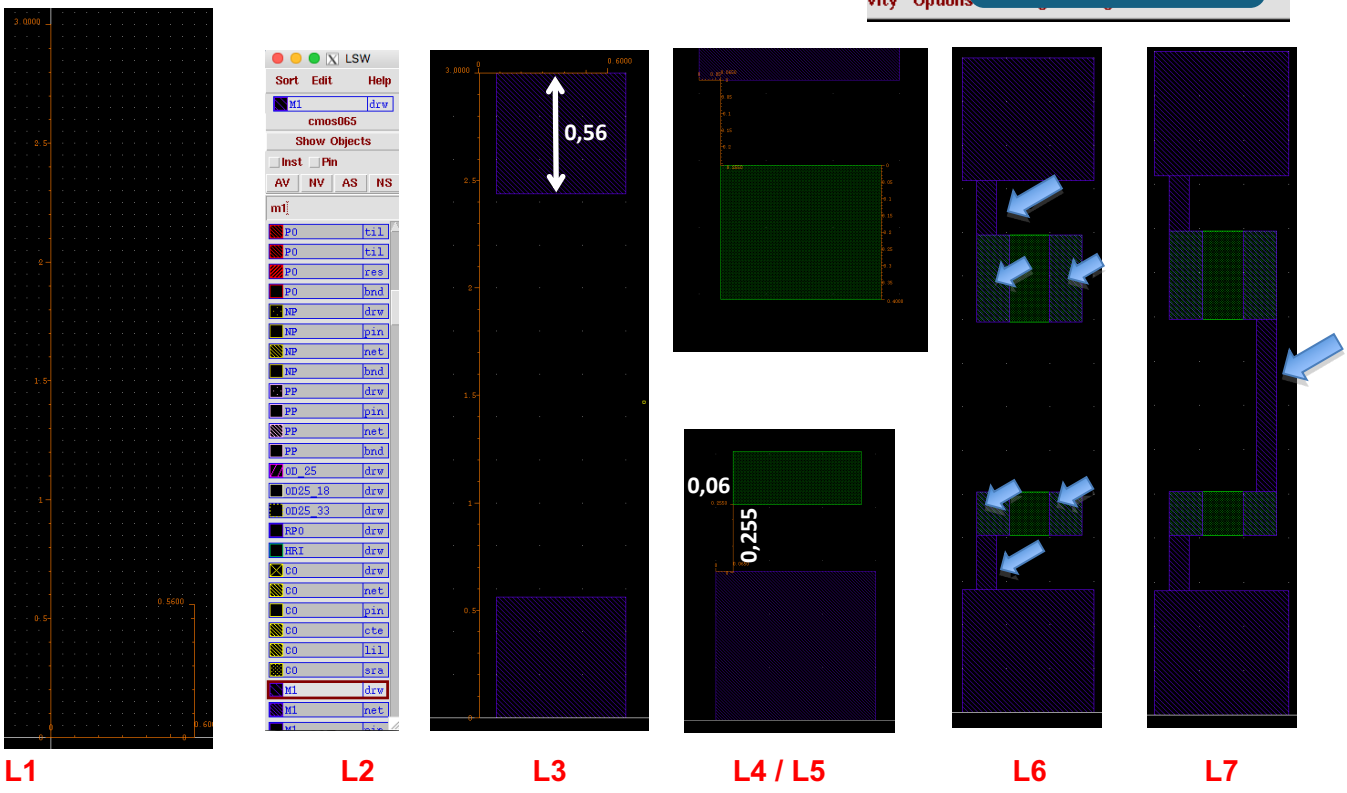
L5. Desenhar outro retângulo de difusão com dimensão 0,48 µm x 0,4 µm (figura L5)

L6. Selecionar a camada M1 (drawing) na janela LSW (figura L2). Desenhar 4 retângulos de M1 sobre os drenos/sources – nos transistores P os retângulos são de 0,15 µm x 0,4 µm, e nos transistores N os retângulos são de 0,15 µm x 0,2 µm. Desenhar dois retângulos de M1 conectando os *sources* à alimentação. Largura do M1: 0,09 µm (figura L6).

L7. Desenhar um retângulo de M1 conectando os *drenos*. Largura do M1: 0,09 µm (figura L7).

DICA: acompanhar o tamanho do retângulo no topo da janela: dX e e dY.





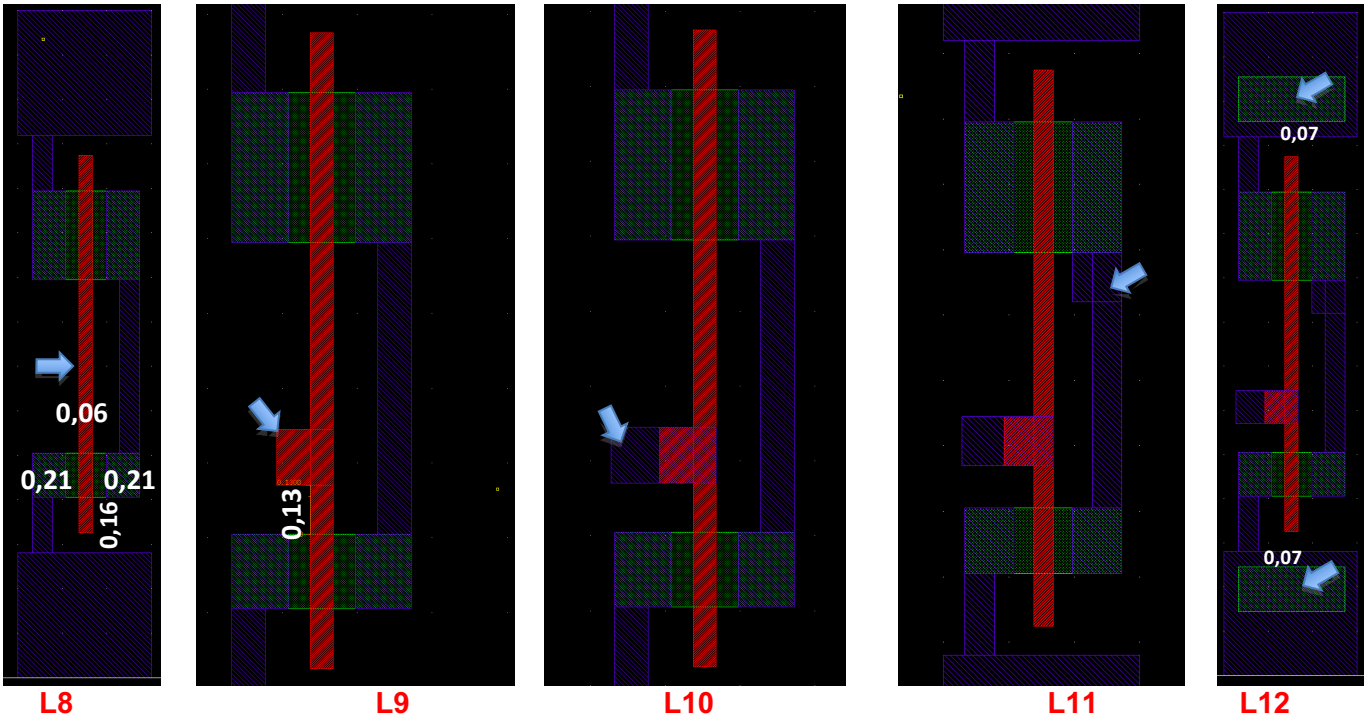
L8. Inserir um retângulo de polisilício, responsável pela definição das regiões de *gate*. Desenhar um retângulo de poli, camada PO (drawing). Esse retângulo deve ter largura de 0,06 µm e se sobrepor à difusão, representando assim o *gate* do transistor (0,21+**0,06**+0,21=**0,48**). Importante: deve haver uma "sobra" mínima de 0,16 µm após as difusões (figura L8).

L9. Inserir um retângulo polisilício, responsável pela conexão do *gate* ao mundo externo – quadrado de 0,15 µm x 0,15 µm, com distância de 0,13 µm em relação à difusão (figura L9).

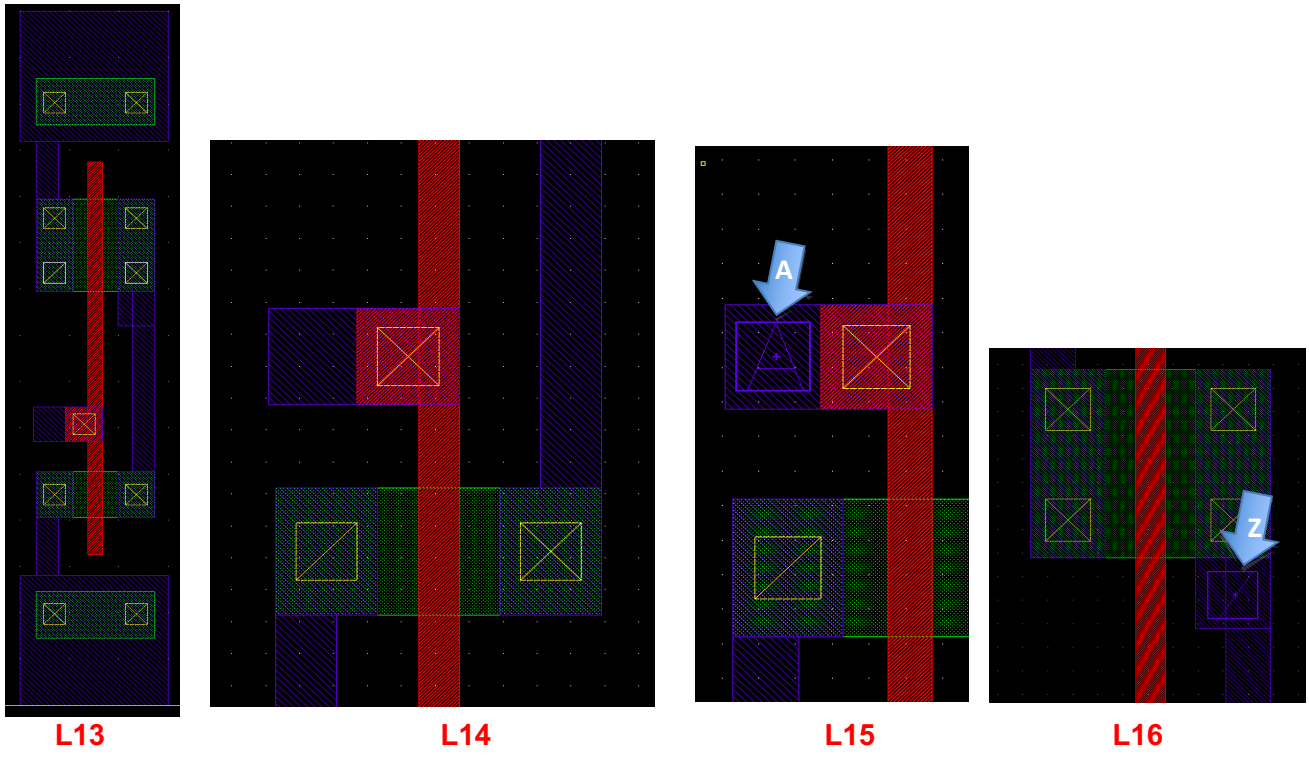
L10. Inserir um retângulo meta1, responsável por "subir" à conexão do *gate* para metal1 – quadrado de 0,28 µm x 0,15 µm (figura L10). Este retângulo deve superpor-se ao retângulo de poli do item anterior.

L11. Inserir um quadrado de 0,15 µm x 0,15 µm em meta1, junto aos drenos, para inserção do pino de saída (figura L11).

L12. **Copiar** o retângulo de difusão da parte N para as regiões de alimentação. Estes retângulos de difusão serão responsáveis pela polarização de substrato. Colar este retângulo com uma margem de 0,07 µm em relação ao metal 1 (figura L12).

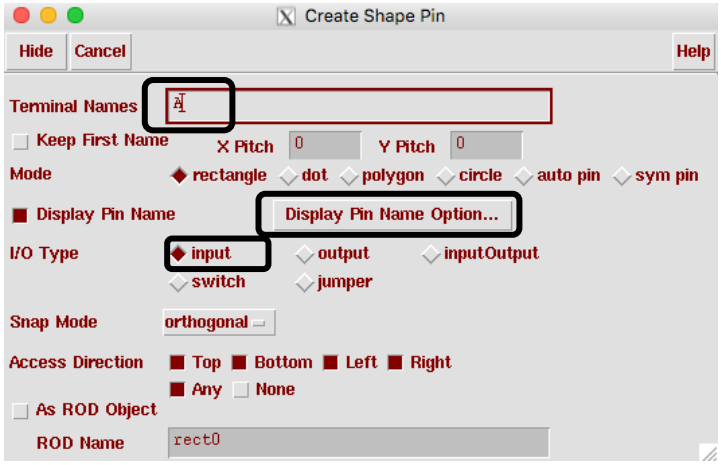


L13. Inserir os contatos – camada CO drw. Os contatos são quadrados de 0,09 µm x 0,09 µm, com espaçamento entre contatos maior ou igual a 0,11 µm, e devem ter uma margem **mínima** de 0,03 µm em relação a qualquer borda. Teremos nos transistores N 1 contato por dreno/source, nos transistores P 2 contatos por dreno/source, nas difusões de polarização 2 contatos, e um contato entre o poli e o metal 1 do *gate* (figura L13).



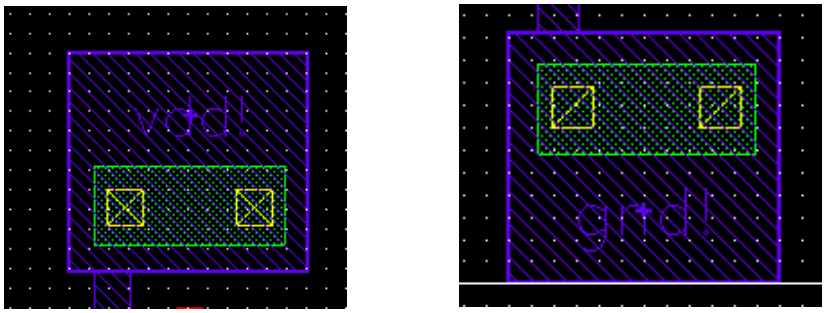
L14. **Inserção dos pinos**. Até o momento trabalhamos apenas com camadas de desenho (drw). Para os pinos iremos utilizar as camadas PIN. Selecionar a camada M1 (pin) na janela LSW. Faça um zoom na região do contato de *gate* (figura L14).

L15. Clicar em *Create* → *Pin* (atalho "ctrl p"), **selecionar o modo shape pin,** definir o pino como input e dar um nome ao pino (*A*). Esse nome deve ser o mesmo dado ao esquemático. Clicar em "*Display Pin Name Option*" e modificar o valor de Height para 0.1. Clicar em Ok. Clicar em hide, clicar em qualquer lugar do editor de layout. Desenhar um quadrado 0,1 x 0,1 µm (deixar o "+" do label dentro do quadrado), centralizado no metal ao lado do contato de *gate* (figura L15).



L16. Repetir o processo de criação para o pino de saída (Z): Desta vez selecionar o I/O type "output" e posicioná-lo no centro do quadrado de metal (figura L16).

L17. Criar dois novos pinos, para as linhas de alimentação. Primeiramente criar um pino com a camada M1 (pin), direção "input" e nome **vdd!** (com exclamação mesmo). Este pino deve ter o mesmo tamanho da camada M1 desenhada para a linha de alimentação na porção superior do layout e deve ser posicionada de forma a se sobrepor essa camada. Este procedimento deve ser repetido, porém agora criando um pino cujo nome deve ser **gnd!** e posicionar sobre a camada de metal 1 da linha de alimentação na porção inferior do layout (figura L17).



L17

Definição dos tipos de transistores

L18. Inserção dos implantes. Os implantes definem o tipo da difusão: N ou P. Desenhar os quatro retângulos de implante N e P, na seguinte ordem:

(1) PP drw, sobre a linha de alimentação gnd!, retângulo de **0,78** (X) x **0,36** (Y) (**0,09** de cada lado do metal1 → 0,78-0,6= 0,18 → 0,18/2= 0,09)

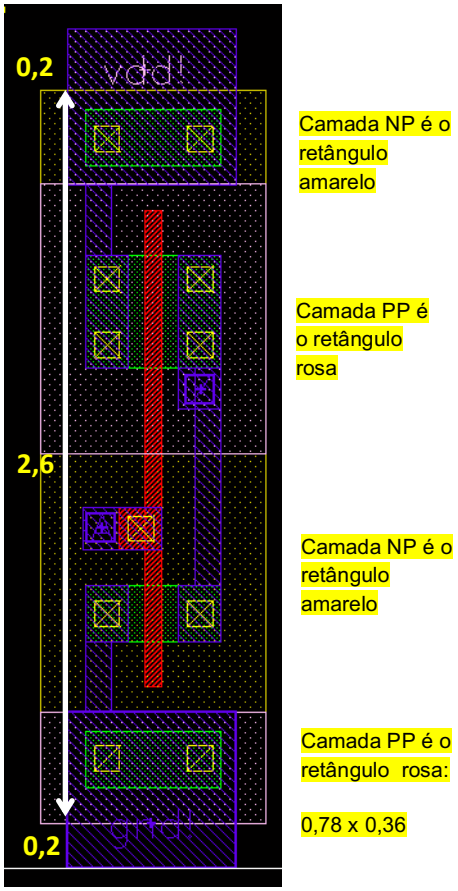
(2) NP drw, do primeiro retângulo até o centro da célula (**0,78** (X) x **0,94** (Y));

(3) PP drw, do retângulo NP drw até o metal de vdd! (**0,78** (X) x **0,94** (Y));

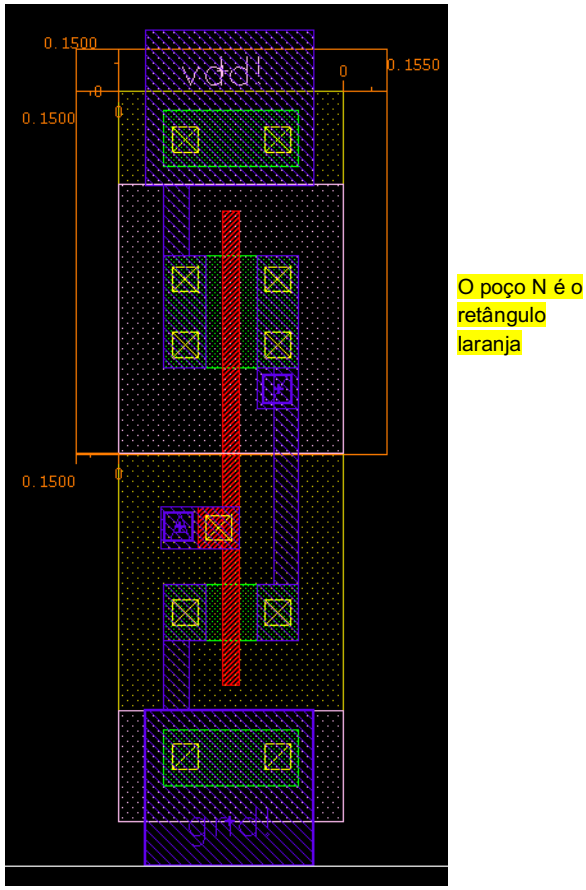
(4) NP drw, sobre a linha de alimentação vdd!, retângulo de **0,78 x 0,36**;

Todos os 4 retângulos devem ter a mesma largura e devem ser justapostos (figura L18)

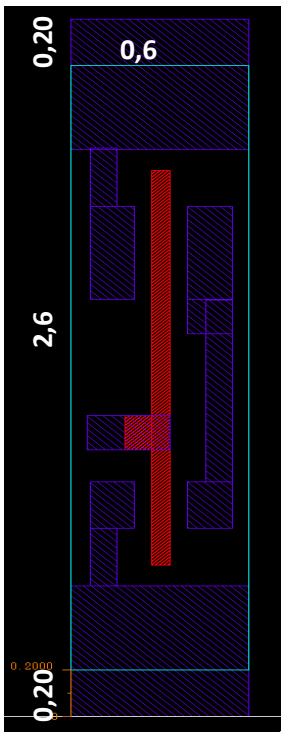
L19. Inserção do poço N. Selecionar a camada de poço N (NW drw), e fazer um retângulo com margem de 0,15 µm ao redor dos dois retângulos de implante superiores (figura L19).



L18



L19



L20

L20. Terminando a célula

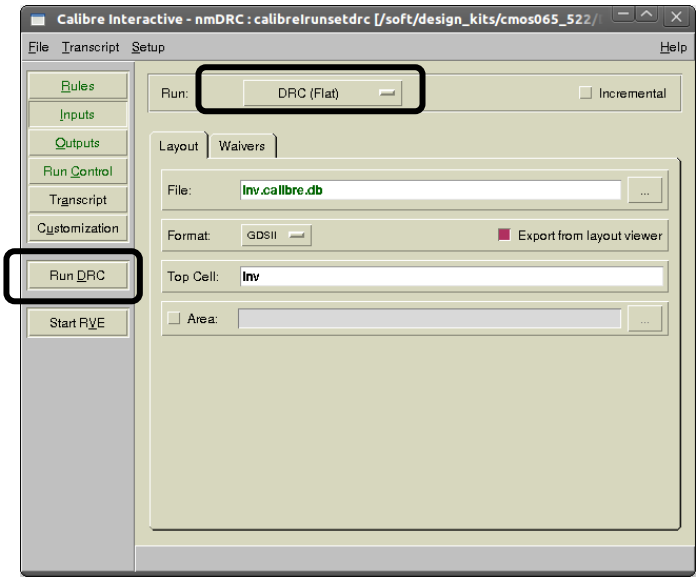
A última etapa do layout é inserir as camadas de fronteira: *prBoundary* e *DCO*

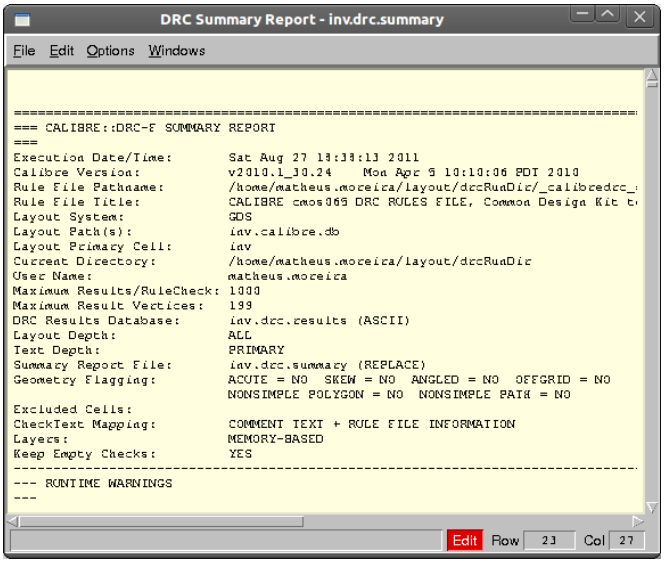
- Exibir apenas o metal 1 (na janela LSW clicar em NV e depois M1). Na figura L20 tem o poli e o metal1.
- Selecionar a camada prBoundary drw, e fazer um retângulo de 0,6 µm x 2,6 µm, como indicado na figura L20.
- Selecionar a camada DCO drw, e fazer um retângulo de 0,6 µm x 2,6 µm, exatamente sobre o retângulo prBoundary.

3 – DRC - VERIFICANDO SE O DESENHO ESTÁ CORRETO

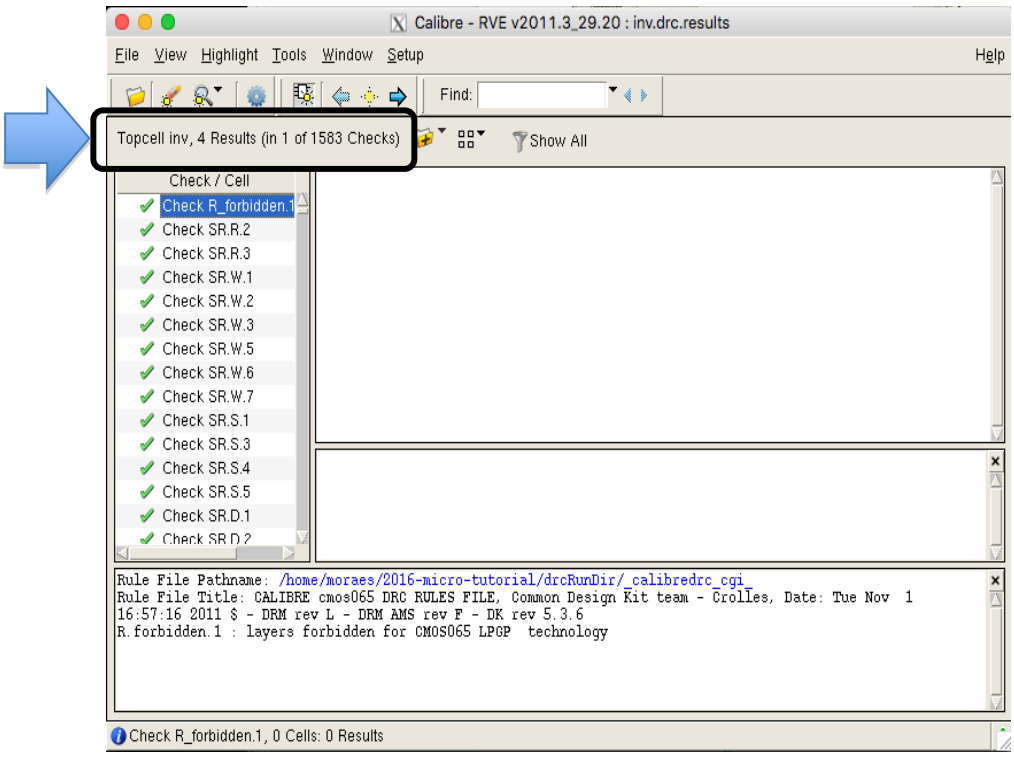
- LEMBRAR DE SALVAR A CÉLULA! Este passo é feito através da ferramenta Calibre DRC (*design rule checker*).
- Ir em "*Calibre → Run DRC*", clicar em Run DK DRC e OK.

- Escolher o modo DRC (Flat), pois desenhamos o inversor sem instanciar nenhum bloco hierárquico. Clicar em Run DRC. Caso apareça a mensagem de sobrescrever o arquivo, clicar em ok. Será gerado um relatório geral da ferramenta, mostrando todos os passos executados na verificação do layout.

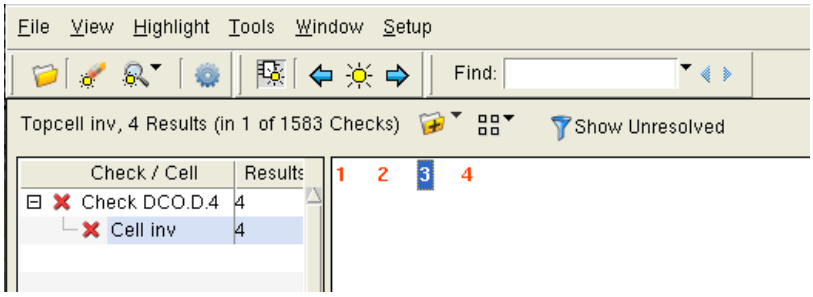




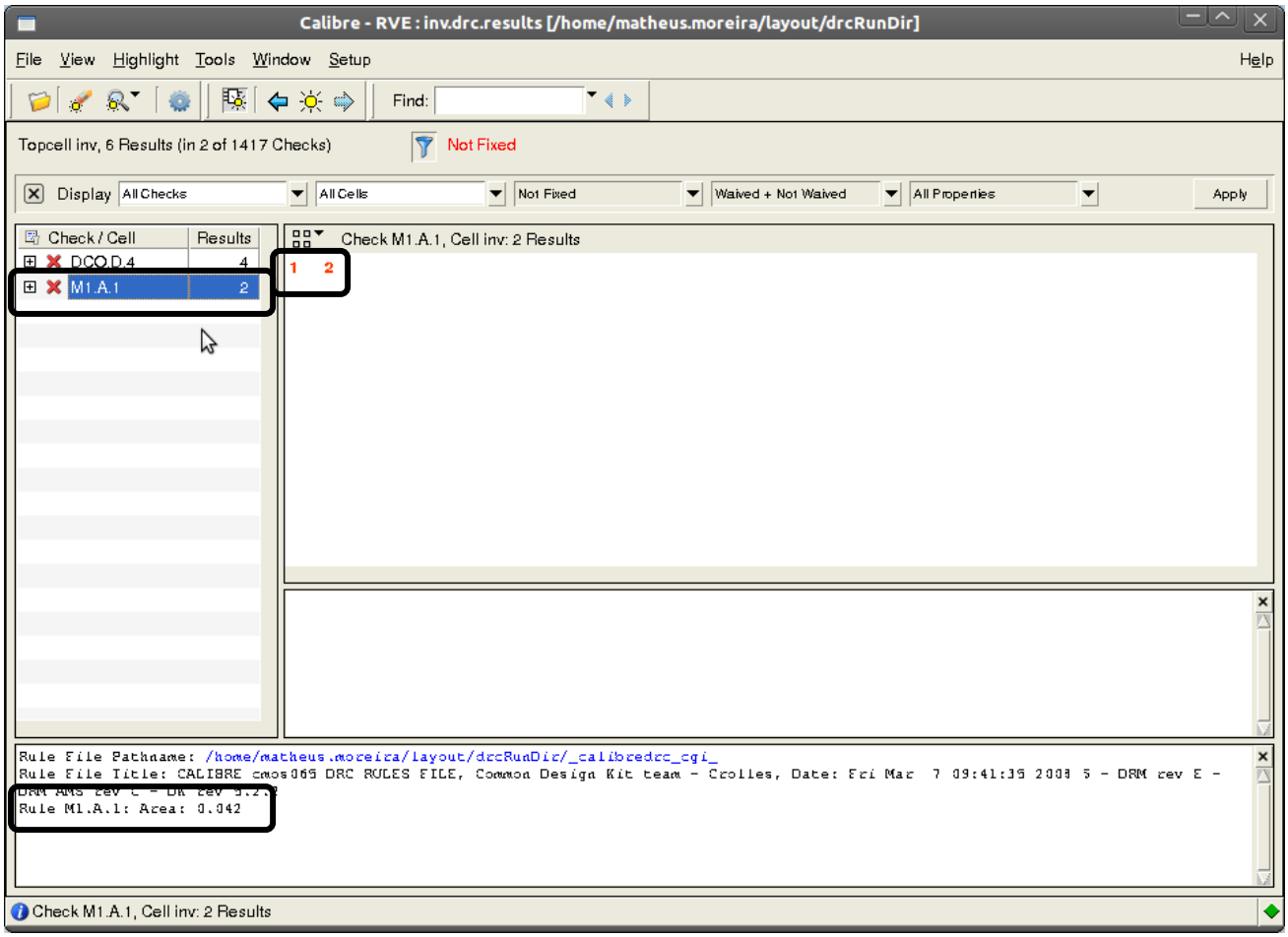
- A janela de resultados é apresentada abaixo. Notar que forma indicados 4 erros de DRC. Clicar no funil para filtrar apenas as regras que podem ter sido violadas, para assim corrigir o layout. Selecionar a opção "Not Unresolved" e clicar em *apply*.



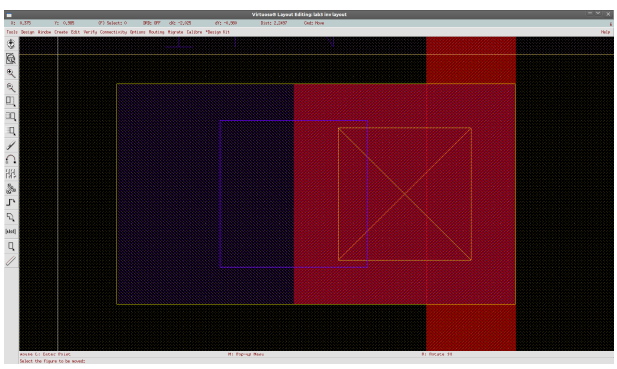
- Os 4 erros de DRC são referentes à camada DCO e podem ser ignorados.



- Havendo erros de DRC em camadas de desenho (*DRW*):
  - Serão sinalizados com o código da regra de projeto. Ao clicar na violação, será dada uma explicação do erro. Nesse caso, a regra de área mínima para metal 1 não foi respeitada em 2 ocasiões. Notar que o valor de área mínima é dado para que o layout seja corrigido.



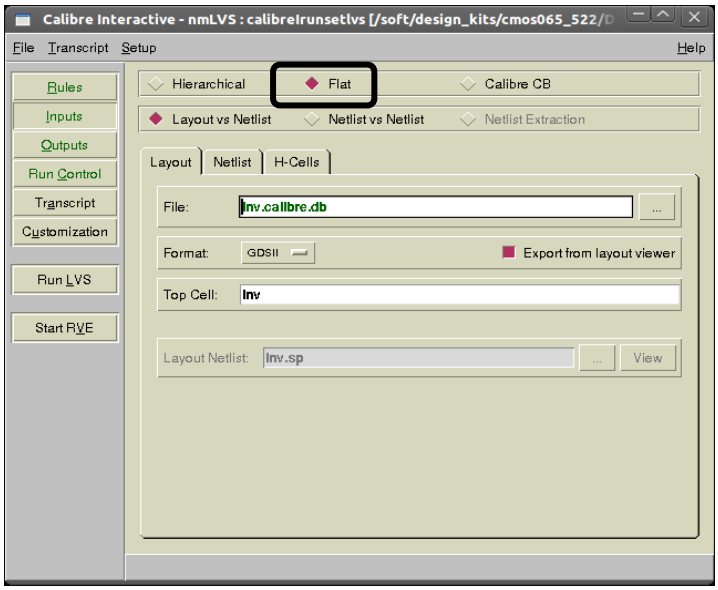
  - Ao clicar duas vezes em uma das violações, a janela do virtuoso (com o layout) mostra exatamente onde essa violação acontece.

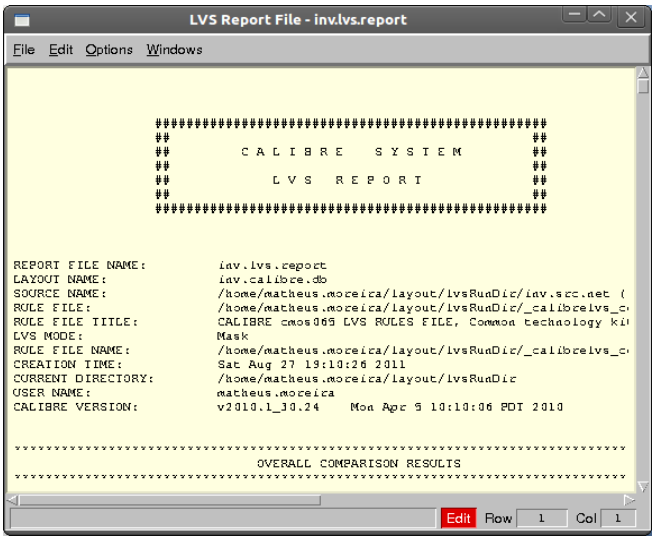


- Corrigir os possíveis erros de DRC e executar novamente a ferramenta de verificação. **Deixar o layout livre de violações nas camadas de desenho**. Essa verificação, garante que o layout projetado é "fabricável". Quando uma fábrica recebe as máscaras de um layout, ela verifica se o layout respeita as regras de projeto físico. Se sim, o projeto será fabricado. Se não, ele retorna ao projetista.

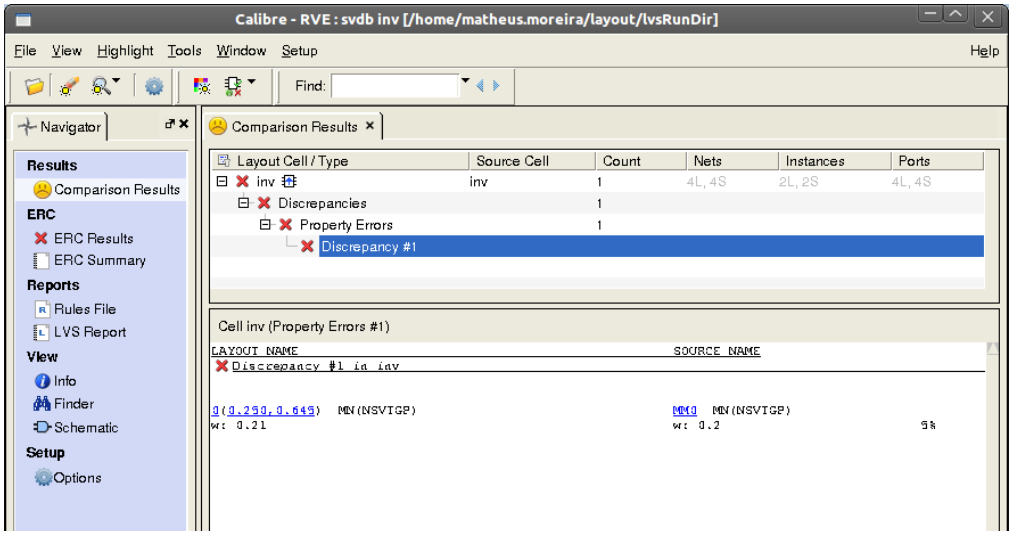
4 – LVS - VERIFICANDO SE O LAYOUT E O ESQUEMÁTICO SÃO EQUIVALENTES

- Essa verificação, LVS (*layout versus schematic*), garante que o layout projetado implementa a mesma lógica que o esquemático. Este passo é feito através da ferramenta Calibre LVS
- Ir em "*Calibre → Run LVS*". Escolher o modo *Flat* novamente e clicar em Run LVS. Caso apareça uma mensagem de sobrescrever arquivo, clicar em OK.

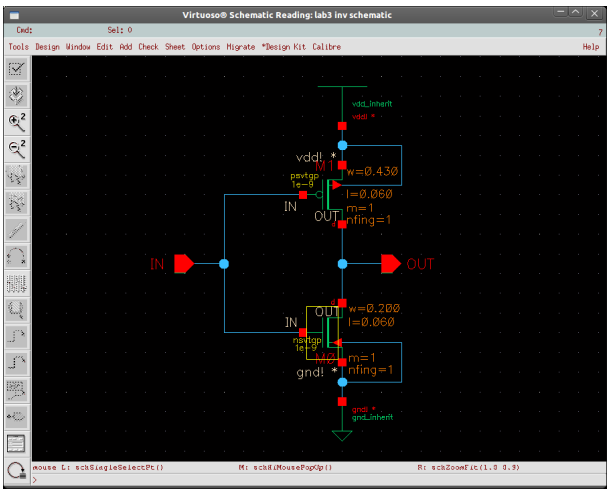


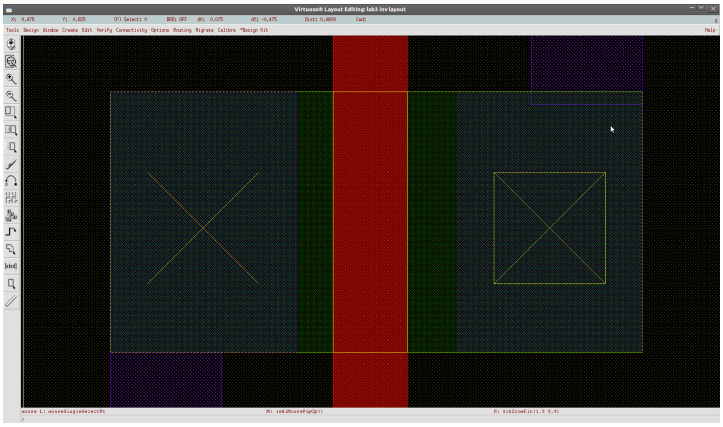


- Na janela de resultados da verificação LVS exibe-se eventuais erros. Nesse exemplo, o layout não fecha com o esquemático. Ao expandir a discrepância, obtemos sua causa. Nesse caso, o transistor NMOS MN, desenhado no layout possui um W de 0.21 µm, enquanto o transistor equivalente, do esquemático, possui um W de 0.20 µm.

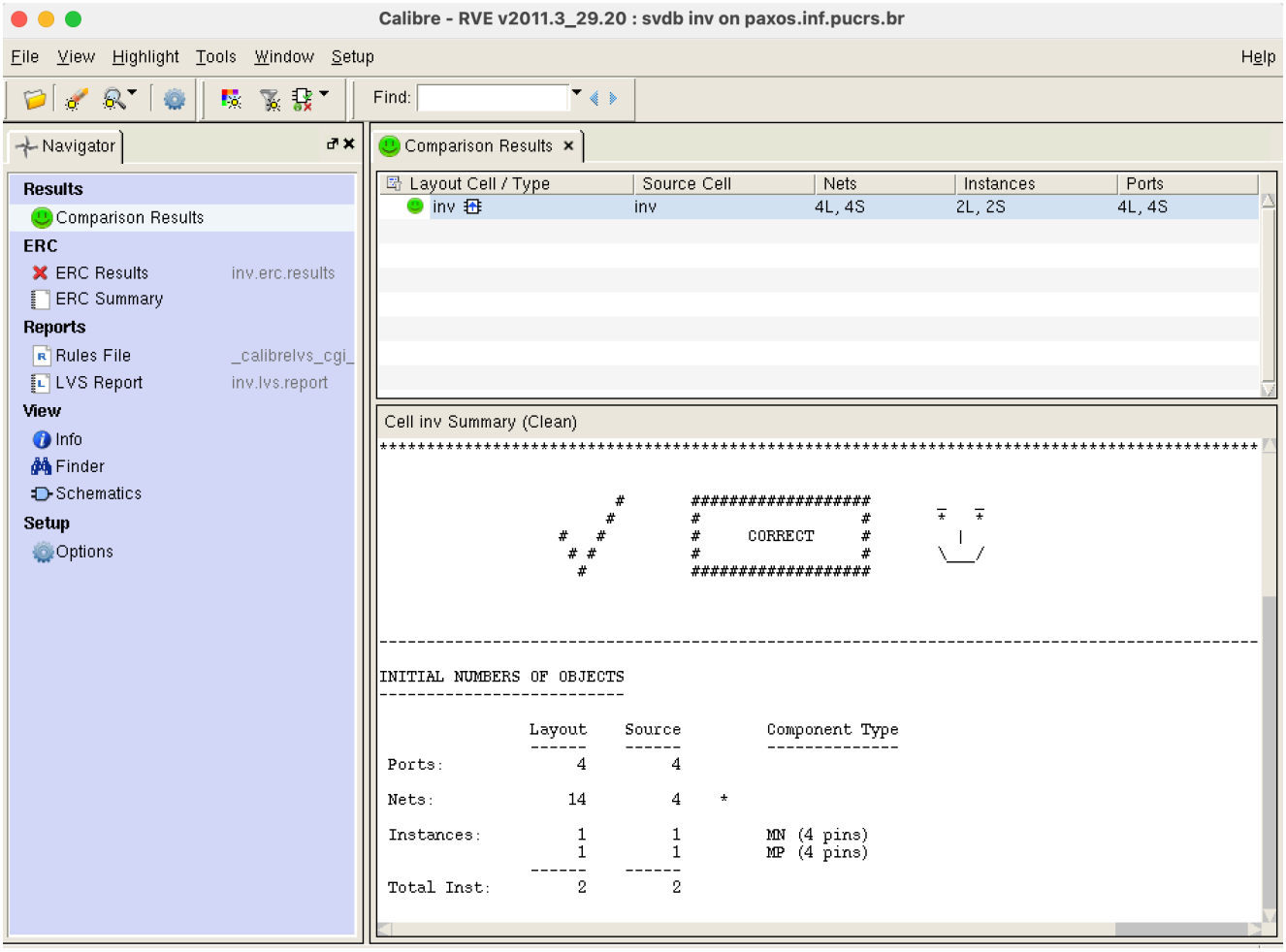


- Ao clicarmos na discrepância, a ferramenta mostra, nas duas *views* (*schematic* e layout), onde encontra-se o problema.





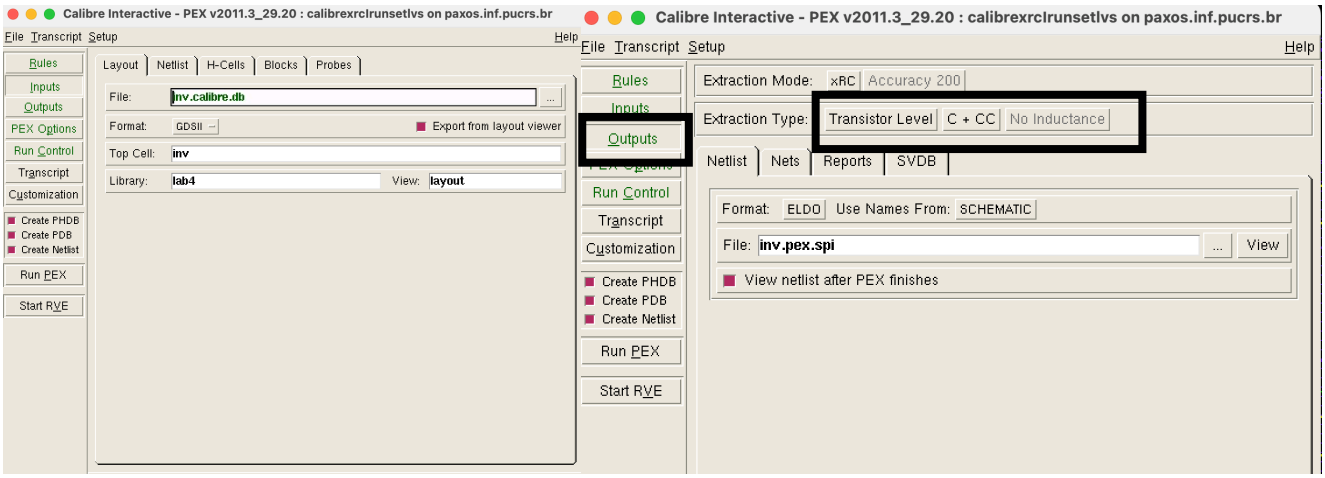
- Caso alguma violação de LVS aconteça, corrigir todos os problemas e executar a ferramenta novamente. Garantir que o esquemático e o layout implementam a mesma função lógica. O resultado do LVS deve ser:



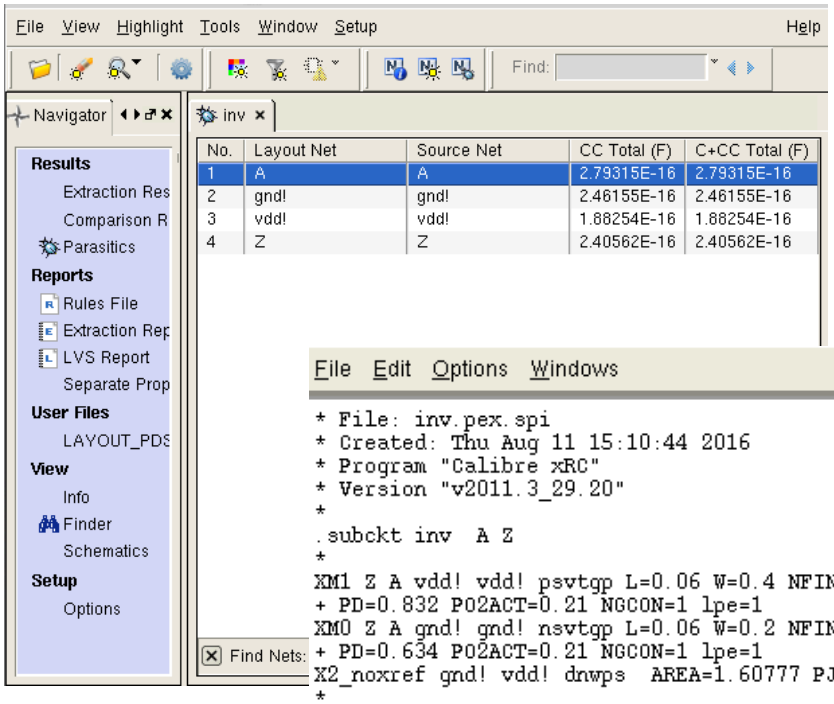
Atenção: os erros mais comuns são relacionados a pinos – os pinos devem ser desenhados com a camada M1 pin

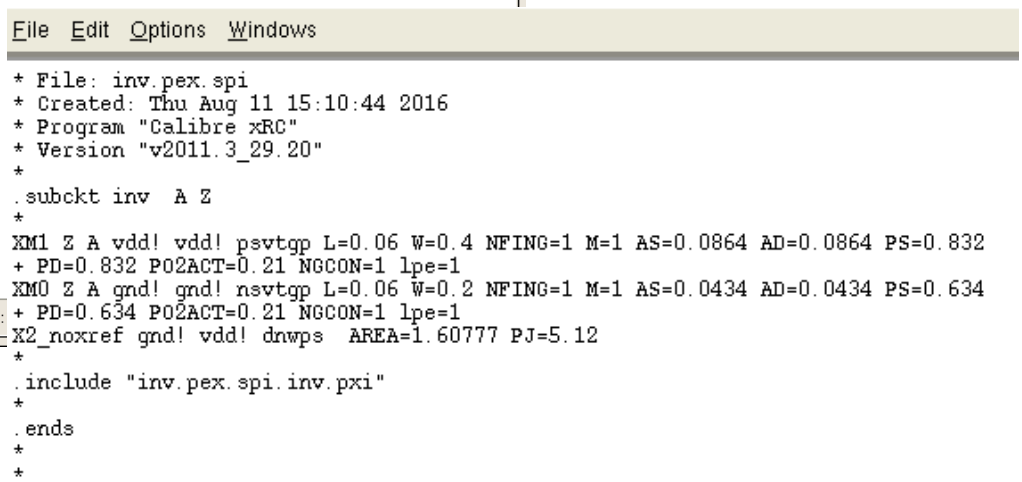
5 - EXTRAÇÃO ELÉTRICA

- Este passo é responsável pela realização da extração elétrica, ou seja, pela geração do netlist *spice*. Ir em "*Calibre → Run PEX*". Usar as opções padrão e clicar em OK. Caso apareça alguma mensagem, solicitando para salvar o arquivo de configurações, clicar em não salvar. A ferramenta irá perguntar se deseja criar uma pasta para o ambiente PEX, clicar em criar. Clicar na opção "*Outputs*" da ferramenta, e escolher as opções de extração, C+CC e sem indutância. Clicar em Run PEX.



- Clicar em Run PEX. Serão geradas duas janelas, uma mostrando os parasitas extraídos e outra mostrando a descrição em *spice* do circuito obtido.





- Reparar que foram criados três arquivos que descrevem, em *spice*, o circuito extraído. No terminal, no diretório pexRunDir, encontram-se os arquivos com essas descrições. Nesse exemplo os arquivos são (eles deverão ser incluídos no relatório):
  - inv.pex.spi (**arquivo a ser utilizado para a simulação**)
  - inv.pex.spi.inv.pxi (capacitâncias parasitas)
  - inv.src.net (netlist)

6 - SIMULAÇÃO ELÉTRICA

Fechar as ferramentas da Cadence, sair do csh, e configurar o simulador elétrico.
Comandos:

```
exit
module purge
module load ic spectre
```

- Abrir o arquivo com a descrição spice do inversor (arquivo pexRunDir/inv.pex.spi) e modificar sua interface para que passe a possuir as entradas para as linhas de alimentação. Na linha que descreve o ".subckt (NOME_DA_CELULA)" é dada a pinagem do circuito. **Adicionar** ao final as duas entradas: vdd! e gnd!. No exemplo aqui descrito, o resultado é o seguinte:

```
* File: inv.pex.spi
* Created: Wed May 11 08:06:01 2022
* Program "Calibre xRC"
* Version "v2011.3_29.20"
*
.subckt inv  A Z vdd! gnd!
*
XM1 Z A vdd! vdd! psvtgp L=0.06 W=0.4 NFING=1 M=1 AS=0.0864 AD=0.0864 PS=0.832
+ PD=0.832 PO2ACT=0.21 NGCON=1 lpe=1
XM0 Z A gnd! gnd! nsvtgp L=0.06 W=0.2 NFING=1 M=1 AS=0.0434 AD=0.0434 PS=0.634
+ PD=0.634 PO2ACT=0.21 NGCON=1 lpe=1
X2_noxref gnd! vdd! dnwps  AREA=1.566 PJ=5.06
*
.include "inv.pex.spi.inv.pxi"
*
.ends
*
```

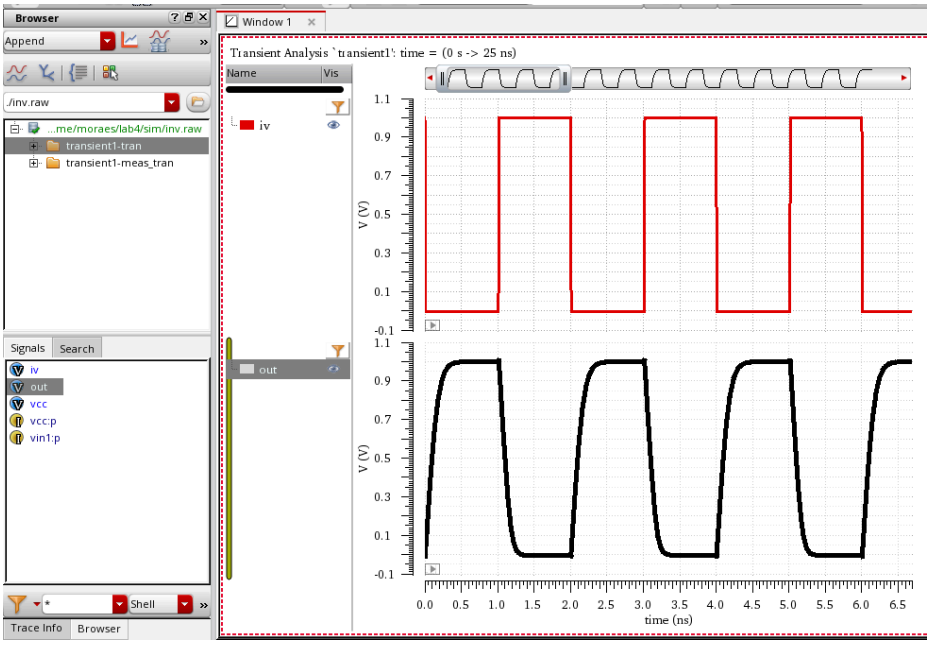
- Ir para o diretório de simulação: **cd sim**
- Abrir o arquivo **inv.sp**. Observar na linha 15 a inclusão do arquivo "../pexRunDir/inv.pex.spi" via comando include. Na linha 20 há a instanciação do subcircuito (X1 iv out vcc 0 inv). **Este é o arquivo que gera os estímulos para o circuito que projetamos.**

- Executar a simulação elétrica: **spectre inv.sp**. Observar o arquivo **inv.measure:**

```
date                 :  7:17:59 PM, Fri Apr 7, 2023
design               :  * inversor - simulação do circuito extraído do layout projetado
simulator            :  spectre
version              :  15.1.0.257


Measurement Name     :  transient1
Analysis Type        :  tran
td                = 84.6095
tdescida          = 8.46095e-11
ts                = 91.964
tsubida           = 9.1964e-11
```

Ou seja, o tempo de subida é de 84.6 ps e o de descida 91.964 ps.

- Executar a ferramenta *viva*. Selecionar os probes no sinal de entrada e saída (iv e out) e enviá-los para o gráfico.



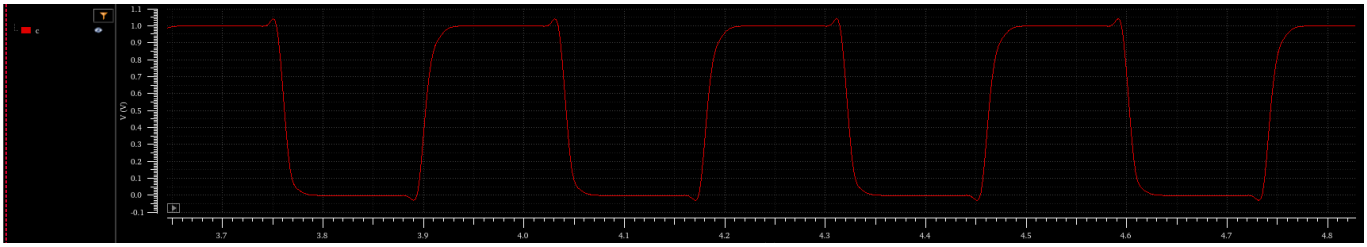
Nosso inversor funciona!

Simulação elétrica de um anel de inversores

- Agora vamos simular um anel de inversores. Verifique se o arquivo **anel.sp** contém a referência correta ao arquivo extraído (../pexRunDir/inv.pex.spi) e as 15 instanciações fazem referência ao inversor. Executar o seguinte comando: **spectre anel.sp**

```
more anel_inv.measure
date                    :  7:20:25 PM, Fri Apr 7, 2023
design                  :  * inversores em anel para cálculo de frequência com inversor
extraído
simulator               :  spectre
version                 :  15.1.0.257


Measurement Name    :  transient1
Analysis Type       :  tran
freq_ghz          = 3.56905
periodo           = 0.280187
tf                = 6.27838e-12
tr                = 1.24008e-11
```

- Visualize a forma de onda do oscilador:



- **Alterar agora o anel para 21 inversores e fornecer no relatório a nova frequência. Lembrar de alterar o measure para considerar os 21 inversores:**

```
.measure tran periodo  param = '(tf+tr) * 1e9 * 21'
.measure tran freq_ghz  param = '1/periodo'
```

7 - MODELO ABSTRATO

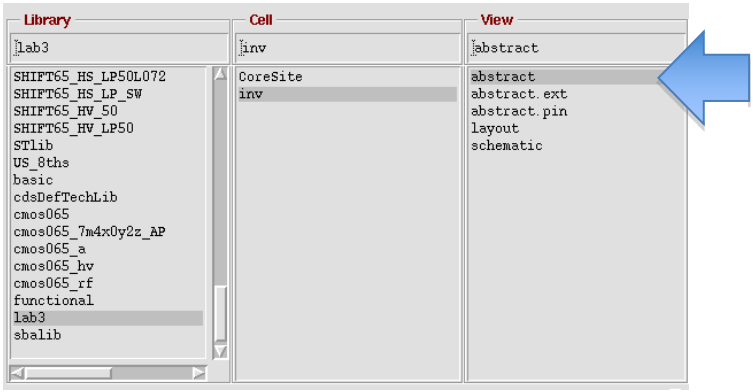
- Agora, devemos criar o modelo abstrato para podermos usar a célula que projetamos em um circuito descrito em VHDL. Este modelo indica para a ferramenta a disposição dos metais de roteamento, que ela precisará utilizar na hora de colocar a células no circuito e interligá-las.

- **Este passo será por script.** Abra o *script* **gera_abstract.txt** e verifique se os comandos a seguir conferem com a célula inversor:

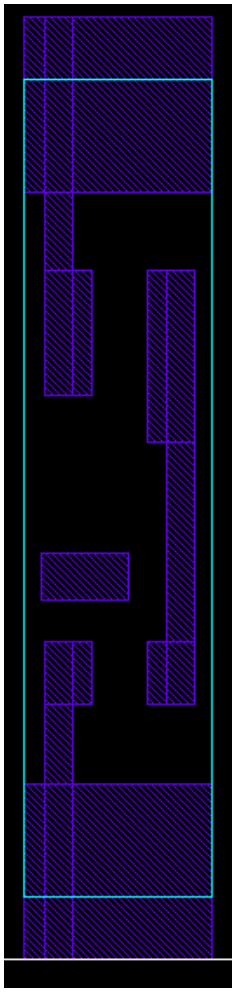
```
absSetLibrary("lab4")
absSetOption("ViewLayout" "layout")
absSetOption("ViewLogical" "schematic")
....
absSelectCell("inv")
```

- **Executar os passos abaixo** – o comando *abstract* pode demorar um pouco. Ele gera a *view* abstract e o arquivo inv.lef

  a) **Executar: module purge**
  b) **carregar "module load ic/5.1.41" (não pode ser outra versão)**
  c) **ir para a pasta ".../<microX>/lab4" e abrir o shell "csh"**
  d) **no csh executar "*source .cshrc_cmos065*"**
  e) **Executar: setenv CDS_AUTO_64BIT**
     (comando opcional, se houver erro na geração do LEF)
  f) **após executar: abstract -replay gera_abstract.txt**
  g) **abrir o icfb (icfb&)**



A *view* abstract (ao lado) contém as camadas de metal da célula, que são obstáculos para roteamento, e a fronteira da célula.

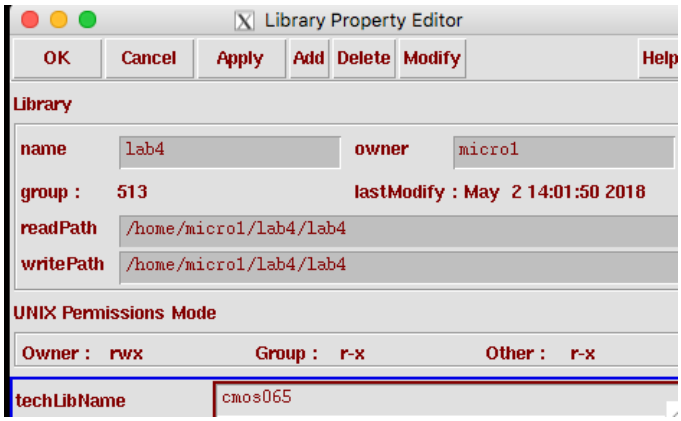


- **SAIR DO icfb**

- **SUBSTITUIR** no arquivo LEF todas as ocorrências de CoreSite por CORE.
  Para esta ação execute: sed -i 's/CoreSite/CORE/g' inv.lef

```
..
MANUFACTURINGGRID    0.005000 ;
MACRO inv
    CLASS CORE ;
    FOREIGN inv 0 0.2 ;
    ORIGIN 0.000 -0.200 ;
    SIZE 0.600 BY 2.600 ;
    SYMMETRY X Y ;
    SITE CORE ;
    PIN A
```

- Um **erro** observado nesta etapa é o **layout** ficar **desvinculado** da biblioteca, e os retângulos do inversor "desaparecerem". Para resolver este problema, no *library manager* exibir as propriedades da biblioteca e em *techLibName* digitar cmos065:



8 - CARACTERIZAÇÃO ELÉTRICA – Geração do LIB

O objetivo desta etapa é gerar um arquivo om informações de atraso e consumo da célula projeta.

Ir para o diretório: **characterization**

- Olhar o conteúdo dos arquivos ***liberate.tcl***, responsável por definir o simulador utilizado, o nome dos nodos de alimentação, além da célula que vamos caracterizar, neste caso o *inv*. Se houver várias células para se caracterizar, podemos por um * e a ferramenta caracteriza tudo.
- Este arquivo define os parâmetros para caracterização, como as condições de operação (nominal – 1V 25C).

Executar os seguintes comandos para realizar a caracterização da célula:

```
cd characterization/
module purge
module load liberate spectre
liberate liberate.tcl
```

- O resultado é o arquivo inv.lib. Agora temos nossa célula no formato *liberty*: **inv.lib**. Este formato é um padrão, que contém informações de consumo de potência da célula, seus pinos de entrada e saída, sua função lógica, os tempos nos arcos da célula (tempo de propagação de um valor da entrada para saída) e as capacitâncias de entrada e saída. Estas informações são utilizadas pela ferramenta para geração de relatórios e validação do funcionamento dos projetos que vierem a utilizar esta célula.

  Por exemplo:

```
...
        cell_fall (delay_4x5) {
           index_1 ("0.003, 0.08, 0.16, 0.31");
           index_2 ("0.001, 0.003, 0.01, 0.025, 0.08");
           values (
             "0.00768426, 0.0140939, 0.0362238, 0.0835454, 0.256948", \
             "0.0165469, 0.0285835, 0.0574656, 0.104479, 0.277346", \
             "0.0185872, 0.0343237, 0.072064, 0.127893, 0.299397", \
             "0.0189462, 0.0390662, 0.0878143, 0.160094, 0.342244" \
           );
         }
..
```

Neste trecho do arquivo *lib* temos o atraso (em *ps*) para 4 valores de carga de saída (index_1) e 5 valores de rampa de entrada no pino A (index_2). Obviamente quanto maior a carga e maior a rampa de entrada, maior o atraso.

- Abrir o datasheet gerado para a célula: firefox datasheet/index.html. Esta página exibe os dados de caracterização elétrica da célula projetada.

inv_datasheet Library

Cell Groups
INV

INV

inv_datasheet Cell Library: Process , Voltage 1.00, Temp 25.00

Truth Table

INPUT	OUTPUT
A	Z
0	1
1	0

Pin Capacitance Information

Cell Name	Pin Cap(pf)	Max Cap(pf)
	A	Z
inv	0.00103	0.08000

Leakage Information

Cell Name	Leakage(nW)		
	Min.	Avg	Max.
inv	0.00000	12.92730	17.05510

Delay Information

Delay(ns) to Z rising :

Cell Name	Timing Arc(Dir)	Delay(ns)		
		Min	Mid	Max
inv	A->Z (FR)	0.00807	0.08152	0.37331

Delay(ns) to Z falling :

Cell Name	Timing Arc(Dir)	Delay(ns)		
		Min	Mid	Max
inv	A->Z (RF)	0.00767	0.07205	0.34223

Power Information

Internal switching power(pJ) to Z rising :

Cell Name	Input	Power(pJ)		
		min	mid	max
inv	A	0.00000	0.00000	0.00000
	A	0.00051	0.00094	0.00089

Internal switching power(pJ) to Z falling :

Cell Name	Input	Power(pJ)		
		min	mid	max
inv	A	0.00000	0.00000	0.00000
	A	0.00000	0.00036	0.00030

9 - SÍNTESE LÓGICA

Este passo é responsável por fazer o mapeamento de uma descrição RTL (VHDL) para a implementação em uma dada tecnologia, usando células disponíveis na biblioteca para implementar a mesma lógica descrita no RTL.

Ir para o diretório: synthesis

Configurar a ferramenta de síntese lógica:

```
module purge
module load genus/211 innovus/211
```

- O primeiro arquivo a analisar é o **load.tcl**.
  - Na linha 11 definimos o TOP a ser sintetizado: *set DESIGN_TOP "**anel**"*.
  - Na linha 23 deste arquivo (comando *set_db library*) indicamos que vamos utilizar o inversor previamente caracterizado: *../characterization/**inv.lib***, além das células da biblioteca cmos065.
  - Na linha 25 deste arquivo (comando *set_db lef_library*) indicamos a geometria da célula projetada: *../**inv.lef***, além das células da biblioteca cmos065.

- O segundo arquivo é o que contém os parâmetros para a síntese - **cmd_genus**:

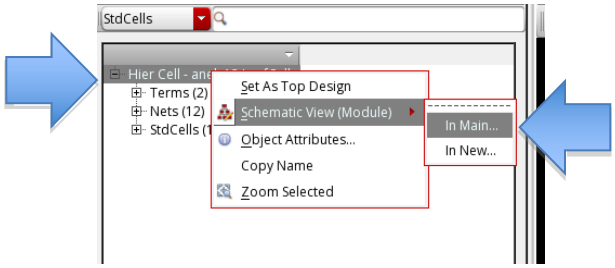
```
include load.tcl
read_hdl -vhdl anel.vhd

elaborate ${DESIGN_TOP}

#Post synthesis reports
report_area   > ${REPORTS_PATH}/${DESIGN_TOP}_area_synth.txt
report_timing > ${REPORTS_PATH}/${DESIGN_TOP}_timing_synth.txt
report_power  > ${REPORTS_PATH}/${DESIGN_TOP}_power_synth.txt
report_gates  > gates.txt

#Generate sdc pos synthesis
write_sdc > ${OUTPUTS_PATH}/${DESIGN_TOP}.sdc

#Generate sdf pos synthesis
write_sdf > ${OUTPUTS_PATH}/${DESIGN_TOP}_synth.sdf

#Build physical synthesis environment
write_design -innovus -base_name layout/${DESIGN_TOP}
```

- No diretório src há o arquivo que iremos sintetizar, um anel de inversores, com 10 instâncias do inversor projetado e uma porta NAND (HS65_GS_NAND2X2) da biblioteca de células: **anel.vhd.**

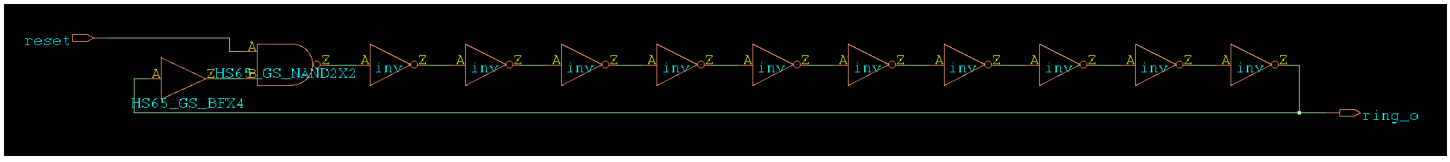
- Invocar a ferramenta de síntese lógica: **genus -gui**

- Executar o seguinte comando para realizar a síntese lógica: **include cmd_genus**

- Agora podemos visualizar o circuito gerado pela ferramenta. Na interface gráfica selecione o *anel* e abrir o esquemático, como na figura ao lado.



- O resultado é apresentado abaixo. Observar que a ferramenta instanciou corretamente os 10 inversores, a NAND da biblioteca, e um buffer (HS65_GS_BFX2) para quebrar o laço de realimentação.



- Executar o relatório de área: **report_gates**

```
@genus:root: 4> report_gates
============================================================
  Generated by:           Genus(TM) Synthesis Solution 22.15-s086_1
  Generated on:           May 15 2024  11:08:58 am
  Module:                 anel
  Operating conditions:   PVT_1V_25C
  Interconnect mode:      global
  Area mode:              physical library
============================================================
```

```
      Gate       Instances   Area      Library
-------------------------------------------------
HS65_GS_BFX4             1   2.080    CORE65GPSVT
HS65_GS_NAND2X2          1   2.080    CORE65GPSVT
inv                     10  15.600    inv
-------------------------------------------------
total                   12  19.760


  Library    Instances  Area  Instances %
-----------------------------------------
CORE65GPSVT          2  4.160        16.7
inv                 10 15.600        83.3


     Type       Instances  Area  Area %
---------------------------------------
inverter               10 15.600   78.9
buffer                  1  2.080   10.5
logic                   1  2.080   10.5
physical_cells          0  0.000    0.0
---------------------------------------
total                  12 19.760  100.0
```

- Para sair: `@genus:root: 5>` **exit**

10 - SÍNTESE FÍSICA

Permanecer no diretório **synthesis**

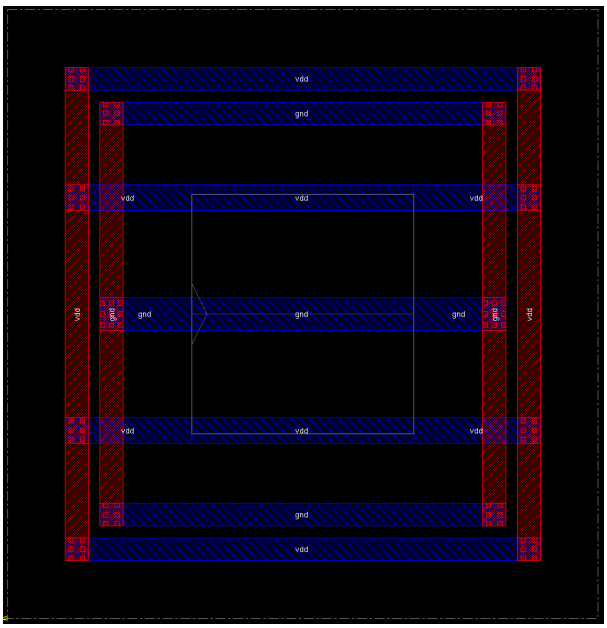
- Invocar a ferramenta de síntese física: **innovus -common_ui**
- Executar o comando no terminal: **source cmd_innovus**

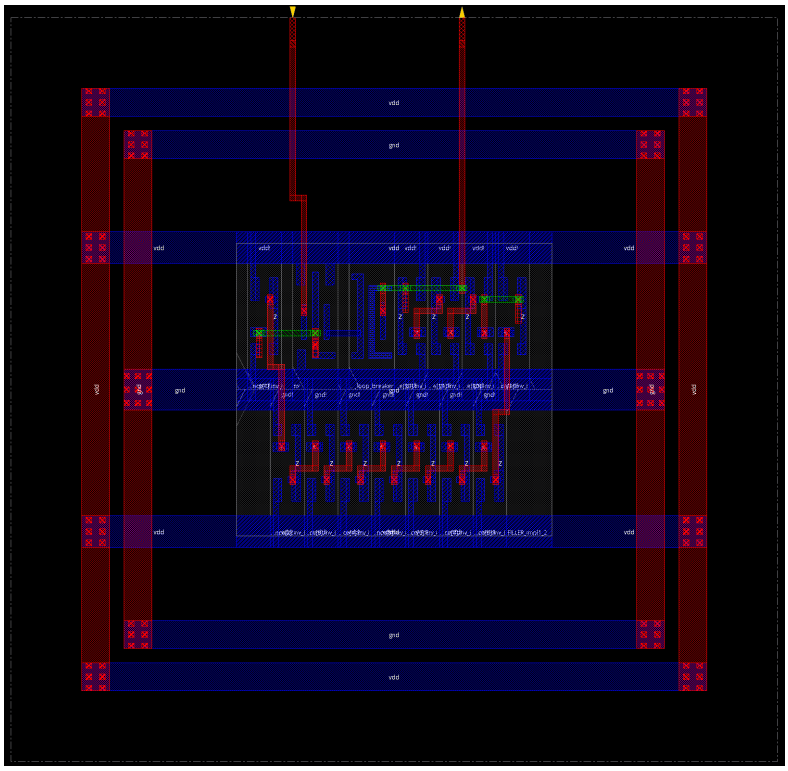
Os 3 primeiros passos do script criam os anéis de alimentação, e definem uma região com duas linhas para as células do circuito. Notar que a linha inferior de células terá as células espelhadas, para a correta conexão à alimentação. A alimentação é implementada na horizontal com metal 1 e na vertical com metal 2. As principais etapas executadas por este script são:

- **Leitura do projeto**: Lê o circuito
- **Floorplanning**: Etapa onde é decidido qual a porcentagem do chip que utilizada para o circuito
- **Power Planning**: Posiciona as linhas de alimentação no chip

Os passos 4 e 5 do script posicionam e roteiam as células do circuito. Notar que há espaços vazios, que ainda devem ser preenchidos. As principais etapas executadas por este script são:

- **Placement**: Posiciona as células no chip
- **Roteamento**: Inserção dos fios de interconexão



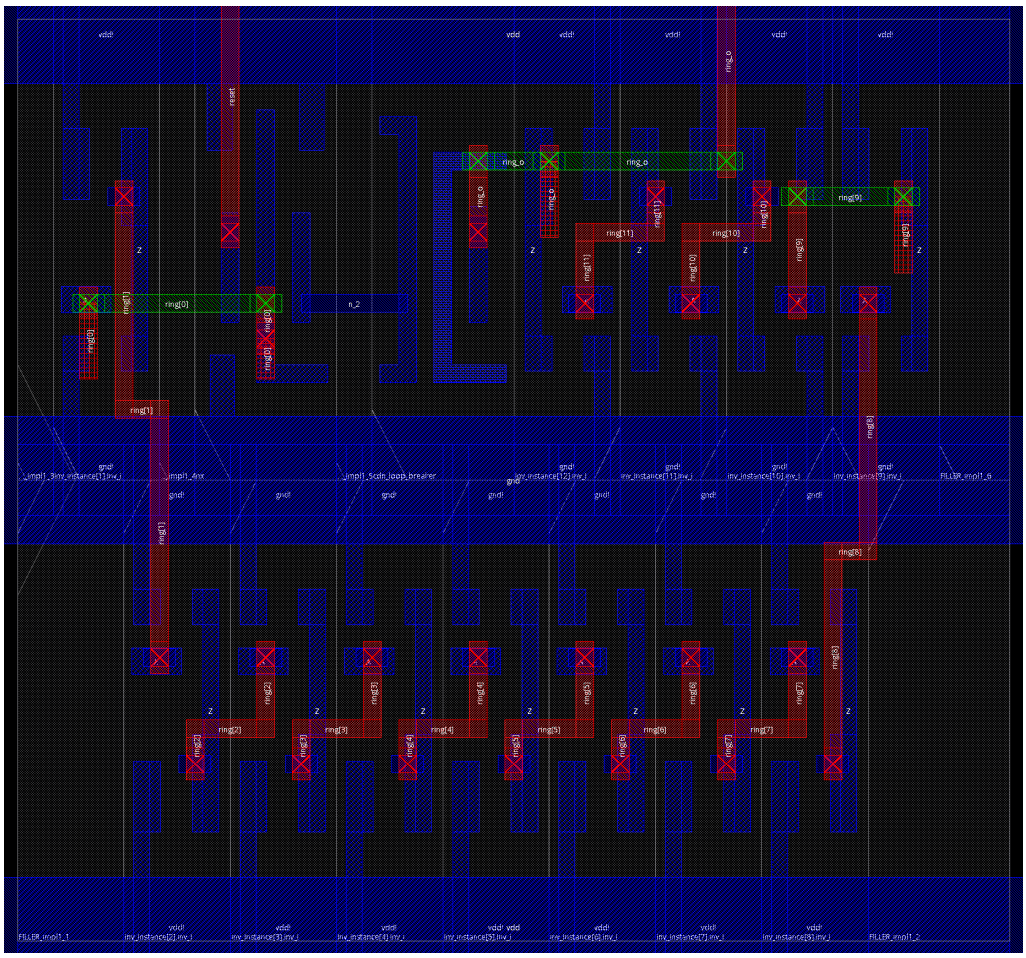


A parte final script completa o layout e gera uma de verificações. As principais etapas executadas por este script são:

- **Inserção das filler-cells**: Inserção de células para preenchimento do espaço do chip
- **DRC**
- **Geração dos relatórios**

Assim, nosso circuito utilizando inversores está finalizado. Selecione a opção para visualizar as linhas de metal internas à célula:

Na tela abaixo pode-se visualizar as conexões entre os inversores, com o metal1 interno das células:



Verificar se não há erros de DRC no circuito: **check_drc**

```
@innovus 8> check_drc
#-check_same_via_cell true              # bool, default=false, user setting
 *** Starting Verify DRC (MEM: 2437.9) ***

  VERIFY DRC ..... Starting Verification
  VERIFY DRC ..... Initializing
  VERIFY DRC ..... Deleting Existing Violations
  VERIFY DRC ..... Creating Sub-Areas
  VERIFY DRC ..... Using new threading
  VERIFY DRC ..... Sub-Area: {0.000 0.000 12.800 13.200} 1 of 1
  VERIFY DRC ..... Sub-Area : 1 complete 0 Viols.

  Verification Complete : 0 Viols.

 *** End Verify DRC (CPU TIME: 0:00:00.0  ELAPSED TIME: 0:00:00.0  MEM: 264.1M) ***.
```

Finalmente, podemos conferir a área do layout gerado:

> report_area

```
@innovus 6> report_area
 Depth  Name      #Inst  Area (um^2)
 ------------------------------------
 0      anel      12     19.76
```

EXIT e FINAL DO LABORATÓRIO DE PROJETO DE CÉLULA