

UNIVERSIDADE FEDERAL DA PARAÍBA  
CENTRO DE INFORMÁTICA  
INTRODUÇÃO A MICROELETRONICA



RELATÓRIO III

THIAGO ALVES DE ARAUJO  
MATRICULA: 2016019787

## 1 – Graal

O Graal (graal layout design) é uma ferramenta utilizada para desenhar o layout das portas lógicas. Esta ferramenta pertence ao Alliance, que nada mais é do que um conjunto de ferramentas e bibliotecas livres que são utilizadas para auxiliar o desenvolvimento de projetos VLSI. Nessa ferramenta, é possível desenhar os circuitos com o tamanho preciso de no mínimo  $1 \lambda$  (*um lambda*). Além disso, nela estão disponíveis diversos tipos de conectores, materiais para os canais de conexão, etc.

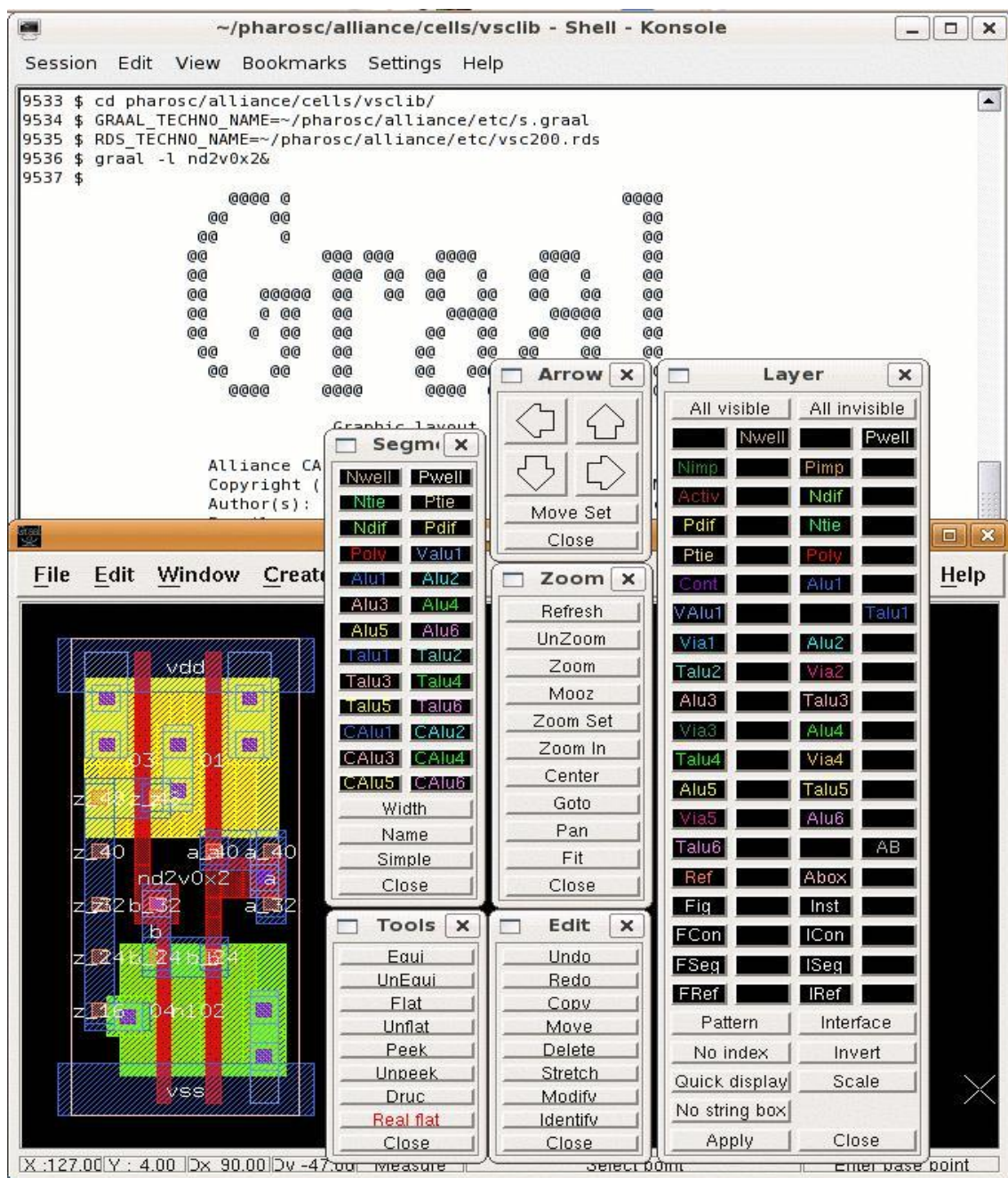
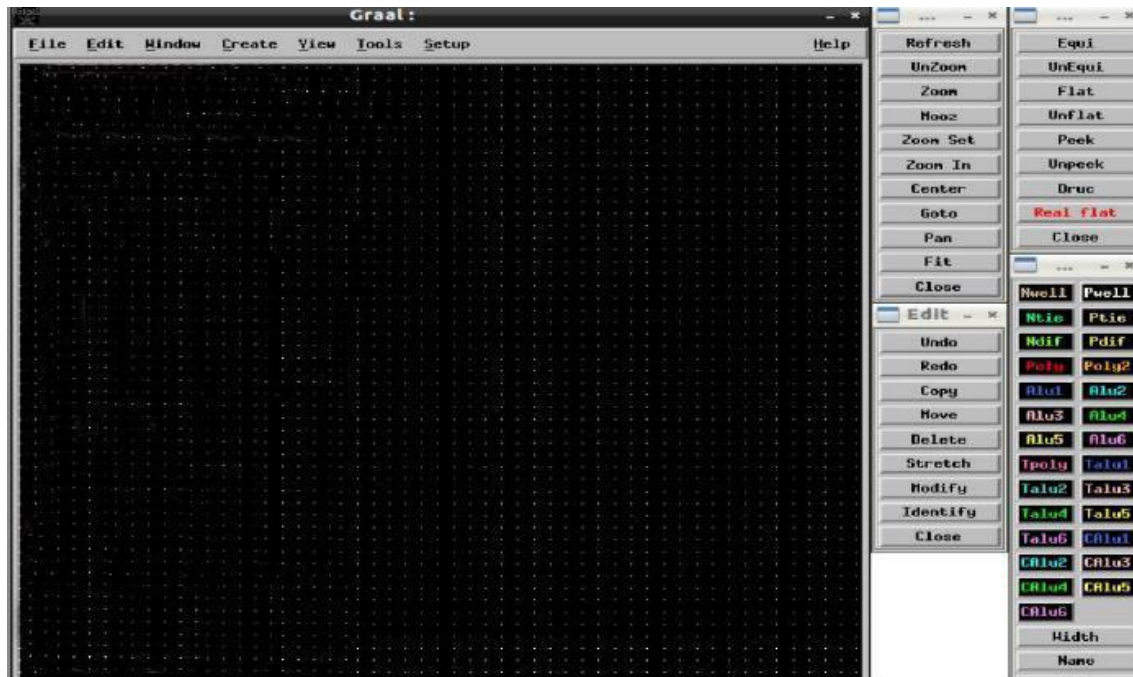


Imagem retirada do site [http://www.vlsitechnology.org/html/linux\\_help3.html](http://www.vlsitechnology.org/html/linux_help3.html)

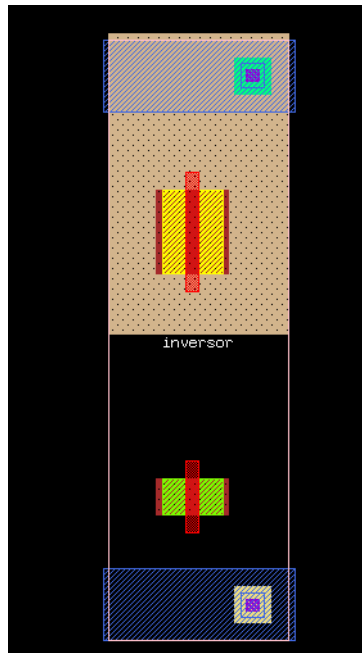
## 2 – Desenvolvendo a Standard Cell

Com o terminal aberto, navegue até a pasta onde deseja salvar o arquivo e execute o Graal. Para a criação da célula standard do circuito inversor, foi utilizado como base o diagrama de palitos do circuito.



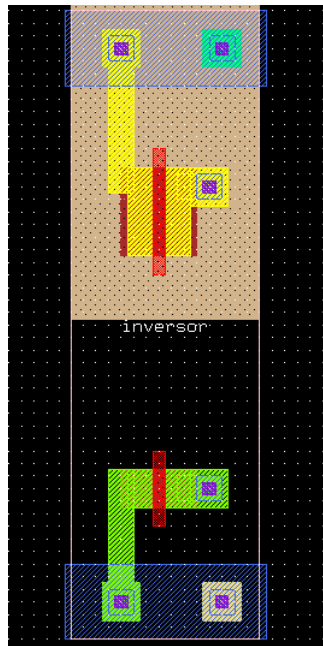
Interface do graal

Primeiro você deve criar um espaço onde os componentes do inversor serão adicionados. Para isso, vá em *create > abutment box* (Ctrl+A) e desenhe uma “caixa” com dimensões  $15\lambda$  de largura por  $50\lambda$  de altura (as coordenadas são mostradas no canto inferior esquerdo). Após isso, vá em *Create > Segment* e crie um seguimento *Nwell* com dimensões  $25\lambda$  por  $15\lambda$  (este segmento deve ser criado na parte superior da “caixa”). Logo após, crie os seguimentos de alumínio (*Create > Segment > Alu1*) colocando-os nas extremidades verticais da “caixa”, com dimensões  $15\lambda$  por  $6\lambda$ . Agora vá em *Create > Via* (Ctrl+V) e posicione as vias *N-Tie* na parte direita do alumínio que está no topo e *P-Tie* na parte direita do alumínio que está na região inferior da “caixa”. Agora vá em *Create > Transistor* (Ctrl + T) e posicione os transistores tipo P em cima do Nwell e o do tipo N entre o Nwell e o seguimento de alumínio da parte inferior. O circuito deve estar da seguinte forma:

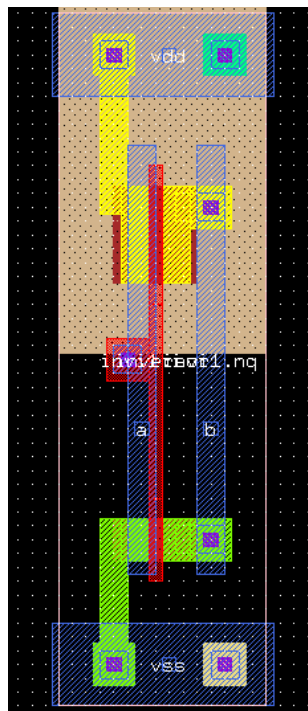


Recomenda-se que a cada passo executado uma verificação de erros seja feita. Para isso, vá em **Tools > Drc** e selecione a área do circuito que foi modificada. Um erro que geralmente acontece é o “Distance Manhattan”. Ele acontece quando dois materiais do mesmo tipo estão muito próximo, desrespeitando a distância mínima de  $3\lambda$ . Para corrigi-lo, basta afastar os componentes ( $3\lambda$  correspondem a três pontos da “grade” pontilhada apresentada na interface do graal).

Agora você deve criar contatos entre o source do transistor P e o seguimento de alumínio superior que será alimentado com VDD, bem como deve conectar o dreno do transistor N ao seguimento de alumínio inferior que será o VSS. Para isso, vá em **Create > Via > Cont Pdiff** e crie dois contatos. O primeiro deve estar dentro do seguimento de alumínio “VDD” e o outro deve estar no dreno do transistor. Para conectar os contatos ao transistor, vá em **Create > Segment > Pdiff** e crie um seguimento entre os dois. Repita o processo para o para o transistor N utilizando **Via > Cont Ndiff** e **Segment > Ndiff**



Para conectar os transistores, crie um seguimento (*Create > Segment > Poly*) e um contato (*Create > Via > Cont Poly*) de polisilício. Agora, crie os conectores com o alumínio. Vá em *Create > Connector > Alu1* e renomeie os conectores com VDD e VSS. Para finalizar isso, crie um seguimento de alumínio (*Create > Segment > Alu1*) conectando os dois transistores e coloque os conectores (*Create > Connector > Alu1*) de entrada e saída. Após isso, basta salvar o arquivo indo em *file > save\_as*. O resultado pode ser visto na imagem abaixo



*Layout do inversor*



### 3 – Cougar

O arquivo criado com Graal possui extensão “.ap”. Para poder simular esse circuito no Spice Opus, é necessário gerar o arquivo “.spi” a partir do mesmo. Para isso é utilizado a ferramenta Cougar que também pertence ao aliante. Com o terminal aberto, vá até a pasta onde o arquivo “.ap” foi salvo e execute o Cougar. Após aberto, execute os seguintes comandos *export MBK\_OUT\_LO=spi* e *cougar -t -ar "nome do arquivo"*. Com isso, um arquivo “.spi” será gerado na pasta selecionada executado.

[illegible]

### Interface do Cougar

## 4 – Simulação

Para a simulação da análise estática, é necessário criar um arquivo “.cir” para descrever o circuito. Para a análise desejada, segue o código abaixo:

```
.include inversor.spi

* a b vdd vss
X1 20 40 10 30 inversor
V3 20 30 0V
V1 10 30 1.8V DC
V2 30 0 0V DC

.model tp pmos level=54
.model tn nmos level=54

.dc V3 0 1.8 .001
.end
```

Arquivo inversor.cir

Para analisar o comportamento, basta exibir a pulso de entrada e a saída.

```
spiceOpus (c) 1 -> inversor_curva.cir
spiceOpus (c) 2 -> run
spiceOpus (c) 3 -> plot v(10) v(40)
spiceOpus (c) 4 -> |
```

#### *Comandos para exibir as curvas*

O correto funcionamento de um inversor se dá quando, ao se aplicar 0V em sua entrada a saída vá para 1,8V. Para o correto funcionamento, o inversor deve inverter aproximadamente no meio do pulso que se encontra entre 0,899 e 0,901 na análise estática. Exibindo o sinal de entrada e saída do inversor criado, percebe-se que seu comportamento está como o esperado. Entretanto, a inversão da saída ocorre um pouco antes do sinal de entrada atingir 0.9v



Para se aproximar de um inversor ideal, deve-se equilibrar as resistências dos transistores de modo que o ponto 0.9V seja alcançado. Essa distância do ponto ideal deve-se ao fato de que as resistências dos transistores são diferentes, ou seja, a condutibilidade (ou mobilidade) de um transistor é diferente da outra. Para isso é necessário modificar a largura do transistor P e do transistor N de modo que o equilíbrio correto entre as resistências seja encontrado. Após algumas tentativas foi encontrado o valor desejado.



*Algumas curvas das diversas tentativas. Curva azul corta o sinal em 0.9003V*

Após conseguir um inversor com um comportamento bem próximo do ideal, analise seu comportamento dinâmico. Para isso, crie um novo arquivo para simulação, onde desta vez o pulso de entrada será uma onda quadrada com variação de 1ns. Abaixo encontra-se o código:

```
.include inversor.spi

.model tp pmos level = 54
.model tn nmos level = 54

x1 10 40 20 30 inversor

V1 10 30 pulse(0 1.8V 1ns 1ps 1ps 1ns 2ns)
V2 20 30 1.8V
V3 30 0 0V

.tran 0.0001 3ns
.end
```





*Curvas dos sinais de entrada e saída*

#### 4.1 – Tempo de Subida

Como a tensão máxima é de 1.8V, o tempo de subida será o tempo gasto para o sinal ir de 0,18V a 1,62V



10% (0,18V) = 2,08745xe-9



90% (1,62V) = 2,20879xe-9

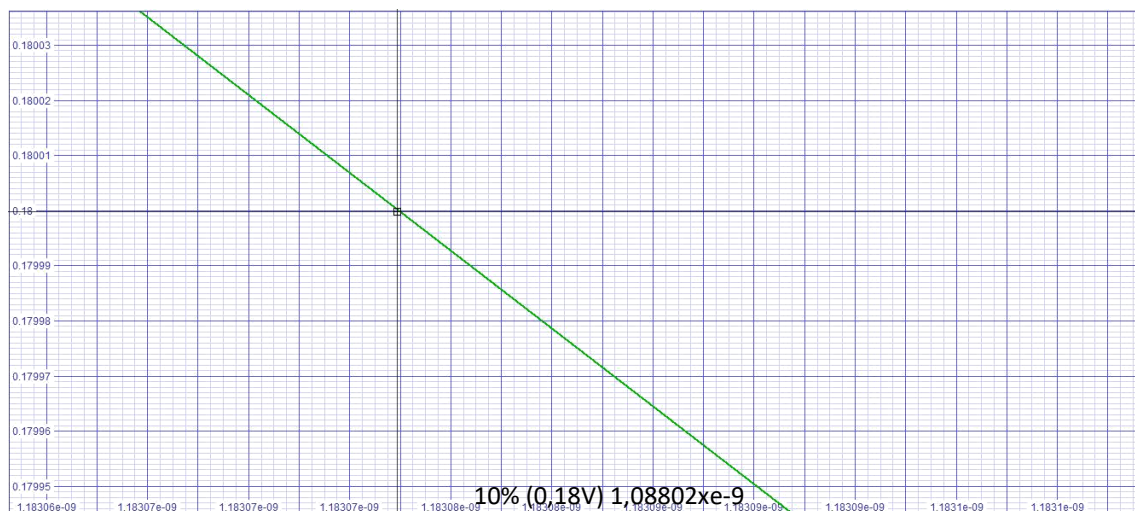
O tempo de subida foi de aproximadamente  $12,134 \times 10^{-11} \text{s}$

## 4.2 – Tempo de Descida

O tempo de descida será o tempo gasto para ir de 1,62V a 0,18V. Segue a análise abaixo:



$$90\% (1,62V) = 1,18307 \times 10^{-9}$$



$$10\% (0,18V) = 1,08802 \times 10^{-9}$$

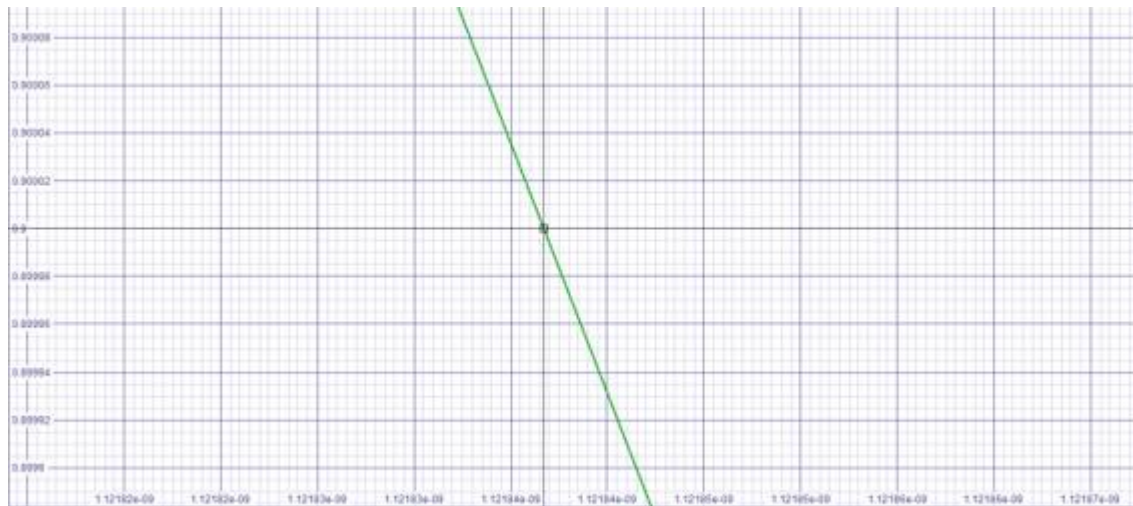
O tempo de subida foi de aproximadamente  $9,505 \times 10^{-11}$

## 4.3 – Tempo de HL e LH

Como a tensão máxima é de 1.8V, o tempo gasto para ir de 1,8V a 0,9V será o HL e 0,0V a 0,9V será o LH.

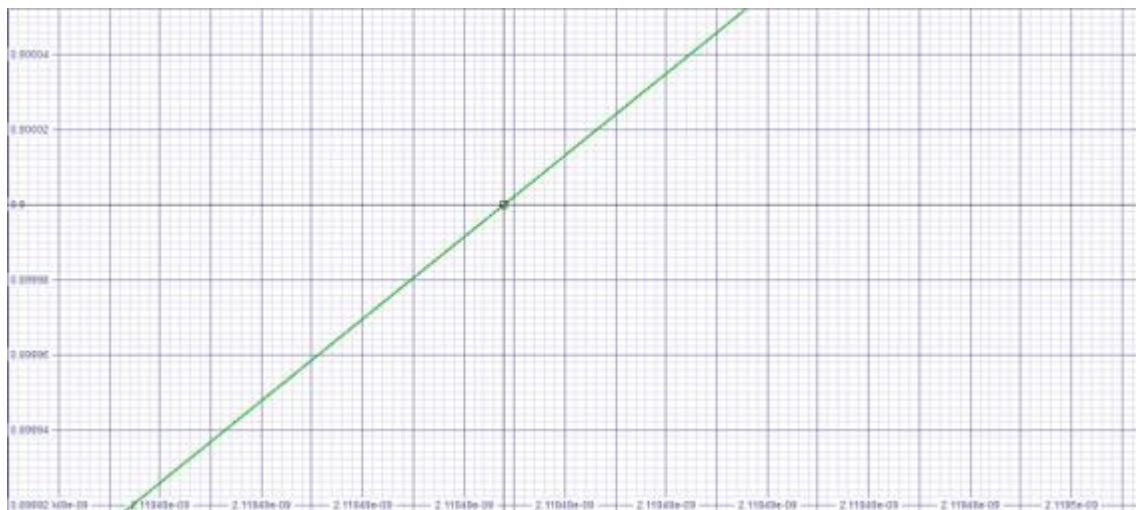






1,12184xe-9

Como o pulso iniciou em 1ns, o tempo gasto para ir de 1,8V até 0,9V foi de 12,184xe-11s



2,11949xe-9

Como o pulso iniciou em 1ns, o tempo gasto para ir de 0,0V até 0,9V foi de 11,949xe-11s

#### 4.4 – Constante de tempo

Como o inversor possui dois transistores (P e N) o circuito vai possuir duas constantes de tempo (uma para cada transistor). Idealmente essas constantes são iguais, porém, devido a alguns efeitos indesejados e muito difíceis de serem corrigidos, as constantes podem apresentar pequenas diferenças em seus valores.

A constante de tempo é o tempo necessário para o capacitor carregar até 63,2% de sua capacidade. Assim,  $(0,632) \cdot (1,8V) = 1,1376V$





Neste gráfico, a onda da direita representa o transistor P e a onda da esquerda representa o transistor N. Para calcular a constante de tempo do transistor P, basta calcular o tempo gasto para a onda ir de 0V a 1,13V (note que em 0V a onda não está cortando exatamente o ponto 2ns, logo, essa diferença deve ser levada em conta). Para calcular a constante de tempo do transistor N, o cálculo é exatamente igual.



2,13478xe-9

Logo, a constante de tempo do transistor P é de aproximadamente 5,578xe-11



1,13467xe-9

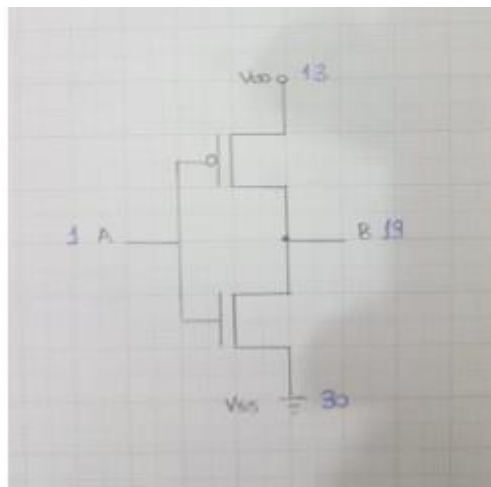
Logo, a constante de tempo do transistor N é de aproximadamente 5,584xe-11

#### 4.5 – Capacitâncias parasitas no canal de saída

Anteriormente quando foi montado o circuito no graal, foi definido nomes para a entrada e a saída do inversor. Quando o arquivo “.spi” do circuito foi criado, todas as características dele foram descritas de forma que é possível saber quais componentes estão conectados entre cada nó.

```
.subckt inversor 1 19 13 30
* NET 1 = a
* NET 13 = vdd
* NET 19 = b
* NET 30 = vss
-----
```

*Trecho do código do Inversor.spi*



*Representação do código acima*

Esta informação vai ser necessária para identificar as capacitâncias parasitas que se encontram entre a saída(b) e o ground. Abrindo o arquivo “inversor.spi”, encontre os valores das capacitâncias parasitas.

```
C2_32 24 30 6.3e-16
R2_4 19 23 0.8
➡ C2_41 19 30 2.24e-15
C2_42 23 30 2.24e-15
R2_5 20 19 0.4
C2_51 20 30 1.12e-15
➡ C2_52 19 30 1.12e-15
R2_6 21 20 0.1
C2_61 21 30 3.5e-16
```

*Trecho do código do arquivo “inversor.spi”*

Pode-se perceber que existem duas capacitâncias parasitas no circuito, umas com 2,24xe-15F e outra de 1,12xe-15. Também é fácil notar que elas estão conectadas em paralelo, logo, o valor total da capacitância parasita é de 3,36xe-15.

#### 4.6 – Resistências do canal

Com os valores da capacitância e constante de tempo de cada um dos capacitores é possível calcular a resistência interna de cada um dos transistores. Abaixo estão os cálculos realizados.

TRANSISTOR P

$$\tau_p = R_p C \quad \therefore R_p = \frac{\tau_p}{C} = \frac{5,578 \times 10^{-11}}{3,36 \times 10^{-15}} = 16,6 \text{ K}\Omega$$

TRANSISTOR N

$$\tau_n = R_n C \quad \therefore R_n = \frac{\tau_n}{C} = \frac{5,584 \times 10^{-11}}{3,36 \times 10^{-15}} = 16,6 \text{ K}\Omega$$

*Calculo para as resistências internas*

#### 4.7 – Capacitâncias parasitas no canal de entrada

Assim como a análise feita para encontrar as capacitâncias parasitas de saída é possível também encontrar as capacitâncias de entrada. Pelo diagrama mostrado anteriormente, podemos ver que as capacitâncias estarão localizadas entre os nós 1 e 30

```

C4_21 4 30 2.17e-15
C4_22 7 30 2.17e-15
R4_3 1 4 0.2
➔ C4_31 1 30 7e-16
C4_32 4 30 7e-16
R4_4 2 1 0.5
C4_41 2 30 1.47e-15
➔ C4_42 1 30 1.47e-15
R4_5 5 6 450

```

*Trecho do código do arquivo "inversor.spi"*