

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
INTRODUÇÃO A MICROELETRONICA

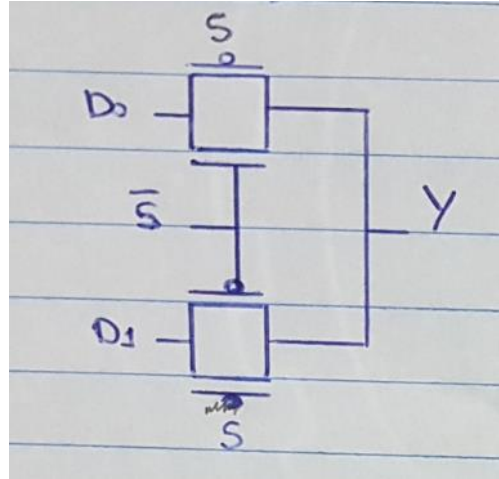


RELATÓRIO VII

THIAGO ALVES DE ARAUJO
MATRICULA: 2016019787

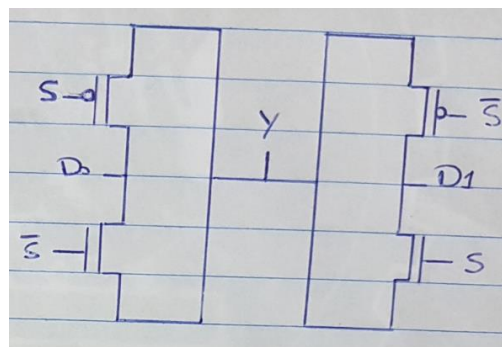
1 – Multiplexador 2:1

Inicialmente para criarmos um multiplexador com duas entradas de dados e uma entrada de controle, precisamos construir o seguinte circuito:



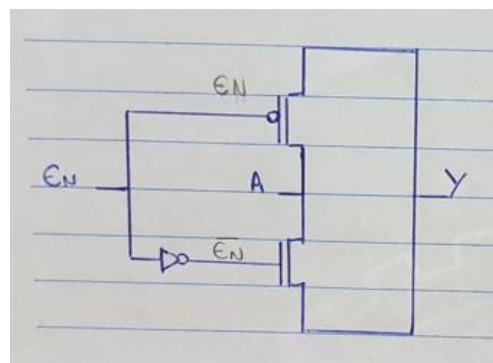
Esquema do mux 2:1

Redesenhando o circuito, obtemos o seguinte layout:

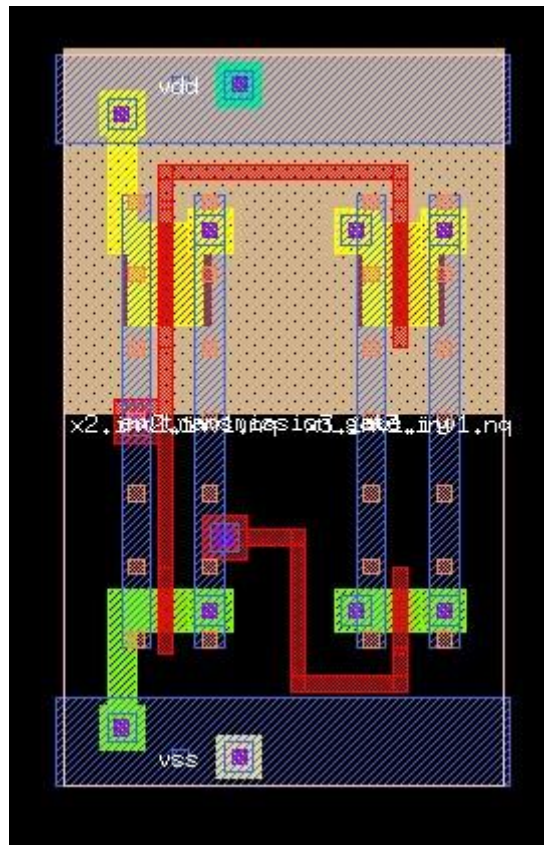


Esquema do Mux 2:1

O circuito mostrado a cima já foi criado em análises anteriores quando construímos um *Transistor com porta de transmissão*. O layout deste circuito apresentado na figura a seguir:



“Transmission gate”



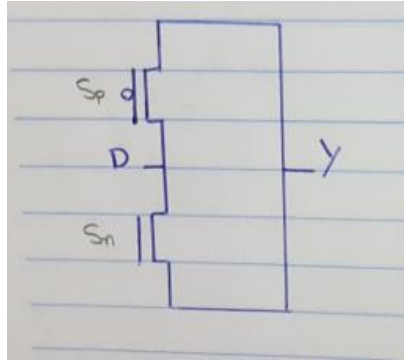
Layout do circuito exibido no graal

Como podemos observar, o circuito criado anteriormente é bem parecido com o circuito do multiplexador. Para que o layout mostrado a cima possa ser utilizado para criar o mux, o inversor de entrada deve ser removido, tendo em vista que as entradas de controle (S) do multiplexador não recebem os mesmos valores

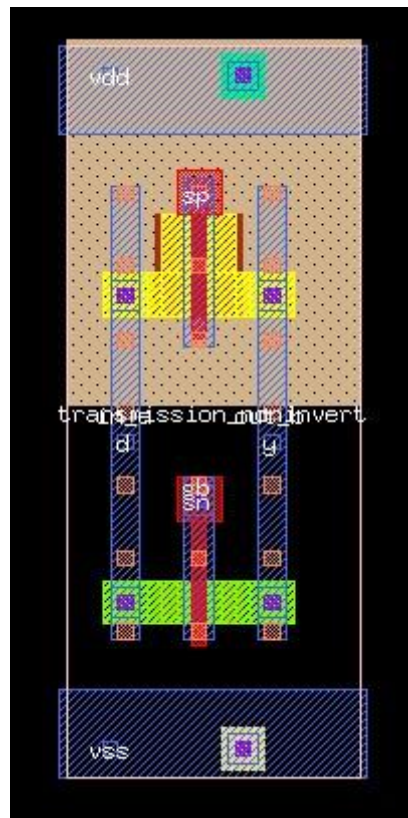
$$Sp(D0) = S \text{ e } Sp(D1) = \sim S$$

$$Sn(D0) = \sim S \text{ e } Sn(D1) = S$$

Removendo o inversor de entrada, podemos utilizar a instancia “transmission gate sem inversor de entrada” para criar o mux.

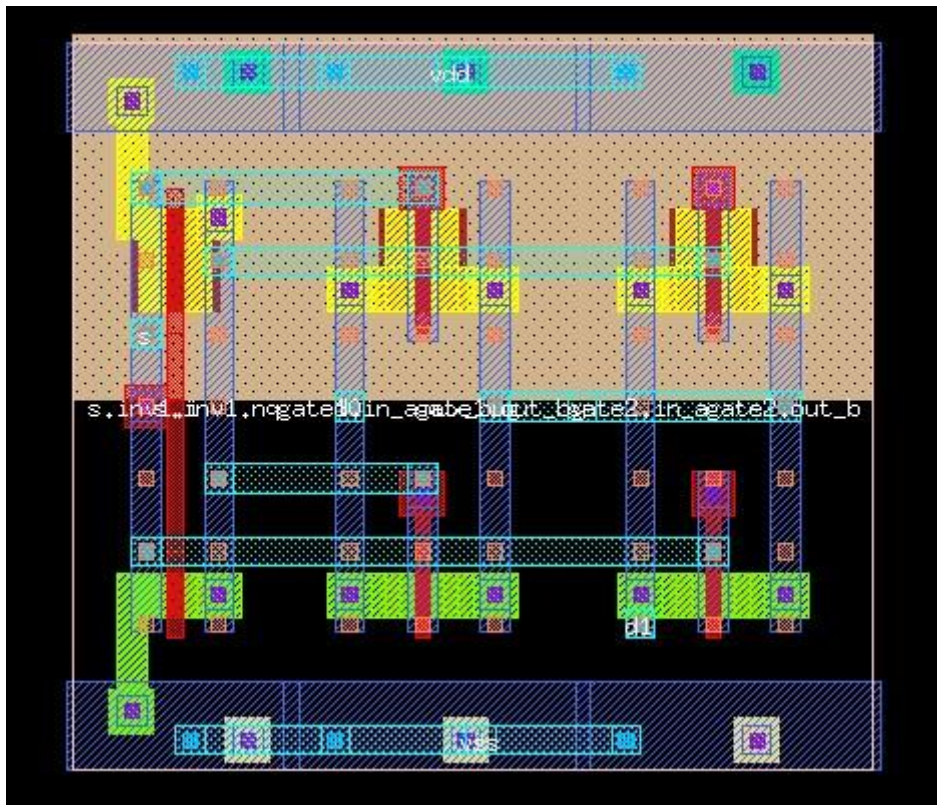


“transmission gate sem inversor de entrada”

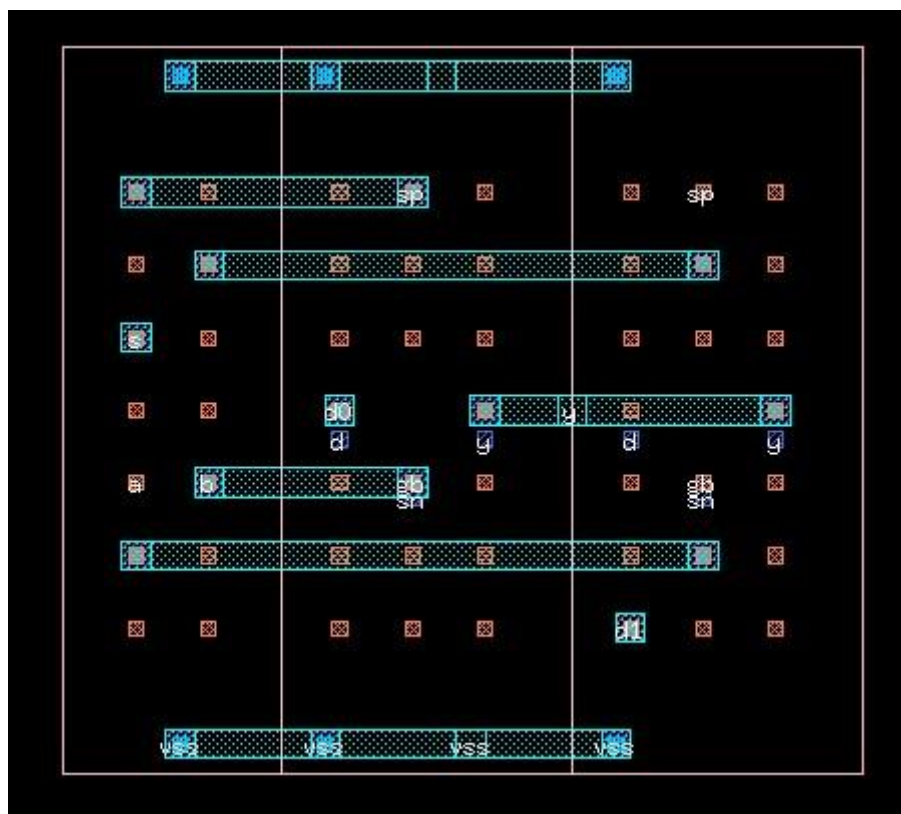


“transmission gate sem inversor de entrada”

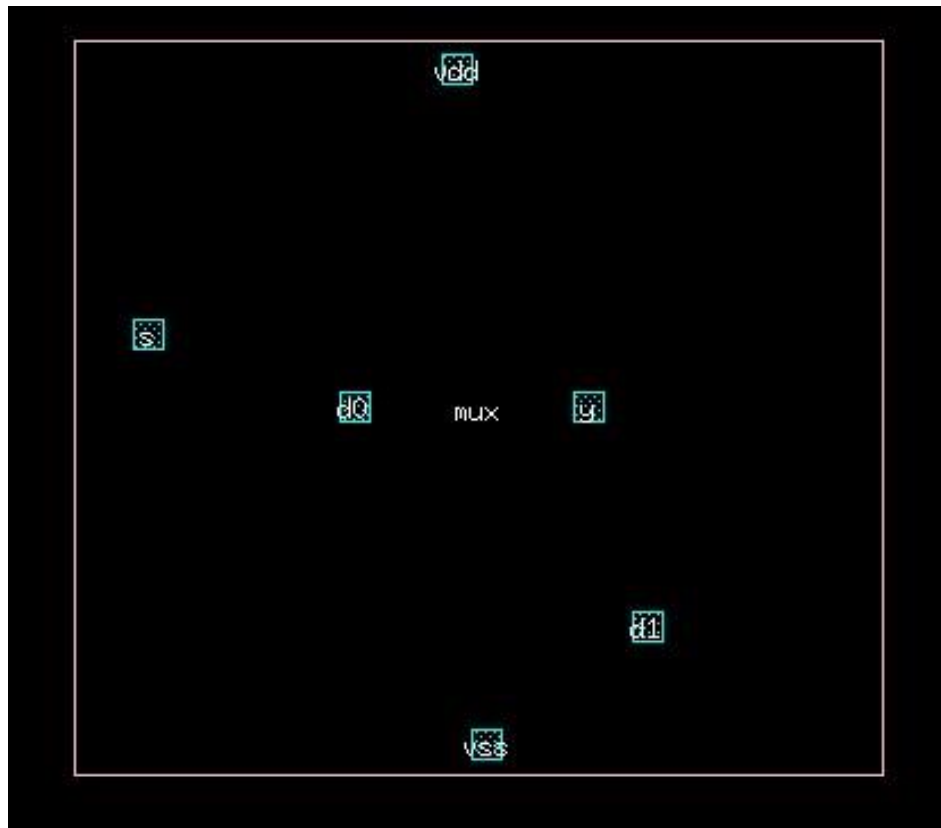
Com o circuito em mãos, instanciamos uma célula inversora e duas células “gate sem inversor e de entrada”. Como os circuitos instanciados possuem suas conexões em Alu1, subimos o nível das conexões para o Alu2 e fazemos as conexões necessárias. Abaixo esta o circuito final do mux 2:1



“Multiplexador 2:1”



“Células conectadas através do Alu2”



“Célula fechada”

1.1 – Simulação

O multiplexador de duas entradas possui o seguinte comportamento:

- Quando $S=0$, independente das entradas, o valor da saída vai ser o valor que está entrando no canal D0.
- Quando $S=1$, independente das entradas, o valor da saída vai ser o valor que está entrando no canal D1.

Logo, a tabela verdade assume os seguintes valores:

S	D1	D0	Y
0	X	0	D0
0	X	1	D0
1	0	X	D1
1	1	X	D1

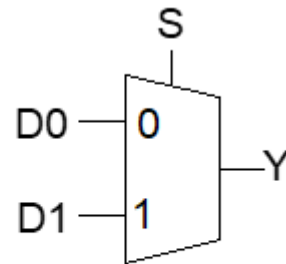


Tabela verdade mux 2:1

Para verificarmos se o circuito criado está funcionando perfeitamente, aplicamos dois sinais de entrada e variamos o controle (S). O circuito utilizado para a simulação está apresentado abaixo:

```
.include mux.spi

*  d0 d1  s vdd vss  y
X1 20 21 10 30 40 50 mux

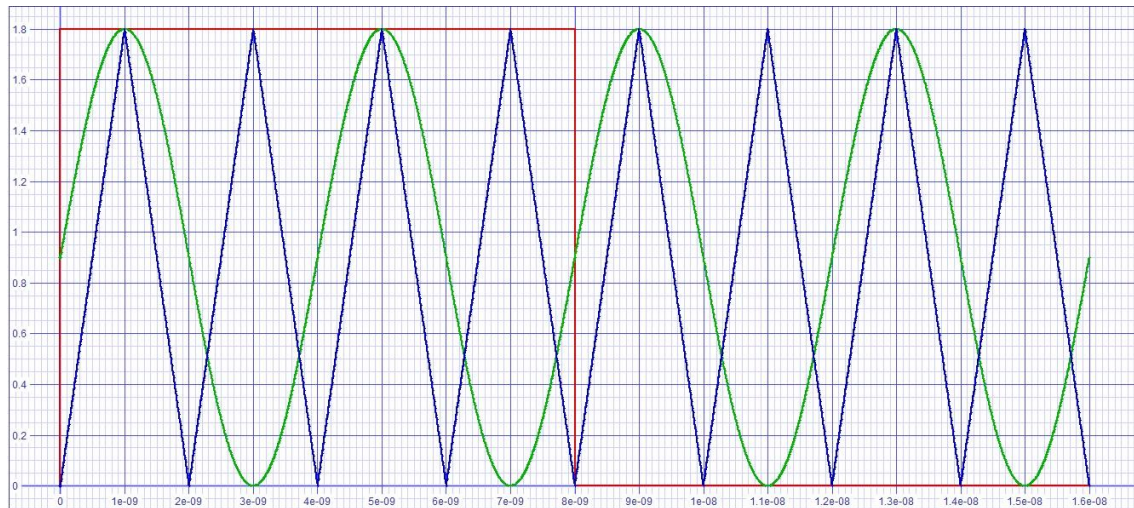
V5 10 40 pulse(0 1.8 0 1p 1p 8n 16n)
V4 30 40 1.8V
V3 40 0 0V
V2 21 40 pulse(0 1.8 0 1n 1n 1p 2n)
V1 20 40 sine(0.9 0.9 0.25G 0 0 0)

.model tp pmos level=54
.model tn nmos level=54
.tran 0.001ns 16ns

.end
```

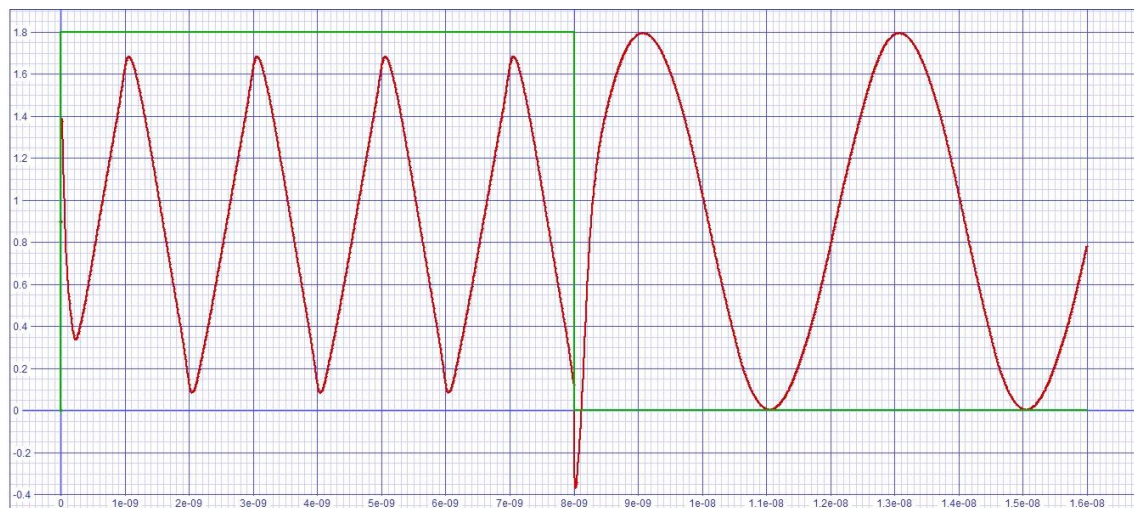
“Arquivo de simulação”

Na entrada D0 aplicamos uma senoide e na entrada D1 aplicamos uma onda triangular. A entrada de controle S recebe um pulso que varia de “1” para “0”. Os sinais de entrada são exibidos abaixo:



Verde - D0(v20), Azul - D1(v21), vermelho - S(v10)

Com isso, podemos observar o comportamento da saída do mux:

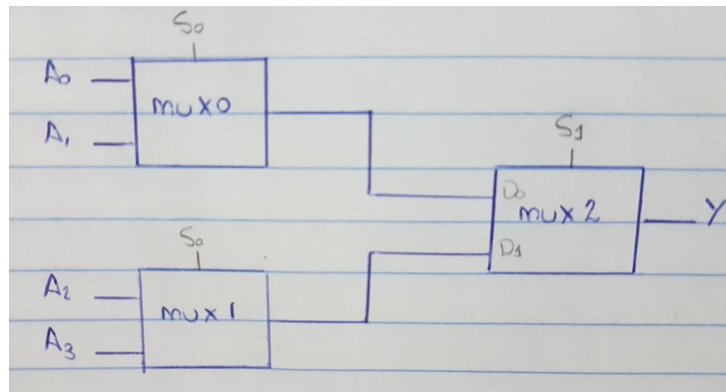


Verde - S(v10), vermelho - Y(v50)

Com isso, podemos observar que, quando o sinal de controle está em “1”, a saída Y assume o valor da entrada D1(onda triangular). Quando o controle está em “0”, a saída assume o valor que está em D0(onda senoidal).

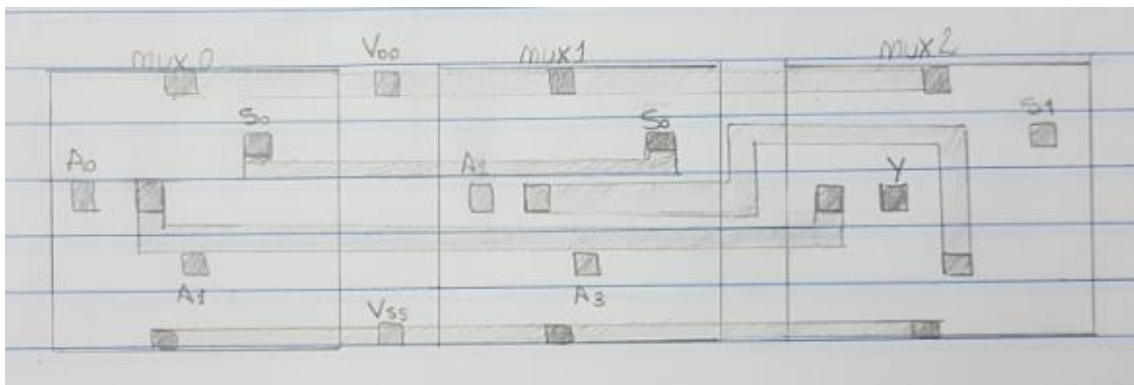
2 – Multiplexador 4:2

Agora, com um multiplexador em mãos, podemos ligar vários deles em “cascata” para analisarmos o comportamento da saída como mostra a esquema a seguir:

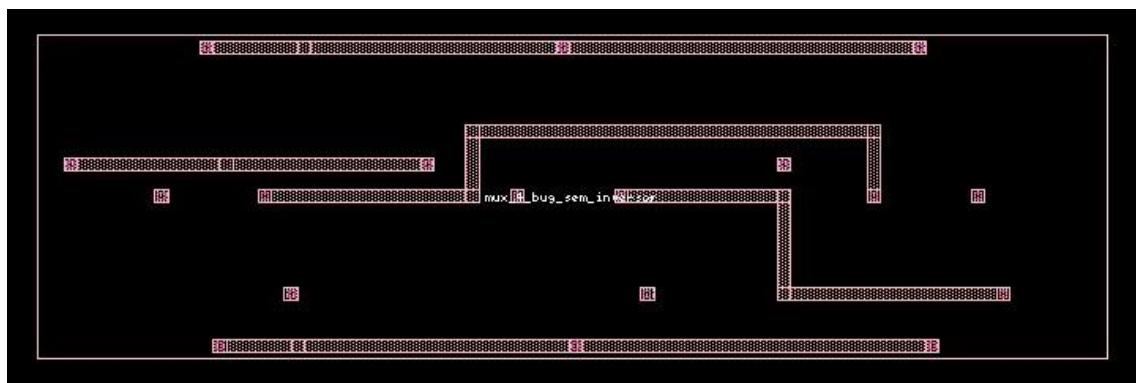


"Mux 4"

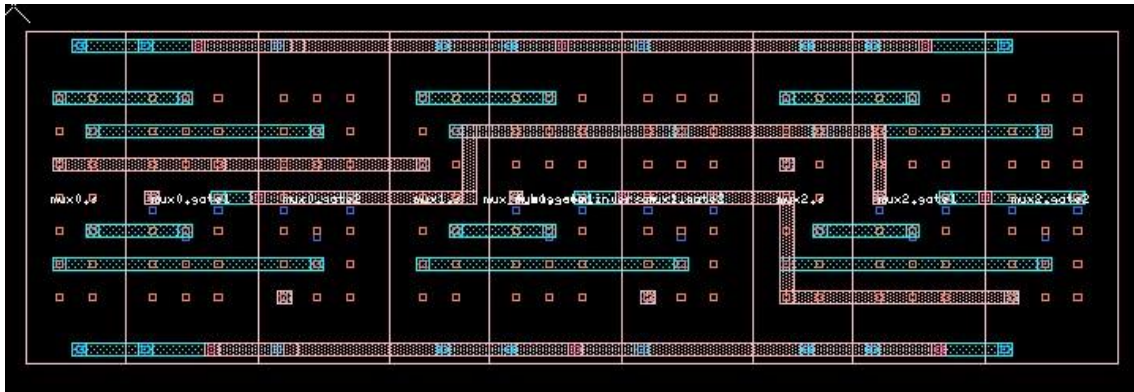
Para criar o circuito acima, instanciamos três multiplexadores criados anteriormente e fazemos as devidas conexões. Como as conexões entre os componentes internos do mux foram feitas em Alu2, precisamos subir de nível novamente para conectarmos os três multiplexadores em uma única célula.



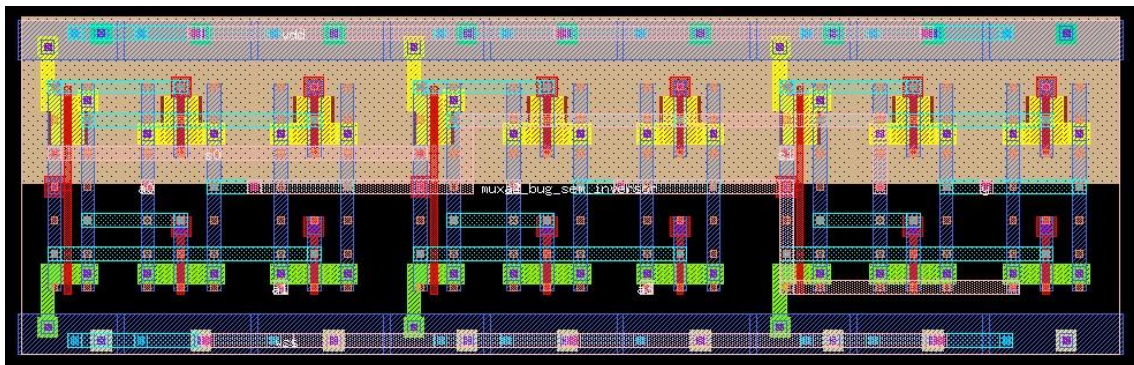
"Esboço do circuito"



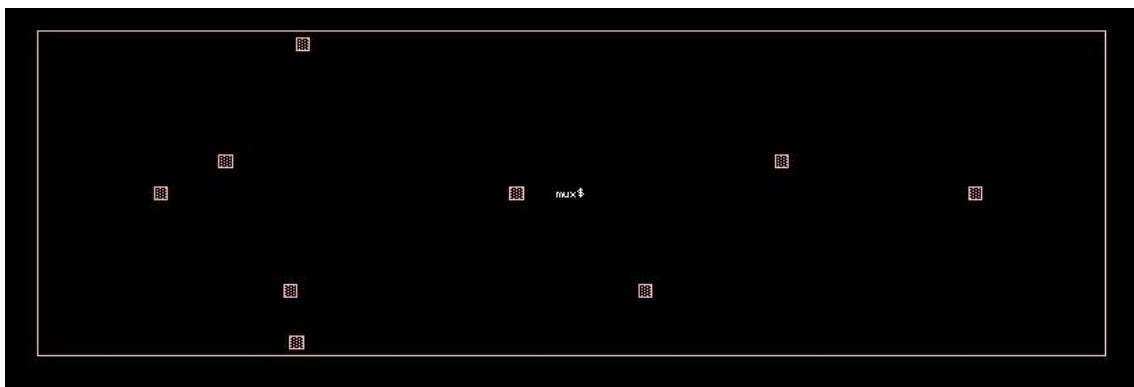
"Célula do Mux4 vista do nível 3"



“Célula do Mux4 vista do nível 2”



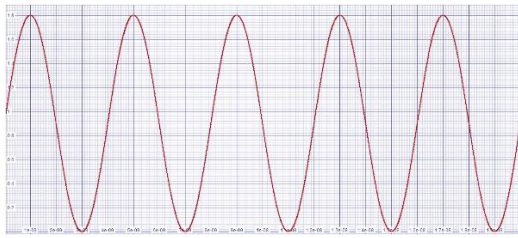
“Célula do Mux4 vista do nível 1”



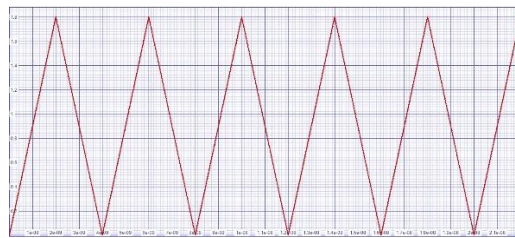
“Célula do Mux4”

2.1 – Simulação

Agora, para simular o circuito criado, aplicamos nas estradas A0, A1, A2 e A3 sinais diferentes. Os controles S0 e S1 são pulsos gerados de tal forma que S1 tenha o dobro do período de S0. Os sinais de entrada foram os seguintes:



A0



A1



A2



A3

O arquivo de simulação ficou da seguinte forma:

```
.include mux_4.spi

* a0 a1 a2 a3 s s1 vdd vss y
xl 20 21 22 23 10 11 30 40 50 mux_4

*A0 input
V1 20 40 sine(0.9 0.9 0.25G 0 0 0)

*A1 input
V2 21 40 pulse(0 1.8V 0ns 2ns 2ns 1ps 4ns)

*A2 input
V3 22 40 pulse(0 1.8V 0ns 4ns 1ps 1ps 4ns)

*A3 input
V4 23 40 pulse(0 1.8V 0ns 1ps 4ns 1ps 4ns)

*S0 input
v5 10 40 pulse(0 1.8V 0ns 1ps 1ps 10ns 20ns)

*S1 input
v6 11 40 pulse(0 1.8V 0ns 1ps 1ps 20ns 40ns)

v7 30 40 1.8v
v8 40 0 0v

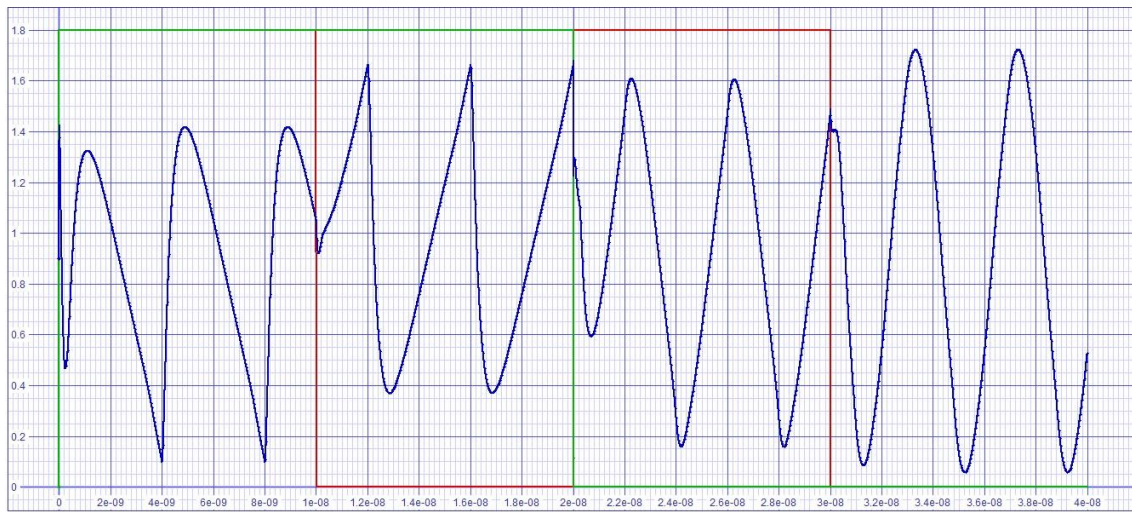
.model tp pmos level=54
.model tn nmos level=54

.tran 1ps 40ns

.end
```

“mux4.cir”

Com isso, o seguinte sinal foi observado na saída:



Vermelho – S0, verde – S1, azul - Y