

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
INTRODUÇÃO À MICROELETRÔNICA



Relatório VIII

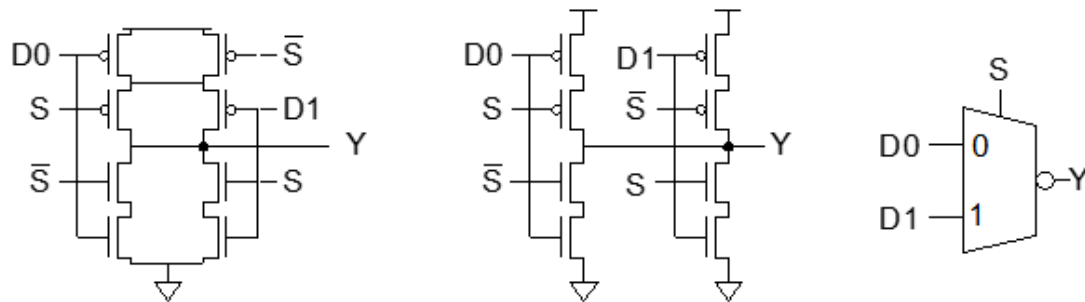
THIAGO ALVES DE ARAUJO

MATRICULA: 2016019789

João Pessoa
2018

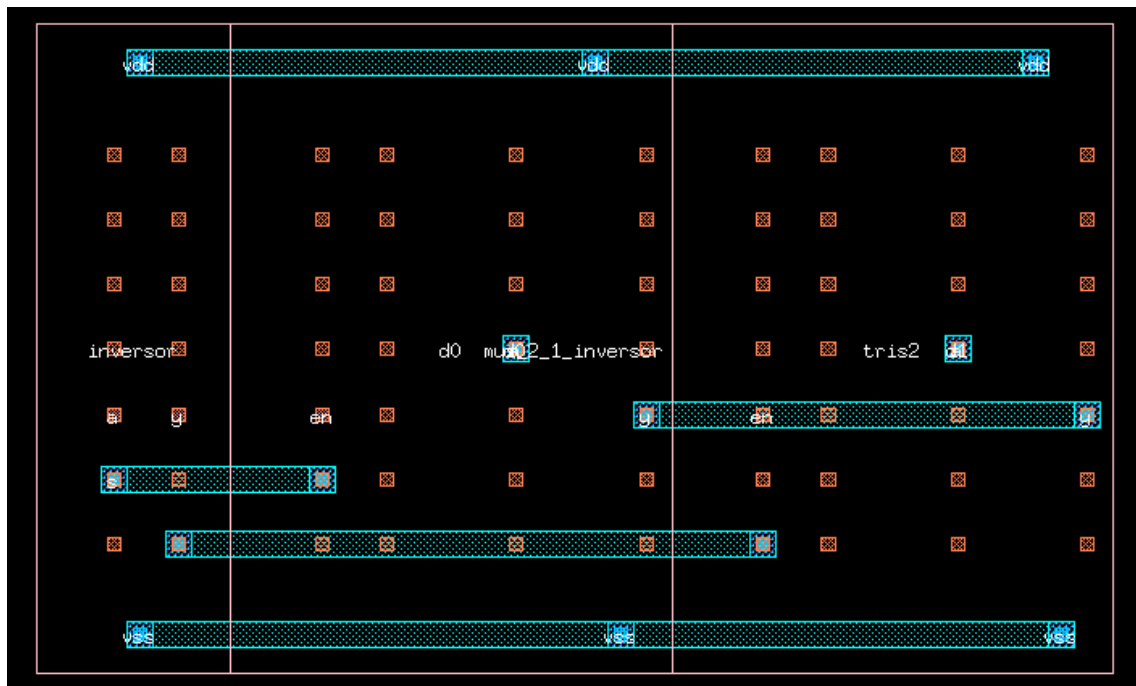
1 – Mux 2:1 Inversor

Para criarmos um multiplexador inversor de duas entradas, precisamos de um inversor e dois tristate inversores como mostra a figura abaixo:



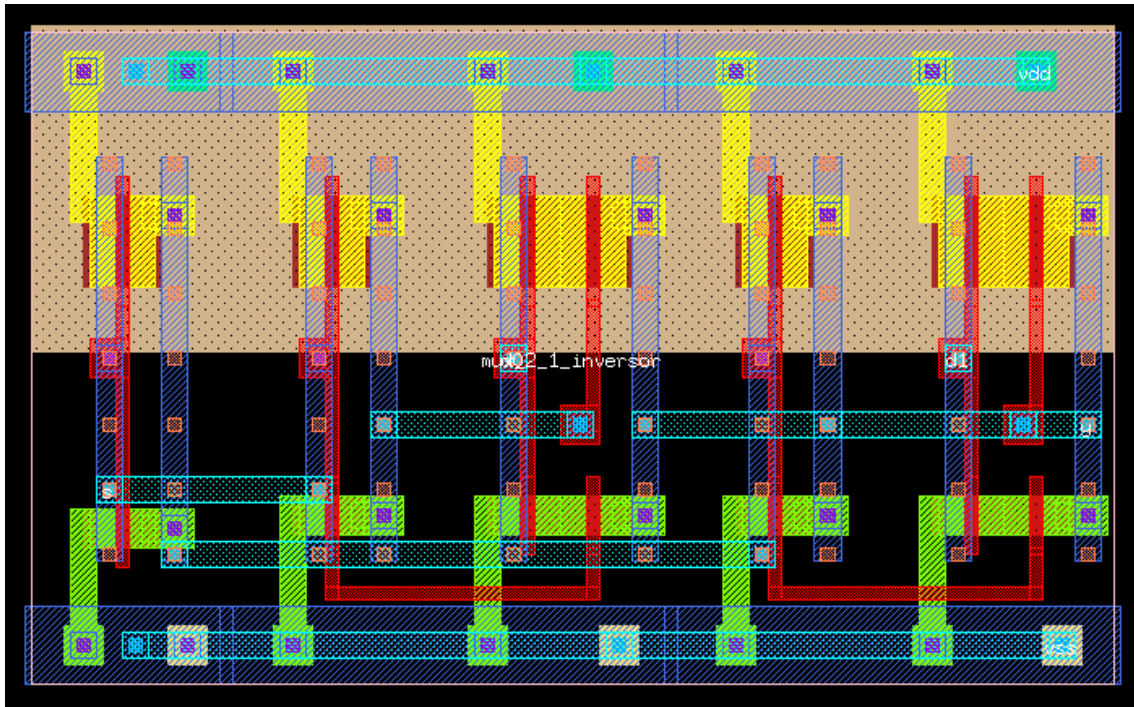
Mux inversor de duas entradas

Instanciando o inversor e os dois tristate (ambos criados anteriormente) e fazendo as devidas conexões apresentadas acima, obtemos o seguinte resultado:



Mux 2:1 inversor

E importante destacar que o tristate inversor possui uma conexão em alu2 internamente, porém mesmo com ela foi possível fazer as conexões das células também em alu2 com as referencias que não estavam sendo utilizadas.



Mux 2:1 inversor (flat)

Para verificarmos as conexões internas, podemos exibir o esquemático do circuito com os comandos “export MBK_IN_LO=spi” e “xsch -l nome”

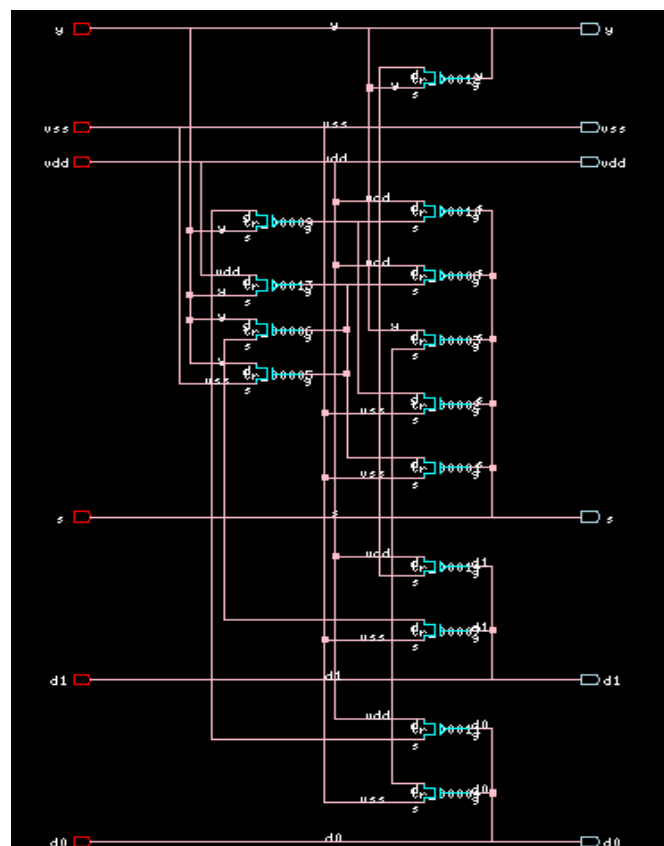


Diagrama de transistores do mux 2:1 inversor

1.1 – Simulação

Para simular o circuito, aplicamos sinais digitais na entrada e observamos o comportamento da saída em relação a entrada e a “chave” seletora.

Abaixo está o arquivo de simulação utilizado:

```
.include mux_2_1_inversor.spi

* INTERF d0 d1 s vdd vss y
X1      20 21 10 30 40 50 mux_2_1_inversor

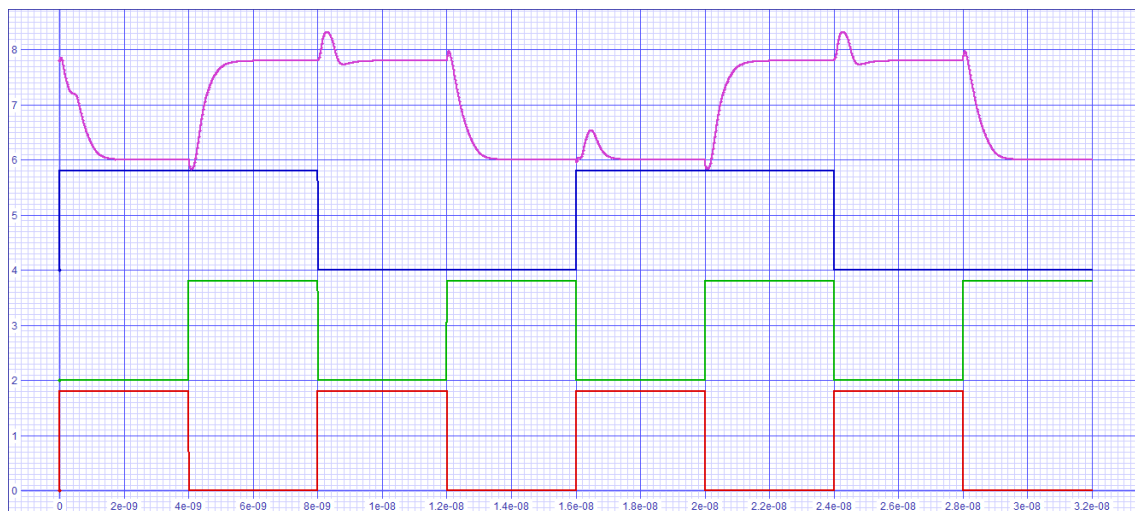
V5 10 40 pulse(0 1.8 0 1p 1p 8n 16n)
V4 30 40 1.8V
V3 40 0 0V
V2 21 40 pulse(0 1.8 4ns 1ps 1ps 4ns 8ns)
V1 20 40 pulse(0 1.8 0ns 1ps 1ps 4ns 8ns)

.model tp pmos level=54
.model tn nmos level=54
.tran 0.001ns 32ns

.end
```

Arquivo de simulação de mux 2:1 inversor

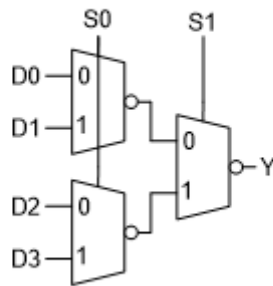
Com isso, podemos observar o seguinte sinal de saída:



D0(vermelho) D1(verde) S(azul) e Y(roxo)

2 – MUX 4:1 INVERSOR

Para criarmos um multiplexador de quatro entradas, instanciamos três multiplexadores inversores de duas entradas como mostra o circuito abaixo:



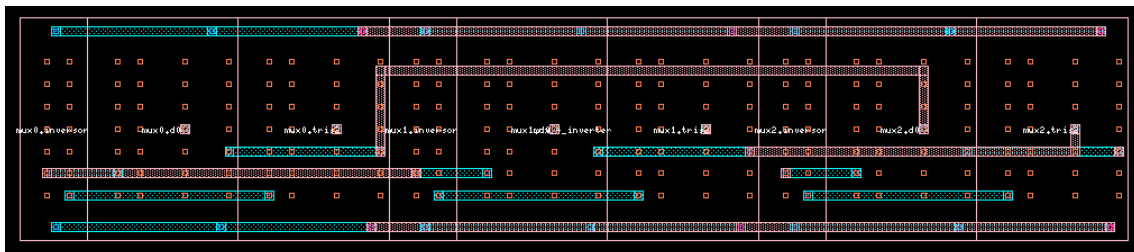
Mux 4:1 inversor

Instanciando os três multiplexadores 2:1 (criados anteriormente) e fazendo as devidas conexões apresentadas acima, obtemos o seguinte resultado:



Mux 4:1

As conexões entre os multiplexadores 2:1 foram feitas em alu3 pois o mux possui conectores em alu2 internamente como mostra a figura abaixo:



Mux 4:1

O circuito interno pode ser visto utilizando os seguintes comandos:

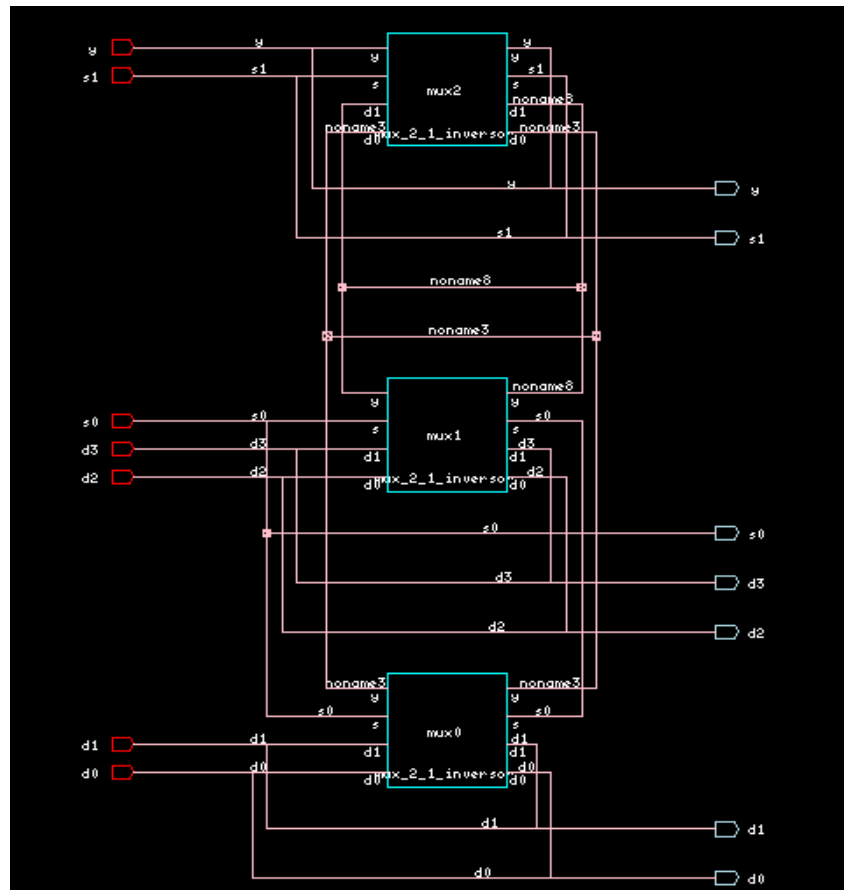
```
export MBK_OUT_LO=vst
```

```
cougar nome
```

```
export MBK_IN_LO=vst
```

```
xsch -l nome
```

O resultado disso pode ser visto abaixo:



MUX 4:1

2.1 – Simulação

Para simular o circuito, aplicamos sinais digitais na entrada e observamos o comportamento da saída em relação as entradas (D0, D1, D2 e D3) e as “chaves” seletoras (S0 e S1). Abaixo está o arquivo de simulação utilizado:

```
.include mux4_inverter.spi

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

* d0 d1 d2 d3 s0 s1 vdd vss y

X1 20 21 22 23 10 15 30 40 50 mux4_inverter

V1 10 40 pulse(0 1.8V 0ns 1ps 1ps 16ns 32ns)
V2 15 40 pulse(0 1.8V 0ns 1ps 1ps 32ns 64ns)

V3 40 0 0V

V4 30 40 1.8V

V5 20 40 pulse(0 1.8V 0ns 1ps 1ps 4ns 8ns)
V6 21 40 pulse(0 1.8V 4ns 1ps 1ps 4ns 8ns)
V7 22 40 pulse(0 1.8V 0ns 1ps 1ps 8ns 16ns)
V8 23 40 pulse(0 1.8V 8ns 1ps 1ps 8ns 16ns)

.tran 8ps 65ns

.end
```

Com isso, podemos observar o seguinte comportamento na saída:



D0(vermelho), D1(verde), D2(azul), D3(rosa), S0(preto), S1(marrom) e Y(roxo)

```
Spiceopus (c) 7 -> mux4_inverter.cir
Spiceopus (c) 8 -> run
Spiceopus (c) 9 -> plot v(20) v(21)+2 v(22)+4 v(23)+6 v(10)+8 v(15)+10 v(50)+12
Spiceopus (c) 10 ->
```

Comandos utilizados para exibir a saída