

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
INTRODUÇÃO À MICROELETRÔNICA



Relatório IX

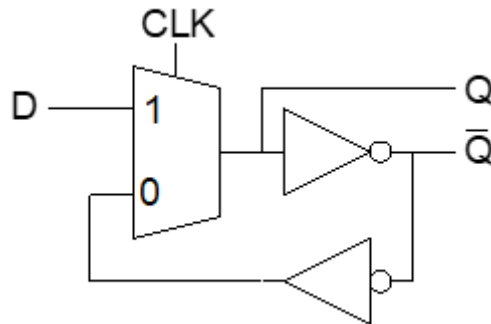
THIAGO ALVES DE ARAUJO

MATRICULA: 2016019789

João Pessoa
2018

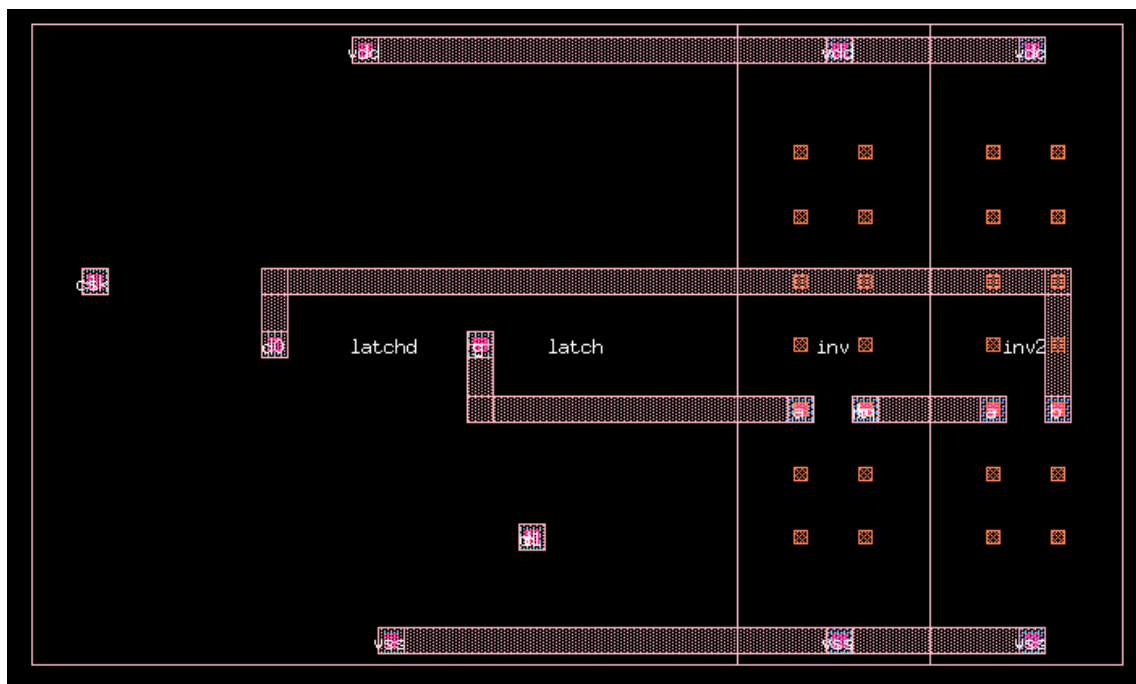
1 – Latch

Para criarmos um latch, precisamos de dois inversores e um multiplexador (não inversor) de duas entradas como mostra a figura abaixo:



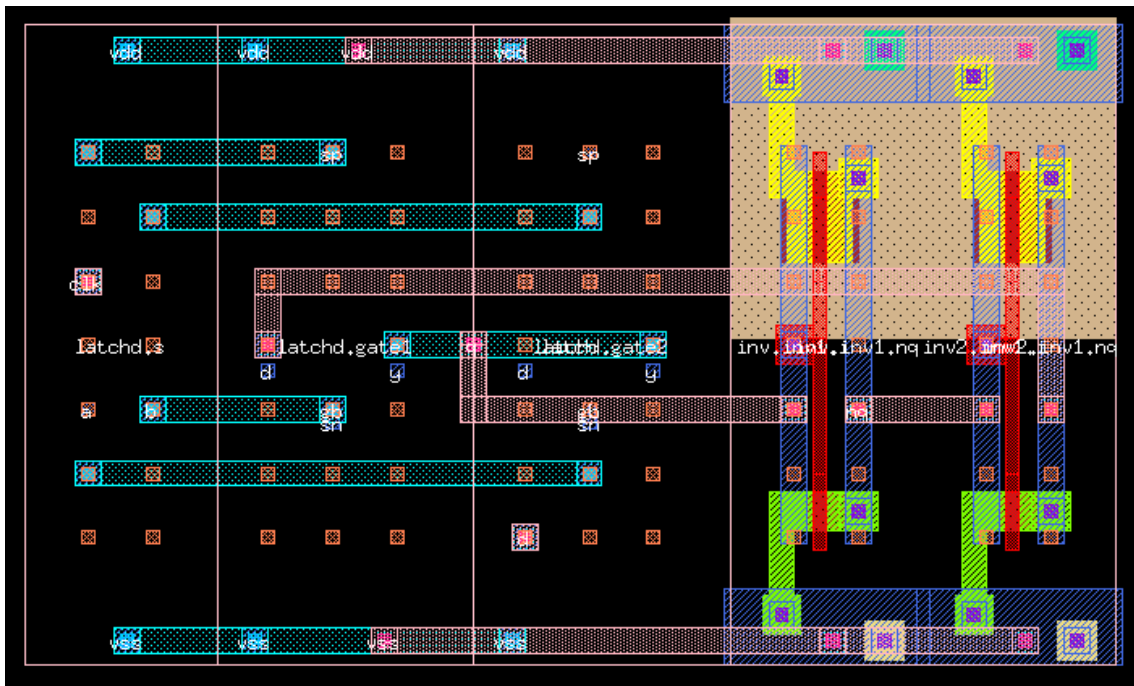
Latch D

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

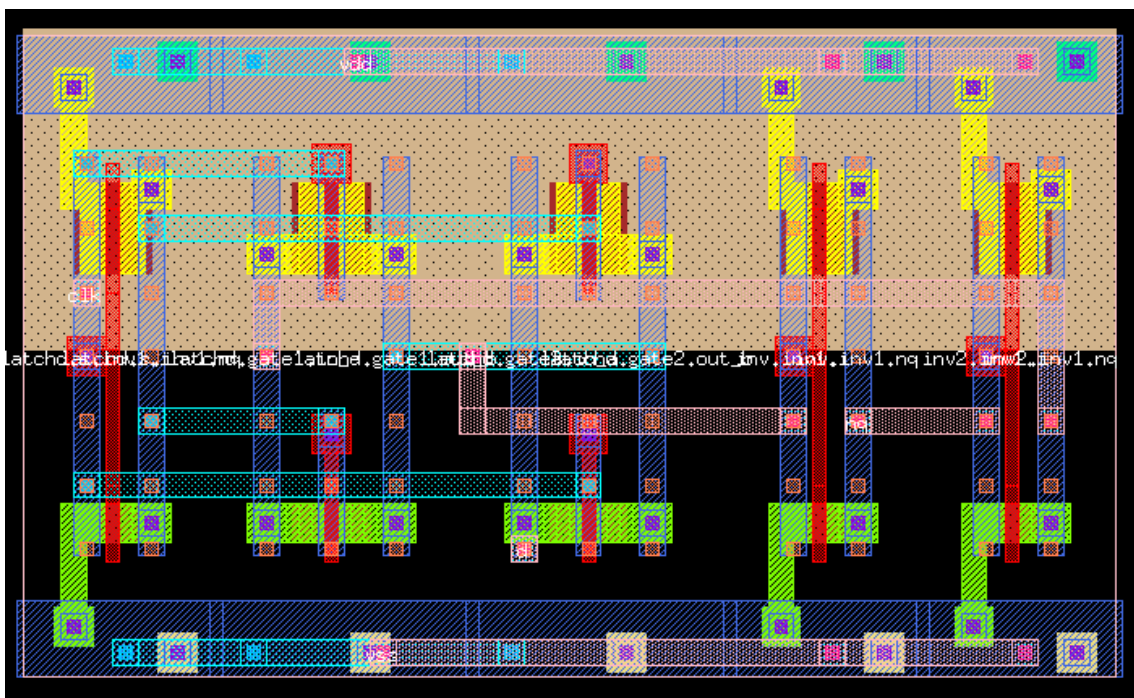


Mux 2:1 inversor

As conexões foram feitas em alu3 tendo em vista que o multiplexador possui trilhas em alu2 internamente como mostra a figura abaixo:



Latch D (flat)



Latch D (flat)

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
```

1.1 – Simulação

```
.include latch.spi

* INTERF clk d nq q vdd vss
X1      10  20 50 51 30  40 latch

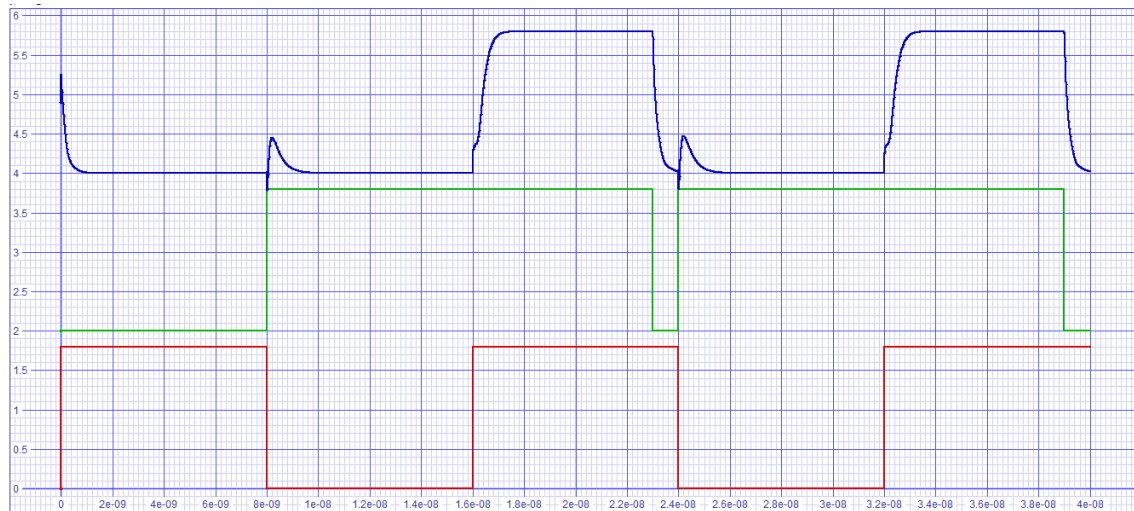
V5 10 40 pulse(0 1.8 0ns 1p 1p 8n 16n)
V4 30 40 1.8V
V3 40 0 0V
V1 20 40 pulse(0 1.8 8ns 1ps 1ps 15ns 16ns)

.model tp pmos level=54
.model tn nmos level=54
.tran 0.001ns 40ns

.end
```

João Pessoa
2018

Com isso, podemos observar o seguinte comportamento na saída:



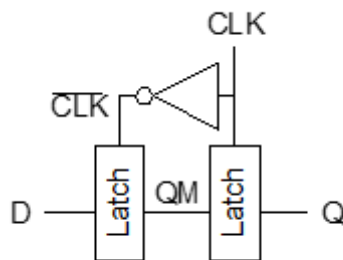
Clock(vermelho), D(verde) e Q(azul)

```
spiceopus (c) 1 -> latchsimu.cir
spiceopus (c) 2 -> run
spiceopus (c) 3 -> plot v(10) v(20)+2 v(51)+4
spiceopus (c) 4 ->
```

Comandos utilizados para exibir a saída

2 – Flip-Flop D (mestre escravo)

Para criarmos um Flip-Flop, precisamos de um inversor e dois Latch D (criado anteriormente) como mostra a figura abaixo:



Flip-Flop MS

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



2.1 – Simulação

Para simular o circuito, aplicamos sinais digitais na entrada e no clock e observamos o comportamento da saída. Abaixo está o arquivo de simulação utilizado:

```
.include flipflop.spi

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

* INTERF clock d2 q2 vdd vss
X1      10    20  50 30  40 flipflop

*ground
V1 40 0 0V

*1.8V total
V2 30 40 1.8V DC

*pulso do clock
V3 10 40 pulse(0 1.8V 7ns 1ps 1ps 12ns 28ns)

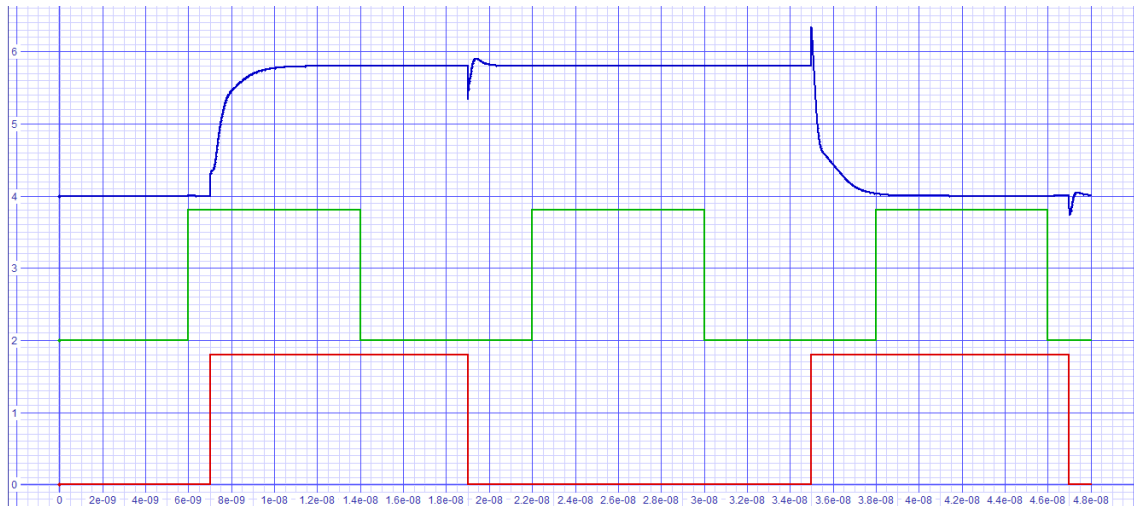
*pulso do D
V4 20 40 pulse(0 1.8V 6ns 1ps 1ps 8ns 16ns)

.tran 4ps 48ns

.end
```

Arquivo de simulação do flip-flop

Com isso, podemos observar o seguinte comportamento na saída:



Clock(vermelho), D(verde) e Q(azul)

```
spiceopus (c) 7 -> flipflop.cir
spiceopus (c) 8 -> run
spiceopus (c) 9 -> plot v(10) v(20)+2 v(50)+4
spiceopus (c) 10 ->
```

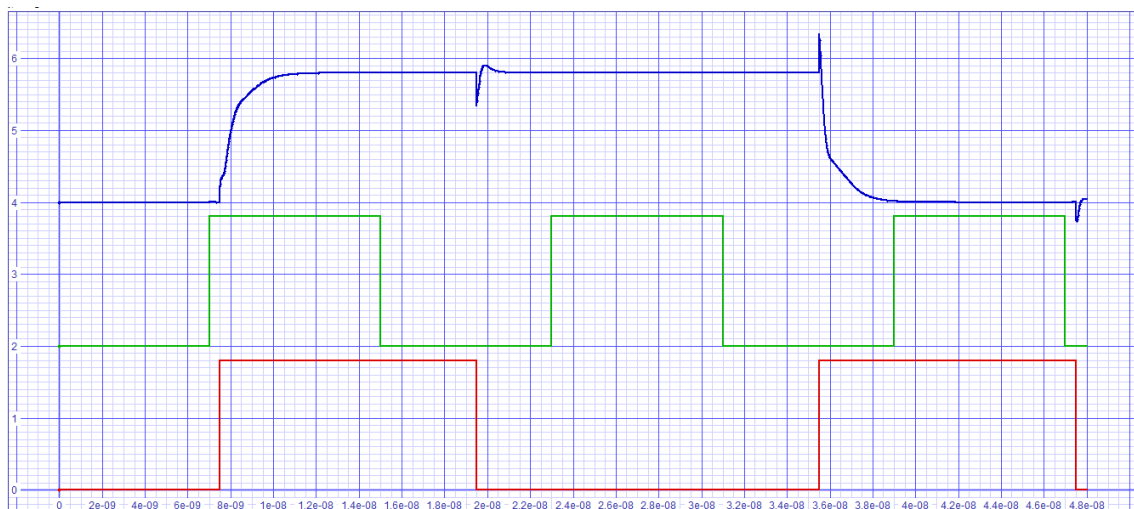
Comandos utilizados para exibir a saída

2.2 – Tempo de Setup

Para calcularmos o tempo de setup, variamos lentamente o sinal de entrada até que o sinal de saída seja afetado.

```
*pulso do clock
V3 10 40 pulse(0 1.8V 7.5ns 1ps 1ps 12ns 28ns)

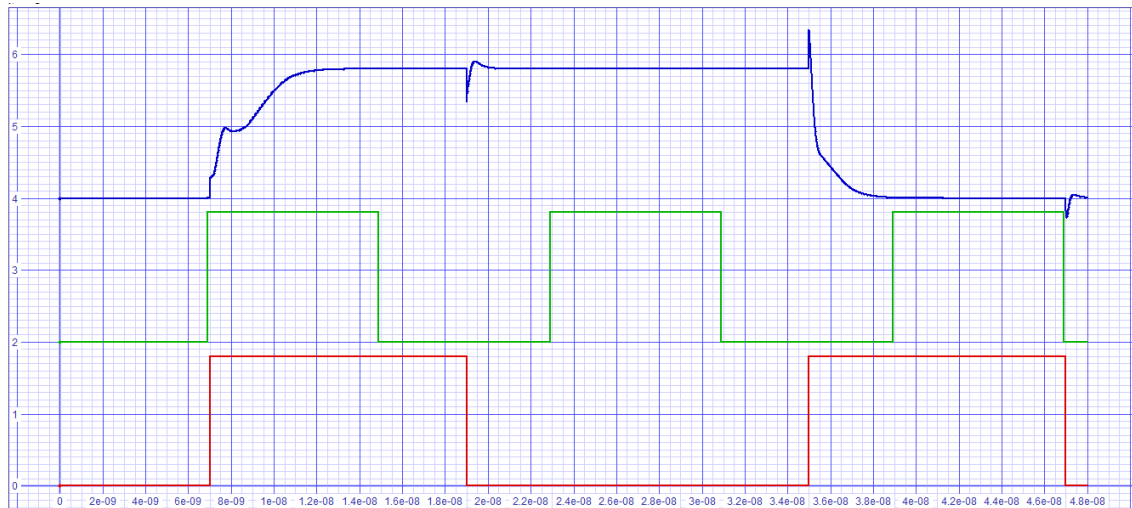
*pulso do D
V4 20 40 pulse(0 1.8V 7ns 1ps 1ps 8ns 16ns)
```



```
*pulso do clock
V3 10 40 pulse(0 1.8V 7ns 1ps 1ps 12ns 28ns)
```

```
*pulso do D
```

```
V4 20 40 pulse(0 1.8V 6.8ns 1ps 1ps 8ns 16ns)
```



```
*pulso do clock
V3 10 40 pulse(0 1.8V 7ns 1ps 1ps 12ns 28ns)
```

```
*pulso do D
```

```
V4 20 40 pulse(0 1.8V 6.9ns 1ps 1ps 8ns 16ns)
```

