

4o projeto: circuito digital multiplexador:

**Integrantes do grupo:** Lara Gama Santos, Mateus Ribeiro Ferraz, Sulamita Ester Costa.

**Objetivos:** Implementar, usando descrição fluxo de dados em Verilog HDL, um circuito Multiplexador de 2 canais e de 4 canais (MUX2:1 e MUX4:1). Este projeto possui 2 partes.

### 1a PARTE (MUX2:1):

**Entradas:** SEL, I0, I1

**Saída:** Saída.

1- Tabela-verdade:

sel	Saída
0	i0
1	i1

Quando a variável de entrada para controle (sel) for 0, então a saída será i0, porém, quando sel tiver valor 1, a saída será i1.

2- Expressões booleanas:

Saída = saída0 + saída1.

saída0= ~sel.i0.

saída1= sel.i1.

Logo,

$Saída = (\sim sel.i0) + (sel.i1)$

3- O circuito é combinacional ou sequencial?

O circuito é combinacional, pois depende apenas das entradas atuais e é modelado por expressões booleanas.

4- A descrição do circuito em Verilog HDL no Quartus:

```

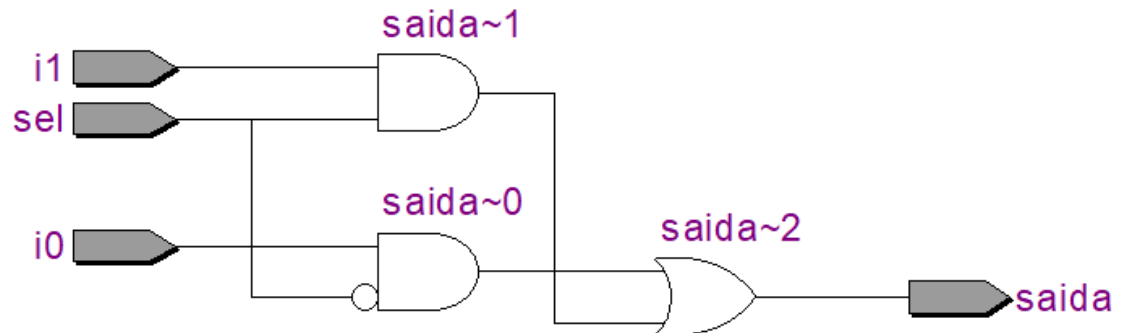
module mux_2l(i0, i1, sel, saida);
input i0, i1, sel;
output saida;

assign saida = (~sel & i0) | (sel & i1);

endmodule

```

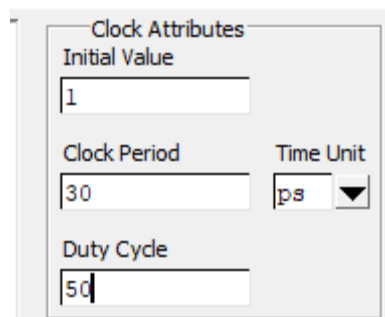
5-Esquemático do circuito obtido no Quartus II:



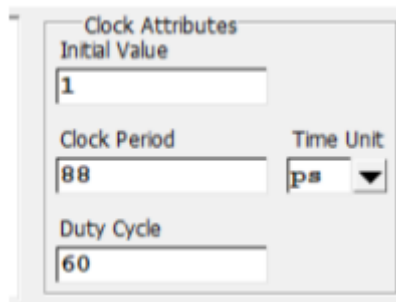
6-Simulação do circuito no ModelSim:

a- As imagens abaixo ilustram quais foram os tipos de sinal e períodos dos sinais de entrada. É possível observar que foram escolhidos períodos e ciclos diferentes para que pudessem ser comparadas diferentes possibilidades de saída. Como a variável de controle “sel” define qual será a saída, é interessante observar diferentes situações para concluir se o circuito é funcional.

**sel:**



i0:



Clock Attributes

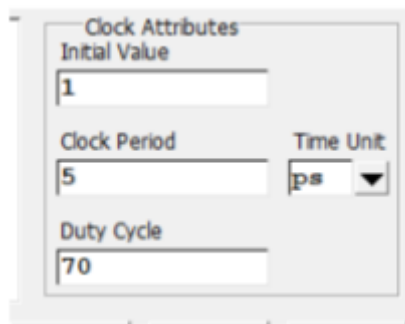
Initial Value  
1

Clock Period  
88

Time Unit  
ps

Duty Cycle  
60

i1:



Clock Attributes

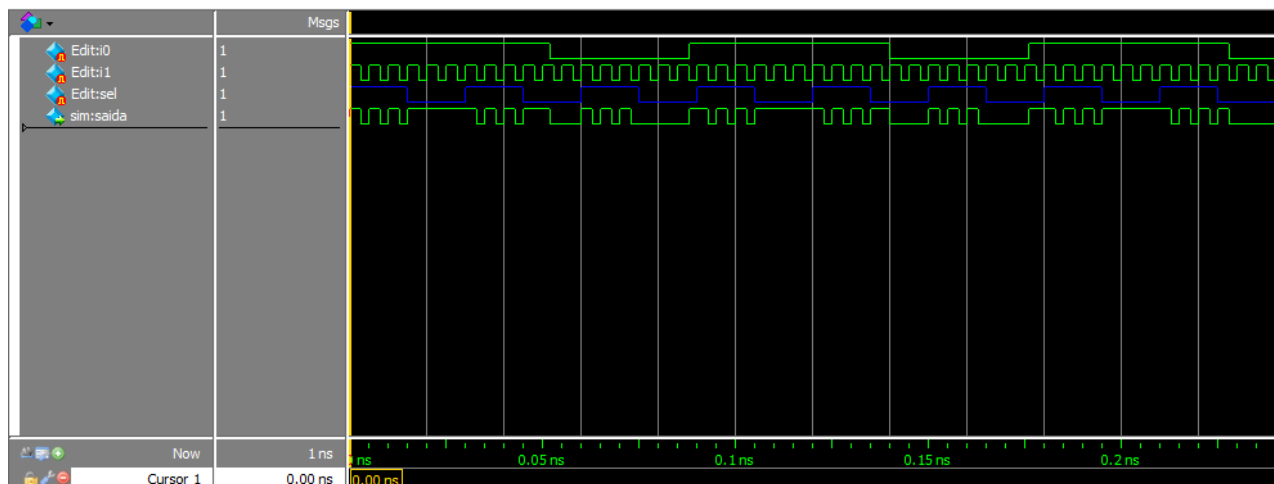
Initial Value  
1

Clock Period  
5

Time Unit  
ps

Duty Cycle  
70

b-



6

## 7- Comente e analise os resultados obtidos:

Para a execução dessa parte do projeto, implementamos, em linguagem Verilog HDL, um programa que utiliza uma variável de entrada para controle (sel) para combinar com outra entrada e regular a saída desejada. Como podemos observar no esquemático, quando "sel" receber nível lógico 0, a saída 1 será 0 ( $i1 \cdot 0 = 0$ ) e a saída 0 será i0 ( $i0 \cdot 1 = i0$ ). Dessa maneira, a saída 2 será i0, pois  $0 + i0 = i0$ . Analogamente, quando "sel" receber nível lógico 1, observamos a

seguinte situação:  $i1.1 = i1$  (saída 1),  $i0.0 = 0$  (saída 0) e  $i1+0 = i1$  (saída 2). Este é o funcionamento de um circuito digital multiplexador.

## 2a PARTE (MUX4:1):

### 1- Tabela-verdade:

A	B	S
0	0	i0
0	1	i1
1	0	i2
1	1	i3

### 2- Expressões booleanas:

$S0 = (\sim B.i0) + (B.i1)$ , baseado no MUX2:1

$S1 = (\sim B.i2) + (B.i3)$ , baseado no MUX2:1

$S = \sim A.S0 + A.S1$

Logo,  $S = \sim A(\sim B.i0 + B.i1) + A(\sim B.i2 + B.i3)$ .

### 3- O circuito é combinacional ou sequencial?

O circuito é combinacional, pois depende apenas das entradas atuais e é modelado por expressões booleanas.

### 4- A descrição do circuito em Verilog HDL no Quartus:

```
//principal - main
module mux_4l(i0, i1, i2, i3, A, B, S0, S1, S);
input i0, i1, i2, i3, A, B;
output S0, S1, S;

mux_2l mux1 (i0, i1, B, S0);
mux_2l mux2 (i2, i3, B, S1);
mux_2l mux3 (S0, S1, A, S);

endmodule
```

```

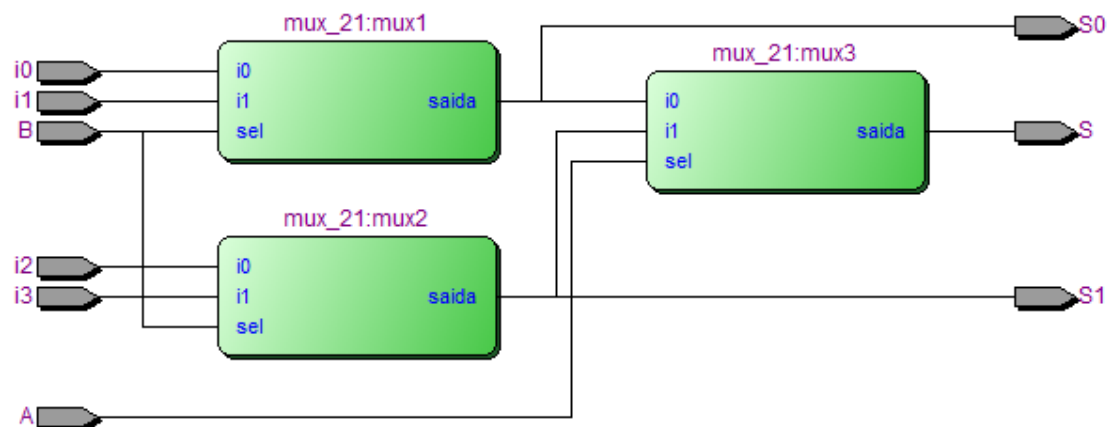
module mux_21(i0, i1, sel, saida);
input i0, i1, sel;
output saida;

assign saida = (~sel & i0) | (sel & i1);

endmodule

```

## 5-Esquemático do circuito obtido no Quartus II:



## 6- Simulação do circuito no ModelSim:

a- As imagens abaixo ilustram quais foram os tipos de sinal e períodos dos sinais de entrada. É possível observar que foram escolhidos períodos e ciclos diferentes para que pudessem ser comparadas diferentes possibilidades de saída. Como as variáveis de controle “A” e “B” definem qual será a saída, é interessante observar diferentes situações para concluir se o circuito é funcional.

i0:

The image shows the 'Clock Attributes' dialog box for a signal named 'i0'. The settings are as follows:

- Initial Value:** 1
- Clock Period:** 88
- Time Unit:** ps (picoseconds)
- Duty Cycle:** 60

**i1:**

Clock Attributes	
Initial Value	
<input type="text" value="1"/>	
Clock Period	Time Unit
<input type="text" value="5"/>	<input type="text" value="ps"/> ▼
Duty Cycle	
<input type="text" value="70"/>	

**i2:**

Clock Attributes	
Initial Value	
<input type="text" value="1"/>	
Clock Period	Time Unit
<input type="text" value="300"/>	<input type="text" value="ps"/> ▼
Duty Cycle	
<input type="text" value="50"/>	

**i3:**

Clock Attributes	
Initial Value	
<input type="text" value="1"/>	
Clock Period	Time Unit
<input type="text" value="15"/>	<input type="text" value="ps"/> ▼
Duty Cycle	
<input type="text" value="50"/>	

**A:**

Clock Attributes	
Initial Value	
<input type="text" value="1"/>	
Clock Period	Time Unit
<input type="text" value="55"/>	<input type="text" value="ps"/> ▼
Duty Cycle	
<input type="text" value="60"/>	

**B:**

Clock Attributes

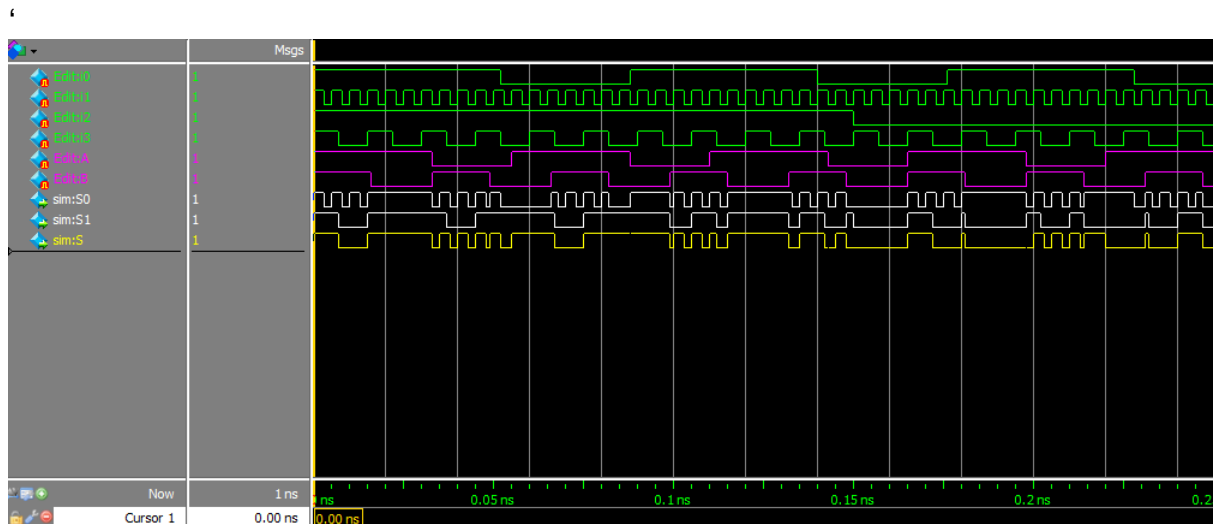
Initial Value  
1

Clock Period  
33

Time Unit  
ps

Duty Cycle  
50

**b-**



**7- Comente e analise os resultados obtidos:**

Nessa segunda parte do projeto, em que fizemos um circuito multiplexador de 4 canais, utilizamos de maneira hierárquica o código do multiplexador de 2 canais (já descrito anteriormente no relatório) para obter as saídas de acordo com as variáveis de entrada para controle. Ambos A e B definem qual será a saída do circuito, sendo que “B” é a variável de controle dos dois primeiros circuitos MUX, que recebem como entradas `i0`, `i1`, `i2` e `i3`. Além disso, “A” será a variável de controle do circuito MUX que recebe como entradas as saídas `S0` e `S1` dos outros dois MUX já citados.