

1o Projeto - sinalizador de Farol:

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

Objetivos: Implementar, usando descrição por fluxo de dados em Verilog HDL, um circuito digital que sinalize quando o motorista esquecer o farol do carro ligado:

Este circuito digital aciona um sinalizador (pode ser um buzzer, led) sempre que o Farol do carro estiver aceso desnecessariamente, quando:

- 1) a Porta estiver aberta e/ou;
- 2) a chave não estiver na ignição

Entradas: porta aberta = 0; porta fechada = 1; chave fora da ignição = 0; chave na ignição = 1; Farol desligado = 0; Farol ligado = 1;

Saída: sinalizador desligado = 0; sinalizador ligado = 1;

Farol	Porta	Chave	Saída/Sinalizador
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

1- A partir das instruções dadas no cabeçalho, como nos quatro primeiros casos da tabela verdade o farol está desligado, o sinalizador estará desligado. No entanto, nos quatro casos subsequentes, em que o farol está ligado, devemos considerar os estados da chave e da porta. Com isso, nos quinto e sexto casos, a porta está aberta, logo o sinalizador está acionado. No sétimo caso, a chave está fora da ignição, portanto, como o farol está ligado, o sinalizador será ativado. Finalmente, no último caso da tabela, o farol está acionado, entretanto a porta está fechada e a chave está na ignição, não ativando o sinalizador.

2- Como o sinalizador só liga se o farol estiver ligado ao mesmo tempo que a porta (P) estiver aberta ou a chave (C) fora da ignição, então, chegamos a seguinte expressão booleana:

$$(F \wedge \sim P \wedge \sim C) \vee (F \wedge \sim P \wedge C) \vee (F \wedge P \wedge \sim C)$$

3- O circuito não é sequencial, ele é combinacional, pois depende apenas das entradas atuais e é modelado por expressões booleanas.

4-

```
/* Projeto de um sinalizador de FAROL ligado
```

```
NOME DOS INTEGRANTES DO GRUPO:
```

```
Lara Gama Santos,  
Mateus Ribeiro Ferraz,  
Sulamita Ester Costa
```

```
DATA: 08/11/2021
```

```
Farol, Porta, Chave - entradas
```

```
Entradas:
```

```
porta aberta = 0;  
chave fora da ignição = 0;  
farol desligado = 0;  
porta fechada = 1;  
chave na ignição = 1;  
farol ligado = 1;
```

```
Saídas:
```

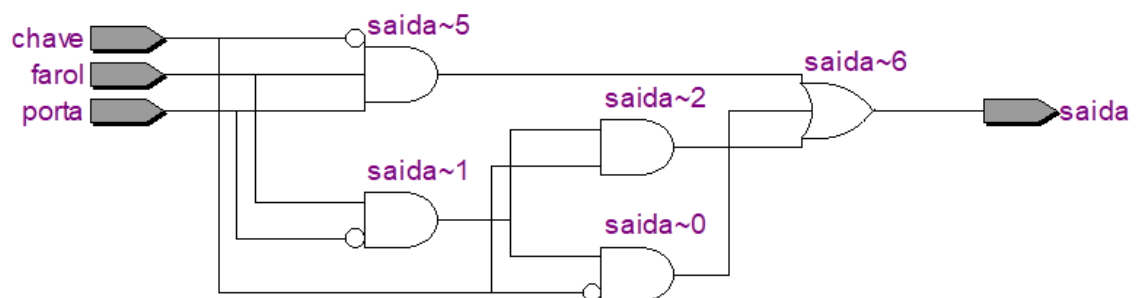
```
sinalizador desligado = 0;  
sinalizador ligado = 1;*/
```

```
module farol(farol, porta, chave, saida);  
input farol, porta, chave;  
output saida;
```

```
assign saida = (farol & ~porta & ~chave) | (farol & ~porta & chave) | (farol & porta & ~chave);
```

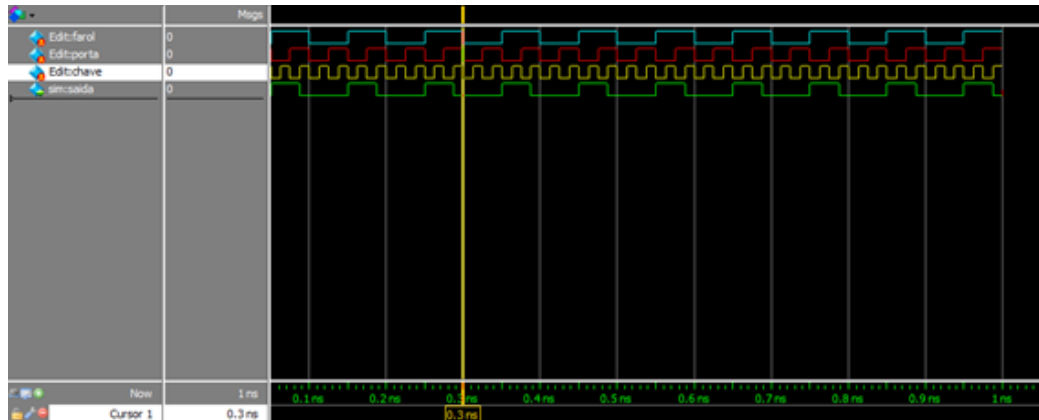
```
endmodule
```

5-



6-a- No projeto em questão, os resultados obtidos foram gerados a partir das saídas adquiridas por meio da tabela verdade. A partir disso, as ondas foram adicionadas ao simulador com períodos diferentes (100,50 e 25) para que todas as combinações da tabela verdade fossem testadas. Com isso, todas as saídas da tabela verdade foram confirmadas pelas saídas obtidas pelo simulador.

b-



7- A partir da tabela verdade, podemos inferir expressões booleanas que, quando inseridas no Quartus II por meio da linguagem Verilog, criam o esquemático de um circuito. Além disso, podemos fazer a simulação deste circuito no ModelSim para colocar as entradas obtidas na tabela verdade e comparar a veracidade das saídas resultantes nas ondas e na tabela.