

Relatório – Farol

Aluno: Alexandre Roque Silva de Paula

Disciplina: Sistemas Digitais. Professora: Mara Coelho.

1 e 2) Tabela Verdade:

F	P	C	S
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

Explicação:

inputs: (F, P, C) Farol, Porta, Chave;

output: S – sinalizador;

Sendo os níveis lógicos 1 e 0, definidos como: porta aberta = 0;

porta fechada = 1; chave na ignição = 1; Farol ligado = 1; sinalizador ligado = 1

i) a Porta estiver aberta; $S = \sim P \& F$

ii) a chave não estiver na ignição; $S = \sim C \& F$

$S = (\sim P \& F) \text{ or } (\sim C \& F)$.

3) (Arquivos em anexo em pastas diferentes)

```
1  /*1 prática de laboratório. Outubro/2020. Aluno: Alexandre Roque
2  Objetivo: Implementar, usando duas formas de descrição em Verilog HDL, um circuito digital que sinalize quando
3  o motorista esquecer o farol do carro ligado:
4  A) Circuito descrito por fluxo de dados em Verilog HDL (expressão booleana);
5  B) Circuito descrito de maneira comportamental em Verilog HDL (use a instrução CASE);
6  Regras:
7  Considere como entradas do circuito as variáveis:
8  inputs: (F, P, C) Farol, Porta, Chave;;
9  E como saída:output: S – sinalizador;
10 Este circuito digital aciona um sinalizador (pode ser um buzzer, led) sempre que o Farol do carro estiver aceso
11 desnecessariamente, quando:
12 1) a Porta estiver aberta;  $S = \sim P \& F$ 
13 2) a chave não estiver na ignição.  $S = \sim C \& F$  */
14
15 module farolFluxo (S, F, P, C); //cabeçalho
16
17     input F, P, C;
18     output S;
19
20     assign S = (F & ~P) | (F & ~C);
21
22 endmodule
23 |
```

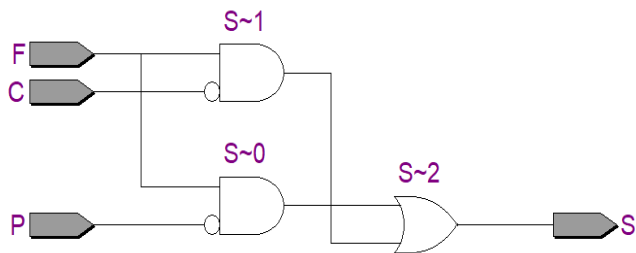
```

6 Regras:
7 Considere como entradas do circuito as variáveis:
8 inputs: (F, P, C) Farol, Porta, Chave,;
9 E como saída: output: S - sinalizador;
10 Este circuito digital aciona um sinalizador (pode ser um buzzer, led) sempre que o Farol do carro estiver aceso
11 desnecessariamente, quando:
12 |
13 Sendo os níveis lógicos 1 e 0, definidos como:
14 porta aberta = 0; porta fechada = 1;
15 chave na ignição = 1; Farol ligado =1; sinalizador ligado =1;
16
17 1) a Porta estiver aberta: S = ~P & F
18 2) a chave não estiver na ignição. S = ~C & F| */
19
20 module farolComportamental (S, F, P, C); //cabeçalho
21
22     input F, P, C;
23     output reg S;
24
25     always@(*)
26     begin
27         case ({F,P,C})
28             3'b000: S = 0;
29             3'b001: S = 0;
30             3'b010: S = 0;
31             3'b011: S = 0;
32             3'b100: S = 1;
33             3'b101: S = 1;
34             3'b110: S = 1;
35             3'b111: S = 0;
36         endcase
37     end
38
39 endmodule
40

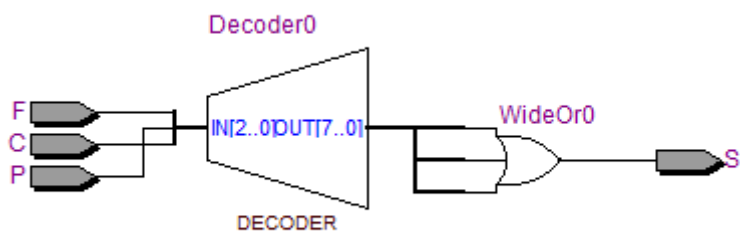
```

4)

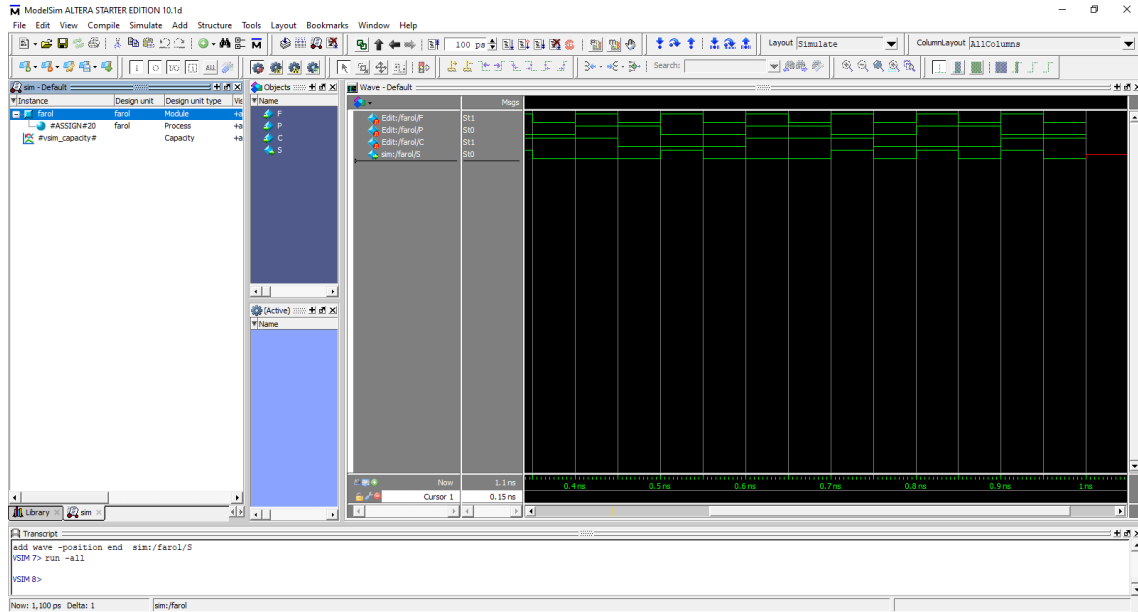
Fluxo de Dados:



Comportamental:



5)



A partir do ModelSim, criei um projeto com as entradas F, P e C, e uma saída S.

A entrada F está variando em 100 ps.

A entrada P está variando em 200 ps.

A entrada C está variando em 300 ps.

O tempo máximo de simulação é 1000 ps.

O resultado era o esperado a partir da tabela verdade e foi o mesmo para os circuitos comportamental e o de fluxo de dados. O gráfico mostra que o sinal será “HIGH”, quando a porta estiver aberta, o farol ligado e a chave estiver na ignição, também quando a porta estiver aberta, chave na ignição e o farol ligado, por último farol ligado, chave não estiver na ignição e a porta fechada.