Experiência *8*
Semáforos de Trânsito II

Disciplina: PCS 3335 – Laboratório Digital A	
Prof.: *Bruno Albertini*	**Data:** *21/05/2020* | *25/06/2020*
Turma: *10*	**Bancada:** *B4*
Membros:	
10696510 Antonio Nigro Zamur	
11027484 Guilherme Akira Alves dos Santos	

1. Introdução

Por meio de uma lógica sequencial e uma lógica combinatória, nessa experiência irá se estruturar um sistema de semáforos de trânsito mais complexo que o da experiência anterior, acrescentando a presença de pedestres na lógica da unidade de controle.

2. Objetivo

Essa atividade visa praticar ainda mais os conceitos de fluxo de dados, unidade de controle, máquina de estados e diagrama de transição de estados enquanto é desenvolvido um projeto de sistema digital, dessa vez, com mais um desafio.

3. Planejamento

a. Projeto

i. Descrição Funcional

Para essa experiência precisaremos atualizar a tabela verdade da experiência passada, tendo uma nova variável “p”, que controlará o semáforo dos pedestres, em dois novos estados, “101” e “111”.
O estado “111” existe para, antes de liberar a passagem de pedestres, manter os semáforos de veículos fechados por 2 segundos, trazendo mais segurança aos pedestres.
Como observado na figura 1 do experimento, há um único comando para todos os semáforos de pedestres, sendo assim, desenvolvemos um sistema em que apenas uma variável é responsável pelo seu acionamento;

Tabela 1 - Tabela verdade para o bloco de lógica combinatória

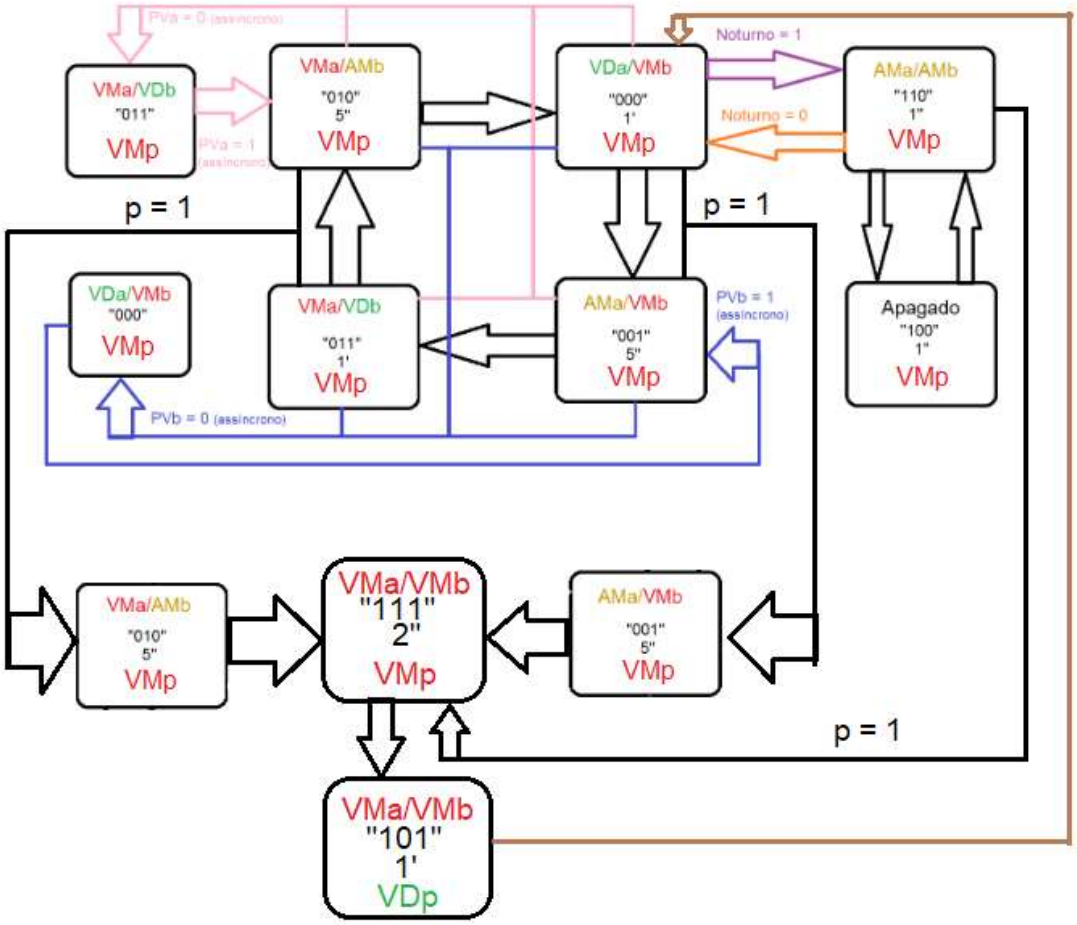
Dec	Q_2	Q_1	Q_0	VMa	AMa	VDa	VMb	AMb	VDb	VMp	VDp
0	0	0	0	0	0	1	1	0	0	1	0
1	0	0	1	0	1	0	1	0	0	1	0
2	0	1	0	1	0	0	0	1	0	1	0
3	0	1	1	1	0	0	0	0	1	1	0
4	1	0	0	0	0	0	0	0	0	1	0
5	1	0	1	1	0	0	1	0	0	0	1
6	1	1	0	0	1	0	0	1	0	1	0
7	1	1	1	1	0	0	1	0	0	1	0

Nota-se que a coluna correspondente à variável “p” é chamada “VDp” e “VMp”, pois quando está em valor lógico 1, ativa o sinal verde para os pedestres e fecha o sinal de todos os carros. Em todos os outros momentos, quando seu valor é “0”, ela manterá o sinal dos pedestres em vermelho.

ii. Diagrama de Blocos

Para seguir uma lógica segura, antes de fechar os dois lados de circulação de veículos, devemos manter o farol vermelho do lado previamente fechado e transformar em amarelo o lado aberto, para apenas então fechar a circulação de automóveis e liberar a travessia de pedestres;

Figura 1 - Transição de estados da unidade de controle:



iii. Diagrama Lógico

Figura 2 - Diagrama lógico do circuito combinatório

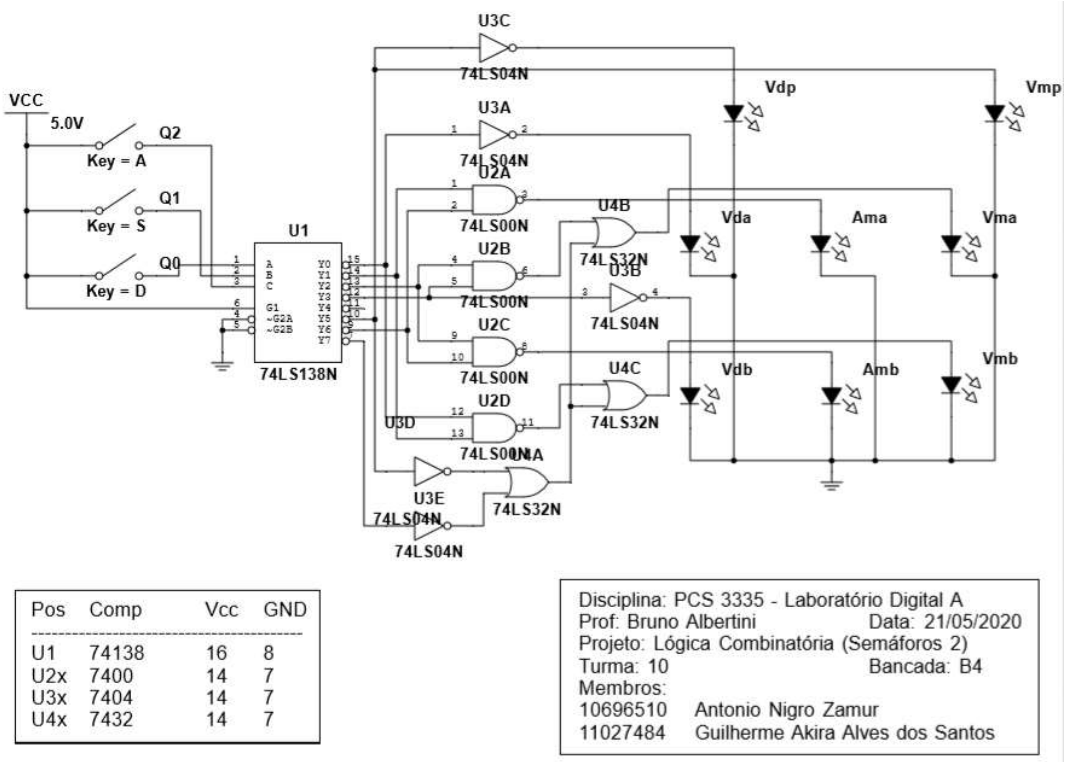


Figura 3 - Código VHDL da nova lógica sequencial parte 1

```
4
5   entity logica_sequencial is
6       generic(
7           short_delay : integer := 5; -- valor em segundos
8           long_delay  : integer := 60 -- valor em segundos
9       );
10      port(
11          pva, pvb, noturno, reset, p: in std_logic;
12          clock: in std_logic;
13          q: out std_logic_vector (2 downto 0)
14      );
15  end logica_sequencial;
16
17  architecture arch_sequencial of logica_sequencial is
18
19      signal troca_dia  :  std_logic := '0';
20      signal Eatual   :  std_logic_vector (2 downto 0) := "000";
21      signal seg_count  :  integer range 0 to 60 := 0;
22      signal pedestre : std_logic := '0'; -- guarda informação de pedestre
23
24  begin
25
26      Counter: process (reset, clock, noturno, Eatual, pva, pvb, p) is
27      begin
28
29          if reset = '1' then      -- reset assincrono
30              Eatual <= "100";
31              seg_count <= 0;
32              pedestre <= '0';
33
34          elsif  p = '1' and pedestre = '0' then -- logica para gravar presença de pedestre
35              pedestre <= '1';
36
37          -- suspende contagem de tempo quando sem carro em uma via
38          elsif noturno = '0' and pva = '0' and Eatual = "011" and pedestre = '0' then
39              seg_count <= 0;
40          elsif noturno = '0' and pvb = '0' and Eatual ="000" and pedestre = '0' then
41              seg_count <= 0;
42
43          -- quando ha subida de clock
44          elsif clock'event and clock = '1' then
45              seg_count <= seg_count + 1;
46
```

Armazena presença de pedestres

Sensivel a pedestres assincronos

Nova condição PVa e PVb

Figura 4 - Código VHDL da nova lógica sequencial parte 2

```
38    elsif noturno = '0' and pva = '0' and Eatual = "011" and pedestre = '0' then
39       seg_count <= 0;
40    elsif noturno = '0' and pvb = '0' and Eatual ="000" and pedestre = '0' then
41       seg_count <= 0;
42
43    -- quando ha subida de clock
44    elsif clock'event and clock = '1' then
45       seg_count <= seg_count + 1;
46
47
48
49
50       if pedestre = '1' and Eatual = "000" then -- transicoes pedestres
51          Eatual <= "001";
52          seg_count <= 0;
53       elsif pedestre = '1' and Eatual = "011" then
54          Eatual <= "010";
55          seg_count <= 0;
56       elsif pedestre = '1' and seg_count = (short_delay -1) and Eatual = "001" then
57          Eatual <= "111";
58          seg_count <= 0;
59       elsif pedestre = '1' and seg_count = (short_delay -1) and Eatual = "010" then
60          Eatual <= "111";
61          seg_count <= 0;
62       elsif pedestre = '1' and Eatual = "110" then -- amarelo piscante (sempre caira aqui, mesmo no estado "100")
63          Eatual <= "111";
64          seg_count <= 0;
65       elsif pedestre = '1' and seg_count = (short_delay - 3) and Eatual = "111" then --TUDO VERMELHO POR 2 SEGUNDOS
66          Eatual <= "101";
67          seg_count <= 0;
68       elsif pedestre = '1' and seg_count = (long_delay -1) and Eatual = "101" then --Carros parados, pedestres atrav
69             Eatual <= "000";
70             pedestre <= '0';
71
72
73       -- transicoes noturnas
74       elsif noturno = '1' and Eatual = "000" then
75          Eatual <= "110";
76          seg_count <= 0;
77       elsif noturno = '1' and Eatual = "110" then
```

Condições para pedestre

iv. Simulações

Figura 5 - Carta de tempo do circuito combinatório

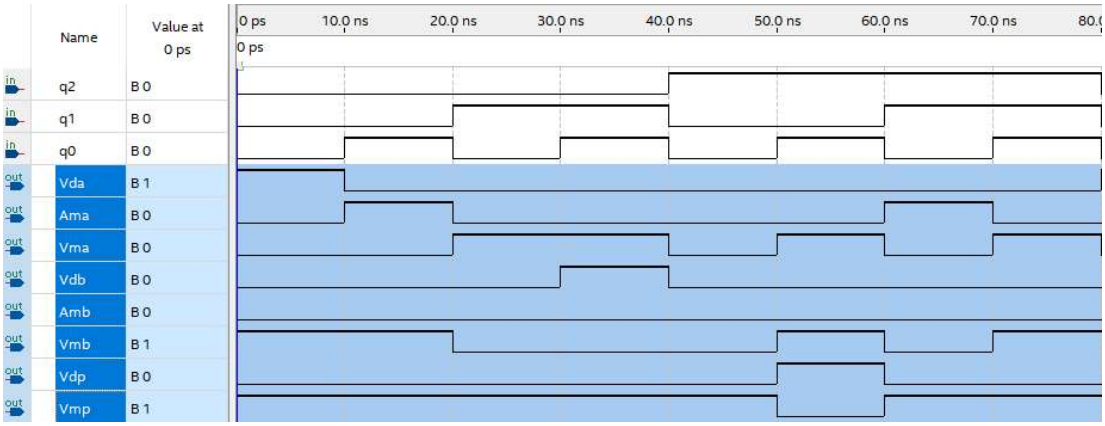


Figura 6 - Ciclo padrão na lógica sequencial

Figura 7 - Ciclo pedestres na lógica sequencial

Figura 8 - Ciclo Pedestres + PVa e PVb

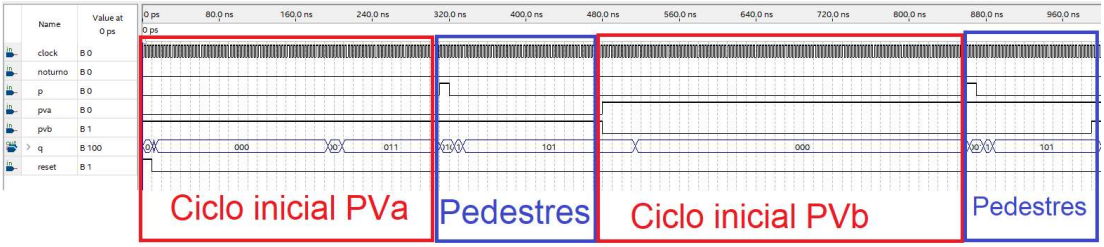
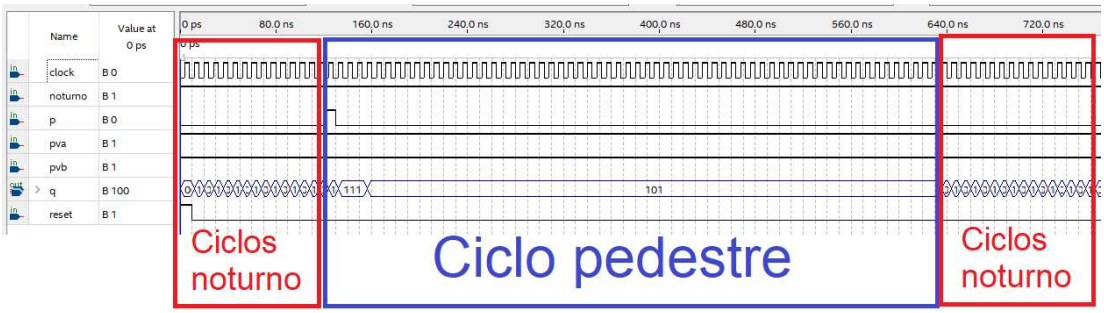


Figura 9 - Ciclo Pedestre + noturno



b. Estratégias de montagens, testes e depuração

i. Estratégia de Montagem

Tabela 2 - Tabela de pinagem unidade_controle

Variável	Tipo	Porta	Pinagem
Clock	Input	CLOCK_50	PIN_M9
Reset	Input	SW0	PIN_U13
Noturno	Input	SW1	PIN_V13
PVa	Input	SW2	PIN_T13
PVb	Input	SW3	PIN_T12
p	Input	SW4	PIN_AA15
semaforos[0]	Output	LEDR0	PIN_AA2
semaforos[1]	Output	LEDR1	PIN_AA1
semaforos[2]	Output	LEDR2	PIN_W2
semaforos[3]	Output	LEDR3	PIN_Y3
semaforos[4]	Output	LEDR4	PIN_N2
semaforos[5]	Output	LEDR5	PIN_N1
semaforos[6]	Output	LEDR6	PIN_U2
semaforos[7]	Output	LEDR7	PIN_U1

ii. Testes e Depuração

Para testar o bloco de lógica combinatória, serão averiguados os valores das saídas do decodificador. Comprovado seu funcionamento, serão posicionadas as portas NAND e em seguida estas serão testadas. O mesmo se aplica para a disposição das portas NOT.
Após confirmado o funcionamento dos componentes, serão conectados *switches* às entradas do decodificador. Com estes, ao chaveá-los, será comparado o resultado obtido com a tabela verdade esperada (vide Tabela 1).

Em seguida, ao acoplar o circuito da FPGA ao circuito combinatório, serão testados alguns cenários que se assemelham a uma situação da vida real, conferindo se o resultado do circuito remete ao esperado.

4. Relatório

a. Resultados obtidos

O experimento atendeu às expectativas logo em sua primeira tentativa. Apenas em um caso de teste o funcionamento deste aparentou estar errado, no entanto, após breve análise constatou-se que o problema se dava pela readaptação dos clocks para que estes se enquadrassem ao tempo proposto ao experimento. Para a execução do desafio, este problema foi corrigido.

b. Pontos positivos e negativos

Positivos: Pudemos observar o funcionamento da placa FPGA e ter nossa atividade avaliada mesmo a distância.

Negativos: Devido ao estado atual do mundo em pandemia, não pudemos ter contato direto com a placa FPGA; dessa forma a experiência de aprendizado não se torna completa.

c. Lições aprendidas

Com essa experiência aprendemos o funcionamento de uma placa FPGA e desenvolvemos uma unidade de controle sensível a variáveis externas (sinais PVa, PVb, p, noturno). Com as técnicas executadas em aula, pudemos aprender o processo de sintetização do código e realmente fazer o upload à placa.

Desafio

Para executar o desafio criamos novas saídas de LED nos estados já existentes; assim alternando entre um sinal vermelho e verde com lógica inversa ao sinal principal (que permite o cruzamento da via em linha reta). Com essa adição ao sistema, o tráfego pode ser otimizado. Como decisão de projeto, ao abrir a passagem de pedestres, todos os sinais para carros são fechados.

Tabela 3 - Novos pinos para controle de conversão à esquerda

Variável	Tipo	Porta	Pinagem
semaforos[8]	Output	LEDR8	PIN_L2
semaforos[9]	Output	LEDR9	PIN_L1

Assim, quando semaforos[9] estiver em estado lógico 1, representa que o semáforo de conversão à esquerda na via A está em verde; quando este estiver em estado lógico 0, representa que o mesmo semáforo se encontra em vermelho. A mesma regra se aplica ao semaforos[8], correspondendo aos sinais da via B.
No primeiro teste do desafio, o parâmetro *generic* referente ao clock de entrada foi passado errado (2Hz ao invés de 50E6Hz), por isso, o circuito temporizador não funcionou como esperado, gerando uma resposta errada do sistema. Após correção do problema mencionado anteriormente, obtivemos sucesso na resolução do desafio, assim criando um sinal de passagem para conversão à esquerda.

Referências

Apostilas e documentos de apoio do site ~/labdig do PCS.
Apostilas disponíveis na plataforma e-Disciplinas.
Tópicos da disciplina Sistemas Digitais 1.