Universidade Federal de São Carlos - UFSCar
Ciência da Computação

Laboratório de Circuitos Digitais
Experimento 5

Turma – A

Caio Vinicius Barbosa Santos, 726503
Mario Luiz Gambim, 744345
Mayk Tulio Bezerra, 727953
Prof. Fredy João Valente

São Carlos
05 de Dezembro de 2018

1. Introdução

O experimento 5 consiste na aplicação das técnicas de construção de máquinas de estados finitas, tanto de Mealy quanto Moore em 2 projetos distintos, onde para a resolução de cada será construído a FSM especificado e gerado seu código equivalente em uma linguagem de descrição de hardware Verilog, após isso uma simulação do código será efetuada para todos estados previstos na máquina, por fim o deploy da placa DE1 Altera será realizado para visualização dos resultados obtidos na prática.

O primeiro projeto é requerido a construção de uma máquina de Moore para resolução e possui o seguinte enunciado:

"Considere o cenário de um controle para um portão de garagem basculante. Em um estado inicial, o portão está fechado. Caso um acionador externo seja selecionado, o portão abre. Caso o portão esteja aberto e o acionador externo seja selecionado, o portão fecha. O portão nunca abre e fecha ao mesmo tempo. O trilho no qual o portão se desloca é equipado com dois sensores que indicam quando o portão está completamente aberto e quando está completamente fechado. O motor não deve tentar abrir o portão quando esse estiver aberto e nem deve fechá-lo quando este já estiver fechado. Para maior segurança dos usuários, o motor está equipado com um aviso luminoso que deve ser aceso quando o portão se desloca. Além disso deve existir um sistema de proteção ao usuário, para que o portão não feche caso tenha um veículo ou pessoa passando no portão (usar um feixe de infravermelho). Deve-se assumir que não é possível parar o portão enquanto ele estiver abrindo ou fechando, mas é possível que o usuário aperte o acionador externo enquanto o portão estiver se deslocando. Nesse caso, se o portão estiver abrindo, ele deve passar a fechar e vice-versa. Considere ainda o caso de falta de energia na máquina de estados. Caso ocorra falta e energia, o portão se em movimento vai parar. Quando a energia voltar, deve-se considerar os sensores de fim de curso no trilho e fazer com que o portão vá para o estado totalmente aberto ou fechado (considerar presença de pessoas ou veículos no portão), antes de retomar a sequência da máquina de estados."

O segundo projeto é a implementação de um circuito sequencial simulando um sistema de irrigação de jardim /horta. Segue as instruções passadas:

"Considere o cenário de um sistema de irrigação de jardim e/ou horta. O sistema possui 7 esguichos de irrigação (sprinklers). Em um estado inicial, os sprinklers estão todos desligados.

Todos os sprinklers são ligados automaticamente ao amanhecer por 30 minutos e ao anoitecer por 1 hora. O sistema também possui um controle remoto que liga/desliga o sistema a qualquer horário, com as seguintes características: Se o sistema estiver ligado, o sistema desliga com o controle remoto. Se o sistema estiver desligado o sistema liga e fica ligado por 1 hora e desliga automaticamente."

2. Procedimento Experimental

2.1 - Etapa 1

Iniciamos o projeto mapeando os possíveis estados da máquina de Moore, e quais informações do enunciado do problema seriam mapeados para as possíveis transições, chegamos no seguinte mapeamento:

Estados:

S1	Portão fechado
S2	Portão abrindo
S3	Portão aberto
S4	Portão fechando

O mapeamento do valor da transição foi realizado através de um binário de 4 bits, onde cada bit representa uma ação, exemplo: controle acionado, leitor de presença de pessoas ativo, trilho indicando que o portão está totalmente fechado, etc.

A seguir será exibido uma tabela contendo o significado de cada bit do BCD de transição de 4 bits.

B3	B2	B1	B0
Controle	Trilho aberto	Trilho fechado	Sensor de presença

Cada um dos bits irá assumir o valor de 0 ou 1 como demonstrado a seguir:

Controle	0 - Não acionado	1 - Acionado
Trilho aberto	0 - Parcialmente aberto	1 - Totalmente aberto
Trilho fechado	0 - Parcialmente fechado	1 - Totalmente fechado
Sensor de presença	0 - Sem presença	1 - Possui presença

2.2.1 Máquina de Moore gerada

A partir do mapeamento realizado anteriormente uma máquina de Moore foi construída como resolução do problema proposto, a máquina é exibida na imagem 1, todas transições que possuem valor ‘x’ em um dos seus bits independe daquele valor.

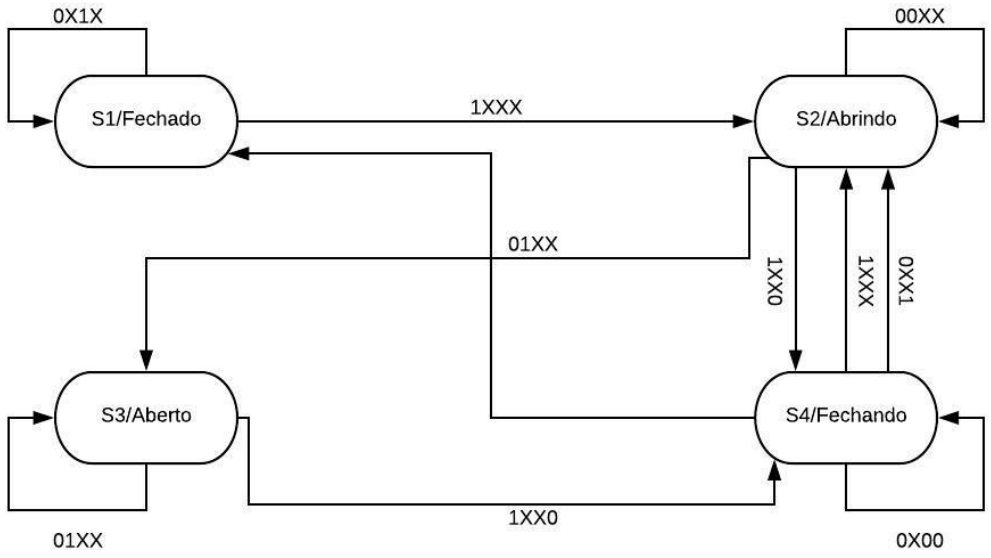

Imagem 1 - Diagrama da máquina de estado construída

.

```
module Experimento5_moore(controle,trilhoAberto,trilhoFechado,sensor,clock);

input controle,trilhoAberto,trilhoFechado,sensor,clock;
reg[1:0] estado;
reg[4:1] entrada;
parameter Fechado = 2'b00, Aberto = 2'b11, Fechando = 2'b01, Abrindo = 2'b10;
initial estado = Fechado;
always @(posedge clock) begin
   entrada[4] = controle;
   entrada[3] = trilhoAberto;
   entrada[2] = trilhoFechado;
   entrada[1] = sensor;

case(estado)
Fechado: begin
         //OQ GERA DE SAIDA AQUI
         if( entrada[4] == (1'b1))
            estado = Abrindo;
         end

         Abrindo: begin
         //OQ GERA DE SAIDA AQUI
         if ((entrada[4] == (1'b0)) && (entrada[3] == (1'b1)))
            estado = Aberto;
         if ((entrada[4] == (1'b1)) && (entrada[1] == (1'b0)))
            estado = Fechando;
         end

         Aberto: begin
            //OQ GERA DE SAIDA AQUI
            if ((entrada[4] == (1'b1)) && (entrada[1] == (1'b0)))
               estado = Fechando;
         end

         Fechando: begin
            //OQ GERA DE SAIDA AQUI
            if ((entrada[4]) == (1'b1))
               estado = Abrindo;
            if ((entrada[4] == (1'b0)) && ((entrada[2]) == (1'b1)))
               estado = Fechado;
            if ((entrada[4] == (1'b0)) && ((entrada[1] == 1'b1)))
               estado = Abrindo;
         end
         default: estado = Fechado;
      endcase
   end
endmodule
```

Imagem 2 - Código da máquina de estado

```
module Experimento5_moore(SW,HEX0,clock);

input [3:0]SW;
input clock;
output reg [0:6]HEX0;
reg[1:0] estado;
reg[4:1] entrada;
parameter Fechado = 2'b00, Aberto = 2'b11, Fechando = 2'b01, Abrindo = 2'b10;
initial estado = Fechado;


always @(posedge clock) begin
    entrada[4] = SW[3]; // CONTROLE
    entrada[3] = SW[0]; //Trilho Aberto
    entrada[2] = SW[1]; // Trilho Fechado
    entrada[1] = SW[2]; // SENSOR

case(estado)
Fechado: begin
            //C = FECHADO
            HEX0 =  7'b0110001;
            if( entrada[4] == (1'b1))
                estado = Abrindo;
            end

Abrindo: begin
            //A = ABRINDO
            HEX0 = 7'b0001000;

            if ((entrada[4] == (1'b0)) && (entrada[3] == 1))
                estado = Aberto;
            if ((entrada[4] == (1'b1)) && (entrada[1] == (1'b0)))
                estado = Fechando;
            end

Aberto: begin
                //O = OPEN
                HEX0 = 7'b0000001;
                if ((entrada[4] == (1'b1)) && (entrada[1] == (1'b0)))
                    estado = Fechando;
            end

Fechando: begin
                //F = FECHANDO
                HEX0 = 7'b0111000;
                if ((entrada[4]) == (1'b1))
                    estado = Abrindo;
                if ((entrada[4] == (1'b0)) && ((entrada[2]) == (1'b1)))
                    estado = Fechado;
                if ((entrada[4] == (1'b0)) && ((entrada[1] == 1'b1)))
                    estado = Abrindo;
            end
            default: estado = Fechado;
        endcase
    end
endmodule
```

Imagem 3 - Código fonte em verilog aplicado a FPGA.

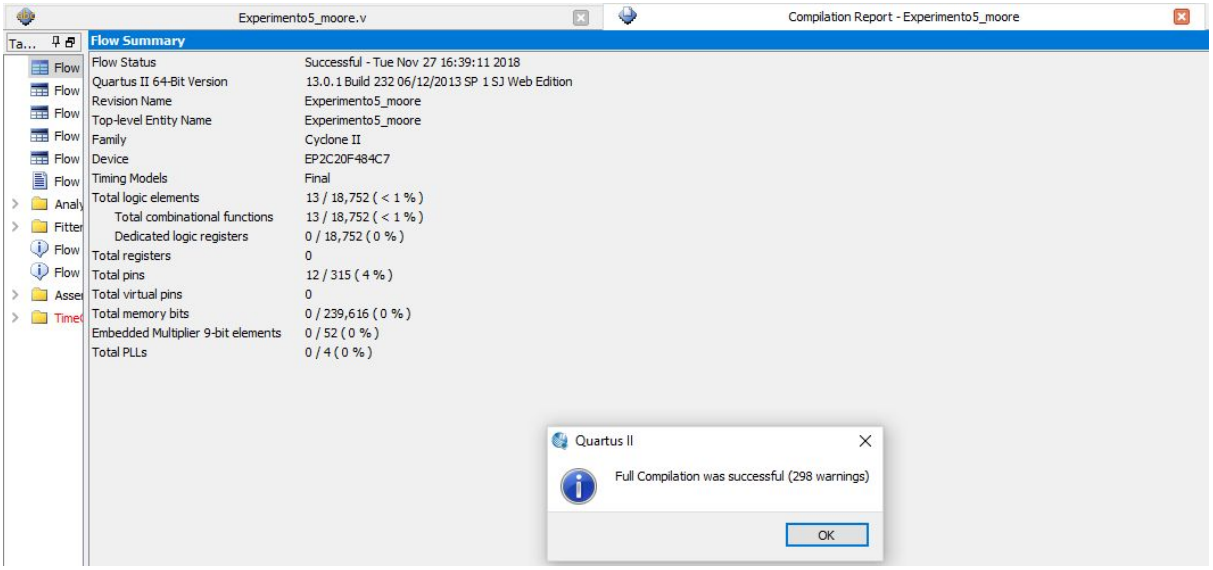

Imagem 4 - Compilação do código em verilog

2.1 - Etapa 2

O circuito a ser implementado é uma simulação de um sistema de irrigação de jardim /horta. Segue na imagem a máquina de estado com as instruções passadas, onde a máquina deveria ligar ao acionar o controle e ligar durante 30 minutos quando amanhecer e 60 minutos quando anoitecer. Segue a máquina desejada:

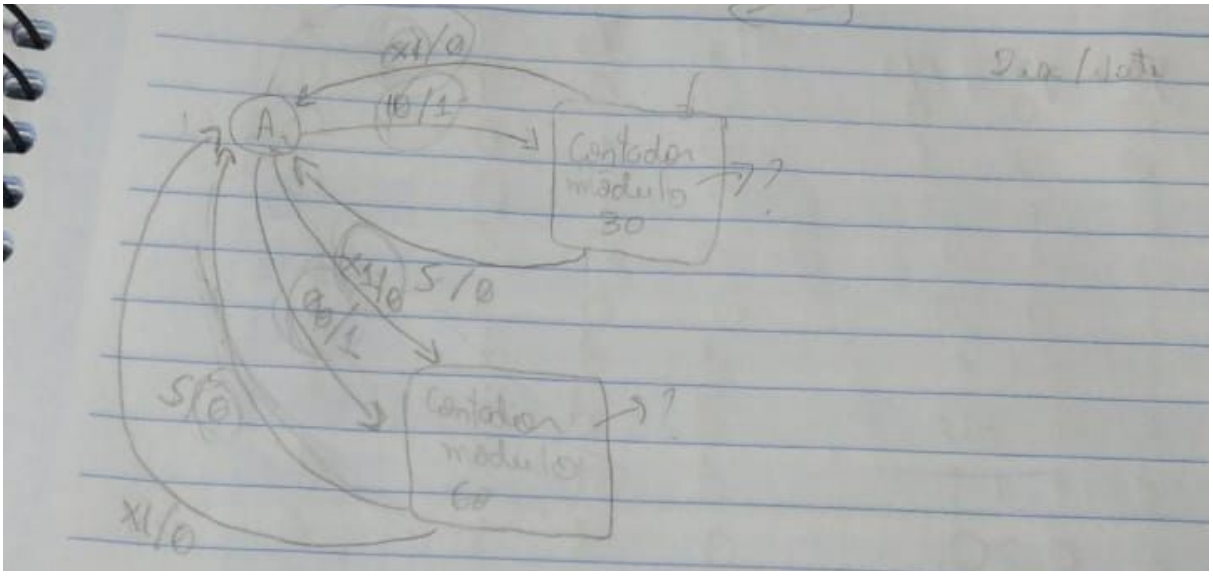

Imagem 5 - Máquina de estado de Mealy do sistema de irrigação

```
module Exp_5_Mealy(SW, HEX1, clk_50mhz, rst_50mhz);


   input [1:0]SW;
   output reg [0:6]HEX1;
   input clk_50mhz, rst_50mhz;
   //ESTADOS
   reg [1:0]estado;
   reg [1:0]entrada;
   reg [17:0] count_reg = 0;
   reg out_100hz = 0;
   parameter S1 = 2'b00, S2 = 2'b01, S3 = 2'b10;
   //S1 INICIAL
   //S2 30 MINUTOS
   //S3 1H
   initial estado = S1;

   always @ (posedge clk_50mhz) begin
      entrada[1] = SW[1];
      entrada[0] = SW[0];
   case(estado)
      S1:begin
         HEX1 = 7'b0000001;
         if(entrada == (2'b10))
            estado = S2;
         if(entrada == (2'b01))
            estado = S3;
      end

      S2:begin
         HEX1=7'b0000110;
         if(entrada[0] == 1)
            //DESLIGADO SE HOUVER CONTROLE
            estado = S1;
         if (clk_50mhz) begin
            count_reg = 0;
         end else begin
            if (count_reg < 150000) begin
                count_reg = count_reg + 1;
            end else begin
               count_reg = 0; //DESLIGADO SE TERMINAR O RELÓGIO
```

Imagem 6 - Codigo Etapa 2 Mealy

```
42                estado = S1;
43             end
44          end
45       end
46
47       S3:begin
48             HEX1 = 7'b0100000;
49             if(entrada[0] == 1) //DESLIGADO SE HOUVER CONTROLE
50                estado = S1;
51
52             if (clk_50mhz) begin
53                count_reg = 0;
54             end else begin
55                if (count_reg < 300000) begin
56                    count_reg = count_reg + 1;
57                end else begin
58                   count_reg = 0;//DESLIGADO SE TERMINAR O RELÓGIO
59                   estado = S1;
60                end
61          end
62       end
63
64    endcase
65    end
66
67 endmodule
68
69
```

Imagem 7 - Código etapa 2 Mealy

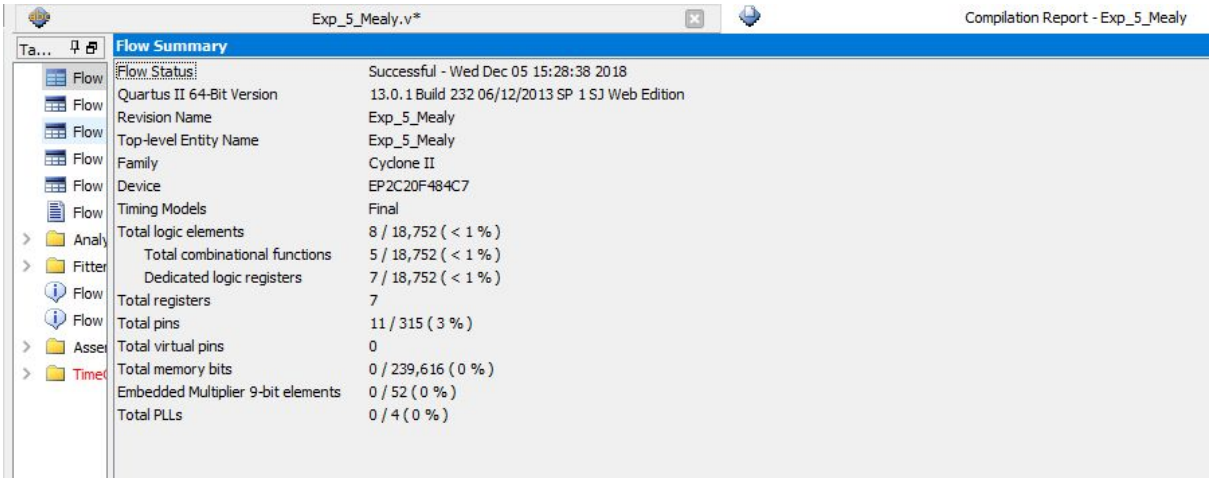

Imagem 8 - Compilação do código em verilog

3 - Apresentação dos Resultados

3.1 - Etapa 1

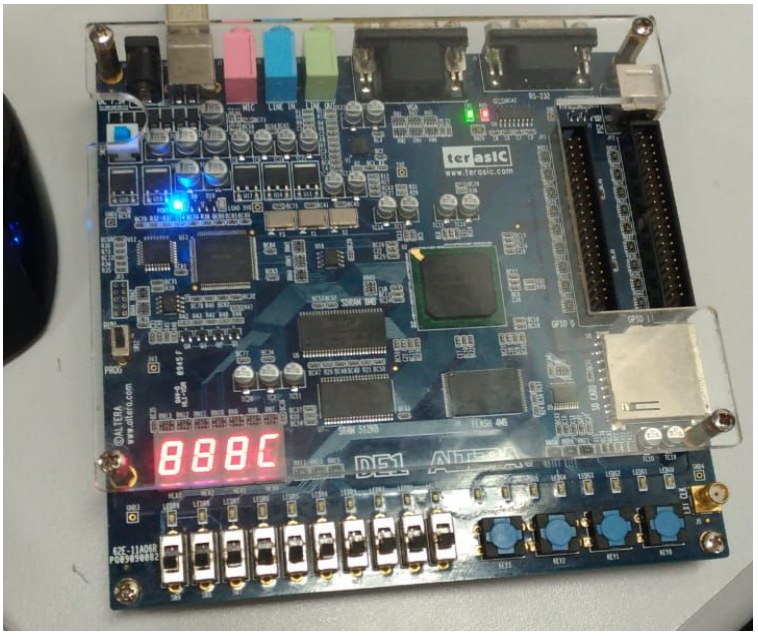

Imagem 9 - Portão fechado (C = Closed)

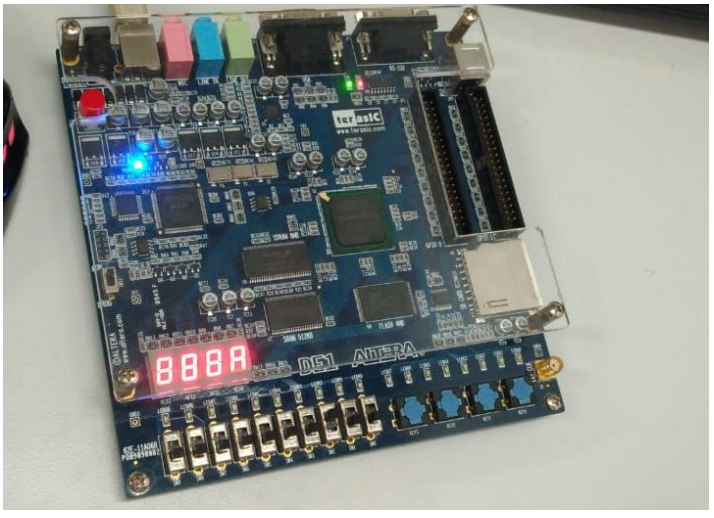

Imagem 10 - Portão abrindo

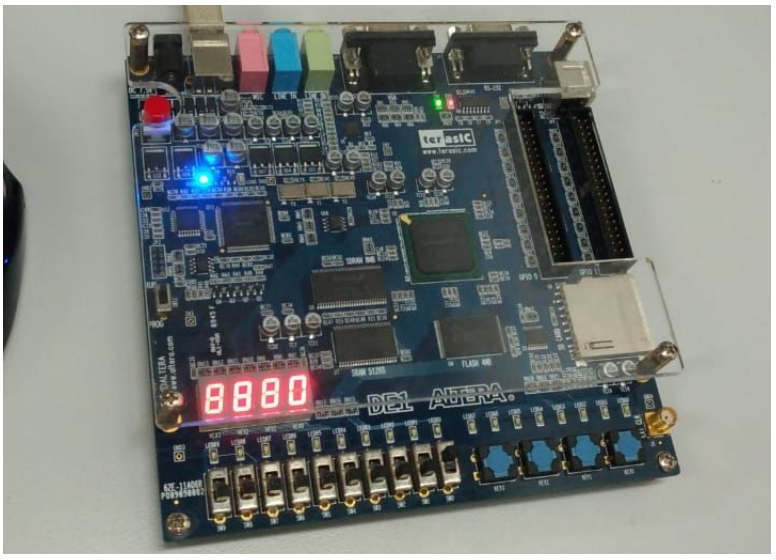

Imagem 11 - Portão aberto (O = open)

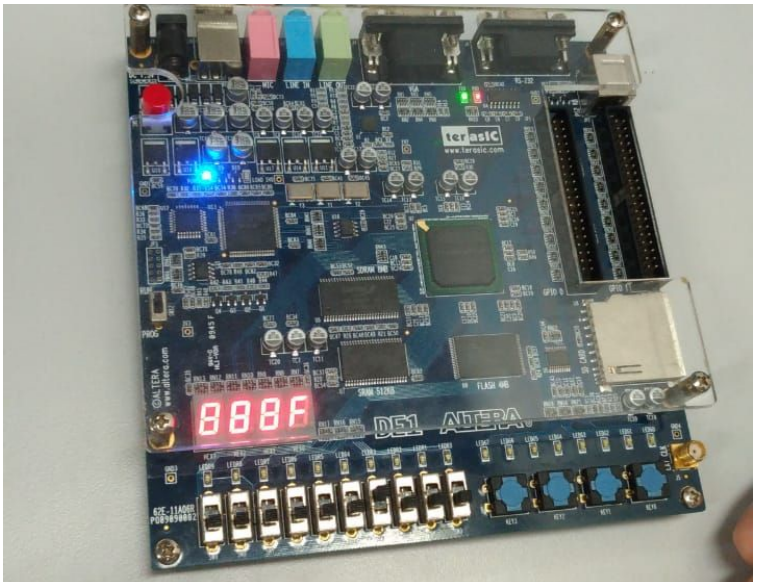

Imagem 12 - Portão fechando

3.1 - Etapa 2

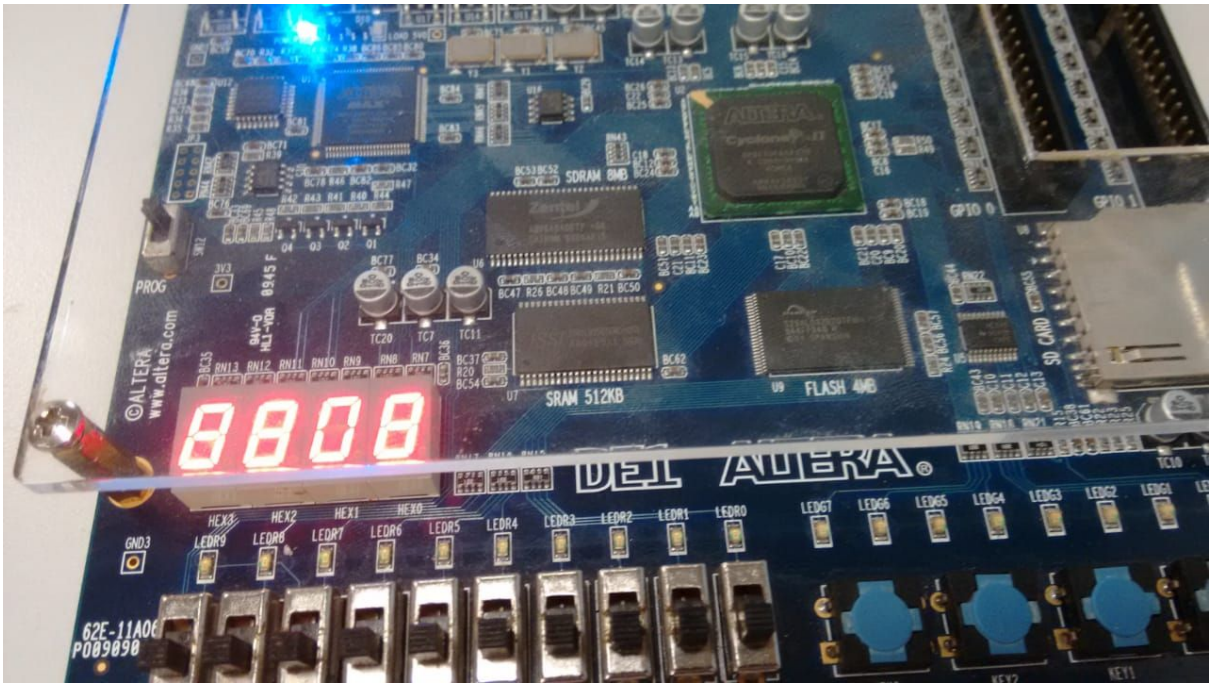

Imagem 13 - Sistema de Irrigação desligado mostrado no HEX1

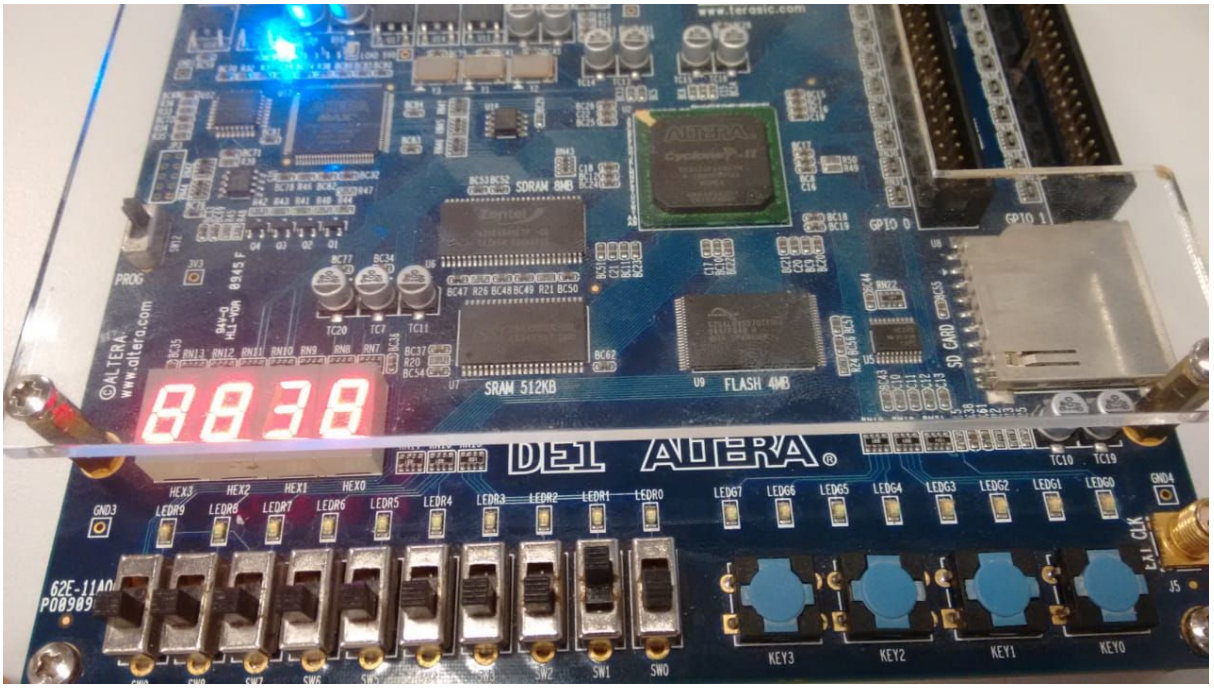

Imagem 14 - Sistema de Irrigação ligado pelo sensor mostrado no HEX1

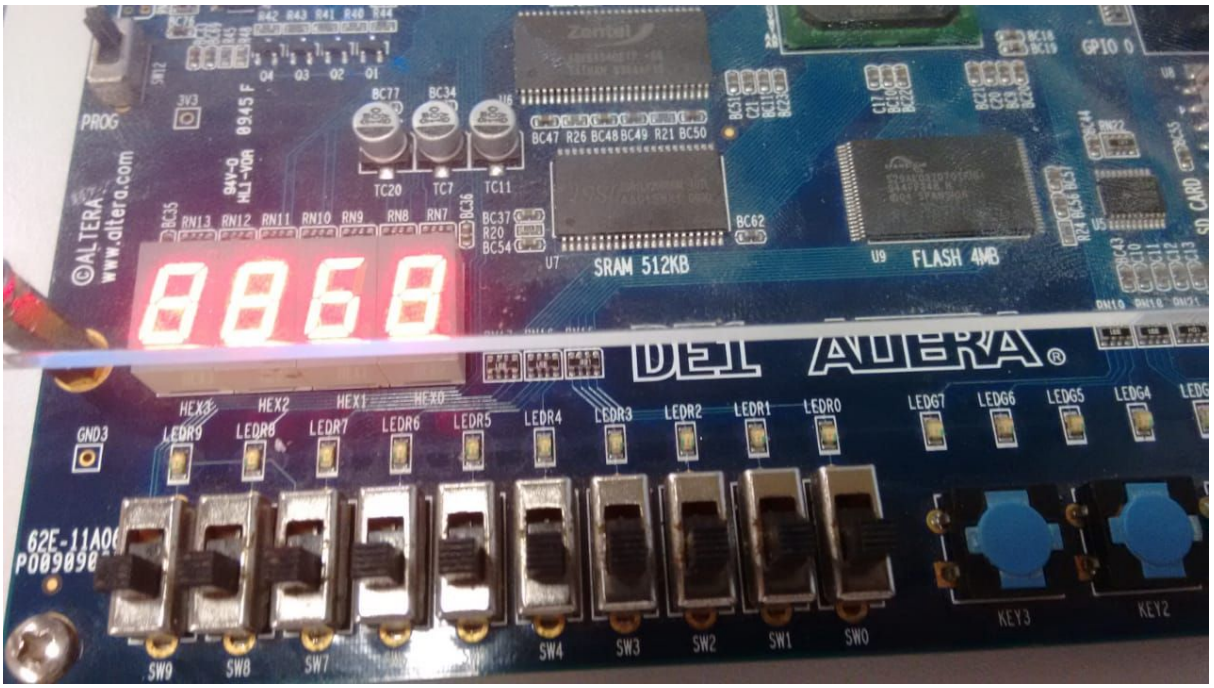

Imagem 15 -Sistema de Irrigação ligado pelo controle mostrado no HEX1

4 - Conclusão

Nesse experimento utilizamos toda nossa bagagem de conhecimento adquirida até agora, cada etapa dos experimentos foi um desafio para nós, desde a construção das máquinas de estado até a sua aplicação na FPGA. De início pensávamos que existiria alguma forma específica de montar a máquina no verilog, logo depois descobrimos que seria um processo bem parecido do qual estamos acostumados no curso. Nesse experimento pudemos entender com clareza as diferenças entre mealy e moore, e suas aplicações, sem dúvidas isso foi de grande importância, gastamos 80% do nosso tempo apenas tentando entender a aplicação e quais complicações e dificuldades seriam encontradas, até chegarmos em uma máquina de estado real e que pudesse ser aplicada de verdade, lógico que respeitando as instruções passadas.