



Universidade Federal de São Carlos - UFSCar

Joao Vitor Azevedo Marciano 743554

Lorhan Sohaky de Oliveira Duda Kondo 740951

Experimento 05 - Implementação de um circuito sequencial utilizando Verilog

São Carlos - SP

2017

Universidade Federal de São Carlos - UFSCar

Joao Vitor Azevedo Marciano 743554

Lorhan Sohaky de Oliveira Duda Kondo 740951

Experimento 05 - Implementação de um circuito sequencial utilizando Verilog

Orientador: Fredy João Valente

Universidade Federal de São Carlos - UFSCar

Departamento de Computação

Ciência da Computação

Laboratório de Circuitos Digitais

São Carlos - SP

2017

Lista de ilustrações

Figura 1 – Ilustração da máquina de estado do problema da garagem.	9
--	---

Lista de tabelas

Lista de quadros

Quadro 1 – Lista das entradas da máquina de estado do problema da garagem. . .	8
Quadro 2 – Significados dos estados relacionando com os estados reais do problema proposto.	9

Lista de abreviaturas e siglas

FPGA	<i>Field Programmable Gate Array</i> - Arranjo de Portas Programáveis em Campo
------	--

Sumário

1	RESUMO	7
2	DESCRIÇÃO DA EXECUÇÃO DO EXPERIMENTO	8
2.1	Passo 1 – Desenhar a máquina de estado	8
2.2	Passo 2 - Escrever um código Verilog para a máquina de estado . .	9
3	AVALIAÇÃO DOS RESULTADOS DO EXPERIMENTO	12
4	ANÁLISE CRÍTICA E DISCUSSÃO	13

1 Resumo

A ideia deste experimento é implementar uma máquina de estado utilizando a linguagem de descrição de *hardware* Verilog. A máquina tenta representar uma situação do mundo real de um portão de garagem.

Considere o cenário de um controle para um portão de garagem. 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 acesso quando o portão se desloca.

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.

Para solucionar tal problema e facilitar sua resolução, dividiu-se o processo em quatro passos:

1. Desenhar a máquina de estado para o cenário em questão;
2. Escrever um código Verilog para a máquina de estado no passo anterior;
3. Executar o código na *Field Programmable Gate Array* - Arranjo de Portas Programáveis em Campo (FPGA) e simulação.

Além disso, escolheu-se uma máquina de estado qualquer para estudar como implementá-la em Verilog.

2 Descrição da execução do experimento

2.1 Passo 1 – Desenhar a máquina de estado

Com o problema em questão, tem-se em mente que trata-se de um portão de garagem que move-se horizontalmente, ou seja, da esquerda para direita e vise e versa. Assim, para a elaboração da máquina¹, considerou-se as entradas conforme o [Quadro 1](#).

Quadro 1 – Lista das entradas da máquina de estado do problema da garagem.

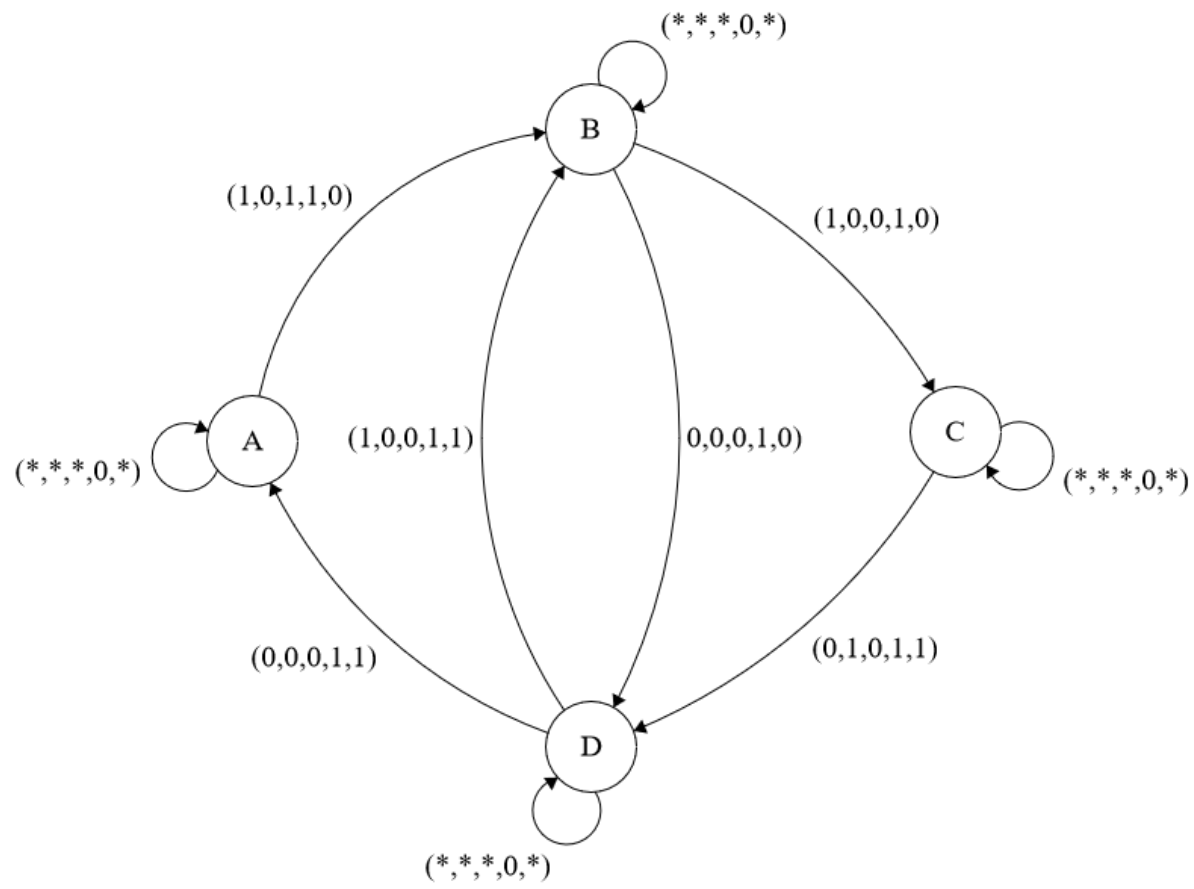
Entrada	Valor lógico 0	Valor lógico 1
Botao (b)	Abrir	Fechar
Aberto (a)	Não aberto	Totalmente aberto
Fechado (f)	Não fechado	Totalmente fechado
Motor (m)	Motor desligado	Motor ligado
Sentido (s)	Esquerda	Direita

Fonte: Próprio Autor

Com as entradas descritas no [Quadro 1](#) elaborou-se a máquina conforme a [Figura 1](#).

¹ Preferiu-se a utilização de uma máquina de Moore por haver conhecimento prévio deste tipo de máquina.

Figura 1 – Ilustração da máquina de estado do problema da garagem.



Nota: A entrada está no formato (b, a, f, m, s). Os significados dos estados estão no [Quadro 2](#).

Quadro 2 – Significados dos estados relacionando com os estados reais do problema proposto.

Estado	Significado
A	Totalmente fechado
B	Abrindo
C	Totalmente aberto
D	Fechando

Fonte: Próprio Autor

2.2 Passo 2 - Escrever um código Verilog para a máquina de estado

Com a máquina de estado pronto, criou-se a código em Verilog, que no estado "totalmente aberto"apresenta A *display*, no estado "totalmente fechado"apresenta F no *display*, no estado 'abrindo' acende um *led* verde e no estado "fechando"acente um *led*

vermelho. Nos estados "abrindo" e "fechando" o *display* apresenta 0. O Verilog encontra-se no [Código 2.1](#).

```
module inicial ( botao, aberto, fechado, motor, sentido, ledVerde,
    ledVermelho, display, clock );
input  botao, aberto, fechado, motor, sentido, clock;
output ledVerde, ledVermelho;
output [6:0] display;

reg [1:0] estado;
reg [4:0] entrada;

reg [6:0] tmpDisplay;
reg tmpLedVerde, tmpLedVermelho;

parameter Fechado = 2'b00, Abrindo = 2'b01, Aberto = 2'b10, Fechando =
    2'b11;

initial estado = Fechado;

always @(posedge clock) begin
    entrada[4] = botao;
    entrada[3] = aberto;
    entrada[2] = fechado;
    entrada[1] = motor;
    entrada[0] = sentido;

    case( estado )
        Fechado: begin
            tmpDisplay = 7'b0001110;
            tmpLedVerde = 0;
            tmpLedVermelho = 0;

            if( entrada == 5'b10110 ) // botao = 1 & aberto = 0
            & fechado = 1 & motor = 1 & sentido = 0
                estado = Abrindo;
        end

        Abrindo: begin
            tmpDisplay = 7'b1000000;
            tmpLedVerde = 1;
            tmpLedVermelho = 0;

            if( entrada == 5'b10010 ) // botao = 1 & aberto = 0
            & fechado = 0 & motor = 1 & sentido = 0
                estado = Aberto;

            if( entrada == 5'b00010 ) // botao = 0 & aberto = 0
            & fechado = 0 & motor = 1 & sentido == 0
```

```

        estado = Fechando;
    end

    Aberto: begin
        tmpDisplay = 7'b0001000;
        tmpLedVerde = 0;
        tmpLedVermelho = 0;

        if( entrada == 5'b01011 ) // botao = 0 & aberto = 1
        & fechado = 0 & motor = 1 & sentido = 1
            estado = Fechando;
        end

    Fechando: begin
        tmpDisplay = 7'b1000000;
        tmpLedVerde = 0;
        tmpLedVermelho = 1;

        if( entrada == 5'b10011 ) // botao = 1 & aberto = 0
        & fechado = 0 & motor = 1 & sentido = 1
            estado = Abrindo;
        if( entrada == 5'b00011 ) // botao = 0 & aberto = 0
        & fechado = 0 & motor = 1 & sentido = 1
            estado = Fechado;
        end

    default: estado = Fechado;

    endcase

end

assign display= tmpDisplay;
assign ledVerde = tmpLedVerde;
assign ledVermelho = tmpLedVermelho;

endmodule

module maquina( SW, LEDG, LEDR, HEX0, CLK );
    input [4:0] SW;
    input CLK;
    output [0:0] LEDG, LEDR;
    output [6:0] HEX0;

    inicial a( SW[4], SW[3], SW[2], SW[1], SW[0], LEDG[0], LEDR[0], HEX0,
    CLK);
endmodule

```

Código 2.1 – Código da máquina de estado do problema da garagem.

3 Avaliação dos resultados do experimento

Apresentar os resultados da simulação em software e da utilização do Kit DE1 e/ou protoboard. Utilizar figuras, descrevê-las e discuti-las.

4 Análise crítica e discussão

Apresentar a visão do grupo sobre o experimento, apresentando pontos fáceis e de dificuldades para a realização do mesmo. Comente se os resultados obtidos representam o comportamento esperado do grupo para o circuito, fazendo relação com o conteúdo teórico.