



UNIVERSIDADE FEDERAL DE SÃO PAULO

NATHÁLIA CRISTINA PIRES VIDAL

RA:139958

**IMPLEMENTAÇÃO POR MÁQUINA DE ESTADOS MEALY:
CONTADOR CRESCENTE E DECRESCENTE**

SÃO JOSÉ DOS CAMPOS, JANEIRO 2023

IMPLEMENTAÇÃO POR MÁQUINA DE ESTADOS MEALY: CONTADOR CRESCENTE E DECRESCENTE

Relatório apresentado à Universidade Federal de São Paulo como parte dos requisitos para aprovação na disciplina de Laboratório de Sistemas Computacionais: Circuitos Digitais.

Docente: Prof. Dr. Lauro Paulo da Silva Neto
Universidade Federal de São Paulo- UNIFESP
Instituto de Ciência e Tecnologia - Campus São José dos Campos-SP

SÃO JOSÉ DOS CAMPOS, JANEIRO/2023

RESUMO

Máquina de estados finitos são autômatos finitos que produzem uma saída de dados baseando-se no estado em que se encontram e na entrada de dados. Neste projeto, utilizou-se uma máquina de estados finitos do tipo Mealy. A máquina de estados Mealy possui dez estados e alterna entre eles através de quatro funções: estado atual, próximo estado, estado anterior e display apagado. A máquina percorre a sequência de números 0-8-2-4-5-4-9-0-3, a qual foi mostrada em um display de sete segmentos. O tempo utilizado é manipulado através de um divisor de frequências, o qual atrasa o sinal do clock. A implementação foi feita no Software Quartus na linguagem de descrição de Hardware Verilog VHDL e carregada na placa FPGA DE2-215.

Palavras-chaves: Máquina de estados, Mealy, Display de Sete Segmentos, Verilog, Quartus, FPGA DE2-215.

LISTA DE ILUSTRAÇÕES

Figura 1 - Estrutura Verilog.....	9
Figura 2 - Máquina de Estados Finitos.....	9
Figura 3 - Modelo Máquina Mealy	10
Figura 4 - Representação Display de Sete Seguintos.....	13
Figura 5 - Decodificador BCD em Verilog	13
Figura 8 - Circuito Final do contador.....	15

LISTA DE TABELAS

Tabela 1 - Tabela verdade para o Display de 7 segmentos.....	13
Tabela 2 - Tabela-verdade máquina de estados	14
Tabela 3 - Transição de estados	14

SUMÁRIO

1. INTRODUÇÃO.....	7
2. OBJETIVOS	8
2.1 Gerais:	8
2.2 Específicos:	8
3. FUNDAMENTAÇÃO TEORICA	9
3.1 Verilog- VHDL	9
3.2 Máquina de estados finitos.....	9
3.2 Divisor de frequência	10
3.3 Decodificador bcd	10
4 DESENVOLVIMENTO DO TRABALHO	12
4.1 Divisor de frequência	12
4.2 Decodificador BCD	12
4.3 Máquina de Estados.....	14
4.4 Circuito Final	15
5. RESULTADOS OBTIDOS E DISCUSSÃO	16
6. CONSIDERAÇÕES FINAIS.....	16
REFERÊNCIAS	17

1. INTRODUÇÃO

Atualmente as novas tecnologias estão presentes no dia a dia de todos, sendo utilizadas para diversas funcionalidades. Partindo desse atual contexto histórico, temos os circuitos digitais, o qual possui diversas vantagens e serve como base para diversos estudos principalmente na área da computação.

Nesse projeto trabalhamos com a linguagem de descrição de Hardware para aplicarmos o conceito de máquina de estados, nesse caso, a máquina de estados Mealy, a qual possui dez estados. Para esse projeto, utilizaremos: Um divisor de frequência e um decodificador BCD para display de sete segmentos, os quais serão explicadas ao decorrer deste relatório, formando-se assim um contador que percorrerá uma sequência de forma crescente e decrescente.

2. OBJETIVOS

2.1 Gerais:

Desenvolver um contador para a sequência 0-8-2-4-5-4-9-0-3 através de uma máquina de estados do tipo Mealy, com um tempo de transição de um segundo, que possua três sinais de entrada *up*, *down* e *reset*. Utilizando um desses sinais, ou a combinação deles devemos fazer com que a sequência seja apresentada em um display de sete segmentos, podendo manter o estado, ir para o estado seguinte, ir para o estado anterior e display desligado.

2.2 Específicos:

- Implementar um divisor de frequências responsável por converter o sinal do clock de 50,0 MHz para 1,0 Hz, fazendo com que um ciclo ocorra a cada um segundo;
- Projetar o circuito da Máquina de Estados;
- Implementar um decodificador de saída BCD;
- Projetar o circuito da Máquina de Estados;
- Combinar os circuitos produzidos;
- Analisar os resultados.

3. FUNDAMENTAÇÃO TEORICA

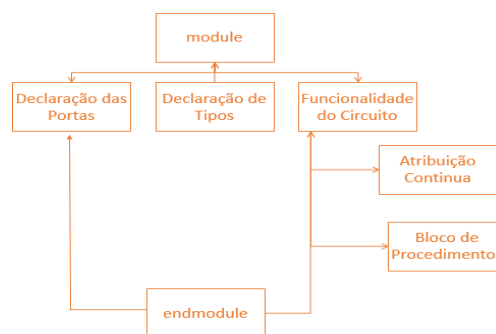
3.1 Verilog- VHDL

Verilog é uma linguagem de descrição de Hardware que proporciona ao programador, através de uma sintaxe, descrever um sistema digital por meio de um arquivo de texto. Esse arquivo pode ser implementado em um dispositivo programável, como as placas FPGAs.

Além de a linguagem Verilog simplificar os circuitos digitais, ela permite a reutilização de bibliotecas e seu código é compatível a diversos dispositivos, ou seja, seu código independe do Hardware.

Sua estrutura fica dentro do cabeçalho e rodapé que são o *module*, iniciação do código, e o *endmodule*, que é o fechamento do código. Após o *module* é feita a Declaração das portas, ou seja, é colocada todas as entradas e saídas. Após isso, é feita a Declaração de Tipos, podendo ser *wire* ou *reg*. E por fim a Funcionalidade do Circuito podendo ser uma atribuição continua, ou seja, inserida constantemente dentro de uma variável ou um bloco de procedimentos. O código é finalizado com o *endmodule*. Podemos observar na Figura 1 a estrutura da linguagem.

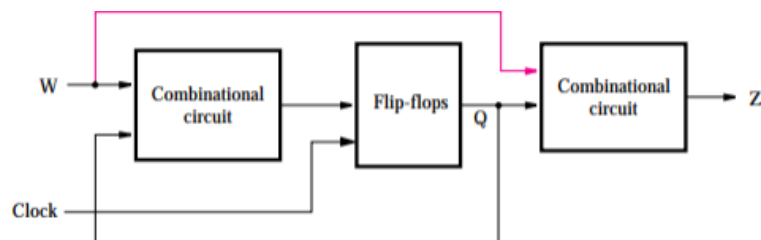
Figura 1 - Estrutura Verilog



3.2 Máquina de estados finitos

Máquina de Estados Finitos é uma máquina abstrata que possui um número finito de estados. A máquina está em apenas um estado por vez, chamado de estado atual, podendo fazer transições. Essas transições são mudanças de estado, armazenando sempre informações de entrada e saída.

Figura 2 - Máquina de Estados Finitos



Fonte: (FLOYD)(2)

A Máquina de Mealy é uma máquina de estados que produz um resultado baseado no estado em que se encontra e na entrada de dados, ou seja, a entrada também influencia diretamente o valor de saída.

Figura 3 - Modelo Máquina Mealy

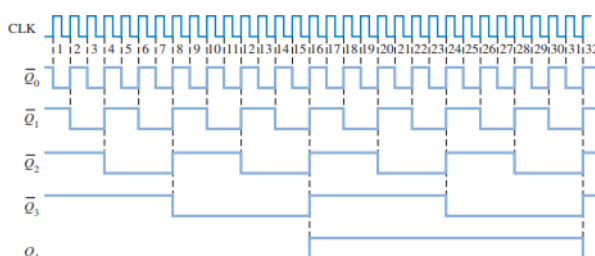


Fonte: Autor

3.2 Divisor de frequência

Um divisor de frequência possui a função de dividir a frequência fundamental de entrada por dois, ou seja, a frequência cai pela metade, como podemos observar na figura:

Figura 3- Forma de onda de um Divisor de Frequência



Fonte: Sistemas Digitais: Fundamentos e Aplicações, p. 473 (1)

3.3 Decodificador bcd

Decodificador bcd é um decodificador que transforma um número binário que representa um certo estado, em um número decimal a ser projetado no display de sete segmentos, como mostrado na Figura 3 a seguir:

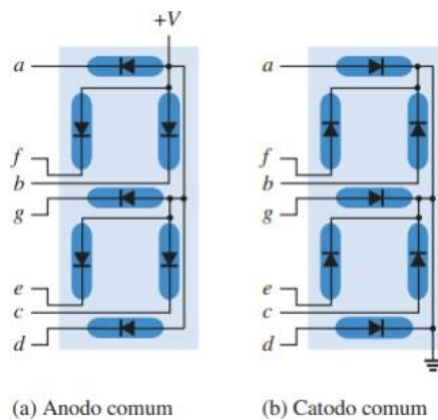
Figura 4- Bloco decodificador de 7 segmentos



Fonte: João Marcos de Aguiar

Esse decodificador é ligado ao display de 7 segmentos, o qual é um grupo de LEDs que em conjunto são capazes de formar letras e números de acordo com as entradas. O display pode ser do tipo cátodo ou ânodo comum, o que significa que ou os LEDs estão com os terminais positivos em comum ou os negativos, portanto para cada tipo, o valor de entrada para ligá-los é diferente, como observado na *Figura 5*.

Figura 5- Display de 7 segmentos.



Fonte: Sistemas Digitais (1)

4 DESENVOLVIMENTO DO TRABALHO

A proposta do projeto é construir um contador que funciona de forma crescente e decrescente e percorresse por uma sequência destinada a cada aluno, de forma individual, através de uma Máquina de Estados Finitos desenvolvido em Verilog no software Quartus Prime.

A sequência a ser percorrida é 0-8-2-4-5-4-9-0-3 e o tipo de máquina de estados é a Mealy. Para melhor compreensão e entendimento, o projeto foi dividido em: Projetar um divisor de frequência para a contagem de 1 segundo, projetar um decodificador BCD para o Display de 7 segmentos e projetar uma Máquina de Estados.

4.1 Divisor de frequência

O divisor de frequência deverá atrasar o sinal do clock recebido, a cada segundo, ou seja, a cada 1 hertz, possibilitando assim, a manipulação do tempo da troca de estados. A saída terá o nível alto “1” após 1 segundo ou 1 Hz, portanto é necessário que se utilize um divisor de frequências.

O divisor de frequência precisará de 26 bits para contar 50 milhões de ciclos de relógio, que corresponde a um segundo.

Figura 6- Divisor de frequência

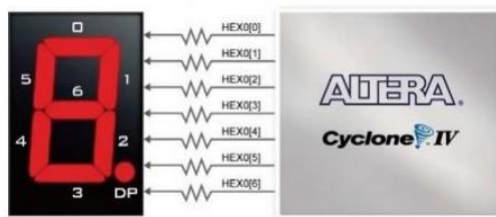
```
27 module divisor_freq_1 (clk, saida_divisor);
28     input clk;
29     output reg saida_divisor;
30     reg [25:0] out_divisor;
31
32     always @ (posedge clk)
33     begin
34         if (out_divisor == 26'd50000000 )
35         begin
36             out_divisor <= 26'd0;
37             saida_divisor <= 1;
38         end
39     else
40     begin
41         out_divisor <= out_divisor+1;
42         saida_divisor <= 0;
43     end
44 end
45 endmodule
46
47
48
```

Fonte: Autor.

4.2 Decodificador BCD

Para esse projeto, devemos projetar um circuito que satisfaça a projeção de números de 0 a 9 no Display de 7 segmentos, transformando assim, os 4 bits de entrada em um número decimal a ser representado. Assim como representado na Figura 11, a seguir:

Figura 4 - Representação Display de Sete Seguimentos



Fonte: Manual DE2-115 (3).

Assim, teremos a seguinte tabela verdade:

Tabela 1 - Tabela verdade para o Display de 7 segmentos

ENTRADAS				SAÍDA DISPLAY 7 SEGMENTOS							NÚMERO
W	X	Y	Z	a	b	c	d	e	f	g	-
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	2
0	0	1	1	1	1	1	1	0	0	1	3
0	1	0	0	0	1	1	0	0	1	1	4
0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	0	1	1	1	1	1	6
0	1	1	1	1	1	1	0	0	0	0	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	1	1	0	1	1	9

Fonte: Autor.

A partir da tabela-verdade acima podemos codificar o Decodificador para display de sete segmentos. O Decodificador pegará os quatro bits de entrada (W,X,Y,Z) e transformará em números, as saídas serão 7 bits (a,b,c,d,e,f,g) onde os bits colocados em “1” terão seus leds apagados, por conta de ser do tipo anodo comum, ou seja, todos os segmentos que receberem valor zero terá o seu led aceso. Além disso, para números maiores que 9, o display apaga, assim todas as saídas estarão em “1”.

Figura 5 - Decodificador BCD em Verilog

```

249
250 module decodificador_BCD_1(BOX, HEX4);
251     input [3:0]BOX;
252     output [6:0]HEX4;
253     reg [6:0]segmentos;
254     always @ (*)
255     begin
256         case (BOX)
257             4'b0000: segmentos=7'b0000001; //0
258             4'b0001: segmentos=7'b1001111; //1
259             4'b0010: segmentos=7'b0010010; //2
260             4'b0011: segmentos=7'b0000110; //3
261             4'b0100: segmentos=7'b1001100; //4
262             4'b0101: segmentos=7'b0100100; //5
263             4'b0110: segmentos=7'b0100000; //6
264             4'b0111: segmentos=7'b0001111; //7
265             4'b1000: segmentos=7'b0000000; //8
266             4'b1001: segmentos=7'b0000100; //9
267             default: segmentos = 7'b1111111; //display apagado
268         endcase
269     end
270     assign HEX4 = segmentos;
271 endmodule
272
273

```

Fonte: Autor

4.3 Máquina de Estados

A máquina de estados deve conter dez estados, onde nove serão representados por números decimais, e o décimo estado será o display apagado. A mudança de estados é coordenada por duas entradas denominadas UP e DOWN, como mostrado a seguir:

Tabela 2 - Tabela-verdade máquina de estados

UP	DOWN	SAÍDA
0	0	MANTER ESTADO
0	1	DECRESCENTE
1	0	CRESCENTE
1	1	APAGAR DISPLAY

Fonte: Autor

Com base na tabela-verdade, temos que gerar uma tabela que abranja todas as possibilidades de transição.

Tabela 3 - Transição de estados

ESTADO	ESTADO ATUAL				PRÓXIMO ESTADO				SAÍDA
	E3	E2	E1	E0	U,D = 0,0	U,D = 0,1	U,D = 1,0	U,D = 1,1	
A	0	0	0	0	0,0,0,0	1,0,0,0	0,0,0,1	1,0,0,1	0
B	0	0	0	1	0,0,0,1	0,0,0,0	0,0,1,0	1,0,0,1	8
C	0	0	1	0	0,0,1,0	0,0,0,1	0,0,1,1	1,0,0,1	2
D	0	0	1	1	0,0,1,1	0,0,1,0	0,1,0,0	1,0,0,1	4
E	0	1	0	0	0,1,0,0	0,0,1,1	0,1,0,1	1,0,0,1	5
F	0	1	0	1	0,1,0,1	0,1,0,0	0,1,1,0	1,0,0,1	4
G	0	1	1	0	0,1,1,0	0,1,0,1	0,1,1,1	1,0,0,1	9
H	0	1	1	1	0,1,1,1	0,1,1,0	1,0,0,0	1,0,0,1	0
I	1	0	0	0	1,0,0,0	0,1,1,1	0,0,0,0	1,0,0,1	3
APAGADO	1	0	0	1	1,0,0,1	0,0,0,0	0,0,0,0	1,0,0,1	OFF

Fonte: Autor.

Com a tabela-verdade de transição de estados podemos desenvolver nossa máquina de estados. Usaremos os fios de entrada *UP* e *DOWN* para comparar e seguir a sequência numérica. Quando *UP* e *Down* forem “0” a máquina deve manter o número então o próximo número será ele mesmo e a variável “saída” receberá o número em decimal. Quando *UP* for “0” e *DOWN* for “1” a contagem deverá ser de forma decrescente, assim a variável “prox” receberá o número anterior e seguirá a sequência assim. Quando *UP* for “1” e *DOWN* for “0” a contagem deverá ser de forma crescente então a variável “prox” receberá o próximo número da sequência e seguirá a sequência dessa forma. Quando *UP* e *DOWN* forem “1” o display deverá ser completamente apagado. Deveremos fazer essas quatro comparações para todos os casos, ou seja, para todos os números da sequência, que são os dez estados, para que não haja erro na contagem. Após isso, dentro do programa principal é chamado o decodificador para que a contagem seja feita no Display de sete segmentos.

4.4 Circuito Final

Logo, a circuito final será:

Figura 6 - Circuito Final do contador

```
// Máquina de Mealy

module maquina_mealy (HEX4, SW, CLOCK2_50);

input [17:0] SW;
input CLOCK2_50;
output [0:6] HEX4;
wire reset, UP, DOWN;
wire OUT;

// variaveis do tipo registrador, armazena um valor e pode altera-lo a qualquer momento
reg [6:0] segmentos;
reg [3:0] estadoAtual;
reg [3:0] prox;
reg [3:0] saida;

// fios de entrada
assign reset = SW[17];
assign UP = SW[1];
assign DOWN = SW[0];

parameter est0 = 4'd0, est8 = 4'd1, est2 = 4'd2, est4 = 4'd3, est5 = 4'd4,
est41 = 4'd5, est9 = 4'd6, est01 = 4'd7, est3 = 4'd8, nulo = 4'd9;

// Chamando o Divisor de Frequência
divisor TX(.CLK (CLOCK2_50), .SAIDA(OUT));

always @(UP or DOWN or estadoAtual) begin

    case(estadoAtual)
```

```
case(estadoAtual)

// 0 8 2 4 5 4 9 0 3

// Primeiro Estado
est0: if (UP==0 & DOWN==0) //manter o estado
begin
    prox <= est0;
    saida = 4'd0; //0
end
else if (UP==0 & DOWN==1) //decrecente
begin
    prox <= est3;
    saida = 4'd3; //3
end
else if (UP==1 & DOWN==0) //crescente
begin
    prox <= est8;
    saida = 4'd8; //8
end
else begin
    prox <= nulo;
    saida = 4'd15; //15
end

// Segundo Estado
est8: if (UP==0 & DOWN==0) //manter o estado
begin
    prox <= est8;
    saida = 4'd8; //8
end
else if (UP==0 & DOWN==1) //decrecente
begin
```

Fonte: Autor.

5.RESULTADOS OBTIDOS E DISCUSSÃO

As etapas durante o projeto foram testadas no ambiente virtual fornecido pela UNIFESP. Podemos observar que os resultados ocorreram como o esperado na contagem crescente e decrescente percorrendo pela sequência numérica pré estabelecida, além de manter um número e apagar o display.

O contador implementado pela máquina de estados Mealy funcionou fazendo a contagem em um segundo tanto na interação crescer e decrescer, demonstrando a sequência no display HEX 0, seguindo o planejado para o projeto.

6.CONSIDERAÇÕES FINAIS

Com o término do trabalho podemos observar que o conceito de máquina de estados pode ser muito útil na resolução de alguns problemas. Sua implementação através da linguagem Verilog VHDL é menos complexa e menos sujeita a erros.

O projeto foi descarregado na placa Intel FPGA e mostrado ao professor na bancada.

REFERÊNCIAS

1. 32x8.com. (s.d.). Fonte: ONLINE Karnaugh map solver with circuit for up to 8 variables.2014: 32x8.com
2. BROWN, S. D., & VRANESIC, Z. (2009). *Fundamentals of Digital Logic with VHDL Desgin*. New York: McGraw-Hill.
3. DE2-115. (s.d.). Fonte: User Manual:
https://grad.sead.unifesp.br/pluginfile.php/371633/mod_resource/content/1/DE2-115%20Manual.pdf
4. FLOYD, T. L. (s.d.). *Sistemas Digitais: Fundamentos e Aplicações. 9a.*
5. Neto, P. D. (s.d.). *Aula 1 Verilog- Linguagem de*. Fonte: Aula 1 Verilog- Linguagem de.