

Fernanda Silva Bucheri RA: 135529

**Projeto de um contador crescente/decrecente
utilizando uma máquina de estados finitos de
Mealy**

São José dos Campos - Brasil

Fevereiro de 2021

Fernanda Silva Bucheri RA: 135529

Projeto de um contador crescente/decrecente utilizando uma máquina de estados finitos de Mealy

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

Docente: Prof^ª. Dr^ª.: Fernanda Quelho Rossi

Universidade Federal de São Paulo - UNIFESP

Instituto de Ciência e Tecnologia - Campus São José dos Campos

São José dos Campos - Brasil

Fevereiro de 2021

Resumo

Este projeto visa implementar uma máquina de estados do tipo Mealy comportando-se como um contador capaz de realizar uma contagem crescente ou decrescente da seguinte sequência numérica não ordenada: 9 - 4 - 6 - 5 - 8 - 2 - 1 - 0 - 5. Utilizou-se o software *Intel Quartus Prime* para elaboração do diagrama esquemático do circuito digital e o Kit de Desenvolvimento FPGA (*Field-programmable gate array*) DE2-115 da ALTERA como ambiente de testes para a implementação real do projeto: de acordo com o comando dado pelo usuário, será exibido em um display de 7 segmentos a contagem crescente, decrescente, irá manter um número no display ou apagá-lo.

Palavras-chaves: Circuitos Digitais, Máquina de Estados, Mealy.

Lista de ilustrações

Figura 1 – Portas lógicas e seus símbolos, tabela verdade e expressão booleana correspondentes.	14
Figura 2 – Algumas identidades da álgebra de Boole.	15
Figura 3 – Latch que utiliza portas NAND e sua tabela verdade.	16
Figura 4 – Latch que utiliza portas NOR e sua tabela verdade.	16
Figura 5 – Flip-flop S-R com clock que responde apenas à borda de subida do pulso de clock e sua tabela verdade.	17
Figura 6 – Circuito interno de um flip-flop S-R disparado por borda.	17
Figura 7 – Flip-flop J-K com clock que responde apenas às bordas positivas do clock e sua tabela verdade.	17
Figura 8 – Circuito interno de um flip-flop J-K disparado por borda.	18
Figura 9 – Diagrama em bloco das máquinas de estado.	19
Figura 10 – Placa DE2-115	20
Figura 11 – Displays de sete segmentos presentes na placa.	21
Figura 12 – Implementação completa do contador: o circuito da máquina de Mealy, o circuito divisor de frequência e decodificador BCD para display de 7 segmentos.	23
Figura 13 – Tabela-verdade para o processo de contagem.	23
Figura 14 – Esquema de um divisor de frequência.	24
Figura 15 – Circuito completo do divisor de frequência.	25
Figura 16 – Circuito do decodificador BCD para display de 7 segmentos (parte I).	27
Figura 17 – Circuito do decodificador BCD para display de 7 segmentos (parte II).	27
Figura 18 – Diagrama de estados.	28
Figura 19 – Circuito completo da máquina de Mealy.	30
Figura 20 – Mais detalhes do circuito da máquina de Mealy: as entradas, uma black box que calcula os próximos estados e uma black box com a memória.	31
Figura 21 – Mais detalhes do circuito da máquina de Mealy: circuito que gera a saída S0.	31
Figura 22 – Mais detalhes do circuito da máquina de Mealy: circuito que gera a saída S1.	32
Figura 23 – Mais detalhes do circuito da máquina de Mealy: circuito que gera a saída S2.	32
Figura 24 – Mais detalhes do circuito da máquina de Mealy: circuito que gera a saída S3.	33
Figura 25 – Circuito da black box "ProximoEstado".	33
Figura 26 – Circuito da black box "zeroprox".	34

Figura 27 – Circuito da black box "umprox".	34
Figura 28 – Circuito da black box "doisprox".	35
Figura 29 – Circuito da black box "tresprox".	35
Figura 30 – Circuito da black box "Memoria".	36
Figura 31 – Waveform da simulação que utiliza as entradas $UP = 1$, $DOWN = 0$ (ou seja, uma contagem crescente) e reset ativado aos 250 ns e se mantendo por 250 ns.	37
Figura 32 – Waveform da simulação de uma contagem crescente. Entretanto, aos 200 ns, $DOWN = 1$, mudando para a função que apaga o display. O display fica apagado por 200 ns. O reset é ativado aos 400 ns e permanece ativado por 400 ns.	37
Figura 33 – Waveform da simulação que utiliza as entradas $UP = 0$, $DOWN = 1$ (ou seja, uma contagem decrescente) e reset ativado aos 250 ns e se mantendo por 250 ns.	38
Figura 34 – Waveform da simulação de uma contagem decrescente. Entretanto, aos 200 ns, $UP = 1$, mudando para a função que apaga o display. O reset é ativado aos 400 ns.	38
Figura 35 – Waveform da simulação onde $DOWN = 1$, UP varia de 150 ns em 150 ns e reset varia de 300 ns em 300 ns.	39
Figura 36 – Waveform da simulação onde $DOWN = 0$ e UP varia de 30 ns em 30 ns.	39
Figura 37 – Waveform da simulação onde UP varia de 30 ns em 30 ns e $DOWN$ varia de 90 ns em 90 ns.	39
Figura 38 – Waveform da simulação onde está ocorrendo uma contagem crescente, aplica-se o reset e, após isso, ocorre uma contagem decrescente.	40
Figura 39 – Waveform da simulação do decodificar BCD para display de 7 segmentos (I).	40
Figura 40 – Waveform da simulação do decodificar BCD para display de 7 segmentos (II).	41

Lista de tabelas

Tabela 1 – Tabela-verdade do decodificador BCD.	26
Tabela 2 – Tabela-verdade dos circuitos combinacionais de entrada e saída.	29

Sumário

1	INTRODUÇÃO	9
2	OBJETIVOS	11
2.1	Objetivos Gerais	11
2.2	Objetivos específicos	11
3	FUNDAMENTAÇÃO TEÓRICA	13
3.1	Variáveis booleanas	13
3.2	Tabelas-verdade	13
3.3	Operações básicas e portas lógicas	13
3.4	Álgebra booleana e Leis de Morgan	15
3.5	Dispositivos lógicos biestáveis (latches e flip flops)	15
3.6	Máquina de estados de Mealy	18
3.7	Placa FPGA, Display de 7 segmentos e Conversor BCD	19
4	DESENVOLVIMENTO	23
4.1	Contador completo	23
4.2	Divisor de frequência	24
4.3	Decodificador BCD para Display de 7 segmentos	25
4.4	Diagrama de estados, tabela verdade e expressões booleanas	27
4.5	Máquina de Mealy	30
4.6	Próximo estado	33
4.7	Memória	35
5	RESULTADOS OBTIDOS	37
5.1	Simulação no software Quartus (waveforms)	37
5.2	Simulação no kit FPGA remoto	41
6	CONCLUSÃO	43
	REFERÊNCIAS	45

1 Introdução

Circuitos digitais são essenciais na vida moderna. Possuem inúmeras aplicações que facilitam e proporcionam benefícios para a humanidade. Computadores, aparelhos de celular, Blu-ray, equipamentos hospitalares e muitos outros, são exemplos de aparelhos que baseiam parte do seu funcionamento em circuitos digitais.

O presente trabalho visa abordar mais profundamente diferentes conceitos que englobam circuitos digitais com uma aplicação que consiste em elaborar uma máquina de estados finita (*Finite State Machine*), mais especificamente, uma máquina de Mealy.

2 Objetivos

2.1 Objetivos Gerais

Desenvolver uma máquina de estados de Mealy que possui 10 estados com quatro funções diferentes, sendo elas: agir como um contador crescente ou decrescente da seguinte sequência: 9-4-6-5-8-2-1-0-5, manter seu estado atual ou não mostrar nada no display. O tempo de transição de um estado para o outro deve ser de 1 segundo, aproximadamente.

2.2 Objetivos específicos

- Criar um diagrama de estados com todos os estados que serão implementados na máquina de Mealy;
- Elaborar as tabelas verdades com as funções de próximo estado e de saída;
- Gerar os mapas de Karnaugh, necessários obter as expressões lógicas e, a partir delas, transformar as tabelas verdades nos seus circuitos lógicos correspondentes;
- Implementar todo o circuito da máquina de Mealy;
- Implementar o circuito divisor de frequência para gerar um clock de 1 segundo a partir do clock interno de 50 MHz do kit FPGA;
- Implementar o circuito decodificador BCD para display de sete segmentos, com o propósito de exibir a contagem no display HEX[3] do kit FPGA;
- Analisar os resultados obtidos através das Waveforms e implementar o circuito no kit FPGA;

3 Fundamentação teórica

A primeira descrição de um sistema numérico binário foi apresentada por volta do século III a.C., pelo matemático indiano Pingala. Já Gottfried Leibniz, publicou no século XVIII seu artigo "Explication de l'Arithmétique Binaire", apresentando o sistema numérico binário moderno, utilizando 0 e 1, tal como nos dias de hoje. George Boole, em 1854, publicou um artigo detalhando um sistema lógico, que ficou conhecido como Álgebra Booleana. Em 1937, foi produzida uma tese por Claude Shannon intitulada "A Symbolic Analysis of Relay and Switching Circuits", que implementava álgebra booleana e aritmética binária utilizando circuitos elétricos pela primeira vez, dando início assim ao projeto de circuitos digitais.

Para se projetar um circuito, podemos utilizar diversas ferramentas, sendo elas: álgebra booleana (de suma importância para se obter uma determinada relação entre as entradas e saídas e para simplificação de circuitos), tabelas-verdade, diagramas, entre outras.

3.1 Variáveis booleanas

Existem apenas dois valores possíveis para uma variável booleana assumir, os quais podemos denominar “verdadeiro” ou “falso”, “alto” ou “baixo”, “0” ou “1”.

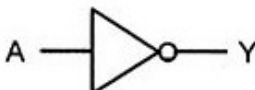






3.2 Tabelas-verdade

São utilizadas no estudo da lógica. Lista-se todos os possíveis valores de entradas e se obtém a saída correspondente para cada combinação. É indispensável para circuitos digitais.

3.3 Operações básicas e portas lógicas

Existem três portas lógicas básicas. São elas: “OR”, “AND” e “NOT” (complementação). É possível representar todas as funções Booleanas em termos destas operações básicas. A partir destas, surgem outras portas lógicas, tais quais: NOT, AND, OR, NOR, NAND, XOR e XNOR.

Figura 1 – Portas lógicas e seus símbolos, tabela verdade e expressão booleana correspondentes.

Função lógica	Símbolo lógico	Tabela verdade	Expressão booleana															
Porta NOT - Inversora		<table><tr><td>A</td><td>Y</td></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	Y	0	1	1	0	$Y = \bar{A}$									
A	Y																	
0	1																	
1	0																	
Porta AND		<table><tr><td>A</td><td>B</td><td>Y</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Y	0	0	0	0	1	0	1	0	0	1	1	1	$Y = A \bullet B$
A	B	Y																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
Porta NAND		<table><tr><td>A</td><td>B</td><td>Y</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Y	0	0	1	0	1	1	1	0	1	1	1	0	$Y = \overline{A \bullet B}$
A	B	Y																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
Porta OR		<table><tr><td>A</td><td>B</td><td>Y</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	1	$Y = A + B$
A	B	Y																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
Porta NOR		<table><tr><td>A</td><td>B</td><td>Y</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	0	$Y = \overline{A + B}$
A	B	Y																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
Porta XOR		<table><tr><td>A</td><td>B</td><td>Y</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	0	$Y = A \oplus B$
A	B	Y																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
Porta XNOR		<table><tr><td>A</td><td>B</td><td>Y</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	1	$Y = \overline{A \oplus B}$
A	B	Y																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

Fonte: tecnicas¹

¹ Disponível em: <https://tecnicas.com/como-descrever-um-circuito-logico-algebricamente/>; Acesso em Fev.2021.

3.4 Álgebra booleana e Leis de Morgan

Na Fig 2, temos algumas propriedades da álgebra de Boole.

Figura 2 – Algumas identidades da álgebra de Boole.

Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\bar{A} = 0$	$A + \bar{A} = 1$
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$
De Morgan's law	$\overline{AB} = \bar{A} + \bar{B}$	$\overline{A + B} = \bar{A}\bar{B}$

Fonte: Site²

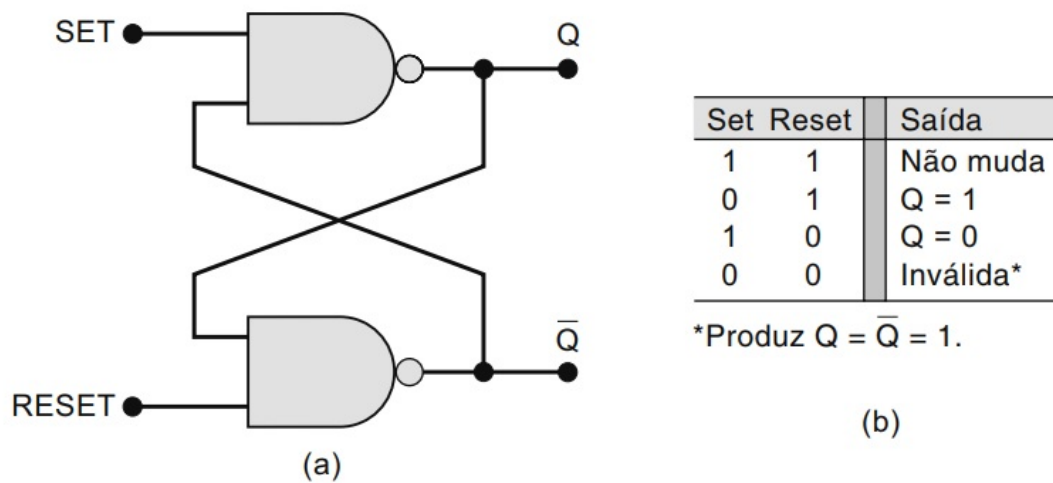
Na última linha observa-se as Leis de Morgan, formuladas por Augustus de Morgan, matemático e lógico indiano. Essas duas leis são de extrema utilidade na simplificação de expressões nas quais uma soma ou um produto de variáveis aparecem negados.

3.5 Dispositivos lógicos biestáveis (latches e flip flops)

Utilizando os conceitos citados acima, podemos desenvolver dispositivos lógicos biestáveis (possuem dois estados estáveis: SET e RESET) que são muito utilizados como dispositivos de armazenamento. Exemplos destes dispositivos são os latches e os flip flops, sendo o flip flop essencial para construção de contadores, registradores e outras lógicas de controle sequencial além de ser utilizado em certos tipos de memórias. A seguir temos alguns exemplos de latches e flip flops e suas respectivas tabelas verdade.

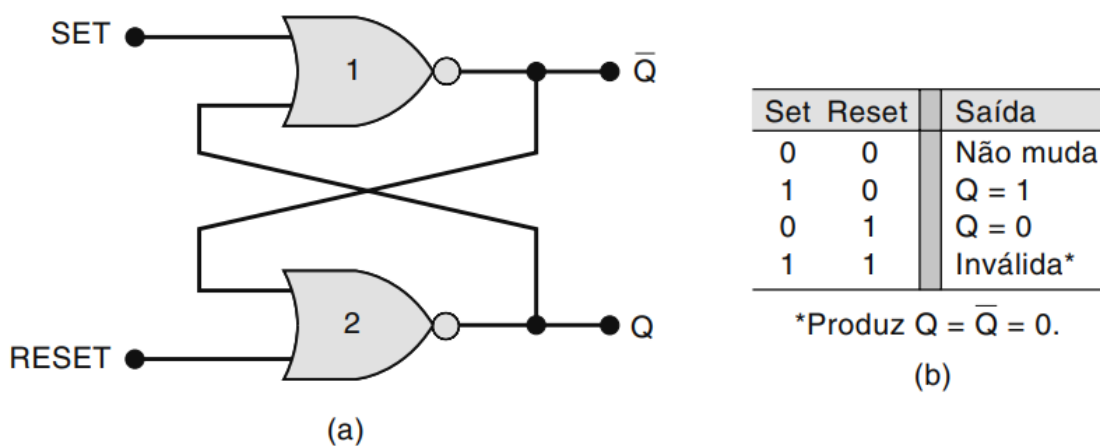
² Disponível em: http://www.dpi.inpe.br/carlos/Academicos/Cursos/ArqComp/aula_5.html; Acesso em Fev.2021.

Figura 3 – Latch que utiliza portas NAND e sua tabela verdade.



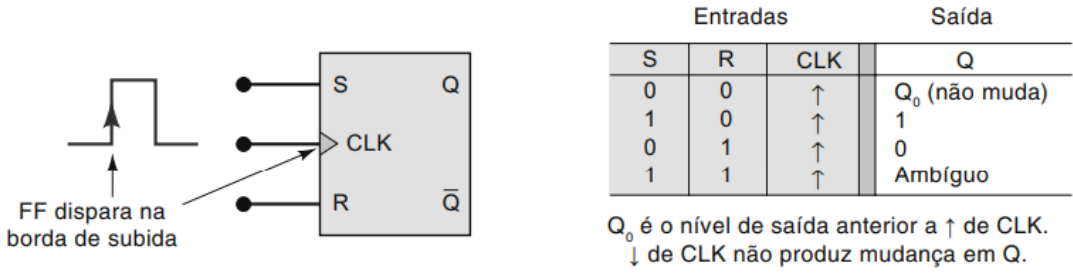
Fonte: Tocci, Ronald J.; Widner, Neal S.; Moss, Gregory L. (2011, pág. 178)

Figura 4 – Latch que utiliza portas NOR e sua tabela verdade.



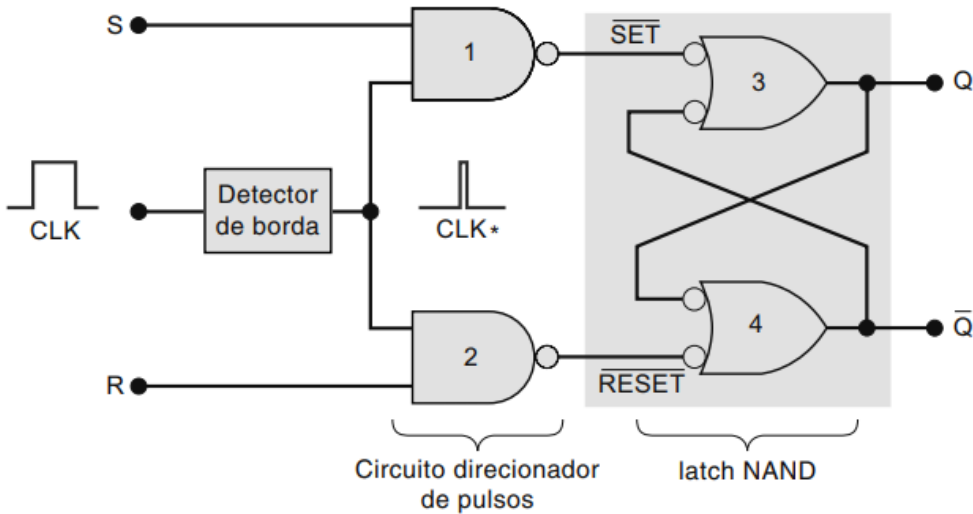
Fonte: Tocci, Ronald J.; Widner, Neal S.; Moss, Gregory L. (2011, pág. 181)

Figura 5 – Flip-flop S-R com clock que responde apenas à borda de subida do pulso de clock e sua tabela verdade.



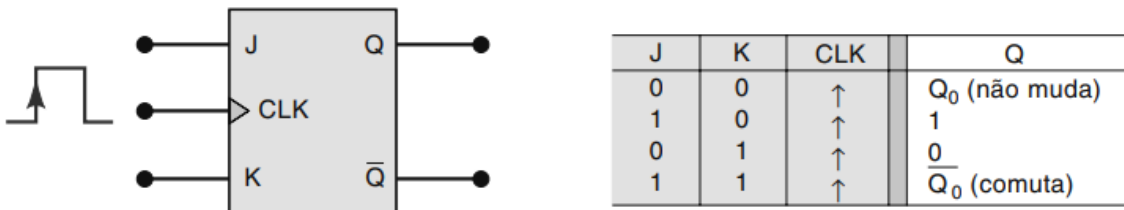
Fonte: Tocci, Ronald J.; Widner, Neal S.; Moss, Gregory L. (2011, pág. 188)

Figura 6 – Circuito interno de um flip-flop S-R disparado por borda.



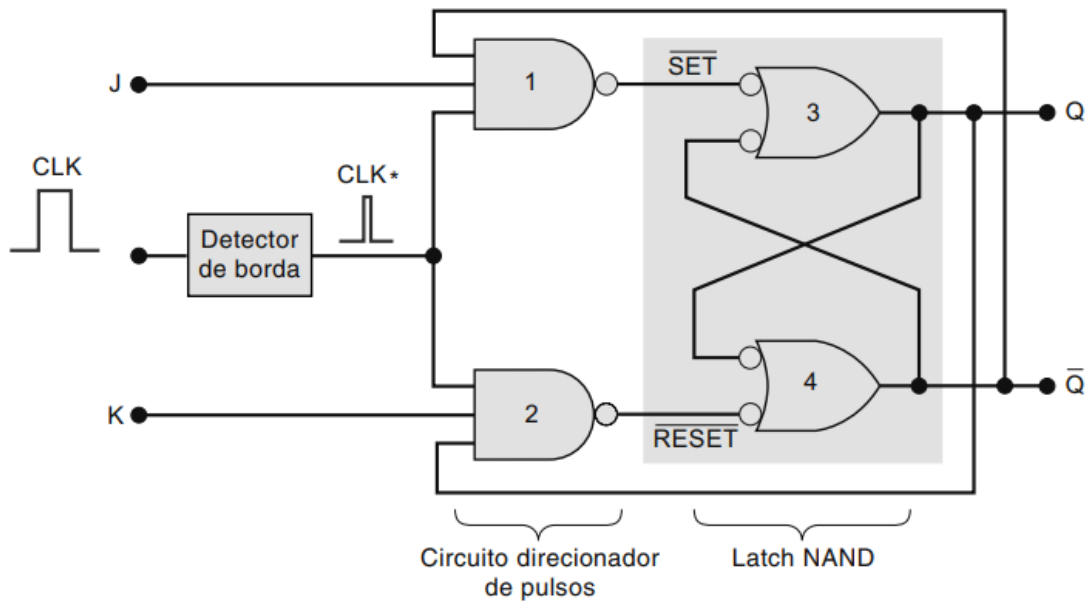
Fonte: Tocci, Ronald J.; Widner, Neal S.; Moss, Gregory L. (2011, pág. 189)

Figura 7 – Flip-flop J-K com clock que responde apenas às bordas positivas do clock e sua tabela verdade.



Fonte: Tocci, Ronald J.; Widner, Neal S.; Moss, Gregory L. (2011, pág. 191)

Figura 8 – Circuito interno de um flip-flop J-K disparado por borda.

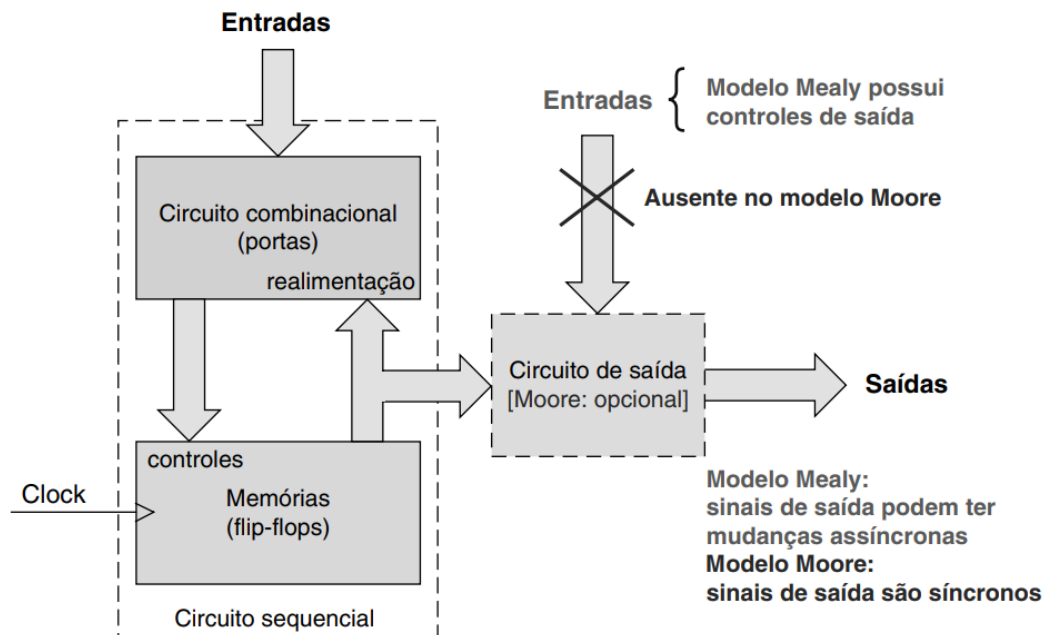


Fonte: Tocci, Ronald J.; Widner, Neal S.; Moss, Gregory L. (2011, pág. 192)

3.6 Máquina de estados de Mealy

Temos que uma Máquina de Estados Finitos é um circuito sequencial, que possui um número finito de estados (e estes são implementados através de flip flops) pré determinados, onde a máquina fica em apenas um estado por vez, sendo este estado chamado de estado atual. A transição de um estado para o outro ocorre de acordo com um clock. Diferente do modelo de Moore, onde as saídas dependem apenas do estado atual, na máquina de Mealy (nome em homenagem ao desenvolvedor desta máquina, George H. Mealy), a saída depende não somente do estado atual, mas também das entradas.

Figura 9 – Diagrama em bloco das máquinas de estado.

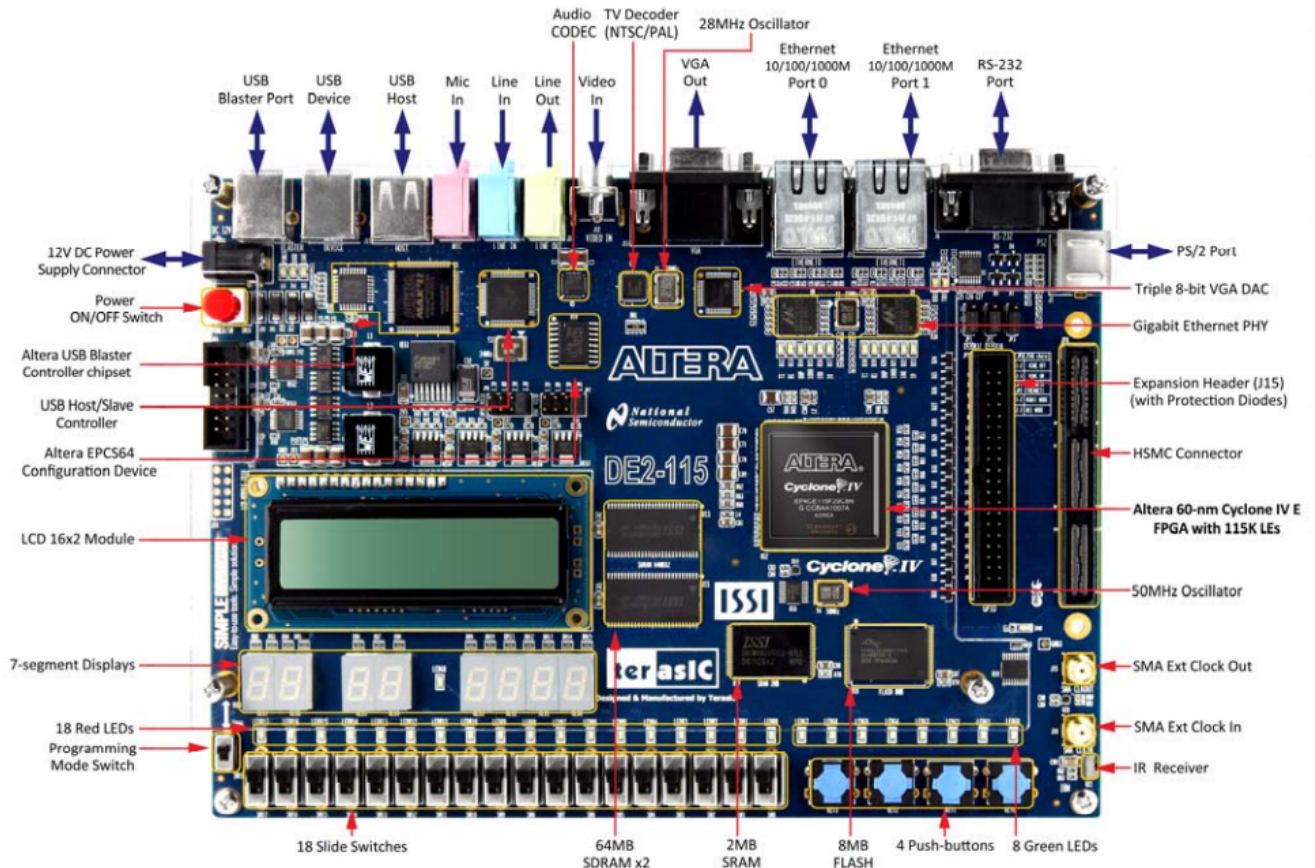


Fonte: Fonte: Tocci, Ronald J.; Widner, Neal S.; Moss, Gregory L. (2011, pág. 367)

3.7 Placa FPGA, Display de 7 segmentos e Conversor BCD

Neste projeto é utilizado o kit de Desenvolvimento FPGA (*Field Programmable Gate Arrays*) DE2-115 da Altera, a qual possui muitos recursos que permitem aos usuários implementar uma ampla gama de circuitos projetados, que vão desde circuitos simples a vários projetos de multimídia, como consta no manual do kit.

Figura 10 – Placa DE2-115

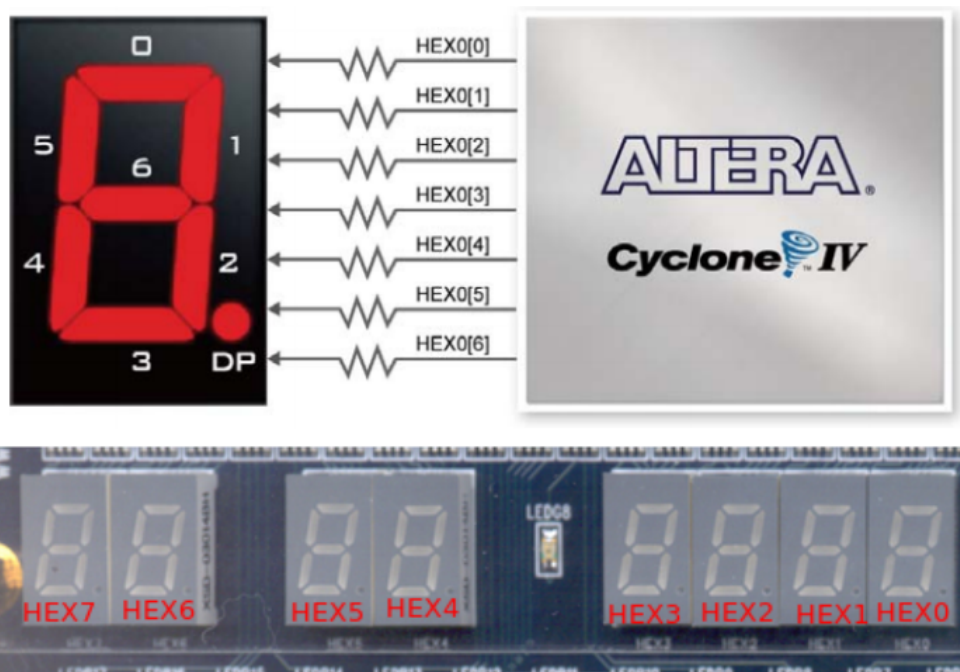


Fonte: Wiki IFSC do Campus São José³

Esta placa possui oito displays de sete segmentos. Estes displays são do tipo anodo comum.

³ Disponível em: https://wiki.sj.ifsc.edu.br/index.php/Interfaces_de_entrada_e_saida_da_DE2-115. Acesso em Fev.2021.

Figura 11 – Displays de sete segmentos presentes na placa.



Fonte: Wiki IFSC do Campus São José⁴

Neste projeto utilizaremos um conversor BCD para transformar a saída de cada estado, ou seja, um número binário, em um número decimal e exibi-lo no display HEX[3].

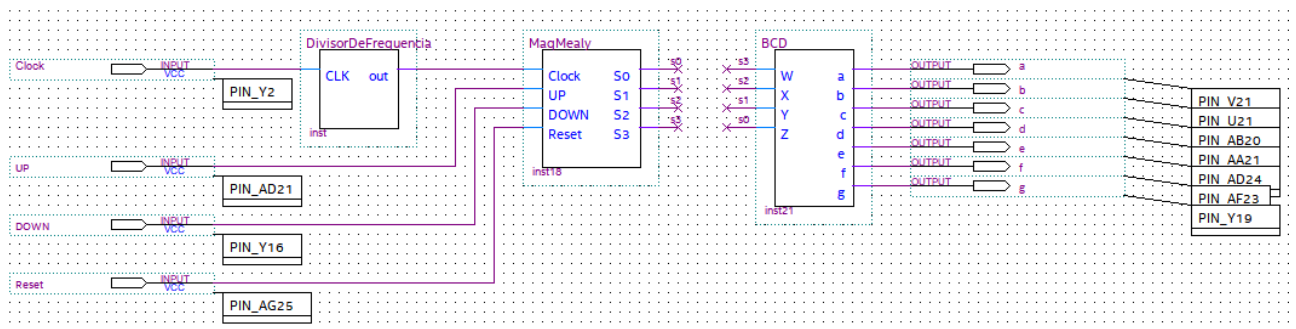
⁴ Disponível em: https://wiki.sj.ifsc.edu.br/index.php/Interfaces_de_entrada_e_saida%C3%ADda_da_DE2-115. Acesso em Fev.2021.

4 Desenvolvimento

4.1 Contador completo

O circuito completo do contador crescente/decrescente é o mostrado na figura [12].

Figura 12 – Implementação completa do contador: o circuito da máquina de Mealy, o circuito divisor de frequência e decodificador BCD para display de 7 segmentos.



Fonte: elaborado pela autora.

O contador possui, ao todo, quatro entradas: "Clock", "UP", "DOWN" e "RESET".

O clock utilizado será o clock interno do kit FPGA. As chaves UP e DOWN são as responsáveis pela designação da função que a máquina irá realizar, de acordo com a Fig 13.

Figura 13 – Tabela-verdade para o processo de contagem.

UP (SW1)	DOWN (SW0)	Contagem
0	0	Mantém
0	1	Decrescente
1	0	Crescente
1	1	Display Apagado

Fonte: Slide da semana 8 da Prof^a. Dr^a. Fernanda Quelho Rossi

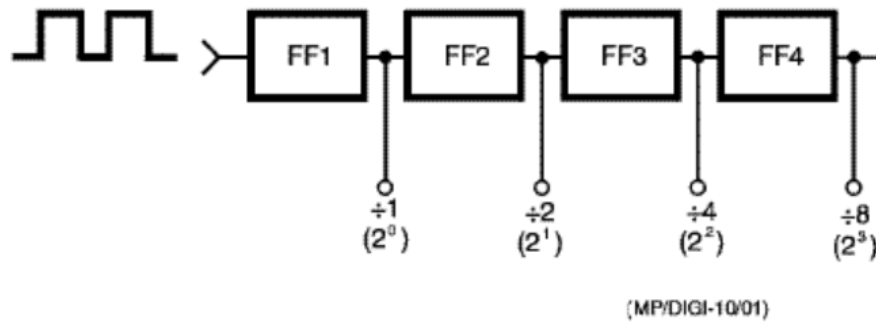
A chave UP utiliza a chave do tipo interruptor SW1 do kit FPGA, enquanto que DOWN utiliza a chave SW0. O reset (assíncrono com o clock) utiliza a chave do tipo interruptor SW17.

As saídas "a", "b", "c", "d", "e", "f" e "g" correspondem as entradas do display de 7 segmentos. Será utilizado o display HEX[3].

4.2 Divisor de frequência

O divisor de frequência é utilizado com o objetivo de obter um clock com uma frequência menor que a frequência original. Utilizando flip-flops conectados em cascata, temos a frequência de entrada dividida por dois a cada flip flop inserido.

Figura 14 – Esquema de um divisor de frequência.



Fonte: RADIOPOINT¹

O cálculo da quantidade de flip flops que serão necessários para o presente projeto é o seguinte: Período é igual ao inverso da frequência:

$$T = \frac{1}{f}$$

Sendo a frequência f no último flip flop igual a frequência do clock dividida pelo módulo do contador (2^n), temos que

$$f = \frac{f_{clk}}{2^n}$$

Então,

$$T = \frac{2^n}{f_{clk}}$$

Queremos $T = 1$ e sabemos que $f_{clk} = 50MHz$, logo

$$1 = \frac{2^n}{50 \cdot 10^6}$$

O que implica que

$$n = \frac{\ln(500000000)}{\ln(2)}$$

Portanto

$$n = 25,57542...$$

Concluimos, então, que são necessários, aproximadamente, 26 flip flops. Entretanto, podemos utilizar algumas estratégias para atingir um tempo mais preciso e chegar o mais

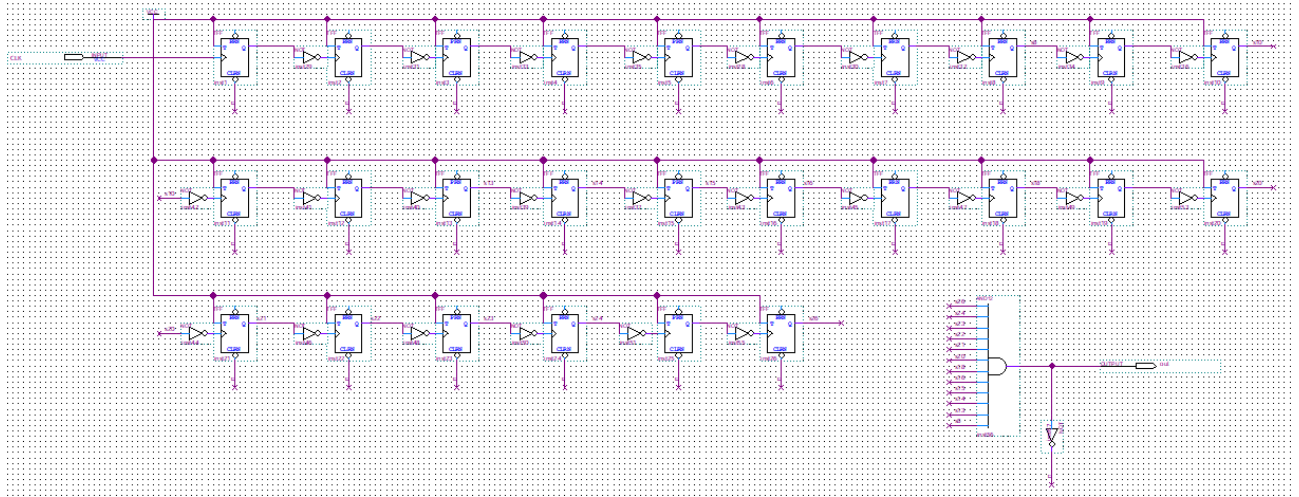
¹ Disponível em: <http://www.radiopoint.com.br/divisores%20digitais.htm>; Acesso em Fev.2021.

próximo possível de 1 segundo. Com 26 flip flops podemos contar até $2^{26} = 67108864$, entretanto precisamos contar até $50 \cdot 10^6$. Convertendo $50 \cdot 10^6$ para binário, obtemos

1011111010111100001000000010111110101111000010000000.

No circuito conectamos todas as saídas iguais a 1, ou seja, as saídas dos flip flops de número 26, 24, 23, 22, 21, 20, 18, 16, 15, 14, 13 e 8 em uma porta AND. Utiliza-se como pulso de reset a própria saída "out", com um inversor. É necessário o reset pois os flip flops devem ser zerados cada vez que eles atingirem o valor binário acima citado.

Figura 15 – Circuito completo do divisor de frequência.



Fonte: elaborado pela autora.

4.3 Decodificador BCD para Display de 7 segmentos

Para representar as saídas no display presente na placa FPGA precisamos de um circuito decodificador responsável por fazer com que os bits que compõem a saída acendam os leds corretos. Vale ressaltar que os displays da placa DE2-115 são do tipo anodo comum, o que significa que aplicando um nível lógico baixo no pino correspondente fará com que o segmento se acenda, enquanto que a aplicação do nível lógico alto fará com que ele fique se apague. A tabela verdade do decodificador BCD segue abaixo

Tabela 1 – Tabela-verdade do decodificador BCD.

Números (decimal)	Entradas				Saídas para o display						
	W	X	Y	Z	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1
10	1	0	1	0	0	0	0	0	0	0	0
11	1	0	1	1	0	0	0	0	0	0	0
12	1	1	0	0	0	0	0	0	0	0	0
13	1	1	0	1	0	0	0	0	0	0	0
14	1	1	1	0	0	0	0	0	0	0	0
15	1	1	1	1	0	0	0	0	0	0	0

Fonte: elaborado pela autora.

Através da tabela, foram construídos os mapas de Karnaugh e obtidas as expressões booleanas através do site 32x8 ².

Lembrando que, nas expressões, "+" representa a função OR, enquanto que as entradas juntas representam a função AND.

$$a = \bar{W}Y + \bar{W}\bar{X}\bar{Z} + \bar{W}XZ + W\bar{Z}\bar{Y}$$

$$b = \bar{W}\bar{X} + \bar{X}\bar{Y} + \bar{W}\bar{Y}\bar{Z} + \bar{W}YZ$$

$$c = \bar{X}\bar{Y} + \bar{W}Z + \bar{W}X$$

$$d = \bar{W}X\bar{Y}Z + \bar{W}\bar{X}Y + \bar{W}Y\bar{Z} + \bar{X}\bar{Y}\bar{Z}$$

$$e = \bar{X}\bar{Y}\bar{Z} + \bar{W}Y\bar{Z}$$

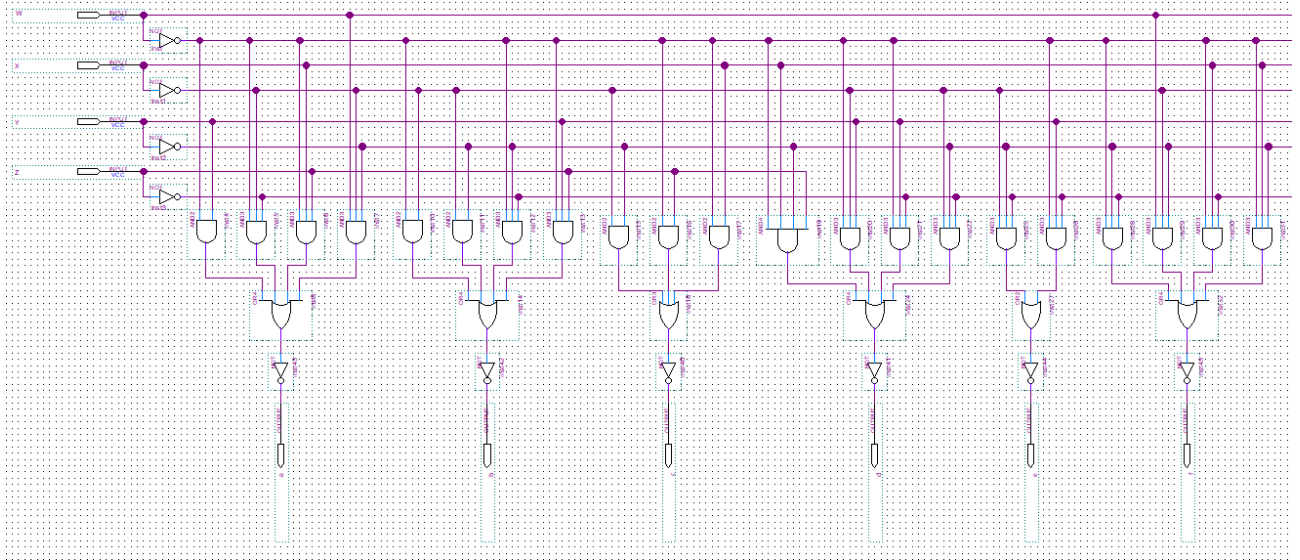
$$f = \bar{W}\bar{Y}\bar{Z} + W\bar{X}\bar{Y} + \bar{W}X\bar{Z} + \bar{W}X\bar{Y}$$

$$g = \bar{W}\bar{X}Y + \bar{W}Y\bar{Z} + \bar{W}X\bar{Y} + W\bar{X}\bar{Y}$$

A partir destes resultados montamos o circuito abaixo:

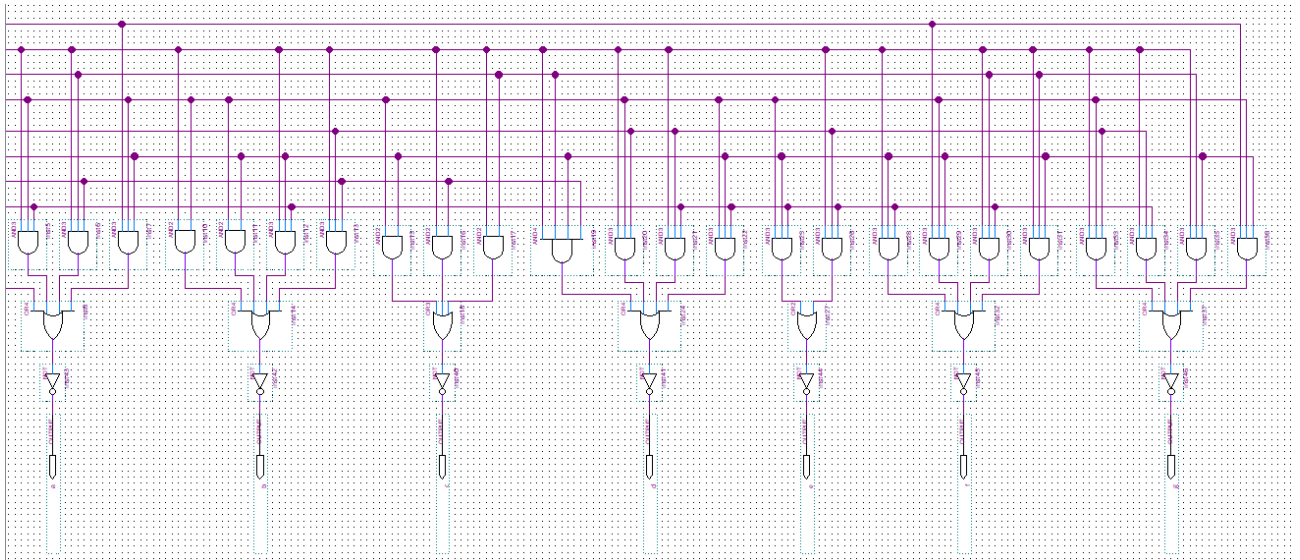
² Disponível em: <http://www.32x8.com/index.html>; Acesso em Fev.2021.

Figura 16 – Circuito do decodificador BCD para display de 7 segmentos (parte I).



Fonte: elaborado pela autora.

Figura 17 – Circuito do decodificador BCD para display de 7 segmentos (parte II).



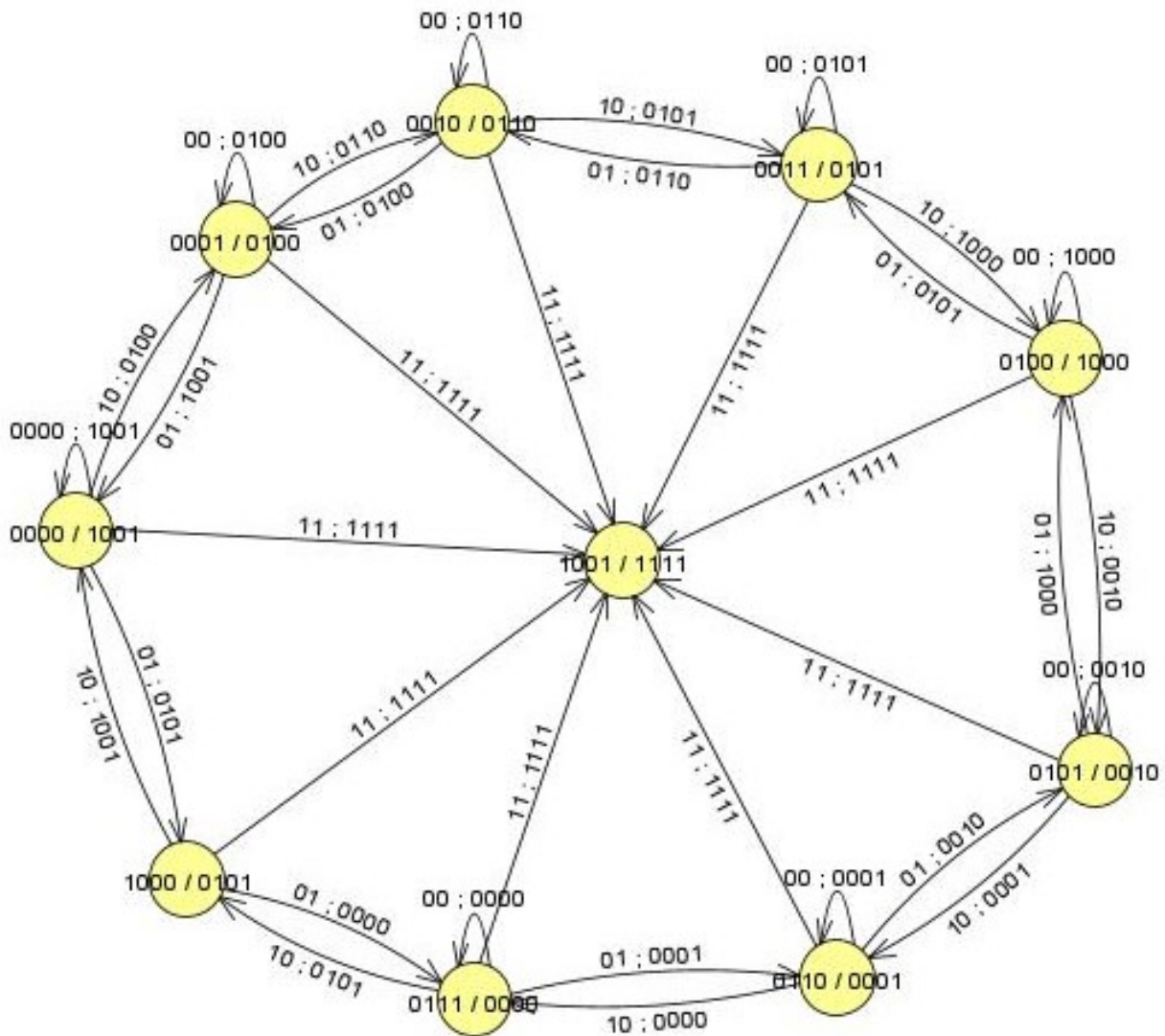
Fonte: elaborado pela autora.

4.4 Diagrama de estados, tabela verdade e expressões booleanas

Para desenvolver a máquina de estados, primeiramente é necessário elaborar o diagrama de estados. Este mostra os possíveis estados de um objeto e as ações responsáveis pelas suas mudanças de estado. Foi utilizado o software JFLAP ³ para a criação deste diagrama.

³ Disponível em: <http://www.jflap.org/>. Acesso em Fev.2021.

Figura 18 – Diagrama de estados.



Fonte: elaborado pela autora.

A partir disto, desenvolve-se a tabela verdade [2] do circuito que possui os estados atuais, os próximos estados, as saídas e o modo (UP/DOWN).

Tabela 2 – Tabela-verdade dos circuitos combinacionais de entrada e saída.

Estado inicial	Próximo Estado: (E3',E2',E1',E0') / Saída: (S3,S2,S1,S0)			
(E3, E2, E1, E0)	UD(00)	UD(01)	UD(10)	UD(11)
0000	0000 / 1001	1000 / 0101	0001 / 0100	1001 / 1111
0001	0001 / 0100	0000 / 1001	0010 / 0110	1001 / 1111
0010	0010 / 0110	0001 / 0100	0011 / 0101	1001 / 1111
0011	0011 / 0101	0010 / 0110	0100 / 1000	1001 / 1111
0100	0100 / 1000	0011 / 0101	0101 / 0010	1001 / 1111
0101	0101 / 0010	0100 / 1000	0110 / 0001	1001 / 1111
0110	0110 / 0001	0101 / 0010	0111 / 0000	1001 / 1111
0111	0111 / 0000	0110 / 0001	1000 / 0101	1001 / 1111
1000	1000 / 0101	0111 / 0000	0000 / 1001	1001 / 1111
1001	1001 / 1111	0000 / 1001	0000 / 1001	1001 / 1111

Fonte: elaborado pela autora.

Com base nesta tabela, criamos os mapas de Karnaugh e obtemos as expressões booleanas. Há uma expressão para cada bit do próximo estado e para cada bit da saída, levando em consideração o estado atual e as entradas UP e DOWN. Além disso, existem algumas combinações de bits de entrada que não correspondem a nenhum estado implementado. Nesses casos, nos mapas de Karnaugh, foi utilizado “don’t care” como valor para as entradas que não importam, pois não correspondem a estados válidos.

De acordo com a Tabela 2, obtemos o seguinte:

Próximo Estado:

$$E3' = EF + A\bar{E}\bar{F} + BCDE + \bar{A}\bar{B}\bar{C}\bar{D}F$$

$$E2' = B\bar{C}\bar{F} + B\bar{D}\bar{F} + BD\bar{E} + BC\bar{E} + A\bar{D}\bar{E}F + \bar{B}CDE\bar{F}$$

$$E1' = C\bar{D}\bar{F} + CD\bar{E} + A\bar{D}\bar{E}F + \bar{A}\bar{C}DE\bar{F} + B\bar{C}\bar{D}\bar{E}F$$

$$E0' = EF + \bar{A}\bar{D}\bar{E} + D\bar{E}\bar{F} + C\bar{D}F + B\bar{D}F + A\bar{D}F$$

Saídas

$$S3 = AD + \bar{A}\bar{C}\bar{D} + EF$$

$$S2 = A + \bar{B}D + \bar{B}C + EF$$

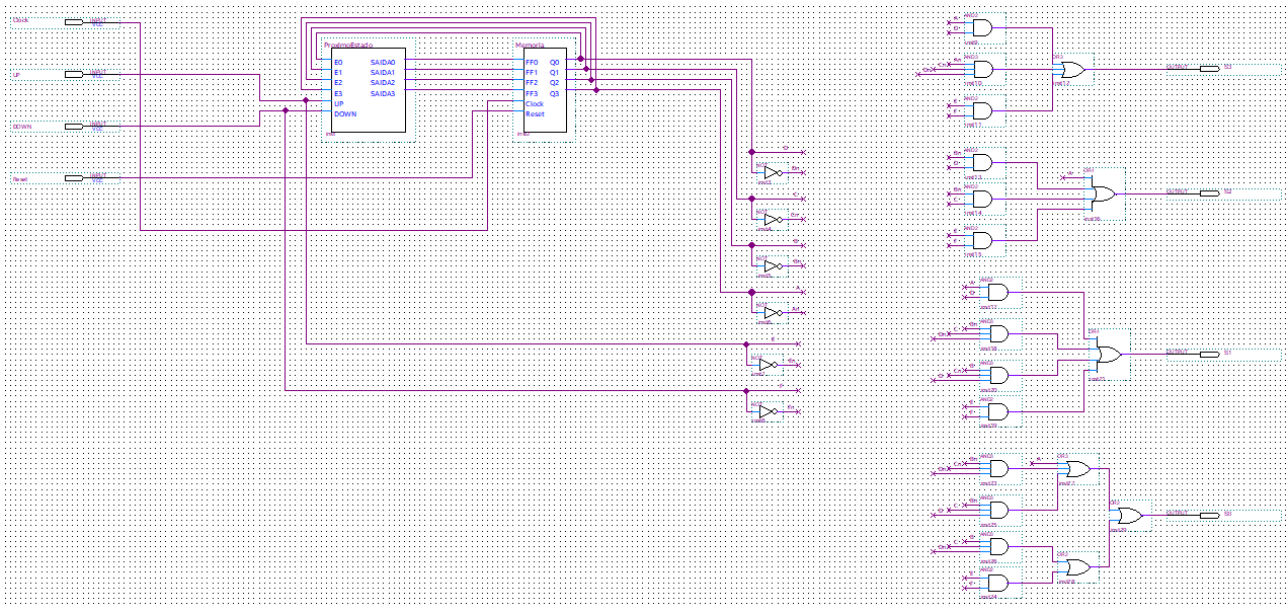
$$S1 = AD + \bar{B}C\bar{D} + B\bar{C}D + EF$$

$$S0 = A + \bar{B}\bar{C}\bar{D} + \bar{B}CD + B\bar{C}\bar{D} + EF$$

4.5 Máquina de Mealy

A partir dos dados acima citados, implementa-se a máquina de Mealy.

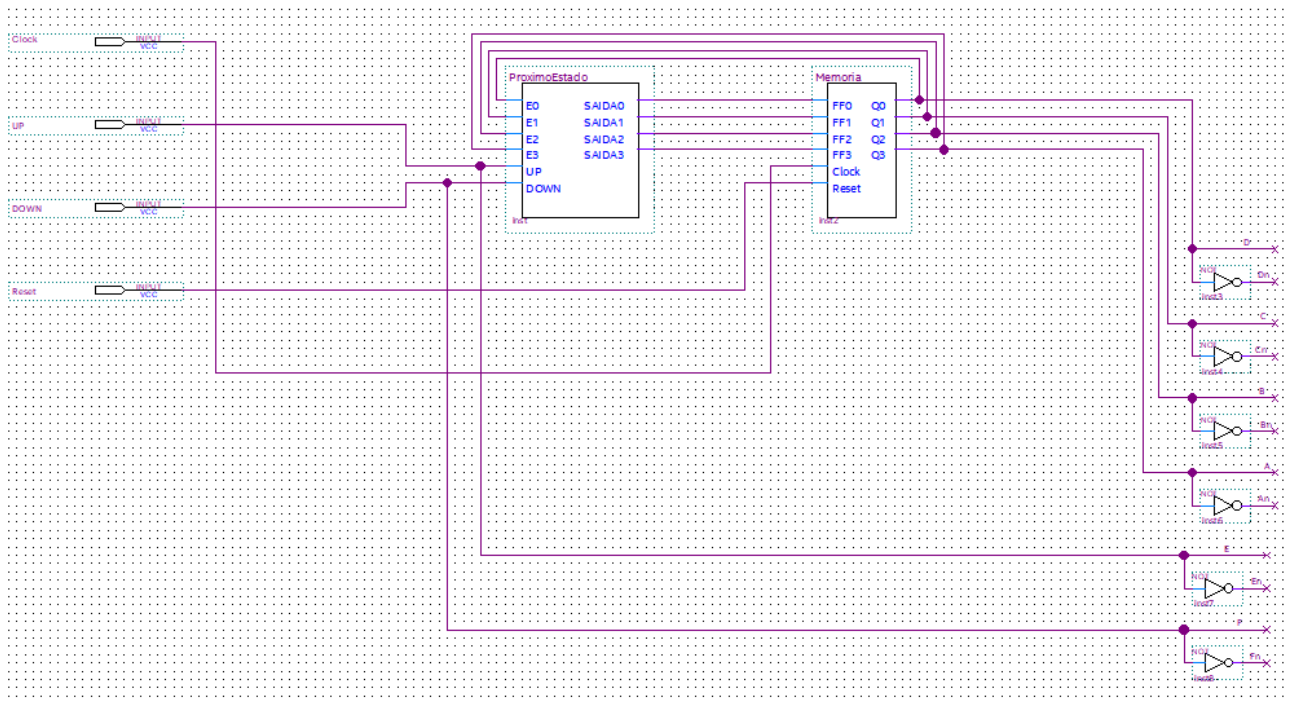
Figura 19 – Circuito completo da máquina de Mealy.



Fonte: elaborado pela autora.

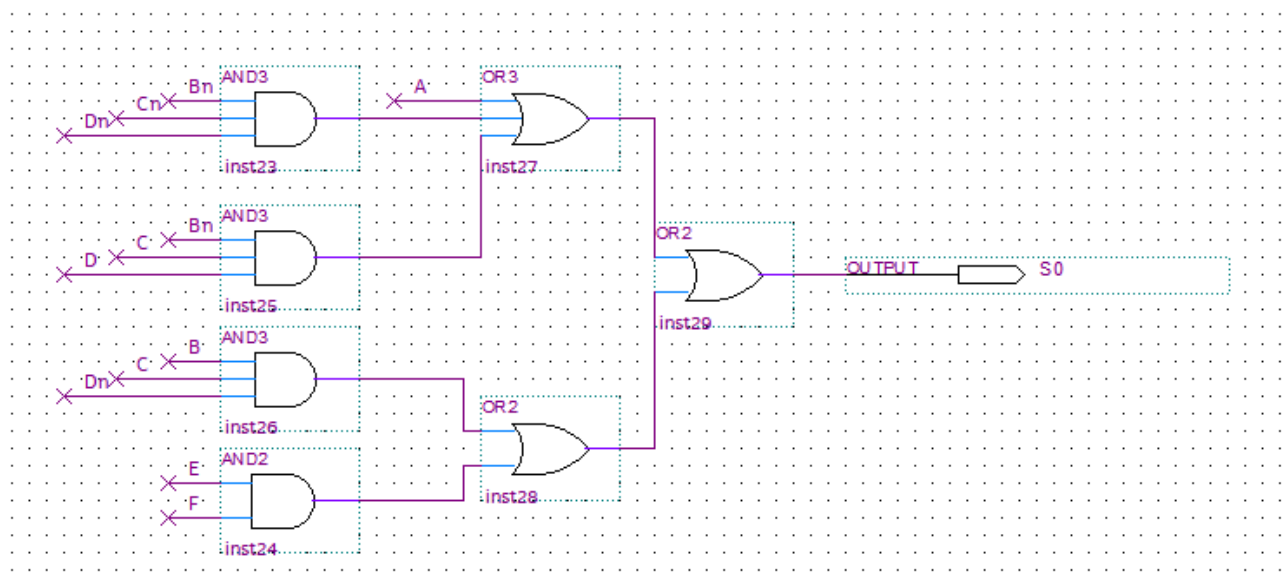
O circuito da máquina de Mealy possui as entradas de clock, UP, DOWN e de reset. Além disso, contém duas black box: uma com o circuito para se calcular o próximo estado e outra com a memória. Por fim, possui os circuitos que geram as saídas S0, S1, S2 e S3.

Figura 20 – Mais detalhes do circuito da máquina de Mealy: as entradas, uma black box que calcula os próximos estados e uma black box com a memória.



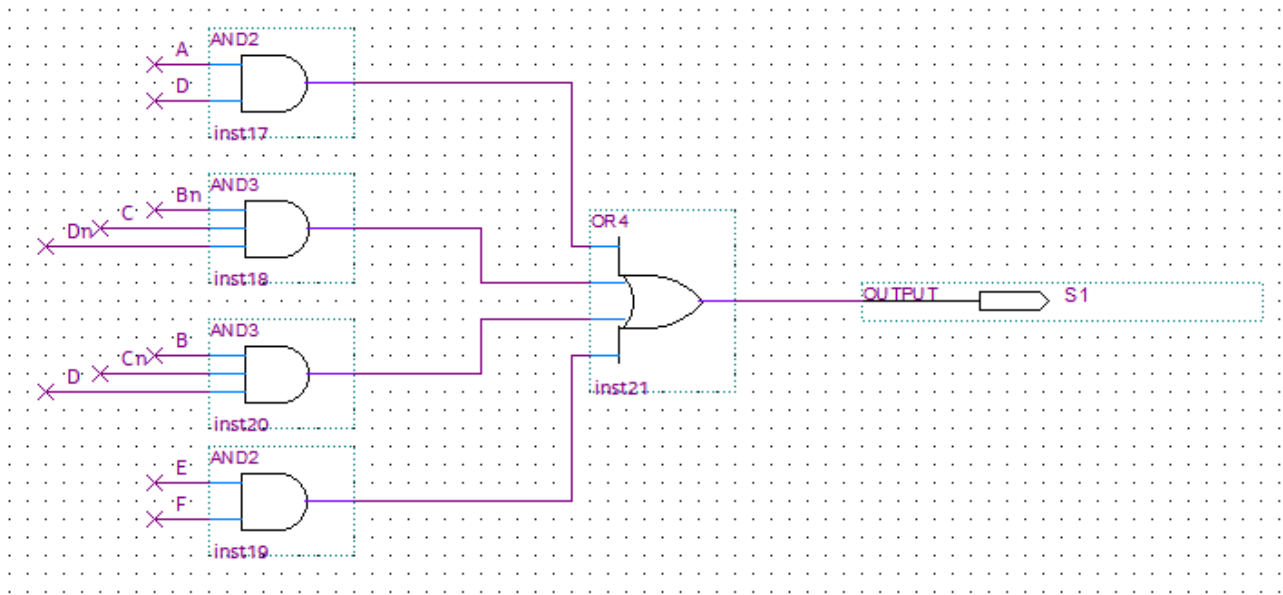
Fonte: elaborado pela autora.

Figura 21 – Mais detalhes do circuito da máquina de Mealy: circuito que gera a saída S0.



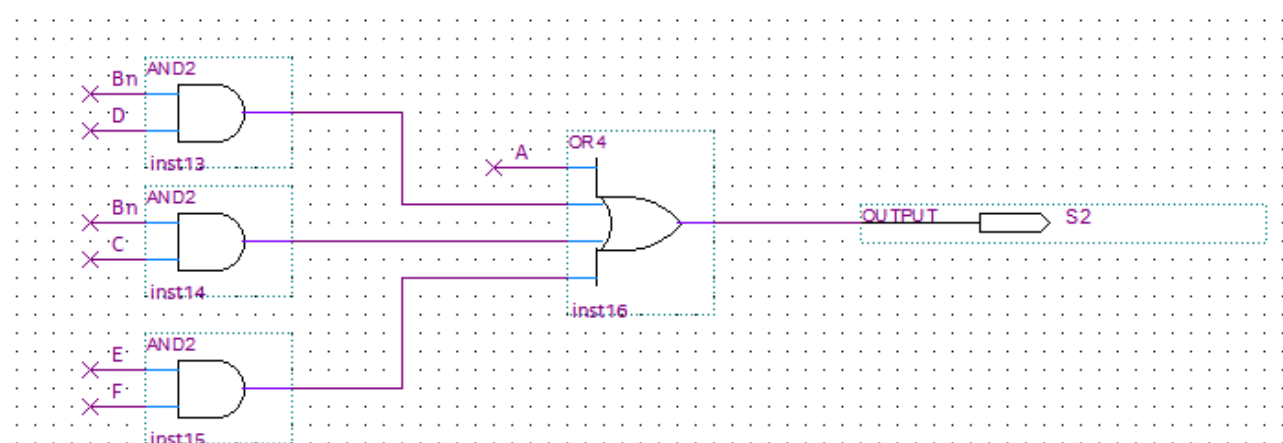
Fonte: elaborado pela autora.

Figura 22 – Mais detalhes do circuito da máquina de Mealy: circuito que gera a saída S1.



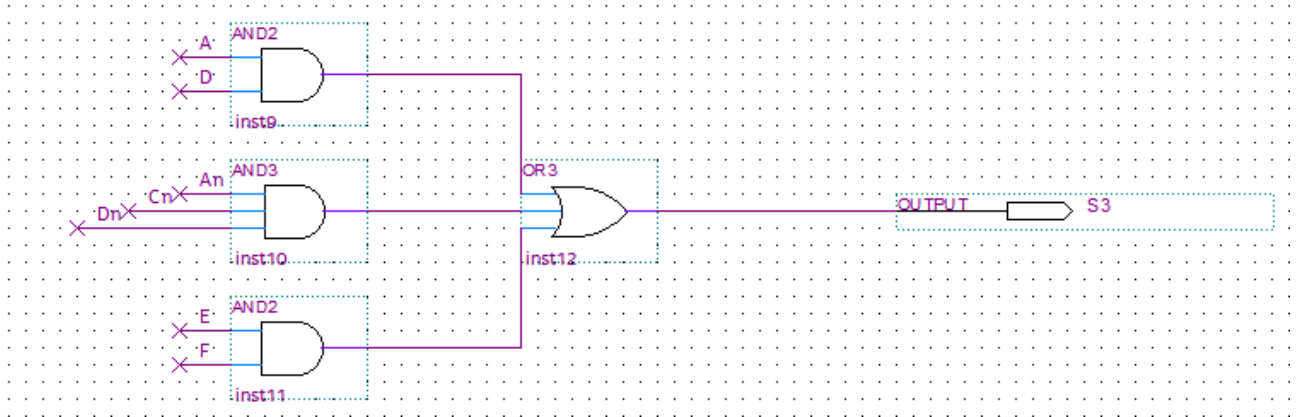
Fonte: elaborado pela autora.

Figura 23 – Mais detalhes do circuito da máquina de Mealy: circuito que gera a saída S2.



Fonte: elaborado pela autora.

Figura 24 – Mais detalhes do circuito da máquina de Mealy: circuito que gera a saída S3.

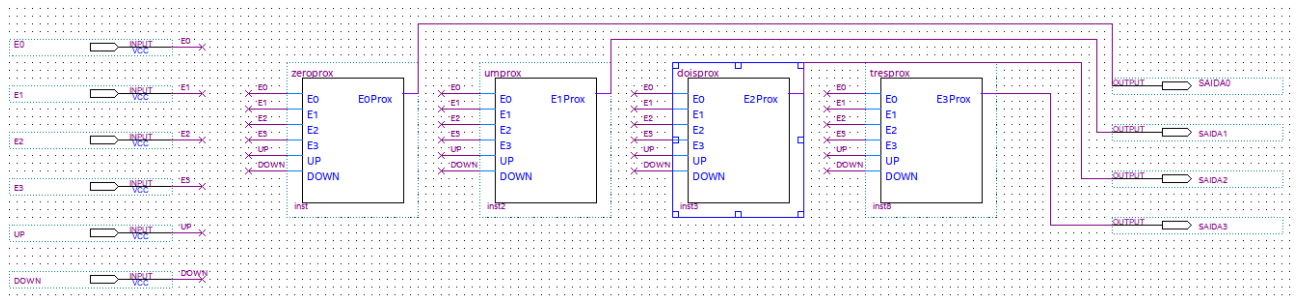


Fonte: elaborado pela autora.

4.6 Próximo estado

O circuito combinacional para calcular o próximo estado foi implementado utilizando quatro black box, uma para cada bit.

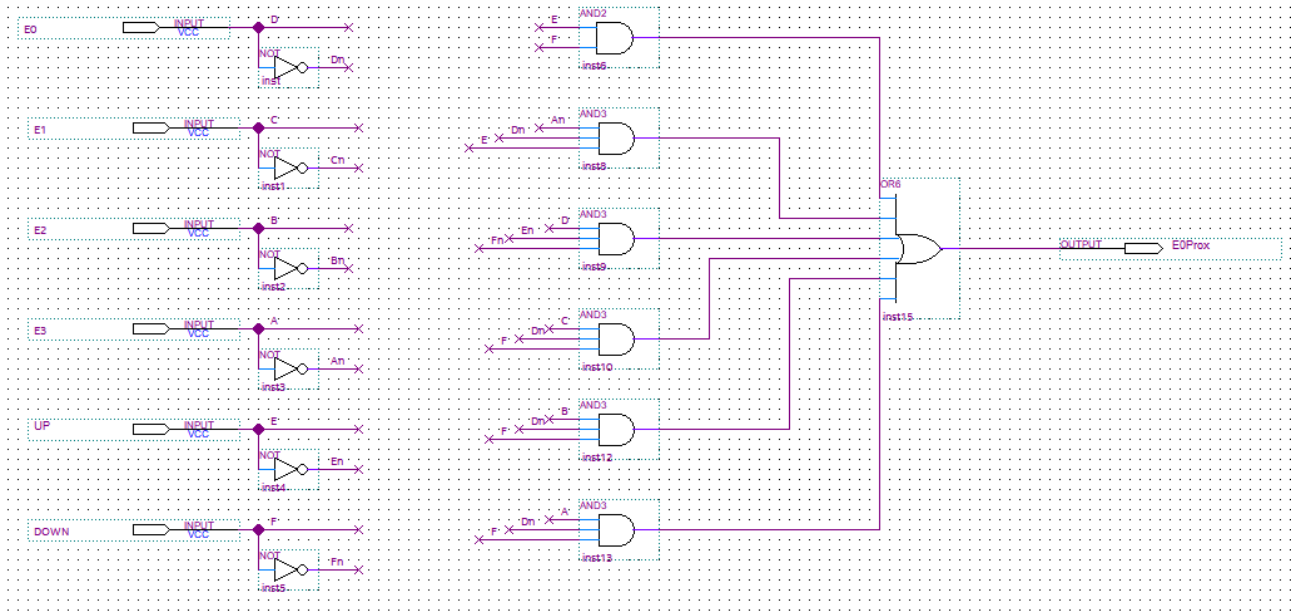
Figura 25 – Circuito da black box "ProximoEstado".



Fonte: elaborado pela autora.

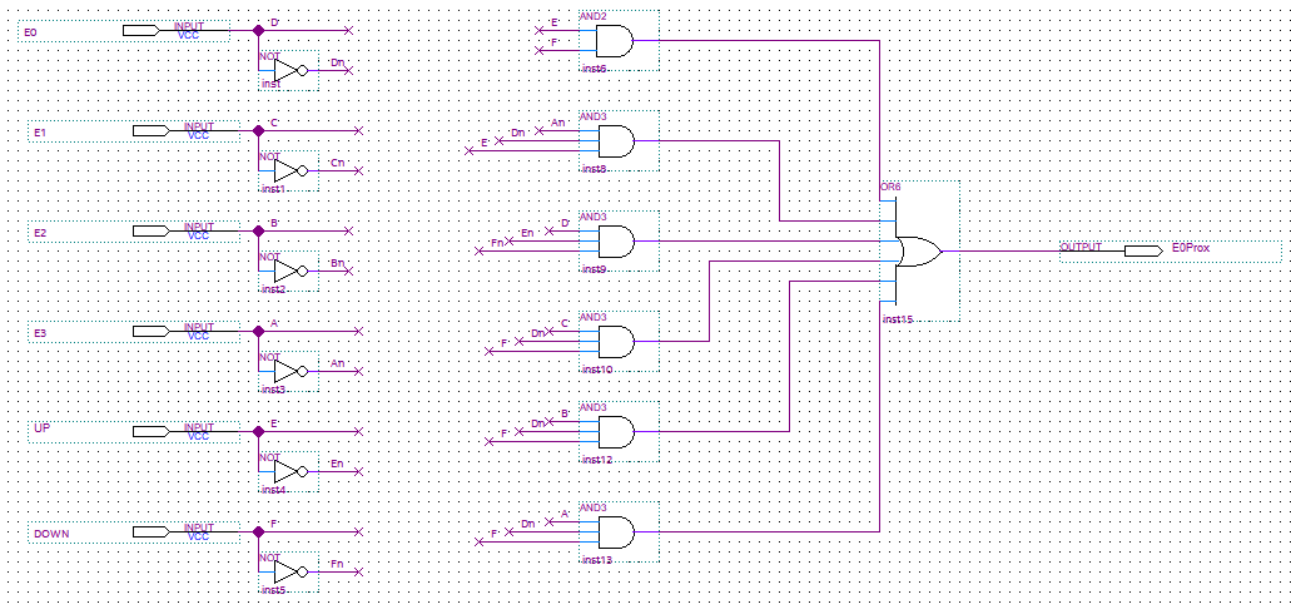
Abaixo observa-se as imagens referentes a cada black box, iniciando pela qual calcula o próximo estado do bit menos significativo, intitulada "zeroprox" e terminando na qual calcula o próximo estado do bit mais significativo, intitulada "tresprox".

Figura 26 – Circuito da black box "zeroprox".



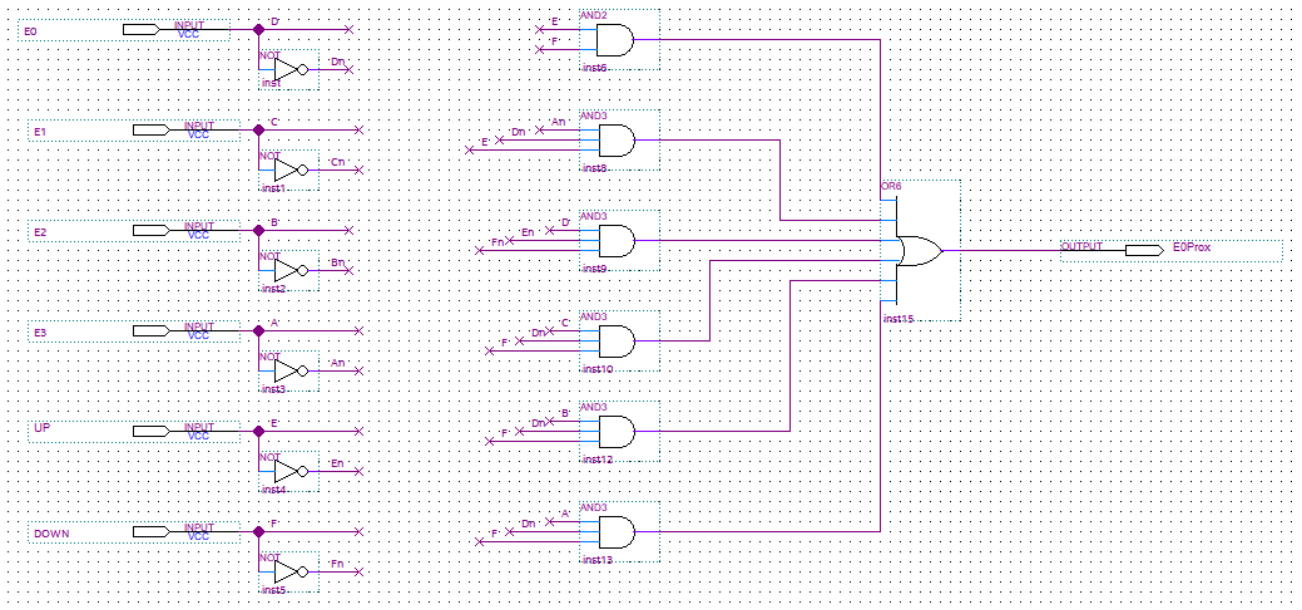
Fonte: elaborado pela autora.

Figura 27 – Circuito da black box "umprox".



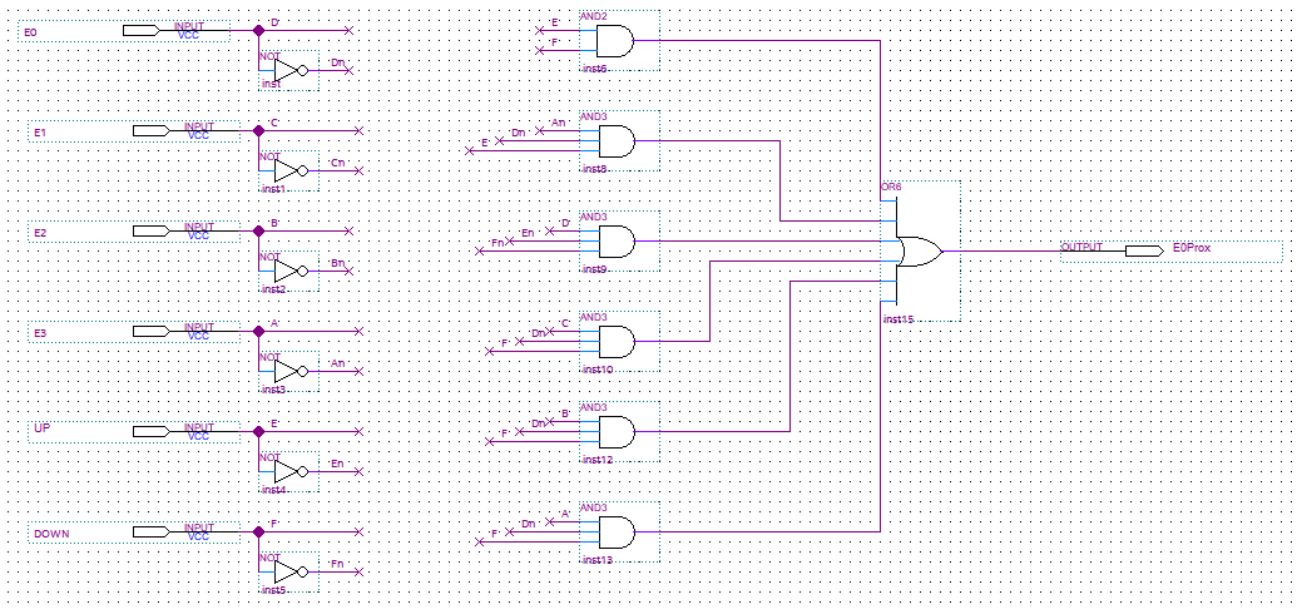
Fonte: elaborado pela autora.

Figura 28 – Circuito da black box "doisprox".



Fonte: elaborado pela autora.

Figura 29 – Circuito da black box "tresprox".

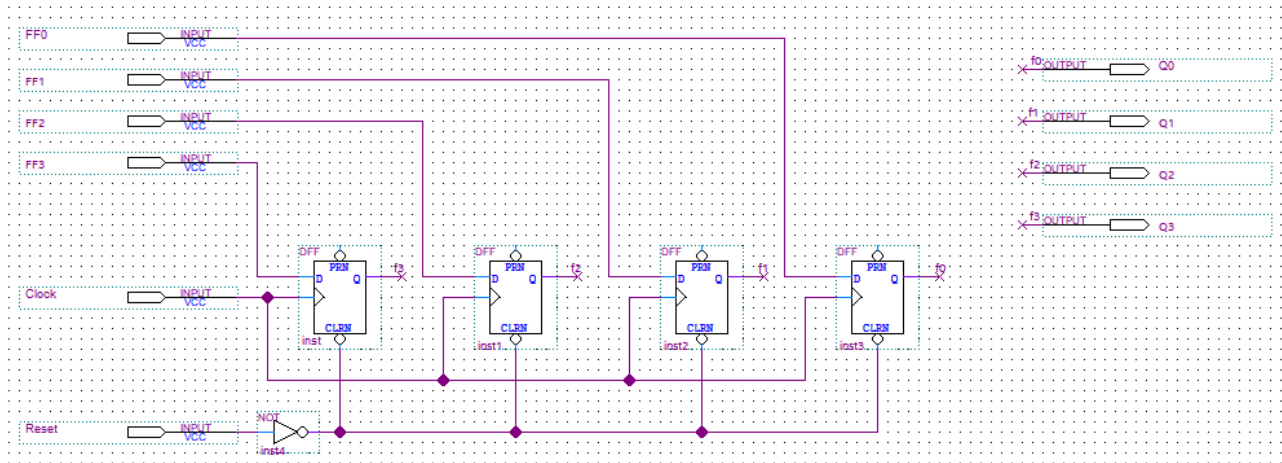


Fonte: elaborado pela autora.

4.7 Memória

A memória é composto por flip flops tipo D com um reset, responsável por armazenar o estado atual que servirá de entrada para a função de próximo estado e de saída.

Figura 30 – Circuito da black box "Memoria".



Fonte: elaborado pela autora.

5 Resultados obtidos

5.1 Simulação no software Quartus (waveforms)

Abaixo seguem as imagens das *waveforms* geradas pelo software *Intel Quartus Prime*.

Figura 31 – Waveform da simulação que utiliza as entradas UP = 1, DOWN = 0 (ou seja, uma contagem crescente) e reset ativado aos 250 ns e se mantendo por 250 ns.

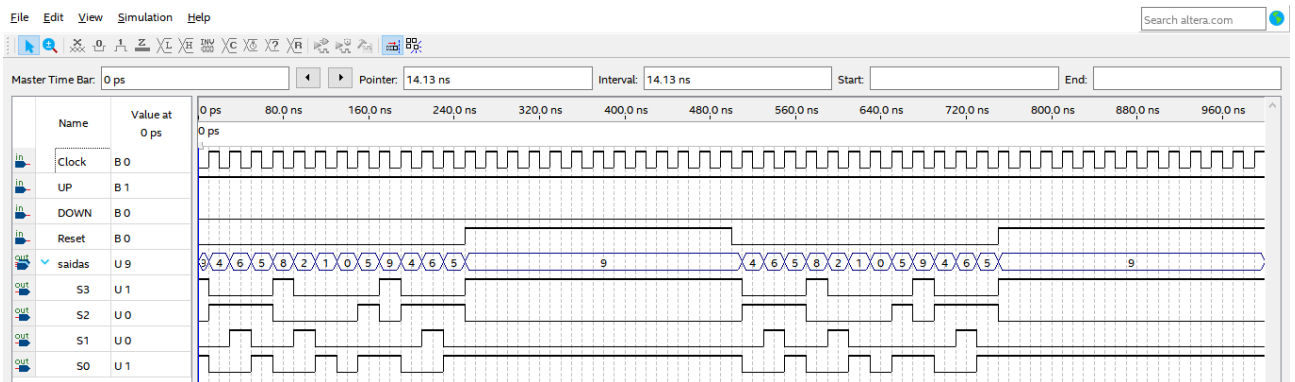
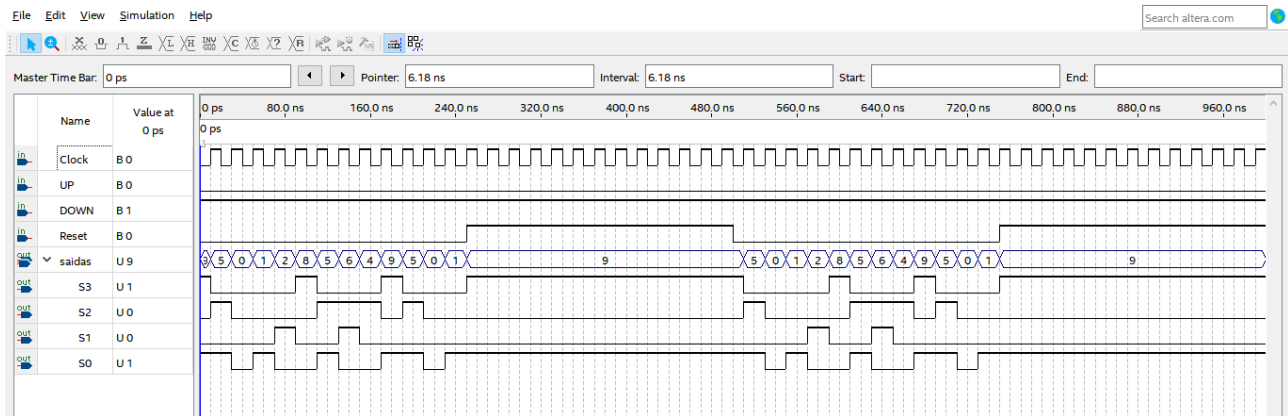
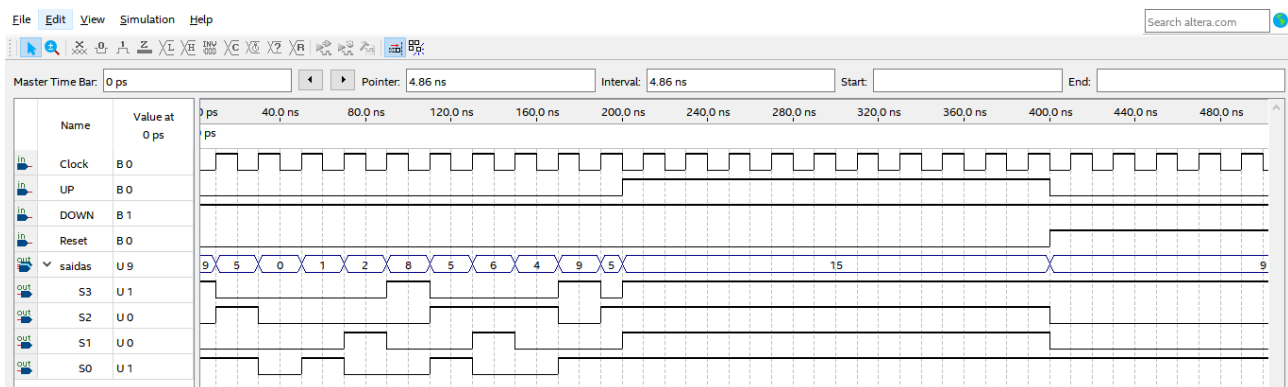


Figura 33 – Waveform da simulação que utiliza as entradas $UP = 0$, $DOWN = 1$ (ou seja, uma contagem decrescente) e reset ativado aos 250 ns e se mantendo por 250 ns.



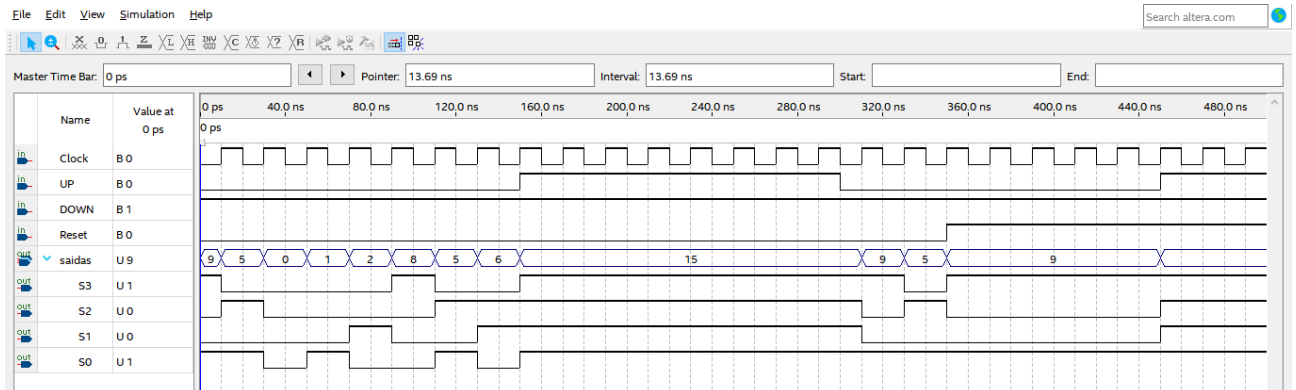
Fonte: elaborado pela autora.

Figura 34 – Waveform da simulação de uma contagem decrescente. Entretanto, aos 200 ns, $UP = 1$, mudando para a função que apaga o display. O reset é ativado aos 400 ns.



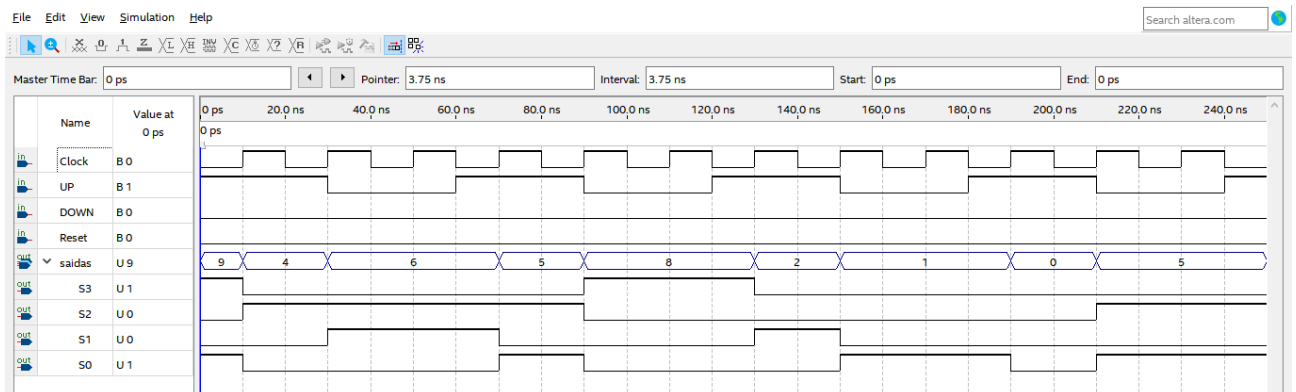
Fonte: elaborado pela autora.

Figura 35 – Waveform da simulação onde DOWN = 1, UP varia de 150 ns em 150 ns e reset varia de 300 ns em 300 ns.



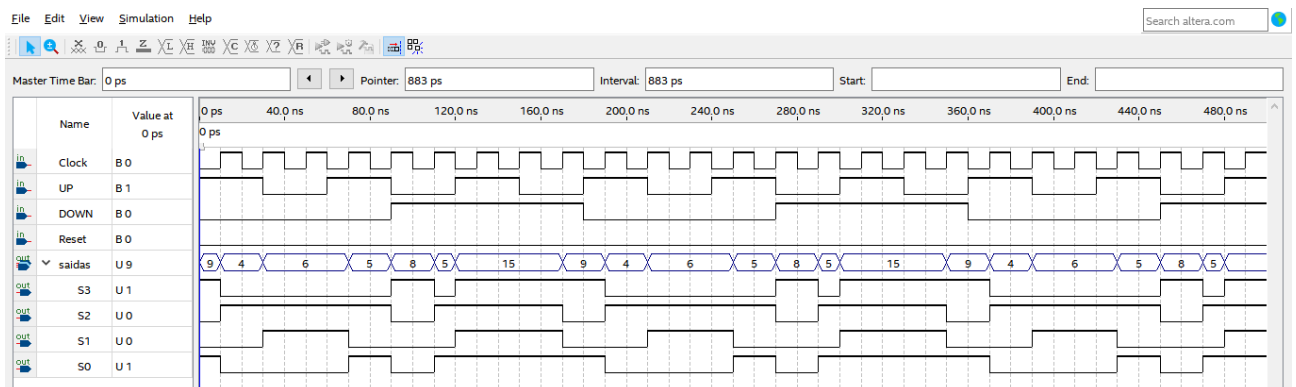
Fonte: elaborado pela autora.

Figura 36 – Waveform da simulação onde DOWN = 0 e UP varia de 30 ns em 30 ns.



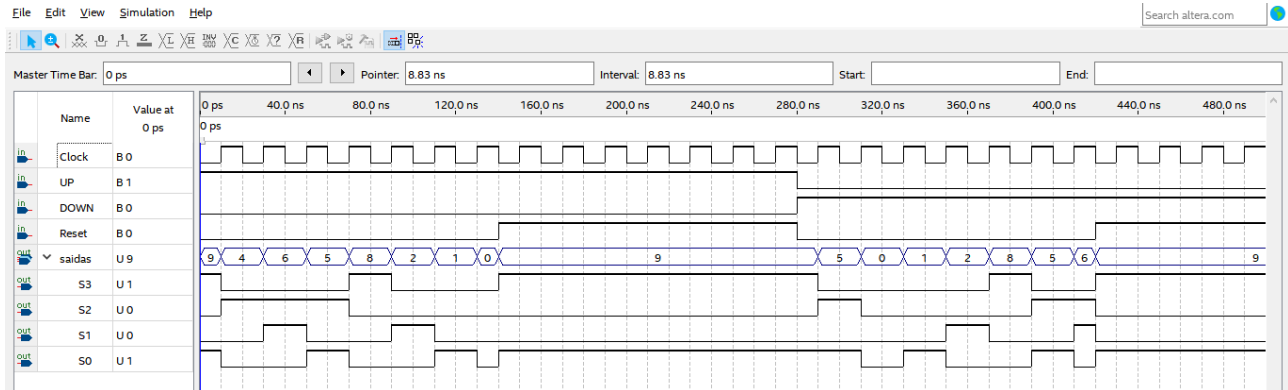
Fonte: elaborado pela autora.

Figura 37 – Waveform da simulação onde UP varia de 30 ns em 30 ns e DOWN varia de 90 ns em 90 ns.



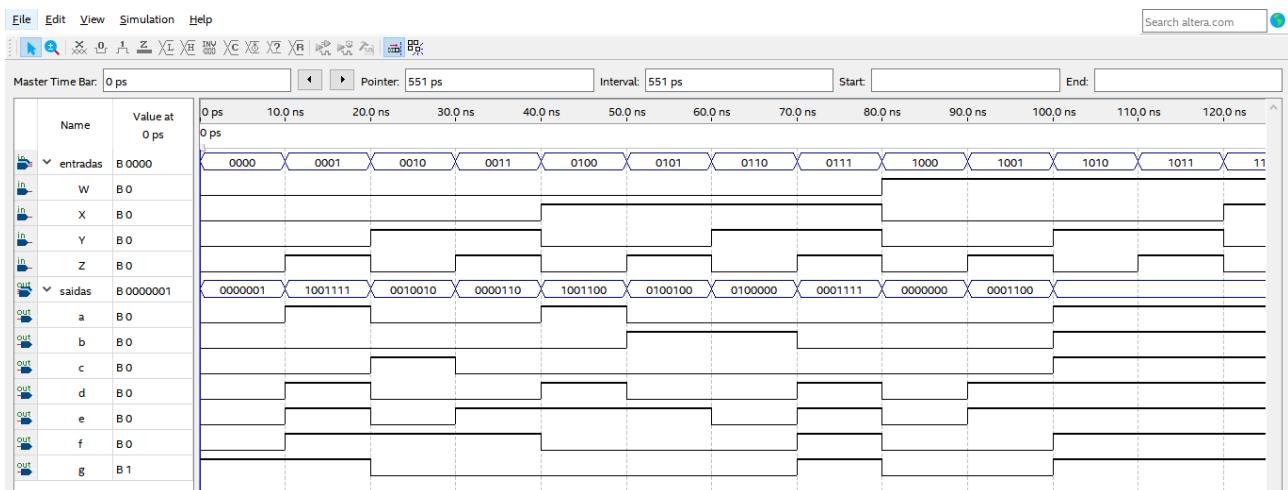
Fonte: elaborado pela autora.

Figura 38 – Waveform da simulação onde está ocorrendo uma contagem crescente, aplica-se o reset e, após isso, ocorre uma contagem decrescente.



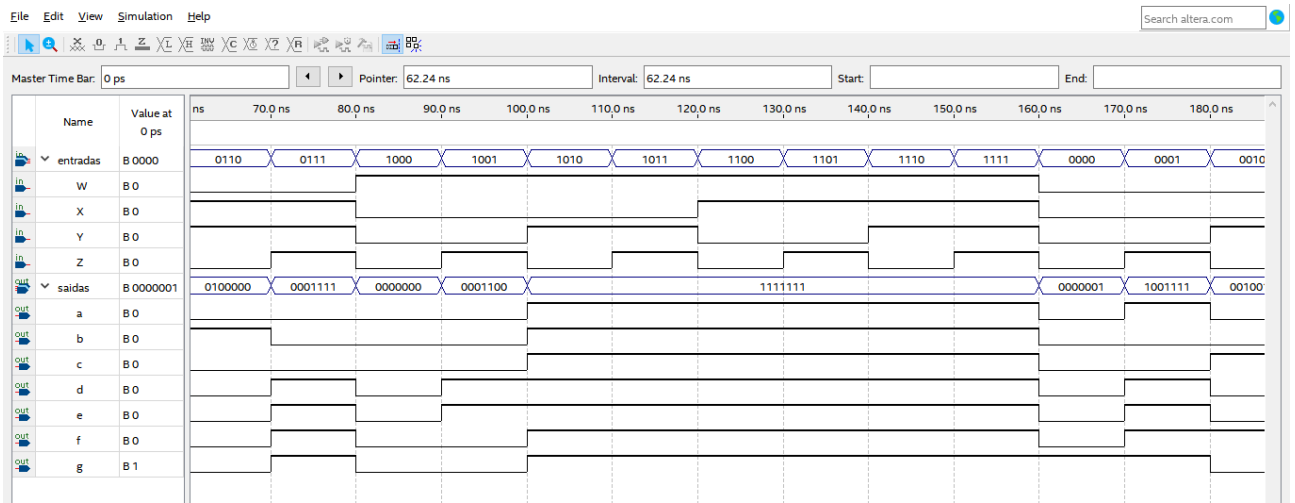
Fonte: elaborado pela autora.

Figura 39 – Waveform da simulação do decodificar BCD para display de 7 segmentos (I).



Fonte: elaborado pela autora.

Figura 40 – Waveform da simulação do decodificar BCD para display de 7 segmentos (II).



Fonte: elaborado pela autora.

5.2 Simulação no kit FPGA remoto

Foram realizadas, no kit FPGA, as seguintes simulações:

- 1) Apresentar todas as transições de estado crescente.
- 2) Apresentar todas as transições de estado decrescente.
- 3) Escolher um estado e manter por aproximadamente 3 segundos.
- 4) Na contagem crescente, entrar no estado apagado e manter neste estado por 3 segundos. Após isto, retornar à contagem crescente.
- 5) Escolher um estado para aplicar o reset. Após isto, iniciar a contagem decrescente.

Em todas elas o circuito apresentou bom funcionamento, mostrando o resultado esperado e correto.

6 Conclusão

A máquina de estados de Mealy não apresentou falhas ao representar os estados. Tanto as simulações em forma de onda realizadas no software *Intel Quartus Prime*, quanto as simulações realizadas no kit remoto foram um sucesso.

Referências

TOCCI, Ronald J.; WIDNER, Neal S.; MOSS, Gregory L. Sistemas Digitais: Princípios e Aplicações. 11^o ed. – São Paulo : Pearson Prentice Hall, 2011.

FLOYD, Thomas. Sistemas Digitais: Fundamentos e Aplicações. 9^a ed. São Paulo: Bookman Companhia Editora Ltda, 2007.

VAHID, Frank. Sistemas Digitais: Projeto, Otimização e HDLs. 1^a ed. São Paulo: Bookman Companhia Editora Ltda, 2008.

EMBARCADOS: Portas lógicas em sistemas digitais. 31 jul. 2015. Disponível em: <https://www.embarcados.com.br/portas-logicas/>. Acesso em: fev. 2021.

SILVATRONICS: O que são portas lógicas em circuitos digitais?. 24 fev. 2020. Disponível em: <https://blog.silvatronics.com.br/portas-logicas/>. Acesso em: fev. 2021.

TECDICAS: Como descrever um circuito lógico algebricamente. 17 nov. 2019. Disponível em: <https://tecdicas.com/como-descrever-um-circuito-logico-algebricamente/>. Acesso em: fev. 2021.

George H. Mealy. In: WIKIPÉDIA: a enciclopédia livre. 14 ago. 2020. Disponível em: https://en.wikipedia.org/wiki/George_H._Mealy. Acesso em: fev. 2021.

Máquina de estados finita. In: WIKIPÉDIA: a enciclopédia livre. 19 jun. 2020. Disponível em: https://pt.wikipedia.org/wiki/M%C3%A1quina_de_estados_finita. Acesso em: fev. 2021.

BINARY Numbers in Ancient India. . Disponível em: <http://home.ica.net/roy-manju/Binary.htm>. Acesso em: fev. 2021.

MATHIGON: Timeline of Mathematics. Disponível em: <https://pt.mathigon.org/timeline/pingal>. Acesso em: fev. 2021.

ESCOLA, Equipe Brasil. "Álgebra Booleana"; Brasil Escola. Disponível em: <https://brasilecola.booleana.htm>. Acesso em: fev. de 2021.

PELEGRINI, Christian Hugo. Claude Elwood Shannon e A Symbolic Analysis of Relay and Switching Circuits: tornando o computador uma máquina semiótica. Contemporanea, v. 7, n. 1, jun. 2019. Disponível em: <https://periodicos.ufba.br/index.php/contemporaneaposcom/a>. Acesso em: fev. 2021.