

ID: 133691

Máquina de Estados Contador através de uma Máquina de Moore

São José dos Campos - Brasil

Outubro de 2019

ID: 133691

Máquina de Estados

Contador através de uma Máquina de Moore

Relatório apresentado à Universidade Federal de São Paulo para apresentar o desenvolvimento de um contador por meio de Máquinas de Estados.

Docente: Prof. Dr. Lauro Paulo da Silva Neto

Universidade Federal de São Paulo - UNIFESP

Instituto de Ciência e Tecnologia - Campus São José dos Campos

São José dos Campos - Brasil

Outubro de 2019

Resumo

Este relatório apresentará o desenvolvimento e suas conclusões em relação ao contador, crescente e decrescente, através do uso de máquina de estados do tipo Moore. O contador será de uma sequência preestabelicida(2-4-6-1-3-6-9-0-5) e deverá contar a cada 1 segundo, isto é, na frequência de 1 Hz, além disso, o contador deverá manter o número e também apagar os leds do *display* de 7 segmentos do kit FPGA. Todo o projeto será desenvolvido no *Quartus Prime*, aplicativo desenvolvido para mapear circuitos lógicos.

Palavras-chaves: contador. máquina de estados. Moore.

Lista de ilustrações

Figura 1 – <i>Display</i> de 7 segmentos - Ânodo comum	11
Figura 2 – <i>Display</i> de 7 segmentos - Representação dos números	11
Figura 3 – <i>Latch</i> com Set e Reset	12
Figura 4 – <i>Flip-Flop</i> D	13
Figura 5 – <i>Flip-Flop</i> T	14
Figura 6 – Máquina de Moore	14
Figura 7 – Contador Assíncrono	15
Figura 8 – Contador Síncrono	16
Figura 9 – Decodificador BCD	20
Figura 10 – Conversor do <i>clock</i> para 1 Hz	20
Figura 11 – Diagrama de estados	21
Figura 12 – Circuito de $Q'0$	24
Figura 13 – Circuito de $Q'1$	24
Figura 14 – Circuito de $Q'2$	25
Figura 15 – Circuito de $Q'3$	25
Figura 16 – Circuito de Saída	26
Figura 17 – Contador com a máquina de Moore	26
Figura 18 – Contador Final	27
Figura 19 – <i>Waveform</i> com Up = 0 e Down = 0	29
Figura 20 – <i>Waveform</i> com Up = 1 e Down = 0	29
Figura 21 – <i>Waveform</i> com Up = 0 e Down = 1	29
Figura 22 – <i>Waveform</i> com Up = 1 e Down = 1	30
Figura 23 – <i>Waveform</i> com combinações aleatórias	30

Lista de tabelas

Tabela 1 – Tabela de estados do <i>Latch</i>	13
Tabela 2 – Tabela de decodificação para o <i>display</i> de 7 segmentos	19
Tabela 3 – Tabela de estados.	22
Tabela 4 – Tabela de transição de estados e saídas.	22
Tabela 5 – Tabela de expressões dos próximos estados	23
Tabela 6 – Tabela de expressões para as saídas	23
Tabela 7 – Tabela de conexão de fios.	23

Sumário

1	INTRODUÇÃO	7
2	OBJETIVOS	9
2.1	Geral	9
2.2	Específico	9
3	FUNDAMENTAÇÃO TEÓRICA	11
3.1	Kit FPGA - <i>Display</i> de 7 segmentos	11
3.2	<i>Flip-Flop's</i>	12
3.2.1	<i>Latch</i>	12
3.3	Máquina de Moore	14
3.4	Conversor do <i>clock</i> para 1 Hz	15
3.4.1	Contador	15
4	DESENVOLVIMENTO	19
4.1	Decodificador BCD para o <i>display</i> de 7 segmentos	19
4.2	Conversor do <i>clock</i> para 1Hz	20
4.3	Máquina de Moore - Contador	21
4.3.1	Cirucito de entrada	23
4.3.2	Circuito de saída	26
4.3.3	Contador com a Máquina de Moore	26
4.3.4	Circuito Final	27
5	RESULTADOS OBTIDOS E DISCUSSÕES	29
6	CONSIDERAÇÕES FINAIS	31
	REFERÊNCIAS	33

1 Introdução

A utilização de máquinas de estados para criação de projetos pode ser bem eficiente comparado à outros métodos, tomando como exemplo, o próprio contador, criar apenas um contador crescente é bem simples sem o uso de máquina de estados, basta apenas utilizar um circuito com *flip-flop's* do tipo t encadeados; no entanto quando se trata de mais funcionalidades o circuito pode ficar bem mais complexo e de difícil interpretação.

O aprendizado em relação aos circuitos digitais é essencial para o desenvolvimento de variados tipos de circuitos, em que pode-se reduzir sua complexidade e facilitar a compreensão do projeto.

Sendo assim, resolveremos o problema do contador com uso de Máquina de Estados do tipo Moore.

2 Objetivos

2.1 Geral

Desenvolver um contador por meio de Máquina de Moore, contendo as funções de crescer, decrescer, manter e apagar o estado do contagem, que serão controladas por *switches*.

2.2 Específico

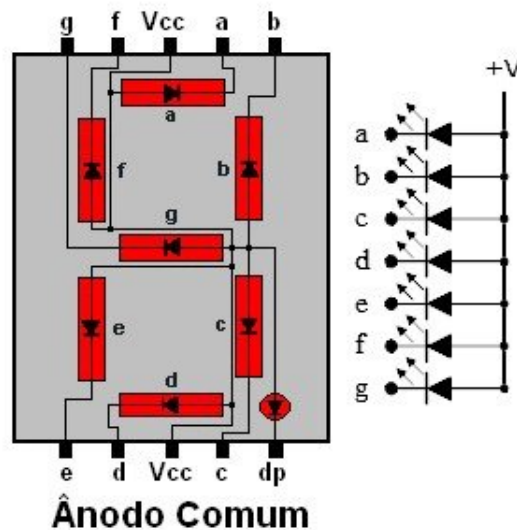
1. Utilizar o *Software Quartus Prime* para realizar o projeto.
2. Desenvolver os circuitos combinacionais de entrada e saída da máquina de estados, com o uso de duas entradas, *up* e *down*.
3. Desenvolver um circuito para a criação de um *clock* de 1 Hz, visto que do kit FPGA que é utilizado contém um *clock* de 50MHz.
4. Desenvolver um circuito de decodificação do circuito de saída da máquina de Moore para o *display* de 7 segmentos do kit.

3 Fundamentação Teórica

3.1 Kit FPGA - *Display* de 7 segmentos

O kit FPGA utilizado nos laboratórios é o Altera DE2-115, e seu display é do tipo ânodo comum como na Figura 1.

Figura 1 – *Display* de 7 segmentos - Ânodo comum

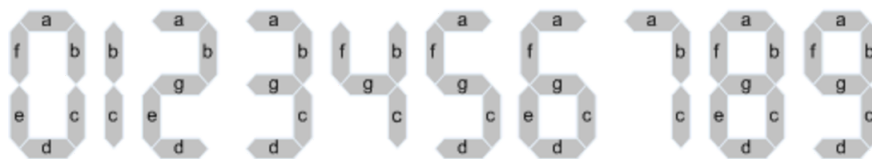


Fonte: <www.electronica-pt.com/eletronica-digital/display-7-segmentos>

Nota-se, pela figura, que todos os segmentos estão ligados ao VCC, isto é, ao nível alto, portanto, para que se possa acender o led do segmento desejado é necessário aplicar o nível lógico baixo (GND ou 0).

Sendo assim podemos construir todos os números, basta tomar as letras desejadas da figura 1 e formamos os números, segundo à figura 2.

Figura 2 – *Display* de 7 segmentos - Representação dos números



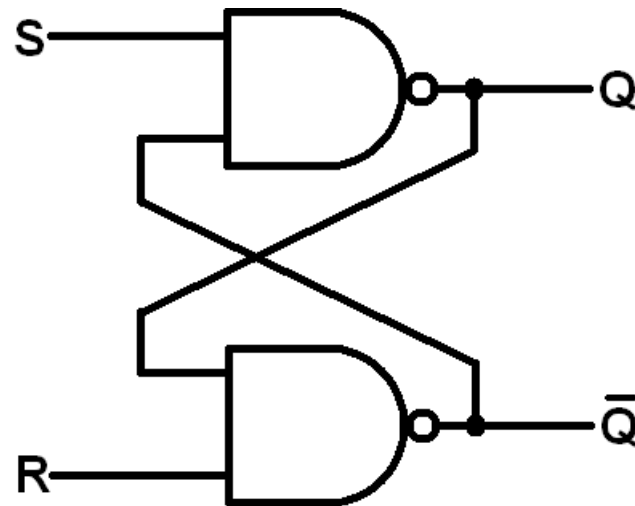
Fonte: <https://www.electronics-tutorials.ws/combinational/comb_6.html>

3.2 Flip-Flop's

Flip-Flop's ou elementos de memória são um tipo de circuito que utilizam de *latch's*. *Latch* também é um tipo de circuito de memória, no entanto, elas não se utilizam de *clocks*.

3.2.1 Latch

Figura 3 – Latch com Set e Reset



Fonte: <<https://learn.sparkfun.com/tutorials/logicblocks-experiment-guide/6-sr-latch>>

A figura 3 apresenta um latch básico, com portas nand, com set e reset. Analisando o circuito(a análise é feita pelo estudo das saídas) podemos definir quatro estados:

1. $S = 1, R = 1$:

Se $q = 0$, o NAND de cima produzirá 0, enquanto o de baixo produzirá saída 1. A mesma coisa se $q = 1$, o NAND de cima produzirá a saída 1 e a de baixo produzirá a saída 0; portanto, nada se altera.

2. $S = 0, R = 1$:

Se $q = 0$; então o NAND de baixo produzirá a saída 0, e em seguida, como este liga-se ao NAND de cima, teremos $q = 1$, e, por fim, o NAND de baixo produzirá a saída 0.

Se $q = 1$, então o NAND de baixo produzirá a saída 0, e este não irá alterar a saída q , que está relacionada à saída de cima.

3. $S = 1, R = 0$:

Se $q = 0$, então temos que este irá produzir nível alto no NAND de baixo, que, então criará uma saída de nível baixo no q , portanto não alterando nenhum estado do circuito.

Se $q = 1$, então, este irá alterar o estado de $-q$ para 1, e este, em seguida irá alterar o estado de q para o nível baixo, que portanto altera-se o estado final.

4. $S = 0, R = 0$:

Este estado, segundo o Tocci(1) é considerado um estado inválido, uma vez que produz saída $q = 1$ e $-q = 1$, que eram pra ser estados complementares, portanto, não devemos considerá-lo.

Portanto, obtemos:

Tabela 1 – Tabela de estados do *Latch*.

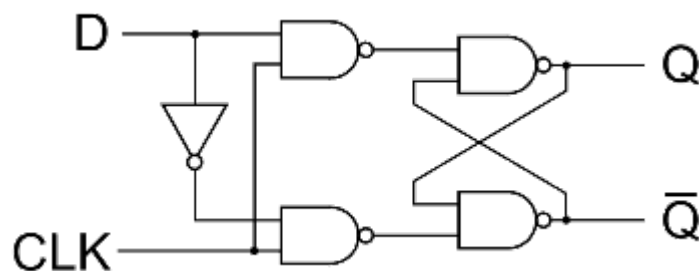
Set	Reset	Q	\overline{Q}
1	1	Mantém	Mantém
1	0	0	1
0	1	1	0
0	0	Inválido	Inválido

Fonte:(1), tabela feita pelo Autor.

Dado a definição dos *latch's*, pode-se definir o conceito dos *flip-flop's*. Apesar, da existencia de vários tipos de *flip-flop's* estudaremos em particular, dois: *flip-flop D* e *flip-flop T*:

1. *Flip-Flop D*:

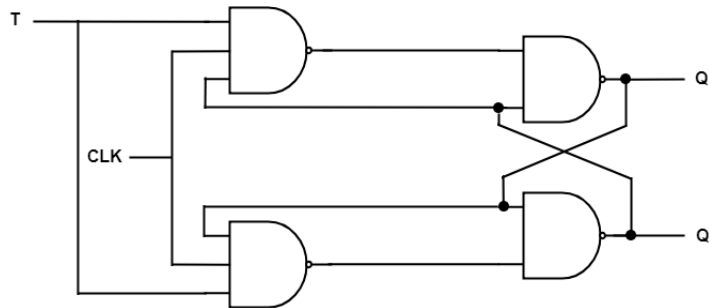
Figura 4 – *Flip-Flop D*



Fonte:<<https://electronics.stackexchange.com/questions/346969/what-is-the-output-when-d-and-c-on-d-flip-flop-are-connected>>

A figura 4 representa um *flip-flop D*, com um *clock*. Nota-se que ele possui um *latch* em sua frente, portanto ele também é um elemento de memória, no entanto as transições de estado só ocorrem durante a subida do clock, o que caracteriza o *flip-flop*.

2. *Flip-Flop T*:

Figura 5 – *Flip-Flop T*

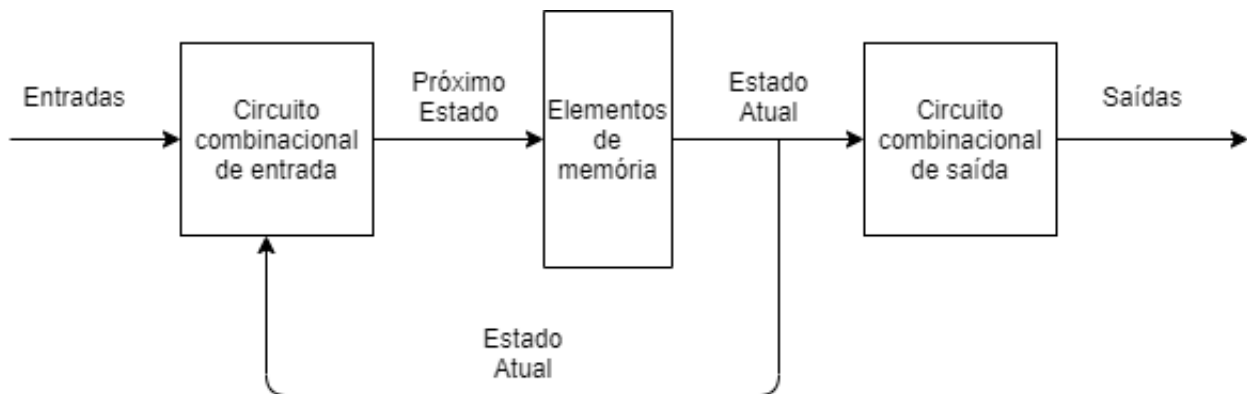
Fonte: <<https://www.csetutor.com/types-of-flip-flops-in-digital-electronics/>>

A figura acima representa o *flip-flop T*, apesar de parecido com o *flip-flop D*, nota-se que entrada dos primeiros nands (que são alimentados pelo *clock*) são realimentados pelas saídas q e $-q$, portanto, provocará na intercalação dos sinais, pois, se tomarmos como exemplo, $q = 1$, temos que o o primeiro NAND de baixo produzirá saída 1, na subida do *clock*, que, em seguida, provocará saída $-q = 1$, e portanto tornará $q = 0$, repetindo esse ciclo.

3.3 Máquina de Moore

Segundo Tocci(1), “máquina de estados se refere a um circuito que sequencia um conjunto de estados predeterminados controlados por um *clock* e outros sinais de entrada.” Seguindo essa definição, a máquina de Moore é uma máquina que sequencia estados, na qual os próximos estados são definidos a partir do estado anterior e as entradas, no entanto, a saída depende somente do estado em que está, não ligando às entradas. Ela pode ser descrita como na figura 6.

Figura 6 – Máquina de Moore



Fonte: O Autor, desenvolvido em (2)

A primeira caixa da figura contém um circuito que combina as entradas e o estado atual, com isso, determinamos a função do próximo estado. Em seguida essa função de próxima estado passa por elementos de memória, os *flip-flop's*, que guardam o estado atual da máquina, e que é atualizado a cada ciclo do *clock*; por fim as saídas do estado atual passam pelo circuito combinacional de saída, que decodificam na saída desejada.

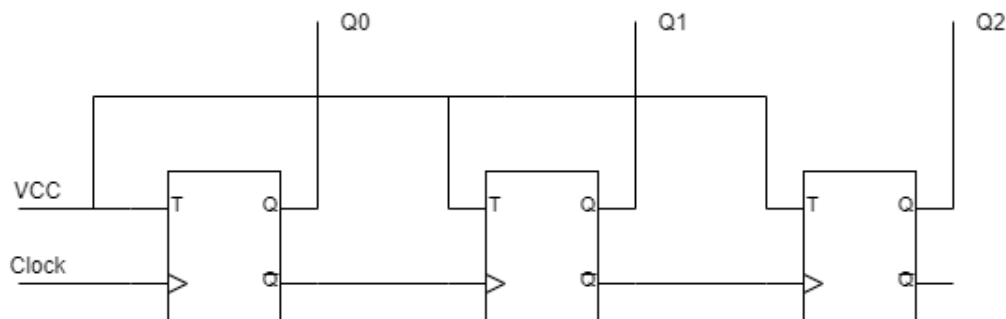
3.4 Conversor do *clock* para 1 Hz

O kit possui um clock de 50 MHz, sendo assim devendo converter para 1 Hz. Um método possível é utilizarmos um divisor de frequência, que tem como princípio o contador.

3.4.1 Contador

Um contador é um tipo de circuito estudado em circuitos digitais, na qual conta números, mas em binário. Esse circuito se utiliza dos *flip-flops*, e este em específico é do tipo T. Pode ser do tipo assíncrono ou síncrono, sendo o primeiro, apresentado um *delay* em cada transição de número e o outro, não apresentando esse delay. Podemos observar o primeiro tipo na Figura 7.

Figura 7 – Contador Assíncrono

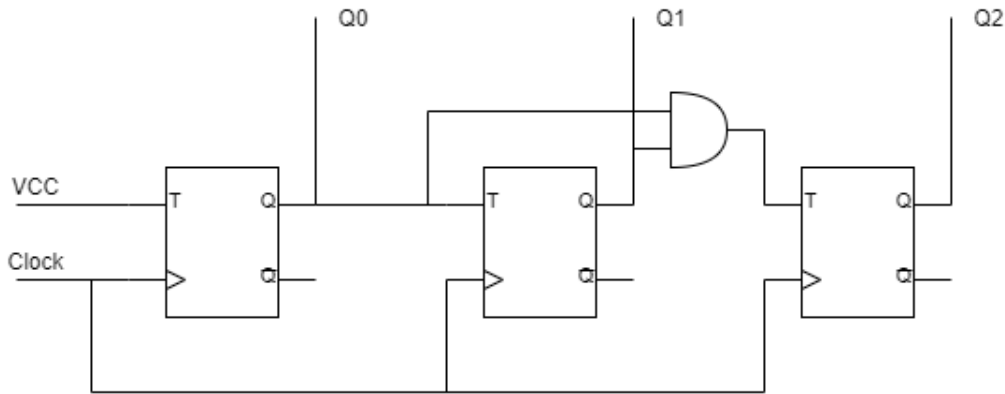


Fonte: O Autor, desenvolvido em (2)

Nota-se que nem todos os *flip-flop's* estão ligados com o clock principal, e sim com a saída negada do elemento de memória, isso caracteriza a assincronicidade do circuito, uma vez que cada *clock* do *flip-flop* depende do anterior, podendo gerar o atraso. Esse contador é crescente e denominado de resto-8, pois conta 8 valores, do 0 até 7, em binário, sendo Q0 o *bit* menos significativo e Q2, o mais significativo.

Já a figura 8 apresenta o contador síncrono.

Figura 8 – Contador Síncrono



Fonte: O Autor, desenvolvido em (2)

Nesse contador temos todos os *flip-flop's* ligados ao ao mesmo *clock*, o que caracteriza a sincronicidade do circuito. Esse contador é crescente e resto-8.

No caso, temos um *clock* de 50 MHz, então:

$$f = \frac{1}{t},$$

Sendo f a frequência e t o tempo do *clock*. Então:

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

E portanto:

$$t = \frac{1}{50 \cdot 10^6} \Rightarrow t = 20 \text{ ns}$$

Se em um ciclo temos 20 ns, precisa-se calcular quantos ciclos são necessários para contar 1 s.

$$t' = t \times x$$

,

Sendo x , a quantidade de ciclos, t' o tempo de desejado e t o tempo do *clock* original.

$$x = \frac{t'}{t} \Rightarrow x = \frac{1 \text{ s}}{20 \text{ ns}} = 50 \cdot 10^6 \text{ ciclos}$$

Agora, necessita-se descobrir a quantidade de *flip-flop's*(n) para contar até 50 milhões, para isso:

$$\lceil 2^n \rceil = 50 \cdot 10^6 \Rightarrow n = 26$$

Portanto, precisamos de 26 *flip-flop's* para contar até 50 milhões, o que permitirá criar o conversor preciso de 1 Hz do Contador da Máquina de Moore.

4 Desenvolvimento

4.1 Decodificador BCD para o *display* de 7 segmentos

Para o *display* de 7 segmentos tem-se que converter os *bits* em binário para a respectiva letra do *display*, por exemplo, como o decodificador é de 4 *bits*, toma-se 0000, portanto suas respectivas letras, na saída, são 0000001(a-g), lembrando que o *display* é anôdo comum, portanto o led só acende ao nível baixo. Para os números maiores que 10 não se deve aparecer nada no *display* portanto suas sequências serão 1111111(a-g).

Portanto se tomarmos a tabela 2 (sendo e3, o *bit* mais significativo e e0, o menos), temos que:

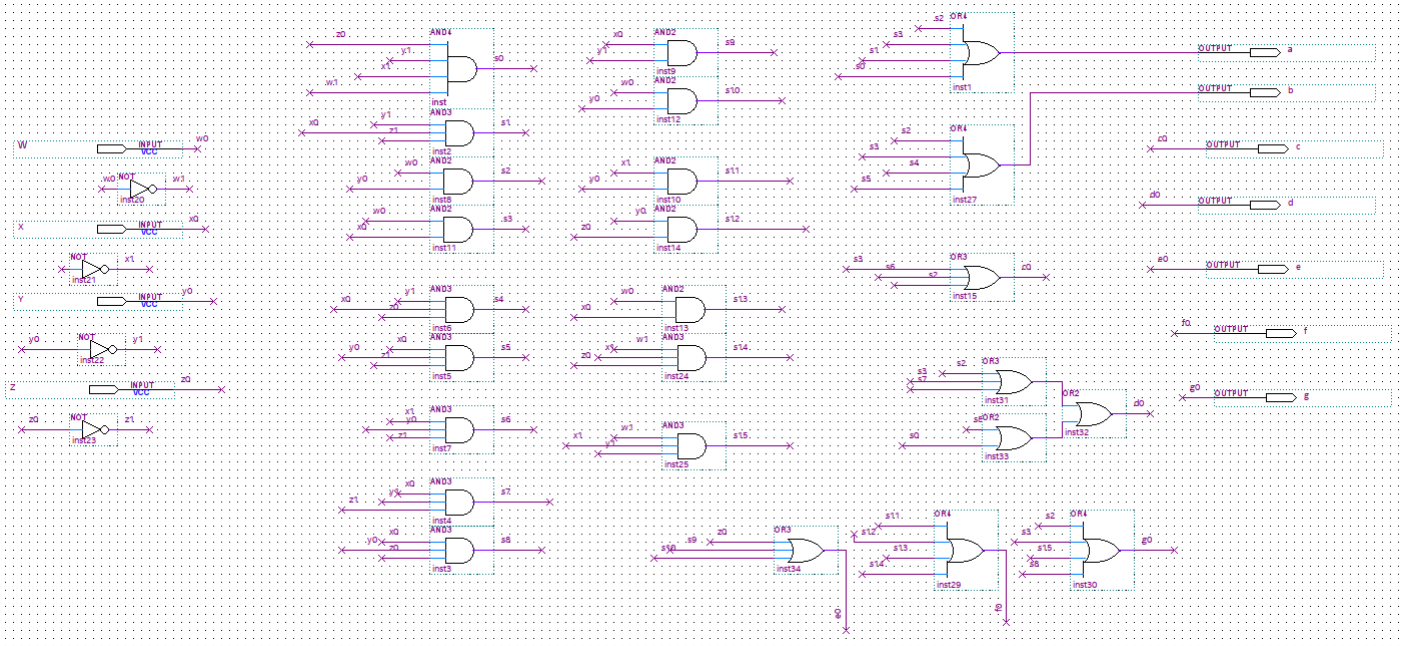
Tabela 2 – Tabela de decodificação para o *display* de 7 segmentos

Entradas				Saídas						
e3	e2	e1	e0	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	1	0	0	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0
0	0	1	1	0	0	0	0	1	1	0
0	1	0	0	1	0	0	1	1	0	0
0	1	0	1	0	1	0	0	1	0	0
0	1	1	0	0	1	0	0	0	0	0
0	1	1	1	0	0	0	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	1	0	0
1	0	1	0	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1

Fonte: O Autor

Utilizando o site 32x8.com(3) podemos formar o circuito da figura 9 (W é a entrada mais significativa e Z, a menos).

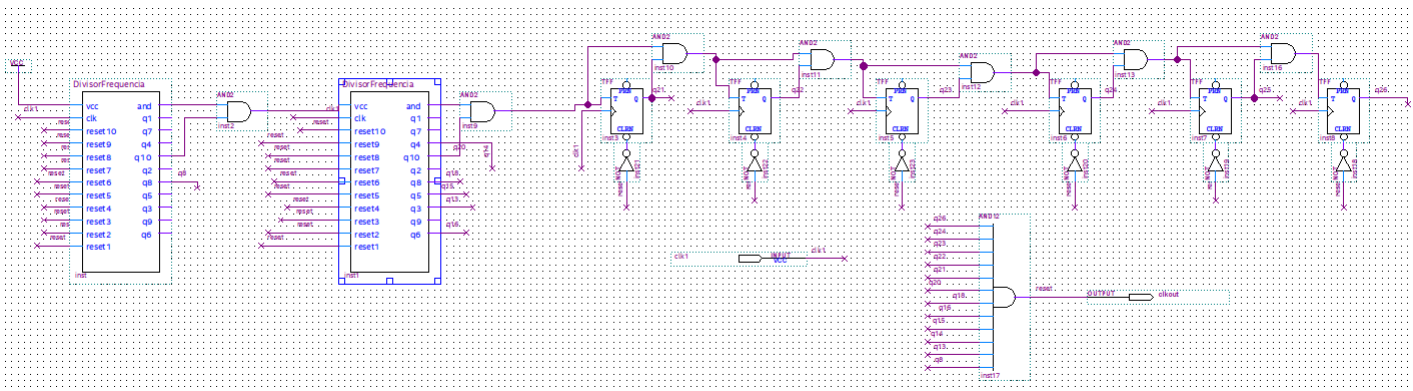
Figura 9 – Decodificador BCD



Fonte: O Autor, desenvolvido no *software Quartus Prime*

4.2 Conversor do *clock* para 1Hz

Para o conversor, já foram feitas as observações quanto à quantidade de *flip-flop's* para o divisor de frequência (seção 3.4), no entanto se utilizarmos todos os *flip-flop's* temos $2^{26} = 67108864$, que não é o número desejado, para isso precisa-se contar até 50 milhões, portanto, toma-se os *bits* que formam 50 milhões em binário. Esse *bits* são 10111110101111000010000000 (do *bit* mais significativo ao menos), basta então tomar esses *bits* de cada *flip-flop* do divisor de frequência, obtendo o circuito da figura 10.

Figura 10 – Conversor do *clock* para 1 Hz

Fonte: O Autor, desenvolvido no *software Quartus Prime*

Dentro de cada caixa do “DivisorFrequencia” da figura 10 temos um contador com 10 *flip-flop's*. A saída “clkout” apresenta o “clock” de 1 Hz.

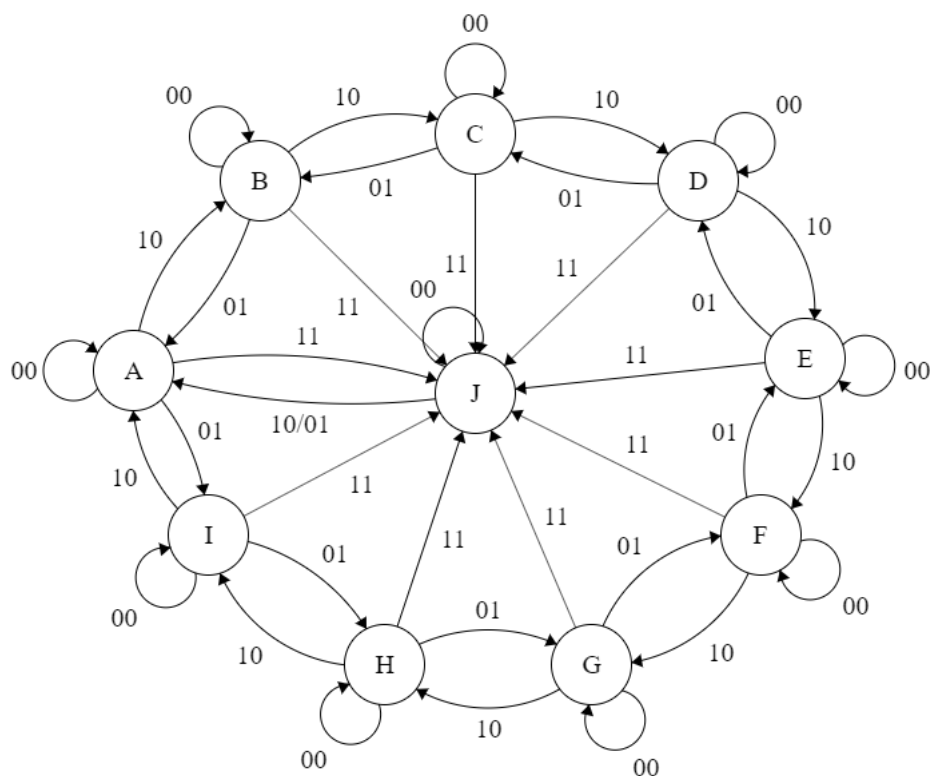
4.3 Máquina de Moore - Contador

Para projetar o contador, primeiro, precisamos definir os estados da Máquina de Estados. Cada estado representará o número do contador, mais um estado extra, que apaga o *display*, portanto, 10 estados. As transições dos estados dependem das entradas, sendo elas o *up* e o *down*, tem-se então, quatro opções:

1. $up = 0$ e $down = 0$: Mantém o estado;
2. $up = 1$ e $down = 0$: Contador crescente;
3. $up = 0$ e $down = 1$: Contador decrescente;
4. $up = 1$ e $down = 1$: Apagado.

Construindo-se um diagrama de estados, obtemos a figura 11

Figura 11 – Diagrama de estados



Fonte: O Autor, desenvolvido em <<http://madebyevan.com/fsm/>>

Definindo cada estado com uma codificação binária temos a tabela 3.

Tabela 3 – Tabela de estados.

Estado	Codificação Binária
A	0000
B	0001
C	0010
D	0011
E	0100
F	0101
G	0110
H	0111
I	1000
J	1001

Fonte: O Autor.

Definindo a tabela de transição de estados, obtém-se a tabela 4.

Tabela 4 – Tabela de transição de estados e saídas.

Estado atual	Próximo Estado				Saídas
$Q3, Q2, Q1, Q0$	UpDown(00)	UpDown(01)	UpDown(10)	UpDown(11)	$S3, S2, S1, S0$
0000	0000	1000	0001	1001	0010(2d)
0001	0001	0000	0010	1001	0100(4d)
0010	0010	0001	0011	1001	0110(6d)
0011	0011	0010	0100	1001	0001(1d)
0100	0100	0011	0101	1001	0011(3d)
0101	0101	0100	0110	1001	0110(6d)
0110	0110	0101	0111	1001	1001(9d)
0111	0111	0110	1000	1001	0000(0d)
1000	1000	0111	0000	1001	0101(5d)
1001	1001	0000	0000	1001	1010(10d)

Fonte: O Autor. Q3 é o *bit* mais significativo e Q0, o menos. O mesmo se aplica as saídas.

Utilizando o site (3) novamente pode-se rapidamente achar as expressões lógicas de cada função de próximo estado e das saídas.

Tabela 5 – Tabela de expressões dos próximos estados

Próximo Estado	Expressão lógica
$Q'0$	$UD + \bar{Q}3\bar{Q}0U + Q0\bar{U}\bar{D} + Q1\bar{Q}0D + Q2\bar{Q}0D + Q3\bar{Q}0D$
$Q'1$	$Q1\bar{Q}0\bar{D} + Q1Q0\bar{U} + Q3\bar{Q}0\bar{U}D + \bar{Q}3\bar{Q}1Q0U\bar{D} + Q2\bar{Q}1\bar{Q}0\bar{U}D$
$Q'2$	$Q2\bar{Q}1\bar{D} + Q2\bar{Q}0\bar{D} + Q2Q1\bar{U} + Q2Q0\bar{U} + Q3\bar{Q}0\bar{U}D + \bar{Q}2Q1Q0U\bar{D}$
$Q'3$	$UD + Q3\bar{U}\bar{D} + Q2Q1Q0U + \bar{Q}3\bar{Q}2\bar{Q}1\bar{Q}0D$

Fonte: O Autor. U = Up; D = Down

Tabela 6 – Tabela de expressões para as saídas

Saída	Expressão lógica
$S0$	$Q2\bar{Q}0 + Q3\bar{Q}0 + \bar{Q}2Q1Q0$
$S1$	$Q2\bar{Q}1 + Q3Q0 + \bar{Q}3\bar{Q}2\bar{Q}0$
$S2$	$Q3\bar{Q}0 + \bar{Q}3\bar{Q}1Q0 + \bar{Q}2Q1\bar{Q}0$
$S3$	$Q3Q0 + Q2Q1\bar{Q}0$

Fonte: O Autor.

Portanto, temos o circuito combinacional de entrada (tabela 5) e o circuito combinacional de saída (tabela 6).

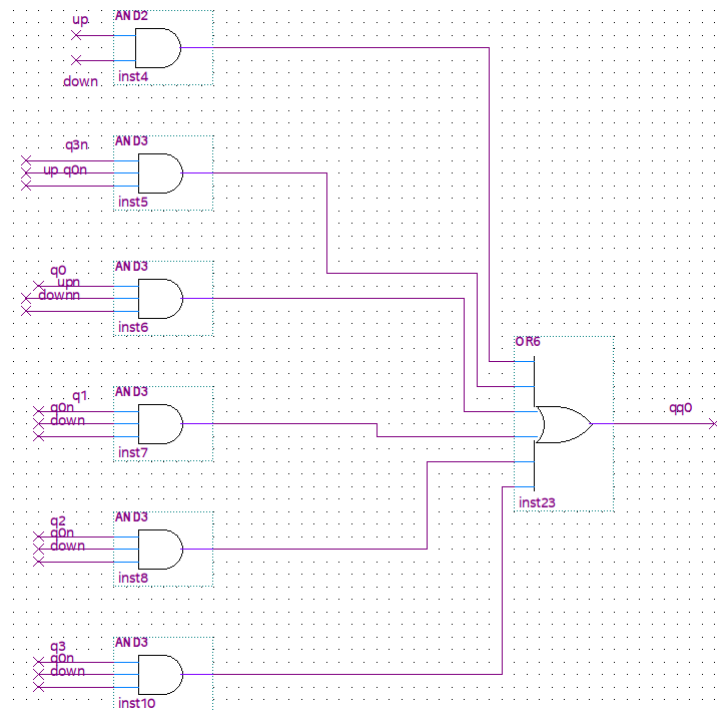
4.3.1 Circuito de entrada

Usando o *software Quartus Prime* obtemos os seguintes circuitos (vide a tabela 7 para conferir as entradas e saídas das portas lógicas):

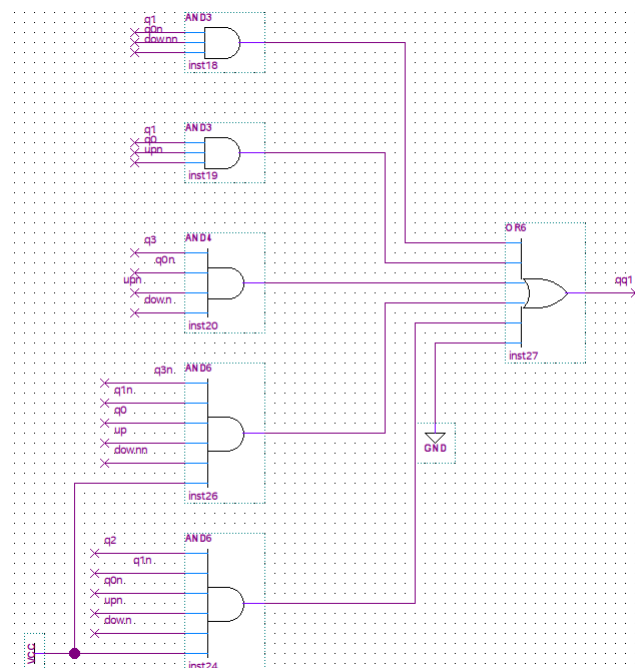
Tabela 7 – Tabela de conexão de fios.

Label do fio	Sigificado
q0/q0n/qq0	$Q0/\bar{Q}0/Q'0$
q1/q1n/qq1	$Q1/\bar{Q}1/Q'1$
q2/q2n/qq2	$Q2/\bar{Q}2/Q'2$
q3/q3n/qq3	$Q3/\bar{Q}3/Q'3$
up/upn	U/\bar{U}
down/downn	D/\bar{D}

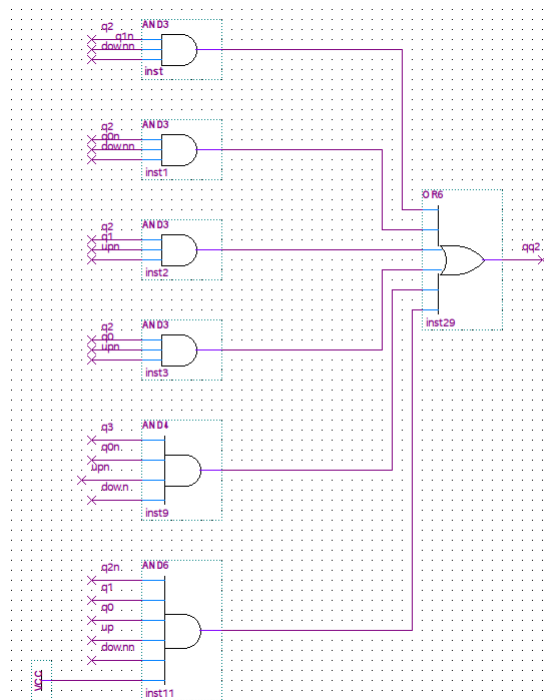
Fonte: O Autor.

Figura 12 – Circuito de $Q'0$ 

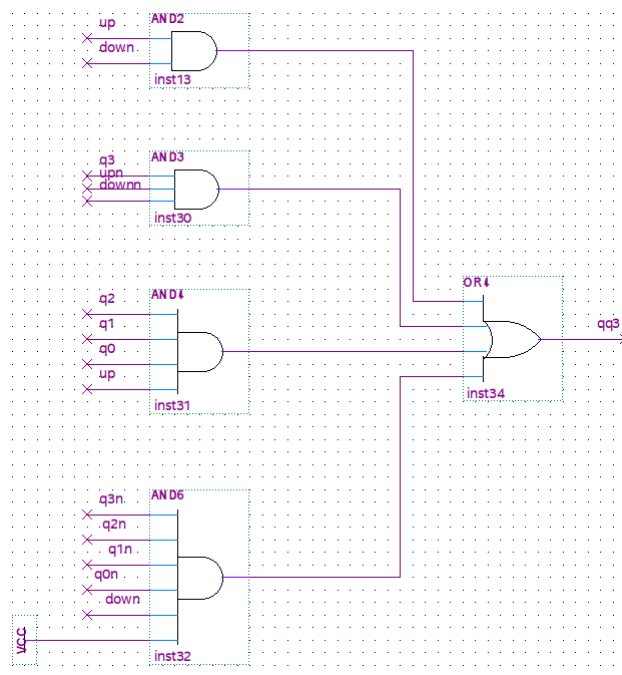
Fonte: O Autor

Figura 13 – Circuito de $Q'1$ 

Fonte: O Autor

Figura 14 – Circuito de $Q'2$ 

Fonte: O Autor

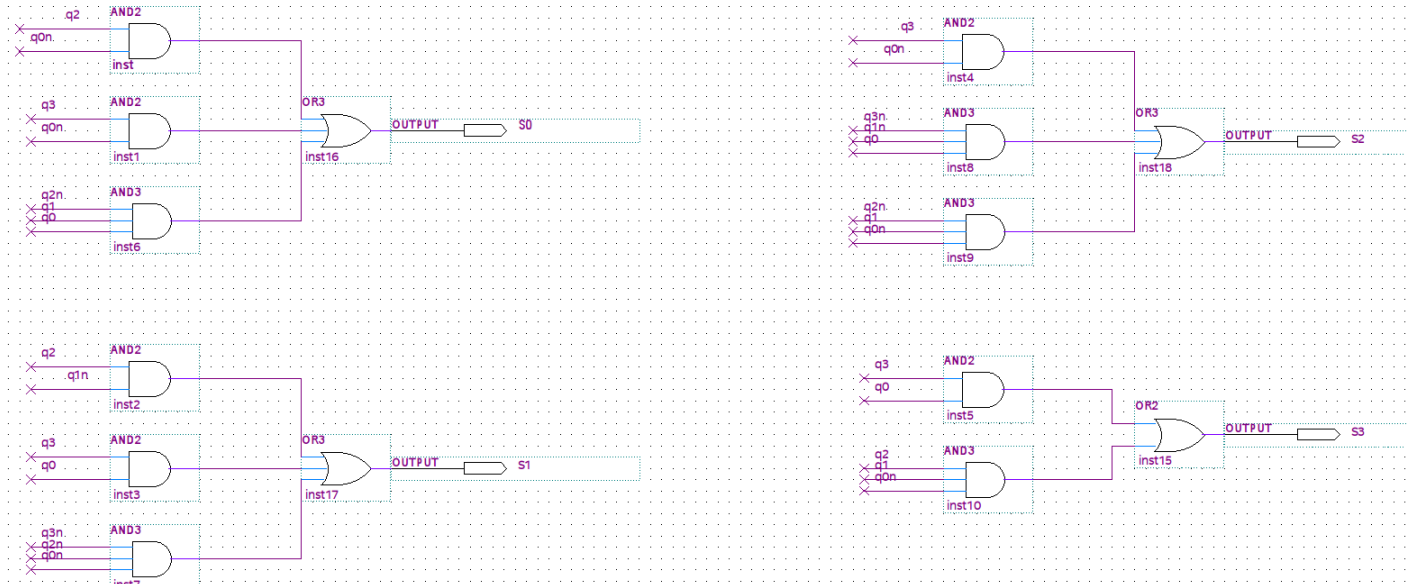
Figura 15 – Circuito de $Q'3$ 

Fonte: O Autor

4.3.2 Circuito de saída

Para o circuito de saída temos a figura 16 (vide tabela 7 para os *inputs*).

Figura 16 – Circuito de Saída

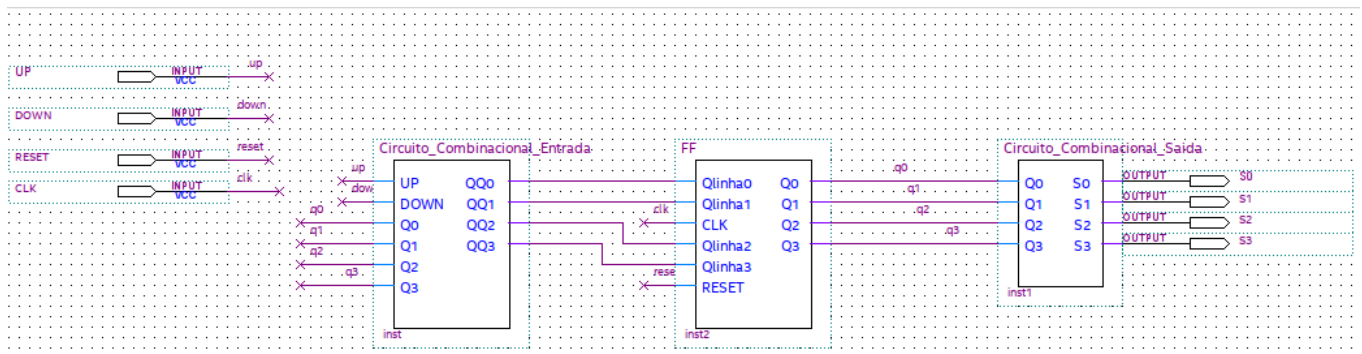


Fonte: O Autor

4.3.3 Contador com a Máquina de Moore

Juntando os circuitos de entrada e saída e inserindo os elementos de memória obtemos o contador, por máquina de estados.

Figura 17 – Contador com a máquina de Moore



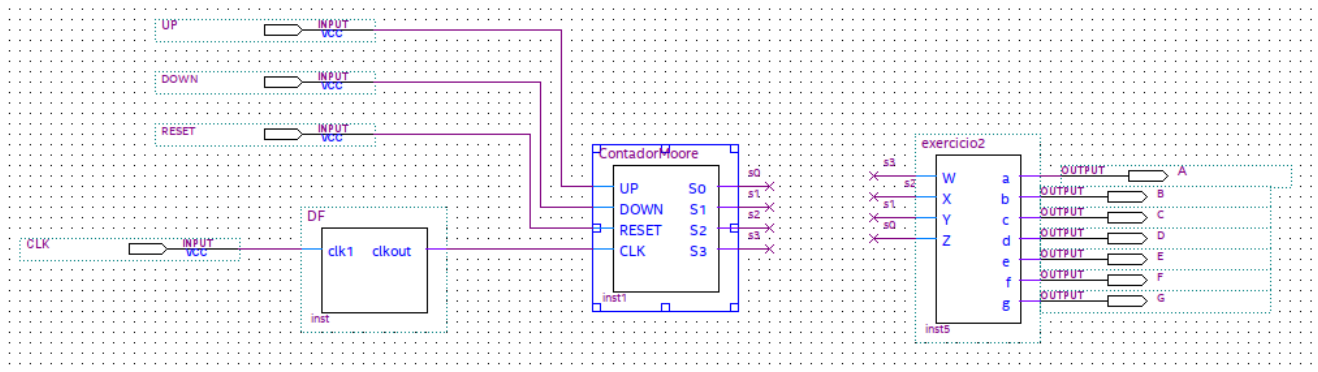
Fonte: O Autor

Dentro da caixa “FF” temos os elementos de memória, os *flip-flop’s* do tipo D, cada *flip-flop* carrega uma entrada de estado, que é atualizada a cada ciclo do clock. Note a entrada “RESET”, ela aplica a saída 0 em todos os *flip-flop’s* em que, portanto, retorna ao primeiro estado(A).

4.3.4 Circuito Final

Ao juntarmos tudo, obtemos o circuito da figura 18.

Figura 18 – Contador Final



Fonte: O Autor

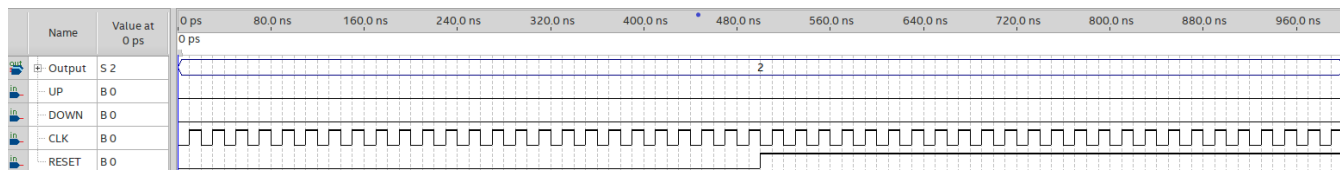
Dentro da caixa “DF” temos o conversor do *clock* para 1 Hz, em seguida, na caixa “ContadorMoore” temos o contador pela máquina de estados e por fim, na caixa “exercicio2” temos o decodificador BCD para o *display*.

5 Resultados Obtidos e Discussões

Feito o contador, testa-se o circuito no kit FPGA, e obtém-se o resultado. Podemos imprimir as formas de onda também para cada tipo de entrada, demonstrando a funcionalidade do circuito.

- Up=0, Down = 0.

Figura 19 – *Waveform* com Up = 0 e Down = 0

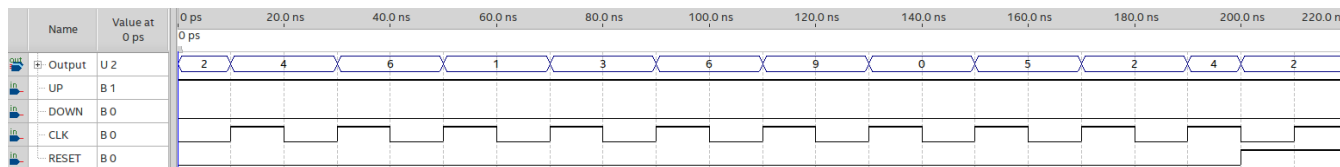


Fonte: O Autor

Nota-se que ele mantém no primeiro estado, portanto está com funcionalidade correta.

- Up=1, Down = 0.

Figura 20 – *Waveform* com Up = 1 e Down = 0

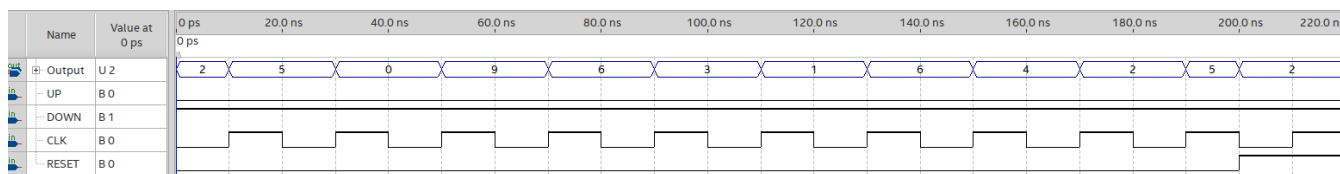


Fonte: O Autor

Neste caso, o contador está crescente, mantendo a sequência estabelecida, e com o “RESET” funcionando.

- Up=0, Down = 1.

Figura 21 – *Waveform* com Up = 0 e Down = 1

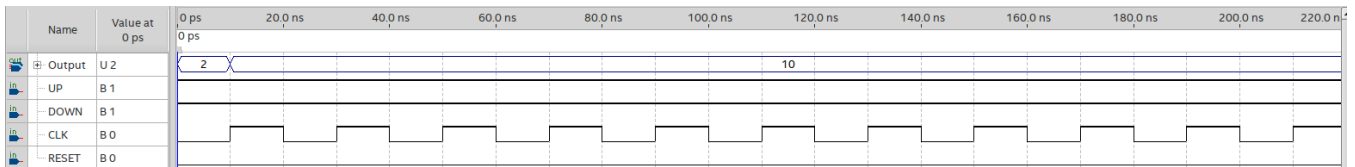


Fonte: O Autor

Aqui tem-se o contador decrescente, que também mantém a ordem preestabelecida, também com a funcionalidade do “RESET”.

- Up=1, Down = 1.

Figura 22 – *Waveform* com Up = 1 e Down = 1

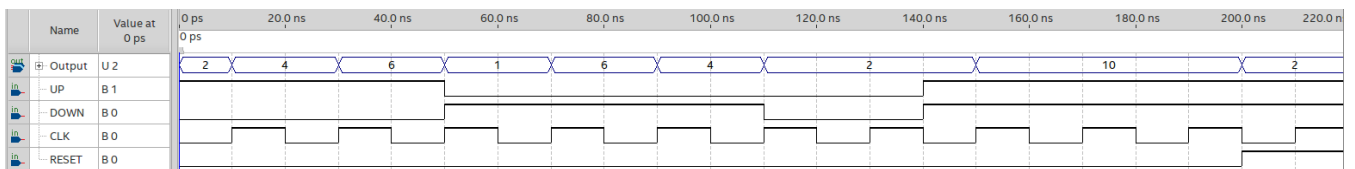


Fonte: O Autor

Quando temos esse estado a saída é 10d, uma vez que o decodificador BCD configura essa saída para apagar o *display*, portanto está correto também.

- Combinação Aleatória

Figura 23 – *Waveform* com combinações aleatórias



Fonte: O Autor

Por fim, temos a combinação aleatória, que demonstra a funcionalidade geral do circuito.

6 Considerações Finais

Com a conclusão do projeto podemos afirmar que o estudo aprofundado de circuitos digitais aumenta o repertório para a produção de circuitos complicados, no entanto, por se tratar de um circuito complexo, erros mínimos cometidos podem corromper o bom funcionamento do circuito, e muitas vezes, achar o erro é o mais complicado.

Finalmente, o uso de máquina de estados pode ser um bom viés para construção de diversos tipos de circuitos, pois permite atualização de dados em tempo constante e também, a implementação de diversas funcionalidades no circuito.

Referências

- 1 TOCCI, R. J.; WIDMER, N. S.; MOSS, G. L. *Sistemas Digitais: princípios e aplicações*. 11th edition. ed. São Paulo/SP, Brasil: Pearson, 2011. Citado 2 vezes nas páginas 13 e 14.
- 2 ALDER, G. *Free online diagram software for making flowcharts, process diagrams, org charts, UML, ER and network diagrams*. <<https://www.draw.io>>. Citado 3 vezes nas páginas 14, 15 e 16.
- 3 ONLINE Karnaugh map solver with circuit for up to 6 variables. <<http://www.32x8.com/index.html>>. Citado 2 vezes nas páginas 19 e 22.