

Gabriel Reale Marques de Oliveira
RA – 148401

PROJETO DE UMA MÁQUINA DE ESTADOS

Máquina de Moore

São José dos Campos – Brasil

2023

Gabriel Reale Marques de Oliveira
RA – 148401

PROJETO DE UMA MÁQUINA DE ESTADOS

Máquina de Moore

Relatório descrevendo o projeto de uma máquina de estados finitos conforme a arquitetura Moore. A máquina foi confeccionada para matéria de Laboratório de Sistemas Computacionais: Circuitos Digitais no ICT Unifesp

Universidade Federal de São Paulo – UNIFESP

Docente – Prof. Dr. Lauro Paulo da Silva Neto

Instituto de Ciência e Tecnologia – Campus São José dos Campos

São José dos Campos – Brasil

2023

Resumo

Este relatório descreve a implementação de uma máquina de estados finitos com arquitetura de Moore, essa é capaz de iterar por uma sequência de valores pré-definidos exibindo-os em um *display* de 7 segmentos. As transições da máquina são bidirecionais, circulares e periódicas (uma transição ocorre por segundo), o usuário é capaz de selecionar o sentido da transições, pausa-las, apagar o *display* ou reinicializar a máquina por meio de *switches*. A implementação foi feita para a família de FPGAs *Cyclone IV E*, por meio de diagramas esquemáticos e utilizando o software *Intel Quartus Prime 20.1 Lite Edition*.

Palavras-chaves: Circuito digital, Máquina de estados, Moore, FPGA

Lista de ilustrações

Figura 1 – Forma de onda de um <i>Flip-Flop</i> tipo T	13
Figura 2 – <i>Display</i> de 7 segmentos ânodo/cátodo comum	15
Figura 3 – Divisor de Frequência	17
Figura 4 – Detector de sequencia	17
Figura 5 – Cascata de tipos T	18
Figura 6 – <i>Flip-flop</i> T de saída e circuito de <i>reset</i>	18
Figura 7 – Grafo da Máquina de Moore	20
Figura 8 – Circuito de estado	21
Figura 9 – Circuito de estado inicial	22
Figura 10 – Padrão de conexões	23
Figura 11 – Máquina de Moore	23
Figura 12 – Máquina de Moore com zoom	24
Figura 13 – Conversor BCD com zoom	24
Figura 14 – Conversor BCD com zoom parte 2	25
Figura 15 – Conversor BCD	26
Figura 16 – Circuito final	27
Figura 17 – Circuito final, entradas e divisor de frequência	27
Figura 18 – Circuito final, conversor binário decimal e saídas	27

Lista de tabelas

Tabela 1 – <i>Flip-Flop T</i>	13
Tabela 2 – Contagem	14
Tabela 3 – Comandos da máquina	19

Sumário

1	INTRODUÇÃO	9
2	OBJETIVOS	11
2.1	Geral	11
2.2	Específico	11
3	FUNDAMENTAÇÃO TEÓRICA	13
3.1	Divisor de Frequência	13
3.2	Display de 7 Segmentos	14
3.3	Máquina de Moore	15
4	DESENVOLVIMENTO	17
4.1	Divisor de Frequência	17
4.2	Máquina de Moore	18
4.3	Conversor Binário Decimal	24
4.4	Circuito Final	26
5	RESULTADOS E DISCUSSÕES	29
6	CONSIDERAÇÕES FINAIS	31
	REFERÊNCIAS	33

1 Introdução

A eletrônica digital existe na fronteira entre diversas áreas da engenharia, englobando desde a matemática discreta até a arquitetura de computadores e comunicação digital, sendo um pilar fundamental da computação. Portanto o autor, como engenheiro em formação, acredita que um estudo aprofundado dessa disciplina é essencial.

Com essas ideias em mente, o presente relatório descreve uma máquina de estados Moore capaz de transitar por uma sequência arbitrária de valores, cuja a implementação abrange diversos conceitos de eletrônica digital.

2 Objetivos

2.1 Geral

Implementar uma Máquina de Moore capaz de iterar por uma sequência pré-determinada de forma bidirecional, circular e periódica isto é:

1. **Bidirecional** : O usuário deve ser capaz de escolher o sentido da iteração, crescente ou decrescente;
2. **Circular** : Chegando ao final da sequência, essa deve ser reiniciada a depender do sentido da iteração;
3. **Periódica** : A máquina deve realizar uma iteração por segundo;

Finalmente, o usuário também deve ser capaz de apagar o *display* ,pausar e reiniciar a máquina, podendo fazê-los por meio de *switches*.

2.2 Específico

1. Projetar um divisor de frequência de converta o *clock* nativo do FPGA de 50Mhz para 1hz;
2. Projetar um conversor binário/decimal para um *display* de 7 segmentos;
3. Projetar a máquina de Moore com a sequencia desejada, o que envolve:
 - a) Projeto do grafo que descreve a máquina desejada;
 - b) Projeto do circuito capaz de descrever um estado genérico desse grafo;
 - c) Conexão correta dos circuitos de estado;

3 Fundamentação Teórica

3.1 Divisor de Frequência

Um divisor de frequência nada mais é que um circuito capaz de reduzir a frequência de um sinal de *clock* de forma controlada. Segue uma descrição de seu funcionamento.

Flip-flops do tipo T, quando ativos, alteram o estado de sua saída a cada borda de subida, conforme a [Tabela 1](#).

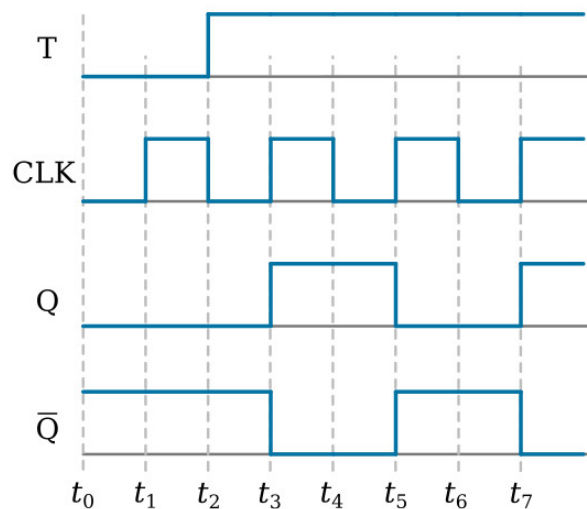
Tabela 1 – *Flip-Flop*
T

T	CLK	Q
1	↑	troca
X	X	mantém

Fonte: O autor.

Para que um tipo T produza um único período completo (1 borda de subida e 1 borda de descida) é necessário que o sinal de *clock* realize 2 períodos (2 bordas de subida e 2 bordas de decida), como na [Figura 1](#).

Figura 1 – Forma de onda de um *Flip-Flop* tipo T



Fonte: <https://ecstudiosystems.com/discover/textbooks/basic-electronics/flip-flops/t-flip-flop/>

Em outras palavras, ao receber 2 de oscilações completas, o *Flip-flop* T produz

apenas uma em sua saída. Essa relação de 2:1 permite que o período do sinal de saída seja dobrado, e conseqüentemente que a frequência caia pela metade, daí o nome divisor de frequência.

Tabela 2 – Contagem

B_2	B_1	B_0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Fonte: O autor.

Esse processo de divisão é análogo ao de contagem. Cada dígito em uma contagem binária possui uma frequência sobre a qual transita de 0 para 1.

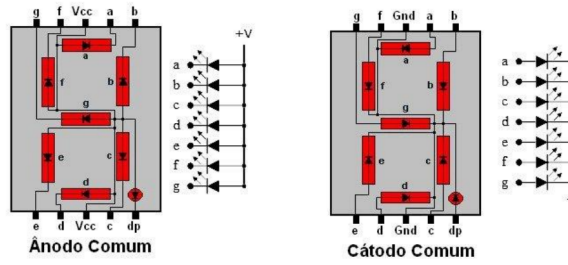
Na [Tabela 2](#) o dígito menos significativo transita de 0 para 1 a cada linha, seu sucessor a cada duas linhas, o sucessor desse a cada quatro, e assim por diante. Fica claro o padrão, a cada dígito a frequência de transição cai pela metade, esse efeito é idêntico ao gerado pelo *Flip-flop* T, logo conclui-se que um divisor de frequência é simultaneamente um contador.

3.2 Display de 7 Segmentos

Esse componente é composto por 7 segmentos de *LED* que, com a sequência de ativação correta, podem exibir números. Os segmentos são rotulados de 'a' até 'g', para cada um desses é disponibilizado um terminal de ativação que liga/desliga o *LED* associado conforme a tensão fornecida. Existem dois tipos de *display*:

1. **Cátodo comum:** os *LEDS* compartilham o mesmo nó em seus cátodos, implicando a ativação dos segmentos em nível lógico alto;
2. **Ânodo comum:** os *LEDS* compartilham o mesmo nó em seus ânodos, implicando a ativação dos segmentos em nível lógico baixo;

Figura 2 – Display de 7 segmentos ânodo/cátodo comum



Fonte: Material do Prof. Dr. Lauro Paulo da Silva Neto

3.3 Máquina de Moore

Uma máquina de Moore é um modelo computacional descrito por:

1. S um conjunto de estados finitos
2. $q_0|q_0 \in S$ um estado inicial
3. Σ um alfabeto de entrada finito
4. O um alfabeto de saída finito
5. δ uma função de transição de entrada
6. γ uma função de transição de saída

A computação ocorre da seguinte maneira, o símbolo $\sigma_a|\sigma_a \in \Sigma$ é lido, em seguida é invocada a função $\delta(\sigma_a, q_a)$ onde q_a é o estado atual e $q_a \in S$, essa chamada retorna um novo estado $q_b|q_b \in S$ para qual a máquina transita, sobre esse novo estado q_b um novo símbolo pode ser lido, e uma nova chamada á δ é realizada, reiniciando o processo. Simultaneamente, a cada troca de estados a função $\gamma(q)$ é chamada, determinado $o \in O$, o símbolo saída da máquina.

For a Moore machine, at each reaction, the output produced is defined by the current state (at the start of the reaction, not at the end). Thus, the output at the time of a reaction does not depend on the input at that same time. The input determines which transition is taken, but not what output is produced by the reaction. (LEE; SESHIA, 2017)

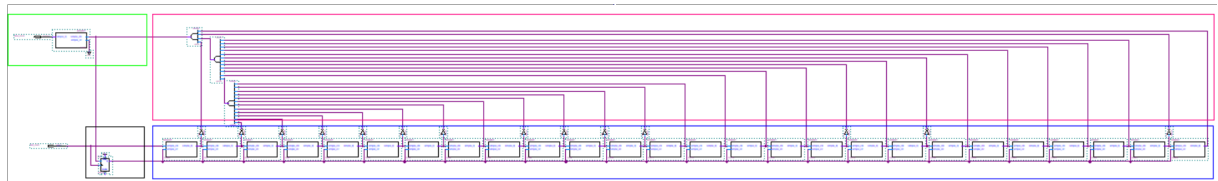
4 Desenvolvimento

4.1 Divisor de Frequência

Assumindo o *clock* nativo do FPGA com frequência de 50Mhz, foi projetado um divisor de frequência que conta o número de oscilações contidas em 1 segundo (50.000.000), e em seguida, inverte o estado de um *flip-flop* t , gerando um sinal de 1hz.

Para contar até 50.000.000 são necessários 26 *flip-flops*, quando esses configurarem a sequência 10111110101111000010000000 passou-se 1 segundo, nesse momento o estado da saída é alternado e a memória é limpa reiniciando a contagem.

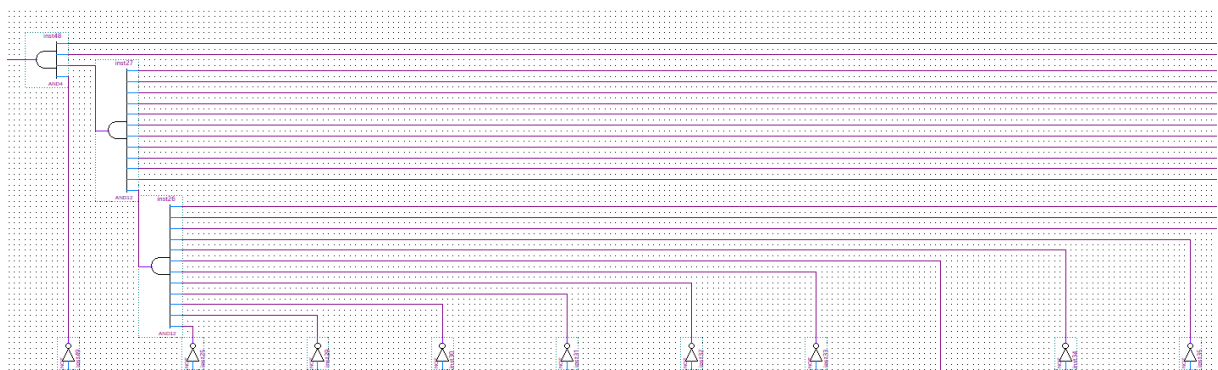
Figura 3 – Divisor de Frequência



Fonte: O autor.

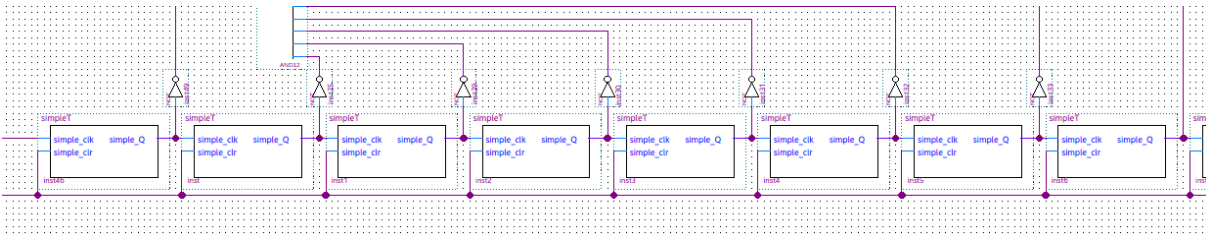
Para a conveniência do leitor, seguem imagens dedicadas para as partes relevantes do divisor:

Figura 4 – **Detector de sequencia**

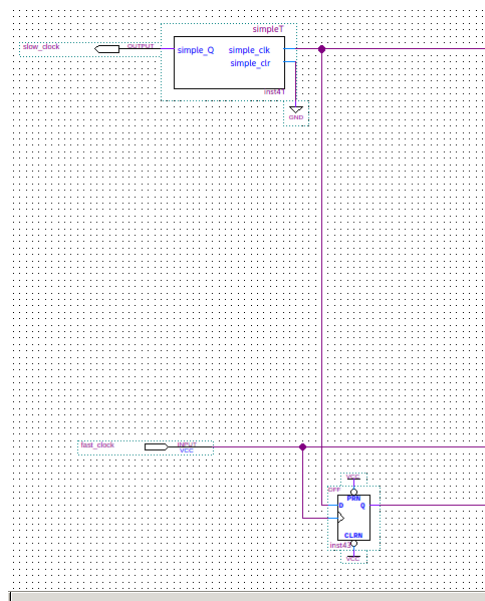


Fonte: O autor.

Figura 5 – Cascata de tipos T



Fonte: O autor.

Figura 6 – *Flip-flop T de saída* e circuito de *reset*

Fonte: O autor.

4.2 Máquina de Moore

A máquina implementada itera sobre a sequência 2-8-3-4-5-4-1-0-7, de acordo com a combinação dos *switches up edown* como descrito na Tabela 3.

Tabela 3 – Comandos
da
má-
quina

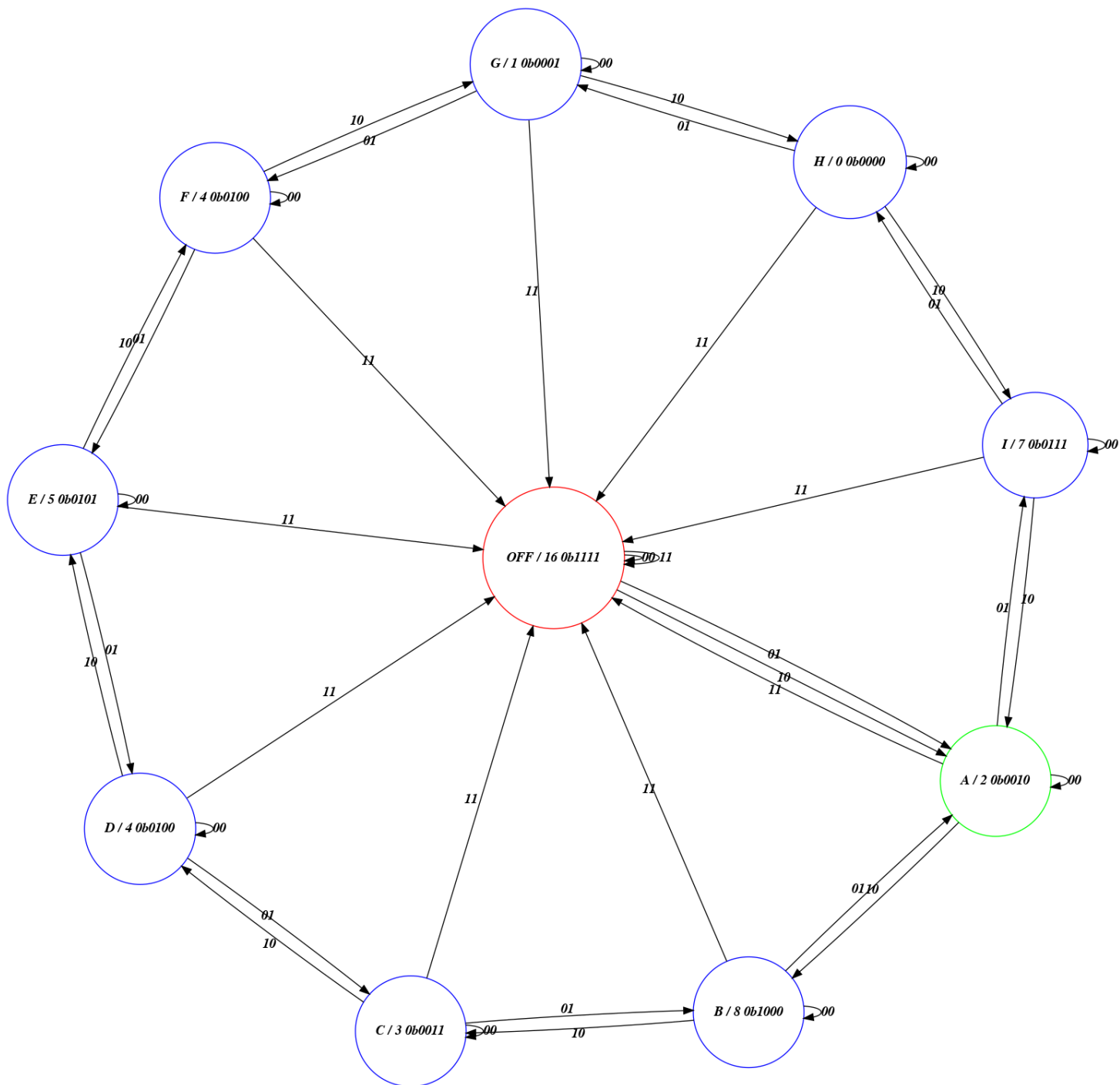
Up	$Down$	Comando
0	0	manter estado atual
0	1	iterar sobre os estados de forma decrescente
1	0	iterar sobre os estados de forma crescente
1	1	desligar o <i>display</i>

Fonte: O autor.

Com os comandos bem definidos, junto às restrições de circularidade, bidirecionalidade e desligamento do *display* foi definido o grafo da figura [Figura 7](#), sobre esse note que:

1. O **estado inicial** foi destacado em verde e o de *display desligado* em vermelho;
2. Todas as transições são acompanhadas de um número de dois dígitos, onde o primeiro representa o valor do *switch up* e o segundo *down*;
3. Todo estado é composto por um rótulo alfabético seguido por, uma barra(/), o número que deve ser imprimido na saída da máquina e a representação binária desse número.

Figura 7 – Grafo da Máquina de Moore



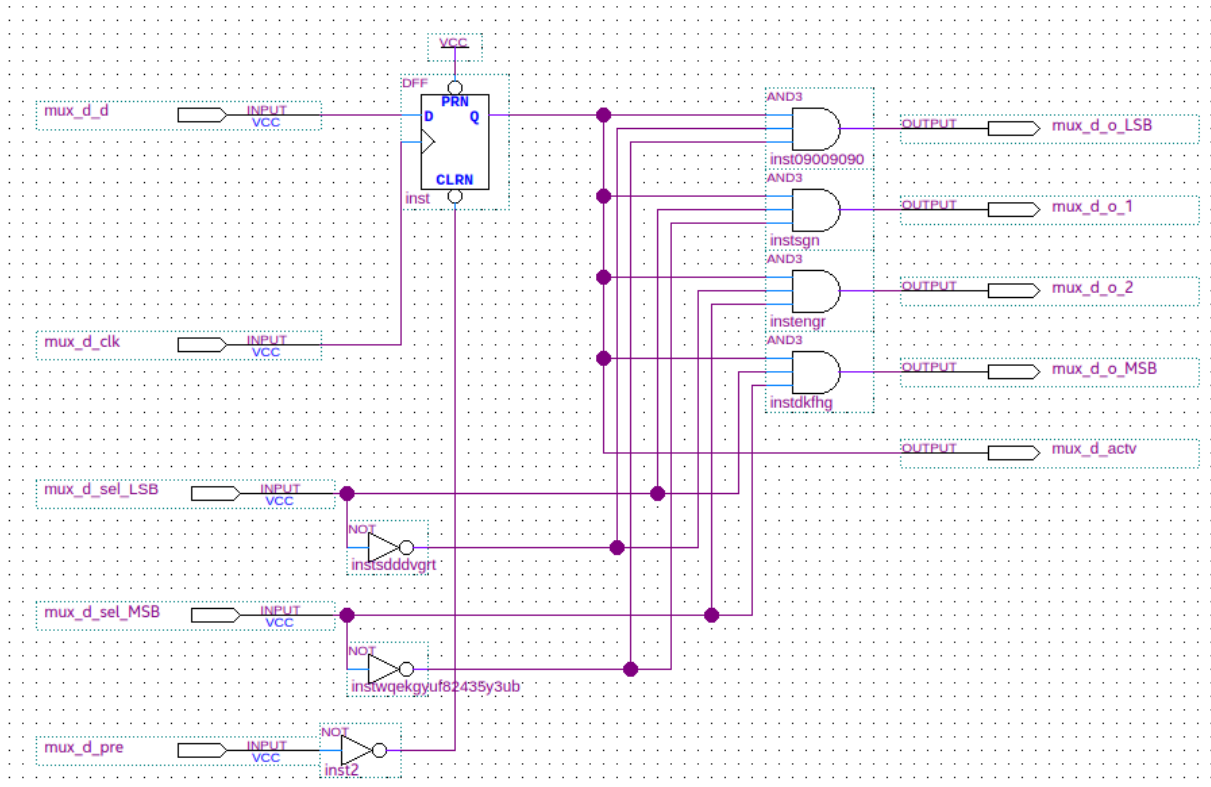
Fonte: O autor.

Conforme a representação, foram extraídos os atributos da Máquina Moore:

1. $S = \{A, B, C, D, E, F, G, H, I, OFF\}$ (um conjunto de estados finitos)
2. $q_0 = A$ (um estado inicial)
3. $\Sigma = \{00, 01, 10, 11\}$ (um alfabeto de entrada finito)
4. $O = \{0010, 1000, 0011, 0100, 0101, 0001, 0000, 0111, 1111\}$ (um alfabeto de saída finito)
5. δ é descrita pelas transições do grafo (uma função de transição de entrada)
6. γ é descrita pelo número associado ao estado (uma função de transição de saída)

Feita a análise inicial, fez-se necessário o projeto de um circuito que implemente um estado qualquer descrito na Figura 7. Para isso foi utilizado o seguinte arranjo:

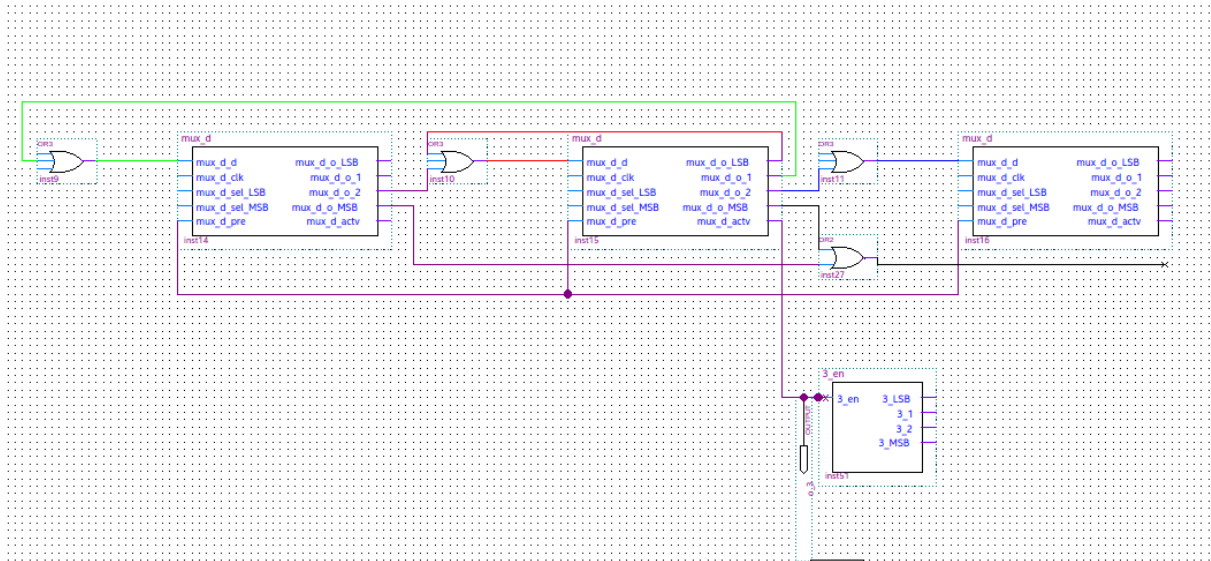
Figura 8 – Circuito de estado



Fonte: O autor.

A cada borda de subida em `mux_d_clk` o valor de `mux_d_d` é amostrado pelo *flip-flop* D e disponibilizado em sua saída Q. Como Máquinas de Moore são estritamente causais cada estado deve possuir 4 transições definidas, uma para cada simbolo de Σ ,

Figura 10 – Padrão de conexões



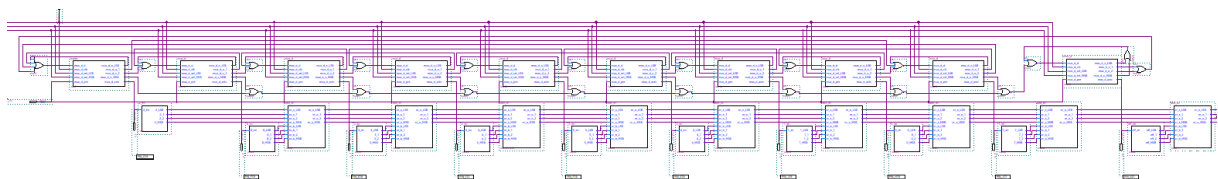
Fonte: O autor.

Esse diagrama foi montado em relação a perspectiva de `mux_d inst15`, para esse a saída:

1. Em vermelho (`mux_d_o_LSB`) volta para a própria `inst 15` (mantém);
2. Em verde (`mux_d_o_1`) segue para a instância anterior (conta decrescente);
3. Em azul (`mux_d_o_2`) segue pra a próxima instancia (conta crescente);
4. Em preto (`mux_d_o_MSB`) é combinada com `mux_o_MSB` dos outros estados e leva à instancia de `display` apagado;
5. Em roxo (`mux_d_o_act`) conecta-se à caixa `3_en` que produz o valor constante `0b0011` (3) caso o estado esteja ativo, emulando a função de saída γ (caixas com outras constantes foram implementadas a depender do estado);

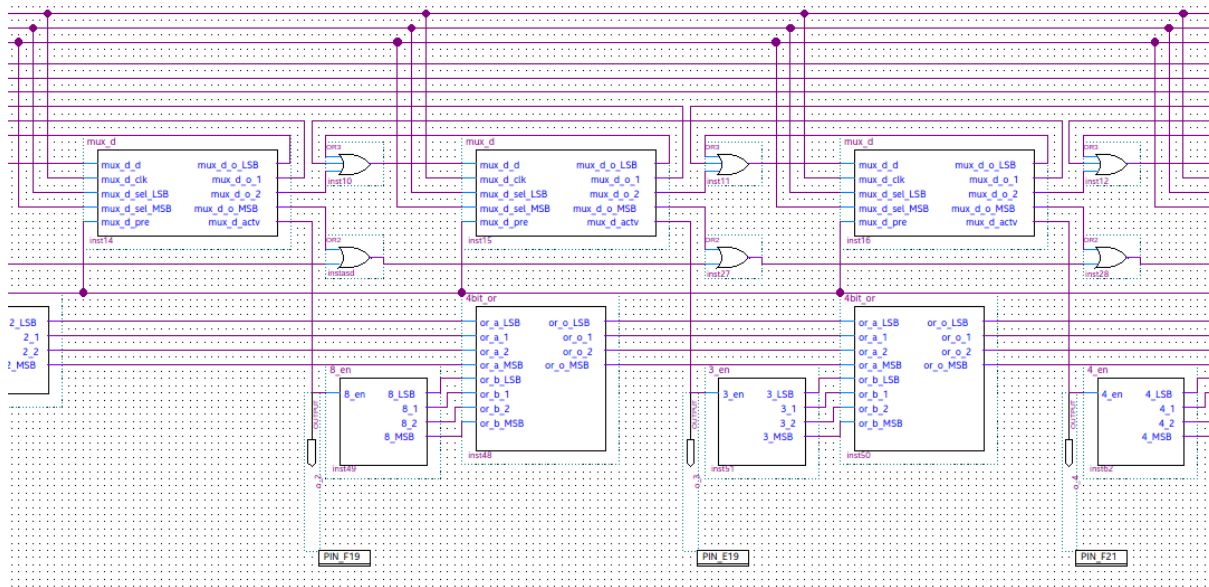
Generalizando esse padrão de conexões, foi implementada a máquina final como na [Figura 11](#) e [Figura 12](#)

Figura 11 – Máquina de Moore



Fonte: O autor.

Figura 12 – Máquina de Moore com zoom

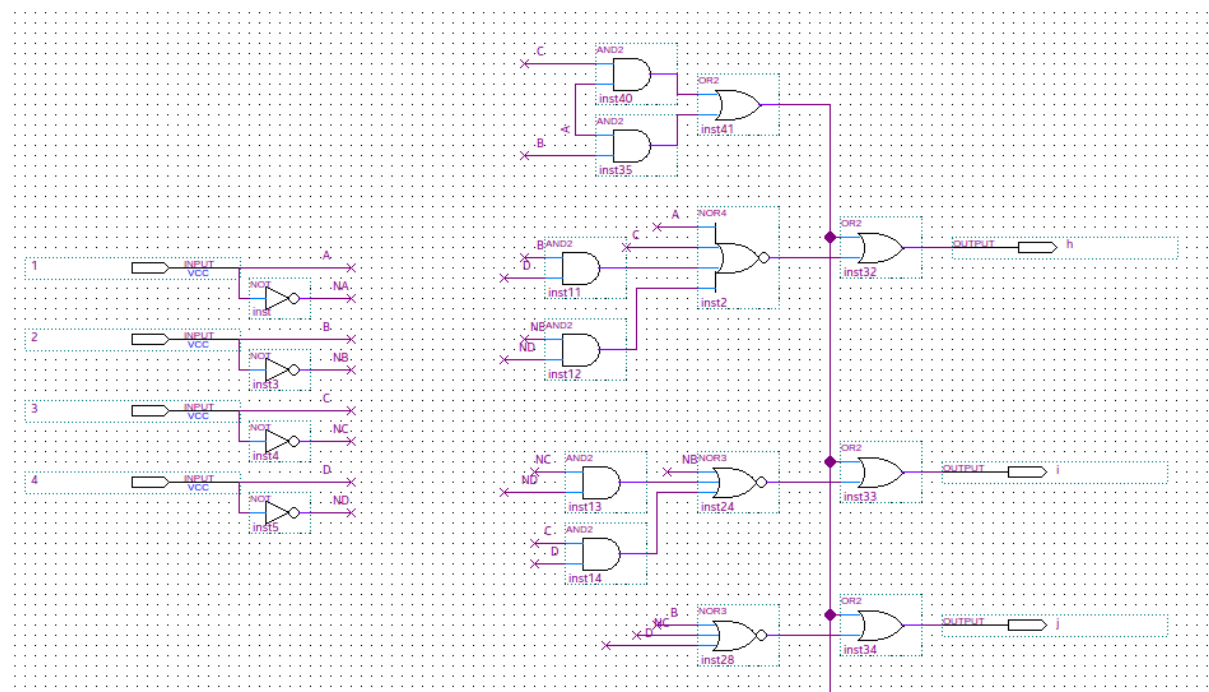


Fonte: O autor.

4.3 Conversor Binário Decimal

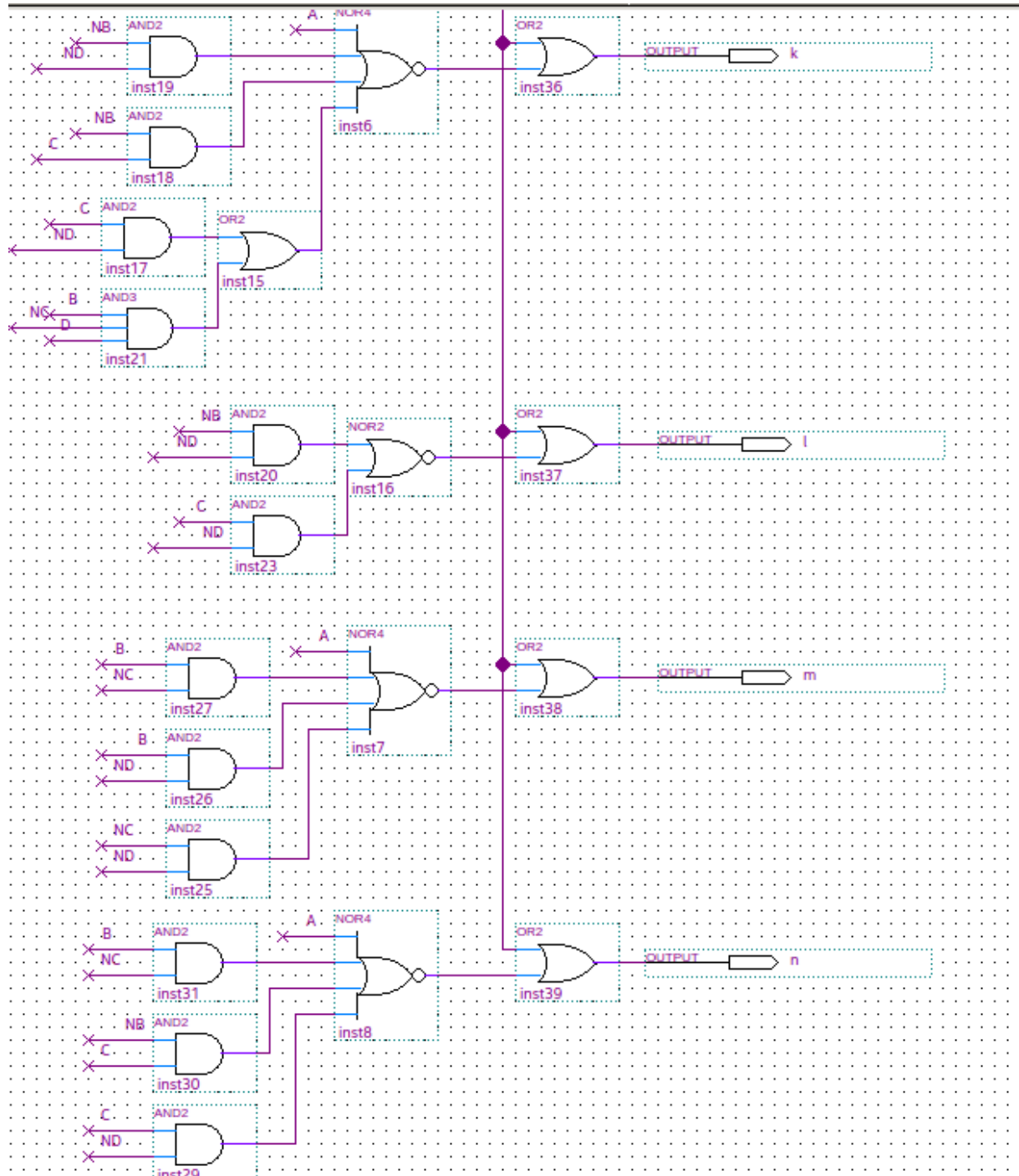
Também foi implementado o conversor binário/decimal, esse circuito toma um número de 4 bits e produz a saída correta para representa-lo em um *display* de 7 segmentos de anodo comum. Caso o número na entrada seja maior do que 9 (0b1001) o *display* apaga.

Figura 13 – Conversor BCD com zoom



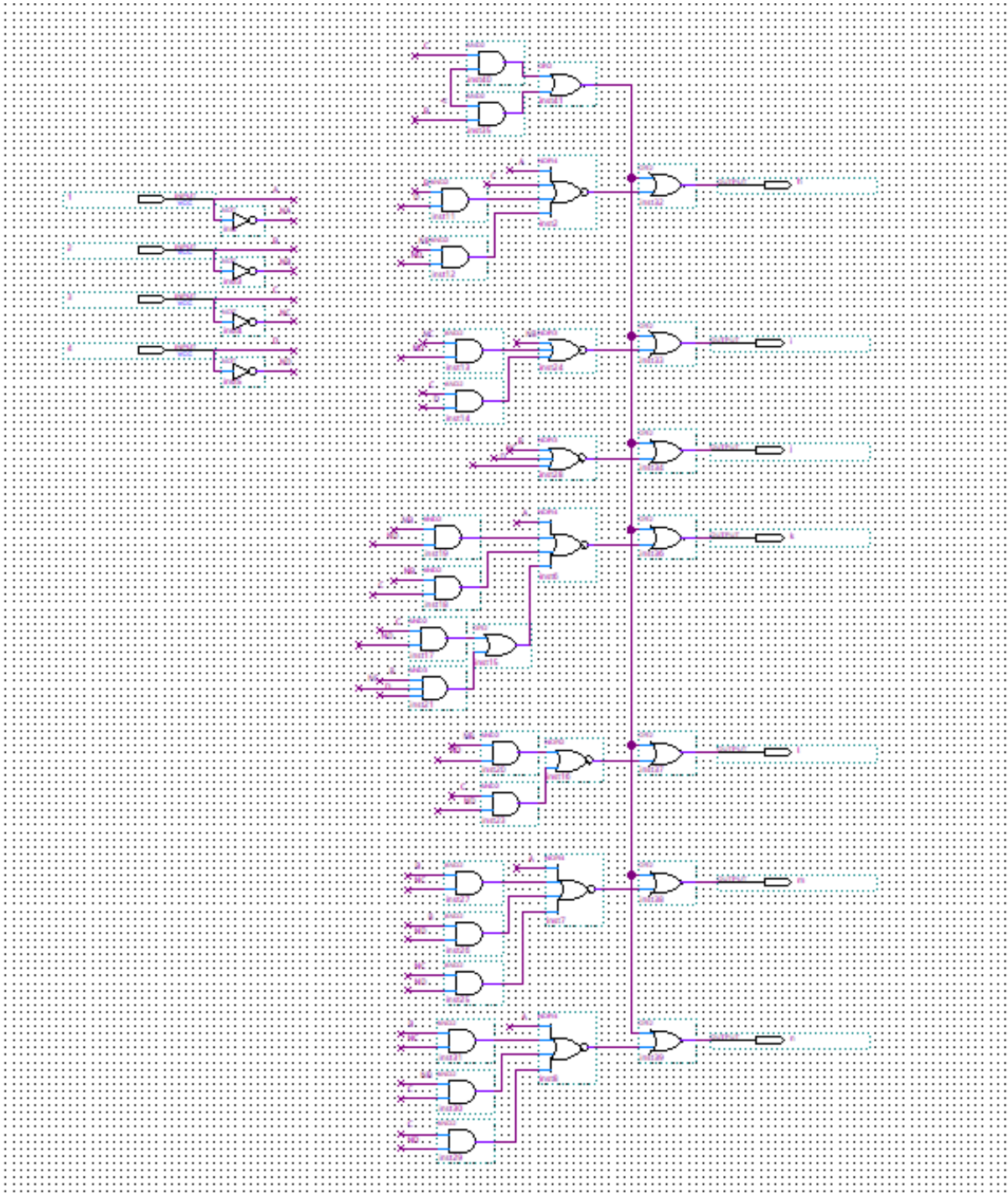
Fonte: O autor.

Figura 14 – Conversor BCD com zoom parte 2



Fonte: O autor.

Figura 15 – Conversor BCD

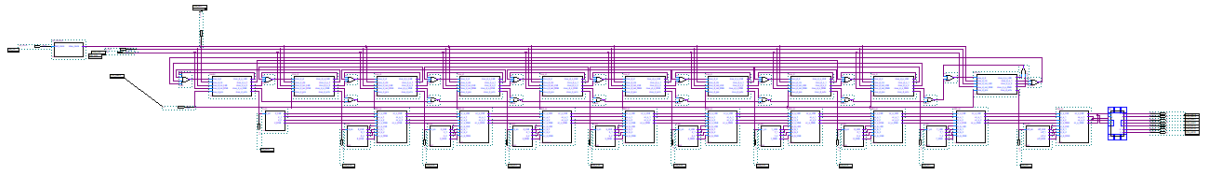


Fonte: O autor.

4.4 Circuito Final

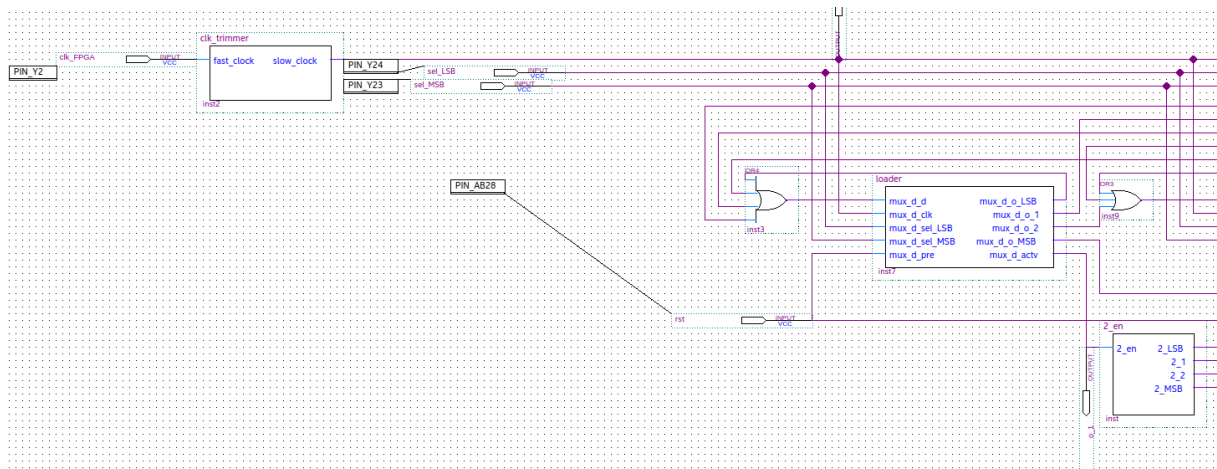
Finalmente, foram conectados os circuitos de divisão de frequência, conversão binária para decimal e Máquina de Moore.

Figura 16 – Circuito final



Fonte: O autor.

Figura 17 – Circuito final, entradas e divisor de frequência



5 Resultados e Discussões

O circuito final atingiu todos os objetivos propostos tanto no laboratório remoto quanto no presencial. O único elemento que gera incerteza é o divisor de frequência, o autor não conseguiu confirmar sem sombra de dúvidas se a frequência de saída realmente é de 1hz, testes na bancada com um cronômetro indicam fortemente que sim, mas devido a dificuldades com o *software Quartus* o recurso de forma de onda não pôde ser utilizado para sanar a dúvida.

6 Considerações Finais

A Máquina de Moore performou corretamente apesar da implementação não usual. A modelagem da Máquina utilizando grafos trouxe contigo vantagens e desafios particulares, por exemplo, utilizando os circuitos de estado não foi necessário o uso de álgebra Booleana para definir o circuito combinacional de entrada e saída como é feito tradicionalmente para Máquinas de Moore, em contraponto, encontrou-se dificuldade em ativar o estado inicial assim que a máquina fosse energizada.

O uso do *Quartus Prime 20.1 Lite Edition* também trouxe dificuldades, mesmo depois de múltiplas reinstalações e trocas de sistema operacional o autor não conseguiu realizar simulações de forma de onda, portanto, a validação do projeto foi feita exclusivamente no laboratório remoto, sem esse o projeto não seria entregue.

Referências

LEE, E. A.; SESHIA, S. A. *Introduction to Embedded Systems - A Cyber-Physical Systems Approach*. [S.l.]: MIT Press, 2017. Citado na página [15](#).