

Letreiro em matriz de LEDs

Luis Felipe Pereira de Carvalho
Naylane do Nascimento Ribeiro

Departamento de Tecnologia - Engenharia de Computação

Universidade Estadual de Feira de Santana (UEFS)
Feira de Santana – BA – Brazil

luisoftbr@outlook.com.br

naylaneribeiro4@gmail.com

Tutor: João Bosco Gertrudes

Abstract. *This project details the construction of a circuit, which aims to be a LED sign that displays the acronym "UEFS". Its development focused on the use of combinational code and shifting systems through the descriptive hardware language verilog. All the requirements of the request were fully met using the resources of the development kit.*

Resumo. *Este projeto detalha a construção de um circuito, que tem como objetivo ser um letreiro de LEDs que exibe a sigla "UEFS". Seu desenvolvimento teve foco no uso de código combinacional e sistemas de deslocamento através da linguagem descritiva de hardware verilog. Todos os requisitos da solicitação foram totalmente atendidos utilizando os recursos do kit de desenvolvimento.*

1. Introdução

O seguinte relatório técnico, no âmbito da disciplina de MI Projeto de Circuitos Digitais da Universidade Estadual de Feira de Santana (UEFS), tem como objetivo relatar e esclarecer as informações acerca da resolução do segundo problema: Letreiro em matriz de LEDs.

Segundo Mundodeled, Os letreiros de LEDs (Painéis Eletrônicos Digitais) tem se tornado uma ferramenta fundamental na sociedade atual, sendo utilizados em diversas áreas e contextos. Na comunicação visual, eles são amplamente utilizados em publicidade, propaganda e sinalização de empresas, lojas, aeroportos, estações de metrô, ônibus e outras instalações públicas, transmitindo informações importantes e mensagens de forma clara e visível para o público.

Na indústria de entretenimento, eles são usados em shows e eventos esportivos para exibir imagens, vídeos e animações em alta definição, proporcionando experiências imersivas para o público presente. Na área da educação, painéis digitais são utilizados em salas de aula para apresentações e exposições, tornando o processo de aprendizagem mais dinâmico e interativo.

Além disso, os painéis eletrônicos digitais têm sido utilizados na gestão do trânsito, em aeroportos e em terminais de ônibus, auxiliando os usuários na localização de rotas e horários de transporte público. Eles também têm sido utilizados em hospitais para exibição de informações de saúde, como avisos e orientações para pacientes e visitantes.

Para atender a construção do segundo problema, os alunos da turma de MI Projeto de Circuitos Digitais, tutorados pelo professor João Bosco Gertrudes, ficaram responsáveis por criar um protótipo de sistema de exibição da sigla “UEFS” em uma matriz de LEDs.

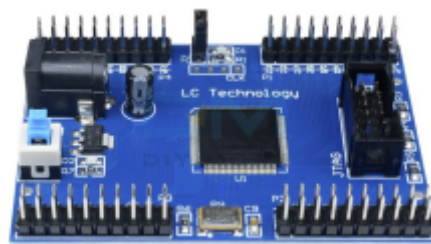
2. Materiais

Abaixo são apresentados o Kit de Desenvolvimento LEDS-CPLD e o Software Quartus que são os materiais necessários para a construção do protótipo.

2.1 Kit de Desenvolvimento LEDS-CPLD

Segundo o Manual do Kit de Desenvolvimento LEDS-CPLD, ele é composto por duas placas. A primeira é a CPLD que possui o circuito integrado CPLD (Complex Programmable Logic Device em inglês ou Dispositivo Lógico Complexo Programável no português) pertencente à família MAX II, com o modelo especificado EPM240T100C5N de dispositivos Intel Altera.

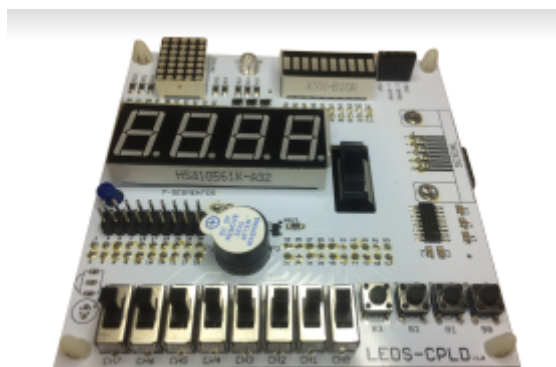
Figura 1. Placa de desenvolvimento CPLD MAX II EPM240



Fonte: Manual LEDS-CPLD

A outra placa é a LEDS-CPLD, que foi desenvolvida para a realização das atividades referentes à área de Projeto de Circuitos Digitais do Curso de Engenharia de Computação da UEFS.

Figura 2. Placa LEDS-CPLD



Fonte: Manual LEDS-CPLD

Este kit possui recursos que permitem ao usuário desenvolver diversas aplicações. Desde projetos de circuitos simples, até circuitos complexos, como ULAs (Unidades Lógica e Aritmética). As interfaces e conectores da plataforma são ligadas diretamente aos pinos de E/S da placa de desenvolvimento CPLD MAX II EPM240.

2.2 Quartus

Para o desenvolvimento, análise e testes do protótipo, foi utilizado o software Quartus. Ele é um software de design de dispositivo lógico programável (FPGA) produzido pela Intel Altera e através deste software é desenvolvida a linguagem de descrição de hardware Verilog (VHDL).

Essa linguagem proporciona a atribuição de funções ao hardware de uma placa CPLD através de códigos em blocos de texto dispostos em forma de módulos. Além disso, o programa apresenta soluções como o University Program VWF, o Chip Planner e o Pin Planner. O VWF é um tipo de arquivo de simulação utilizado, ele contém uma descrição dos sinais elétricos em um circuito digital e é usado para simular o comportamento deste circuito em um ambiente de teste virtual. O Chip Planner permite visualizar e gerenciar a alocação física de componentes de um projeto em um FPGA e o Pin Planner permite designar os pinos de entrada e saída de um dispositivo lógico programável (FPGA ou CPLD).

A linguagem Verilog é utilizada neste projeto como meio controlador das funções do kit de desenvolvimento LEDS-CPLD, de modo a atender os requisitos para a solução do problema proposto.

Na seguinte seção é apresentada a fundamentação teórica necessária para a utilização do kit de desenvolvimento LEDS-CPLD e para a construção do código a ser implementado no kit através do software Quartus.

3. Fundamentação Teórica

De modo a desenvolver o protótipo, foram realizados estudos e resolução de atividades laborais, além da realização das sessões tutoriais de PBL para orientação e discussão acerca da consolidação do projeto.

A seguir são apresentados e detalhados uma breve definição sobre níveis lógicos, circuitos combinacionais e condicionais, clock, flip-flops, multiplexadores e demultiplexadores e condicional if/else (se/senão).

3.1 Níveis Lógicos

“As tensões usadas para representar os binários 1 (alto) e 0 (baixo) são denominados níveis lógicos. Teoricamente, um nível de tensão representa um nível alto e o outro representa um nível baixo. Entretanto, em um circuito digital prático, um nível alto pode ser qualquer tensão entre um valor mínimo e um valor máximo especificados. Da mesma forma, um nível baixo pode ser qualquer valor de tensão entre um valor mínimo e máximo especificados. Não existe sobreposição entre as faixas aceitáveis para os níveis alto e baixo.” [Floyd, 2007]

Desse modo, os níveis lógicos binários são fundamentais para o funcionamento de sistemas digitais e no projeto proposto, os níveis lógicos são a base para todo o funcionamento, desde a entrada, até a saída dos dados.

3.2 Circuitos combinacionais e condicionais

Segundo Capuano (2000), Os circuitos combinacionais são compostos por portas lógicas e outros componentes que geram saídas com base apenas nas entradas

fornecidas, sem armazenar qualquer informação sobre o estado anterior do sistema, de modo que sua abordagem se concentra em como os componentes do sistema se relacionam e interagem entre si.

Por outro lado, os circuitos condicionais envolvem a utilização de elementos de memória, como flip-flops e registradores, para armazenar informações sobre o estado anterior do sistema, ou seja, se concentra em como o sistema se comporta em resposta a diferentes estímulos e eventos. Esses circuitos podem tomar decisões com base nas entradas e no estado atual do sistema para gerar as saídas.

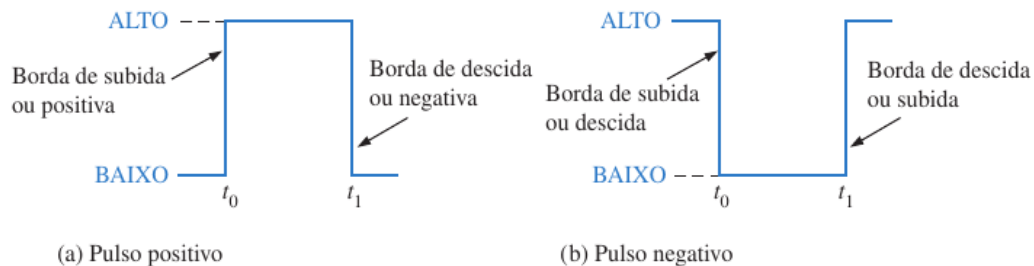
3.3 Clock

Segundo Tocci (2011), os sistemas digitais podem ser operados no modo assíncrono ou síncrono. No modo assíncrono, as saídas de circuitos lógicos podem mudar de estado em qualquer momento em que uma ou mais entradas também mudarem.

No entanto, a concepção e a análise de defeitos são mais complexas em sistemas assíncronos. Já em sistemas síncronos, as transições de saída são determinadas por um sinal chamado clock, que geralmente é um trem de pulsos retangulares ou uma onda quadrada.

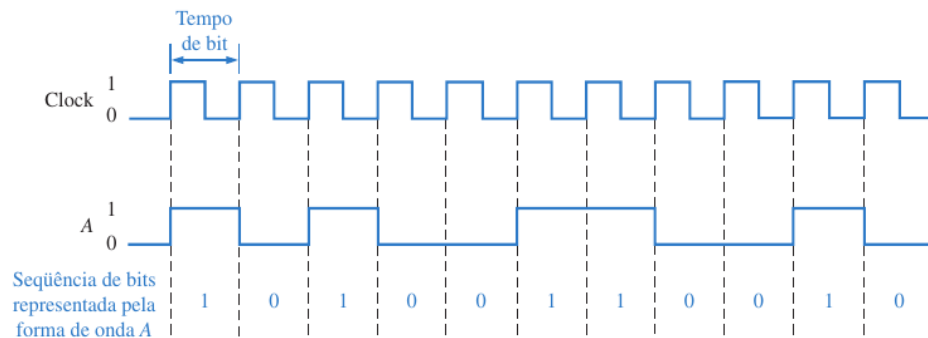
Esse sinal é distribuído para todas as partes do sistema e a maioria das saídas muda de estado somente quando há uma transição no sinal de clock. Quando o clock muda de 0 para 1, é chamado de transição positiva (borda de subida), e quando muda de 1 para 0, é chamado de transição negativa (borda de descida). A figura 3 apresenta um sinal de clock com suas bordas de subida e de descida.

Figura 3. Sinal de clock



(a) Pulso positivo

(b) Pulso negativo



Fonte: Floyd (2007)

3.4 Flip Flops

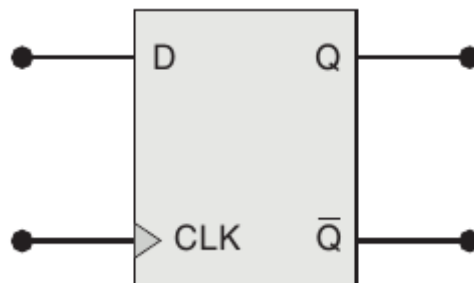
Segundo Floyd (2007), os flip-flops são dispositivos biestáveis síncronos, também conhecidos como multivibradores biestáveis. O termo "síncrono" nesse contexto indica que a mudança de estado da saída só ocorre no momento determinado pela entrada de disparo "clock" (CLK), que é tratada como uma entrada de controle (C). Em outras palavras, as alterações na saída são sincronizadas com o clock.

Existem vários tipos de flip-flops, como o D, JK, T e SR. Cada tipo tem sua própria função e características específicas. A principal diferença entre os tipos é como eles são ativados e atualizados.

- D flip-flops são ativados pelo sinal de clock e armazenam o valor de sua entrada D na borda de subida ou descida do clock.
- JK flip-flops são ativados pelo sinal de clock e armazenam o valor de sua entrada J quando $J=1$ e $K=0$, ou armazenam o inverso do valor atual quando $J=K=1$, ou mantêm o valor atual quando $J=K=0$.
- T flip-flops são ativados pelo sinal de clock e alternam o valor de sua saída Q a cada borda de subida ou descida do clock.
- SR flip-flops são ativados pelo sinal de clock e armazenam o valor de sua entrada S quando $S=1$ e $R=0$, ou armazenam o valor de sua entrada R quando $R=1$ e $S=0$, ou mantêm o valor atual quando $S=R=0$.

Para a solução do problema, foram utilizados flip-flops do tipo T. A figura 4 apresenta um flip-flop do tipo T.

Figura 4. Flip-flop do tipo T

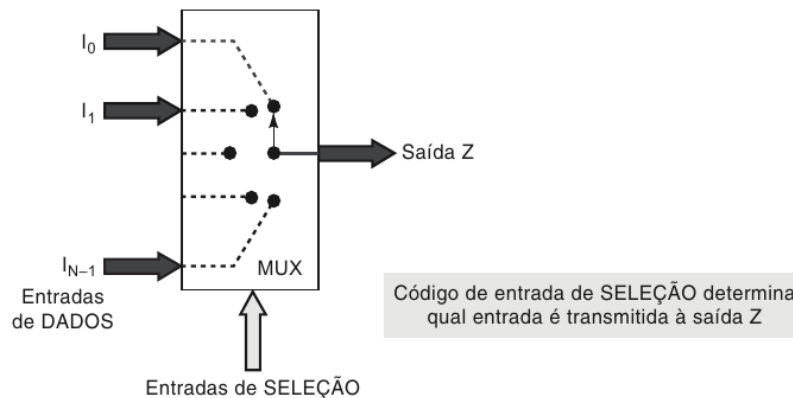


Fonte: Tocci (2011)

3.5 Multiplexador e Demultiplexador

Segundo Tocci (2011), Um multiplexador, ou MUX, é um circuito combinacional que seleciona um sinal de entrada entre vários possíveis para enviar para a saída. Ele é composto por um conjunto de entradas, uma ou mais entradas de seleção e uma única saída. A seleção das entradas é feita com base no valor das entradas de seleção. Na figura 6, é apresentado o diagrama de um multiplexador.

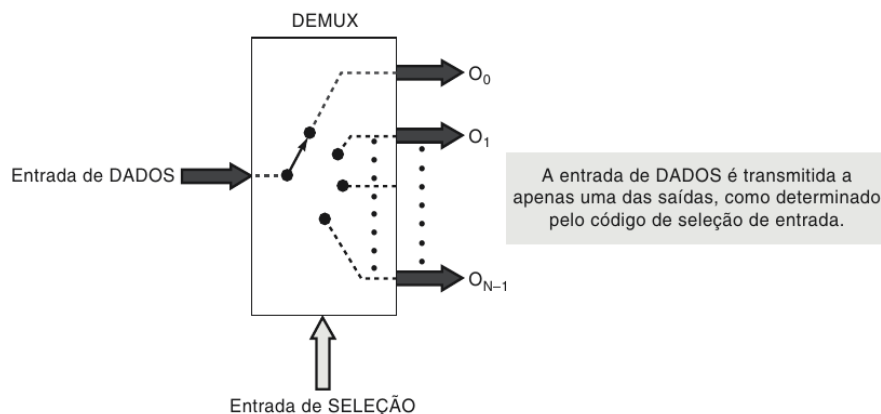
Figura 6. Diagrama de um multiplexador



Fonte: Tocci (2011)

Um demultiplexador, ou DEMUX, basicamente inverte a função da multiplexação. Ele recebe informações digitais a partir de uma linha e as distribui para um determinado número de linhas de saída. Por essa razão, o demultiplexador também é conhecido como distribuidor de dados. Na figura 7 é apresentado o diagrama de um demultiplexador.

Figura 7. Diagrama de um demultiplexador



Fonte: Tocci (2011)

3.6 If e Else

Os condicionais “If/Else” são uma estrutura de controle de fluxo em programação que permite executar diferentes blocos de código dependendo de uma condição booleana. Em circuitos digitais, a estrutura "If/Else" (Se/Senão) é implementada por meio de portas lógicas, através da construção de um sistema que opera de forma comportamental (condicional). No protótipo, essa estrutura foi utilizada no processamento dos elementos biestáveis

De modo geral, a estrutura, em código, de um “If/Else” ocorre da seguinte forma:

```
if condição:
    bloco de código a ser executado se a condição for verdadeira
else:
    bloco de código a ser executado se a condição for falsa
```

É importante ressaltar, que podem haver mais condições para que um fluxo seja executado, de modo que essas demais condições são determinadas por um “Else If” (Senão se) com condições que são distintas das demais.

Na seção seguinte é apresentada a metodologia utilizada para o desenvolvimento do protótipo.

4. Metodologia

Após a apresentação dos materiais e da base didática necessária para a resolução do problema, partiremos para a apresentação dos requisitos solicitados e a construção do protótipo. Nas seguintes subseções são apresentados os requisitos solicitados e todo o desenvolvimento do protótipo desde a escolha das interfaces de entrada e saída até a construção e funcionamento dos módulos desenvolvidos em linguagem de descrição de Hardware Verilog (VHDL).

4.1 Definição de requisitos

Como citado anteriormente, para a concretização do problema, são apresentados os requisitos que estão dispostos na tabela 1:

Tabela 1. Requisitos para a construção execução do protótipo

Requisito	Detalhamento
Interface de entrada (IE01)	O protótipo deve conter uma interface de entrada (IE01) composta por duas chaves, CH0 e CH1.
Interface de saída (IS01)	O protótipo deve exibir, bit a bit, em uma interface de saída (IS02) representada por uma matriz de LEDs, o código 2 de 5 inserido.
Comutação das chaves	<ul style="list-style-type: none">• Quando somente a chave CH0 está em nível lógico alto, a mensagem deve ser exibida com deslocamento pela coluna da direita para a esquerda.• Quando somente a chave CH1 está em nível lógico alto, a mensagem deve ser exibida com deslocamento pela coluna da esquerda para a direita.• Quando ambas as chaves estão em nível lógico baixo, não há mensagem para ser exibida.• Quando ambas as chaves estão em

	nível lógico alto, o deslocamento é interrompido.
--	---

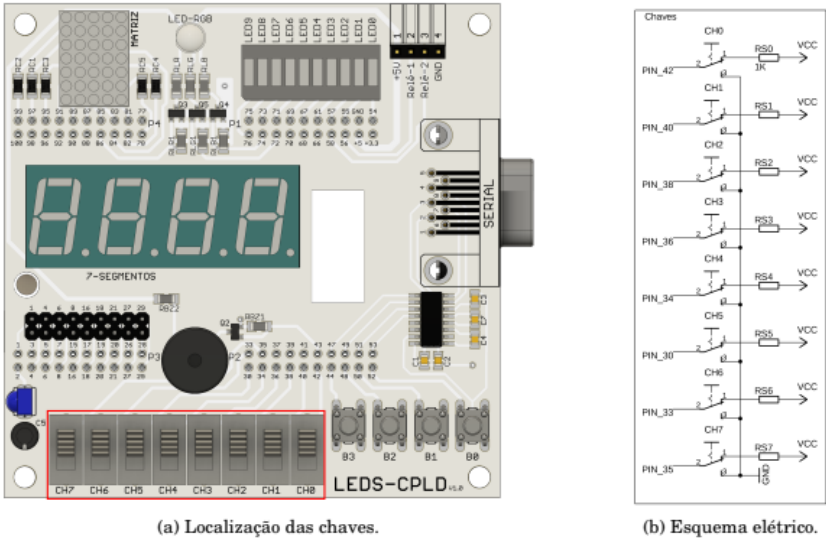
Fonte: Elaborada pelos autores

4.2 Apresentação das interfaces de entrada e saída

Para o desenvolvimento do protótipo é imprescindível a definição das interfaces de entrada (IE01) e saída (IS01). Foi realizada a escolha das chaves CH0 e CH1 como interface de entrada para IE01 e para a interface de saída IS01 foi escolhida a matriz de leds.

4.2.1 Chaves

Figura 8. Conjunto de Chaves utilizado como interface da entrada IE01



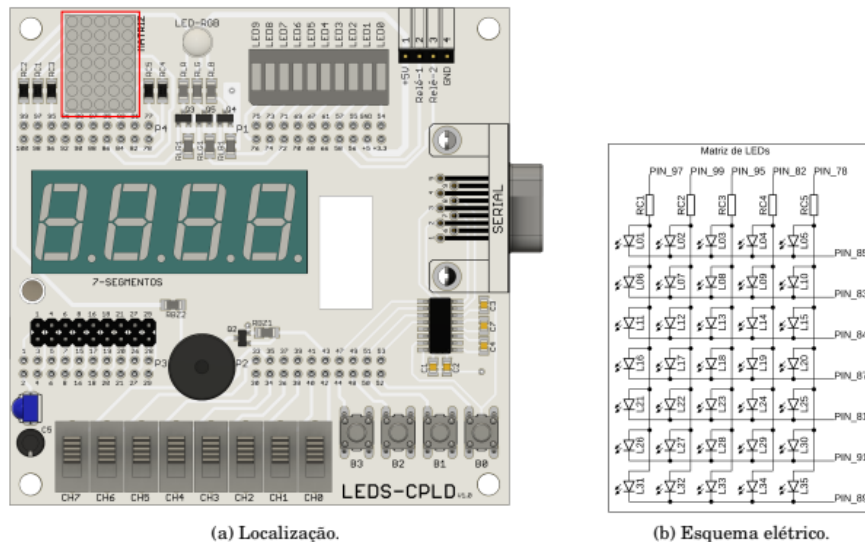
Fonte: Manual LEDS-CPLD

Segundo o manual do kit LEDS-CPLD, as chaves fornecem um sinal de nível lógico alto (1) (chave para cima) ou nível lógico baixo (0) (chave para baixo) para o CPLD.

Para o protótipo, as chaves CH0 e CH1 são utilizadas para que o usuário faça a escolha do modo de exibição da mensagem na matriz de LEDs, conforme descrito na tabela 1.

4.2.2 Matriz de LEDs

Figura 9. Matriz de leds utilizada como a interface de saída IS01



Fonte: Manual LEDS-CPLD

Segundo o manual do kit LEDS-CPLD, a matriz de LEDs é composta por 5 linhas e 7 colunas conforme mostrado na figura 9. Cada linha e cada coluna está conectada a um pino do CPLD. As colunas são ativadas com nível lógico alto (1) e as linhas com nível lógico baixo (0), de modo que cada um dos 35 LEDs podem ser controlados individualmente pela combinação de sinais lógicos na linha e na coluna.

4.3 Varredura de Matriz

Para a varredura da matriz de LEDs, um controlador gera um sinal de controle que indica qual linha de LEDs deve ser acesa em um determinado momento. O sinal de controle é enviado para um decodificador de linha, que ativa os circuitos integrados correspondentes a essa linha. Cada linha é ativada rapidamente, um após o outro, em uma sequência rápida para criar a ilusão de que todos os LEDs estão acesos simultaneamente.

Em seguida, o controlador fornece a informação de exibição para cada LED individual na linha acesa, para exibir o padrão ou imagem desejados. Isso geralmente é feito por meio de uma memória de exibição ou buffer de imagem, que armazena as informações necessárias para exibir a imagem na matriz de LEDs. O controlador atualiza rapidamente as informações da memória de exibição para cada linha, enquanto a matriz de LEDs é varrida. A figura 10 mostra a varredura da matriz sendo feita linha a linha.

Figura 10. Varredura da matriz

	CO	C1	C2	C3	C4				CO	C1	C2	C3	C4					CO	C1	C2	C3	C4					CO	C1	C2	C3	C4	
L0	○	○	○	○	○				L0	○	○	○	○	○				L0	○	○	○	○	○				L0	○	○	○	○	○
L1	○	○	○	○	○				L1	○	○	○	○	○				L1	○	○	○	○	○				L1	○	○	○	○	○
L2	○	○	○	○	○				L2	○	○	○	○	○				L2	○	○	○	○	○				L2	○	○	○	○	○
L3	○	○	○	○	○				L3	○	○	○	○	○				L3	○	○	○	○	○				L3	○	○	○	○	○
L4	○	○	○	○	○				L4	○	○	○	○	○				L4	○	○	○	○	○				L4	○	○	○	○	○
L5	○	○	○	○	○				L5	○	○	○	○	○				L5	○	○	○	○	○				L5	○	○	○	○	○
L6	○	○	○	○	○				L6	○	○	○	○	○				L6	○	○	○	○	○				L6	○	○	○	○	○
	CO	C1	C2	C3	C4				CO	C1	C2	C3	C4					CO	C1	C2	C3	C4										
L0	○	○	○	○	○				L0	○	○	○	○	○				L0	○	○	○	○	○									
L1	○	○	○	○	○				L1	○	○	○	○	○				L1	○	○	○	○	○									
L2	○	○	○	○	○				L2	○	○	○	○	○				L2	○	○	○	○	○									
L3	○	○	○	○	○				L3	○	○	○	○	○				L3	○	○	○	○	○									
L4	○	○	○	○	○				L4	○	○	○	○	○				L4	○	○	○	○	○									
L5	○	○	○	○	○				L5	○	○	○	○	○				L5	○	○	○	○	○									
L6	○	○	○	○	○				L6	○	○	○	○	○				L6	○	○	○	○	○									

Fonte: Elaborada pelos autores

4.4 Quadros

Para a construção do protótipo, como informação de exibição (Ver seção 4.3), foram criados quadros binários representativos para o movimentos das letras “UEFS” no display de modo a facilitar a construção do software do protótipo. Na figura 11, são apresentados os 8 quadros construídos.

Figura 11. Tabelas verdade dos quadros da sigla “UEFS”

QUADRO 0							
ENTRADAS			SAÍDAS				
A	B	C	C0	C1	C2	C3	C4
0	0	0	1	1	1	1	1
0	0	1	0	0	0	0	1
0	1	0	1	1	1	1	1
0	1	1	0	0	0	0	0
1	0	0	1	1	1	1	1
1	0	1	1	0	1	0	1
1	1	0	1	0	1	0	1
1	1	1	-	-	-	-	-

QUADRO 1							
ENTRADAS			SAÍDAS				
A	B	C	C0	C1	C2	C3	C4
0	0	0	1	1	1	1	1
0	0	1	0	0	0	0	0
0	1	0	1	1	1	1	1
0	1	1	1	0	1	0	1
1	0	0	1	0	1	0	1
1	0	1	0	0	0	0	0
1	1	0	1	1	1	1	1
1	1	1	-	-	-	-	-

QUADRO 2							
ENTRADAS			SAÍDAS				
A	B	C	C0	C1	C2	C3	C4
0	0	0	1	1	1	1	1
0	0	1	1	0	1	0	1
0	1	0	1	0	1	0	1
0	1	1	0	0	0	0	0
1	0	0	1	1	1	1	1
1	0	1	1	0	1	0	1
1	1	0	1	0	1	0	0
1	1	1	1	0	1	0	0
1	1	1	-	-	-	-	-

QUADRO 3							
ENTRADAS			SAÍDAS				
A	B	C	C0	C1	C2	C3	C4
0	0	0	1	0	1	0	1
0	0	1	0	0	0	0	0
0	1	0	1	1	1	1	1
0	1	1	1	0	1	0	0
1	0	0	1	0	1	0	0
1	0	1	0	0	0	0	0
1	1	0	1	1	1	0	1
1	1	1	-	-	-	-	-

QUADRO 4							
ENTRADAS			SAÍDAS				
A	B	C	C0	C1	C2	C3	C4
0	0	0	1	1	1	1	1
0	0	1	1	0	1	0	0
0	1	0	1	0	1	0	0
0	1	1	0	0	0	0	0
1	0	0	1	1	1	0	1
1	0	1	1	0	1	0	1
1	1	0	1	0	1	1	1
1	1	1	-	-	-	-	-

QUADRO 5							
ENTRADAS			SAÍDAS				
A	B	C	C0	C1	C2	C3	C4
0	0	0	1	0	1	0	0
0	0	1	0	0	0	0	0
0	1	0	1	1	1	0	1
0	1	1	1	0	1	0	1
1	0	0	1	0	1	1	1
1	0	1	0	0	0	0	0
1	1	0	1	1	1	1	1
1	1	1	-	-	-	-	-

QUADRO 6							
ENTRADAS			SAÍDAS				
A	B	C	C0	C1	C2	C3	C4
0	0	0	1	1	1	0	1
0	0	1	1	0	1	0	1
0	1	0	1	0	1	1	1
0	1	1	0	0	0	0	0
1	0	0	1	1	1	1	1
1	0	1	0	0	0	0	1
1	1	0	1	1	1	1	1
1	1	1	-	-	-	-	-

QUADRO 7							
ENTRADAS			SAÍDAS				
A	B	C	C0	C1	C2	C3	C4
0	0	0	1	0	1	1	1
0	0	1	0	0	0	0	0
0	1	0	1	1	1	1	1
0	1	1	0	0	0	0	1
1	0	0	1	1	1	1	1
1	0	1	0	0	0	0	0
1	1	0	1	1	1	1	1
1	1	1	-	-	-	-	-

Fonte: Elaborada pelos autores

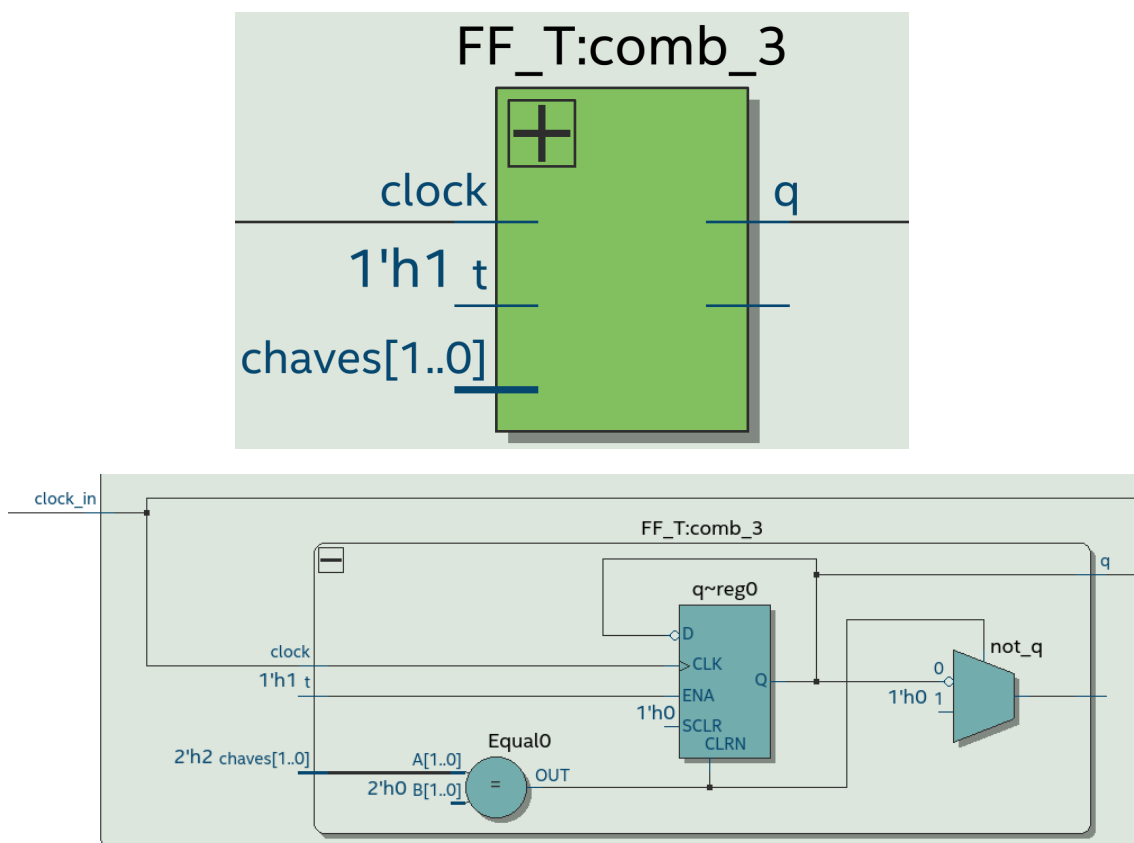
Note que para cada quadro é determinada uma entrada de 3 bits (A, B e C) onde as combinações, exceto (111), são relacionadas a cada expressão das 7 linhas do quadro de modo a determinar quais colunas recebem nível alto ou baixo para a formação das listras ou parte delas.

4.5 Apresentação da Linguagem Descritiva de Hardware Verilog (VHDL)

A Linguagem Descritiva de Hardware Verilog (VHDL) foi utilizada para o controle dos recursos utilizados no protótipo, seu código foi construído em forma subdividida em 13 arquivos, com um deles sendo o principal. Foram desenvolvidos módulos, sendo eles: Flip-Flop T, Contador, Divisor Clock, Quadro X (são oito módulos numerados de 0 a 7), Multiplexador e Varredura Matriz. Estes módulos são apresentados a seguir.

4.5.1 Módulo Flip-Flop T

Figura 12. Módulo FF_T



Fonte: Representação gerada com o RTL Viewer

O módulo FF_T é responsável por descrever um flip-flop T, que é um tipo de circuito lógico bistável. O flip-flop T tem duas entradas: uma entrada de clock e uma entrada T. Também tem duas saídas: uma saída Q e uma saída NOT-Q.

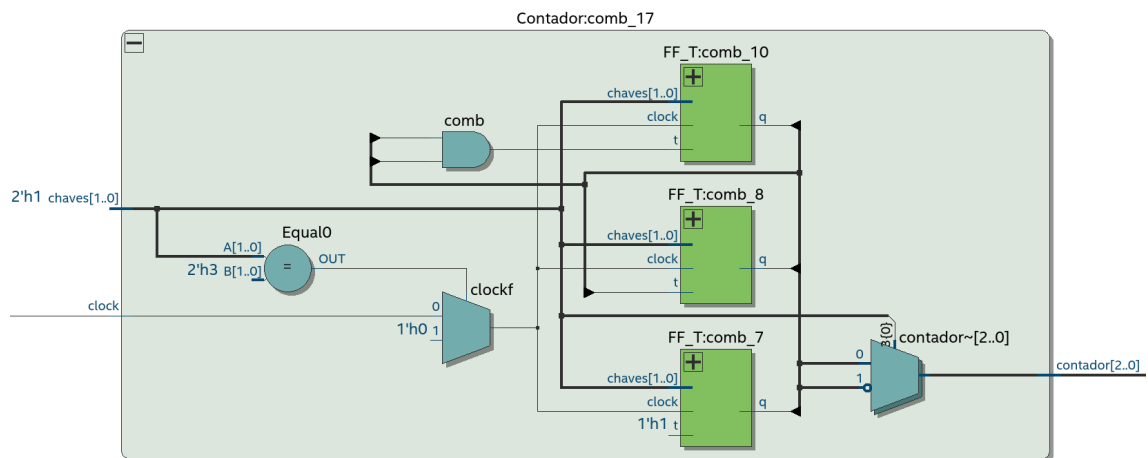
Quando as chaves CH0 e CH1 estiverem em 0, é feito o reset, e a saída Q recebe nível lógico baixo. Caso contrário, se a entrada T estiver em nível lógico alto, a saída Q

muda de estado, ou seja, se estiver em nível lógico baixo, passa para nível lógico alto, e vice-versa.

A saída NOT-Q é a saída complementar da saída Q. Se o reset for acionado, a saída NOT-Q será nível lógico baixo, caso contrário, será o complemento da saída Q. O circuito é implementado usando um registrador, que armazena o valor atual da saída Q, e a porta NOT é usada para gerar a saída NOT-Q.

4.5.2 Módulo Contador

Figura 13. Módulo Contador



Fonte: Representação gerada com o RTL Viewer

O módulo Contador é responsável por implementar um contador assíncrono de 3 bits que utiliza flip-flops T. O contador é controlado por duas chaves (CH0 e CH1) e um sinal de clock. Os flip-flops T são definidos em um módulo separado (FF_T), que é instanciado três vezes neste módulo.

O sinal de clock é passado por um filtro (wire clockf) que mantém o clock em nível lógico alto, exceto quando ambas as chaves estão em nível lógico alto. Nesse caso, o clock é forçado a nível lógico baixo (1'b0), o que impede a contagem.

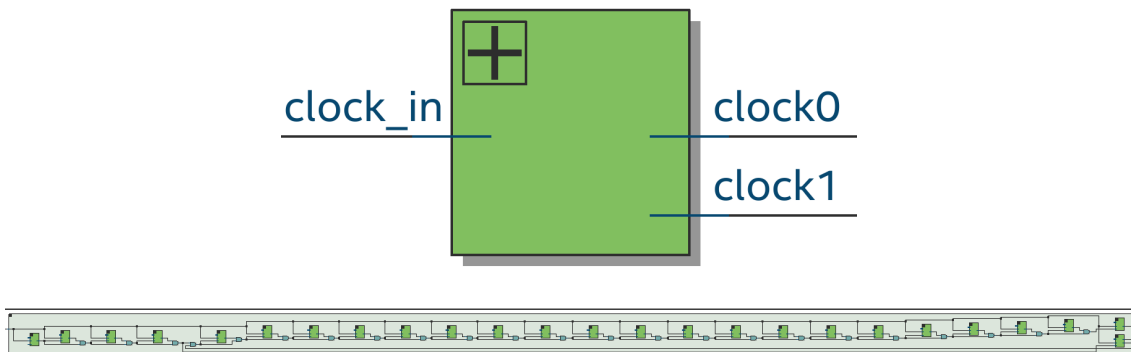
A saída do primeiro flip-flop T (a) é usada como entrada do segundo flip-flop T. A saída do segundo flip-flop T (b) é combinada com a saída do primeiro (a) em um AND, cuja saída é usada como entrada do terceiro flip-flop T (c).

As saídas dos flip-flops T (a, b e c) e suas negações (not_a, not_b e not_c) são usadas para gerar a saída do contador. O bit menos significativo (contador[0]) é a saída do primeiro flip-flop T (a), invertida ou não, dependendo do valor da chave CH0. O bit do meio (contador[1]) é a saída do segundo flip-flop T (b), invertida ou não, dependendo do valor da chave CH0. O bit mais significativo (contador[2]) é a saída do terceiro flip-flop T (c), invertida ou não, dependendo do valor da chave CH0.

4.5.3 Módulo Divisor_Clock

Figura 14. Módulo Divisor_Clock

Divisor_Clock:comb_3



Fonte: Representação gerada com o RTL Viewer

O módulo Divisor_Clock é responsável por ser um divisor de frequência que gera dois sinais de clock (Ver seção 3.3) com frequências diferentes a partir de um sinal de clock de entrada. O módulo possui como entrada, o clock_in de 50Mhz presente no CPLD e como saída:

- clock0: sinal de clock de saída com frequência de 6,25 MHz.
- clock1: sinal de clock de saída com frequência de 1,49 Hz.

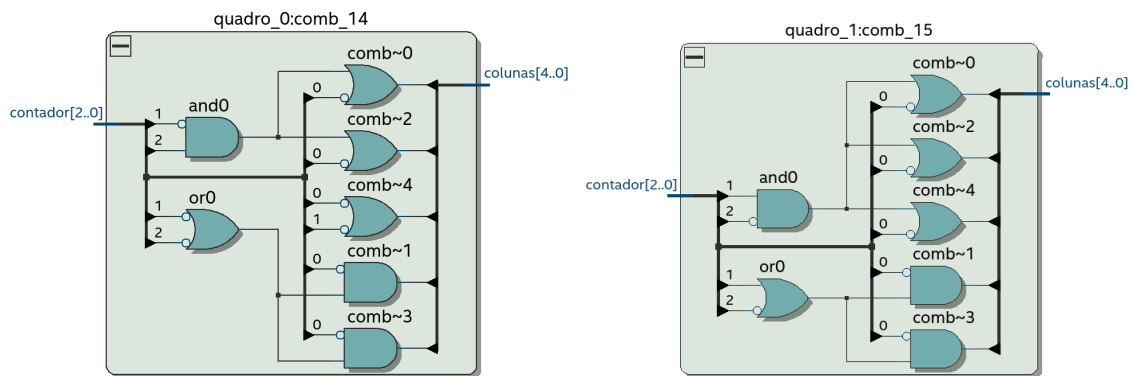
Para dividir a frequência do sinal de entrada, o módulo usa flip-flops T (toggle) que são inicializados com um valor de entrada "1" na borda de subida do sinal "clock_in". O sinal de saída dos flip-flops é armazenado em um fio "out" que é usado para gerar os sinais de saída.

Para gerar o sinal "clock0", o módulo usa uma sequência de operações AND entre os sinais de saída dos flip-flops armazenados no fio "out". Cada operação AND é gerada por um bloco de flip-flops T e o resultado é armazenado em um fio "ands". No final da sequência de operações AND, o sinal "clock0" é gerado a partir da saída do último bloco de flip-flops T. O mesmo processo é usado para gerar o sinal "clock1", com sequências de operações AND diferentes.

Cada bloco de flip-flops T é implementado através da instância de um módulo chamado "FF_T", que é responsável por realizar a operação toggle. Esse módulo é instanciado várias vezes com diferentes entradas e saídas, conforme necessário para gerar as sequências de flip-flops.

4.5.4 Módulos dos Quadros

Figura 15. Módulos Quadro 0 e 1



Fonte: Representação gerada com o RTL Viewer

O módulo Quadro é responsável por ser um decodificador de ativação de LEDs que representam as letras (ou partes delas) que são exibidas na matriz. Eles têm dois sinais: um sinal do contador de 3 bits e um sinal de saída de 5 bits, representando as cinco colunas da matriz.

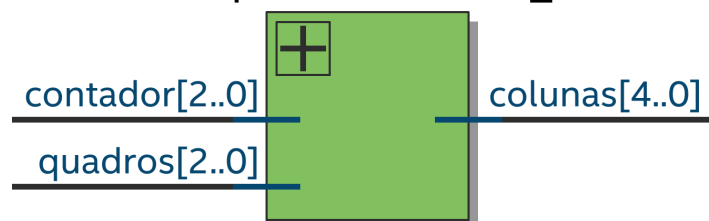
O código usa portas lógicas básicas como NOT, AND e OR para converter o quadro binário em uma representação adequada para exibição da parte correspondente. Cada porta lógica é usada para gerar uma coluna. As portas lógicas são interconectadas de maneira específica para gerar as cinco colunas da matriz.

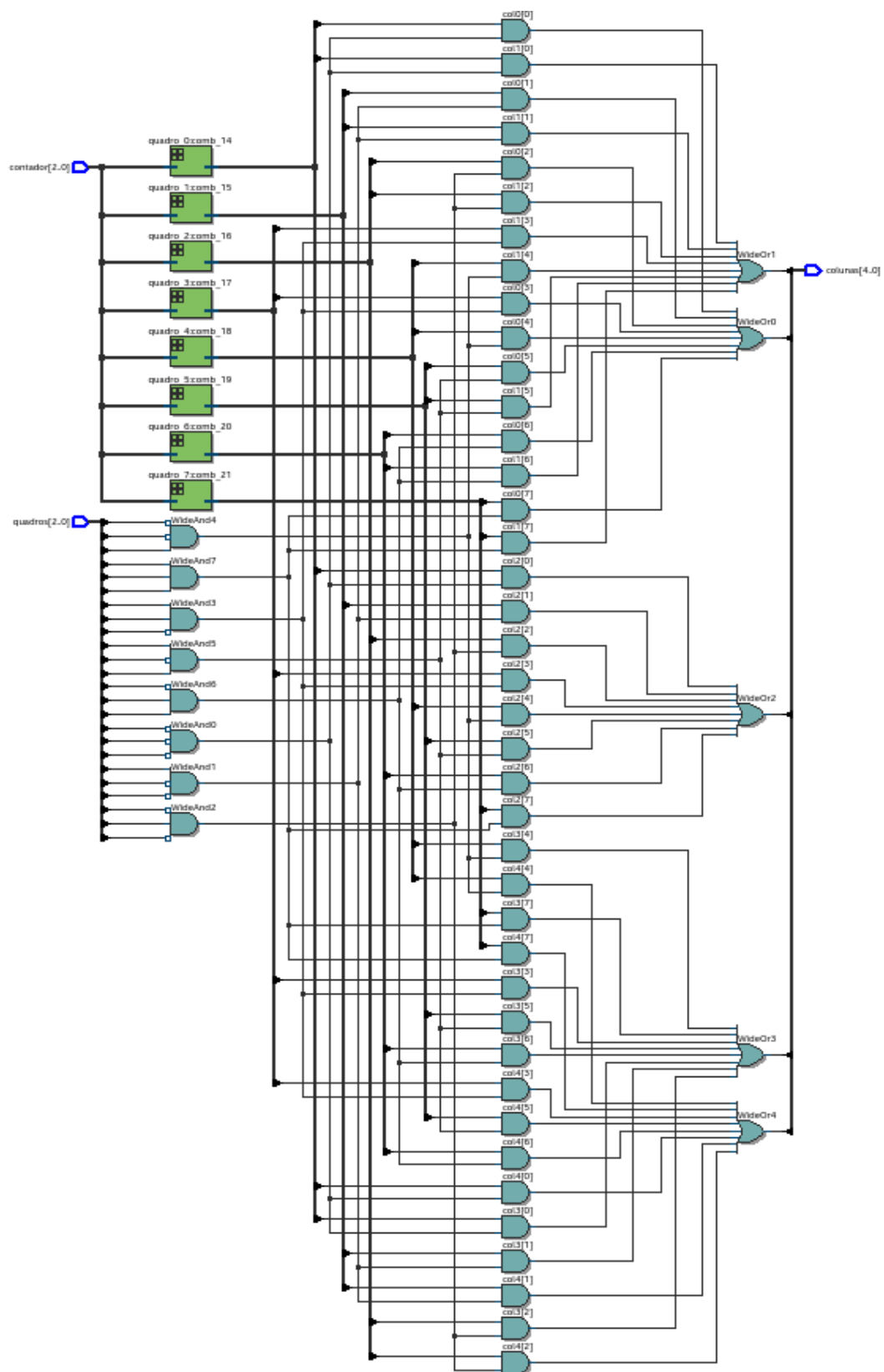
Na figura 15 são apresentados os módulos quadro correspondente aos quadros 0 e 1 porém os demais quadros (Ver seção 4.4) também são construídos de maneira semelhante, há apenas mudança nas expressões resultante das combinações de entrada de forma a representar corretamente as partes de cada uma das letras.

4.5.5 Módulo Multiplexador

Figura 16. Módulo Multiplexador

Multiplexador:comb_16





Fonte: Representação gerada com o RTL Viewer

O módulo Multiplexador é responsável por ser um multiplexador de 8 entradas e 4 saídas utilizando um circuito combinacional. O multiplexador recebe os sinais de 8

colunas de um conjunto de quadros (placas eletrônicas) e seleciona uma das colunas de cada quadro para ser a saída de cada uma das 4 saídas do multiplexador.

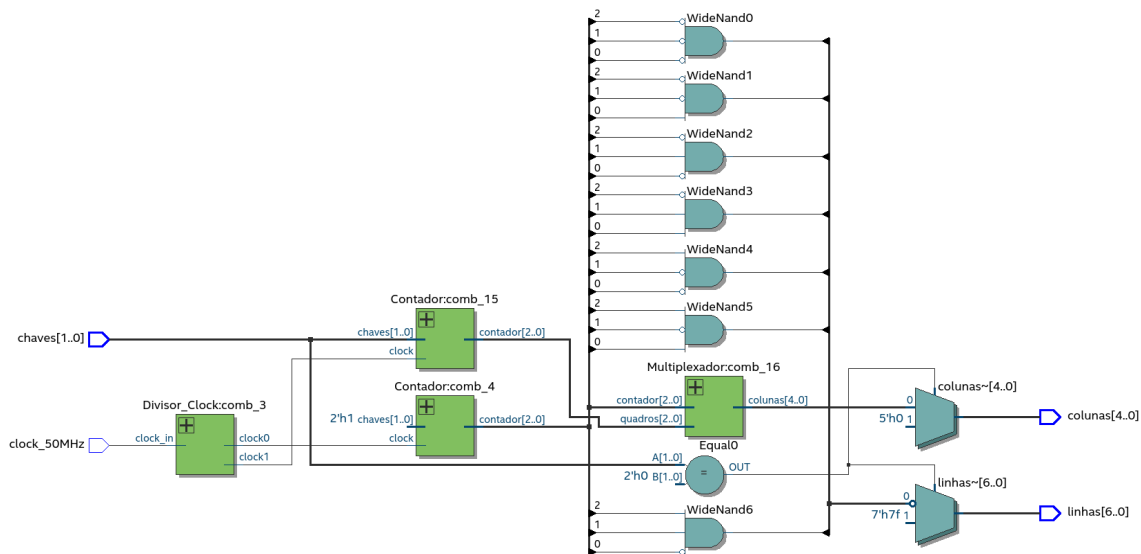
As entradas do multiplexador são os sinais de contador e os sinais dos quadros. O contador é utilizado para selecionar a coluna do quadro que será escolhida como saída. O sinal de quadros é utilizado para indicar quais colunas de cada quadro estão disponíveis para seleção.

A lógica do multiplexador é implementada com portas AND e OR. Primeiro, as portas AND são utilizadas para gerar sinais de seleção para cada coluna de cada quadro. Depois, as portas OR são utilizadas para selecionar qual coluna será a saída de cada uma das 4 saídas do multiplexador.

Existem oito fios de saída para as entradas de seleção de quadro (f0 a f7) e oito fios de saída para cada coluna de cada quadro (q0 a q7). O multiplexador usa portas AND para combinar esses sinais e produzir quatro sinais de saída para cada coluna do quadro selecionado. Em seguida, esses sinais são combinados usando portas OR para produzir o sinal final para cada coluna do quadro selecionado.

4.5.6 Módulo Varredura_Matriz

Figura 17. Módulo Varredura_Matriz



Fonte: Representação gerada com o RTL Viewer

O módulo Varredura_matriz é responsável por controlar e interligar todos os módulos, ele é o principal módulo do protótipo. O principal objetivo é criar a matriz de LEDs que exibe o letreiro “UEFS”, onde cada LED é controlado individualmente e é ligado ou desligado em um determinado momento, de acordo com a informação fornecida pelo circuito.

O módulo Varredura_Matriz recebe como entradas o sinal de clock, um sinal de controle com o estado das chaves, e tem como saídas as linhas e colunas que controlam os LEDs da matriz.

O código implementa a varredura da matriz de LED, definindo as linhas e colunas correspondentes para acender os LEDs de acordo com os sinais de controle. A lógica da varredura é baseada em um contador e em portas lógicas NAND, que definem as linhas que serão acionadas em cada momento.

A função do multiplexador é selecionar a coluna correspondente ao quadro (ou frame) atual da matriz, que é definido pelo sinal de controle de entrada. O multiplexador passa essa informação para o próximo bloco do circuito que controla a exibição de caracteres no display.

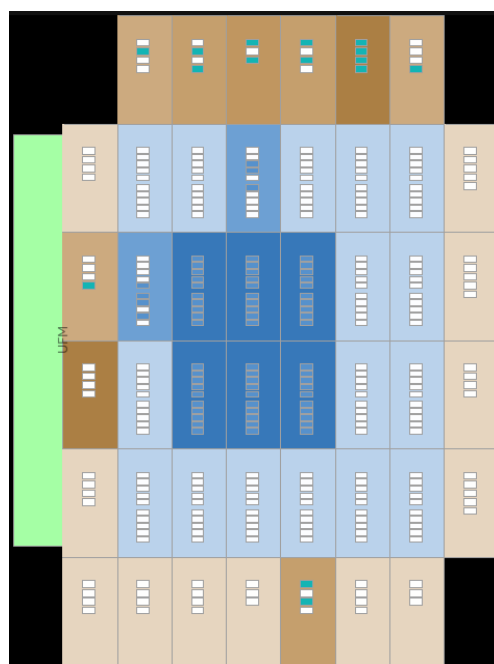
5. Resultados e discussões

O protótipo tem como objetivo principal ser um letreiro de matriz de leds que exibe a sigla “UEFS”. A construção do sistema foi feita de maneira que sua utilização fosse a mais simples possível, facilitando a experiência do usuário e usando o menor número de recursos do kit de desenvolvimento.

5.1 Uso de LABs, LEs, LUTs e Pinos

No protótipo, foram constatados os usos de 8 dos 24 LABs presentes no kit de desenvolvimento, o que corresponde a aproximadamente 33,3%. A quantidade de uso de LEs (elementos lógicos) foi de 67 dos 240 disponíveis, aproximadamente 27,9%, onde 32 foram utilizados com LUTs de quatro entradas, 17 com LUTs de três entradas, 15 com LUTs de duas entradas e 3 com LUTs de uma entrada. Em relação ao uso de pinos, foram utilizados 15 dos 80 disponíveis, o que corresponde a 18,75%. A figura 18 apresenta os LABs usados (blocos destacados em azul escuro), os LEs usados de cada um dos LABs e os pinos usados (blocos destacados em marrom escuro) na ferramenta Chip Planner.

Figura 18. Imagem gerada pela ferramenta Chip planner



Fonte. Elaborada pelos autores

5.2 Manual de Uso

Após código verilog ser implementado no kit de desenvolvimento LEDS-CPLD, o usuário pode iniciar o uso do sistema. O funcionamento se dá da seguinte forma:

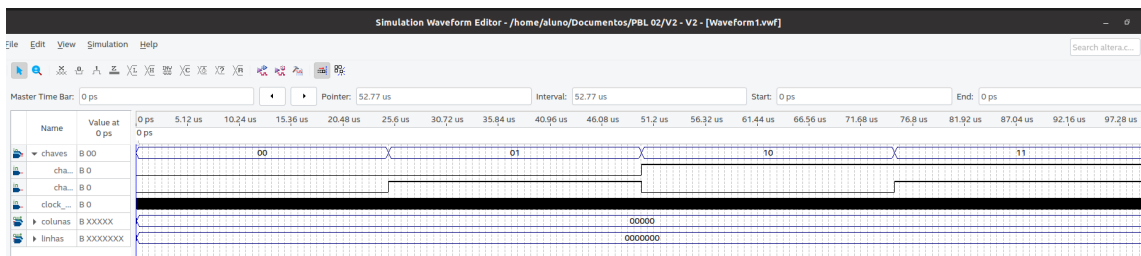
- Chaves CH1 e CH0 em nível lógico baixo (0) desligam a matriz de LEDs;
- CH1 em nível lógico alto (1) e CH0 em baixo (0) deslocam o letreiro da esquerda para a direita;
- CH1 em nível lógico baixo (0) e CH0 em alto (1) deslocam o letreiro da direita para a esquerda;
- CH1 e CH0 em nível lógico alto (1) interrompem o deslocamento do letreiro;

5.3 Testes

Para a realização dos testes virtuais, um arquivo University Program VWF (Vector Waveform File) foi criado na pasta do projeto para observação do comportamento e simulação dos códigos Verilog escritos para o protótipo.

Inicialmente o arquivo foi iniciado na ferramenta Simulation Waveform Editor e editado para que o tempo total da simulação fosse de 100 μ s, onde, cada uma das quatro combinações das chaves CH1 e CH0, tem um tempo de simulação de 25 μ s. O clock foi definido com uma variação de 50 MHz, que é o clock presente no CPLD.

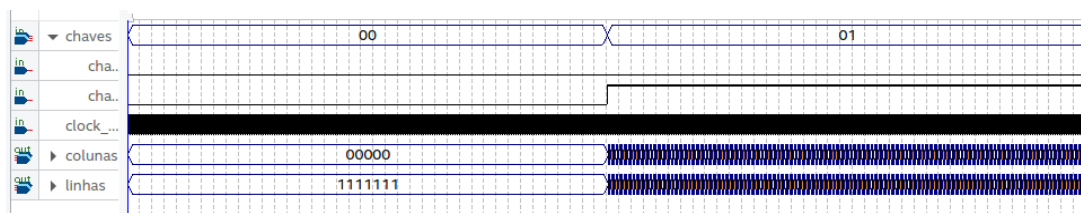
Figura 18. Edição de dados das entradas para simulação

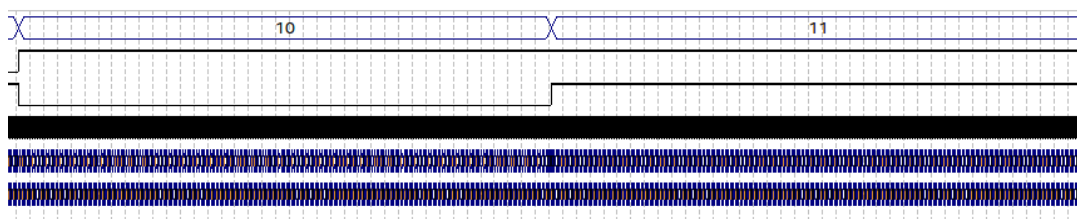


Fonte: Ferramenta Simulation Waveform Editor do software Quartus

Após a edição, a simulação foi gerada de maneira funcional para observar o comportamento do circuito conforme as entradas das chaves.

Figura 19. Simulação funcional do circuito



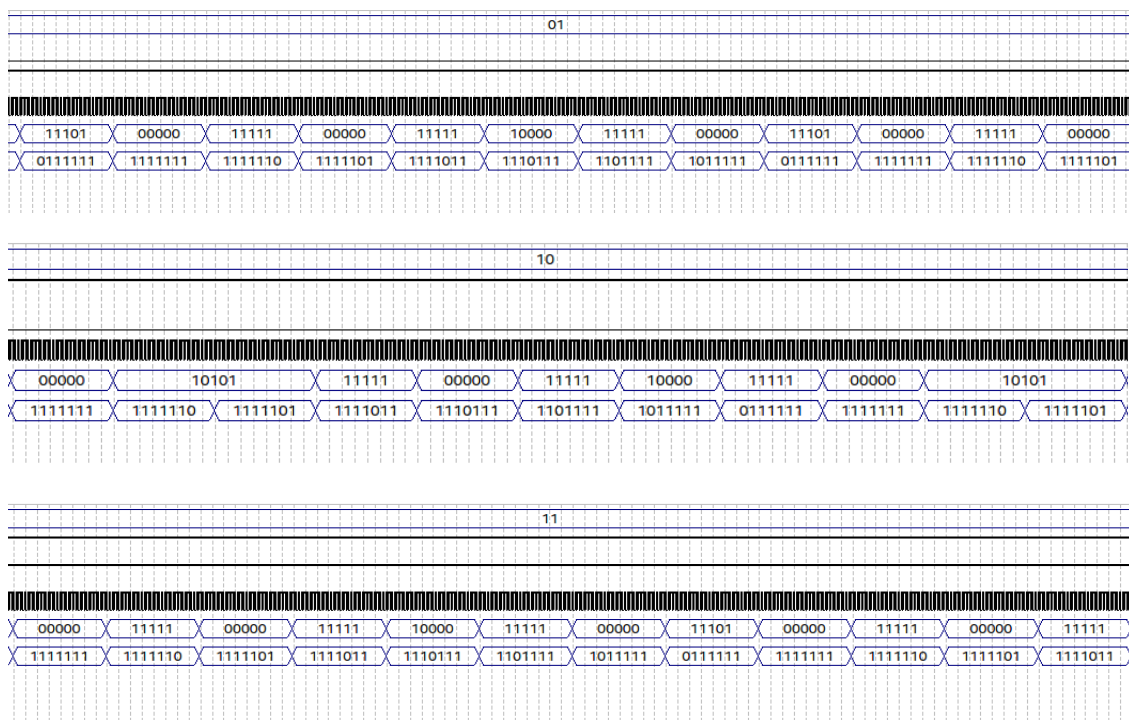


Fonte: Ferramenta Simulation Waveform Editor do software Quartus

Através da figura 19, é possível observar que, quando ambas as chaves estão em nível lógico baixo (0), as colunas recebem bits de nível lógico baixo e as linhas recebem bits de nível lógico alto, de modo com que a matriz de leds seja desativada conforme atendimento dos requisitos (Ver tabela 1). Para as demais chaves, há a varredura linha a linha (Ver seção 4.3) e há uma variação muito rápida dos níveis lógicos que a combinação de bits das colunas permite que os quadros (Ver seção 4.4) sejam formados e exibidos na matriz.

Na figura 20 são apresentadas as ativações das colunas e a varredura linha a linha quando as chaves estão em nível lógico 01, 10 e 11, vale ressaltar que, como mesmo que o circuito interrompa o deslocamento ao colocar ambas as chaves em nível lógico alto (1), ainda sim, o circuito faz o processamento dos quadros que apenas esperam pela parada com o acionamento das chaves.

Figura 20. Variação dos níveis lógicos recebidos pelas linhas e pelas colunas

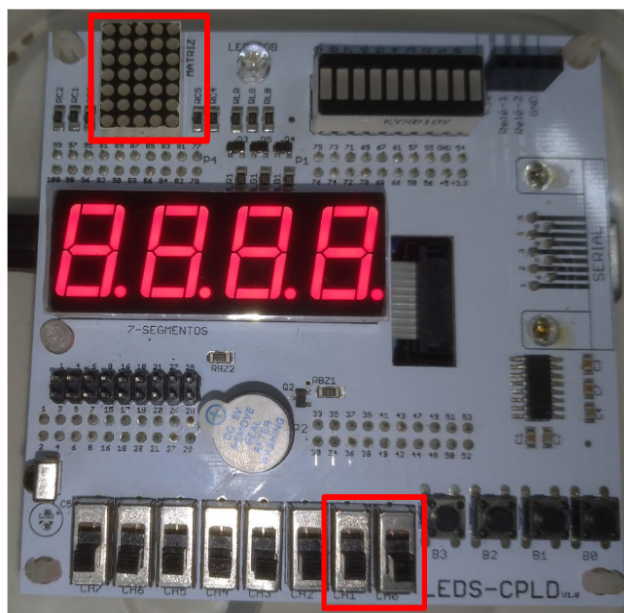


Fonte: Ferramenta Simulation Waveform Editor do software Quartus

Após a constatação do funcionamento em meio virtual, foram realizados testes com o código devidamente descarregado no kit de desenvolvimento. A seguir é apresentado o teste prático.

No primeiro teste, ambas as chaves foram postas em nível lógico baixo (0) o que resultou na desativação da matriz de LEDs conforme apresentado na simulação de ondas (Waveform). A figura 21 apresenta a CPLD com as chaves CH1 e CH0 em nível lógico baixo.

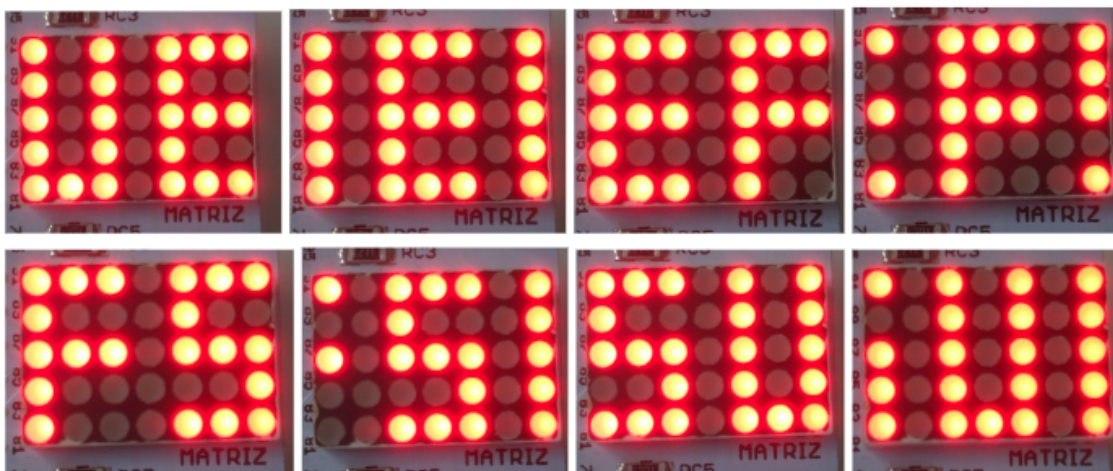
Figura 21. Matriz de LEDs desligada e chaves em 00



Fonte. Elaborada pelos autores

No segundo teste, a chave CH1 posta em nível lógico alto (1) e a CH0 em nível lógico baixo (0), que é responsável pelo deslocamento das letras da esquerda para a direita conforme solicitado nos requisitos (ver Tabela 1). O deslocamento ocorreu conforme apresentado na simulação em waveform. A figura 22, mostra o deslocamento dos quadros.

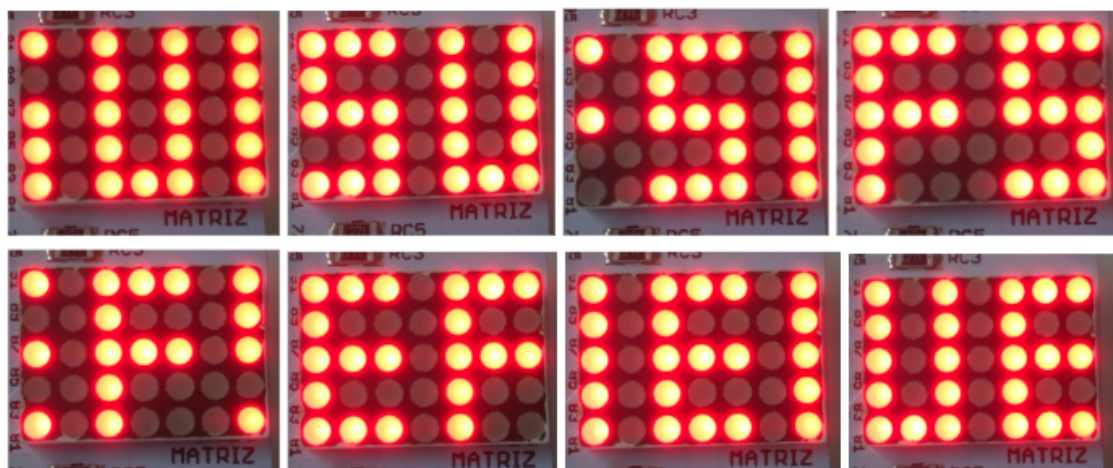
Figura 22. Quadros do deslocamento do letreiro na matriz de LEDS da esquerda para a direita.



Fonte. Elaborada pelos autores

O terceiro teste ocorreu de maneira semelhante, mas a Chave CH1 foi posta em nível lógico baixo (0) e a CH0 em nível lógico alto (1) que é responsável pelo deslocamento das letras da direita para a esquerda conforme solicitado nos requisitos (Tabela 1). O deslocamento ocorreu conforme apresentado na simulação em waveform.

Figura 23. Quadros do deslocamento do letreiro na matriz de LEDS da direita para a esquerda.



Fonte. Elaborada pelos autores

No quarto e último teste as chaves CH1 e CH0 foram postas em nível lógico alto (1) que é responsável pelo interrompimento do deslocamento do letreiro. Este teste se fez importante de modo foi possível observar que, devido ao tempo de acionamento das chaves, do atraso de processamento dos circuitos físicos e lógicos e de deslocamento e processamento dos quadros no protótipo, nem sempre o interrompimento se dá com o quadro observado, podendo ser interrompido no quadro atual ou no anterior ou no subsequente.

6. Conclusão

O desenvolvimento do protótipo resultou em um circuito que atendeu plenamente às especificações solicitadas, com excelente desempenho e uso reduzido de elementos lógicos. A otimização de hardware utilizada permitiu uma economia de recursos operacionais do CPLD, uma vez que apenas 27,91% dos LEs, 33,3% dos LABs e 18,75% dos pinos disponíveis foram utilizados, e o uso de LUTs de diferentes entradas possibilitou uma implementação eficiente sem comprometer a funcionalidade do circuito.

A baixa latência do desempenho garantiu uma resposta rápida e precisa na exibição das letras na matriz de LEDs, com uma comunicação visual clara e fácil observação da mensagem.

Os resultados dos testes virtuais e reais confirmaram o correto funcionamento do protótipo, o que indica uma solução efetiva para a exibição de mensagens em uma matriz de LEDs. Em suma, o circuito apresenta-se como uma opção eficiente e confiável para a aplicação desejada.

Referências

CAPUANO, Francisco G. ; IDOETA, Ivan Valeije. Elementos de Eletrônica Digital. 40ª ed. São Paulo: Érica. 544 p.

TOCCI, Ronald J. ; WIDMER, Neal S.; MOSS, Gregory L..Sistemas Digitais: Princípios e Aplicações. 11ª ed. São Paulo: Pearson, 2011. 830 p.

FLOYD, Thomas. Sistemas Digitais: Fundamentos e Aplicações. 9ª ed. São Paulo: Artmed, 2007. 888 p.

DIAS, Anfranserai M. Manual do kit LEDS-CPLD. Rev. 1.0. Disponível em: <https://drive.google.com/file/d/168zWIJU0rbnq3q8QJXnrwRY8iO6Ds2xQ/view>
Acesso em: 01 mar. 2023

Mundo de LED. 10 Motivos para usar painel de LED na sua empresa. Disponível em: <https://www.mundodeled.com.br/blog/10-motivos-para-usar-painel-de-led-na-sua-empresa>
Acesso em: 20 abr. 2023