

UNIVERSIDADE ESTADUAL DE FEIRA DE SANTANA

DEPARTAMENTO DE TECNOLOGIA

TEC498 PROJETO DE CIRCUITOS DIGITAIS

PROTÓTIPO DE LEITURA E EXIBIÇÃO DE CÓDIGO DE BARRAS 2 DE 5

Douglas Oliveira de Jesus, Laiza Araujo Gordiano Oliveira, Kevin Cordeiro Borges

Tutor: Armando Sanca Sanca

1 INTRODUÇÃO

Com a primeira patente atribuída em 1952 por Joseph Woodland e Bernard Silver, o código de barras surgiu como uma alternativa de solucionar o grande atraso e falta de precisão de leitura, digitação e registro de dados de maneira digital. Pois, devido a grande quantidade de produtos no varejo, por exemplo, seguindo a terceira revolução industrial, os métodos antigos não estavam mais dando conta de tanto registro, impossibilitando o acompanhamento de produção da época ([MILIES, 2020](#)). Atualmente, a utilização do código de barras é imprescindível em diversas áreas, como na logística, o qual é fundamental para o controle de estoque e distribuição de produtos; saúde, possibilitando a utilização para identificação de pacientes e medicamentos, prevenção de erros de administração e controle de medicamentos; educação, indústria, divulgação de dados, entre outras.

O desenvolvimento de projetos que trabalham com códigos de barras tem sido essencial para tornar a utilização dessa tecnologia cada vez mais abrangente e eficiente ([MILIES, 2020](#)). Com o avanço da tecnologia, além da implementação do código ter se tornado mais barata e acessível, surgiram novos tipos de códigos de barras, como o QR Code, que permitem o armazenamento de mais informações em um espaço menor, além dos EAN, DataBar, UPC, ITF-14, código 2 de 5, entre outros. Além disso, a integração dos códigos de barras com sistemas de informação têm permitido o armazenamento e análise de dados em tempo real, possibilitando uma gestão mais eficiente de processos.

O código 2 de 5, tendo destacado neste artigo, por sua vez, tem diversas funcionalidades. O PostNet 74210, por exemplo, é utilizado internacionalmente como um código identificador de registros postais, sendo utilizado, principalmente, nos Estados Unidos. Além desse, tem o modelo "entre 5", o de uso de caracteres, o EAN e o ITF. Os dois últimos são bastante utilizados em sistemas de identificação de produtos industriais e do varejo, sendo aplicado, também, mundialmente. Vale destacar que são várias as maneiras de implementação do código 2 de 5.

Portanto, na tentativa de desenvolver competências e habilidades voltadas para a identificação e conversão de códigos, desenvolvimento e interpretação de tabelas verdade, desenvolvimento de funções de portas lógicas e criação de códigos de descrição, a disciplina Módulo Integrador de Circuitos Digitais da UEFS usa, como problema incentivador, a criação de um protótipo de leitura e exibição de código de barras 2 de 5.

O problema solicita que o sistema deva receber um código 2 de 5 (interface de entrada IE01) e representá-lo de forma visual no display de sete segmentos (interface de saída IS01) ou na matriz de LEDs (interface de saída IS02) do kit de desenvolvimento LEDS-CLPD. A escolha da interface (IS01 ou IS02) que deverá ser exibida será selecionada pelo usuário (interface de entrada IE02). Caso o código inserido seja inválido, deverá ser exibido a identificação de um erro. Por fim, toda a construção do protótipo deverá ser feita utilizando conceitos teóricos de circuitos digitais combinacionais. As interpretações, escolhas e determinações do projeto foram feitas em sessões tutoriais, utilizando a metodologia Problem-based Learning (PBL).

2 METODOLOGIA

O desenvolvimento do projeto foi elaborado a partir de sessões tutoriais, laboratórios, criação de código e simulação e testes físicos na placa.

2.1 SEÇÕES

A maior parte das discussões e produções foram realizadas durante as sessões PBL, com toda a turma. Desde a leitura do problema até a elaboração de alguns trechos no Verilog, linguagem de descrição solicitada para o problema. Embora tenha existido uma dificuldade de adaptação aos conceitos da disciplina nas primeiras sessões, a turma conseguiu redirecionar o foco para o que interessava inicialmente: criação de tabelas verdade, identificações de funções lógicas e desenvolvimento do diagrama. Com esses dados, a criação do código de descrição no Quartus II (plataforma de desenvolvimento da Altera/Intel) ficou mais clara e concisa.

Sessões em laboratórios foram realizados para a noção física dos equipamentos e a interpretação de representação do código na placa. Esses laboratórios tiveram o suporte de roteiros que indicavam a utilização da plataforma Quartus, ensinando desde a criação de um projeto até a simulação dele.

2.2 RECURSOS UTILIZADOS

Em todo o processo de criação do protótipo, algumas ferramentas foram utilizadas como auxiliadoras no processo.

O Quartus II, em suas diversas versões, foi utilizado para a criação do projeto em Verilog ([RIBAS, 2017](#)). O programa conta com simuladores, apresentação de diagramas, determinação de pinos na placa, identificação de utilização de elementos lógicos, entre outros.

O Verilog é uma linguagem de descrição de hardware de fácil sintaxe e compreensão ([PALNITKAR, 2003](#)), possibilitando seu uso para modelagem de sistemas eletrônicos físicos. Ele tem uma parte estrutural, parecida com a sintaxe da linguagem de programação C, mas também conta com uma linguagem de mais baixo nível, a qual permite criação e relação de portas lógicas, que foram bastante utilizadas nesse problema.

Na parte de registro, o Overleaf foi utilizado como uma ferramenta que permite escrever em LaTeX, utilizando o modelo da ABNT disponibilizado no site do Módulo Integrador de

Circuitos Digitais da UEFS. O site foi escolhido por ter uma ótima estruturação de escrita e permitir o desenvolvimento em grupo dentro da plataforma. Além do Overleaf, também para registros, o Trello foi utilizado como uma tabela de registro do que foi tratado nas sessões, armazenando as ideias, fatos, questões e metas propostos pela metodologia PBL.

Fisicamente, o laboratório de hardware da UEFS (LEDS) foi utilizado como uma sala para a realização das sessões de laboratório, permitindo a realização dos testes nas placas, além da utilização de computadores como auxiliares na produção.

Os recursos bibliográficos estão apresentados na sessão de referência deste relatório.

2.3 ESCOLHA DO TIPO DO CÓDIGO 2 DE 5

Embora o modelo do código 2 de 5 EAN esteja mais próximo da utilização proposta, ele conta com 4 tamanhos de barras diferentes para a representação, enquanto o modelo ITF utiliza 2 tamanhos ([MILIES, 2020](#)) 1 e 0 em binário) para representar o código 2 de 5. Sendo assim, se torna mais viável na implementação do problema. Além disso, os outros tipos possuem pesos e aplicações diferentes do setor varejista e atacadista, o que acaba sendo um grande obstáculo para a escolha dos mesmos. Com isso, o código mais viável escolhido pela equipe foi o código de barras 2 de 5 padrão ITF com peso 12470.

Os códigos de barras Intercalados 2 de 5 (ITF) são os códigos numéricos de duas larguras que podem codificar informações de qualquer comprimento, desde que haja um número par de dígitos no código ([RUSS, 2013](#)). A informação é codificada com base na largura das barras e espaços, e exatamente 2 de cada 5 barras são largas. Seu antecessor, o código de barras Padrão ou Industrial 2 de 5, pode codificar a informação só na largura das barras, mas não nos espaços. Os códigos de barras ITF são geralmente utilizados para fins de distribuição e identificação em armazéns.

O código 2 de 5 possui 5 dígitos, dos quais 2 são nível lógico alto (1) e 3 são nível lógico baixo (0) ([RUSS, 2013](#)). Para entender o funcionamento e ter o representante em decimal, basta somar os pesos dos dígitos que são 1 em binário. Por exemplo, o código 01001 representa 2 em decimal, pois o primeiro dígito (LSB) tem peso 0, e o quarto dígito tem peso 2. Como 2 mais 0 é 2, então esse é o representante em decimal. Para representar o 0 em decimal, é utilizado os pesos 7 e 4, pois passam do número 9 em decimal.

2.4 ESCOLHA DE ENTRADAS E SAÍDAS

O kit de desenvolvimento LEDS-CPLD possui diversas entradas e saídas que podem ser utilizadas por diversos fins. Como o problema do componente curricular informa sobre a exibição do código 2 de 5 representado em decimal, a equipe optou por utilizar o display de sete segmentos. Referente a matriz de LEDs, o próprio kit de desenvolvimento já possui uma configuração para essa saída.

Como entrada, a equipe percebeu que o conjunto de chaves (8) conseguem representar as duas interfaces solicitadas (de inserção do código 2 de 5 e da escolha em qual interface de saída o código será exibido). Com as 5 últimas chaves, identificadas no Manual do Kit LEDS-CPLD

como CH4, CH3, CH2, CH1 e CH0 e representadas no projeto como E4, E3, E2, E1 e E0, respectivamente, a interface IE01 permite a entrada do código 2 de 5. Com as 3 primeiras chaves, CH7, CH6 e CH5 e no projeto como E7, E6 e E5, por permitir 8 possibilidades (2 elevado a 3), visto que são 2 para cada entrada, com 3 entradas, a interface IE02 permite a escolha da interface a qual será exibida o código 2 de 5. Essas entradas da IE02 funcionam com código binário padrão, o qual se as 3 entradas forem nível lógico baixo o display irá exibir o código e, caso seja diferente disso, irá exibir na linha colocada em binário nessa interface. Na Figura 2 da sessão de Descrição do Código é representado a tabela verdade que demonstra isso.

2.5 DECODIFICADORES E MULTIPLEXADORES

Antes de apresentar a descrição do código, é interessante pontuar conceitos de circuitos digitais que foram desenvolvidos no código: uso de decodificadores e multiplexadores.

Um decodificador em circuitos digitais é um componente eletrônico que recebe um sinal de entrada codificado e o converte em um sinal de saída decodificado ([FLOYD, 2007](#)), geralmente em formato binário. Ele é utilizado para decodificar informações codificadas em diferentes formatos, tais como códigos de barras, números em formato BCD (Binary Coded Decimal), ou outras formas de codificação. O decodificador recebe um conjunto de bits de entrada e ativa uma das suas saídas correspondentes, dependendo da combinação de bits recebida. Cada saída é ativada para uma combinação específica de bits de entrada, e todas as outras saídas permanecem desativadas. O número de saídas ativas pode variar de acordo com o número de bits de entrada, mas normalmente é uma potência de dois.

Para controlar um display de sete segmentos, por exemplo, é necessário um decodificador que converta o número, ou caractere, a ser exibido em uma combinação de segmentos a serem ligados. O decodificador recebe os bits de entrada e ativa as saídas correspondentes para ligar os segmentos apropriados do display ([TOCCI; WIDMER; MOSS, 2010](#)). Por exemplo, se o número 7 for enviado ao decodificador, ele ativará as saídas correspondentes aos segmentos a, b, c para exibir o dígito 7 no display. Assim, o uso de um decodificador é fundamental para a operação de um display de sete segmentos, pois permite que os dados sejam convertidos em sinais de controle que ligam ou desligam os segmentos do display de forma adequada.

O multiplexador, por sua vez, é um componente eletrônico que permite selecionar um de vários sinais de entrada e direcioná-lo para a saída ([FLOYD, 2007](#)). Ele é usado para economizar espaço e reduzir a complexidade em sistemas digitais, permitindo que múltiplos sinais sejam roteados para um único canal de saída. O funcionamento de um multiplexador é relativamente simples: ele possui várias entradas e uma saída única. Além disso, ele tem um ou mais sinais de controle que determinam qual das entradas deve ser selecionada e enviado para a saída.

Uma matriz de LEDs é um conjunto de LEDs organizados em linhas e colunas, formando uma matriz bidimensional. Para acender um LED específico na matriz, é necessário selecionar a linha e a coluna correspondentes e enviar um sinal para a posição desejada. O uso de um multiplexador pode simplificar o processo de controle dos LEDs na matriz. Um multiplexador pode ser usado para selecionar a linha correta. Isso pode ser feito de forma rápida e eficiente, permitindo que os LEDs sejam controlados com precisão.

2.6 DESCRIÇÃO DO CÓDIGO

Nomeado como "identificadorDeCodigoDeBarras", o projeto principal recebe 8 entradas (E7, E6, E5, E4, E3, E2, E1, E0) a partir das chaves da placa LEDS-CPLD e possui 25 saídas, a serem descritas abaixo. Além disso, faz a importação de dois módulos: um responsável pela matriz de LEDs e o outro pela conversão do código 2 de 5 em decimal para o display de sete segmentos.

O módulo "matrizDeLeds", recebe todas as entradas e possui as saídas de todas as linhas (L6, L5, L4, L3, L2, L1, L0) e colunas (C4, C3, C2, C1, C0). A justificativa das escolhas foi de acordo com a capacidade da matriz de LEDs disponível na placa utilizada, que possui 7 linhas e 5 colunas. Como condutores de saídas de portas lógicas para outras portas lógicas (denominados 'wires', ou fios, em português), wENABLE e wERRO são responsáveis por verificar se as saídas da matriz devem ser acionadas ou não e os fios (w9-w0) conduzem as saídas da entrada do código 2 de 5.

Para definir a linha da matriz de leds (Figura 1), foi criado 7 portas do tipo 'not and' (NAND), em que ele altera o valor de 0 para 1 caso alguma entrada seja 0 e de 1 para 0 caso todas as entradas sejam 1. A escolha dessa porta lógica para a construção se deu pois as linhas são ativadas em nível lógico baixo. Então, quando um código binário padrão for escrito nas entradas E7, E6, E5, na ordem, respectivamente, do MSB para LSB, será enviado o nível lógico baixo para a saída. Sendo assim, as seguintes entradas representam a linha correspondente em binário (Tabela 1).

Como visto, o código 000 em binário não é representado na linha, pois esse define a escolha de ativação do display de sete segmentos. Ou seja, para escolher qual das interfaces será escolhida para exibir o código, basta manipular as entradas E7, E6, E5 da placa, o qual 000 ativa o display e, qualquer outra entrada, das 7 possibilidades restantes, definem a escolha da linha de acordo com a tabela (Tabela 1).

```
nand linha0 (L0, ~E7, ~E6, E5);  
nand linha1 (L1, ~E7, E6, ~E5);  
nand linha2 (L2, ~E7, E6, E5);  
nand linha3 (L3, E7, ~E6, ~E5);  
nand linha4 (L4, E7, ~E6, E5);  
nand linha5 (L5, E7, E6, ~E5);  
nand linha6 (L6, E7, E6, E5);
```

Figura 1 – Definição da linha da matriz

Após a escolha da linha, o código manipula a verificação se todo o código 2 de 5 foi colocado corretamente. Para isso, a equipe decidiu representar 10 possibilidades de acionamento para cada dígito (0-9) através de min-terms. Então, caso o código inserido nas chaves E4-E0 esteja representando um código 2 de 5, uma das 10 saídas terá nível lógico alto. Consequentemente, caso o código inserido não seja válido, todas as saídas terá nível lógico baixo (Figura 4). A escolha dessa lógica de construção foi definida pela facilidade em atribuir uma definição de erro. Com outras palavras, caso todos os fios de saída das portas AND (w0-w9) sejam 0, significa que o código inserido está incorreto e, por isso, será exibido erro ao usuário. Para isso, uma porta NOR tendo como entrada todas os wires w0-w9, define nível lógico alto em wERRO caso todos os fios sejam 0 (Figura 5).

Número binário	Linha da matriz de LEDs
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Tabela 1 – Conversão binário-decimal para escolha da linha da matriz

ENTRADAS			SAÍDA						
E7	E6	E5	L0	L1	L2	L3	L4	L5	L6
0	0	0	1	1	1	1	1	1	1
0	0	1	0	1	1	1	1	1	1
0	1	0	1	0	1	1	1	1	1
0	1	1	1	1	0	1	1	1	1
1	0	0	1	1	1	0	1	1	1
1	0	1	1	1	1	0	1	1	1
1	1	0	1	1	1	1	0	1	1
1	1	1	1	1	1	1	1	0	0

Figura 2 – Tabela Verdade das linhas da matriz de LEDs

ENTRADAS					SAÍDA				
E4	E3	E2	E1	E0	C0	C1	C2	C3	C4
0	0	1	1	0	0	0	1	1	0
1	0	0	0	1	1	0	0	0	1
0	1	0	0	1	0	1	0	0	1
1	1	0	0	0	1	1	0	0	0
0	0	1	0	1	0	0	1	0	1
1	0	1	0	0	1	0	1	0	0
0	1	1	0	0	0	1	1	0	0
0	0	0	1	1	0	0	0	1	1
1	0	0	1	0	1	0	0	1	0
0	1	0	1	0	0	1	0	1	0

Figura 3 – Tabela Verdade das colunas da matriz de LEDs

Por fim, 5 portas são criadas para repassar o valor das entradas (E4-E0) para as saídas (C0-C4) caso o erro seja 0, ou seja, caso o código 2 de 5 inserido seja válido. Em resumo, caso as entradas E7-E5 sejam diferentes de 000 e não haja erro na inserção do código 2 de 5, o código 2 de 5 será exibido na linha definida (E7-E5) nas colunas de acordo com a entrada (E5-E0) do próprio código 2 de 5. Se houver erro, a matriz é apenas desativada, como demanda o problema.

```

nor ENABLE (wENABLE, E7, E6, E5);
and dig0 (w0, ~E4, ~E3, E2, E1, ~E0);
and dig1 (w1, E4, ~E3, ~E2, ~E1, E0);
and dig2 (w2, ~E4, E3, ~E2, ~E1, E0);
and dig3 (w3, E4, E3, ~E2, ~E1, ~E0);
and dig4 (w4, ~E4, ~E3, E2, ~E1, E0);
and dig5 (w5, E4, ~E3, E2, ~E1, ~E0);
and dig6 (w6, ~E4, E3, E2, ~E1, ~E0);
and dig7 (w7, ~E4, ~E3, ~E2, E1, E0);
and dig8 (w8, E4, ~E3, ~E2, E1, ~E0);
and dig9 (w9, ~E4, E3, ~E2, E1, ~E0);

```

Figura 4 – Verificação se todo código inserido é válido

```

nor ERRO (wERRO, w0, w1, w2, w3, w4, w5, w6, w7, w8, w9);

```

Figura 5 – Definição do erro

No segundo módulo, "cod2outof5_to_7seg" é definido o acionamento do valor decimal no display de sete segmentos de acordo com a conversão feita a partir do código 2 de 5. Esta descrição foi feita através de uma tabela verdade, que ativa as saídas dos LEDs representantes no display de 7 segmentos de acordo com as entradas do código 2 de 5. Sendo assim, cada saída (LED do display), será ativa de acordo com a entrada possível. Devido a configuração do código 2 de 5, não foi possível realizar otimizações das entradas, apenas correlacionar as saídas das portas AND com outras portas OR de outros LEDs do display de 7 segmentos. O display só liga se as entradas E5, E6 e E7 forem todas nível lógico baixo (ENABLE). Se o ENABLE tiver em nível lógico baixo mas o código inserido estiver incorreto, o código de erro também é exibido.

O código começa definindo todas as entradas (E7-E0) e 13 saídas: a, b, c, d, e, f, g, que são representantes dos LEDs do display; dig1seg, dig2seg, dig3seg, dig4seg que são todas as possibilidades de acionamentos dos dígitos disponíveis no display; ledR, ledG, que representam as cores vermelho e verde no LED que foi incluída como uma experiência adicional ao que é solicitado no problema para ajudar a mensagem de erro a identificar se o código inserido é válido ou não. Os fios são os mesmos representantes da matriz de LEDs.

Como na matriz, a identificação se o código 2 de 5 inserido é válido ocorre da mesma maneira. 10 portas AND definem se alguma saída é nível lógico alto para representar que o código inserido é válido ou se é inválida, caso todas as saídas das portas (w0-w9) sejam 0 (Figura 4).

Sendo contrário da matriz de LEDs, o wENABLE funciona para validar se as entradas E7-E5 são 000 em binário. Caso sejam, wENABLE recebe nível lógico alto e entra em uma porta AND junto com as saídas (w0-w9). Portanto, as novas saídas dessas portas, definidas como w00-w09, determinarão se um número decimal será exibido ou não no display. Caso alguma desses fios seja nível lógico alto, o display irá exibir e, caso todos sejam nível lógico baixo, significa que ou houve uma entrada inválida, ou o wENABLE não foi ativado (não foi selecionado 000 nas portas E7-E5, que seleciona o display) (Figura 6).

Os LEDs acionados no display de sete segmentos são representados pelas letras a-g (Figura 8) e o acionamento correto desses LEDs podem representar vários códigos e, dentre eles, de 0 a 9 em decimal e a letra "E" que foi escolhido como representante de erro, a ser exibido, pela

equipe. A tabela verdade do display de sete segmentos demonstra quais saídas são ativadas a cada código 2 de 5 válido é inserido (Figura 7).

```

and dig0Enabled (w00, w0, wENABLE) ;
and dig1Enabled (w01, w1, wENABLE) ;
and dig2Enabled (w02, w2, wENABLE) ;
and dig3Enabled (w03, w3, wENABLE) ;
and dig4Enabled (w04, w4, wENABLE) ;
and dig5Enabled (w05, w5, wENABLE) ;
and dig6Enabled (w06, w6, wENABLE) ;
and dig7Enabled (w07, w7, wENABLE) ;
and dig8Enabled (w08, w8, wENABLE) ;
and dig9Enabled (w09, w9, wENABLE) ;

```

Figura 6 – Determinação da escolha do dígito

TABELA VERDADE DO DISPLAY DE SETE SEGMENTOS															
ENTRADAS								SAÍDA							DECIMAL
E7	E6	E5	E4	E3	E2	E1	E0	G	F	E	D	C	B	A	
0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	1	1	1	0	0	1	1	1
0	0	0	0	1	0	0	1	0	1	0	0	1	0	0	2
0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	3
0	0	0	0	0	0	1	0	1	0	1	0	0	1	0	4
0	0	0	0	1	0	1	0	0	0	1	0	0	1	0	5
0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	6
0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	7
0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	8
0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	9

Figura 7 – Tabela Verdade do display de sete segmentos

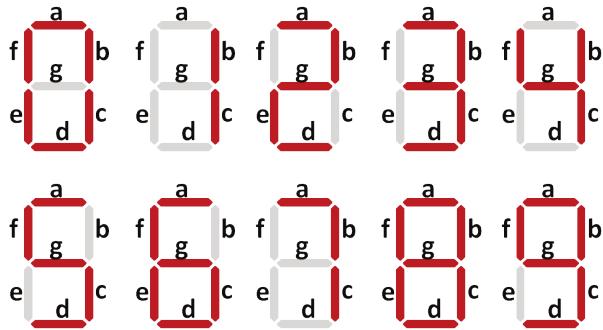


Figura 8 – Representação dos LEDs no display de sete segmentos

Se houver erro, ou seja, algum código não for escrito corretamente os LEDs a, f, g, e, d serão ligados – é importante destacar que os leds são ligados no nível lógico baixo. Se todos os wires w0-w9 forem 0, ou seja, o código inserido não for válido, haverá mensagem de erro (wERRO receberá 1). A necessidade de utilizar 20 wires (w0-w9 e w00-w09) surge para conseguir criar o wire de erro (Figura 5), pois deve ser independente do enable. As portas G-A só poderão ser ligadas caso algum dígito esteja correto e o ENABLE esteja ativado, ou seja, as saídas w00-w09 estejam ativas, ou caso haja erro, ou seja, caso algum código inserido seja inválido.

Os LEDS do display de sete segmentos são ligados quando algum número, em decimal, ou do erro, que esse LED representa no acionamento, está ativado. Sendo assim, um número

pode acionar vários LEDs, a representar em decimal de acordo com a imagem formada no display. Tomando a figura (Figura 9), é possível identificar um exemplo com o LED "D". Ele só será acionado se algum dos fios w00, w02, w03, w05, w06, w08, wERRO, estiver em nível lógico alto, devido a sua porta lógica NOR, que define saída em nível lógico baixo (que aciona o LED) quando pelo menos alguma das entradas é nível lógico alto. Em alto nível, se o dígito 0, 2, 3, 5, 6, 8 ou houver entrada inválida, ativando o wERRO, for nível lógico alto, o LED "D" será acionado. Pois, de acordo com a figura (Figura 8) esse LED pode representar todos esses números e a letra "E", que representa erro.

```
//As possibilidades de ligação de cada LED
nor G (g, w02, w03, w04, w05, w06, w08, w09, wERRO); //led g do sete segmentos
nor F (f, w00, w04, w05, w06, w08, w09, wERRO); //led f do sete segmentos
nor E (e, w00, w02, w06, w08, wERRO); //led e do sete segmentos
nor D (d, w00, w02, w03, w05, w06, w08, wERRO); //led d do sete segmentos
nor C (c, w00, w01, w03, w04, w05, w06, w07, w08, w09); //led c do sete segmentos
nor B (b, w00, w01, w02, w03, w04, w07, w08, w09); //led b do sete segmentos
nor A (a, w00, w02, w03, w05, w06, w07, w08, w09, wERRO); //led a do sete segmentos
```

Figura 9 – Possibilidades de ligação de cada LED

Como a lógica de acionamento do dígito também é ativa em nível lógico baixo, deve-se colocar em nível alto todas que não devem ser acessas. Sendo assim, dig1seg, dig2seg e dig3seg recebem 1 de maneira pré-definida, deixando apenas o dig4seg ser possível o acionamento ou não. O acionamento do dígito 4 só será efetuado caso algum dos dígitos (representados pelos fios w00-w09) ou wERRO esteja em nível lógico alto.

Em resumo do segundo módulo, o dígito será acionado no display de sete segmentos caso o conjunto das entradas (E4-E0) seja válida e o wENABLE estiver ativado, ou seja, caso E7-E5 seja 000 em binário ou um erro será exibido caso o código inserido nas entradas E4-E0 seja inválido, independente do wENABLE. Com isso, possibilita que o problema proposto seja resolvido de maneira completa. Adicionando funcionalidade, o ledR é acionado quando há erro e o ledG é acionado quando não há erro.

2.7 IMPLEMENTAÇÃO FÍSICA

A implementação do código escrito em Verilog deve ser repassada para os circuitos integrados de lógica programável complexa (CPLD) para utilização do protótipo físico.

Os CPLDs tem um funcionamento interno semelhante, mas mais complicados, aos integrados PAL e PLD. No CPLD existem vários sub-circuitos interconectados por uma matriz programável de interconexão global. Esses sub-circuitos recebem a nomenclatura de Logic Array Blocks (LABs) (Figura 10). Cada LAB contém blocos de I/O (entrada/saída), um circuito de expansão de termo produto e a quantidade de 4 a 16 macrocélulas. Cada macrocélula é estruturada com uma matriz AND conectada a uma matriz OR, para a implementação de soma de produtos, e flip-flops que, em alguns casos, são responsáveis pela realimentação de sinais na matriz AND (BROWN; ROSE, 1996) O modelo utilizado foi o CPLD MAX II EPM240T100C5N.

Quando o código é ainda compilado no Quartus, uma série de informações acerca da implementação são geradas. No resumo do uso de recursos do Fitter, é possível ver todas essas informações. A partir do código apresentado neste relatório, temos as seguintes informações: 51/240 (21%) elementos lógicos foram utilizados, todos no modo normal. Desses, no uso de

elemento lógico por número de entradas LUT, 28 são de 4 entradas, 13 são de 3 entradas e 10 são de 2 entradas. Por fim, são utilizados 33 pinos, de 80 disponíveis (41%).

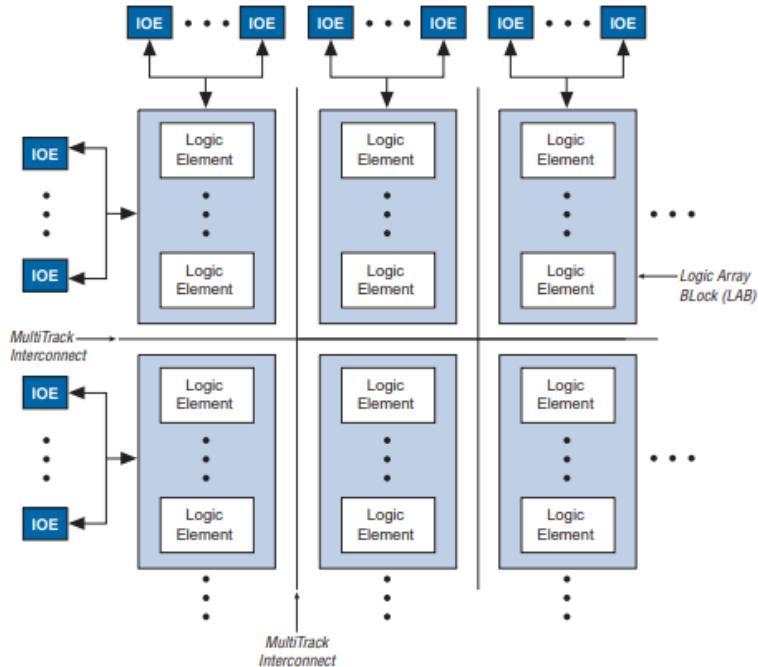


Figura 10 – Diagrama de blocos do dispositivo MAX II

2.8 TESTE E SIMULAÇÃO

Para realizar os testes físicos na placa, é recomendável realizar algumas simulações virtuais inicialmente, na tentativa de perceber os problemas antes de estarem conectadas na placa. Caso a simulação esteja correta, o ideal é que a aplicação física tenha um resultado adequado.

No primeiro teste, apresentado na ferramenta de simulação (Figura 13) na forma de onda da Figura X, as entradas E4-E0 seguem respectivamente 01100, acionando, em nível lógico baixo, os LEDs a, c, d, e, f, g. Como as entradas E7, E6 e E5 estão todas em nível lógico baixo e não há erro no código 2 de 5 inserido, o dígito 4 do display, que aciona em nível lógico baixo, é ativado. Na implementação física (Figura 11), o sistema funciona corretamente, tal qual a simulação.

No segundo teste apresentado (Figura 14), E4 recebe 0, E3 recebe 0 e E2, E1 e E0 recebem 1. Sendo assim, o código informado é inválido, pois tem mais de 2 entradas em nível lógico alto. É importante notar, também, que as chaves E7, E6 e E5 estão nos níveis baixo, alto e alto. Ou seja, representam a linha 3 da matriz de LEDs. Entretanto, devido ao invalidez, a matriz não é acionada e os LEDs a, d, e, f, g são acionados, representando o código de erro. Na placa (Figura 12), depois que o código é passado, acontece o que há na simulação: a matriz é desativada e é exibido no display de sete segmentos a letra "E", representando um erro de inserção de código.

Com isso, os testes físicos apresentam coerência com o que foi gerado nas simulações virtuais através das formas de onda.

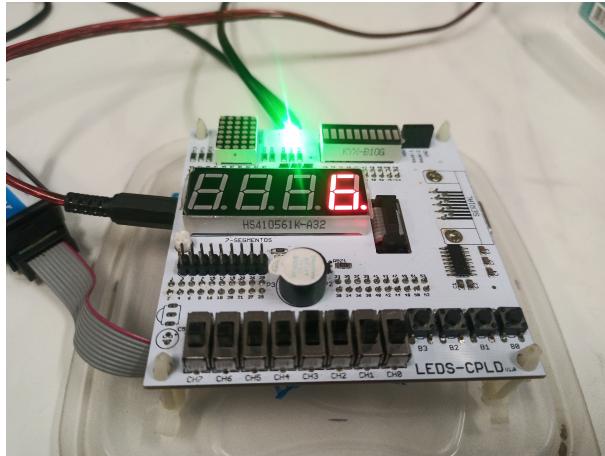


Figura 11 – Teste de entrada 01100 no código 2 de 5 no display

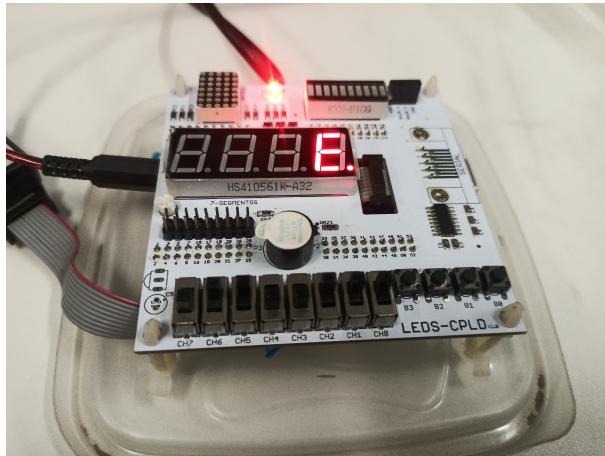


Figura 12 – Teste de entrada 00111 na linha 3 da matriz

3 RESULTADOS

O protótipo passou por alguns problemas de implementação, que foram corrigidos depois de algumas discussões e sessões. Entretanto, por fim, após realizar as simulações e todos os testes possíveis depois da implementação, foi considerado o preenchimento de todos os requisitos solicitados. O protótipo cumpre com maestria a conversão do código 2 de 5 em decimal, manipula o display de sete segmentos e a matriz de LEDs e lida com o erro nas variadas situações possíveis.

Por outro lado, a equipe aprendeu a manipular os códigos de descrição de hardware no mais baixo nível, relacionando-o com as variadas funções lógicas de saídas. Ademais, desenvolveu habilidade de interpretação, manipulação e conversão de alguns códigos numéricos existentes e de reconhecimento de padrões e relações entre portas lógicas, desenvolvendo decodificadores e multiplexadores.

De forma adicional, mas fora do que foi solicitado, o projeto poderia passar por atualizações para importação de um módulo que lidasse com a leitura real do código de barras, além da inserção manual do código. Ainda dentro do tema proposto, ao invés de simplesmente converter, poderia armazenar os dados lidos e manusear o possível gerenciamento desses dados.

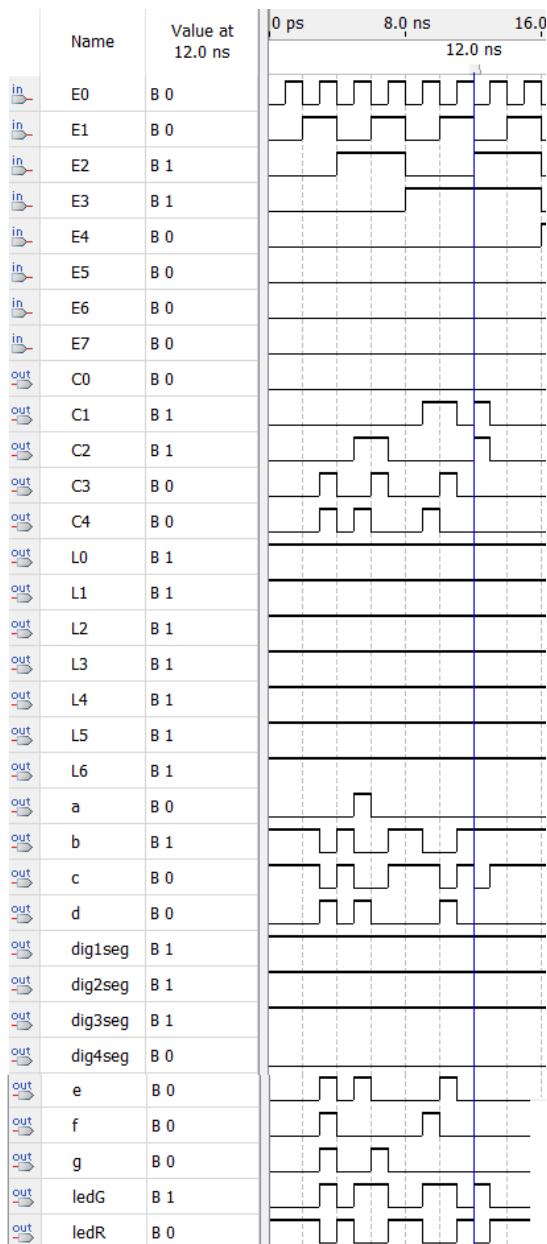


Figura 13 – Captura de tela da simulação de entrada 01100

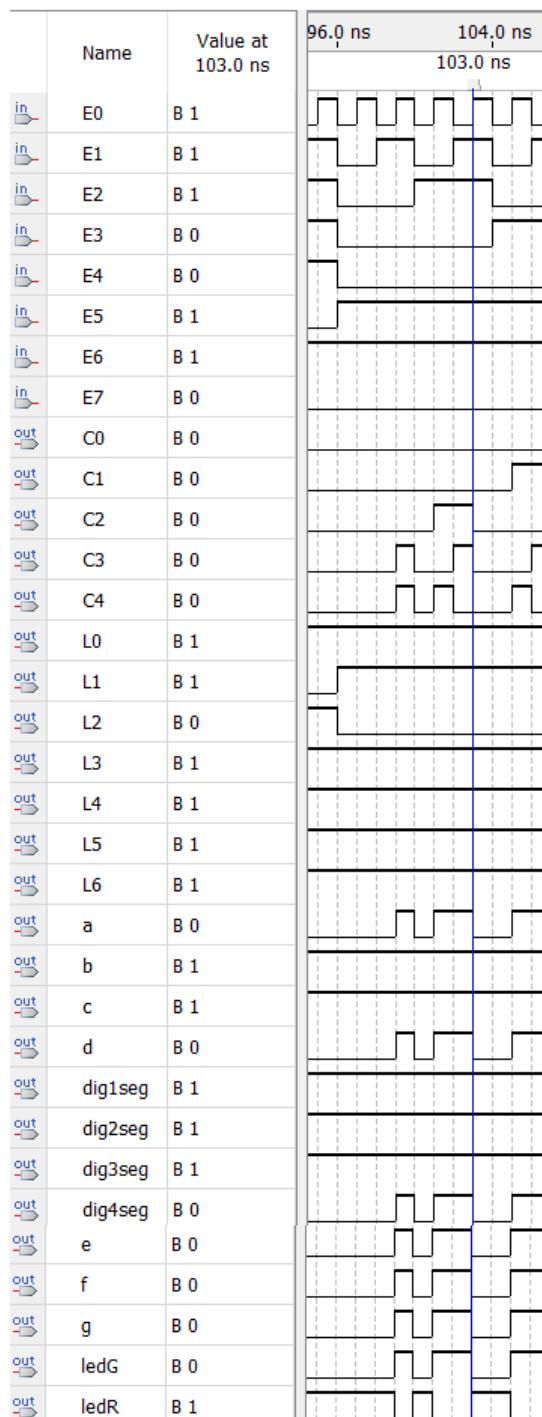


Figura 14 – Captura de tela da simulação de entrada 00111

REFERÊNCIAS

- BROWN, S.; ROSE, J. Fpga and cpld architectures: a tutorial. *IEEE Design Test of Computers*, v. 13, n. 2, p. 42–57, 1996. Citado na página [9](#).
- FLOYD, T. *Sistemas Digitais: fundamentos e aplicações*. 7. ed. [S.l.]: Bookman Editora, 2007. ISBN 978-85-7780-107-7. Citado na página [4](#).
- MILIES, C. P. A matemática dos códigos de barras. *RPM 65, IME, USP.*, 2020. Citado na página [3](#).
- PALNITKAR, S. *Verilog HDL: a guide to digital design and synthesis*. [S.l.]: Prentice Hall Professional, 2003. v. 1. Citado na página [2](#).
- RIBAS, C. E. Arquitetura fpgas e cplds da altera. 2017. Citado na página [2](#).
- RUSS, A. All about interleaved 2 of 5 barcode. *BarCode*, 2013. Citado na página [3](#).
- TOCCI, R. J.; WIDMER, N. S.; MOSS, G. L. *Sistemas digitais*. [S.l.]: Pearson Educación, 2010. Citado na página [4](#).