

Circuito identificador de códigos de intercalado 2 de 5

Lucas Gabriel Cerqueira Santos Lima
Luis Felipe Pereira de Carvalho

Departamento de Tecnologia - Engenharia de Computação

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

lucasfacul20222@gmail.com

luisoftbr@outlook.com.br

Tutor: João Bosco Gertrudes

Abstract. *This project details the construction of a circuit, which aims to be an identification and barcode display system, the code type chosen was the interleaved 2 of 5. Its development focused on the use of basic logic gates arranged in a combinational way 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 sistema de identificação e exibição de códigos de barras, o tipo de código escolhido foi o intercalado 2 de 5. Seu desenvolvimento teve foco no uso de portas lógicas básicas dispostas de forma combinacional 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 primeiro problema: Identificador de código de barras.

Segundo o Sebraeatende (2021), os Códigos de Barras (Barcodes) são representações gráficas de números ou caracteres alfanuméricos que reunidos, fornecem informações, de algo ou alguém (como exemplo, o número de funcionário) de modo a ser utilizado como instrumento de identificação e comunicação.

Existem vários tipos de códigos de barras diferentes, como UPC, Code 39 ou 128, EAN 8 ou 13, Codabar e interleaved 2 of 5 (ITF). Para a construção e execução do projeto é utilizado o código intercalado 2 de 5 (interleaved 2 of 5). Segundo o Keyence, o código 2 de 5 é um código que consiste na existência de apenas dois bits de nível alto (1) entre cinco bits. Este código é implementado com um bit de paridade, em geral o quinto bit, e em forma ponderada (pesos variando conforme a posição) nos demais quatro bits.

Para atender a construção do problema 1, 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 identificação e exibição de código de barras no qual é realizada a identificação de um código 2 de 5 e sua representação de forma visual.

Na seção seguinte 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.

2. 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 os materiais utilizados, os requisitos solicitados para a conclusão do problema e uma breve definição sobre níveis lógicos, álgebra booleana, circuitos combinacionais e portas lógicas.

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) que é pertencente à família MAX II, com o modelo especificado EPM240T100C5N de dispositivos Intel Altera.



Figura 1. Placa de desenvolvimento CPLD MAX II EPM240

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.

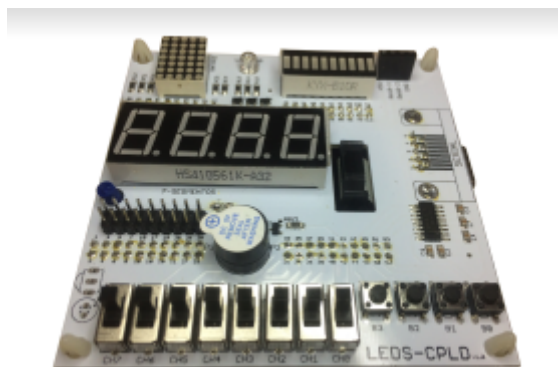


Figura 2. Placa LEDS-CPLD

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, é 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.

2.3 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.

2.4 Álgebra Booleana

Segundo Tocci (2011), a álgebra booleana é utilizada para a análise e simplificação de sistemas lógicos, ela emprega o uso de símbolos e operações para representar uma expressão lógica que possui um de dois valores possíveis: verdadeiro ou falso, no caso de circuitos lógicos 0 ou 1 (nível baixo ou alto).

A álgebra booleana possui esse nome pois em 1854, o matemático chamado George Boole escreveu a obra “Uma investigação das leis do pensamento”, em que ele descrevia o modo como se toma decisões lógicas com base em circunstâncias verdadeiras ou falsas. O método que ele descreveu é hoje conhecido como lógica booleana.

Ainda Segundo Tocci (2011), a álgebra booleana apresenta três operações básicas, OR (ou), AND (e) e NOT (não) que são denominadas operações lógicas. A partir dessas operações lógicas, foram permitidas as construções de métodos e técnicas como as expressões booleanas para a análise de circuitos lógicos com o uso da tabela verdade, que descreve como a saída de um circuito lógico depende dos níveis lógicos presentes nas entradas do circuito.

Figura 3. Exemplo de expressões booleanas

$$\begin{aligned}
 z &= ABC + A\bar{B}(\bar{A} + \bar{C}) && [\text{teorema (17)}] \\
 &= ABC + A\bar{B}(A + C) && [\text{cancela inversões duplas}] \\
 &= ABC + A\bar{B}A + A\bar{B}C && [\text{multiplica}] \\
 &= ABC + A\bar{B} + A\bar{B}C && [A \cdot A = A]
 \end{aligned}$$

Fonte: Tecnicas (2019)

Para a construção da lógica utilizada no protótipo, utilizamos das expressões booleanas conhecidas como mintermo e maxtermo.

Segundo o IFSC (2021), o mintermo é o termo produto associado à cada linha da tabela verdade, no qual todas as variáveis de entrada estão presentes. Dado um dado mintermo, se substituirmos os valores das variáveis associadas, obteremos 1. Porém, se substituirmos nesse mesmo mintermo quaisquer outras combinações de valores, obteremos 0.

Ainda segundo o IFSC (2021), maxtermo é o termo soma associado à cada linha da tabela verdade, no qual todas as variáveis de entrada estão presentes. Dado um dado maxtermo, se substituirmos os valores das variáveis associadas, obteremos 0. Porém, se substituirmos nesse mesmo maxtermo quaisquer outras combinações de valores, obteremos 1.

De modo a exemplificar, são apresentados na figura 3 os mintermos e maxtermos de um circuito combinacional de 3 entradas.

Figura 4. mintermos e maxtermos de um circuito combinacional de 3 entradas

A	B	C	Maxtermo	Mintermo
0	0	0	$A+B+C$	$\bar{A}.\bar{B}.\bar{C}$
0	0	1	$A+B+\bar{C}$	$\bar{A}.\bar{B}.C$
0	1	0	$A+\bar{B}+C$	$\bar{A}.B.\bar{C}$
0	1	1	$A+\bar{B}+\bar{C}$	$\bar{A}.B.C$
1	0	0	$\bar{A}+B+C$	$A.\bar{B}.\bar{C}$
1	0	1	$\bar{A}+B+\bar{C}$	$A.\bar{B}.C$
1	1	0	$\bar{A}+\bar{B}+C$	$A.B.\bar{C}$
1	1	1	$\bar{A}+\bar{B}+\bar{C}$	$A.B.C$

Fonte: IFSC(2021)

2.5 Circuitos combinacionais e Portas Lógicas

Partindo do esclarecimento sobre os níveis lógicos e a álgebra booleana, é importante salientar que, em circuitos mais complexos, nos nós (nodes) não há a dependência de uma única fonte de informação, em razão disto, os circuitos são definidos como circuitos combinacionais.

“Um circuito combinacional é todo circuito cuja saída depende única e exclusivamente das várias combinações das variáveis de entrada.”[Capuano, 2000]

De acordo com essa definição e segundo Floyd (2007), para realizarmos ações em níveis lógicos em circuito combinacional, são realizadas operações envolvendo portas lógicas. elas são consideradas circuitos eletrônicos que controlam a trajetória de uma informação (corrente elétrica), de modo a proporcionar uma saída especificamente construída conforme às entradas do circuito lógico.

Para tal, existem três portas lógicas básicas construídas a partir da álgebra booleana (Ver seção 2.4), NOT, AND e OR. A porta lógica NOT, ou inversora, tem como função, realizar a inversão ou complementação de um sinal recebido, ou seja, troca um nível lógico para seu nível oposto de 0 para 1 ou de 1 para 0).


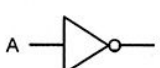




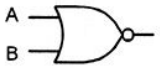
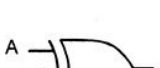
A porta AND, é utilizada para combinar duas ou mais entradas para que tenham uma só saída e ela ainda pode realizar a operação de multiplicação lógica. Temos ainda que, a saída de uma porta AND é em nível alto apenas quando todas as entradas estiverem em nível alto e a saída será em nível baixo quando uma ou mais entradas estiverem em nível baixo. Logo, para que seu funcionamento ocorra, todas as suas entradas devem estar no mesmo nível lógico.

A porta OR pode possuir duas ou mais entradas para uma única saída e realiza a operação de adição lógica, sua saída é nível alto quando uma ou mais entradas estiverem em nível alto e, ela será em nível baixo somente quando todas as entradas estiverem em nível baixo.

As demais portas lógicas, NAND, NOR, OR exclusivo (XOR) e NOR exclusivo (XNOR), são resultado da combinação das três portas básicas descritas, além destas,

existe a porta Buffer, que é uma entrada sem qualquer alteração lógica em sua saída. Na figura 4 são apresentadas as portas, suas representações mais comuns, suas tabelas verdade e suas expressões booleanas correspondentes.

Figura 5. Portas lógicas e sua tabela e equação booleana equivalentes

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

Fonte: Tecnicas (2019)

Na seguinte seção, é apresentado como se decorreu o desenvolvimento do protótipo de modo a atender a resolução do problema proposto.

3. Metodologia

Após a apresentação 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, o código intercalado 2 de 5 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).

3.1 Definição de requisitos

Como citado anteriormente, para a concretização do problema, são apresentados 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) para o código 2 de 5.
Interface de entrada (IE02),	O protótipo deve permitir ao utilizador escolher, através de uma interface de entrada (IE02), em qual linha da matriz de LEDs o código 2 de 5 inserido será exibido.
Interface de saída (IS01)	O protótipo deve conter uma interface de saída (IS01) para exibir o dígito decimal referente ao código 2 de 5 inserido.
Exibição na interface de saída (IS02)	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.
Escolha de interface de saída	O protótipo deve permitir ao utilizador escolher em qual interface de saída (IS01 ou IS02) o código 2 de 5 inserido será exibido, respeitando os respectivos formatos de saída de cada interface.
Processamento individual das interfaces de saídas	As interfaces de saída IS01 e IS02 devem ser mutuamente exclusivas.
Procedimento em caso de erro	Caso o código inserido não seja um código 2 de 5, o protótipo deve exibir um código de erro na IS01, desativando a IS02.
O circuito digital do protótipo deve ser puramente combinacional	Utilização apenas de combinação das portas lógicas básicas e suas variações.

Fonte: Elaborada pelos autores

3.2 Código Intercalado 2 de 5

Conforme apresentado na introdução, o código intercalado 2 de 5 que é utilizado neste protótipo consiste em um código que possui apenas dois bits de nível alto (1) entre cinco bits (exemplo: 10010).

De modo a realizar a utilização deste código no projeto, os conceitos de tabela verdade e valor ponderado posicional, foram aplicados na criação de uma tabela verdade para um código de 5 bits. Nesta tabela foram obtidas 32 variações posicionais das quais, 10 variações que possuem apenas dois dígitos binários em alto nível (1) foram validadas para representar os dígitos decimais de 0 a 9 além de essas variações respeitam a ponderação posicional que possui como sequência os pesos 0,1,2,3,6. Na tabela 2 estão dispostas as 10 combinações do código intercalado 2 de 5 e seus decimais correspondentes.

Tabela 2. Tabela verdade do código intercalado 2 de 5

Dígitos binários do código intercalado 2 de 5					
n4 (peso 0)	n3 (peso 1)	n2 (peso 2)	n1 (peso 3)	n0 (peso 6)	Decimal
0	1	1	0	0	0
1	1	0	0	0	1
1	0	1	0	0	2
1	0	0	1	0	3
0	1	0	1	0	4
0	0	1	1	0	5
1	0	0	0	1	6
0	1	0	0	1	7
0	0	1	0	1	8
0	0	0	1	1	9

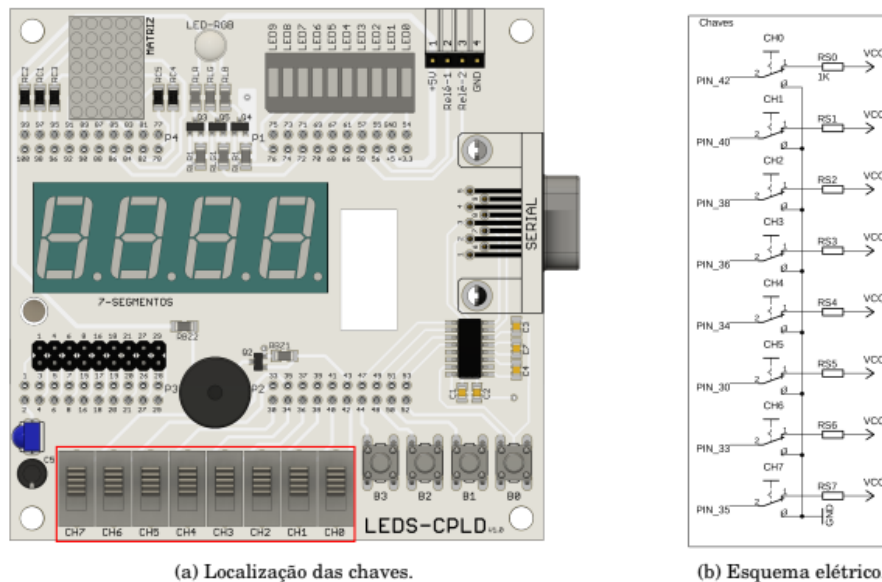
Fonte: Elaborada pelos autores

3.3 Apresentação das interfaces de entrada e saída

Para o desenvolvimento do protótipo foi imprescindível a definição de interfaces de entrada (IE01 e IE02) e saída (IS01 e IS02) dos dados inseridos. Foi realizada a escolha do conjunto de chaves (CH0 até CH7) como interface de entrada para IE01 e IE02. Para a interface de saída IS01 foi realizada escolha do quarto dígito do display de 7 segmentos e para a interface de saída IS02 foi escolhida a matriz de leds.

3.3.1 Chaves

Figura 6. Conjunto de Chaves utilizado como interface das entradas IE01 e IE02



Fonte: Manual Leds-CPLD

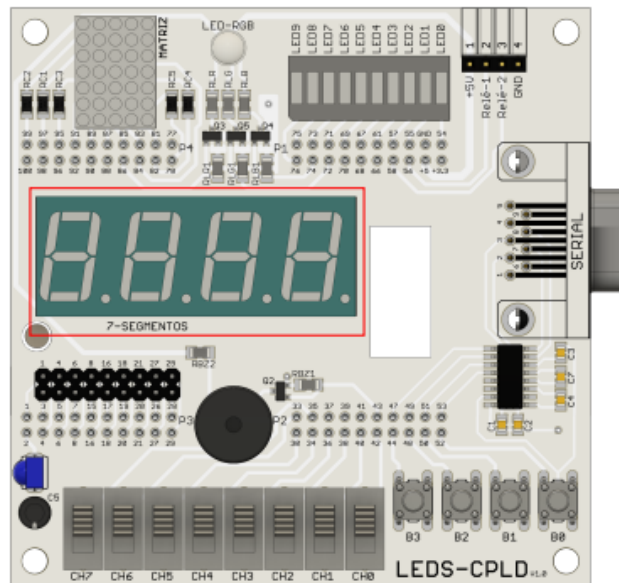
Segundo o manual do kit Leds-CPLD, as chaves fornecem um sinal de nível lógico alto (1) (chave para cima) ou nível lógico baixo (0) (chave para baixo) para o CPLD.

Para o protótipo, as chaves de CH2 a CH0 são utilizadas para que o usuário faça a escolha de qual linha da matriz de leds (ver figura 9) (ISO2) ele deseja utilizar para exibir os dados ou se ele deseja exibir no display de 7 segmentos (IS01) mediante combinação binária que será descrita adiante.

As chaves CH7 a CH3 são utilizadas pelo usuário para inserir o código binário intercalado 2 de 5. Cada chave controla um dos bits, podendo ele ser inserido como em nível alto ou baixo.

3.3.2 Display de 7 segmentos

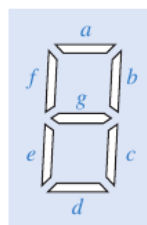
Figura 7. Display de 7 segmentos utilizado como a interface de saída IS01



Fonte: Manual LEDS-CPLD

Os displays de sete segmentos são métodos visuais para se observar caracteres alfanuméricos de um circuito. Eles são também usados em circuitos lógicos que decodificam um número codificado em binário (BCD) de modo a exibi-lo de forma apropriada.

Figura 8. Display de 7 segmentos e seu arranjo.

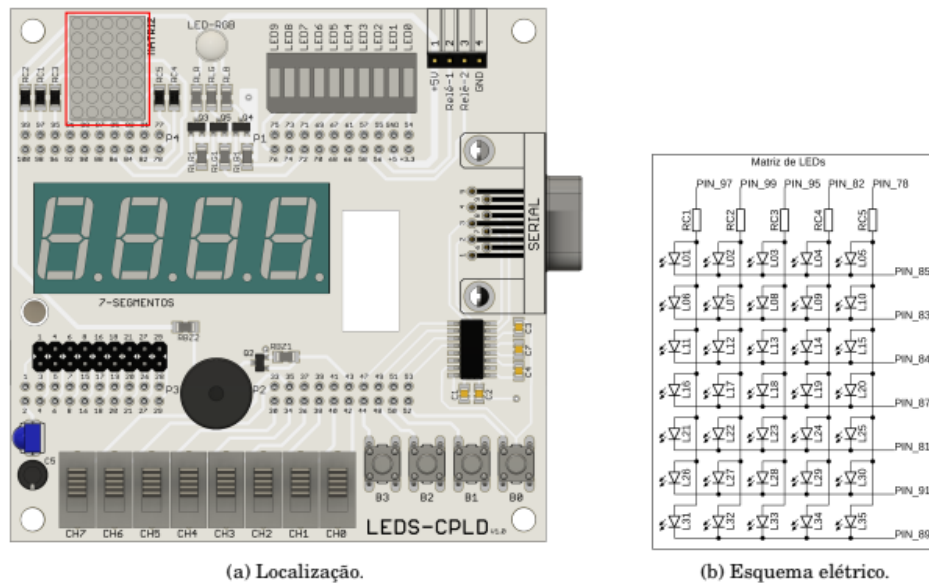


Fonte: Floyd (2007)

Para a construção da solução do protótipo, o display de 7 segmentos multiplexados do kit LEDS-CPLD foi analisado e obteve-se como constatação que, seu funcionamento se dá com o uso de uma alimentação para o dígito escolhido (neste caso foi utilizado o quarto dígito do display) que se mostra com seu segmento sempre ativos. Para que haja a exibição de um caractere é basilar o uso de uma lógica invertida nos pinos de alimentação dos segmentos, ou seja, para desligar um segmento do display, deve-se fornecer um sinal de nível lógico alto (1).

3.3.3 Matriz de LEDs

Figura 9. Matriz de leds utilizada como a interface de saída IS02



Fonte: Manual Leds-CPLD

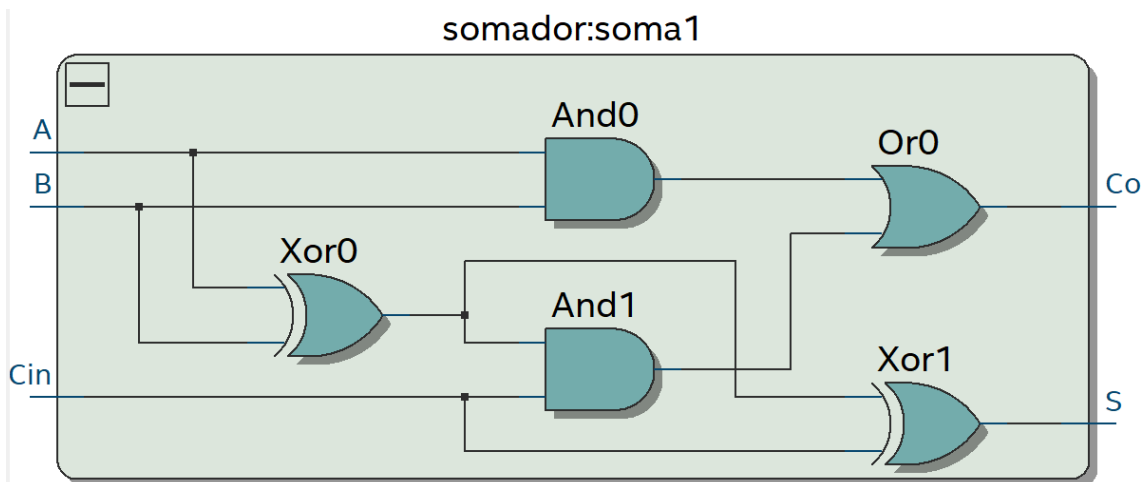
Segundo o manual do kit Leds-CPLD, a matriz de LEDs é composta por 5 linhas e 7 colunas conforme mostrado na figura 9. Cada linha e cada coluna está conectada a um pino do CPLD. As colunas são ativadas com nível lógico alto(1) e as linhas com nível lógico baixo(0) de modo que cada um dos 35 LEDs podem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

3.4 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 de módulos que foram subdivididos em 6 arquivos, com um deles sendo o principal. Foram desenvolvidos módulos, sendo eles: Main, Somador, Verificador2de5, Matriz_linhas, Matriz_colunas, e display7seg. Estes módulos são apresentados a seguir.

3.4.1 Módulo Somador

Figura 10. Módulo Somador



Fonte: Representação gerada com a ferramenta RTL Viewer

O módulo Somador é responsável por realizar a soma de dois bits (A e B) e um bit de “carry-in” (Cin) para gerar um resultado de soma (S) e um bit de “carry-out” (Co). O módulo possui cinco portas: A, B, Cin (entradas) e S, Co (saídas). O módulo também faz uso de três fios (T1, T2 e T3) para o processo de soma.

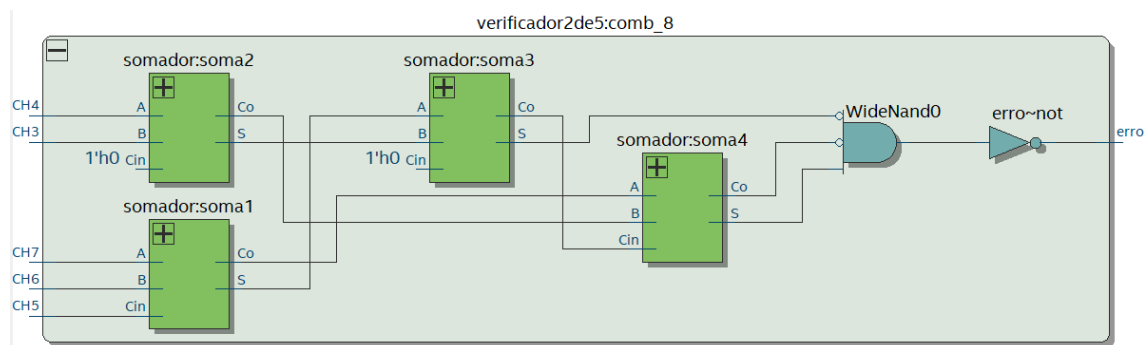
O primeiro fio T1 é usado para armazenar o resultado da operação lógica XOR entre as entradas A e B, o que gera a soma parcial. O segundo fio T2 é usado para armazenar o resultado da operação lógica AND entre as entradas A e B, o que gera o “carry gerado”. O terceiro fio T3 é usado para armazenar o resultado da operação lógica AND entre o somatório parcial T1 e o carry-in Cin, o que gera o “carry propagado”.

A saída Co é gerada a partir de uma operação lógica OR entre o carry gerado T2 e o carry propagado T3. A saída S é gerada a partir de uma operação lógica XOR entre o somatório parcial T1 e o carry propagado T3.

Este módulo usa três tipos de portas lógicas: XOR, AND e OR para realizar as operações lógicas necessárias para a soma. Este módulo somador é utilizado no módulo “Verificador2de5” para realizar e validar a entrada dos 5 bits correspondentes ao código intercalado 2 de 5.

3.4.2 Módulo Verificador2de5

Figura 11. Módulo Verificador2de5



Fonte: Representação gerada com a ferramenta RTL Viewer

O Módulo Verificador2de5 é responsável por verificar se o código intercalado 2 de 5 inserido pelo usuário, é ou não, válido. O módulo recebe cinco entradas, sendo elas CH7, CH6, CH5, CH4 e CH3 que são as chaves usadas para inserir os dígitos do código 2 de 5 e possui também uma porta de saída, chamada de “erro”, que se manifesta conforme a validade do código inserido (Ver Seção 3.2), como sendo ele válido (0) ou inválido (1).

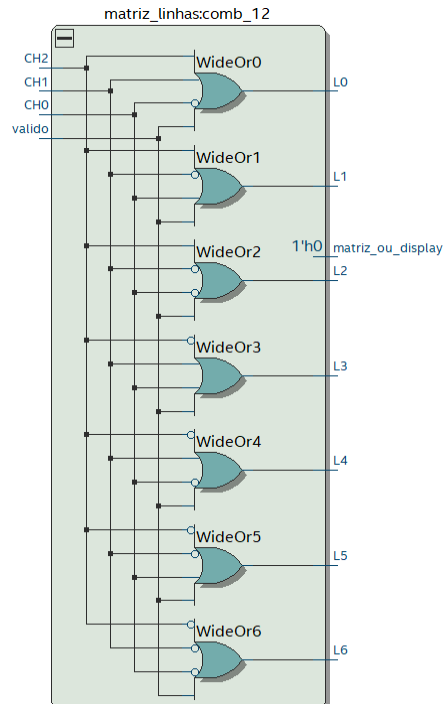
O módulo usa quatro módulos "Somador"(ver seção 3.4.1) para contar a quantidade de números um (1) disponíveis no código inserido. Os módulos de somador recebem os sinais de entrada correspondentes a cada par de dígitos do código e produzem as somas parciais (s1, s2, s3 e s4) e os resultantes dessas somas (Cout1, Cout2, Cout3 e Cout4).

O módulo Verificador2de5 usa dois inversores (NOT) para inverter os sinais de dois dos resultados das somas: o resultado final (s3) e o resultante final de soma (Cout4). Por último, o módulo usa uma porta lógica NAND para verificar se algum dos

três bits (nCout4, s4 e ns3) é zero. Se algum deles for zero, a porta NAND produz um sinal de erro (1), indicando que o código inserido é inválido. Caso todos possuam sinal lógico alto (1), é enviado 0 para a saída “erro”, indicando que o código inserido é válido.

3.4.3 Módulo Matriz_linhas

Figura 12. Módulo Matriz_linhas



Fonte: Representação gerada com a ferramenta RTL Viewer

O módulo Matriz_linhas é responsável por receber as entradas das chaves CH2, CH1 e CH0 que são usadas para inserir um código binário de 3 bits. Este código serve para realizar a escolha da linha da matriz a qual o usuário deseja exibir o código ou para exibir o decimal equivalente no display de 7 segmentos. Como o acionamento das linhas da matriz ocorre com sinal de nível lógico baixo, as entradas de sinais altos (1) são invertidas para os fios nCH2, nCH1 e nCH0, utilizando o conceito de maxtermo.

O sinal de entrada "válido" é utilizado para verificar se o código binário inserido é válido ou não. Este sinal vem do módulo “verificador2de5”, descrito anteriormente. Caso seja inválido, todas as linhas da matriz continuam apagadas pois recebem nível lógico alto (1), indicando que há um erro no código inserido.

O sinal "matriz_ou_display" é o sinal de controle resultado da combinação inversa dos bits das três entradas que determina se a matriz deve permanecer ligada ou não. Caso a combinação seja “000”, ou seja, as três chaves responsáveis pela escolha da linha da matriz estejam para baixo, o sinal “matriz_ou_display” recebe sinal lógico 1 que posteriormente será utilizado para desligar ou não o dígito 4 do display de 7 segmentos.

As sete linhas da matriz são representadas pelos sinais de saída L0, L1, L2, L3, L4, L5 e L6. Cada linha é controlada por uma porta OR que recebe como entrada os sinais de entrada "CH2", "CH1", "CH0" e o sinal "valido". O sinal "válido" é incluído na

porta OR para que, caso o código inserido for inválido, todas as linhas da matriz continuem apagadas. Cada porta OR é controlada por um conjunto diferente de sinais de entrada, resultando em uma exibição diferente na matriz para cada código binário inserido. Na tabela 3 são apresentadas as combinações das três entradas, a linha da matriz que é acionada por cada uma dessas combinações e a combinação responsável por ligar o display de 7 segmentos.

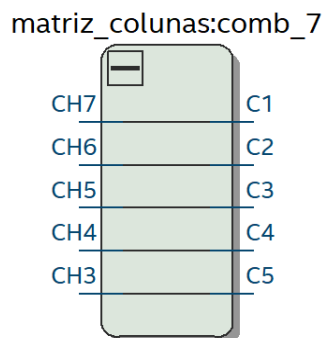
Tabela 3. Tabela combinacional de escolha de interface de saída

	Bits de sinal inseridos em cada chave		
Interface de saída	CH2	CH1	CH0
Linha 1 (L0)	0	0	1
Linha 2 (L1)	0	1	0
Linha 3 (L2)	0	1	1
Linha 4 (L3)	1	0	0
Linha 5 (L4)	1	0	1
Linha 6 (L5)	1	1	0
Linha 7 (L6)	1	1	1
Display de 7 segmentos	0	0	0

Fonte: Elaborada pelos autores

3.4.4 Módulo Matriz_colunas

Figura 13. Módulo Matriz_colunas



Fonte: Representação gerada com a ferramenta RTL Viewer

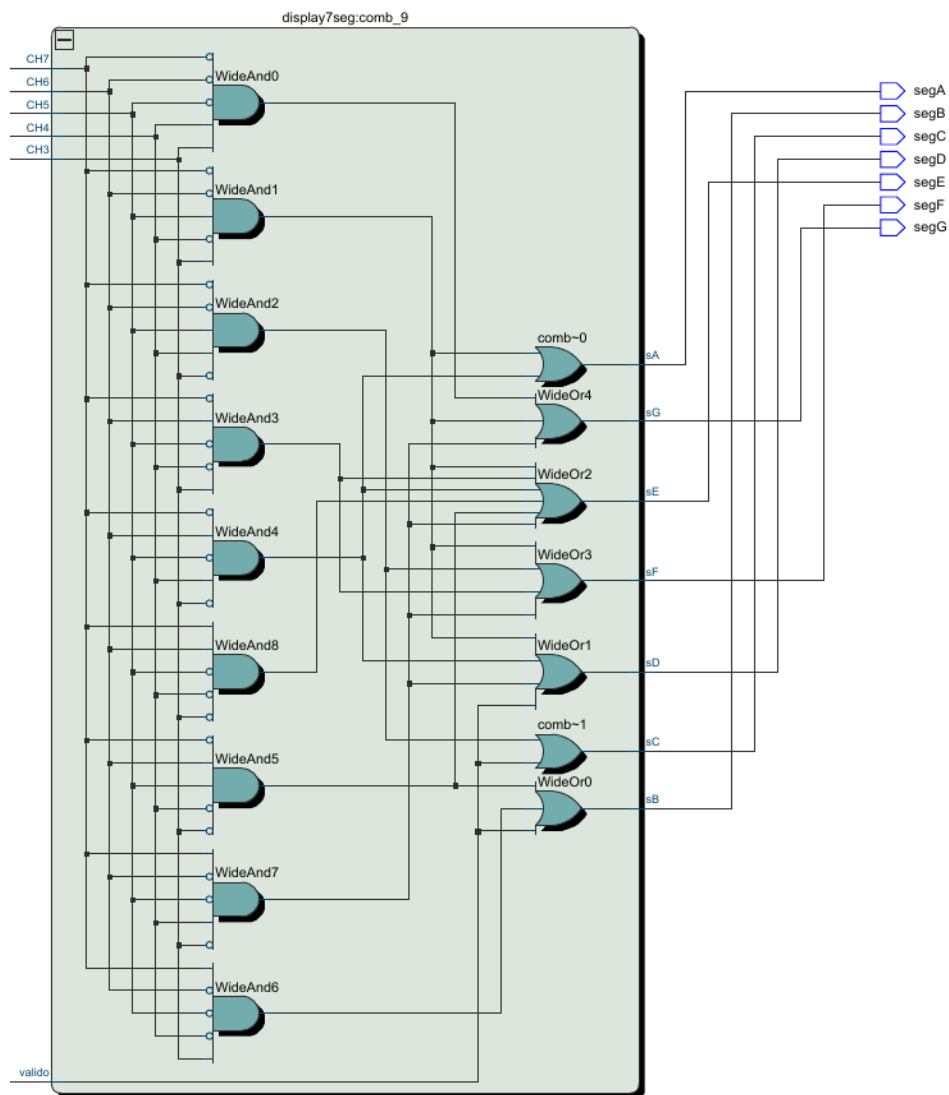
O módulo Matriz_colunas é responsável por conectar as entradas das chaves CH7 a CH3 diretamente às saídas das colunas da matriz de LEDs C1 a C5, sem a necessidade de codificação.

O módulo possui cinco saídas C1, C2, C3, C4 e C5, que representam as colunas da matriz de LEDs. As entradas são CH3, CH4, CH5, CH6 e CH7, que representam as chaves associadas a cada uma das colunas.

As entradas estão diretamente relacionadas às saídas das colunas. Nesse caso, C1 é atribuído a CH7, C2 é atribuído a CH6, C3 é atribuído a CH5, C4 é atribuído a CH4 e C5 é atribuído a CH3. Portanto, quando uma chave é acionada, o led correspondente na matriz de LEDs é ativado naquela posição.

3.4.5 Módulo Display7seg

Figura 14. Módulo Display7seg



Fonte: Representação gerada com o RTL Viewer

O módulo Display7seg é responsável por receber o código intercalado 2 de 5 e um sinal de validade que indica se o código de entrada é válido ou não. Os dados recebidos são codificados para serem exibidos no display de 7 segmentos (ver seção 3.3.2).

O código resultado da combinação das entradas CH7, CH6, CH5, CH4 e CH3 é primeiramente invertido em cada dígito, utilizando portas NOT. Em seguida, é feita a representação numérica dos códigos de entradas válidos usando portas AND, onde cada fio recebe a combinação referente a seu número decimal. A representação que é utilizada é a que está presente na tabela 2 (ver seção 3.2). Toda lógica de combinação desse módulo segue o conceito de soma de produtos.

A partir dessa representação numérica, são determinados quais segmentos devem ser desligados para exibir o número correspondente no display de 7 segmentos. Para isso, são utilizadas portas OR para cada segmento (A, B, C, D, E, F, G), conforme segue:

- O segmento A é desligado para as combinações referentes aos números decimais 1 e 4;
- O segmento B é desligado para as combinações referentes aos números decimais 5 e 6, mas é desligado também caso o código de entrada não seja válido;
- O segmento C é desligado para as combinações referentes ao número decimal 2, mas é desligado também caso o código de entrada não seja válido;
- O segmento D é desligado para as combinações referentes aos números decimais 1, 4 e 7, mas é desligado também caso o código de entrada não seja válido;
- O segmento E é desligado para as combinações referentes aos números decimais 1, 3, 4, 5, 7 e 9;
- O segmento F é desligado para as combinações referentes aos números decimais 1, 2, 3 e 7;
- O segmento G é desligado para as combinações referentes aos números decimais 0, 1 e 7;
- Como os segmentos estão sempre ativos, para a exibição do dígito 8, não é desligado nenhum segmento portanto, não há uma porta OR definida para este dígito;
- A desativação dos segmentos B, C e D, exibe o caractere F que representa a inserção de um código 2 de 5 inválido.

Ao final, o módulo possui as saídas sA, sB, sC, sD, sE, sF e sG, que correspondem aos segmentos do display de 7 segmentos. As variações apresentadas estão dispostas na tabela 4.

Tabela 4. Tabela-verdade do decodificador de código 2 de 5 para display de 7 segmentos

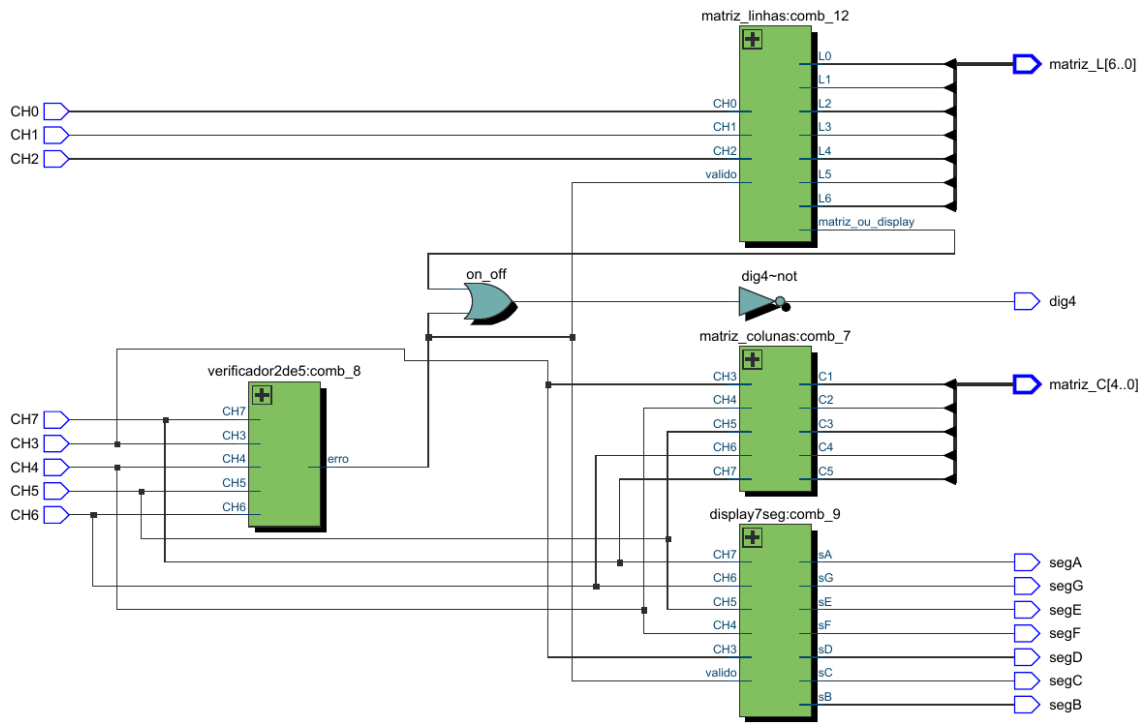
Decimal/Caractere representativo	Segmentos do display						
	a	b	c	d	e	f	g
0	0	0	0	0	0	0	1

1	1	0	0	1	1	1	1
2	0	0	1	0	0	1	0
3	0	0	0	0	1	1	0
4	1	0	0	1	1	0	0
5	0	1	0	0	1	0	0
6	0	1	0	0	0	0	0
7	0	0	0	1	1	1	1
8	0	0	0	0	0	0	0
9	0	0	0	0	1	0	0
F	0	1	1	1	0	0	0

Fonte: Elaborada pelos autores

3.4.6 Módulo Main

Figura 15. Módulo Main



Fonte: Representação gerada com o RTL Viewer

O módulo Main é responsável por controlar e interligar todos os módulos, ele é o principal módulo do protótipo.

As entradas do módulo são as chaves CH7, CH6, CH5, CH4 e CH3 que representam o código binário 2 de 5 inserido pelo usuário, e as chaves CH2, CH1 e CH0 que são utilizadas para selecionar a linha da matriz de LEDs que será acionada ou a exibição numérica do código inserido. Além disso, o módulo recebe as informações dos outros módulos utilizados no sistema: `matriz_colunas`, `verificador2de5`, `display7seg` e `matriz_linhas`.

As saídas do módulo são: as linhas da matriz de LEDs (`matriz_L`), as colunas da matriz de LEDs (`matriz_C`), os segmentos do display de 7 segmentos (`segA`, `segB`, `segC`, `segD`, `segE`, `segF`, `segG`), os dígitos do display de 7 segmentos (`dig1`, `dig2`, `dig3` e `dig4`) e o ponto (presente no display de 7 segmentos).

O módulo começa, através do comando “buf”, enviando sinal lógico 1 para o ponto e os dígitos não utilizados do display de 7 segmentos, deixando eles sempre desligados. Após isto, relaciona as chaves CH7, CH6, CH5, CH4 e CH3 com as colunas da matriz de LEDs utilizando o módulo `matriz_colunas`. Em seguida, verifica se o código binário 2 de 5 inserido é válido utilizando o módulo `verificador2de5` e converte o código para os segmentos do display de 7 segmentos utilizando o módulo `display7seg`.

Depois, é verificado se as chaves CH2, CH1 e CH0 estão na posição 000 (bit 1) através da saída “chave_000” do módulo `matriz_linhas` ou se o código binário 2 de 5 inserido é inválido através da saída “error” do módulo `verificador2de5`. Se pelo menos um desses fios estiverem com sinal lógico alto (1), ele é invertido e o dígito 4 do display de 7 segmentos permanece ligado e exibe ou a letra F (caso o código 2 de 5 inserido for inválido) ou o número referente ao código inserido (caso o código 2 de 5 inserido for válido). Caso estes dois fios estiverem com sinal lógico baixo (0), devido a inversão, o dígito 4 é desativado pois recebe sinal lógico alto (1).

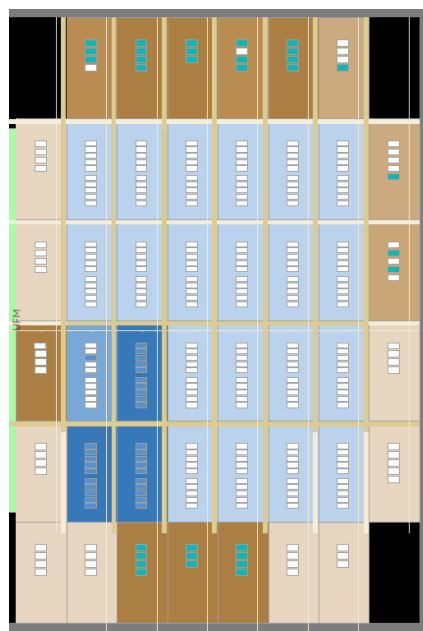
4. Resultados e discussões

O protótipo tem como objetivo principal ser um sistema de identificação e exibição de códigos de barras, o tipo escolhido foi o intercalado 2 de 5. 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.

4.1 Uso de LABs, LEs, LUTs e Pinos

No protótipo, foram constatados os usos de 4 dos 24 LABs presentes no kit de desenvolvimento, o que corresponde a aproximadamente 17%. A quantidade de uso de LEs (elementos lógicos) foi de 31 dos 240 disponíveis (cerca de 13%), onde 22 foram utilizados com LUTs de quatro entradas, 5 com LUTs de três entradas e 4 com LUTs de duas entradas. Em relação ao uso de pinos, foram utilizados 32 dos 80 disponíveis, o que corresponde a 40%. A figura 16 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 16. Imagem gerada pela ferramenta Chip planner



Fonte. Elaborada pelos autores

4.2 Manual de Uso

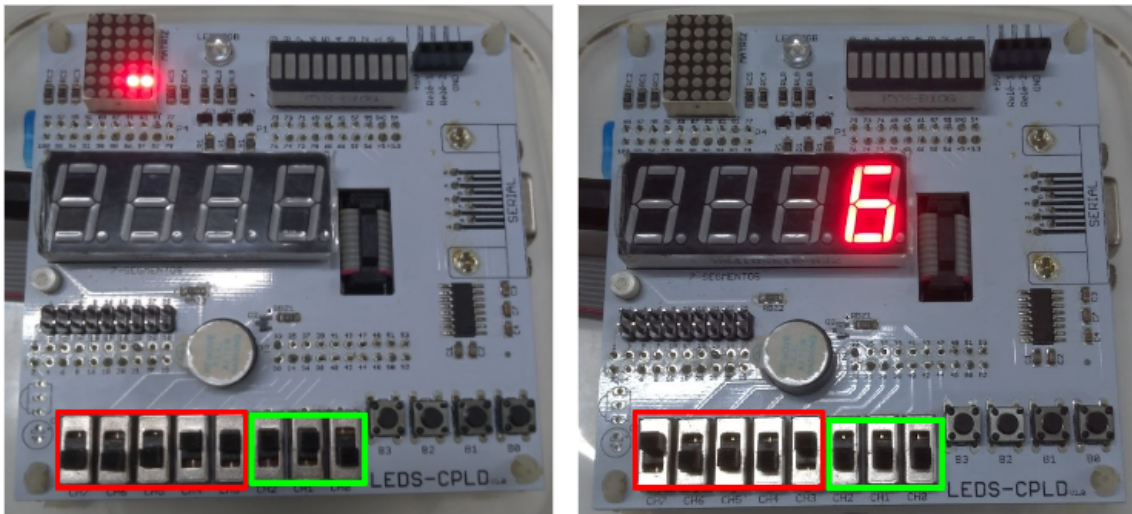
Apos o código verilog ser implementado no kit de desenvolvimento LEDS-CPLD, o usuário pode iniciar o uso do sistema. De modo a exemplificar, são apresentados dois exemplos.

Para que o usuário faça a entrada dos dígitos binários do código, deve-se posicionar as chaves CH7, CH6, CH5, CH4, e CH3 conforme a escolha desejada, onde, para cima é gerado um sinal de nível lógico alto (1) e para baixo, um sinal de nível lógico baixo (0). A tabela 3 apresenta as combinações e seus respectivos dígitos decimais resultantes.

Para que o usuário utilize qualquer uma das interfaces de saídas, deve-se posicionar as chaves CH2, CH1 e CH0, conforme a escolha desejada, seja escolhendo uma das linhas da matriz, seja escolhendo o display de segmentos. A tabela 4 apresenta as combinações e suas respectivas interfaces de saídas.

De modo a exemplificar, foram utilizadas as combinações referente a linha 6 (L5) da matriz de leds para e do dígito decimal 9 e, do display de 7 segmentos para exibir o número 4. as chaves referentes ao código de 5 bits estão destacadas em vermelho e as referentes a escolha da interface da saída, estão destacadas em verde.

Figura 17. Exibição do dígito 9 na matriz de leds e do dígito 6 no display de 7 segmentos.



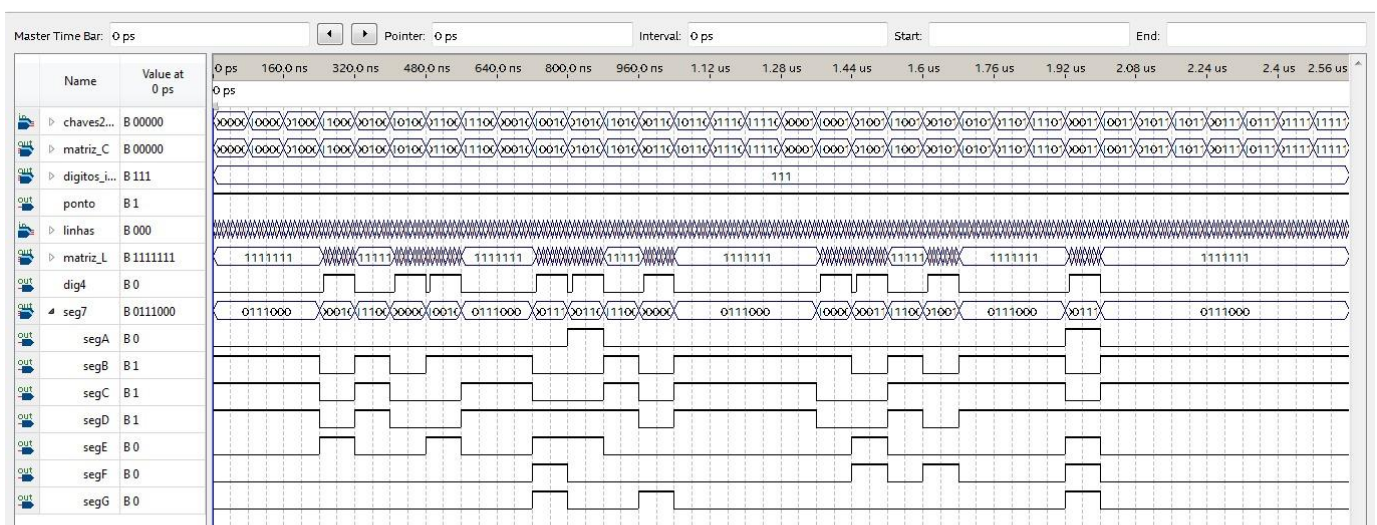
Fonte. Elaborada pelos autores

4.3 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 todos os nós (nodes) do circuitos fossem apresentados, em seguida foi definido o valor de 2560 ns para o fim da simulação com tempo de cada grade para 10ns. É definido 1280ns para o canal CH7, esse valor é dividido pela metade até chegar ao valor de 10ns em CH0. Na figura 18, é apresentada a simulação que possui o funcionamento como correto, pois os níveis lógicos enviados às interfaces de saída se alteram conforme as 32 variações de um código binário de 5 bits.

Figura 18. Simulação funcional do comportamento do circuito

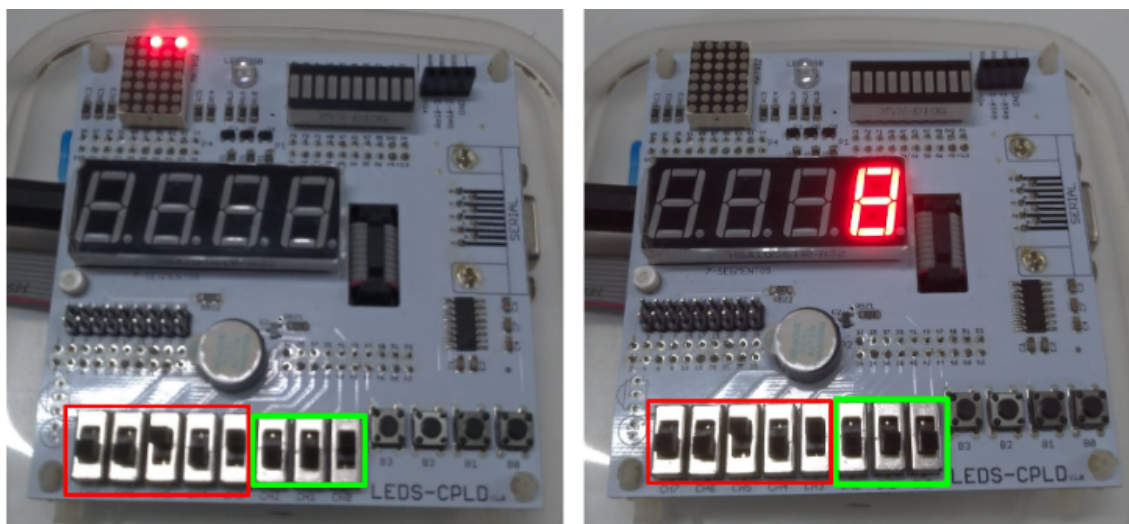


Fonte: Ferramenta Simulation Waveform Editor do software Quartus

Após a constatação do funcionamento em meio virtual, foram realizados testes com o código devidamente descarregado no kit de desenvolvimento. De modo a ilustrar os testes, dois são apresentados.

No primeiro teste, foram utilizadas as combinações referente a linha 1 (L0) da matriz de leds e do display de 7 segmentos. O dígito decimal escolhido para ser exibido foi o 8.

Figura 19. Exibição do dígito 8 no LEDS-CPLD.



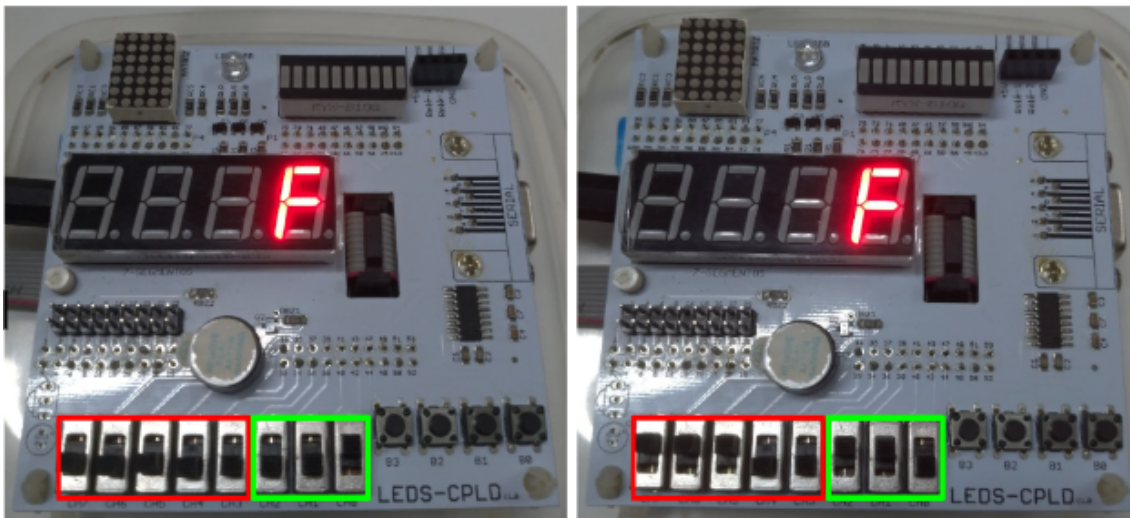
Fonte. Elaborada pelos autores

Note que as chaves destacadas no quadro vermelho, são as chaves responsáveis pela inserção dos dígitos binários referentes ao código intercalado 2 de 5. Neste caso, apenas as chaves CH5 e CH3 enviam sinal lógico alto para o sistema, gerando o código (00101) que representa o dígito 8.

As chaves destacadas em verde são responsáveis pela escolha da interface de saída, note que na imagem a esquerda, elas estão combinadas na forma (001) que é código responsável por ativar a linha 1 (L0) da matriz de leds; na imagem a direita, elas estão combinadas na forma (000) que é responsável por ativar o display de 7 segmentos.

No segundo teste, foram utilizadas as combinações referente à linha 1 (L0) e à linha 6 (L5) da matriz de led, mas os código de 5 bits utilizados foram os considerados inválidos para o sistema. Na parte esquerda da figura 20 é feita a escolha da linha 1 da matriz e o código de 5 bits inserido foi (00000); na parte direita, é feita a escolha da linha 6 da matriz e o código de 5 bits inserido foi (11100).

Figura 20. Inserção de código inválido no LEDS-CPLD.



Fonte. Elaborada pelos autores

Constata-se que mesmo que haja a escolha para que a saída do resultado seja feita na matriz de leds como, o código de 5 bits inseridos é inválido, a matriz de leds não é ativada e o caractere F que foi escolhido para representar códigos inválidos é exibido no display de 7 segmentos, o que atende corretamente um dos requisitos que foram solicitados.

5. Conclusão

O circuito atendeu todas as especificações solicitadas e apresentou excelente desempenho em relação às entradas do código e suas saídas. Os testes virtuais e reais indicaram o correto funcionamento do projeto, a matriz de LEDs acendeu as posições desejadas corretamente e o display foi capaz de exibir os dígitos decimais e o representado de erro corretamente conforme o código escolhido.

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/168zWlJU0rbnq3q8QJXnrwRY8iO6Ds2xQ/view>
Acesso em: 01 mar. 2023

KEYENCE. ITF e código de distribuição padrão. Disponível em: https://www.keyence.com.br/ss/products/auto_id/barcode_lecture/basic/itf/#:~:text=ITF%20%C3%A9%20a%20abreviatura%20de,em%20caixas%20de%20papel%C3%A3o%20corrugado. Acesso em: 02 mar. 2023

WIKIPEDIA. Two-out-of-five code. 2023. Disponível em: https://en.wikipedia.org/wiki/Two-out-of-five_code Acesso em: 02 mar. 2023

SEBRAEATENDE. Como funcionam os códigos de barras? . 2021. Disponível em: <https://www.sebraeatende.com.br/artigo/como-funcionam-os-codigos-de-barras> Acesso em: 02 mar. 2023

IFSC. AULA 8 - Eletrônica Digital 1 - Graduação, 2021. Disponível em: https://wiki.ifsc.edu.br/mediawiki/index.php/AULA_8_-_Eletr%C3%B4nica_Digital_1_-_Gradua%C3%A7%C3%A3o Acesso em: 09 mar. 2023

TECDICAS. Como descrever um circuito lógico algebricamente. 2019. Disponível em: <https://tecdicas.com/como-descrever-um-circuito-logico-algebricamente/> Acesso em: 09 mar. 2023

VIPPALA, Karthik. How to count number of one's in a binary vector using adders? . Youtube, 10 set. 2021. Disponível em: https://youtu.be/I77M2Dux_kg Acesso em: 09 mar. 2023