



UNIVERSIDADE FEDERAL DA FRONTEIRA SUL
BACHARELADO EM CIÊNCIAS DA COMPUTAÇÃO

LUCAS MAHLE, ROGÉRIO PINHEIRO E RAFAELLE ARRUDA

Codificador Braile

Chapecó 2017

Sumário

RESUMO3

1 OBJETIVOS E FUNDAMENTO4

2 METODOLOGIA5

- 2.1 Compreensão do circuito6
- 2.2 Tabela Verdade7 2.3 Mapa de Karnaugh9
- 2.4 BDD11

3 DESENVOLVIMENTO13

- 3.1 Desenvolvimento da Protoboard14
- 3.2 Desenvolvimento no Simulador15
- 3.3 Inserir Texto15
- 3.4 Salvar na memória16
- 3.5 Exibir no Display16

4 RESULTADOS/CONCLUSÃO19

REFERÊNCIA BIBLIOGRAFICA20

RESUMO

Este trabalho tem por objetivo desenvolver uma solução de baixo custo para auxiliar na alfabetização de pessoas com deficiência visual através da leitura táctil em Braille. Foi utilizado um sistema que desenvolve um circuito integrado para os resultados obtidos comprovaram a leitura fiel ao sistema Braille comprovando que o protótipo do circuito em Braille assiste aos deficientes visuais adequadamente. Para que isso funcione, ultimamos a tabela verdade e suas expressões, para que o mesmo mostre sua minimização do circuito, e inclusive foi usada o BDD para minimizar.

O sistema é disponibilizado por memoria, pois, o mesmo recebe os dados, e interpreta e envia os comandos para o circuito, usando um multiplexador e então processa o comando que foi dado, e mostra para o usuário.

1 OBJETIVOS E FUNDAMENTO

O objetivo do trabalho era criar um tradutor de palavras do alfabeto para a sintaxe Braille. Devido a falta de recursos, a letra em Braille foi formada de forma luminosa (LED) ao invés de forma tátil. Cada palavra traduzida poderia conter no máximo 8 letras.

O projeto foi dividido em duas partes. A primeira sendo desenvolvida num software simulador, enquanto a segunda foi desenvolvida usando componentes e uma protoboard.

O fundamento do projeto, era aplicar conceitos básicos para que conseguisse ser construído o projeto por inteiro. Baseando-se em conteúdo estudado previamente nas aulas, mas também exigir o estudo do aluno sobre, principalmente, os componentes disponíveis no software simulador.

2 METODOLOGIA

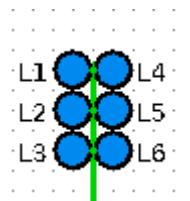
Com seu objetivo exposto sobre a leitura em Braille, devemos ter conhecimento da deficiência visual, é necessário que o mesmo domine alguns materiais básicos como instrumentos e movimentos, exemplo o tato para pegar e sentir objetos ao redor. Entretanto buscamos desenvolver uma estrutura de circuito mecânica para que fosse capaz de formar caracteres em letra para led em alto relevo. As letras são formadas por meio de diferentes combinações na led, mas quando uma letra é identificada, dependendo de qual linha está e se encontra a led, fica acesa ou apagada, pois o mesmo tem cinco entradas e seis leds de saída respectivamente, mas alguns deles irá acender de acordo com que o usuário utilizar na escrita.

Para prosseguir no projeto utilizamos a protoboard para montar o circuito, pois com ela, desenvolvemos dezenas de circuito sem a necessidade soldar, mas se o circuito está com problema, apenas retiramos um simples fio. Entretanto para o braille funcionar, aderimos resistor de 1K e CII, ocorre da seguinte forma, cada pino da led funciona com o auxílio de um dispositivo para alimentar o circuito, mas para acionar um pino, é necessário enviar uma frase ou texto, fazendo com que gere, ressaltando, uma característica do trabalho é que o seu funcionamento necessita de uma corrente elétrica elevada a 5v, mais que isso não funciona.

2.1 Compreensão do circuito

A primeira e principal etapa (após compreender o objetivo do trabalho) era formar uma tabela verdade a partir dos possíveis resultados. Para que isso fosse possível, o foco primário era apena a expressão para acender os leds.

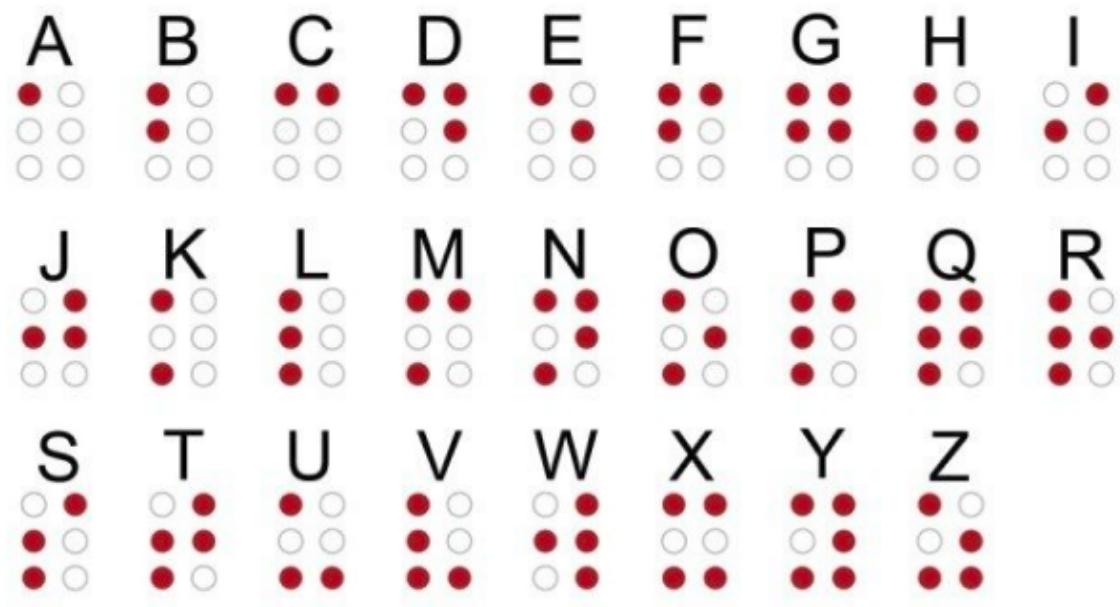
Foi determinado que a ordem dos leds para formação da letra em braille era da seguinte maneira:



A próxima etapa, teve um fator influenciador grande. Levando em consideração que o valor em base binária de uma letra na tabela ASCII possui 8 bits, seria inviável formar a expressão do circuito usando papel e caneta. O que foi averiguado nesse momento, foi que os 3 primeiros bits (da esquerda para direita) na extensão de “a” até “z” sempre eram formados com 011, logo, foi possível diminuir as entradas para apenas 5 bits.

2.2 Tabela Verdade

Comparando com o alfabeto braille, foi montado a tabela verdade, usando a entrada como o valor da tabela ASCII e na saída os leds acesos de acordo com a letra braille:



	A	B	C	D	F	L_0	L_1	L_2	L_3	L_4	L_5
*	0	0	0	0	0	-	-	-	-	-	-
A	0	0	0	0	1	0	0	0	0	0	0
B	0	0	0	1	0	1	0	0	0	0	0
C	0	0	0	1	1	0	0	1	0	0	0
D	0	0	1	0	0	1	0	0	1	0	0
E	0	0	1	0	1	0	0	0	0	1	0
F	0	0	1	1	0	1	0	1	0	0	0
G	0	0	1	1	1	1	0	1	1	0	0
H	0	1	0	0	0	1	1	0	0	1	0
I	0	1	0	0	1	0	1	0	0	0	0
J	0	1	0	1	0	1	0	1	1	0	0
K	0	1	0	1	1	0	1	0	0	0	0
L	0	1	1	0	0	1	1	0	0	0	0
M	0	1	1	0	1	0	1	1	1	0	0
N	0	1	1	1	0	0	1	1	1	1	0
O	0	1	1	1	1	0	1	0	0	1	0
P	1	0	0	0	0	1	1	1	1	0	0
Q	1	0	0	0	1	1	1	1	1	1	0
R	1	0	0	1	0	1	1	0	1	0	0
S	1	0	0	1	1	0	1	1	1	0	0
T	1	0	1	0	0	1	1	1	1	0	0
U	1	0	1	0	1	0	1	0	0	0	1
V	1	0	1	1	0	1	1	1	0	0	1
W	1	0	1	1	1	0	1	0	1	1	1
X	1	1	0	0	0	1	0	1	1	0	1
Y	1	1	0	0	1	0	1	1	1	1	1
Z	1	1	0	1	0	1	0	1	0	1	1

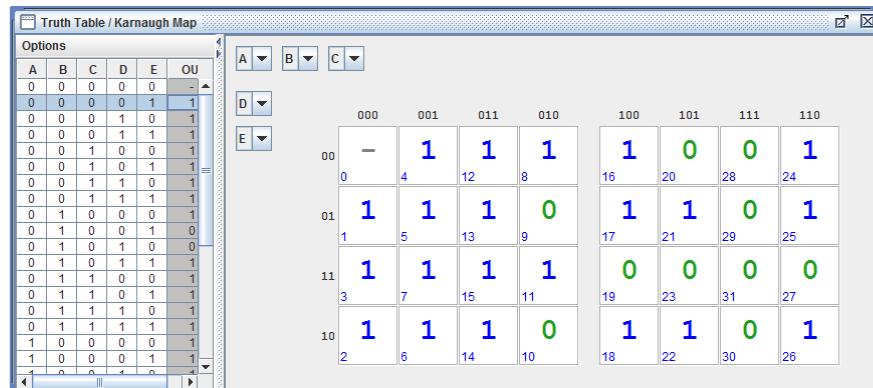
O resultado é uma tabela verdade formada por 27 linhas. A primeira linha, com as 5 entradas zeradas, foi incluída para garantir que o circuito apagasse os leds quando o barramento possuía nível lógico 0.

Das inúmeras maneiras de se extrair uma expressão baseando-se na tabela verdade, foi optado por usar o mapa de karnaugh, já que com 5 variáveis não é tão complexo do que seria com 8 variáveis.

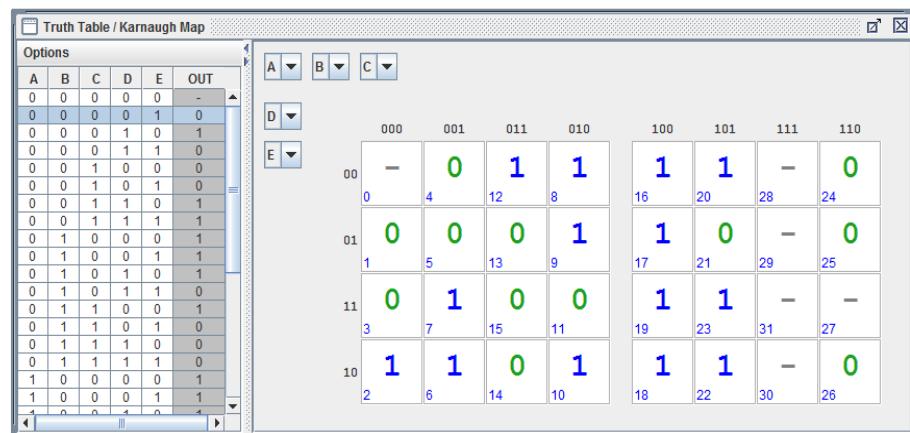
2.3 Mapa de Karnaugh

O mapa foi gerado para todos os 6 leds, lembrando que o resultado já estava na menor expressão, não sendo necessário aplicar nenhuma técnica de minimização de circuito.

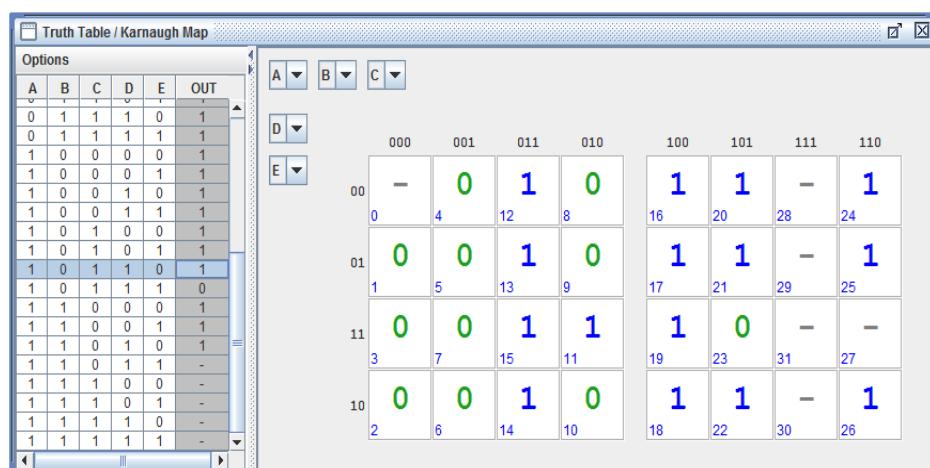
L0



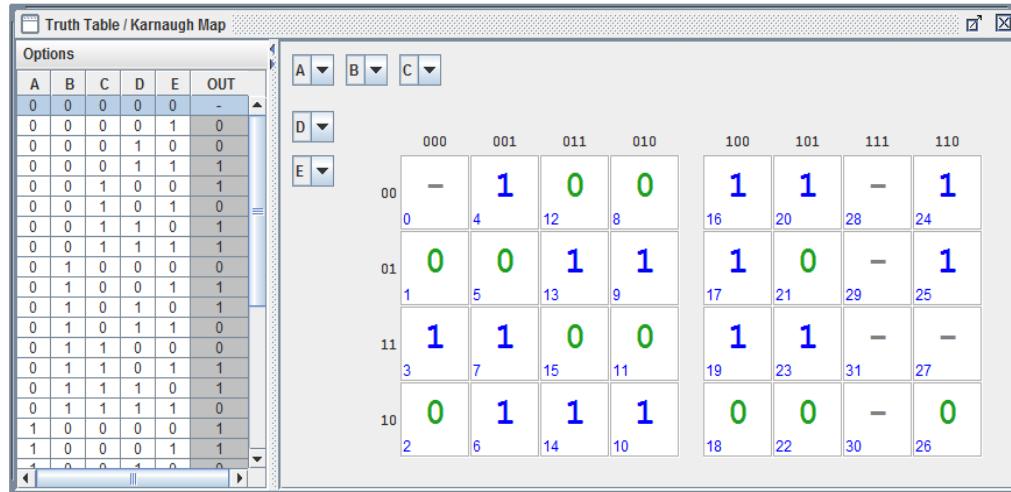
L1



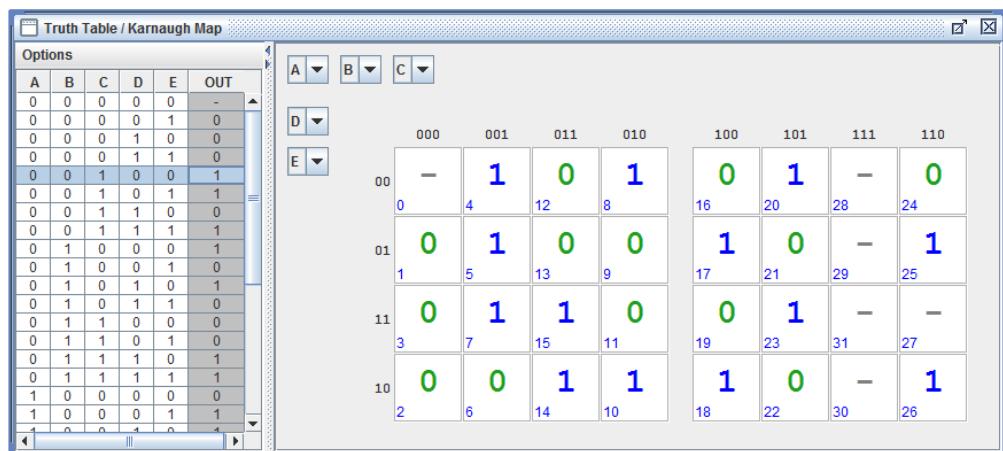
L2



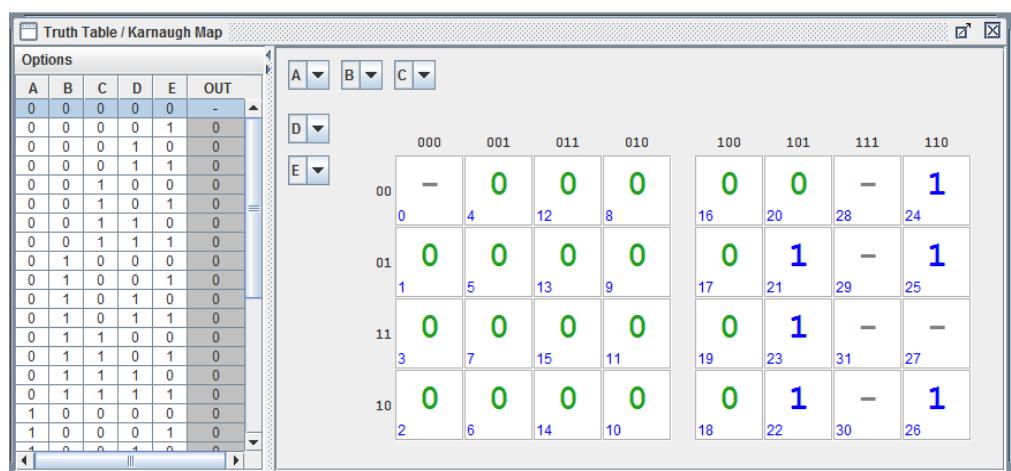
L3



L4



L5

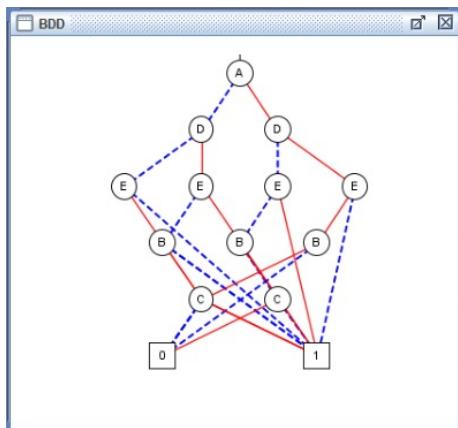


O mapa foi gerado para todos os 6 leds, lembrando que o resultado já estava na menor expressão, não sendo necessário aplicar nenhuma técnica de minimização de circuito.

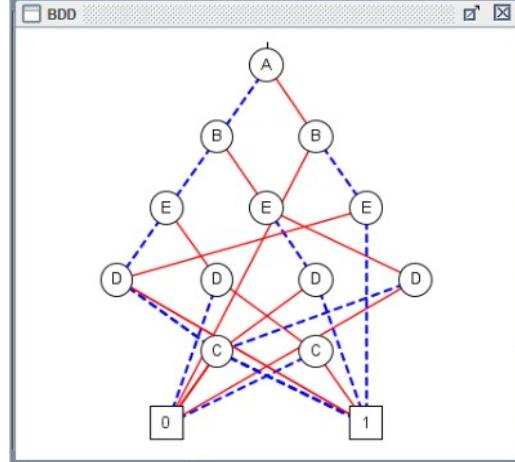
2.4 BDD

O BDD é uma técnica de desenvolvimento ágil que encoraja colaboração entre desenvolvedores, setores de qualidade e pessoas não-técnicas ou de negócios num projeto de circuito antes de simular o projeto, é tática de minimização. Entretanto utilizamos ela, para minimizar a contagem de 0(zero) a 1(um) num digrama com todas as entradas e saídas do mesmo. De acordo com a imagem:

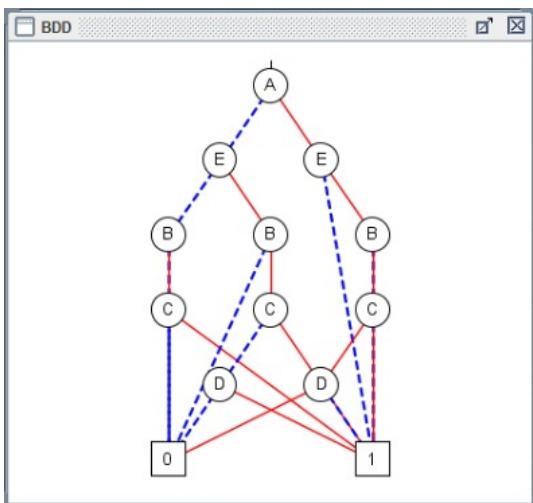
L0



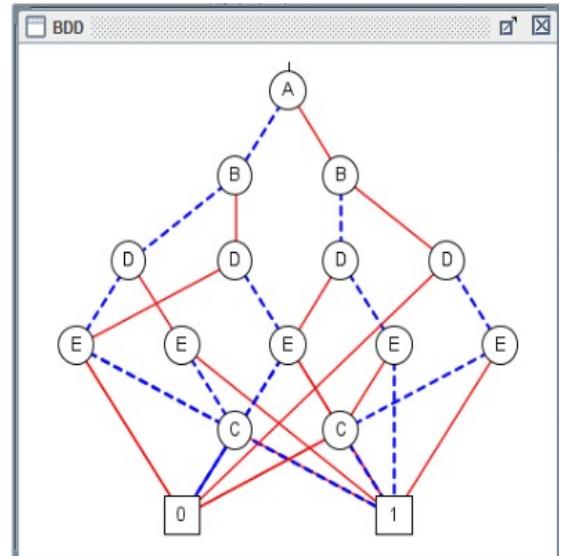
L1



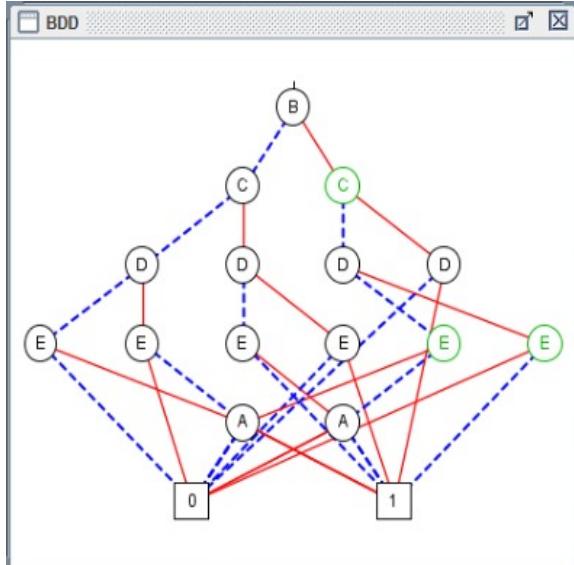
L2



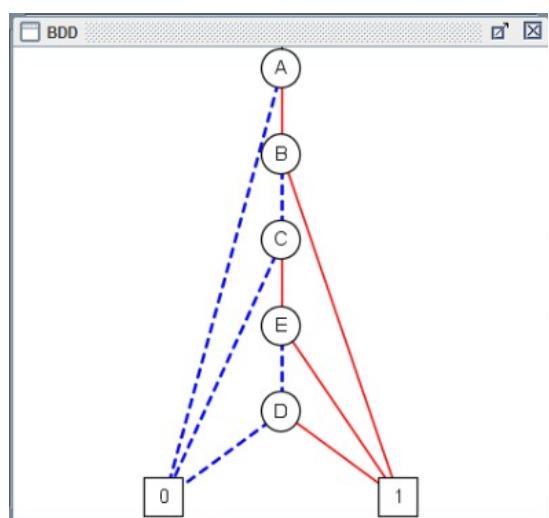
L3



L4



L5



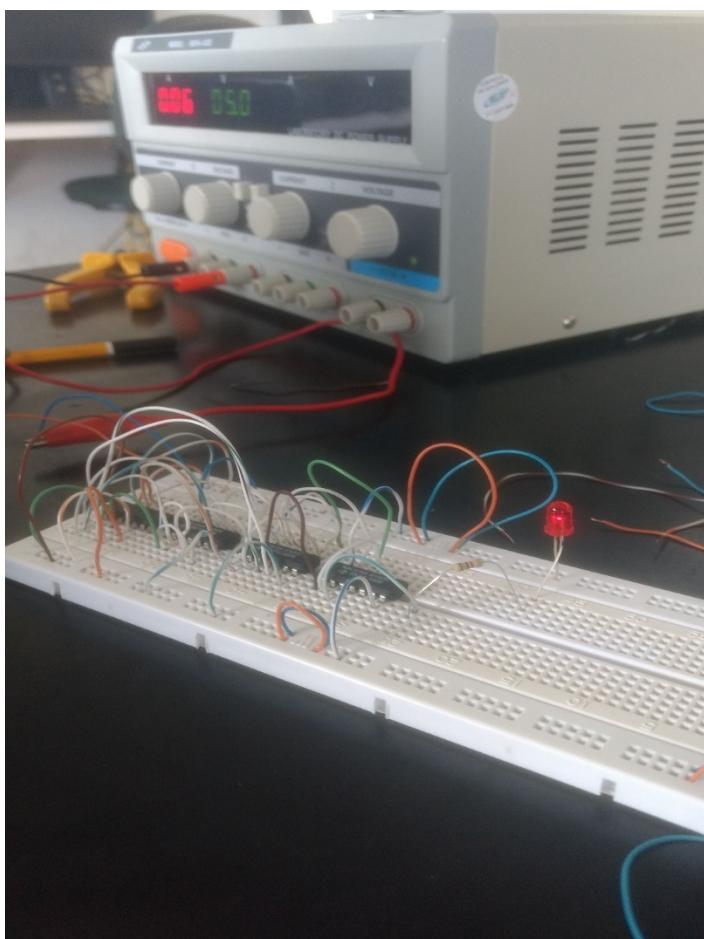
3 DESENVOLVIMENTO

O desenvolvimento do projeto tinha 2 objetivos, a protoboard com expressão de um único led e o circuito completo no software simulador.

3.1 Desenvolvendo a protoboard

O objetivo principal da protoboard foi criar um circuito para verificar o seu funcionamento do Led. Entretanto foi utilizado CI para realizar as ações mais complexas que não podem ser executadas por um único componente. Um circuito integrado pode realizar diferentes funções, como temporizador, oscilador, amplificador, controlador e outras. Mas no projeto o CI foi essencial para conversão em braille, nela usamos apenas o conversor.

Para demonstrar o circuito, escolhemos o Led L5 para verificar seu funcionamento, e demonstrar o trabalho concluído no laboratório. Apenas usamos os fios para representar a entrada, e saída em caracteres (exemplo A, B, e C são as entradas e saída representamos como L0, L1..), mas por fim, utilizamos o Led vermelho, claramente tivemos que fazer teste até ver se realmente está funcionando e analisando a sua alimentação na protoboard. Assim como demonstra a imagem a baixo:



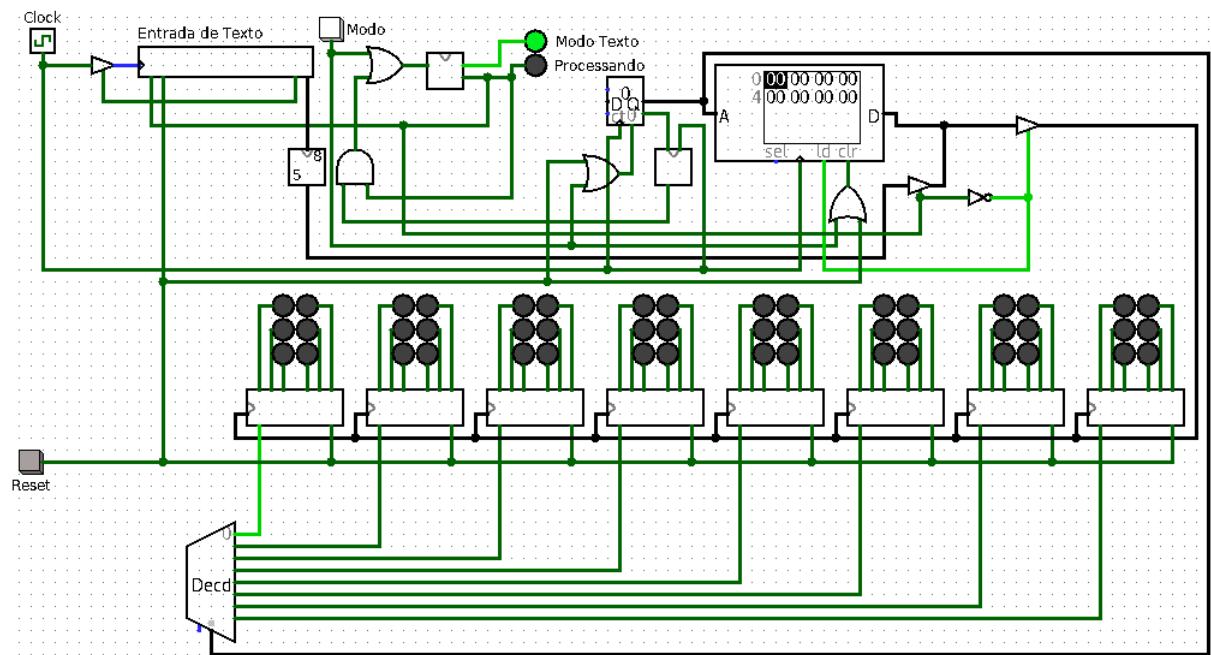
3.2 Desenvolvimento no simulador

O projeto no simulador teve um alto nível de exigência, devido ao fato da falta de experiência dos 3 integrantes no manuseio do mesmo. Com isso, tivemos gastos de muitas horas de pesquisas e várias latas de bebidas energéticas.

O fluxo do projeto foi definido da seguinte maneira:



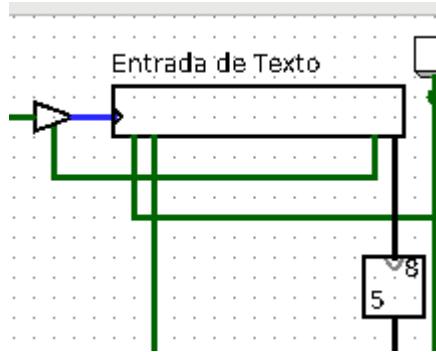
O dois principais componentes de sincronização é *clock* que faz o contador funcionar. O contador foi limitado a 3 bits, então seu valor vai de 0 até 7, o que é suficiente para o projeto, já que possui apenas 8 displays. O contador se conecta à memória RAM e o Decodificador.



3.3 Inserir Texto

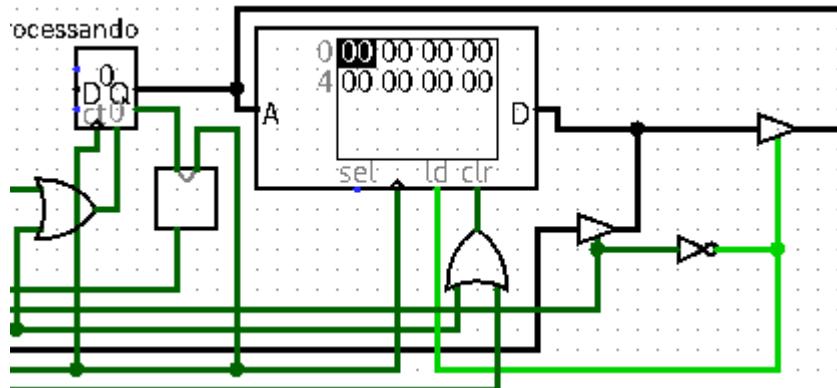
O bloco mais simples do projeto, composto por um teclado para inserir a palavra e um conversor de barramento, de 8 bits para 5.

A saída do conversor está ligado na ponte chaveadora de barramento conectado à memória RAM.



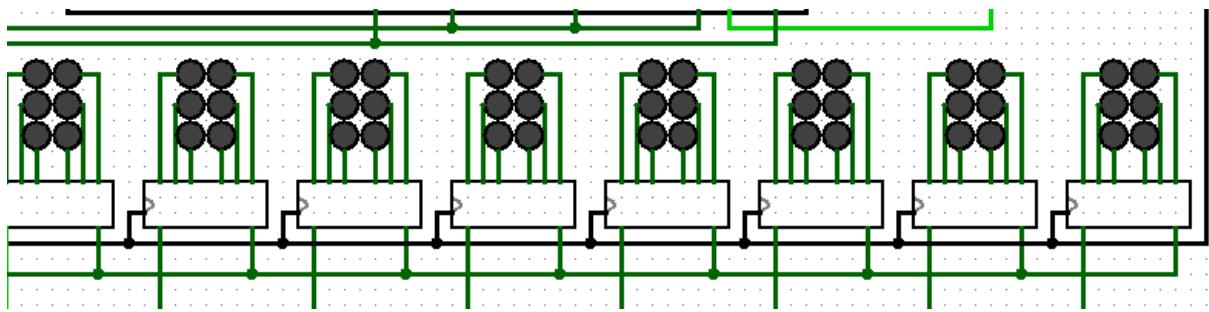
3.4 salvar na memória

Após o texto ser escrito na entrada de dados, é pressionado um botão que altera o estado de um registrador e desencadeia o processo de gravação. O botão está ligado no *clear* da memória, sendo assim, a memória é limpa, o load da memória é desativado e o buffer controlado habilita a ligação entre a caixa de texto e a entrada de dados da memória RAM. A partir daí, a cada clock, o contador é incrementado, alterando assim o endereço apontado na memória, do mesmo modo, a cada clock o teclado seta no barramento o valor da letra e limpa a anterior. Os 7 primeiros clocks fazem a mesma função, mas no 8º clock faz com que o contador sete o Caray como 1, ativando o componente de *Jump Clock*, que consiste em guardar o valor 1 e só permitir sua passagem no próximo clock. Isso foi necessário para conseguir alterar o modo de funcionamento, pois o ciclo de gravação só pode ser executado 1 vez (8 contagens). O modo texto então é reativado. Sem o *Jump Clock*, a última letra não seria gravada.

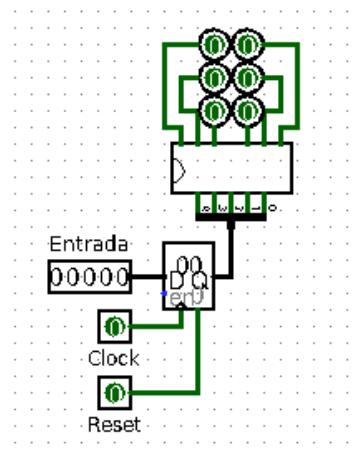


3.5 Exibir no Display

O modo texto é o modo na qual permite a escrita no teclado e ao mesmo tempo é o modo que escreve os valores da memória nos displays. Após o 8º ciclo do contador voltar para o modo texto, o chaveamento de barramento da memória é alterado, fazendo com que a saída da memória se conecte com os displays, o load da memória RAM é ativo, por consequência, o barramento recebe o valor que está sendo apontado pelo endereço da memória. O contador tem sua saída ligada ao decodificador, assim é sincronizado o endereço da memória e a saída do decodificador que é ligado em cada um dos 8 displays.



Quando o valor do decodificador é alterado, uma das 8 portas recebe o valor 1, na qual está conectado ao registrador do display. O registrador estando ativo, grava o valor que está no barramento, logo, o valor que está na memória, o valor da letra. A saída desse registrador, está conectado ao circuito dos leds, fazendo com que os leds se acendam.



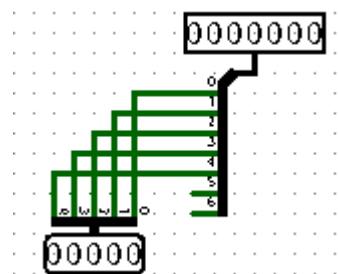
Chave do modo de operação:

Grava o seu próprio valor negado toda vez que o botão é clicado.



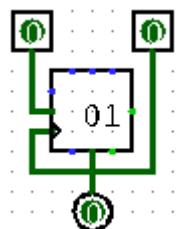
Conversor de dados 8x5:

Distribui os 8 bits recebidos em 5 bits. Dispensando os 2 últimos bits.



Jump Clock:

Usa um deslocador para pular um pulso do clock.



4 RESULTADOS/CONCLUSÃO

Na realização de teste, foram utilizadas aproximadamente 26 letras do alfabeto, com o Braille de seis pinos. Por meio da execução do sistema pôde-se verificar que o Codificador Braille desempenho as funções esperadas, que são: Leitura de informação textuais a partir de uma memória, conversão em sinais analógicos e apresentação das informações, desenvolvida em forma de caracteres. Entretanto com a necessidade de melhoria de vida tanto para pessoas sem necessidade especiais e tanto para aquelas têm deficiência, a tecnologia torna-se alternativa. Ressaltando que esse trabalho buscou prover acessibilidade a pessoas com deficiência visual por meio do desenvolvimento de um sistema que provê a apresentação de textos digitais em alfabeto Braille.

REFERÊNCIAS BIBLIOGRÁFICAS

DESENVOLVIMENTO DE UMA CÉLULA BRAILLE DE BAIXO CUSTO USANDO ARDUINO, **LIMA**, MAURÍCIO. Disponível no site web
<http://www.swge.inf.br/cba2014/anais/PDF/1569935361.pdf>

Disponível no site

http://www.contagem.pucminas.br/pitane/index.php?option=com_content&view=article&id=87:a

Logisim part 4:Multiple inputs and Registers

Disponível no site: <https://www.youtube.com/watch?v=BuPwuVrmCUk>

Logisim part 6:Keyboards

Disponível no site: https://www.youtube.com/watch?v=fWx8x9hk_RI

Logisim part 7:ROM

Disponível no site :<https://www.youtube.com/watch?v=ejpdxbkHgWo>

Logisim part 10:RAM

Disponível no site :<https://www.youtube.com/watch?v=lk5iSGwzmgY>