



**UNIVERSIDADE FEDERAL DA FRONTEIRA SUL
CAMPUS DE CHAPECÓ
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**IGOR ABEL DOS SANTOS BORDIN
MATHEUS SLAMA RIBAS
PABLO RICARDO LODI DE LIMA**

CONVERSOR BRAILLE

**CHAPECÓ
2018**

SUMÁRIO

1. OBJETIVOS.....	3
2. METODOLOGIA.....	4
3. DESENVOLVIMENTO.....	5
3.1. Conversor de Letras para Código ASCII.....	5
3.2. Alfabeto Braille.....	6
3.3. Teste de Mesa.....	6
3.4. Mapa de Karnaugh.....	7
3.5. Explicação do circuito.....	7
4. CONCLUSÃO.....	10
REFERÊNCIA BIBLIOGRÁFICA.....	11

1. OBJETIVOS

O objetivo do trabalho foi criar um tradutor de palavras do alfabeto latino 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 32 letras.

O projeto foi dividido em três partes. A primeira sendo desenvolvida num programa em Python, enquanto a segunda e a terceira foram desenvolvidas usando um software simulador de circuitos integrados (LogiSim).

O fundamento do projeto era aplicar conceitos básicos para que fosse possível a construção do projeto por inteiro. Embasando-se no conteúdo estudado visto anteriormente e também exigindo do aluno o estudo sobre, principalmente, os componentes disponíveis no software simulador.

2. METODOLOGIA

Neste trabalho foi utilizado um sistema que desenvolve um circuito lógico utilizando o simulador LogiSim para a geração de um sistema Braille .

Foi utilizado um programa desenvolvido em Python que converte letras do alfabeto para os códigos da tabela ASCII, o qual sua saída é inserida em uma memória usada para interpretar e enviar comandos para o circuito, que usa um conversor, um registrador, um decodificador e entre outros componentes, processando o comando que foi dado e mostrando ao usuário um sistema Braille com a utilização de LEDs.

3. DESENVOLVIMENTO

O projeto foi dividido em três partes:

1. Um programa na linguagem Python que converte palavras para a tabela ASCII.

(A frase convertida deve conter no máximo 32 caracteres.)

2. O conversor de ASCII para LEDs (Braille).
3. O conversor de LEDs (Braille) para palavras.

3.1. Conversor de Letras para Código ASCII

O código desenvolvido na linguagem Python (versão 3.6.5) armazena um texto inserido pelo usuário usando o terminal dentro de uma variável. Criando um arquivo de imagem de memória ROM no disco compatível com o LogiSim, o programa converte usando um laço de repetição cada caractere encontrado no que foi digitado para o seu respectivo em ASCII.

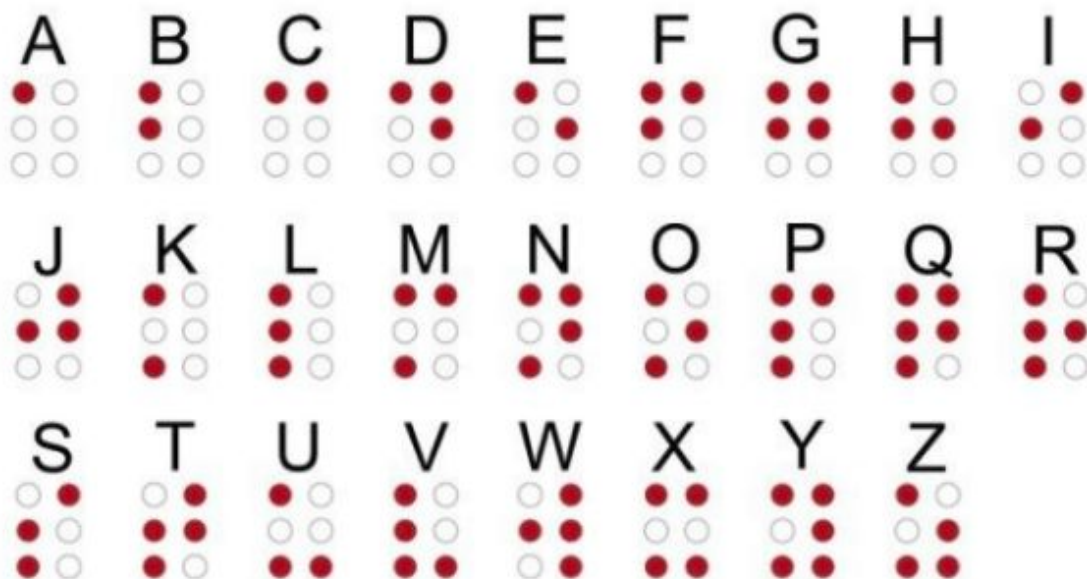
Por último, ele retorna a conversão para o arquivo no disco e imprime na tela do terminal o seu conteúdo final.

```
print("Digite o texto a ser convertido para ASCII (sem acentos):"); x = input(); cont = 0; dat=open('texto','w')
texto= ""; texto+= "v2.0 raw\n"
for a in x:
    b = list(str(hex(ord(a.lower()))[2:])); c = str(int(b.pop(0)) - 2); b.insert(0,c); d = "".join(b); texto+=d;
    cont+=1;
    if cont < 8: texto+=" ";
    else: texto+="\n"; cont=0;
dat.write(texto); dat.close();
print(texto);
```

Código fonte do software criado

3.2. Alfabeto Braille

Comparando com o alfabeto Braille, a tabela verdade foi montada usando a entrada dos valores da tabela ASCII para que na saída, os LEDs sejam acesos de acordo com a determinada letra em Braille:



3.3. Teste de Mesa

O resultado é uma tabela verdade formada por 27 linhas. Foram adicionadas todas as possibilidades (incluindo as possibilidades que tem resultado 0).

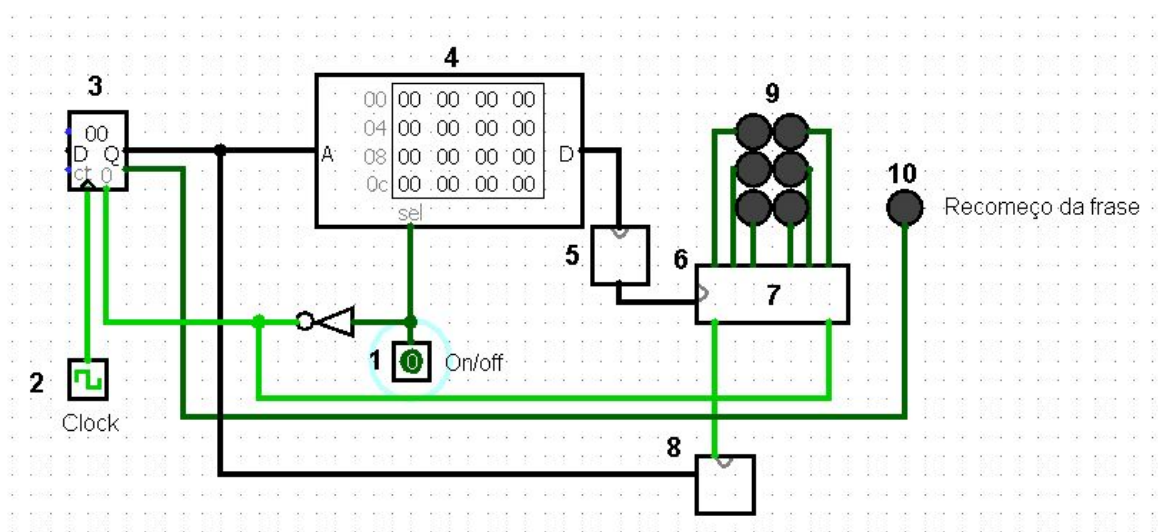
Como existem várias maneiras de gerar uma expressão usando a tabela verdade, optou-se por usar o Mapa de Karnaugh com 5 variáveis, pois sua aplicação é mais simples do que uma de 8 variáveis.

	A	B	C	D	F	L0	L1	L2	L3	L4	L5
.	0	0	0	0	0	-	-	-	-	-	-
A	0	0	0	0	1	1	0	0	0	0	0
B	0	0	0	1	0	1	1	0	0	0	0
C	0	0	0	1	1	1	0	0	1	0	0
D	0	0	1	0	0	1	0	0	1	1	0
E	0	0	1	0	1	1	0	0	0	1	0
F	0	0	1	1	0	1	1	0	1	0	0
G	0	0	1	1	1	1	1	0	1	1	0
H	0	1	0	0	0	1	1	0	0	1	0
I	0	1	0	0	1	0	1	0	1	0	0
J	0	1	0	1	0	0	1	0	1	1	0
K	0	1	0	1	1	1	0	1	0	0	0
L	0	1	1	0	0	1	1	1	0	0	0
M	0	1	1	0	1	1	0	1	1	0	0
N	0	1	1	1	0	1	0	1	1	1	0
O	0	1	1	1	1	1	0	1	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	1	0	1	0
S	1	0	0	1	1	0	1	1	1	0	0
T	1	0	1	0	0	0	1	1	1	1	0
U	1	0	1	0	1	1	0	1	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	1	0	1	1	1	1
Z	1	1	0	1	0	1	0	1	0	1	1

3.4. Mapa de Karnaugh

Foi utilizado o mapa de karnaugh do próprio logisim, que entregou o circuito pronto utilizando as opções de “usar somente porta NAND” e “usar apenas portas com duas entradas” fizemos o circuito Letras/LED. Utilizando 5 variáveis já que $2^5 = 32$, só precisamos de 23 caracteres que corresponde a letras do alfabeto criamos esse circuito

3.5. Explicação do circuito



1-On/Off. Botão que liga e desliga o circuito e limpa o contador (3) e os leds (9).

2-Clock. O pulso de clock que vai aumentando em 1 o valor do contador (3).

3-Contador. O valor do contador reflete na posição da memória ROM (4) e quando chega ao final da memória ROM (4) dá um pulso para o led de Recomeço de frase (10) e reinicia a contagem a partir do 00, a primeira casa da memória ROM (4).

4-memória ROM. Armazena o código da tabela ASK e manda o sinal referente a ele para o conversor (5) e para o codificador de 32 letras (8).

5-conversor 8X5. A memória ROM (4) tem o pulso de saída de 8 bits então o conversor transforma em 5 já que só precisamos de 5 para ligar todos os LEDs (9).

6-Registrador de letra. Recebe o pulso de 5 bits e passa por um contador e irá para o letras/LED (7).

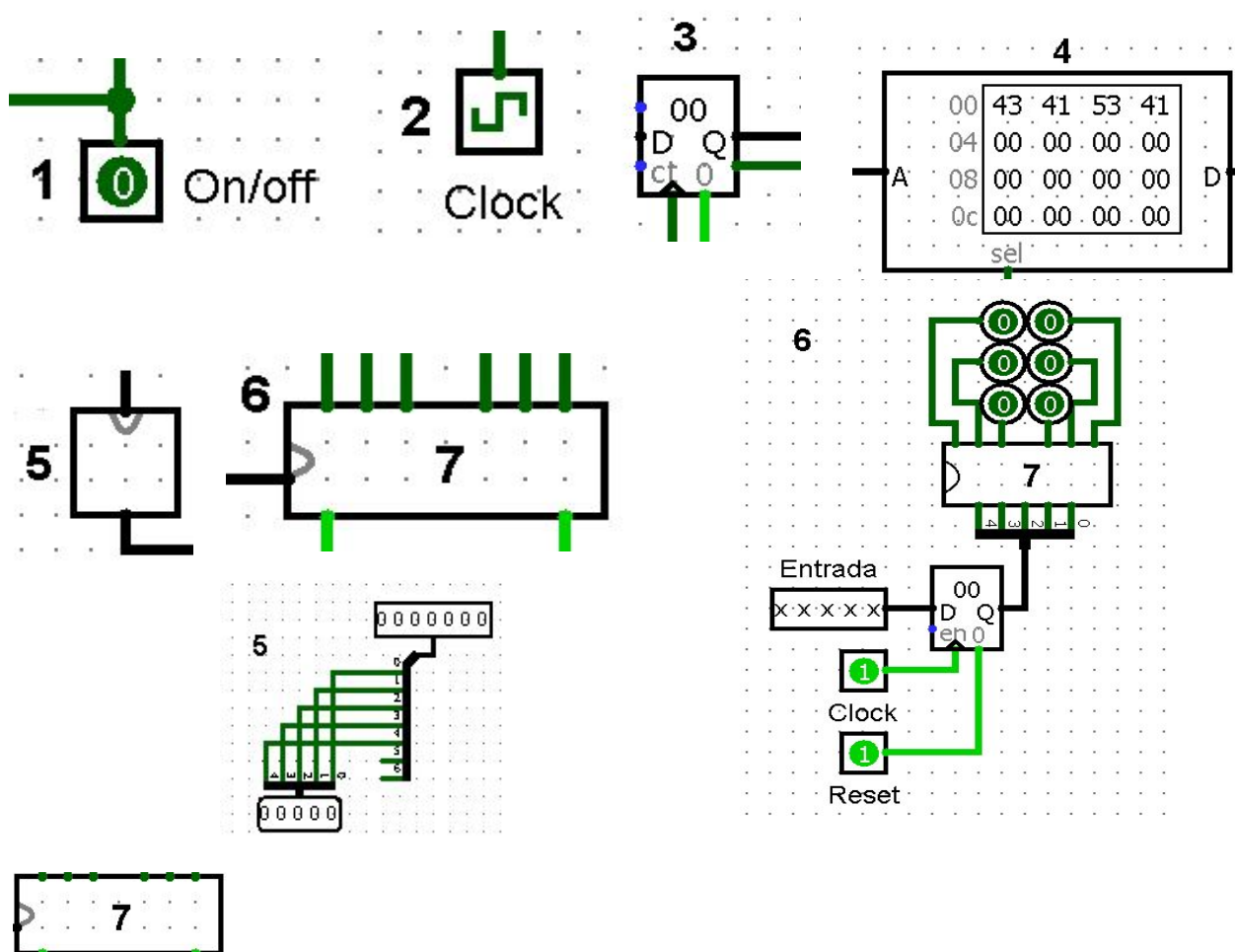
7-Letras/LED. Recebe os valores que vão ser ligados de acordo com a tabela verdade e o mapa de Karnaugh e vai selecionar quais os LEDs (9) que deve acender.

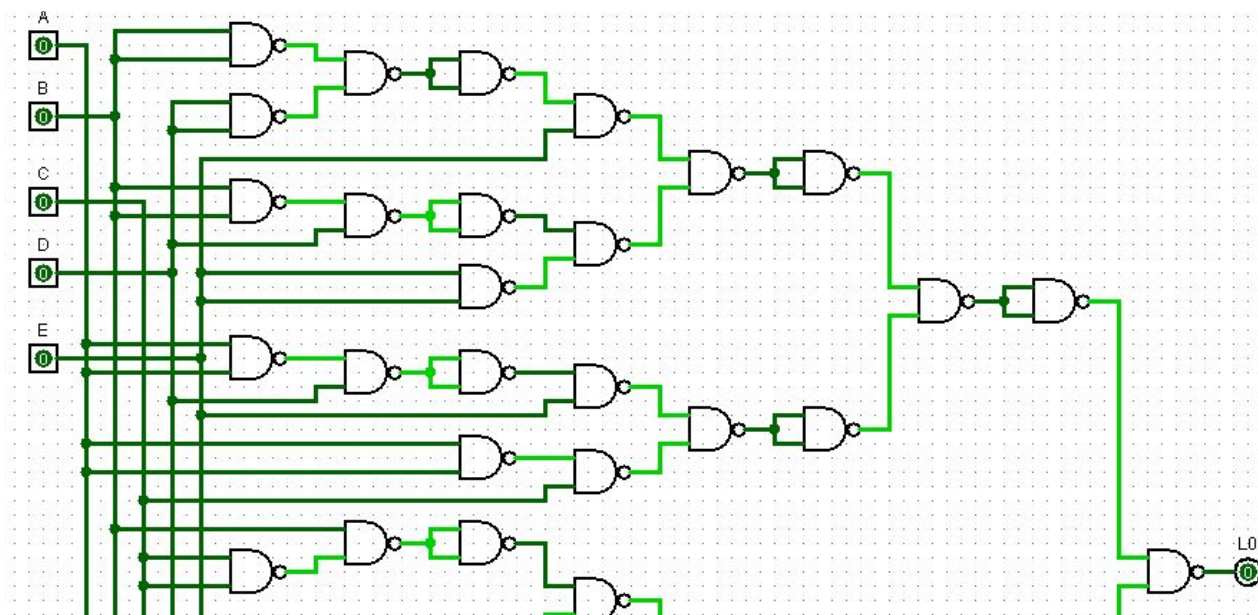
8-Codificador de 32 Letras. Os dados chegam nele onde passam para um decodificador que vai mandar o sinal em ordem e irão para uma porta OR que só é para mandar um sinal por vez.

9-Conjunto de LEDs. Ele receberá o sinal para ligar do Letras/LED (7) e ligar somente os que estiverem de acordo com a letra em braille.

10-Recomeço de frase. Toda vez que a memória ROM (4) chegar ao final irá emitir um pulso para esse LED mostrando que a frase irá recomeçar.

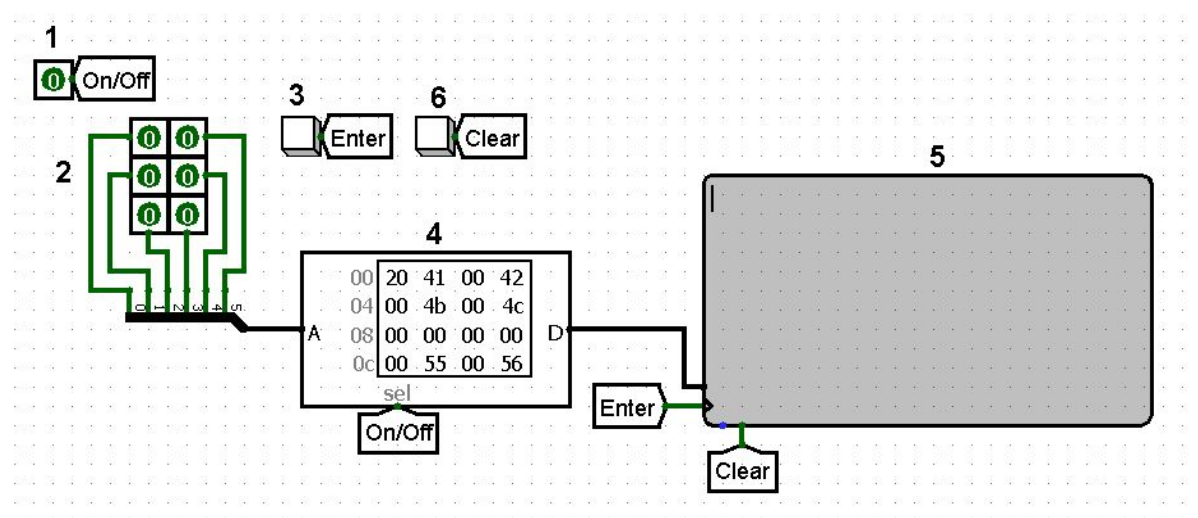
Imagens dos circuitos:





Parte do circuito de Letras/LED (7)

Circuito Braille para Letra:



1-On/Off.Liga a memória ROM (4) para que o circuito funcione.

2-Braille.Leds que representam os caracteres bailes que serão convertidos em letras.

3-Enter.Após escrever no braille(2) aperta o enter para enviar a posição do caractere para a memória ROM (4).

4-Memória ROM. onde estão gravados os dados do caracteres em ascii e a encia para a TTY (5).

5-TTY.Tela que recebe o valor em ascii da ROM (4) e mostra em letra.

6-Clear.Limpa a TTY(5) apagando tudo que foi escrito.

4. CONCLUSÃO

Na execução dos testes, foi utilizado o alfabeto com suas 26 letras, sendo elas representadas em Braille de seis pontos. Por meio da execução do sistema pôde-se verificar que o Codificador Braille desempenhou as funções esperadas, que são: Leitura de informação textuais a partir de uma memória, conversão em sinais analógicos, leitura dos sinais analógicos (Braille) armazenando-os em uma memória, apresentação das informações, desenvolvida em forma de caracteres.

REFERÊNCIAS BIBLIOGRÁFICAS

Documentação do logisim. Disponível em:
<<http://www.cburch.com/logisim/pt/docs.html>>. Acesso em 22 set. 2018.

Braille Virtual. Para aprender Braille na Internet. Disponível em:
<<http://www.braillevirtual.fe.usp.br/pt/index.html>> Acesso em: 25 set. 2018.