ELEC 204 - Final Lab Assignment

Ahmet Akkoç

A Morse Code Generator

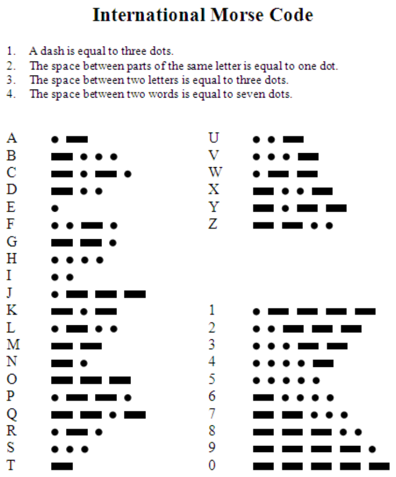
Chapter 1

Introduction

1.1 Goals

The main goal of this lab project is to design a simple device which can encode the English alphabet and Hindu-Arabic numerals as Morse Code.

The device will have a small memory allowing for a number of characters to be typed before playing out a given message. The user will utilize the switches at the bottom of the board to type in characters and the buttons to the right side will be used for functions such as adding new characters or displaying the currently encoded message. The 10 LED's on the bottom of the board will be used to display the message, which will appear as a signal travelling from right to left, indicated by the LED's switching on and off.



International Morse Code

1. A dash is equal to three dots.
2. The space between parts of the same letter is equal to one dot.
3. The space between two letters is equal to three dots.
4. The space between two words is equal to seven dots.

A	● ▬	U	● ● ▬
B	▬ ● ● ●	V	● ● ● ▬
C	▬ ● ▬ ●	W	● ▬ ▬
D	▬ ● ●	X	▬ ● ● ▬
E	●	Y	▬ ● ▬ ▬
F	● ● ▬ ●	Z	▬ ▬ ● ●
G	▬ ▬ ●		
H	● ● ● ●		
I	● ●		
J	● ▬ ▬ ▬		
K	▬ ● ▬	1	● ▬ ▬ ▬ ▬
L	● ▬ ● ●	2	● ● ▬ ▬ ▬
M	▬ ▬	3	● ● ● ▬ ▬
N	▬ ●	4	● ● ● ● ▬
O	▬ ▬ ▬	5	● ● ● ● ●
P	● ▬ ▬ ●	6	▬ ● ● ● ●
Q	▬ ▬ ● ▬	7	▬ ▬ ● ● ●
R	● ▬ ●	8	▬ ▬ ▬ ● ●
S	● ● ●	9	▬ ▬ ▬ ▬ ●
T	▬	0	▬ ▬ ▬ ▬ ▬

Figure 1.1: Morse Code Chart

1.2 Approach

- The device has 6 functions, each mapped to the buttons on the right side of the board.

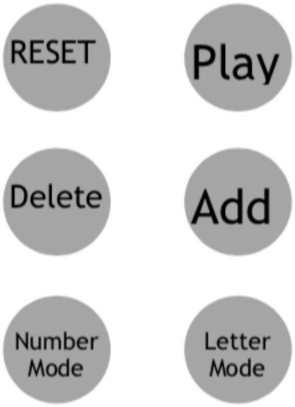
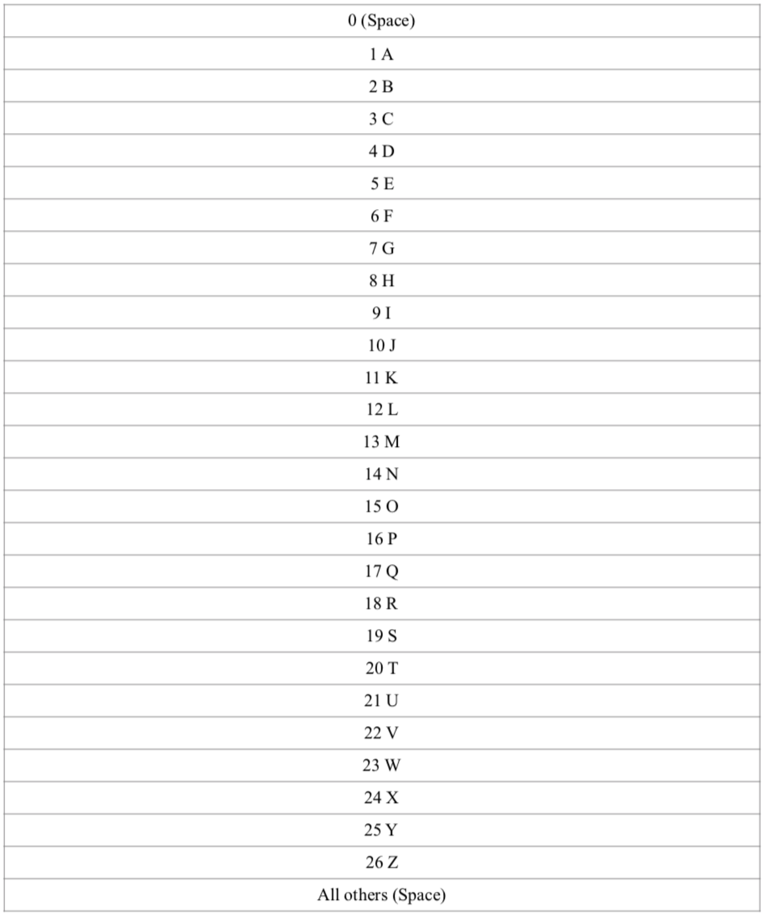


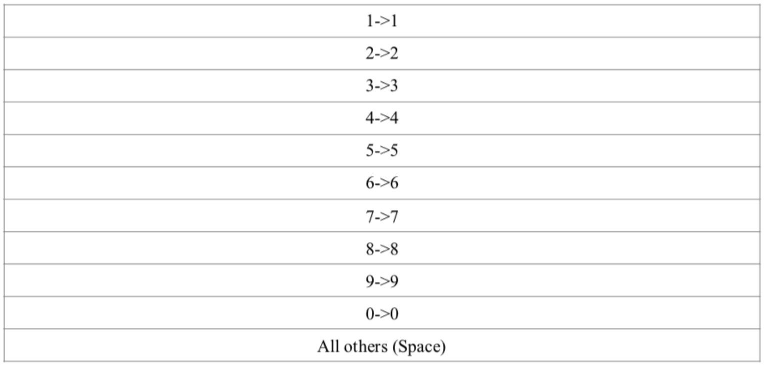
Figure 1.2: Button Mapping

- By default, the device is set to **Letter Mode**. In this configuration the input is read as a binary integer representation (not the ASCII code) of a specific letter.
- When **Number Mode** is pressed, it will instead encode input as a digit rather than attempting to read it as a number. Pressing either button will switch between the two modes.
- **Add** encodes the input depending on the selected mode into a buffer which can store up to 64 characters, including spaces.
- **Delete** functions as a backspace and erases the last input.
- The **Play** button will load from the buffer in a first-in-first-out manner, and play out the Morse Code encoding using the LEDs.
- The **RESET** button is a shortcut to clear the entire buffer without having to mash the **Delete** button.



0 (Space)
1 A
2 B
3 C
4 D
5 E
6 F
7 G
8 H
9 I
10 J
11 K
12 L
13 M
14 N
15 O
16 P
17 Q
18 R
19 S
20 T
21 U
22 V
23 W
24 X
25 Y
26 Z
All others (Space)

Figure 1.3: Letter Input Table



1->1
2->2
3->3
4->4
5->5
6->6
7->7
8->8
9->9
0->0
All others (Space)

Figure 1.4: Numeric Input Table

Chapter 2

Methodology

2.1 Modules

- main.vhd:
  - Single Module Project

2.2 Module Descriptions

2.2.1 main.vhd

- Top Module of the project.
- At startup a short animation is played to indicate the program has been loaded. (LEDs light up from right to left)
- Contains the above tables in the form of arrays.
- Switches between a number of one-hot encoded states at clock intervals.
- The **init** state serves as the default state.
- The **playcode** state plays out from the buffer, **MEMORYARRAY**. It loads the next piece to print out, uses the table representation to display it and finally transitions to the **charSep** state to add a character separator (3 OFFs).
- The **setnummode** and **setletmode** are used to switch between the number and letter modes respectively.
- **addNewChar** and **deleteChar** are used to encode (based on the mode that has been set) or remove characters.
- **resetRec** is used to clear the entire buffer.

- **clear** is an intermediate state used to tell the device to shift all lit LEDs to the left until there are none left. Some of its purposes include playing the activation animation or finalizing a message after the **playcode** state.
- **done** is used as a state to wait for the next input. The state is used to for instance prevent accidentally deleting too many characters by accidentally holding the **Delete** button by checking to see if the button is being held.

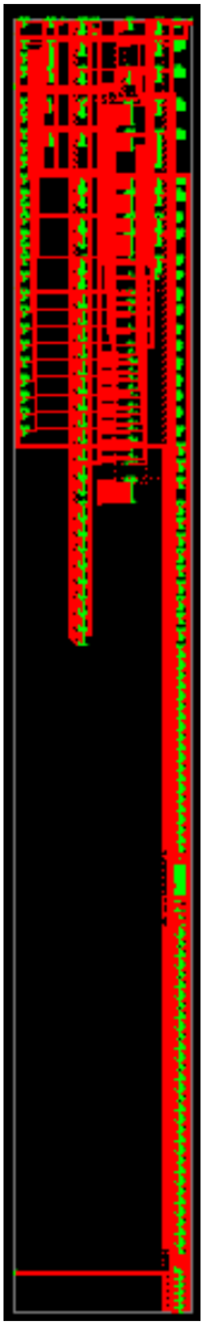


Figure 2.1: Circuit Schematic

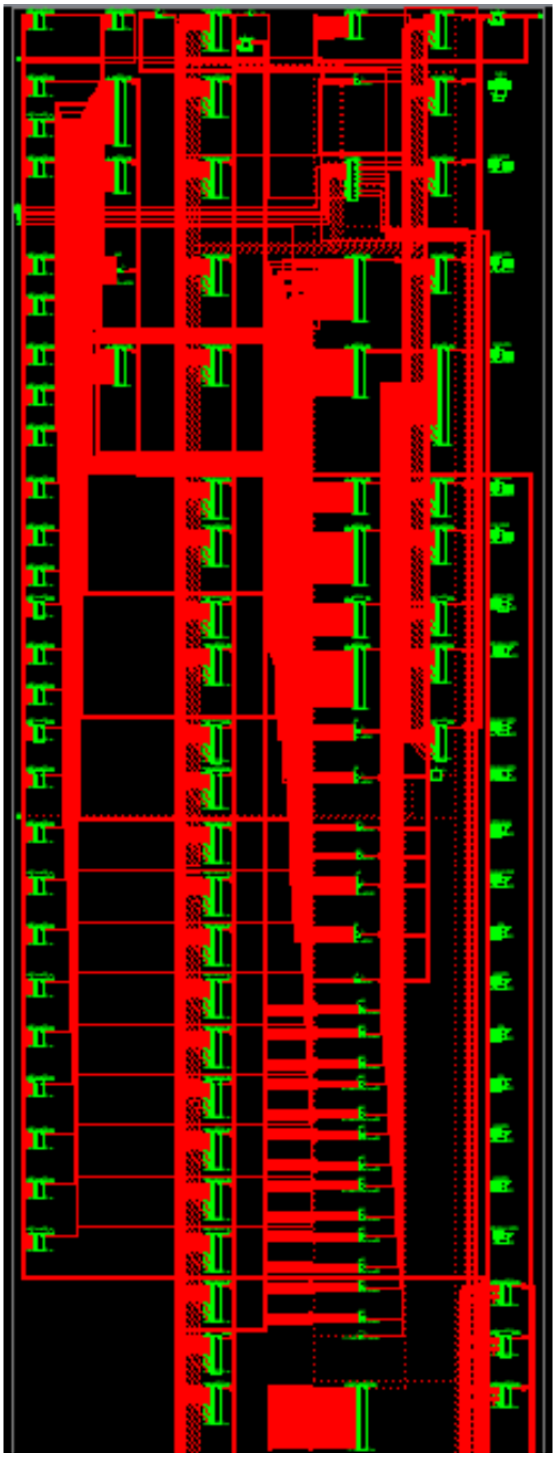


Figure 2.2: Circuit Schematic Close-up

Chapter 3

Results

The circuit was able to run the program as described above:

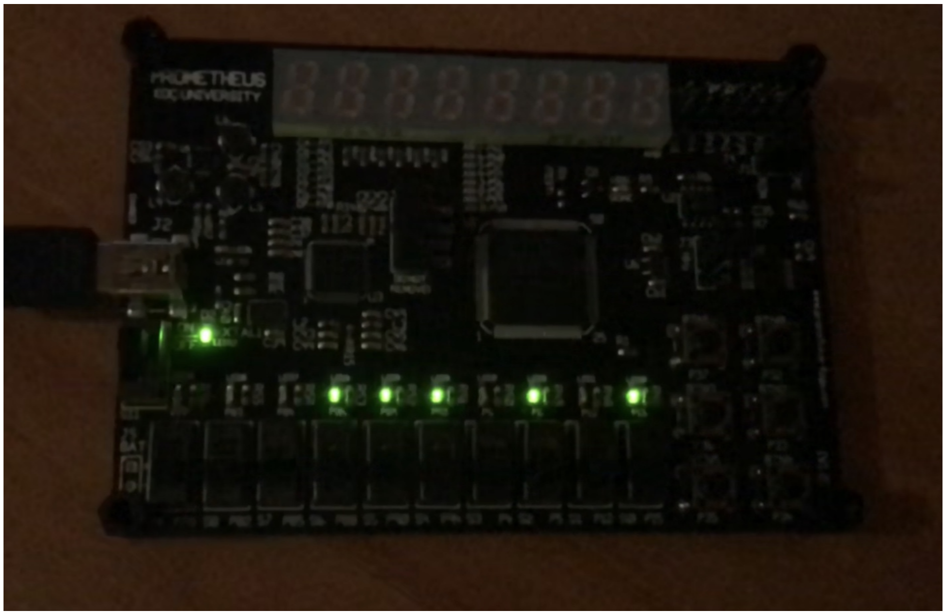


Figure 3.1: The Letter D

After encoding the message ”this is so sad” and playing it, the circuit was able to produce a meaningful result. Every single LED is meant to represent a *dot* as described in Figure 1.1.

Chapter 4

Closing Notes

4.1 Discussion

This project was, no matter how exhausting, quite enlightening in understanding how to use arrays for both storing precomputed values as well as having an efficient way to maintain a system memory. Some additional features that could've been added:

- changing the mode select to a single button and using the 7-segment display to show the mode.
- utilizing the 7-segment display to show the current state.
- using the freed-up button to play an "SOS" message.

4.2 Conclusion

Of all my lab projects, this took by far the most time and effort but I feel that it also taught me the most. As well as being a fun idea, a chance to learn Morse Code it was also a great chance to make a device that I can consider sophisticated. Although very poorly optimized being able to work with memory, giving the user the ability to read/write all seem to me as an important step in my hardware programming career.