**Lebanese American University**

A close up of a logo

Description automatically generated

***COE 322 – Logic Design Lab – Project***

Dia Khalil-

Michelle Baalbaky-

Farah Mikhael-

Karim Nehme

**Table of Contents**

[Table of Figures 3](#_Toc192674675)

[Table Of Tables 4](#_Toc192674676)

[Introduction 5](#_Toc192674677)

[Equipment needed 6](#_Toc192674678)

[Part 1: Experiment 7](#_Toc192674679)

[A- Design and Implementation of a 7-Segment Display 7](#_Toc192674680)

[B- Automation System 11](#_Toc192674681)

[Part2: Theory 25](#_Toc192674682)

[Conclusion 28](#_Toc192674683)

[References 29](#_Toc192674684)

Table of Figures

[Figure 1: 7 segment display chip 7](#_Toc192674685)

[Figure 2: Connections of the 7-segment display 7](#_Toc192674686)

[Figure 3: Circuit that shows our group number implemented on the breadboard 9](#_Toc192674687)

[Figure 4: Circuit Implementation on the breadboard 9](#_Toc192674688)

[Figure 5: Breadboard implementation for 0--- 18](#_Toc192674689)

[Figure 6: Breadboard implementation for 1--- 18](#_Toc192674690)

[Figure 7:Breadboard implementation for 2--- 19](#_Toc192674691)

[Figure 8:Breadboard implementation for 3--- 19](#_Toc192674692)

[Figure 9:Breadboard implementation for 4--- 20](#_Toc192674693)

[Figure 10:Breadboard implementation for 5--- 20](#_Toc192674694)

[Figure 11:Breadboard implementation for 6--- 21](#_Toc192674695)

[Figure 12:Breadboard implementation for FULL 21](#_Toc192674696)

[Figure 13:Schematic of the 1st seven segments 23](#_Toc192674697)

[Figure 14:Waveform of the 1st seven segment 23](#_Toc192674698)

[Figure 15:Schematic of the 2nd ,3rd and 4th seven segments 24](#_Toc192674699)

[Figure 16:Waveform of the 2nd ,3rd and 4th seven segments 24](#_Toc192674700)

[Figure 17: Common Cathode and Common Anode Seven Segment Displays 25](#_Toc192674701)

[Figure 18 74LS47 Pinout 27](#_Toc192674702)

Table Of Tables

[Table 1: Truth Table representing 4 inputs and 7 outputs to be fed to the 7-segment display 8](#_Toc192674703)

[Table 2: Truth table of the first seven segment display 11](#_Toc192674704)

[Table 3:K-map of a 11](#_Toc192674705)

[Table 4: K-map of b 12](#_Toc192674706)

[Table 5:K-map of c 12](#_Toc192674707)

[Table 6:K-map of d 13](#_Toc192674708)

[Table 7:K-map of e 13](#_Toc192674709)

[Table 8:K-map of f 13](#_Toc192674710)

[Table 9:K-map of g 14](#_Toc192674711)

[Table 10:Truth table of the second seven segment: U 14](#_Toc192674712)

[Table 11: : K-map of b, c, d, e, f 15](#_Toc192674713)

[Table 12:K-map of g for “U” 15](#_Toc192674714)

[Table 13: Truth table of the 3rd and 4th seven segment: LL 16](#_Toc192674715)

[Table 14:K-map of d, e, f 16](#_Toc192674716)

[Table 15:K-map of g for “L” 17](#_Toc192674717)

[Table 16: 74LS47 Pinout Explanation 27](#_Toc192674718)

Introduction

# Truth Table Analysis

## State Definition and Encoding

Each state has unique output expressions for L1-L4 and next state logic N1-N3. The ideal encoding for these states using C3, C2, C1 would be:

* BOOT: 000
* SEQ1: 001
* SEQ2: 010
* SEQ3: 011
* SEQ4: 100
* ST: 101
* LOCKED: 111

# Quartus

# Virtual Breadboard

S1: white (7-1)

S2: red (2)

S3: green (3)

S4: purple (5-4)

I6: PIN 13 GROUND

VCC: YELLOW

Ground: Black

PIN 5: L1 OUTPUT BLUE nb on breadboard 8

S2🡪C3 PIN 9 brown

S1🡪C2 PIN 10 orange

S0🡪C1 PIN 11 yellow

1. **Number of MUXes**: We need 4 separate 8-to-1 multiplexers, one for each output (L1, L2, L3, L4).
2. **State Encoding**:
   * BOOT: 000
   * SEQ1: 001
   * SEQ2: 010
   * SEQ3: 011
   * SEQ4: 100
   * ST: 101
   * State 110: GROUNDED
   * LOCKED: 111
3. **Control Signals**: The selection lines for each MUX are the state bits (C3,C2,C1).

**How the MUX Works**

Each 8-to-1 multiplexer has:

* 8 data inputs (i₀ through i₇)
* 3 select lines (C3,C2,C1) that determine which input is connected to the output
* 1 output (L1, L2, L3, or L4)

The select lines correspond to the state encoding bits. For example, when the state is BOOT (000), input I0 of each MUX is selected. When the state is SEQ1 (001), input I1 is selected, and so on.

**Each MUX Is Summarized as Follows:**

**L1 MUX (Blue):**

* BOOT (000): Outputs S₁
* SEQ1 (001): Outputs S₁
* SEQ2 (010): Outputs S₄
* SEQ3 (011): Outputs S₂
* SEQ4 (100): Outputs S₃
* ST (101): Outputs S₂
* 110: Grounded (0)
* LOCKED (111): Outputs S₁

**L2 MUX (Red):**

* BOOT (000): Outputs S₂
* SEQ1 (001): Outputs S₂
* SEQ2 (010): Outputs S₂
* SEQ3 (011): Outputs S₄
* SEQ4 (100): Outputs S₁
* ST (101): Outputs 0
* 110: Grounded (0)
* LOCKED (111): Outputs S₂

**L3 MUX (Green):**

* BOOT (000): Outputs S₃
* SEQ1 (001): Outputs S₃
* SEQ2 (010): Outputs S₁
* SEQ3 (011): Outputs S₃
* SEQ4 (100): Outputs S₂
* ST (101): Outputs S₃
* 110: Grounded (0)
* LOCKED (111): Outputs S₃

**L4 MUX (Orange):**

* BOOT (000): Outputs S₄
* SEQ1 (001): Outputs S₄
* SEQ2 (010): Outputs S₃
* SEQ3 (011): Outputs S₁
* SEQ4 (100): Outputs S₄
* ST (101): Outputs S₁
* 110: Grounded (0)
* LOCKED (111): Outputs S₄

A diagram of a computer program

AI-generated content may be incorrect.