
CS 20 PROJECT 1 DOCUMENTATION

Binary/BCD Braille 3-bit Adder

Second Semester, A.Y. 2022-2023

Submitted By:

Angelo M. Bocobo
Raymund Klien B. Mañago
Ann Gabrielle C. Purisima

Lab 2
Group 4

10 May 2023

1. Project Overview

This project involves a circuit that adds two 3-bit unsigned binary inputs via two 4-position DIP switches and outputs the sum in either as a 4-bit binary or a BCD Braille . The circuit uses a display selector via a 4-position DIP switch that allows the user to choose between binary and BCD Braille output modes. Finally, the output is displayed using a 3 x 4 LED array as arranged in Figure 1.

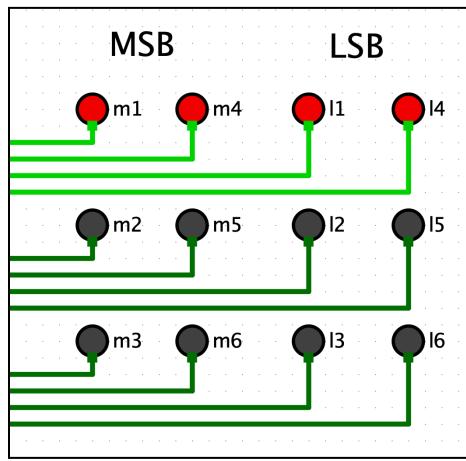


Figure 1. The 3 x 4 LED array.

2. Hardware Materials

The following hardware materials were used to accomplish the circuit:

- 3 x breadboards
- 12 x LEDs
- 3 x 4-position DIP switches
- 1 x 74LS83 IC (14-pin 4-bit full adder)
- 1 x 74LS04 IC (14-pin NOT gate)
- 1 x 74LS21 IC (14-pin 4-input AND gate)
- 6 x 74LS08 ICs (14-pin 2-input AND gate)
- 4 x 74LS32 ICs (14-pin 2-input OR gate)
- 6 x $10k\Omega$ resistors
- 5 x 330Ω resistors
- 4 x 300Ω resistors

3. Truth Tables

The **truth table of the 4-bit full adder** seen in Table 1 has two unsigned 3-bit binary inputs, namely X and Y, where X3 and Y3 are the MSB, respectively. Furthermore, it has a 4-bit binary output represented by B, C, D, and E wherein B is the MSB and E is the LSB.

X3	X2	X1	Y3	Y2	Y1	B	C	D	E
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	1
0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	1	0	0	1	1
0	0	0	1	0	0	0	1	0	0
0	0	0	1	0	1	0	1	0	1
0	0	0	1	1	0	0	1	1	0
0	0	0	1	1	1	0	1	1	1
0	0	1	0	0	0	0	0	0	1
0	0	1	0	0	1	0	0	1	0
0	0	1	0	1	0	0	0	1	1
0	0	1	0	1	1	0	1	0	0
0	0	1	1	0	0	0	1	0	1
0	0	1	1	0	1	0	1	1	0
0	0	1	1	1	0	0	1	1	1
0	0	1	1	1	1	1	0	0	0
0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	1	0	0	1	1
0	1	0	0	1	0	0	1	0	0
0	1	0	0	1	1	0	1	0	1
0	1	0	1	0	0	0	1	1	0
0	1	0	1	0	1	0	1	1	1
0	1	0	1	1	0	1	0	0	0
0	1	0	1	1	1	1	0	1	1

0	1	1	0	0	0	0	0	1	1
0	1	1	0	0	1	0	1	0	0
0	1	1	0	1	0	0	1	0	1
0	1	1	0	1	1	0	1	1	0
0	1	1	1	0	0	0	1	1	1
0	1	1	1	0	1	1	0	0	0
0	1	1	1	1	0	1	0	0	1
0	1	1	1	1	1	1	0	1	0
1	0	0	0	0	0	0	1	0	0
1	0	0	0	0	1	0	1	0	1
1	0	0	0	1	0	0	1	1	0
1	0	0	0	1	1	0	1	1	1
1	0	0	1	0	0	1	0	0	0
1	0	0	1	0	1	1	0	0	1
1	0	0	1	1	0	1	0	1	0
1	0	0	1	1	1	1	0	1	1
1	0	1	0	0	0	0	1	0	1
1	0	1	0	0	1	0	1	1	0
1	0	1	0	1	0	0	1	1	1
1	0	1	0	1	1	1	0	0	0
1	0	1	1	0	0	1	0	0	1
1	0	1	1	0	1	1	0	1	0
1	0	1	1	1	0	1	0	1	1
1	1	0	0	0	0	0	1	1	0
1	1	0	0	0	1	0	1	1	1
1	1	0	0	1	0	1	0	0	1
1	1	0	1	0	0	1	0	1	0
1	1	0	1	0	1	1	0	0	1

1	1	0	1	1	0	1	1	0	0
1	1	0	1	1	1	1	1	0	1
1	1	1	0	0	0	0	1	1	1
1	1	1	0	0	1	1	0	0	0
1	1	1	0	1	0	1	0	0	1
1	1	1	0	1	1	1	0	1	0
1	1	1	1	0	0	1	0	1	1
1	1	1	1	0	1	1	1	0	0
1	1	1	1	1	0	1	1	0	1
1	1	1	1	1	1	1	1	1	0

Table 1. Truth table for the 4-bit full adder with two unsigned 3-bit binary inputs.

Meanwhile, **the truth table of the binary/BCD Braille 3-bit adder** has 5 inputs and 12 outputs as seen in Table 2. In particular, A represents the display selector, i.e. 0 if binary and 1 if Braille BCD; while B, C, D, and E represent the 4-bit binary sum wherein B is the MSB and E is the LSB. On the other hand, m1, m2, m3, m4, m5, and m6 represent the first and second column of the 3 x 4 LED array; while l1, l2, l3, l4, l5, and l6 represent the third and fourth column as arranged in Figure 1.

A	B	C	D	E	m1	m2	m3	m4	m5	m6	l1	l2	l3	l4	l5	l6
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0	0	1	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	1	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	1	0	0	1	0	0	0	0	0
0	0	1	1	1	0	0	0	1	0	0	1	0	0	1	0	0
0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0
0	1	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	1	1	0	0	0	0	0	1	0	0	1	0	0

0	1	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
0	1	1	0	1	1	0	0	1	0	0	0	0	0	1	0	0	0
0	1	1	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0
0	1	1	1	1	1	0	0	1	0	0	1	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0
1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0
1	0	0	1	1	0	0	0	0	0	0	1	0	0	1	0	0	0
1	0	1	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0
1	0	1	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0
1	0	1	1	0	0	0	0	0	0	0	1	1	0	1	0	0	0
1	1	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0
1	1	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0
1	1	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	1	1	0	0	1	1	0
1	1	1	0	1	1	0	0	0	0	0	1	1	0	1	1	0	0
1	1	1	1	0	0	1	0	0	0	0	1	1	0	0	0	0	0
1	1	1	1	1	0	1	0	0	0	0	1	0	0	1	1	0	0
1	1	1	1	1	1	1	0	0	0	0	1	0	0	1	0	0	0

Table 2. Truth table of the binary/BCD Braille 3-bit adder.

4. Logical Expressions

1. K-Map for **m1** with SOP: $A'B + BD + BC$

ABC \ DE	00	01	11	10
000	0	0	0	0
001	0	0	0	0
011	1	1	1	1
010	1	1	1	1
100	0	0	0	0
101	0	0	0	0
111	1	1	1	1
110	0	0	1	1

2. K-Map for **m2** with SOP: 0

ABC \ DE	00	01	11	10
000	0	0	0	0
001	0	0	0	0
011	0	0	0	0
010	0	0	0	0
100	0	0	0	0
101	0	0	0	0
111	0	0	0	0
110	0	0	0	0

3. K-Map for **m3** with SOP: **0**

ABC \ DE	00	01	11	10
000	0	0	0	0
001	0	0	0	0
011	0	0	0	0
010	0	0	0	0
100	0	0	0	0
101	0	0	0	0
111	0	0	0	0
110	0	0	0	0

4. K-Map for **m4** with SOP: **A'C**

ABC \ DE	00	01	11	10
000	0	0	0	0
001	1	1	1	1
011	1	1	1	1
010	0	0	0	0
100	0	0	0	0
101	0	0	0	0
111	0	0	0	0
110	0	0	0	0

5. K-Map for **m5** with SOP: **0**

ABC \ DE	00	01	11	10
000	0	0	0	0
001	0	0	0	0
011	0	0	0	0
010	0	0	0	0
100	0	0	0	0
101	0	0	0	0
111	0	0	0	0
110	0	0	0	0

6. K-Map for **m6** with SOP: **0**

ABC \ DE	00	01	11	10
000	0	0	0	0
001	0	0	0	0
011	0	0	0	0
010	0	0	0	0
100	0	0	0	0
101	0	0	0	0
111	0	0	0	0
110	0	0	0	0

7. K-Map for I1 with SOP: $A'D + B'D + DE + AB'E + AC + ABD'E'$

ABC \ DE	00	01	11	10
000	0	0	1	1
001	0	0	1	1
011	0	0	1	1
010	0	0	1	1
100	0	1	1	1
101	1	1	1	1
111	1	1	1	1
110	1	0	1	0

8. K-Map for I2 with SOP: $AC'E' + AB'CD + ABC'D' + ABD'E'$

ABC \ DE	00	01	11	10
000	0	0	0	0
001	0	0	0	0
011	0	0	0	0
010	0	0	0	0
100	1	0	0	1
101	0	0	1	1
111	1	0	0	0
110	1	1	0	1

9. K-Map for I3 with SOP: 0

ABC \ DE	00	01	11	10
000	0	0	0	0
001	0	0	0	0
011	0	0	0	0
010	0	0	0	0
100	0	0	0	0
101	0	0	0	0
111	0	0	0	0
110	0	0	0	0

10. K-Map for I4 with SOP: A'E + B'DE + BD'E + AB'D'E' + ACD + ABDE'

ABC \ DE	00	01	11	10
000	0	1	1	0
001	0	1	1	0
011	0	1	1	0
010	0	1	1	0
100	1	0	1	0
101	1	0	1	1
111	0	1	1	1
110	0	1	0	1

11. K-Map for **I5** with SOP: $AB'D'E' + AC'D'E' + AB'CE + ABDE'$

ABC \ DE	00	01	11	10
000	0	0	0	0
001	0	0	0	0
011	0	0	0	0
010	0	0	0	0
100	1	0	0	0
101	1	1	1	0
111	0	0	0	1
110	1	0	0	1

12. K-Map for **I6** with SOP: **0**

ABC \ DE	00	01	11	10
000	0	0	0	0
001	0	0	0	0
011	0	0	0	0
010	0	0	0	0
100	0	0	0	0
101	0	0	0	0
111	0	0	0	0
110	0	0	0	0

5. Diagrams

The diagram of the circuit using Logisim is seen in Figure 2. It uses the materials (except the resistors) as enumerated in Section 2 of the documentation.

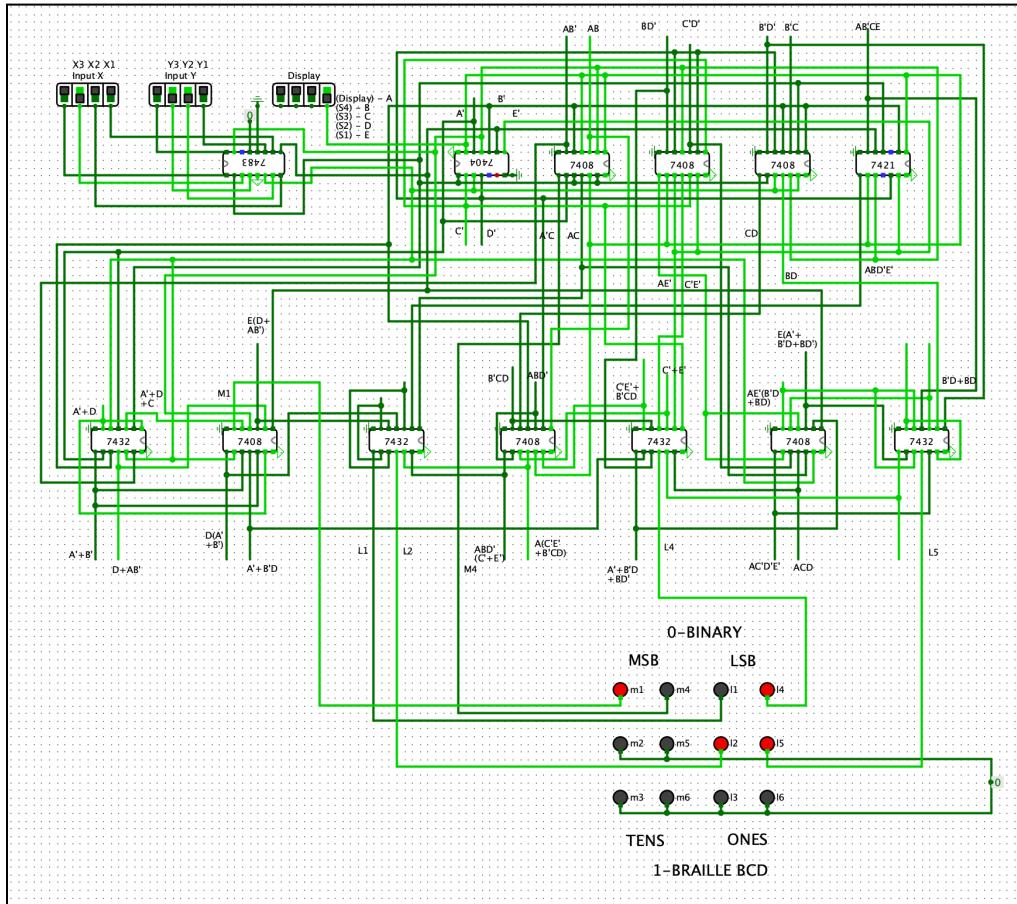


Figure 2. The diagram of the circuit using Logisim.

Moreover, the following are the pinouts of the used ICs, i.e., 74LS83, 74LS04, 74LS21, 74LS08, and 74LS32, shown in Figures 3, 4, 5, 6, and 7 respectively. (Diagrams retrieved from: <https://datasheethub.com>)

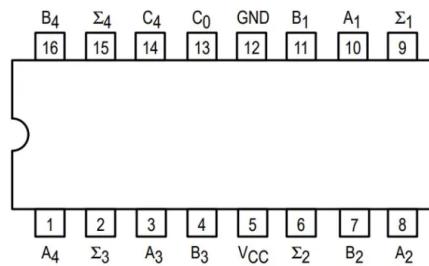


Figure 3. 74LS83 pinout.

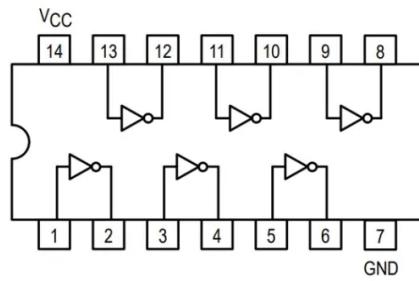


Figure 4. 74LS04 pinout.

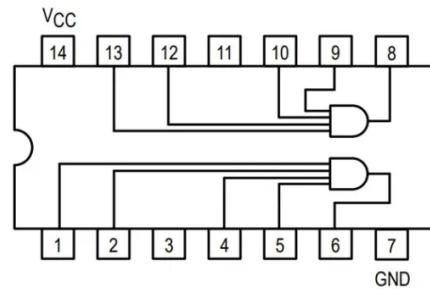


Figure 5. 74LS21 pinout.

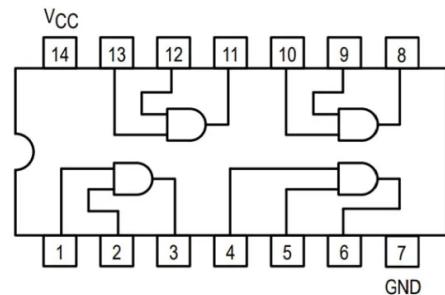


Figure 6. 74LS08 pinout.

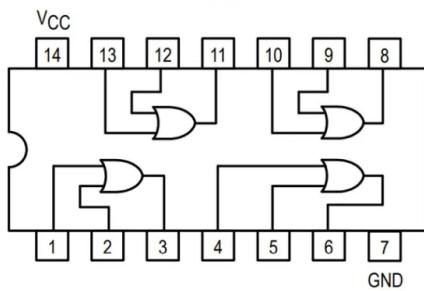


Figure 7. 74LS32 pinout.

6. Physical Implementation

With three breadboards in total, the circuit begins with two 4-position DIP switches, wherein both switches are used for two unsigned 3-bit binary inputs.

The output pins of the two DIP switches are connected to four 300Ω resistors and four 330Ω resistors to resolve floating pins issue when the switch is off. These inputs are then fed into a 4-bit full adder IC which will produce a 4-bit binary output represented by B, C, D, and E, where B is the MSB and E is the LSB.

The 4-bit binary output, together with the display selector bit represented by A, will go through numerous logic gates using ICs that follows the truth table seen in Table 2. The resulting output are then reflected on the the 3 x 4 LED array with $10k\Omega$ resistors for each LED.

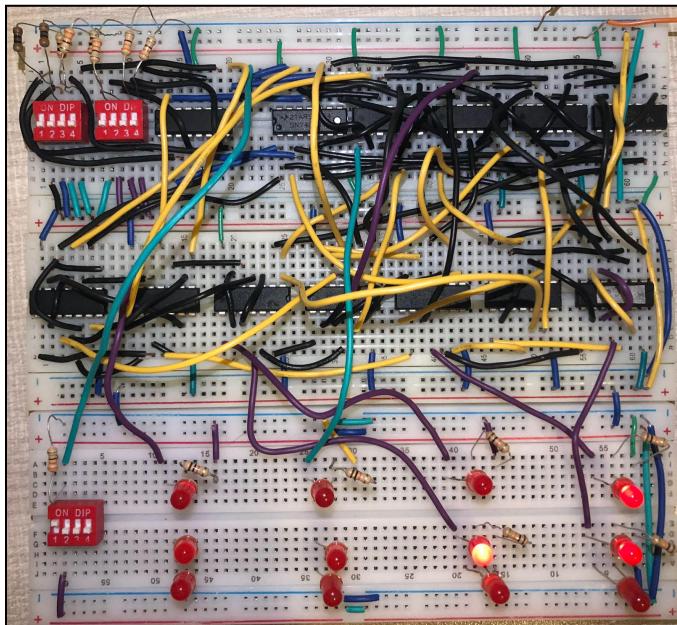


Figure 8. Physical implementation of the circuit in Braille BCD mode when adding two 000s.