ISTANBUL TECHNICAL UNIVERSITY

COMPUTER ENGINEERING DEPARTMENT

BLG 242E

DIGITAL CIRCUITS LABORATORY EXPERIMENT REPORT

EXPERIMENT NO	:	1
EXPERIMENT DATE	:	17.03.2023
LAB SESSION	:	FRIDAY - 16.00
GROUP NO	:	G18

GROUP MEMBERS:

150220770 : ONUR BAYLAM

150200916 : DENIS IURIE DAVIDOGLU

SPRING 2023

Contents

1 INTRODUCTION [10 points]

Briefly describe what you have done during the experiment.

2 MATERIALS AND METHODS [40 points]

Answer all questions and provide everything that are indicated in the "Report" section of the related experiment in "Experiments Booklet". Edit following sub-heading as you wish.

2.1 Preliminary

Figure 1: An Example Figure Caption[**?**]

- 74HC04: Hex Inverters IC.
- 74HC08: 2-input Positive-AND gates IC.
- 74HC32: 2-input Positive-OR gates IC.

2.2 Experiment

- **Part 1:** Design and implement the logic circuits for the given expressions below by using the necessary gates.
  $F_1(a, b) = a + ab$
  $F_2(a, b) = (a + b).(a + \overline{b})$

  NOTE: In order to design and implement boolean functions, we benefit from Logisim tool which enables us to simulate our circuits with different input combinations and detect possible errors. In addition, we will illustrate our implementions of the

circuits using IC schemas.

Design of F_1:

To design and implement F_1, we need to use a 2-input AND gate, a 2-input OR gate.

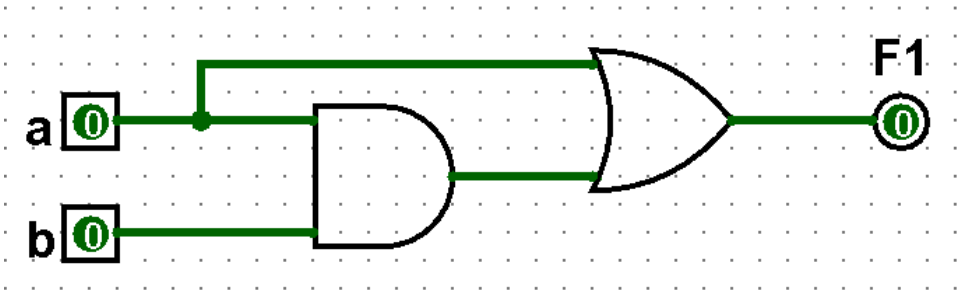

Figure 2: Design of F_1 in Logisim

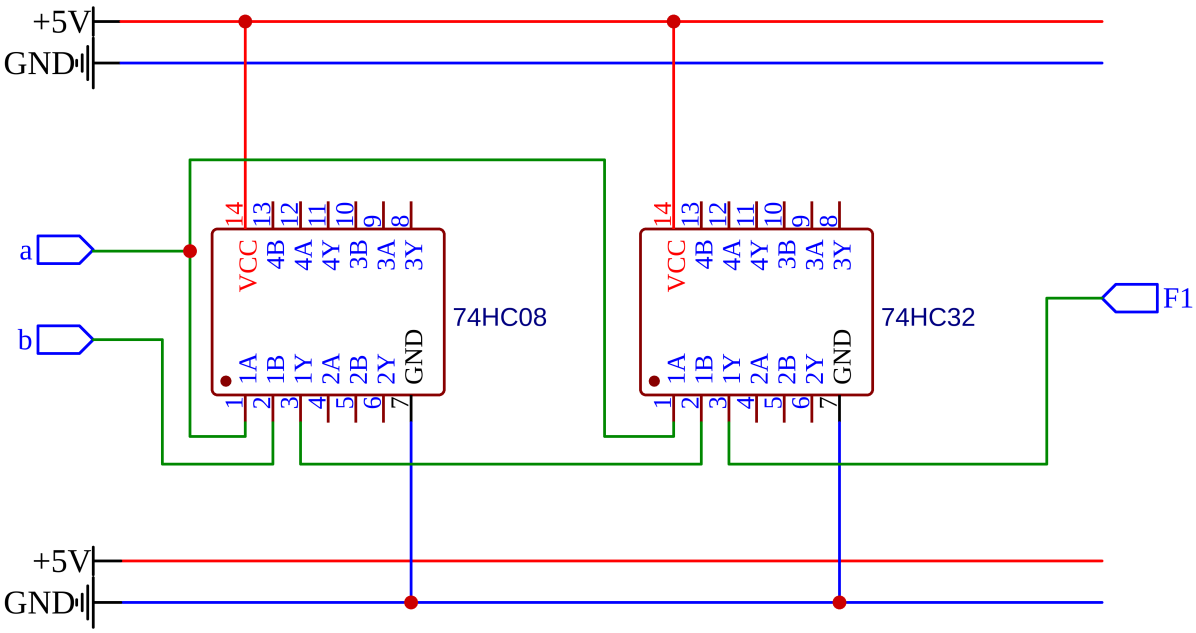

Figure 3: Design of F_1 in EasyEDA

Design of F_2:

To design and implement F_2, we need to use two 2-input OR gates, a 2-input AND gate, a NOT gate.

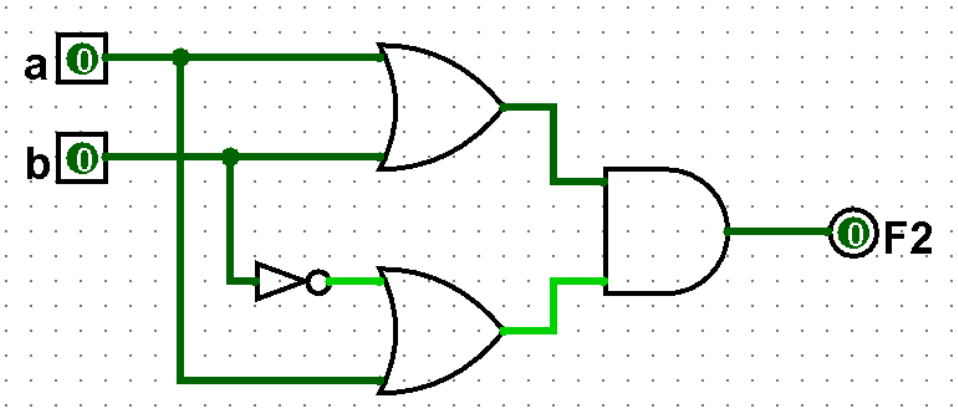

Figure 4: Design of F_2 in Logisim

Functions F_1 and F_2 designed and implemented using ICs:

- 74HC08: 2-input Positive-AND gates IC.
- 74HC32: 2-input Positive-OR gates IC.

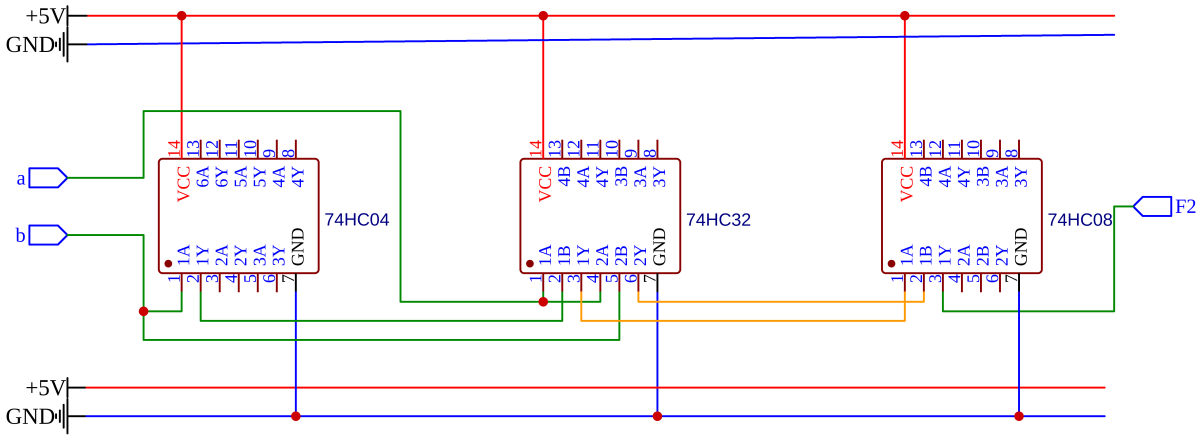

Figure 5: Design of F_2 in EasyEDA

To validate our designs of F_1 and F_2, we need to create truth tables of them. Both F_1 and F_2 functions have the same truth tables:

a	b	F_1
0	0	0
0	1	0
1	0	1
1	1	1

Figure 6: Truth table of the function F_1 and F_2

Our implementations of F_1 and F_2 are compatible with their truth table.
Moreover, we can construct the same implementations with the help of **SPDT** switch which consists of three terminals, one is output and others are input terminals. Using switch button of **SPDT**, we are able to determine which input terminal will be connected to the output terminal.

Implementations of F_1 and F_2 using SPDT switch:

a	b	F_1
0	0	0
0	1	0
1	0	1
1	1	1

Figure 7: Implementations of F_1 and F_2 using SPDT switch

- **Part 2:** A theorem is given as: $a + (a \cdot b) = a$. First, determine the dual of the given theorem and then, implement the functions for both sides of the dual theorem by using logic gates. Validate the truth of the theorem by comparing the changes in the outputs.

  First of all, the dual of a theorem is an expression created by replacing OR with AND, AND with OR, 1 with 0 and 0 with 1. Also, the order of operations should be preserved, so we need to put extra parentheses. In this formula, $(a \cdot b)$ is done first:

$$a + (a \cdot b) = a \qquad \text{(Original)}$$

$$a \cdot (a + b) = a \qquad \text{(Dual)}$$

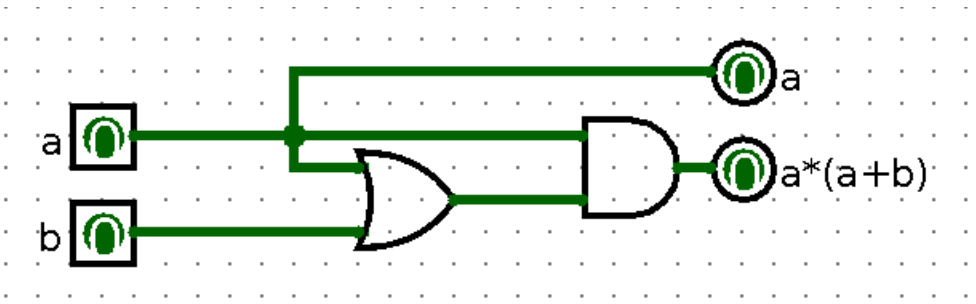

Figure 8: Design in Logisim

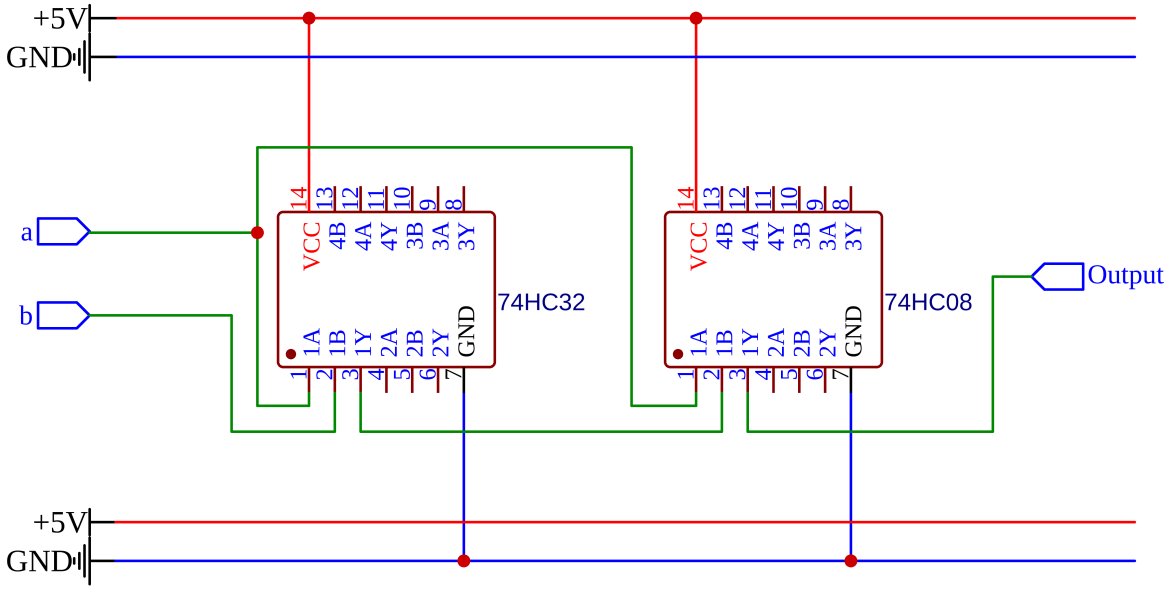

Figure 9: Design in EasyEDA

To check validity of the dual theorem, the following truth table is constructed:

a	b	$a \cdot b$	$a + b$	$a + a \cdot b$	$a \cdot (a + b)$
0	0	0	0	0	0
0	1	0	1	0	0
1	0	0	1	1	1
1	1	1	1	1	1

As it is clear from the table, the last two columns, which are the original expression's and its dual's left-hand side truth values, are identical to the first column, the right-hand side of the both equations. The equivalency of functions a and $a \cdot (a+b)$ confirms the dual theorem.

- **Part 3:** $F_3(a,b,c) = ab + \overline{a}c$ is given. First, determine the complement of the given function (F_3). Then, implement the circuit which realizes the complementary function ($\overline{F_3}$). Validate your implementation by using the truth table.

  Step 1: In the first step, we find the complement of the given function F_3 by applying **De Morgan's** rule.

  Step 1.1: $\overline{F_3(a,b,c)} = \overline{ab + \overline{a}c}$

  Step 1.2: $\overline{F_3(a,b,c)} = (\overline{a} + \overline{b}).(a + \overline{c})$

  Step 2: In the second step, we implement the circuit which realizes the function $\overline{F_3}$ using logic gates and ICs.

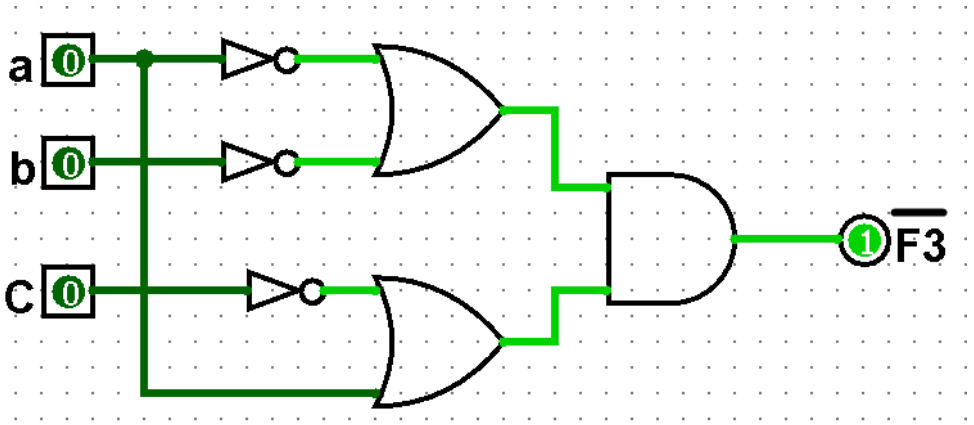

Figure 10: Design of $\overline{F_3}$ in Logisim

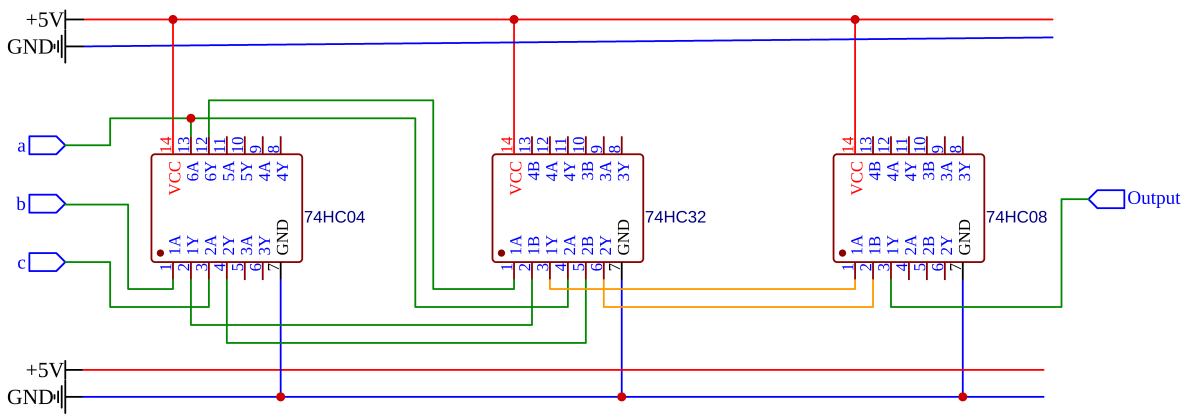

Figure 11: Design of $\overline{F_3}$ in EasyEDA

Step 3: In the third step, we validate our implementation of the function $\overline{F_3}$ using truth table.

a	b	c	$\overline{F_3}$
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Figure 12: Truth table of $\overline{F_3}$

According to the truth table, our implementation is correct.

- **Part 4:** A basic logical function F_4 is defined as follows. $F_4(a, b, c, d) = \cup_1(1, 2, 5, 6, 9, 10, 13, 14)$. First, simplify given logical function and implement the simplified expression using logic gates. Validate your circuit by observing the outputs for each possible input.

  Simplification is done using Karnaugh map. The result suggests only two prime implicants as show below:

ABCD	F_4
0000	0
0001	1
0010	1
0011	0
0100	0
0101	1
0110	1
0111	0
1000	0
1001	1
1010	1
1011	0
1100	0
1101	1
1110	1
1111	0

$\longrightarrow$

AB \ CD	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	1	0	1
10	0	1	0	1

$\longrightarrow$ $C\overline{D} + \overline{C}D$

With the formula obtained, the following circuits are designed:

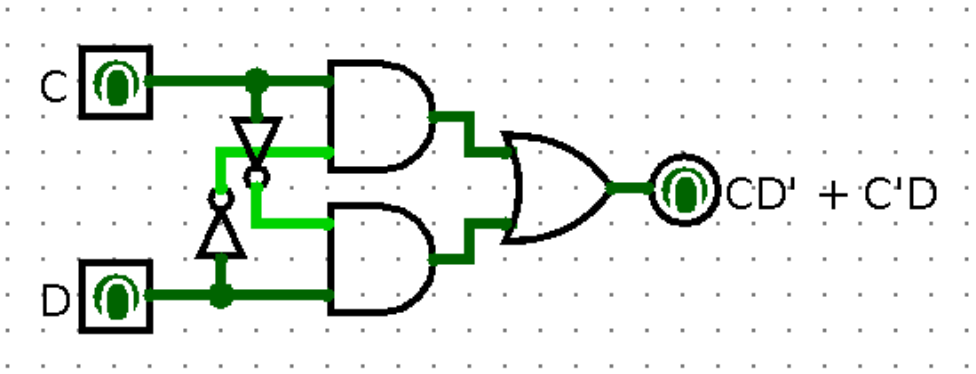

Figure 13: Design of F_4 in Logisim

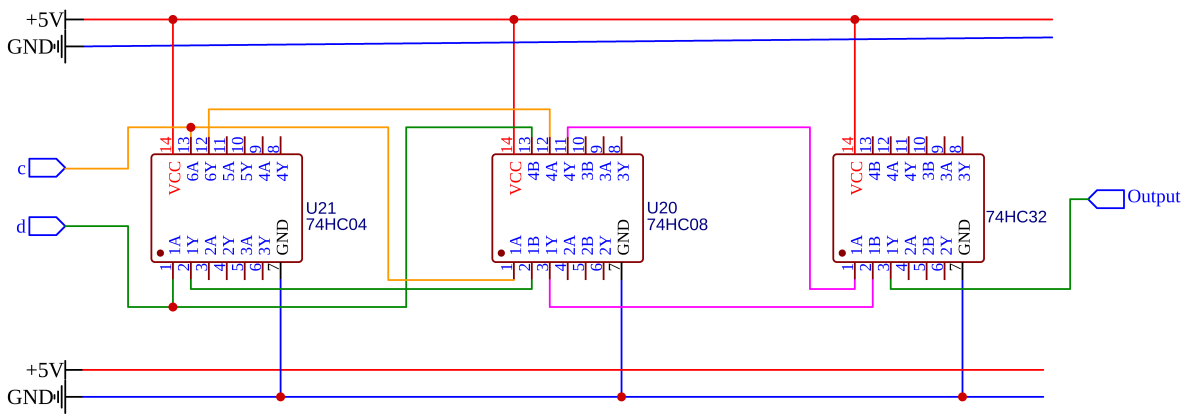

Figure 14: Design of F_4 in EasyEDA

- **Part 5:**
  - **By using voltmeter, observe V_{cc} and G_{nd} voltages:**
    Using voltmeter, we can observe the voltage difference between V_{cc} and G_{nd}. Also, we can alter the voltage difference to a specific value with the help of CADET interface. In our experiments, we set V_{cc} to 5V and G_{nd} to 0V, since these values are optimal for the ICs we use in our implementations. Also, we observed V_{cc} and G_{nd} voltages using voltmeter.
  - **By using potentiometer, build a resistance in 8 Kohm:**
    In CADET, we are able to specify a resistance value benefeting from potentiometer. To achieve this, we also need an ohmmeter to observe current resistance. So, in our experiment, we built a resistance value of 8 Kohm by adjusting the wiper of potentiometer in CADET. Also, we observed the value of the resistance simultaneously through the ohmmeter.
  - **Show "27" in two seven segment display by giving an input using 8 switches:**
    In this part of our experiment, we used eight different switches in CADET, four switches were used to display "2" and others were used to display "7" in seven segment display. During the experiment, we explored that our switches were operating as binary number digits and display the binary value as its corresponding decimal integer; before explaining in detail, we should examine our switch components:

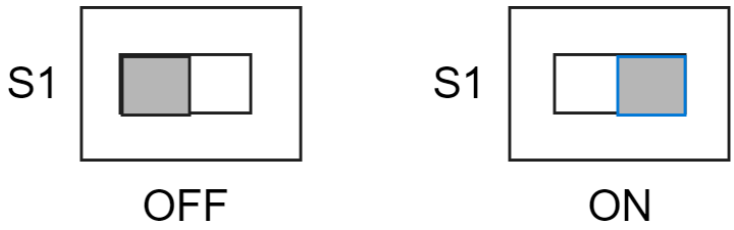

Figure 15: Switch component

Each switch corresponds to one single digit in our binary number. For instance, S0 (switch 0) determines the value of first digit in our binary number. If S0 is OFF, then first digit of the binary number is also 0. To illustrate the working principle of switches in a better way, we can use the schema below:

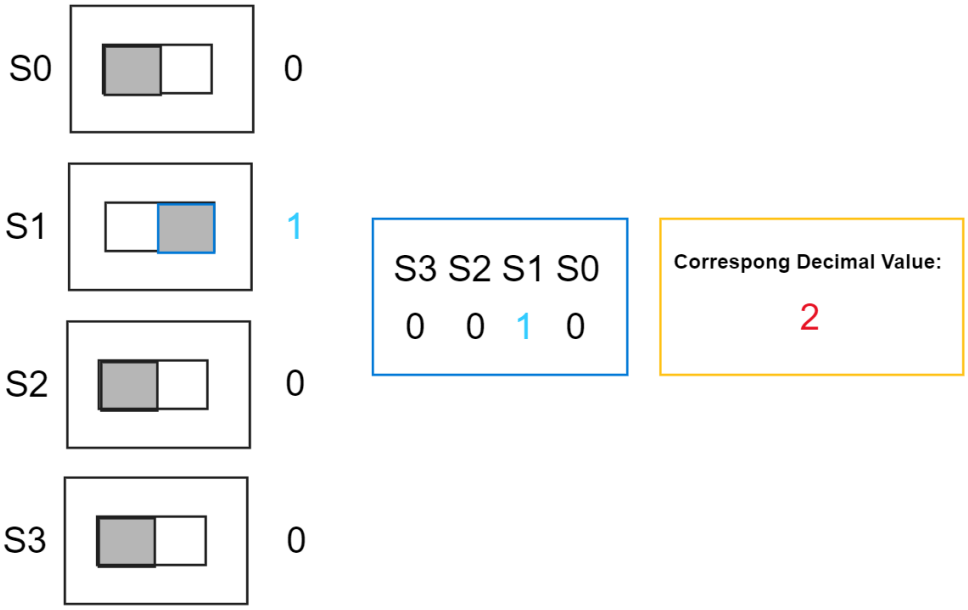

Figure 16: How to display "2" using switches

Therefore, in order to display "2", we set S1 to 1 (ON) and others are set to 0 (OFF).

Similarly, to display "7", we set S0, S1, S2 to 1 (ON), and S3 to 0 (OFF):

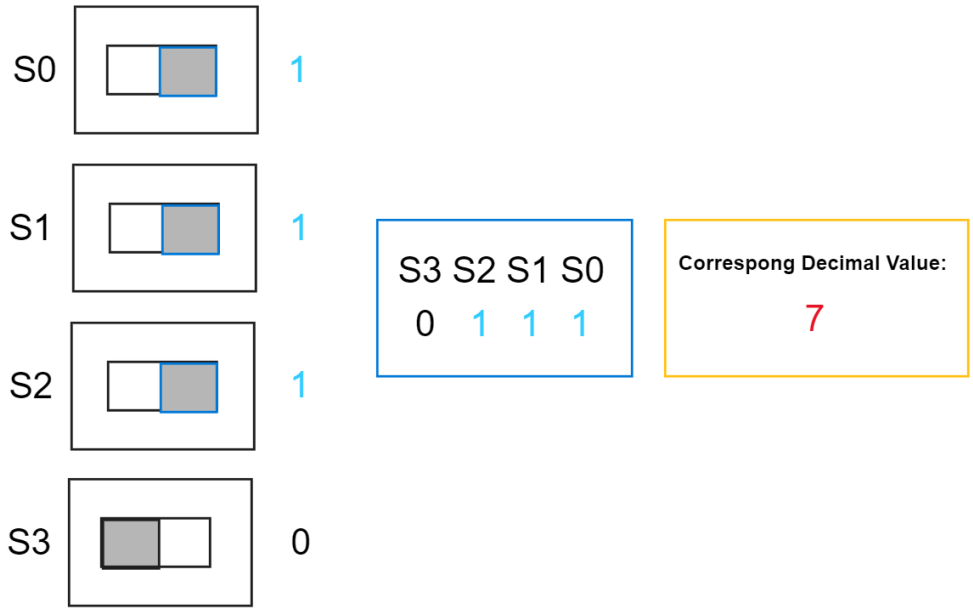

Figure 17: How to display ”7” using switches

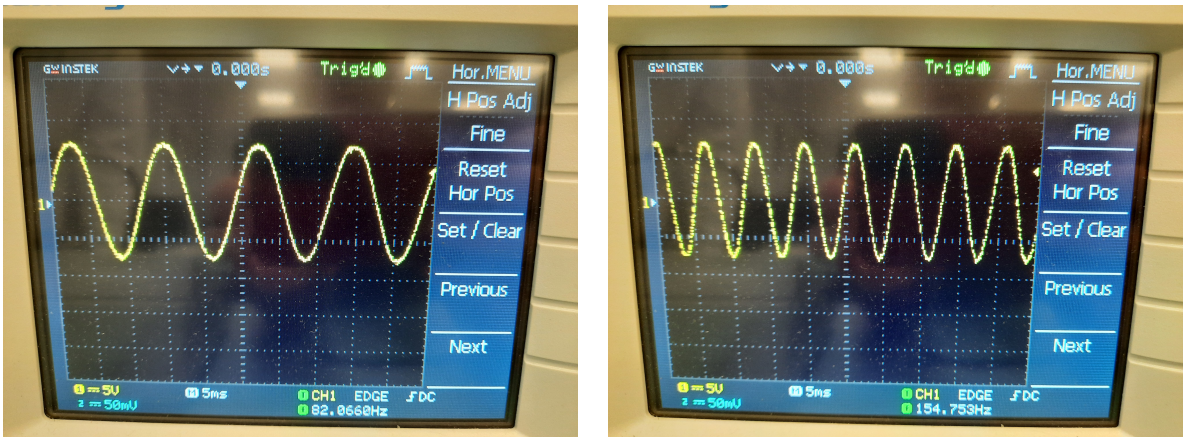

Figure 19: Sine waves generated

Figure 20: Square waves generated

Implementation of seven segment display using ICs:

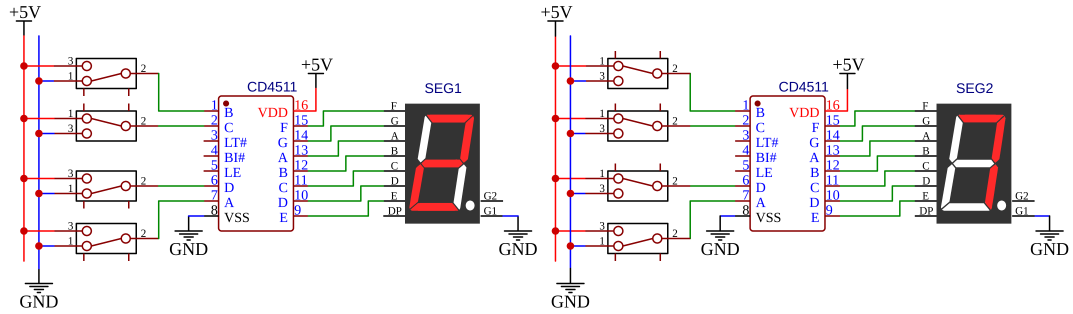

Figure 18: How to display "27" in two segment display using 8 switches

- **Part 6:** C.A.D.E.T. has a built-in 200kHz function generator with a number of knobs, which adjust the function to any desired characteristics. By changing the parameters, functions of different shapes were obtained:

Figure 21: Triangle waves generated

3 RESULTS [15 points]

During this experiment, all of the theoretical predictions made from boolean algebra axioms and truth tables turned out to be true. By using integrated circuits containing AND, OR, and NOT gates, different boolean functions were implemented. Besides that, we practiced the most basic prototyping on breadboard and applied our electronics knowledge.

4 DISCUSSION [25 points]

To elaborate on what we have done during experiment, a few things can be pointed out. First of all, we were required to study the datasheets of the 74 series ICs. The datasheets presented in the experiment booklet contained just the pinout diagrams and function tables. We noticed that the voltage supply and ground pins are in the same locations for most of the chips, and the outputs were directly below the inputs, a standard design choice which made it easier to quickly memorize the pins. Secondly, we learned how to use C.A.D.E.T., breadboards on it, wires and classical debugging with a multimeter. The latter was useful to check for connectivity and measuring high-low logic levels, when a pin was tricky to be accessed and connected to high/low probes. We configured SPDT switches to act as high/low inputs, while keeping in mind that it can be used in other different ways, such as switching between two signals. Although in this experiment there was no need for pull-up or pull-down resistors, we learned how to build desired resistance with a potentiometer. The oscilloscope and function generator were studied, and although digital electronics use only two voltage levels and a logic analyzer is a more practical device, on higher clock frequencies the square wave degrades and using oscilloscope in solving problems gives a full picture of what is going on.

5 CONCLUSION [10 points]

There were difficulties in getting used to the laboratory setup, we had problems primarily with the breadboards, function generator and oscilloscope. The breadboards were unusual by having power lines cut in the middle, and on our first tries of building circuits we were confused of getting high impedance state on output. The problem was not immediately obvious. Then the experiment went fine until the last part, where we were getting nothing meaningful on the oscilloscope display. We had no confidence in whether it was our wiring, function generator parameters or oscilloscope itself. With the help of professor, we found out that the function generator on our C.A.D.E.T. was broken and we tried other set instead. Overall, the experiment went exciting because we learned the practical side of designing digital circuitry.