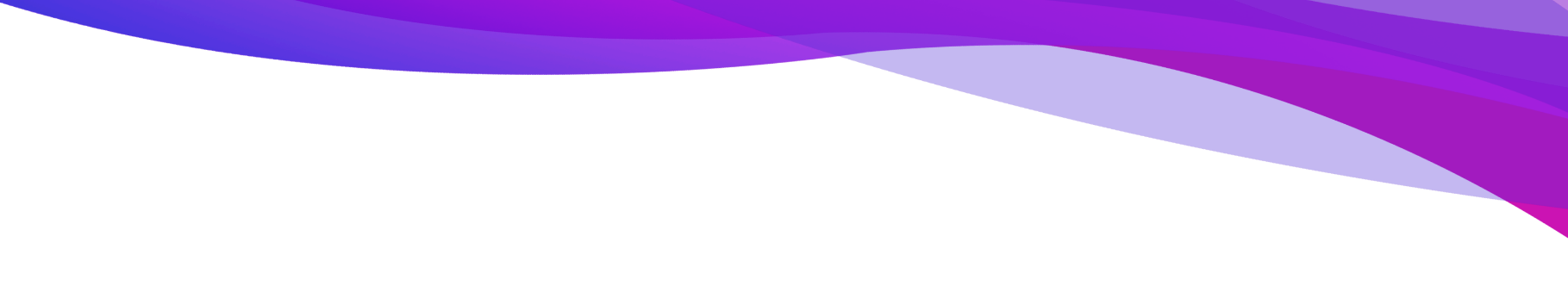
**A REPORT**

**ON**

**Design Assignment**

**In partial fulfilment of the requirements of course**

**CS F215: Digital Design**

|  |  |
| --- | --- |
|  |  |
|  |  |



**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI**

**K. K. Birla Goa Campus**

**November 2020**

**A REPORT**

**ON THE**

Problem Statement

(Topic 14) Design a toy for infants which is made up of an array of LEDs, say minimum 4x4. When a number from (0-9) is input into the

system the number must scroll across this LED screen.

BY

GROUP 103

|  |  |
| --- | --- |
| Vishal Vivek Bharambe  Sudeep Kumar Nemani  Rudra Pratap Singh Chouhan  Jayant Chaudhary  Aviral Kumar Goel  Hitarth Kothari | 2019A7PS0160G  2019A7PS0163G  2019A7PS0164G  2019A7PS0165G  2019A7PS0166G  2019A7PS0178G |

**CHAPTER 1 INTRODUCTION**

The problem statement posed to our group is as follows –

Design a toy for infants which is made up of an array of LEDs, say minimum 4x4. When a number from (0-9) is input into the system, the number must scroll across this LED screen.

Since the toy is supposed to be used by infants, various security features have been considered while modelling it. The major concerns with such toys for infants are –

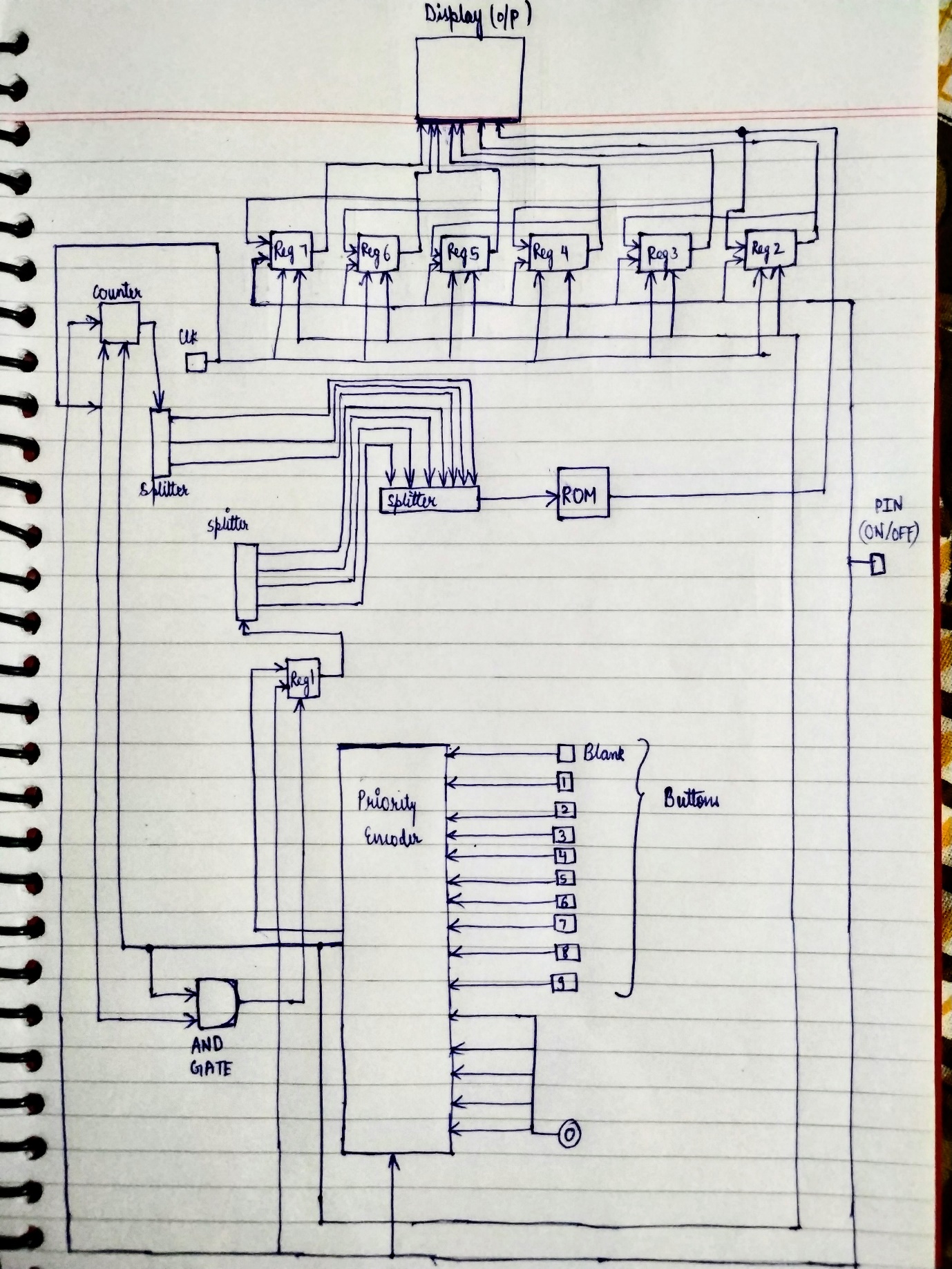
1. Accidently swallowing loose parts.
2. Dropping heavy toys onto themselves.
3. Toys breaking off due to repeated falling and thus resulting in loose parts.
4. Getting hurt by sharp edges.
5. Getting hurt by strangulation by more than seven-inch-long strings/wires.
6. Electric shocks and burns.

We modelled the toy to come in a secured plastic box with no removable parts so that the infant doesn’t accidentally break off swallow anything, addressing concern number 1. Plastic has been chosen so that toy is lightweight, thus addressing concern number 2. The toy is supposed to have eleven buttons: ten for numbers 0 to 9 and one extra button to clear the screen. The buttons will be locked in the casing of the box such that only half of the part is visible to press; thus, again ensuring complete safety for the infant. The box can have a clothed covering so that if some of the plastic parts accidentally break off after repeated falling, none of the contents will come out of that clothed covering minimising all risks and possibilities of infants swallowing the parts; addressing concern 3 and 4. The wired components and batteries will be enclosed in the box and would only be accessible after unzipping the cloth covering. Thus, addressing concern 5 and 6.

Upon pressing the On/Off button, the toy will be switched on or off depending upon the previous state. Upon pressing any of the buttons, the number will keep on scrolling on the LED screen until another numbered button or blank is pressed.

**CHAPTER 2 BLOCK DIAGRAM**

This is Block Diagram for the ‘Logic Design Block’ part of the overall block diagram shown in the System\_Block\_Design.pdf

****

Block diagram for the circuit. The arrows show the flow of the data.

**CHAPTER 3 ASSUMPTIONS**

Assumptions –

1. The conditions are assumed to be ideal. Wires are assumed to be of zero resistance, and hence propagation delay of wires is not considered.

2. The ROM can be hardwired to save cost as it doesn’t need to be updated or reprogrammed.

3. The user is supposed to hold the button for a maximum 2 clock ticks,

(To minimise the time, an idea was to have its clock with a greater frequency so that the two ticks are as short as possible)

The circuit is designed with respect to following assumptions –

1.The register updates in a maximum of 2 ticks in the circuit as soon as the desirable button is pressed.

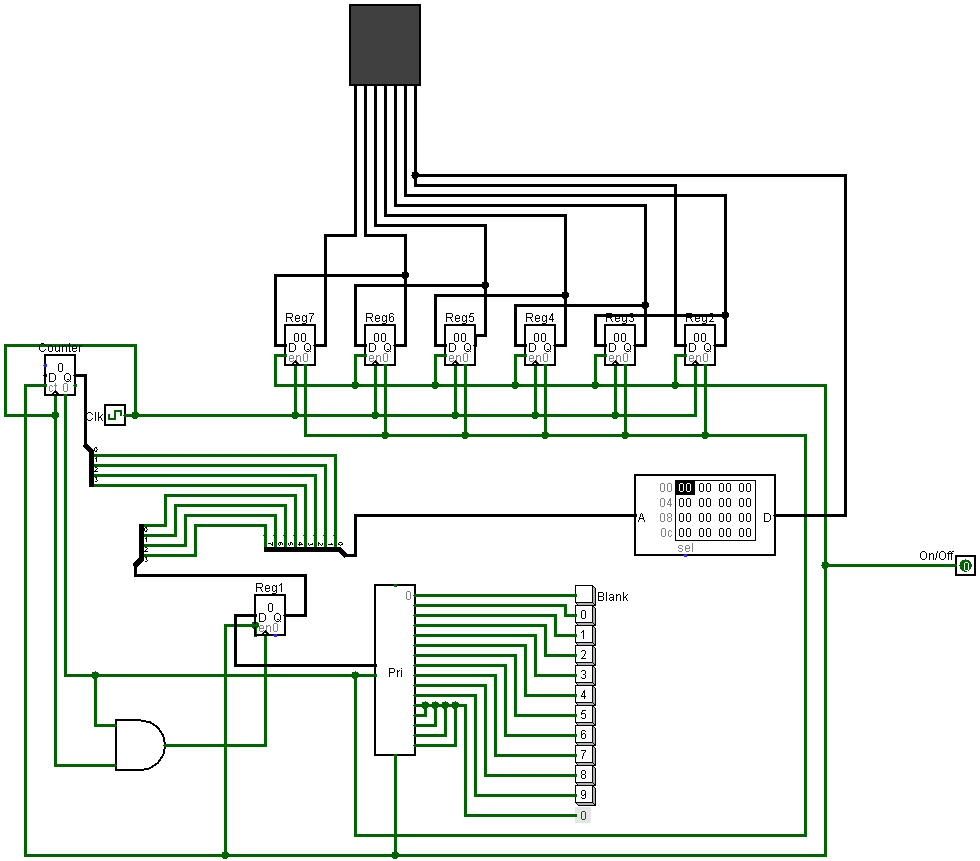
2. The AND gate helps in creating clock pulses to update the register value as in real life implementation, there is going to be a propagation delay and change in the input value might happen later than the occurrence of the positive edge of the clock caused by the button press.

3. The counter helps scroll through the ROM to get desirable 8-bit outputs (The ROM being pre-fed to give values according to input index)

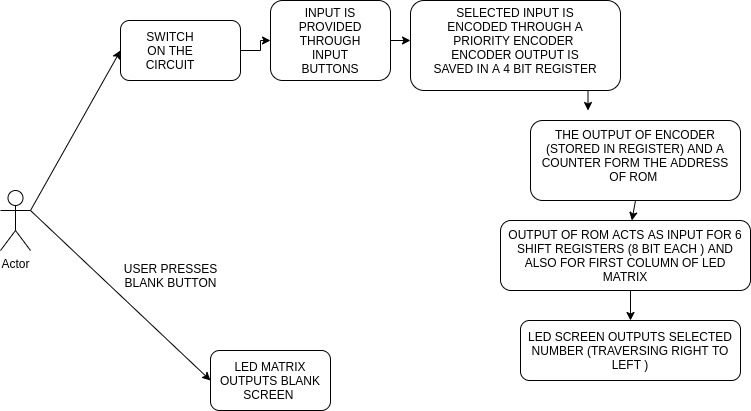
4. then the 8-bit shift register are used to load in the 8-bit values and cascaded to make a shift register of 8 bits - 8 stages.

**CHAPTER 4 LOGISIM IMPLEMENTATION**

**4.1 LOGISIM DIAGRAM**

****

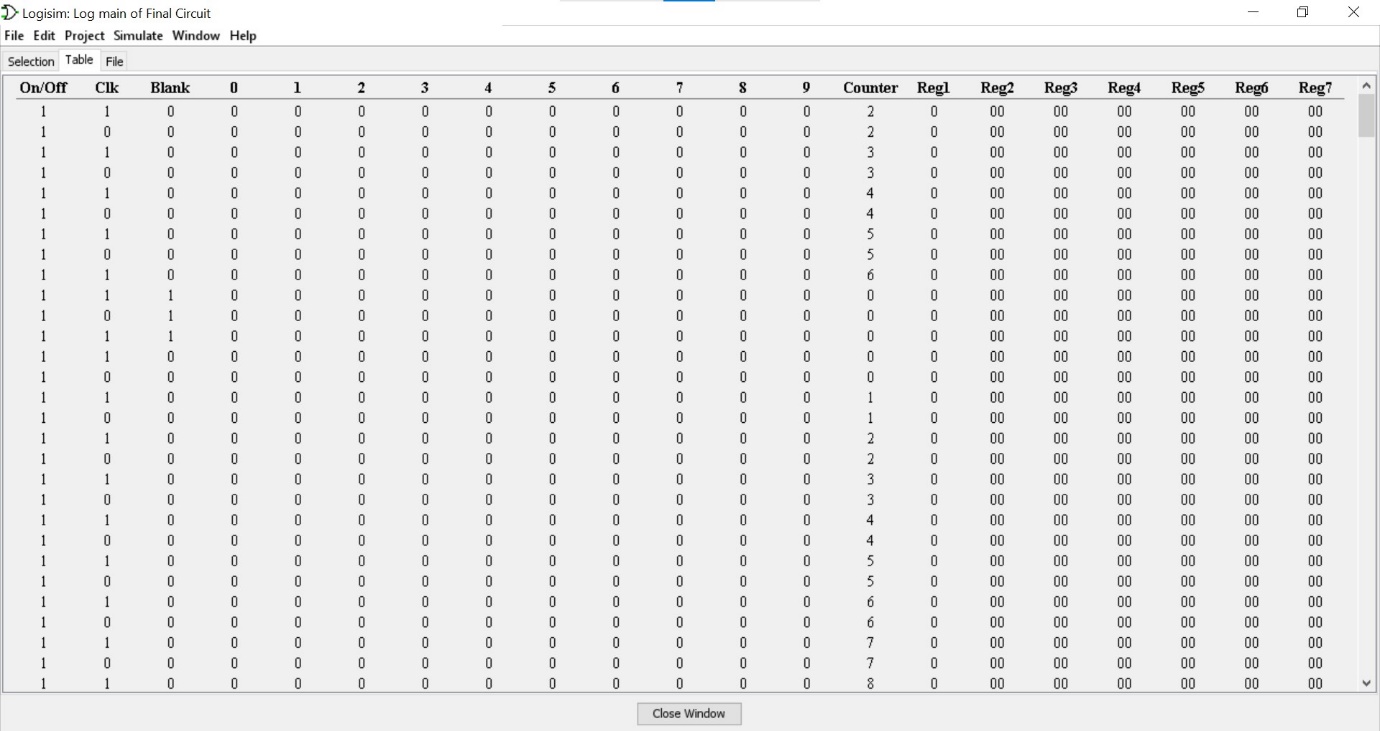
Clock is set at 8Hz.

****

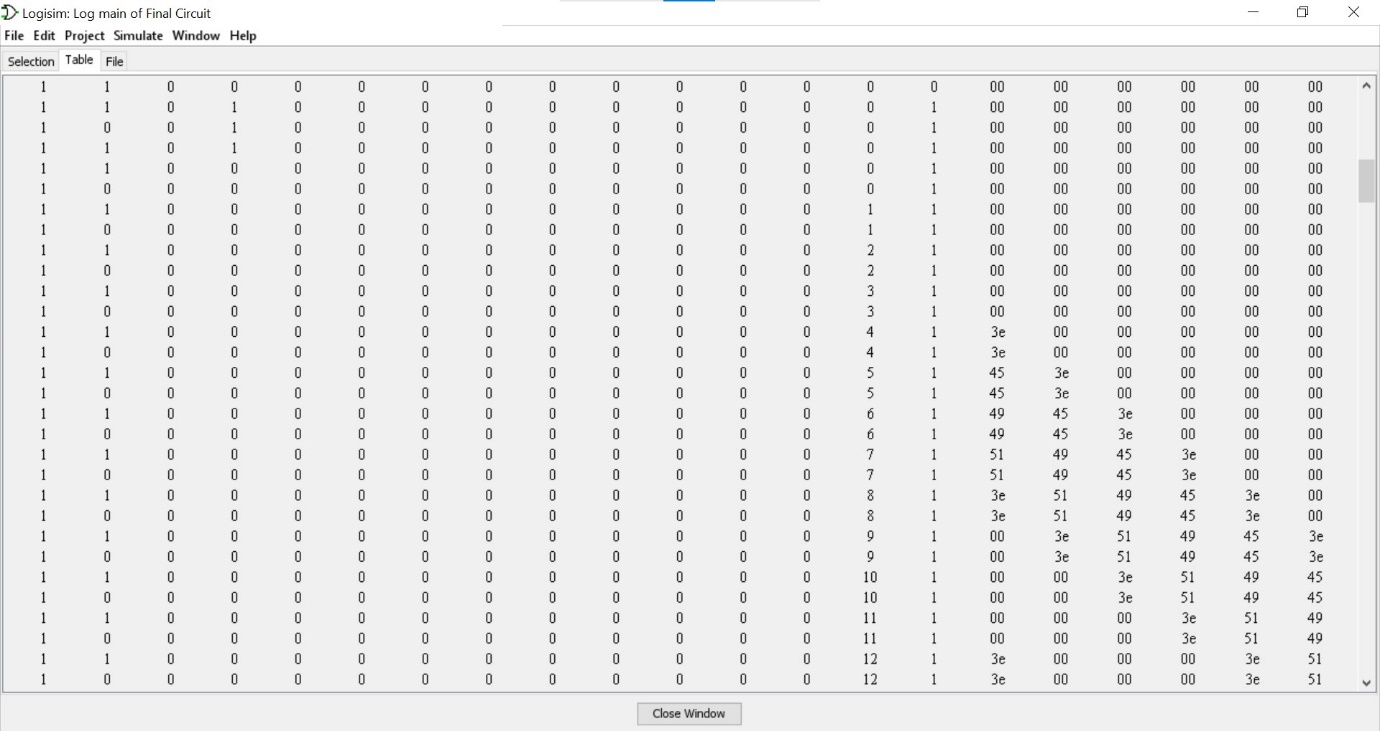
**FLOW OF THE CIRCUIT**

**4.2 SIMULATED OUTPUT**

**4.2.1 LOGGING TABLE ON PRESSING THE BUTTON BLANK**

****

**4.2.2 LOGGING TABLE ON PRESSING THE BUTTON 1**

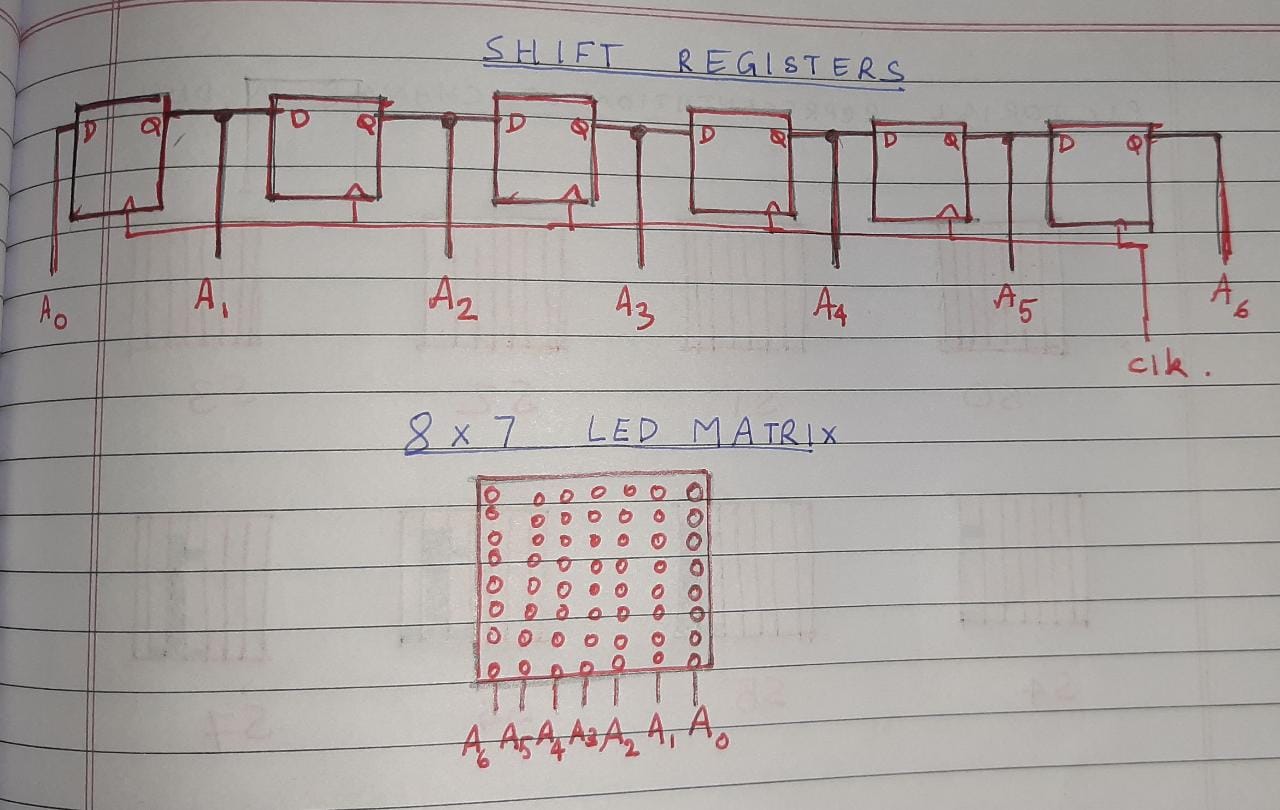


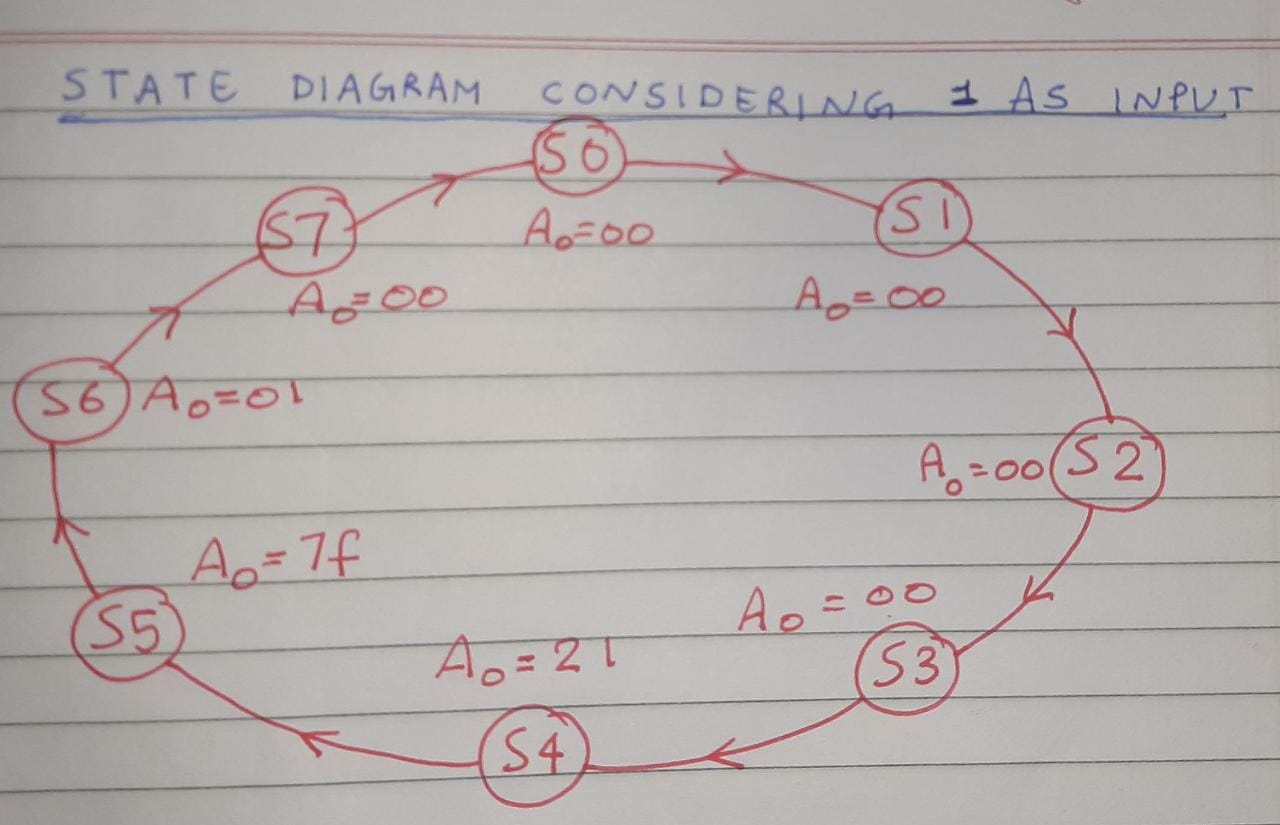
Radix of -

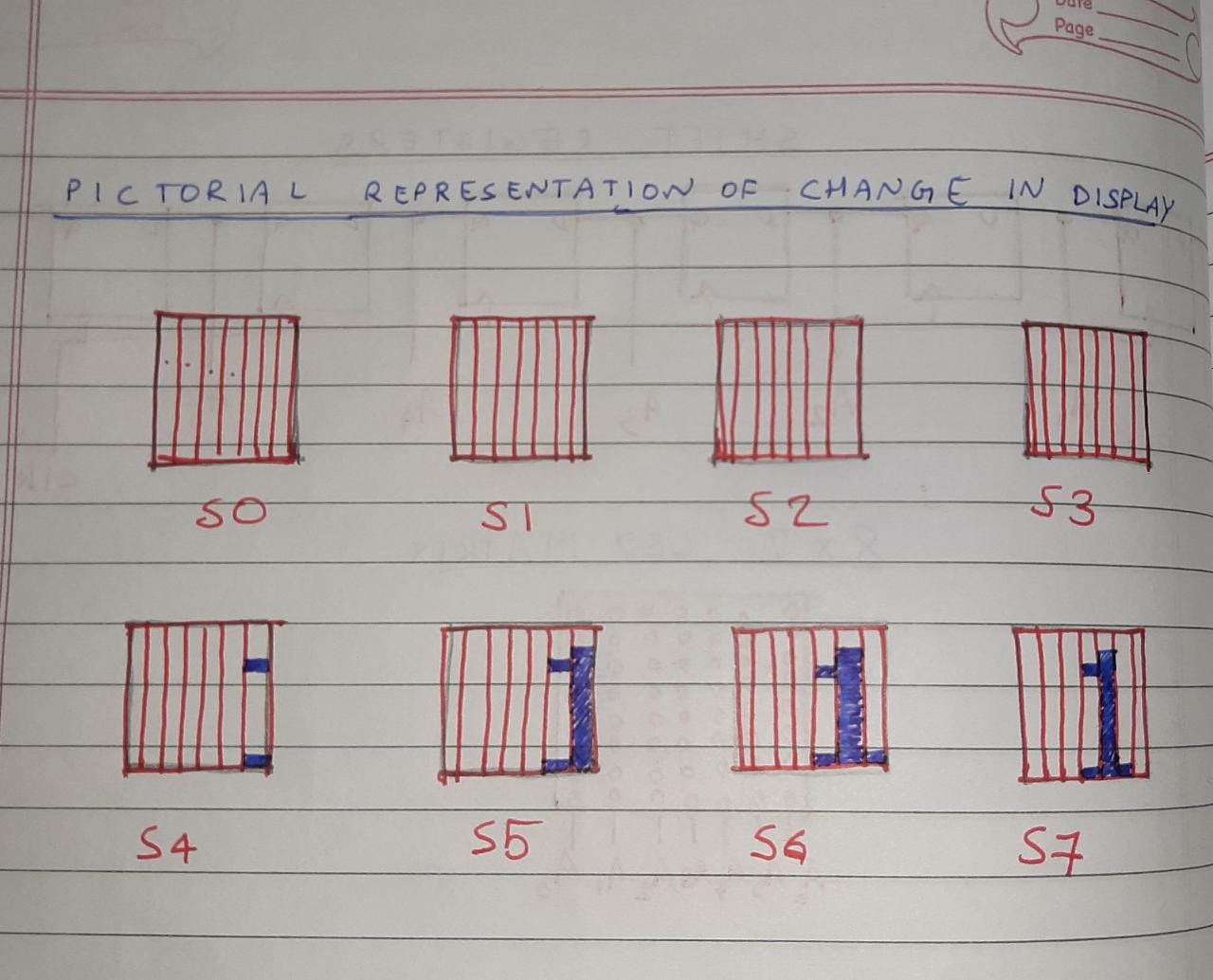
Buttons and Clock and On/Off – 2

Counter – 10

Register – 16

**CHAPTER 5 STATE DIAGRAM**

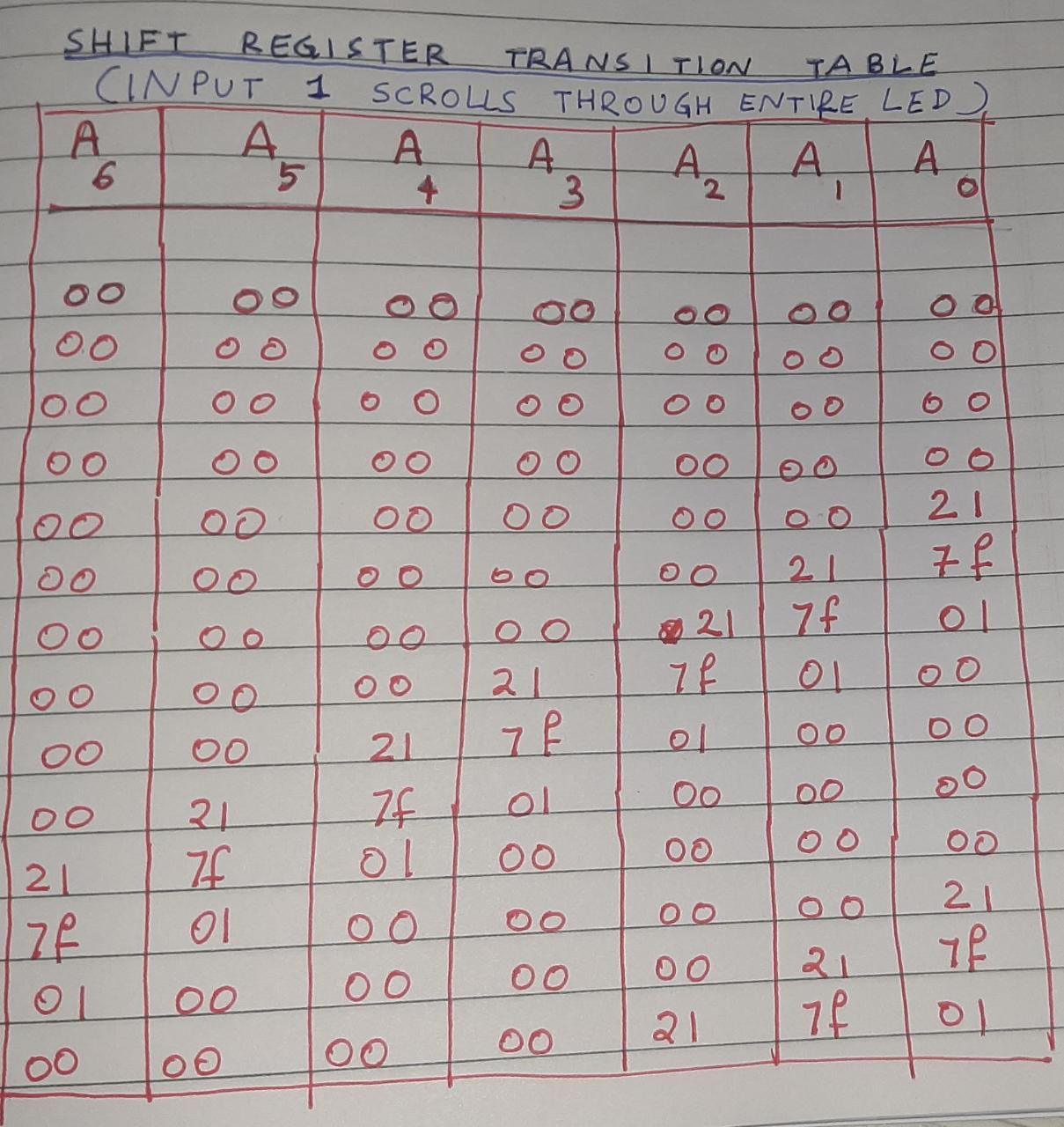
****

****

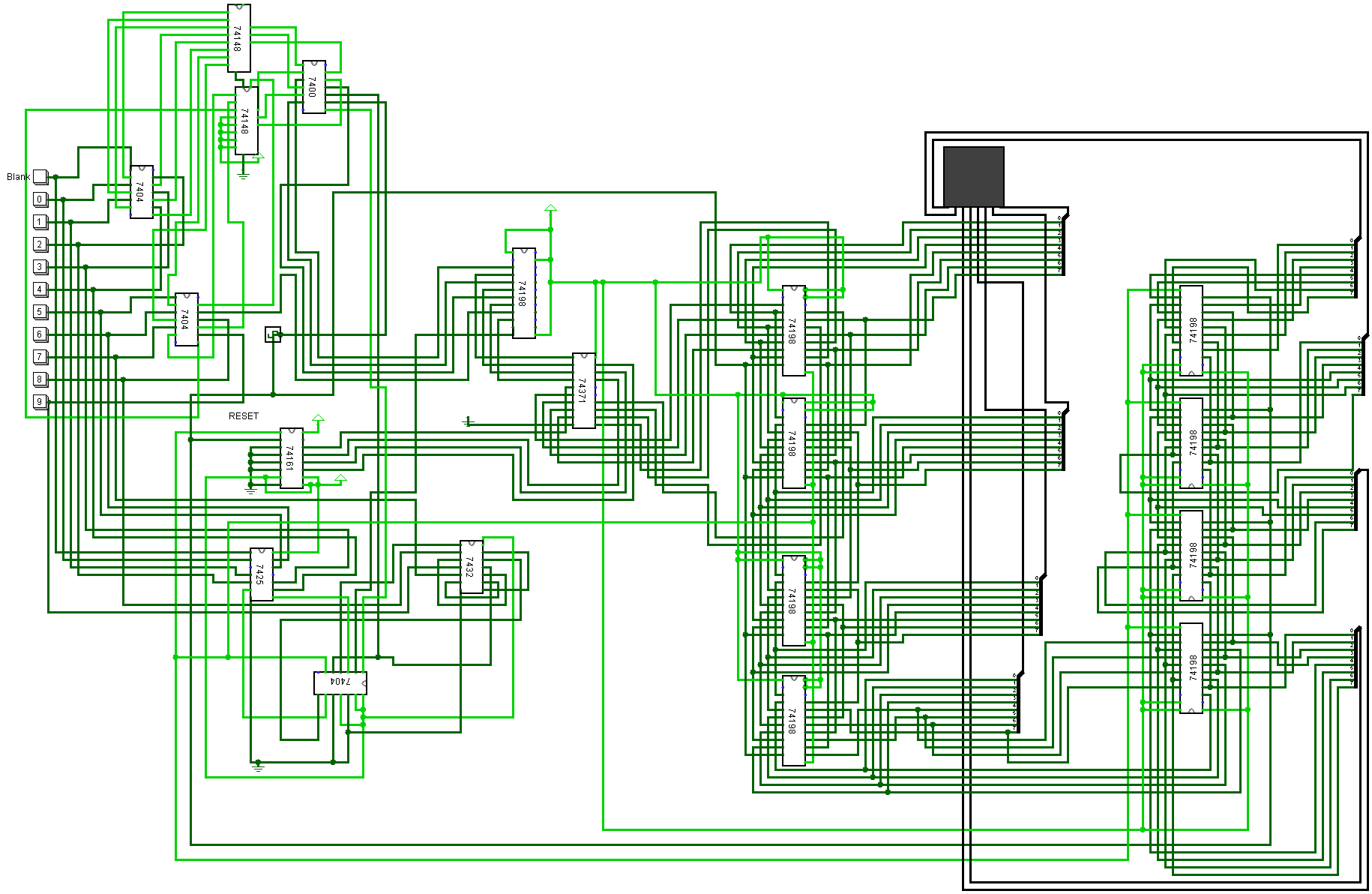
1. The inputs to A0 come from the ROM. Hence the transitions from one state to another occur due to the changing ROM address. The address, in turn, is set at a particular position by the input digit to display and transitioned from one address to another by a counter.

2. The register outputs are connected to each column of the led. The inputs are 8-bit numbers which specify the individual LEDs in a column to remain on/off.

**TRANSITION TABLE FOR REGISTERS**



**CHAPTER 6 IC IMPLEMENTATION**

****

The Logisim implementation in chapter 4 has been implemented here with ICs (Integrated circuits)

All the ICs have been given a common clock and power supply and have been grounded appropriately. The clock frequency is kept at 8Hz and when one of the buttons is pressed, the corresponding number scrolls across the LED matrix.

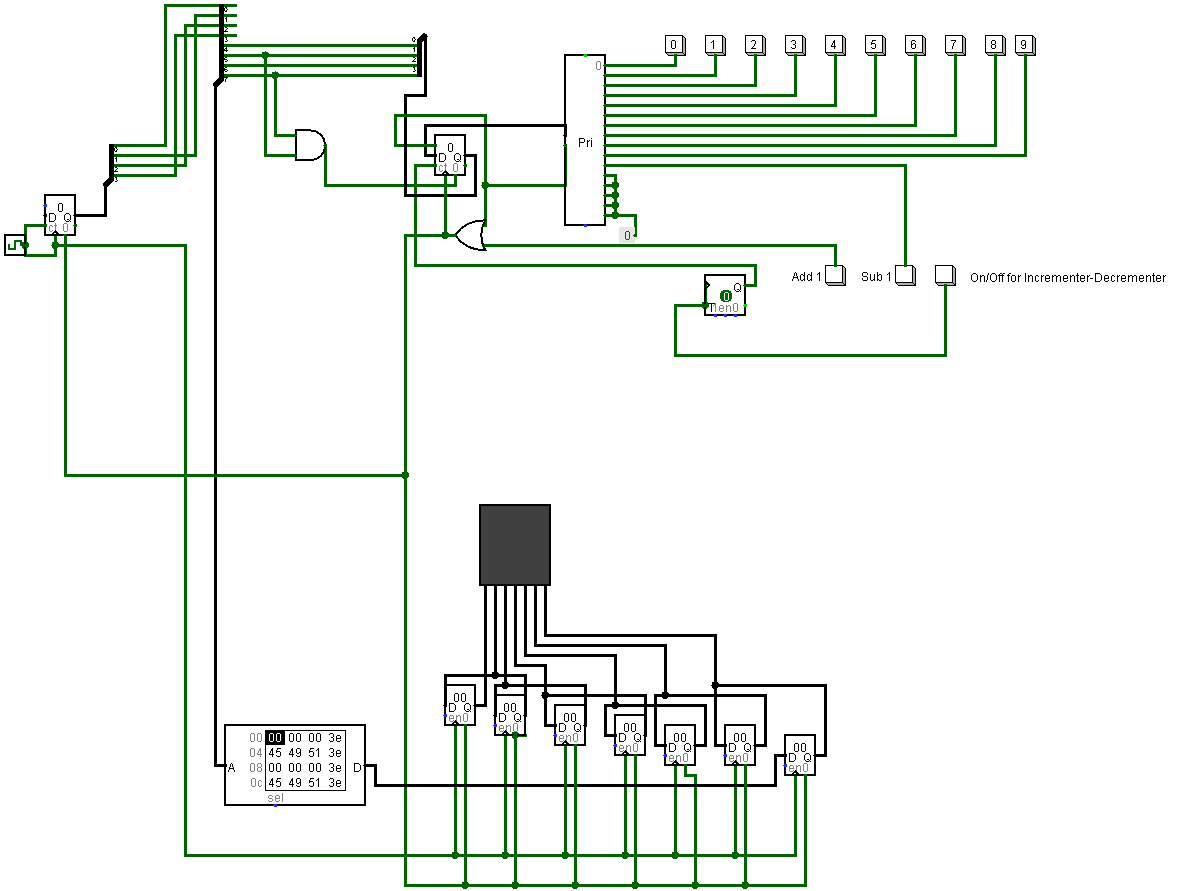
All the ICs are of the 7400 series and belong to the same TTL logic.

**CHAPTER 7 ADDITIONAL FUNCTIONALITY**

The features that our design has –

1. Every enable input has been taken care of to follow ‘Good design’ principles.
2. A single input/output switch has been given to ensure ease of use.
3. The number is made to repeat until another button is pressed. This ensures the kid keeps on seeing the number. Thus, ensuring better memory retention.
4. A priority encoder has been used to for the buttons, and its extra pins(not in use) are being fed the constant 0 thus again ensuring ‘Good design’ principles.
5. The output of the priority encoder has also been given as reset so that if another button is pressed while a number is still scrolling, the screen will become blank first and another number will then come up. If this weren’t ensured, the new number would’ve overlapped the old number.
6. We have used the component sizes which are available in the market, using standardised splitter sizes etc.
7. The ROM has the capacity to be extended to include other numbers.

To extend our circuit design even further, we included an **Incrementor/ Decrementor** also. The Logisim implementation for the same is as follows –

****

As additional functionality, we have implemented an incrementing button (‘Add 1’) that will increment the current number being shown on the LED by 1. Also, the order of number button on board will be shuffled. This can help children to learn counting more visually and interactively.

A decrementing button (‘Sub 1’) has also been added.

Example: if 1 is on screen, then pressing the incrementor button will display 2.

This additional functionality doesn’t increase the price or complexity of the circuit much and at the same time provides a more interactive interface to the user than standard implementation.

The method of operation is, 1. press any number button. 2. Press the On/Off incrementor/decrementor button. 3. Press Add 1 or Sub 1. To go back to number mode, press the On/off button again.

**CHAPTER 8 BILL OF MATERIALS**

1. 11 Button inputs
2. 8x8 LED matrix
3. 7400:

It is a Quad- 2 input NAND Gate. It has 8 input pins and 4 output pins and uses 4 two input NAND gates.

1. 7425:

It is a Dual 4-input NOR gate. It has 8 input pins and 2 output pins and uses two 4 input NOR gates.

1. 7432:

It is a Quad 2-input OR gate. It has 8 input pins and 4 output pins and uses 4 two input OR gates.

1. 74161:

It is a synchronous 4-bit counter. It has 4 input pins for data and one each for clock, clear and enable inputs and 4 data outputs.

1. 74148:

It is a 8-3 line Priority Encoder. It has 8 inputs and 3 outputs all of which are active low.

1. 74198:

It is an 8-bit Shift Register. It has two select inputs, 4 data inputs, one clock input and has 4 data outputs.

1. 74371:

It is 2048 bits sized ROM.

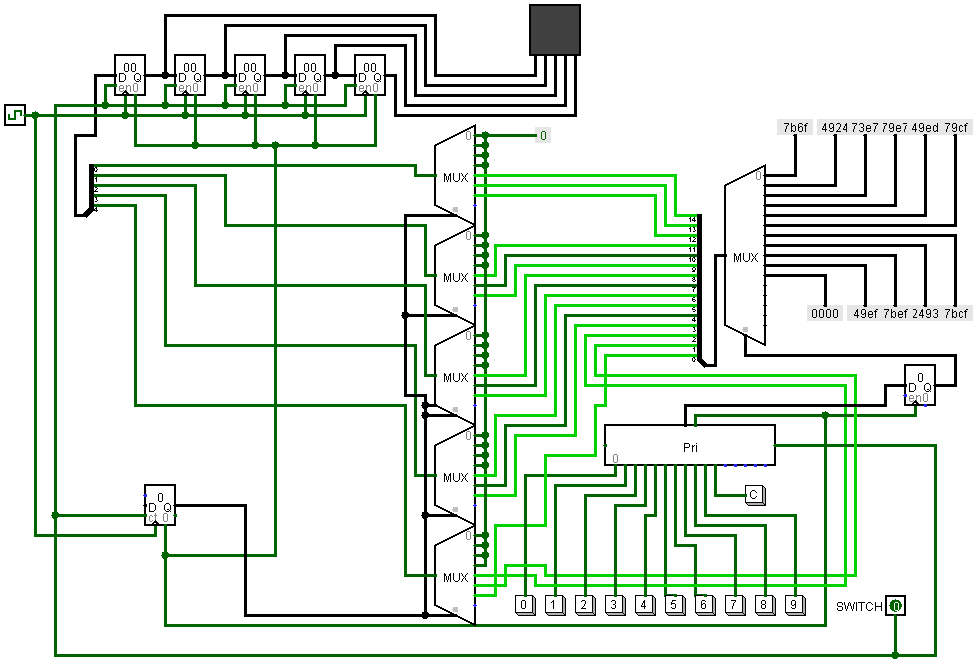
1. 7404:

It is a Hex Inverter. It has 6 input pins and 6 output pins and contains 6 NOT gates.

Our implementation uses nine 74198 ICs, two 74148 ICs, three 7404 ICs, one 7400 IC, one 74371 IC, one 74161 IC, one 7425 IC and one 7432 IC.

**CHAPTER 9 COMPARISON BETWEEN ROM AND WITHOUT ROM IMPLEMENTATION**

Since, ROM costs are a little higher and the circuit implementation could be done without it, we went one step further and did a Logisim implementation without using the ROM element.

****

This implementation uses a series of multiplexers to obtain the same output and functionality as of the given topic.

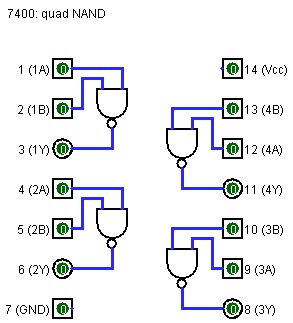
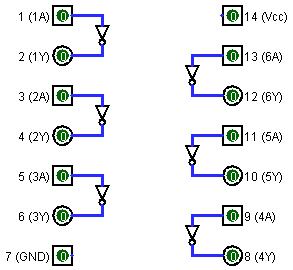
The cost of multiplexers instead of ROM brings down the cost of circuit. Thus, decreasing the cost of the toy overall.

However, we went ahead with the ROM implementation for our project as the empty space in it allows us to increase the numbers beyond 9.

**CHAPTER 10 APPENDIX**

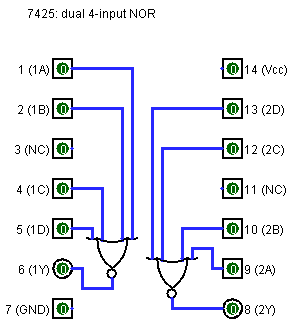
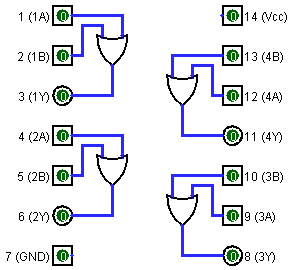
The Datasheets are including in the zip folder as PDFs.

Here, we’re including the Logisim implementation of the ICs used.

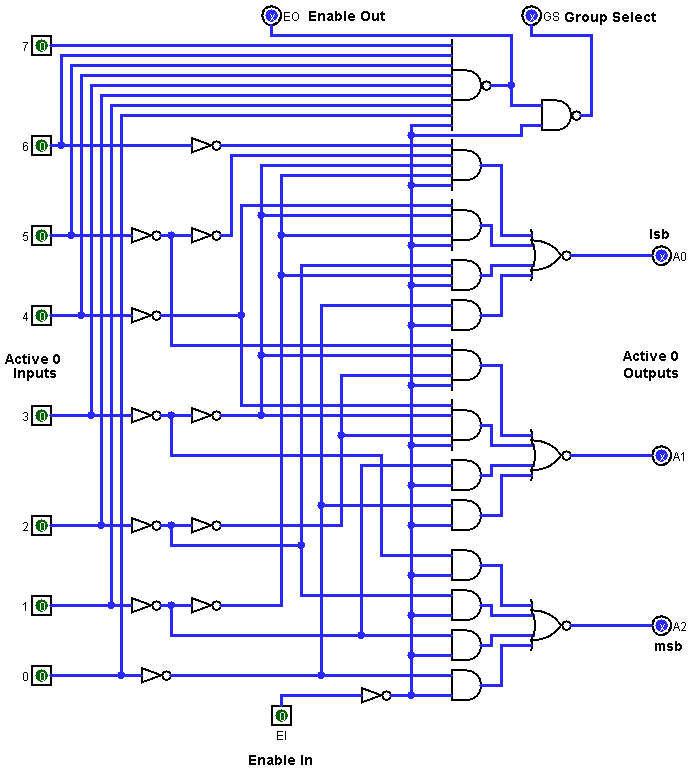
 

7404

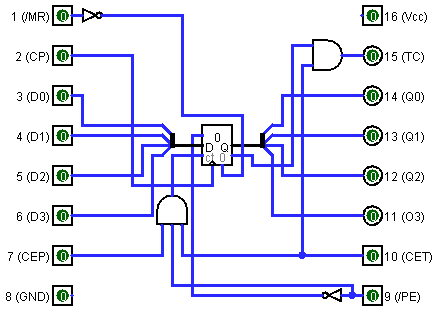
7432

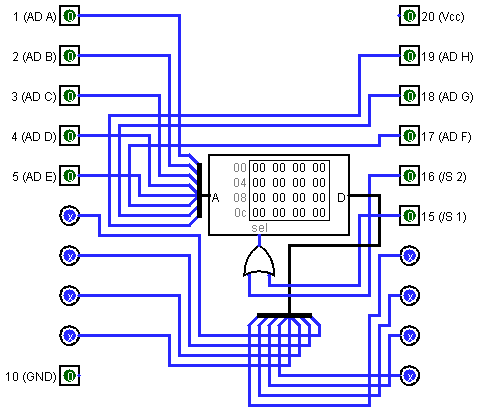
74148

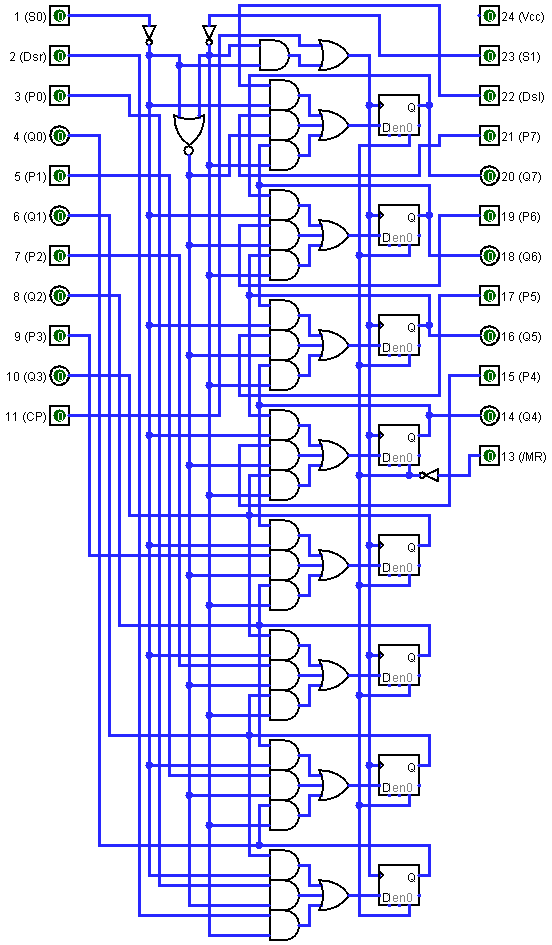


74161

****

74371

****

****

74198

**CHAPTER 11 LIST OF CONTENTS IN ZIP FOLDER**

1. Block Diagram (Block Diagram.jpg)

2. Block Diagram Template posted on Canvas (System\_Block\_Design.pdf)

3. Logisim Implementation circuit file for circuit in chapter 4. (Logisim Implementation.circ)

4. Logisim Implementation image file for circuit in chapter 4. (Logisim Implementation.png)

5. Video demo for Logisim Implementation (Demo Logisim Implementation.mkv)

6. Flow of circuit shown in chapter 4 (Flow of circuit.jpeg)

7. Logging Table data (Logging Table.TXT)

8. Logging Table for Blank in chapter 4.2.1 (Simulation for 1.jpg)

9. Logging Table for 1 in chapter 4.2.2 (Simulation for Blank.jpg)

10. ROM data for Logisim Implementation in chapter 4. (ROM data for Logisim implementation.TXT)

11. IC Implementation circuit file for circuit in chapter 6. (IC Implementation.circ)

12. IC Implementation image file for circuit in chapter 6. (IC Implementation.png)

13. ROM data for IC Implementation in chapter 6. (ROM data for IC implementation.TXT)

14. Video demo for IC Implementation (Demo IC Implementation.mkv)

15. Incrementor – Decrementor circuit file for circuit in chapter 7. (Incrementor-Decrementor.circ)

16. Incrementor – Decrementor image file for circuit in chapter 7. (Incrementor-Decrementor.png)

17. Video demo for Incrementor - Decrementor Implementation (Demo Incrementor Decrementor.mkv)

18. Without ROM circuit file for circuit in chapter 9. (WithoutROM.circ)

19. Without ROM image file for circuit in chapter 9. (WithoutROM.png)

20. Video demo for Without ROM Implementation (Demo Without ROM.mkv)

21. IC Logisim Implementation Images in Chapter 10. (IC Logisim Implementations Images)

22. Datasheets in PDF form for all the 8 ICs used (Datasheets PDFs)

23. This Report. (Report.docx)

**THANKYOU!**