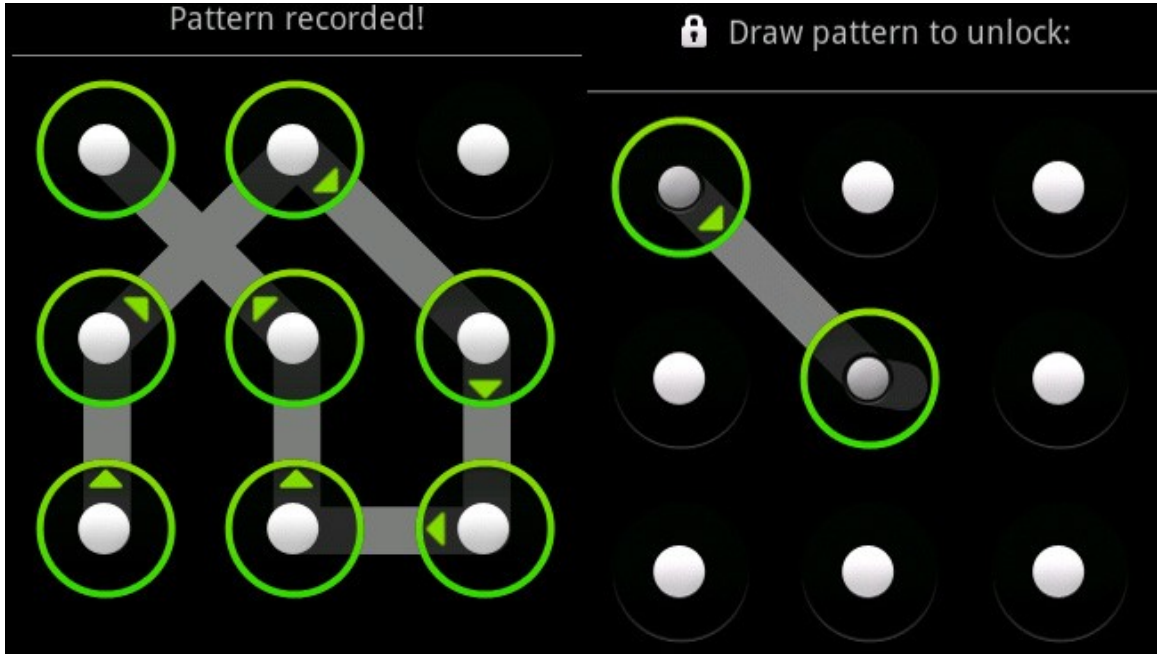
DDS PROJECT - LOGISIM

ndroid pattern lock

Presented by
Chirag Jamadagni 13CO117
Kevin Thomas Mathew 13CO224

Aim:

- To implement an android pattern lock with digital systems.
(android pattern animation)



Analysis:

1. DISPLAY

- Display using an LED matrix the pattern being entered by the user.
- Just like an android lock, nine buttons are used to make any possible pattern.
- Instruction to make pattern:
- -While making the pattern lines cannot be drawn to a button that is not adjacent to the present button.
- -Buttons cannot be repeated. Press clear when done.

Analysis

2. LOCK

- Save the pattern-The entered pattern needs to be stored in memory, in-order to be compared with later for unlocking purpose.
- To UNLOCK, pattern is entered. If entered pattern matches, unlock appears on the TTY output display.
- CLEAR: The led display is cleared and input circuit.
- CLEAR existing pattern: This clears the saved pattern and it only works in the unlocked state.
- SET button is provided to save a new pattern for the lock. Pattern can be set only after clearing the existing saved pattern if present.

Components Required

- AND gates
- D Flip Flops
- Priority Encoder
- Priority Decoder
- OR gates
- XOR gates
- LED matrices

Components Required

- Registers
- 4 bit Comparator
- Input buttons
- Wires
- NOT gates
- 1 kHz Clock pulse
- Multiplexer
- Demux
- TTY

Components and their functionality

- LED matrices: The 4 LED matrices are used to display the pattern made by the user, this allows for the user to take note of what pattern has been set before clearing the circuit. It also serves as a way to make sure that invalid patterns are not accepted while saving a pattern
- D flip flops: They are used in both the LED display and to make shift registers. In the LED display they save the inputs.

Components and their functionality

- TTY box: The TTY box is used to display text. In this circuit, it is used to show whether the circuit is in the locked or unlocked state. This is achieved by connecting the box to an ASCII generating circuit.
- 4 bit comparator: They are used in the circuit to determine if the input pattern matches the saved pattern, their inputs are encoder outputs. We use 9 4-bit comparators in the comparison circuit, one for each button of the pattern.

Components and their functionality

- Priority Encoder and Decoder: They are used to avoid unwanted patterns by filtering out the two most recent entries. Impossible/Unwanted patterns include ones where the third of one row is succeeded by the first of the next row, we can see that such patterns are not possible in the original android pattern lock.

Design

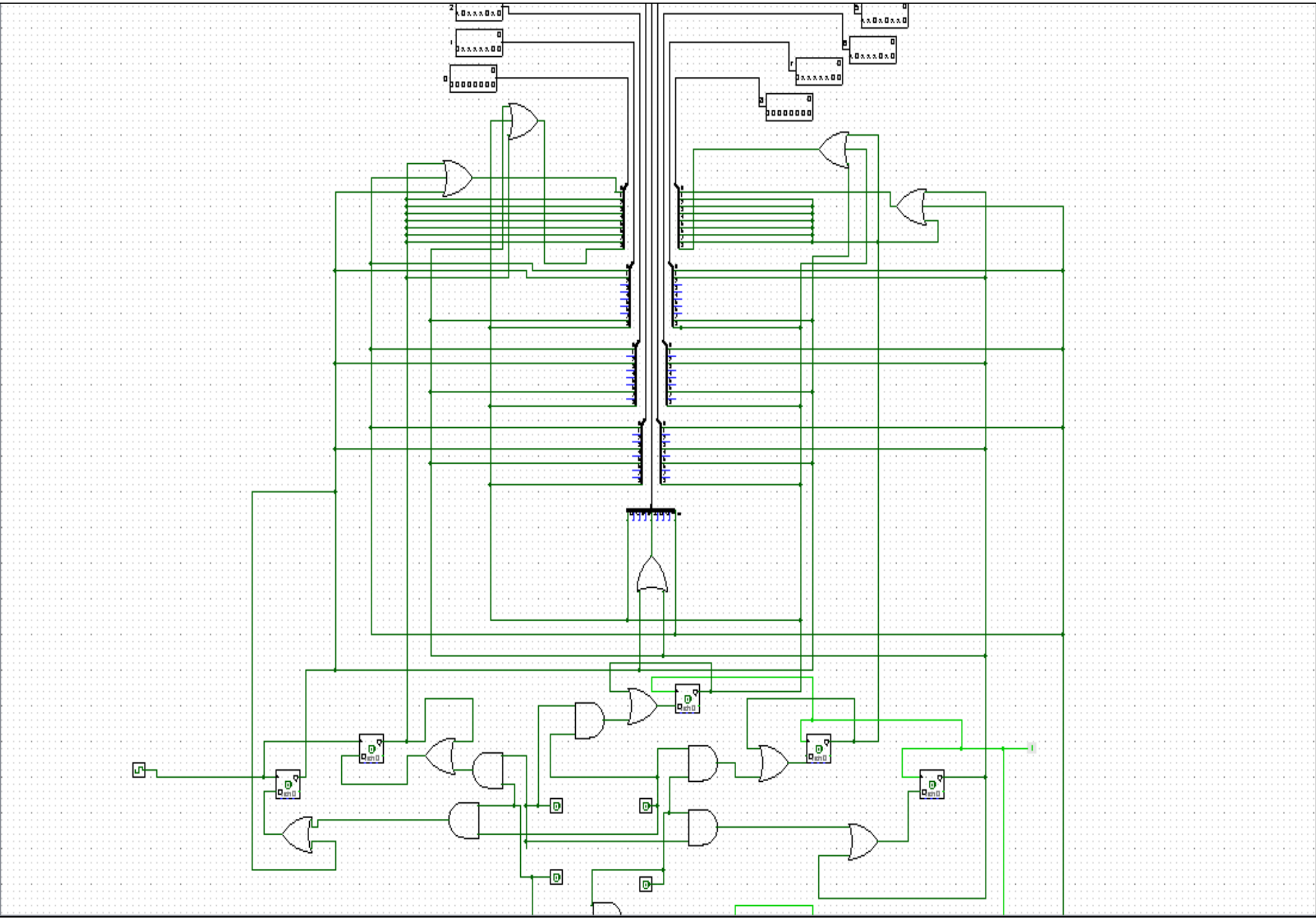
DISPLAY

- Four 9x9 LED matrices used to display the pattern formed by the nine buttons.
- A single LED circuit consists of:

AND gate : for choosing the vertices between which the edge is to be made in the LED

D-flipflop is used to save the input. The output is OR'd and inputed back in the flipflop so that saved input stays.

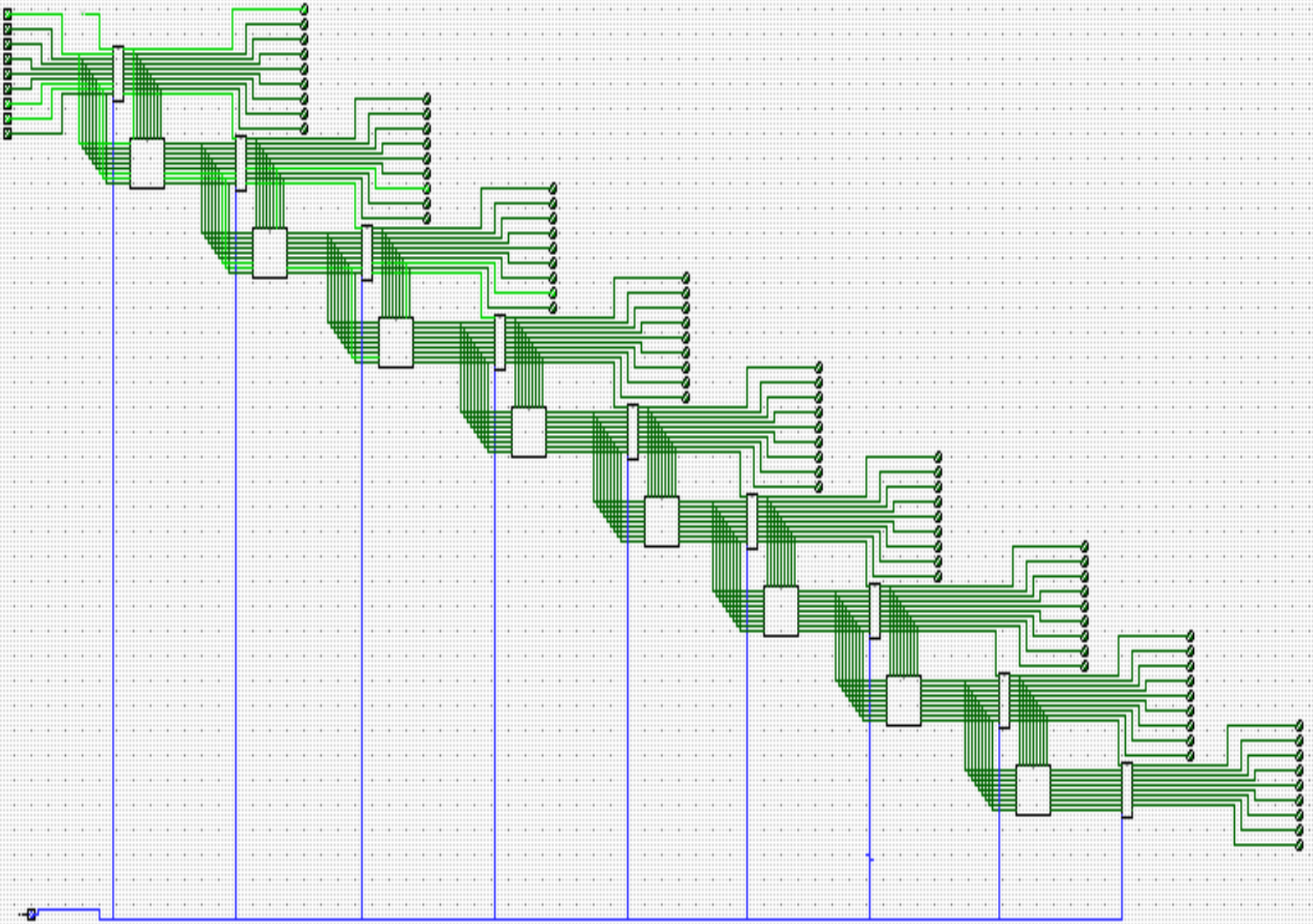
The output from the flipflop is then inputed into the LED inputs according to the grid that is to be lit.



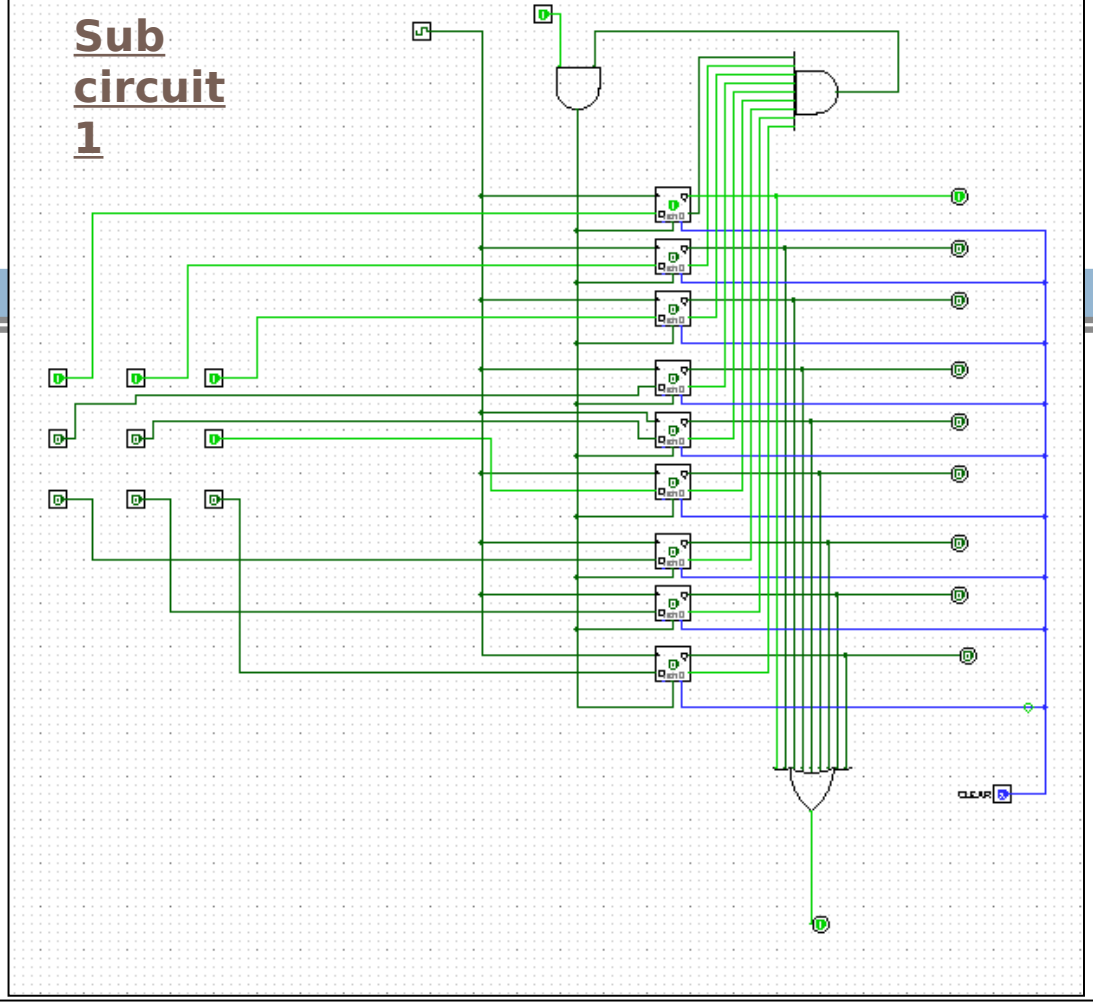
Design

To save input pattern:

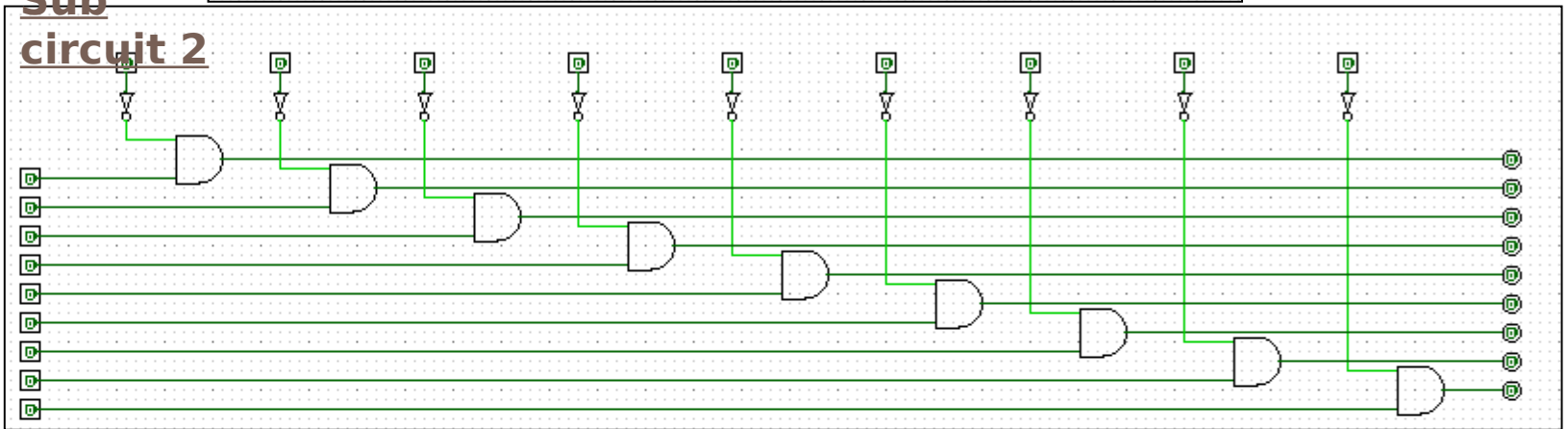
Since the same input device is used to make the entries, a circuit is created to separate out the entries in the entered pattern/order.



Sub circuit 1

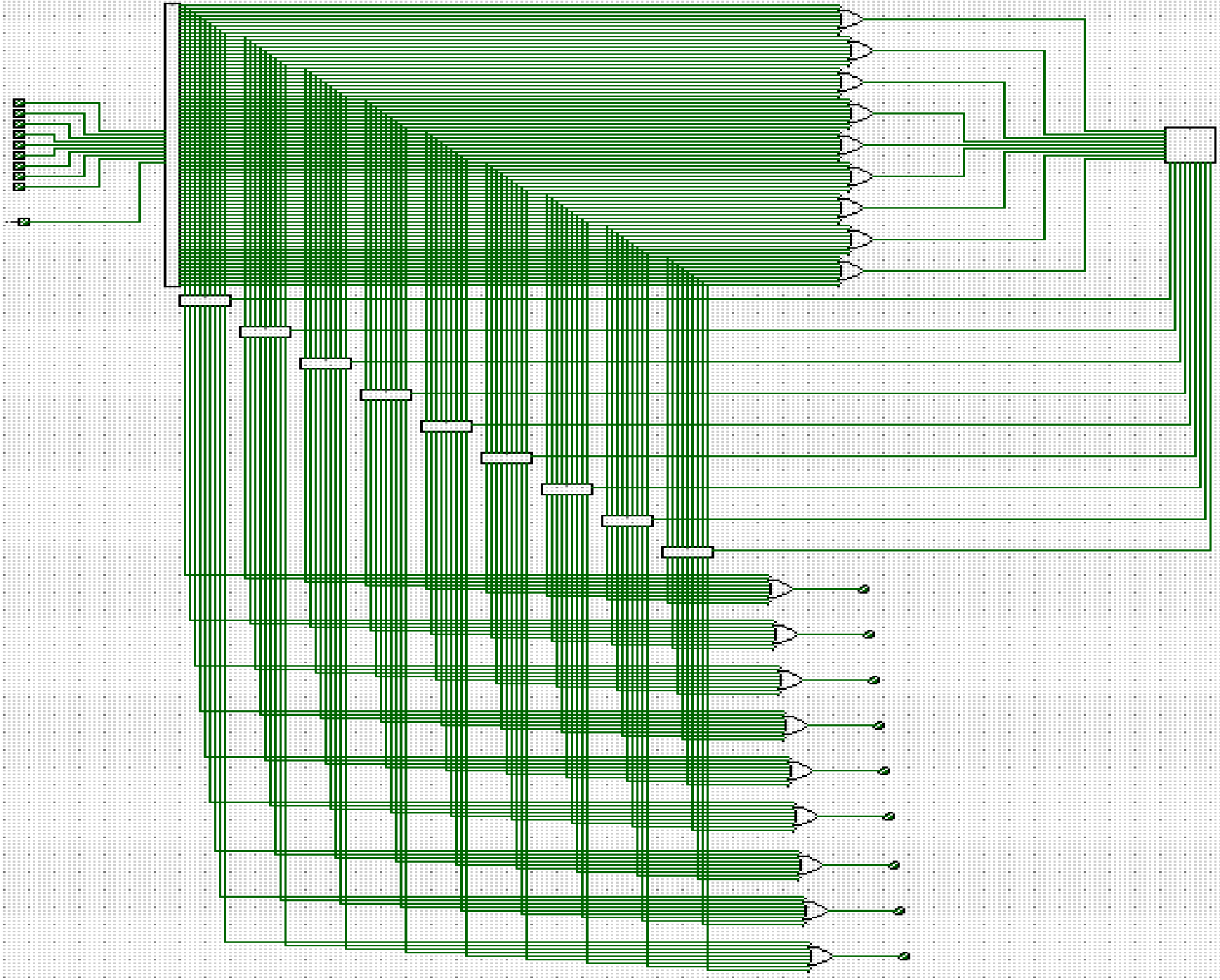


Sub circuit 2



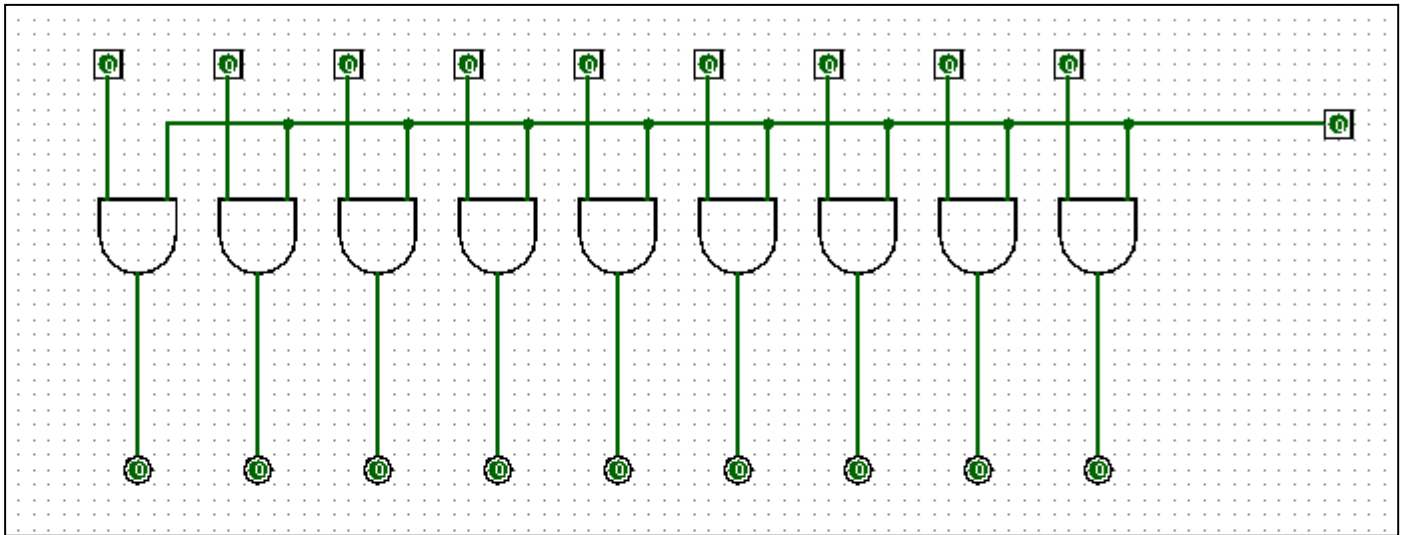
Design

- **Avoid unwanted patterns:**
- When the third button is switched on while the first and second are still on, unwanted pattern appear. To avoid this a sub-circuit to switch off the first button when a third is on is designed.
- This circuit is made using the following:
- Priority encoders and decoders are used to filter out the two most recent entries.
  The first priority encoder gives the most recent entry. This is decoded and NOT'd with all the inputs so that the input with second most priority comes out of the second priority encoder.
- This circuit is added before the inputs are passed into the led circuit.



Sub circuit 1

Sub circuit 2

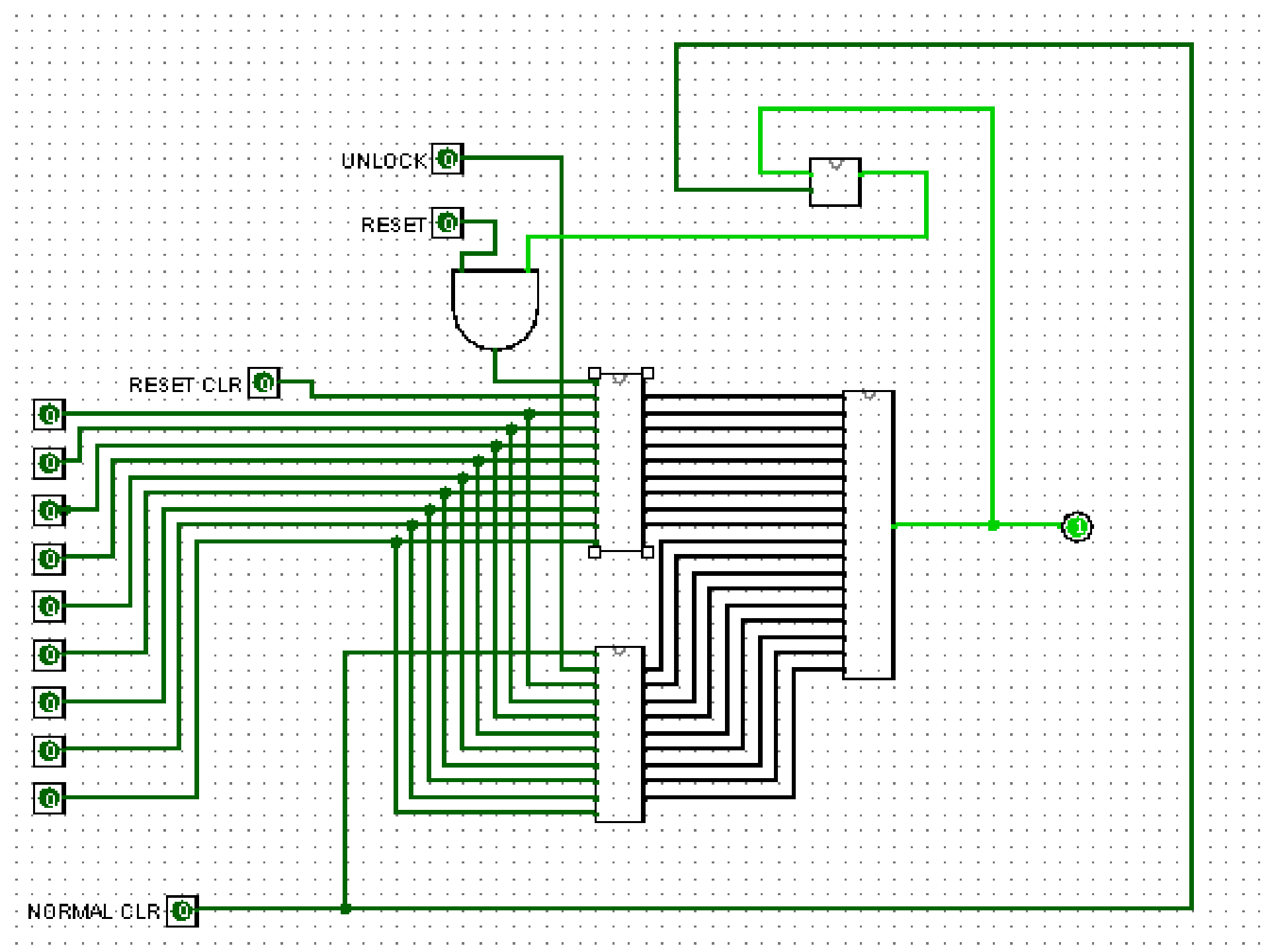


Design

- To match the input pattern with the saved pattern inorder to unlock

-The inputs are passed into an ENCODER whose output is then passed into a nine 4-bit comparators.

-The output from the comparator is AND'd with RESET and inputed into the enable of the saved pattern circuit. This is done so that the pattern can be reset only in the unlocked state.


UNLOCK
RESET
RESET CLR
NORMAL CLR

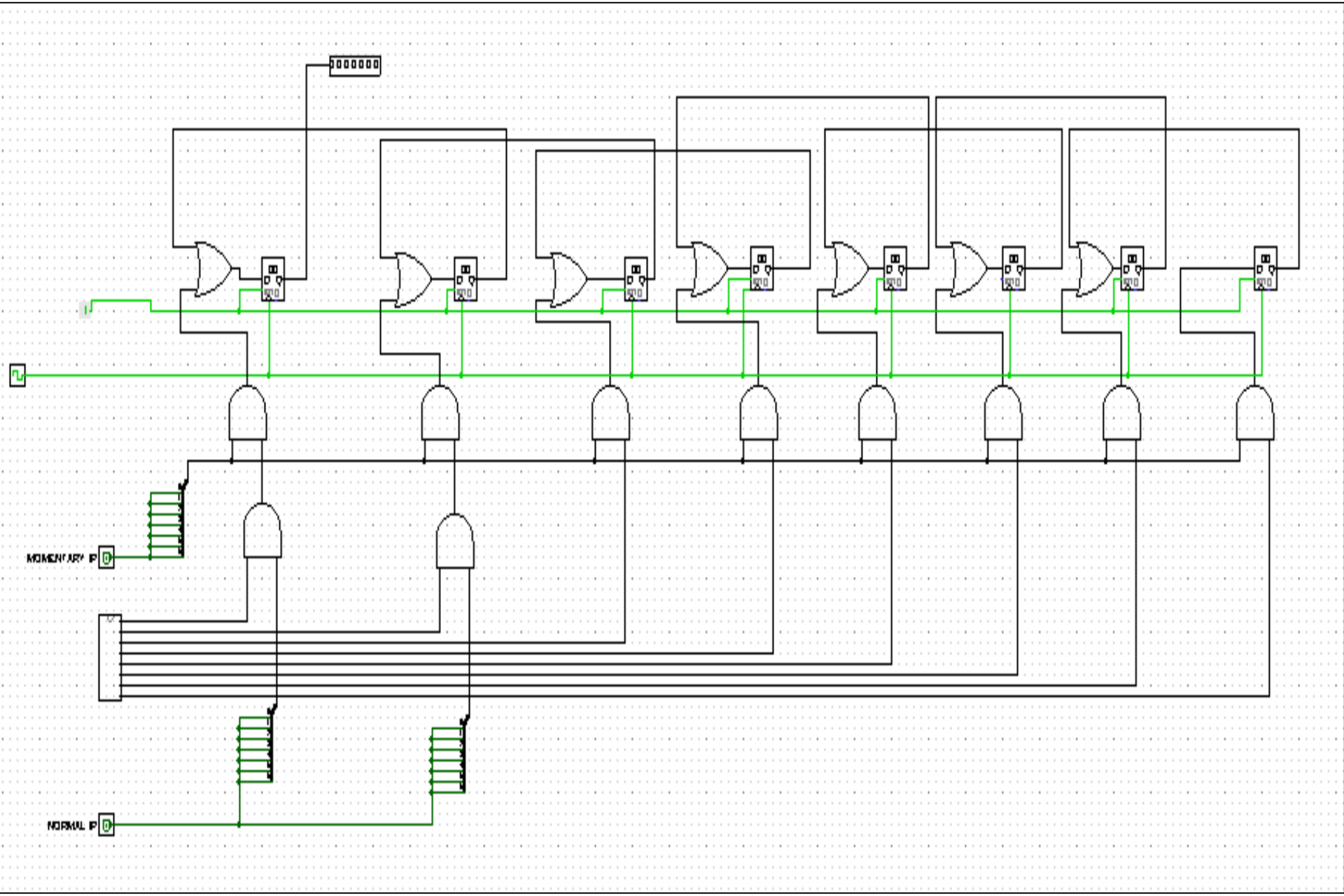
Design

- To Display final message:

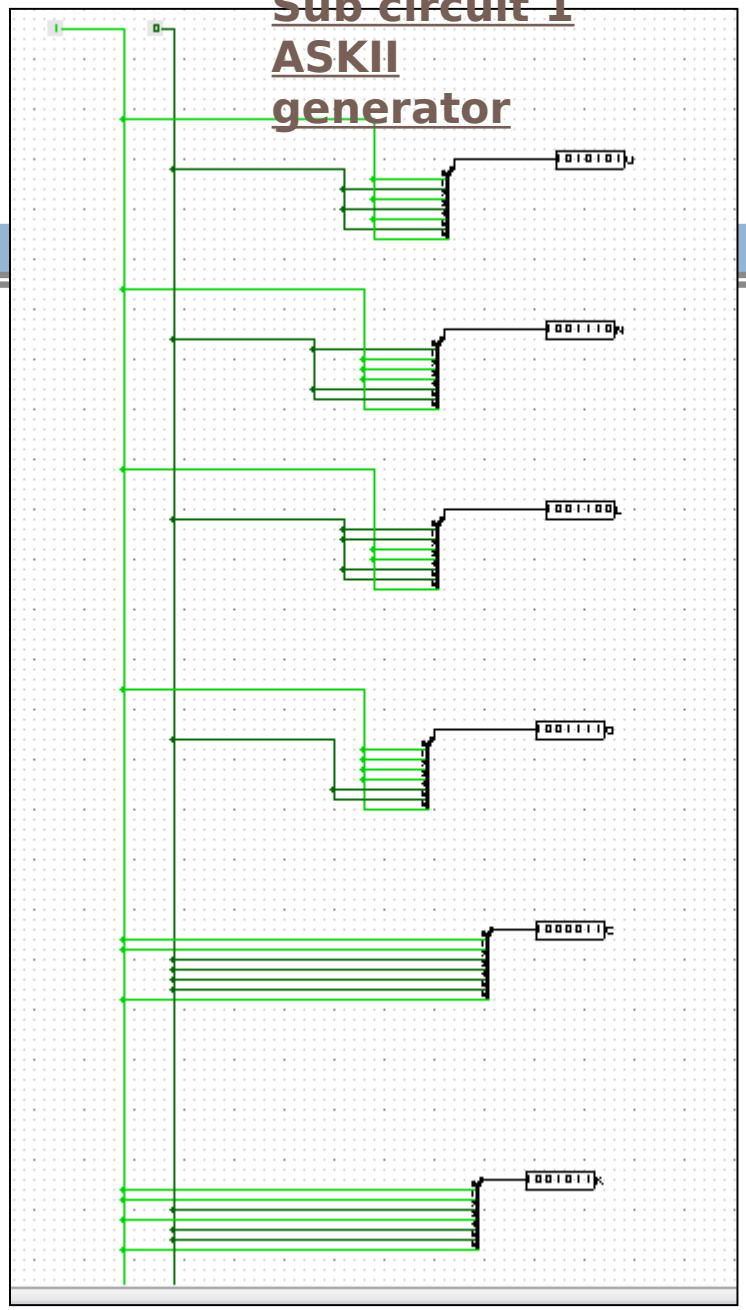
-A circuit that generates ASCII values of ‘U’, ‘N’, ‘L’, ‘O’, ‘C’, ‘K’, ‘E’, ‘D’ is designed.

-This circuit is connected to a circuit of shift register made from D-flip flops.

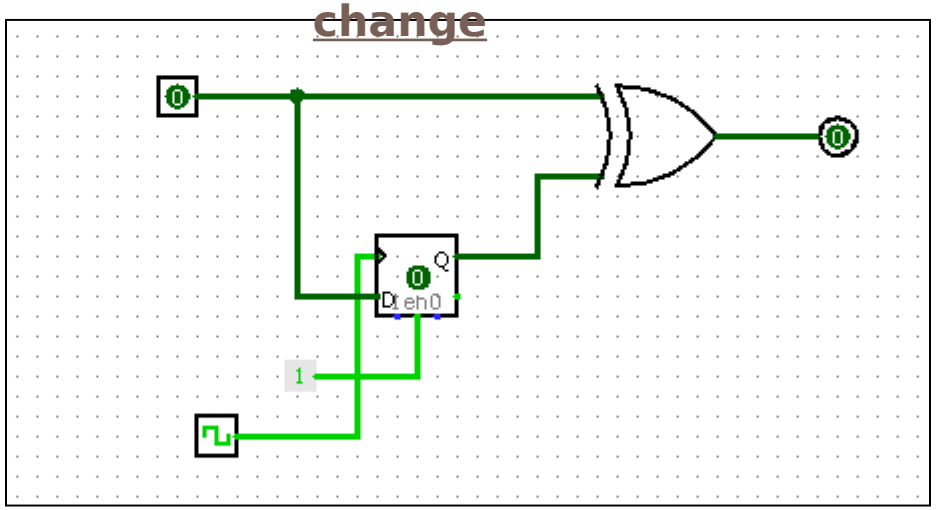
-A circuit that detects a change in the input is added because the stored letters need to be displayed only once on the TTY when output is constant and to clear the previous message shown on the TTY.


MOMENTARY IP
NORMAL IP

Sub circuit 1 ASKII generator



Sub circuit 2: For momentary change

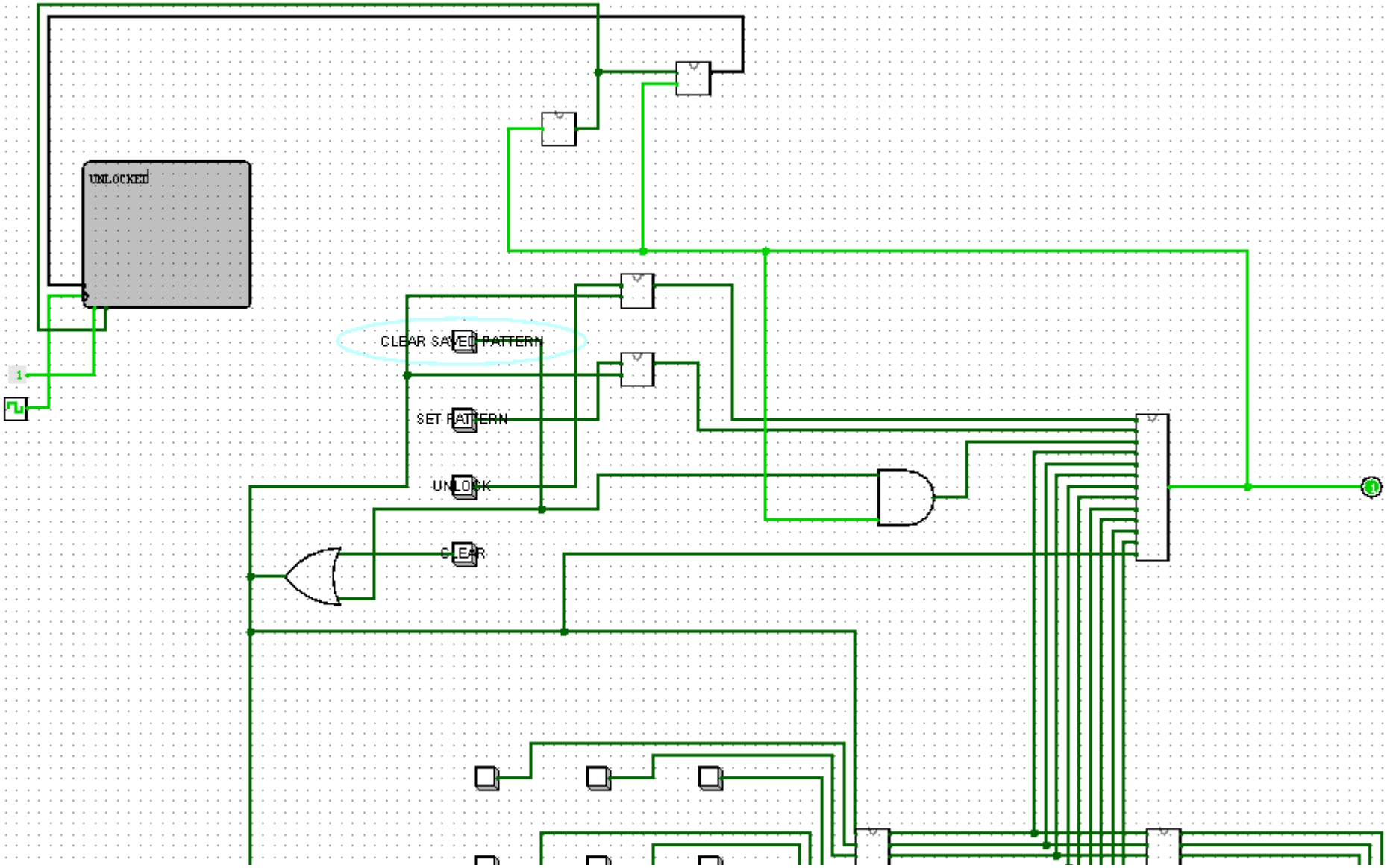


Design

- Clear saved pattern : This button allows the user to clear the pattern currently saved in the circuit. This button only works when the circuit is in the unlocked state.
- Set pattern : This button is used when the circuit is unlocked and a pattern is to be set, this is used only after the previously saved pattern is cleared.

Design

- Unlock: This button is used only after the correct pattern is entered while the circuit is in the locked state to unlock the same.
- Clear: Last but not the least, the clear button allows the user to clear the circuit and the LED matrices so as to make sure that it is not possible to see the saved unlock pattern.


UNLOCKED
1
CLEAR SAVED PATTERN
SET PATTERN
UNLOCK
CLEAR

Implementation

- Clock has been set at 1kHz to avoid the lag in the response by the sub-circuits.
- Initially the circuit is in the 'unlocked' state. The 'set pattern' button is used to save the pattern.

Limitations

- If clock pulse is less than 1kHz there is a significant lag in the circuit
- There is no alternate way to unlock. In the actual android pattern lock there is a way to unlock the phone even if the pattern is forgotten. Our implementation doesn't satisfy this requirement.
- A pattern lock is supposed to be a continuous smooth continuous motion. Even if this circuit is used on a touch screen laptop, the user is required to click each button.

Applications

- Used in all Android based smart phones.
- Used in mobile applications where privacy is a requirement. App's like 'Photo lock' uses the same 3x3 pattern.

Conclusion

- A simple pattern lock used by all android users on an everyday basis can be implemented using digital systems using a combination of gates, plexers and other circuitry.