Mini Project Report

Title :- Design and Implementation of Sequence Detector 1101 using the concept of Finite State Machines.

Objective :- Designing a working model of a Finite State Machine Application 1101 sequence detector using required integrated circuits on breadboard and on Printed Circuit Board.

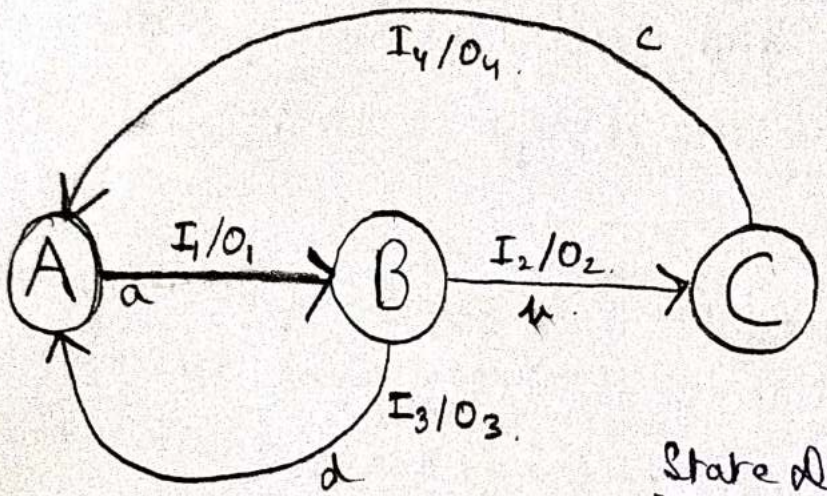
Literature Survey :- Sequence detectors are used in real-world applications such as in digital ~~comman~~ communication to recognise patterns, data compression and protocol implementation. They can also be used in emerging trends such as machine-learning, reconfigurable hardware implementation (eg, FPGA-based detectors) and application-specific optimizations. These detectors work on the principle of finite state machines.

→ Finite state machines.

They are computational models used to describe the behaviour of systems with a finite number of states. The are widely used in various applications, including digital circuit design, control systems and communication protocols.

Here's a breakdown of the key components and concepts of finite state machines:-

Components of Finite State Machines.



State Diagram

States :- A, B, C.

Transitions :- The arrow marks show transition between states (a, b, c, d)

Input :- External stimuli which decides where transition happens (I_1, I_2, I_3, I_4)

Output :- Response of system in each transition (O_1, O_2, O_3, O

1. States :- FSMs consist of a finite set of states, which represent different conditions or configurations of the system. Each state represents a distinct mode of operation or behaviour.

2. Transitions :- They define movement of the system from one state to another in response to input signals or events. They are triggered by input symbols or conditions and are associated with specific state changes.

3. Inputs :- They are external stimuli or signals that trigger state transitions in the FSM.

4. Outputs :- Outputs are generated by the FSM in response to input signals and current states. It represents the system's action corresponding to its current state and input.

5. Transition Function :- It defines the behaviour of FSM by specifying the next state for each possible combination of current state and input.

6. State Diagram :- It is a graphical representation of an FSM depicting states as nodes and transitions as directed edges between states. It provides a visual representation of the system's behaviour and enables easy understanding and analysis.

Finite state machines can be classified into 2 main types based on their output behaviour:-

1. Moore Machine :- In this machine, the outputs are class associated with individual states and remain constant throughout each state's duration. Output generation is solely dependent on the current state of the system.

2. Mealy Machine :- In this machine, outputs are associated with transitions between states and are dependent on both the current state and input. Output generation occurs concurrently with state transitions, potentially resulting in more dynamic behaviour of the system as compared to Moore machines.

→ Sequence detector. (An application of Finite State Machines)
It is a type of digital circuit that identifies specific patterns or sequences of symbols within a stream of input data. The core of it is implemented by finite state machine. The FSM consists of a finite number of states, each representing a different condition or configuration of the detector. Transitions are triggered by input symbols and thus determine the behaviour of the detector.

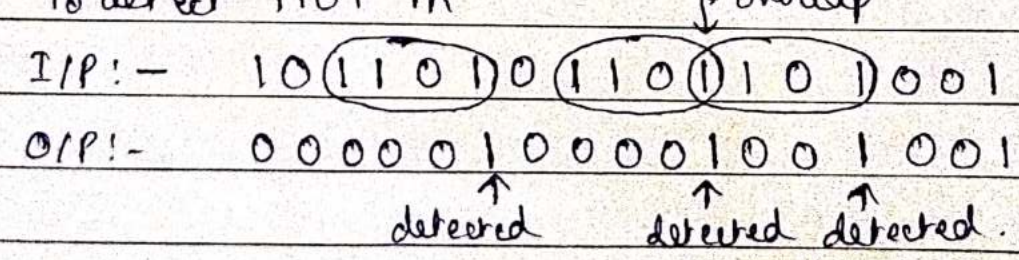
When the sequence detector recognizes the predefined pattern or sequence within the input data stream, it generates an output signal. The output signal indicates the detection of the desired sequence and can be used to trigger further actions or operations in the system.

→ Sequence detector 1101.

The sequence detector designed in this project takes a bit stream of data as its input and output becomes 1 only when the detector detects the input sequence 1101 in the the bit stream. Sequence detector are of two types:- overlapping and non-overlapping.

1. Overlapping sequence detector :- In this type, the detector is capable of detecting multiple occurences of the target sequence within the stream even if they overlap with each other.

Example :- To detect '1101' in



2. Non overlapping sequence detector :- In this type, the detector only detects disjoint occurences of the target sequence within the input data stream. It ignores any instances of the sequence that overlap with previously detected occurences.

Example:- Let us take the same sequence & bit stream data.

I/P :- 1 0 1 1 0 1 0 1 1 0 1 1 0 1 0 0 1

O/P 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0

detected detected.

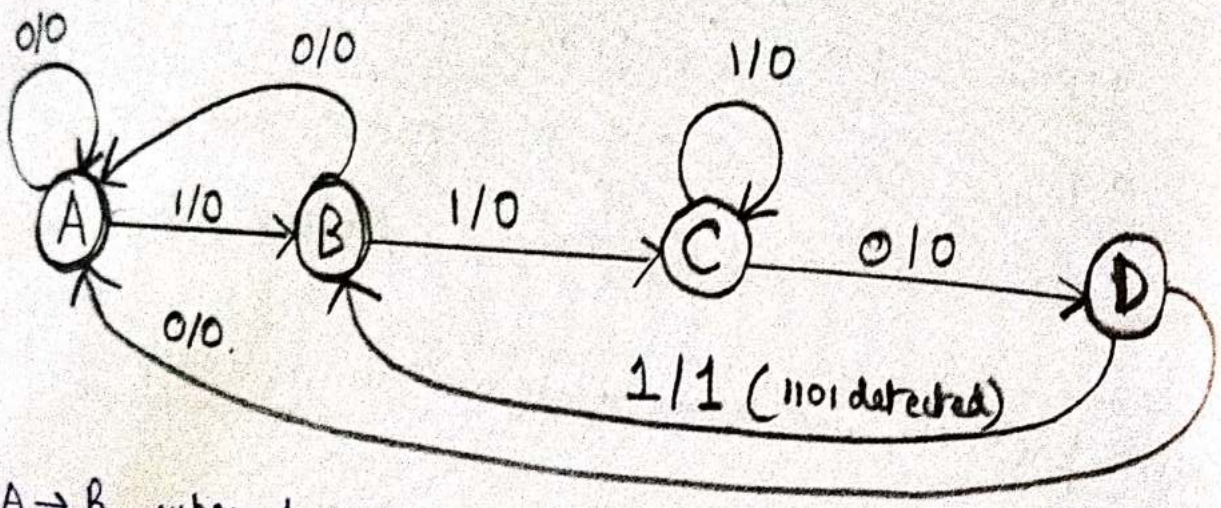
In our project, we have designed an overlapping sequence detector.

Real life applications of sequence detectors:-

→ In communication protocol to detect sync patterns and control characters within data packets
→ Data compression
→ Biomedical signal Processing
→ DNA sequencing
→ Security and cryptography.

These are just a few examples of the wide-ranging applications of sequence detectors in various fields. Their ability to identify specific patterns within data streams makes them invaluable for tasks ranging from communication and data processing to control and automation.

State diagram of 1101 overlapping sequence detecto



A → B, when 1 is detected as 1st bit
A → A, when 0 is detected as 1st bit.
B → A, when 0 is detected as 2nd bit.
B → C, when 1 is detected as 2nd bit.
C → C, when 1 is detected as 3rd bit
C → D, when 0 is detected as 3rd bit.
D → A, when 0 is detected as 4th bit.
D → B, when 1 is detected as 4th bit] → 1101 detected
and it continues ....

DATE 2/4/24

EXPT. NO.

Methodology

1101 overlapping sequence detector.

States required.	Meaning
A	No bits detected.
B	1 detected
C	1 1 detected.
D	1 1 0 detected.

Output = 1 if 4th bit is 1

State Table.

X = Input bit.

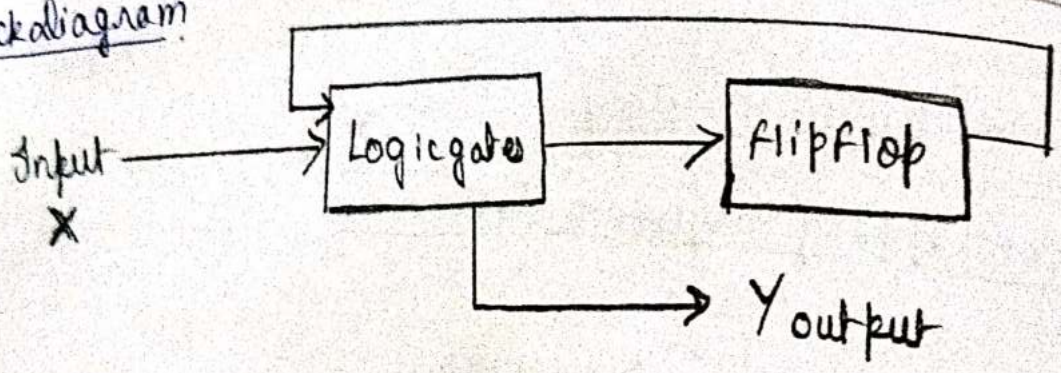
Present State	Next State		Output	
	X=0	X=1	X=0	X=1
A	A	B	0	0
B	A	C	0	0
C	D	C	0	0
D	A	B	0	1

State Table in Binary :-

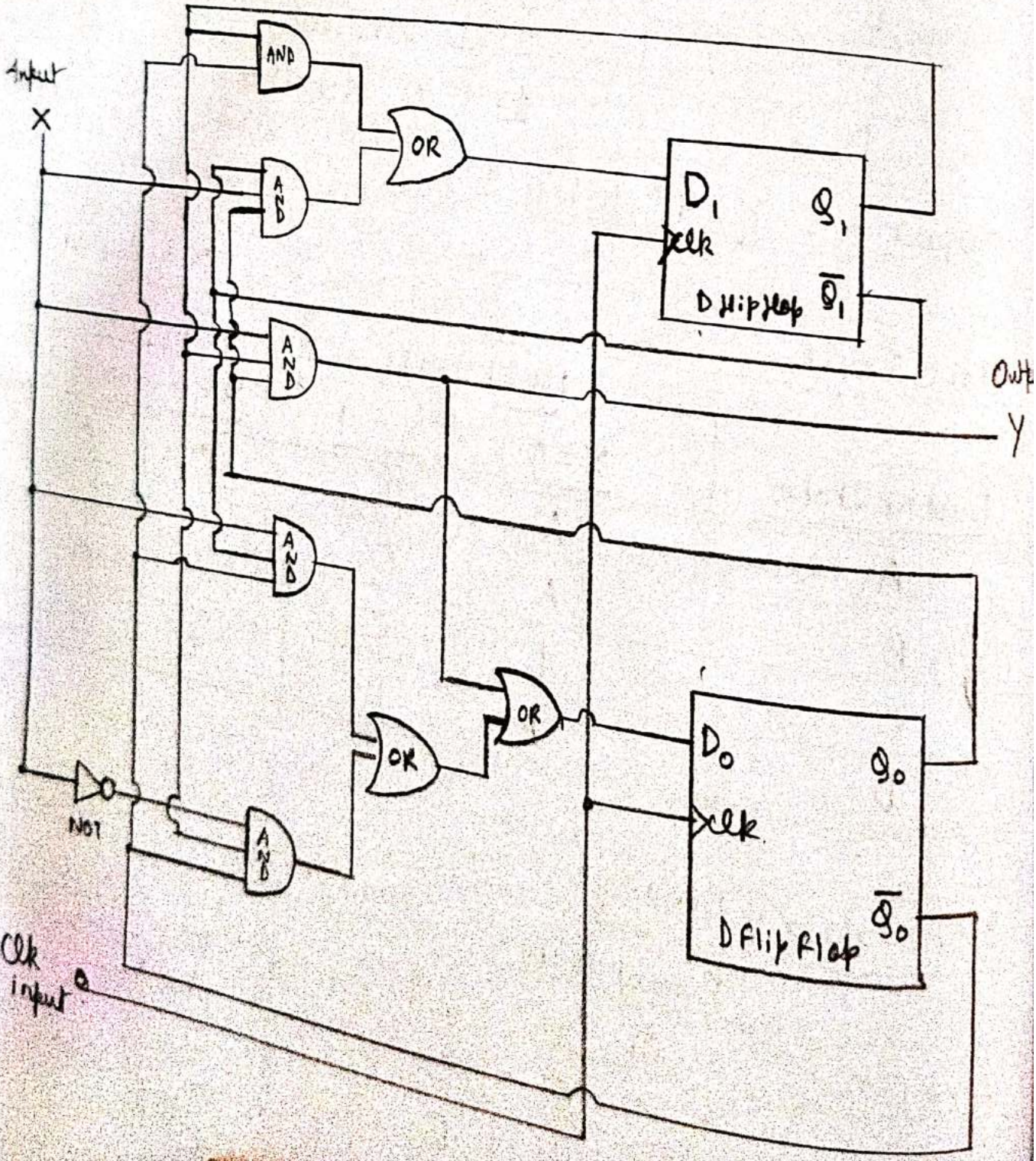
A → 00
B → 01
C → 10
D → 11

Present State		Next State				Output	
		X=0		X=1			
Q_1	Q_0	Q_1^+	Q_0^+	Q_1^+	Q_0^+	X=0	X=1
0	0	0	0	0	1	0	0
0	1	0	0	1	0	0	0
1	0	1	1	1	0	0	0
1	1	0	0	0	1	0	1

Block diagram



Circuit Diagram



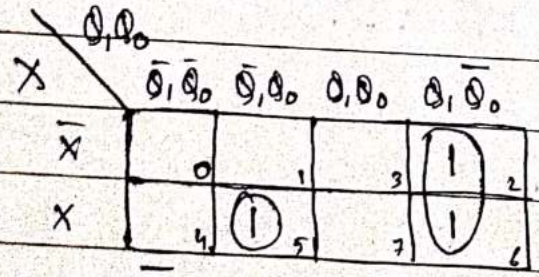
DATE
22/4/24
EXPT. NO.

pioneerpaper.in
PAGE NO. 7

Truth Table :-

Input	Present State		Next State		Output	Input to D Flipflop	
X	Q_1	Q_0	Q_1^+	Q_0^+	Y	D1	D0
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	1	1	0	1	1
0	1	1	0	0	0	0	0
1	0	0	0	1	0	0	1
1	0	1	1	0	0	1	0
1	1	0	1	0	0	1	0
1	1	1	0	1	1	0	1

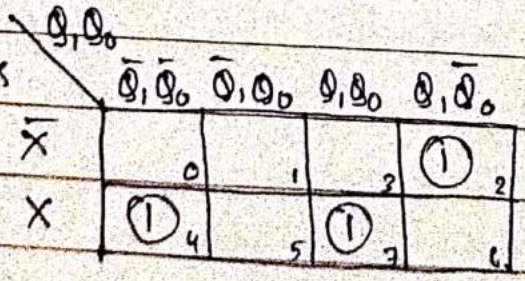
D1 :-



$$D1 = X\bar{Q}_1Q_0 + Q_1\bar{Q}_0$$

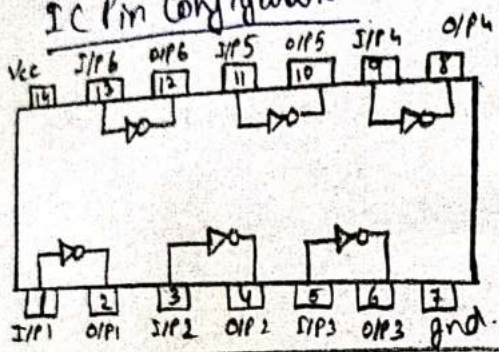
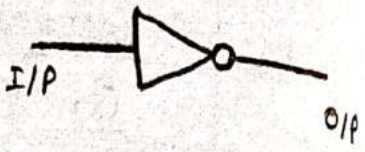
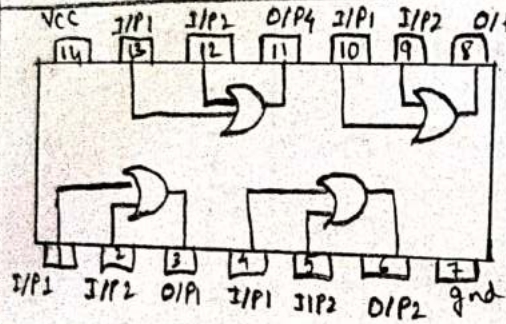
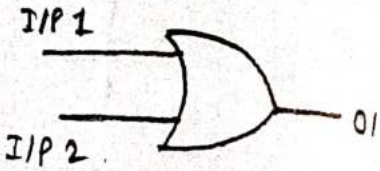
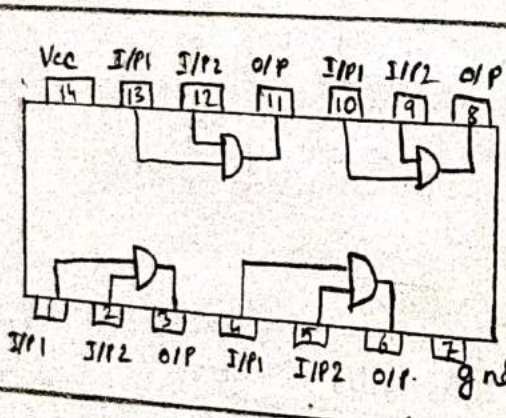
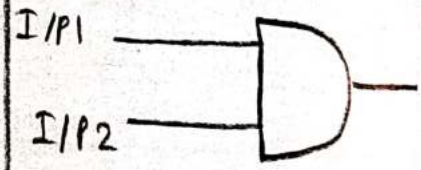
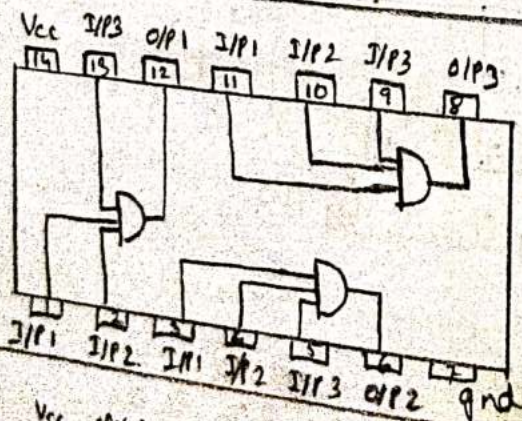
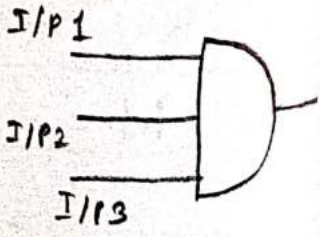
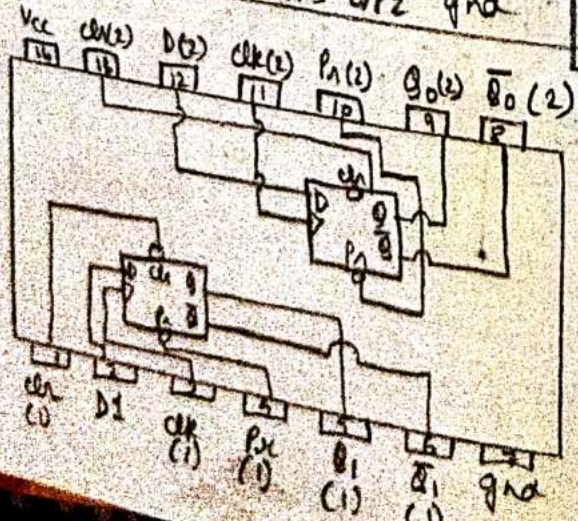
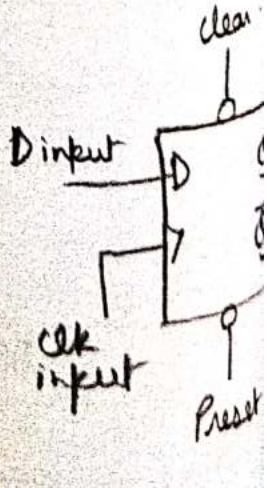
$$Y = XQ_1Q_0.$$

D2 :-



$$D0 = X\bar{Q}_1\bar{Q}_0 + XQ_1Q_0 + \bar{X}Q_1\bar{Q}_0$$

IC symbols and pin configuration

Name	IC Pin Configuration	Symbol
Hex NOT gate		
Quad 2-input OR gate		
Quad 2-input AND gate		
Triple 3-input AND gate		
Dual D-FlipFlop		

~~Compo~~

Resource Utilization

components required :-

	Name	IC number (if any)
①	Dual D Flip Flop. IC.	MC74 ACT 74N.
②	Triple 3-input AND gate IC (x2).	SN74LS11N.
③	Quad 2-input OR gate IC.	SN74HC32N.
④	Quad 2-input AND gate IC.	SN74HC08N.
⑤	Hex NOT gate IC.	SN74HC04N.
⑥	Jumper wires (Male to Male)	–
⑦	Breadboard.	–

Lab equipments.

	Name	Specifications.	Range
①	Power supply.	Scientific Multiple Power supply PSD3304	0 – 5 volt. working voltage :- 4.3 V.
②	Data generator.	8-bit variable data generator ST2111 (Scientech)	0 to 8 bit. working bits :- 11011010
③	Digital storage Oscilloscope.	Gwinstek. GDS-1102B.	100MHz, 1GS/s.
④	External connecting wires	–	–

↑ sequence detector complete setup ↓

Implementation Steps

→ Place the breadboard on a uniform plane region for smooth operation.

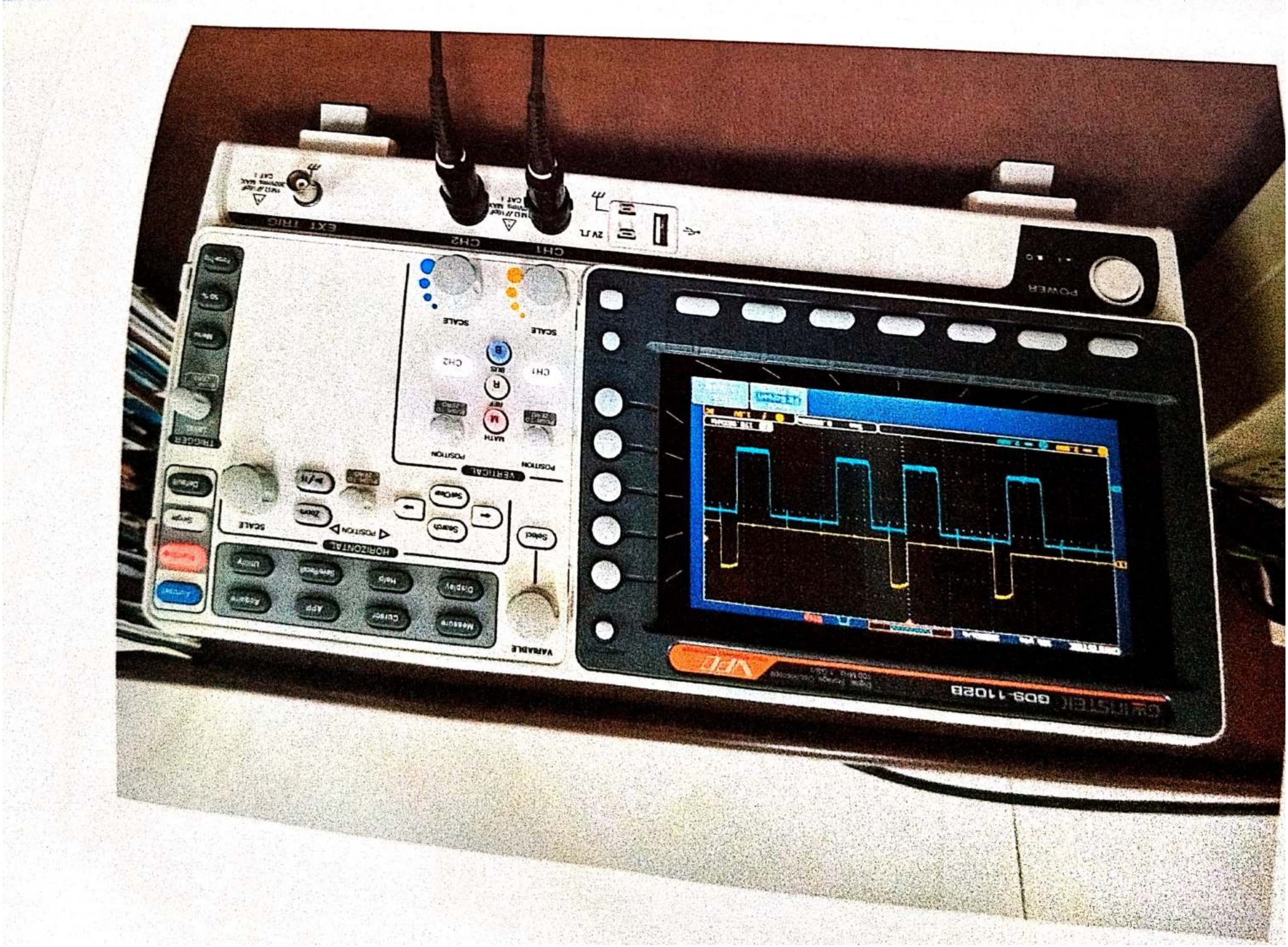
→ Connect all the integrated circuits on the breadboard which are required as per the circuit diagram.

→ Connect the lines across the ICs as shown in the circuit diagram by clearly following the pin configuration structure of respective ICs with the help of jumper wires.

→ After connecting them, connect all the Vcc pin and ground pin of all the ICs to the two respective Vcc line and ground line on the breadboard.

→ Give power supply to the ICs by connecting it with the help of connectors to the Vcc line on the breadboard and also to the preset and clear pins on flip flop.

→ Now connect the data generator to the breadboard connection at 'Input X' terminals on the ICs at required pins. Connect it with the 'data' socket on the generator.



→ Now connect the oscilloscope to the sequence detector, one channel to the input & and another channel to the output with the help of connectors so that we can observe the bit patterns of both parallely and cross-check.

→ Now switch on the power supply and maintain voltage at around 4.3 V.

→ Give the variable 8 bit data with the help of the data generator & 1/0 switches and connect the clock terminal from it to the respective flip-flop's clock pin. Apply a bit stream say '11011010' and select 'load' on the generator.

→ Run the oscilloscope & and capture the waveforms of both channel. You can observe that the output wave has value 1 whenever it detects the complete '1101' from the input waveform.

—x—

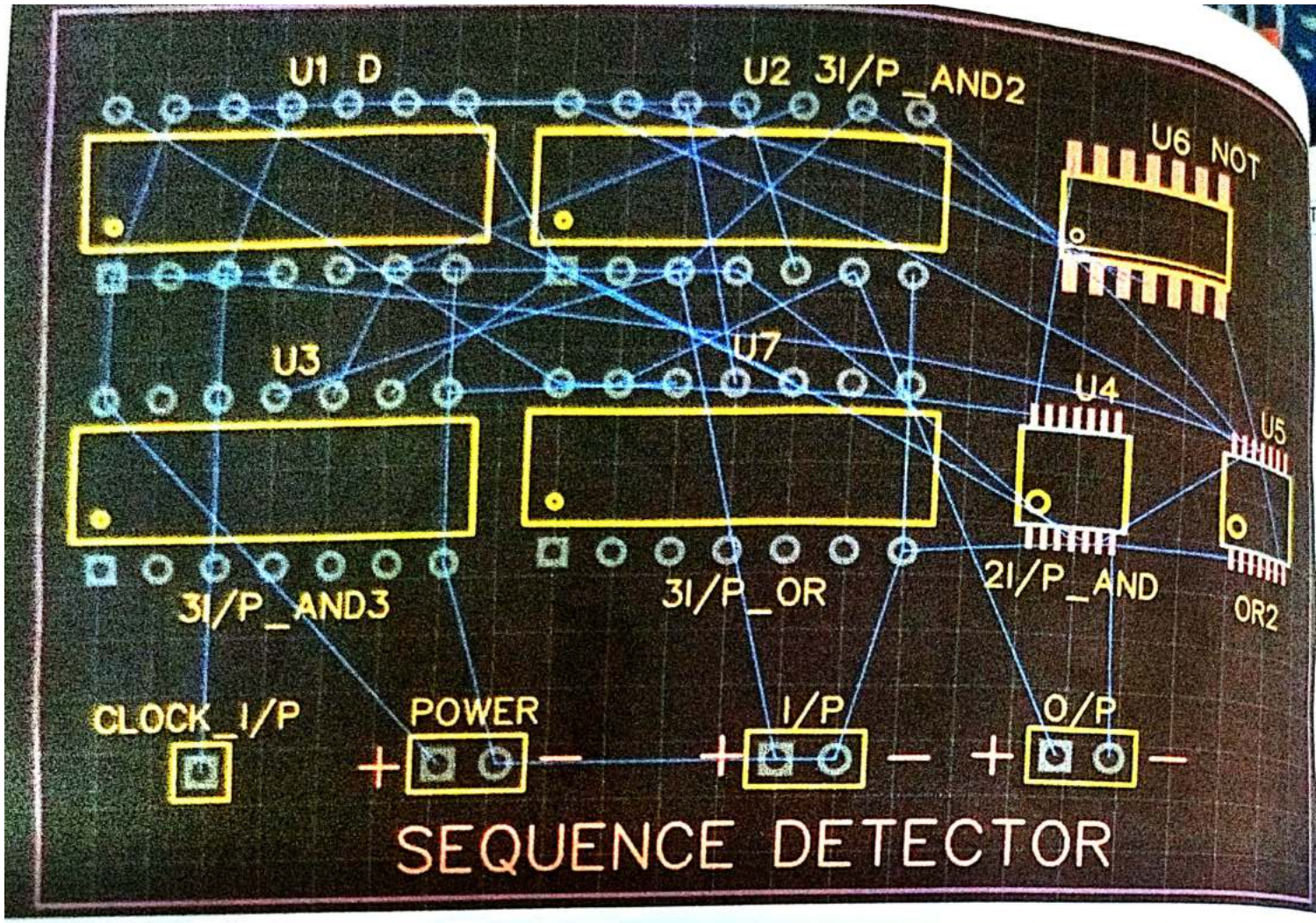
~~Status:- Project~~

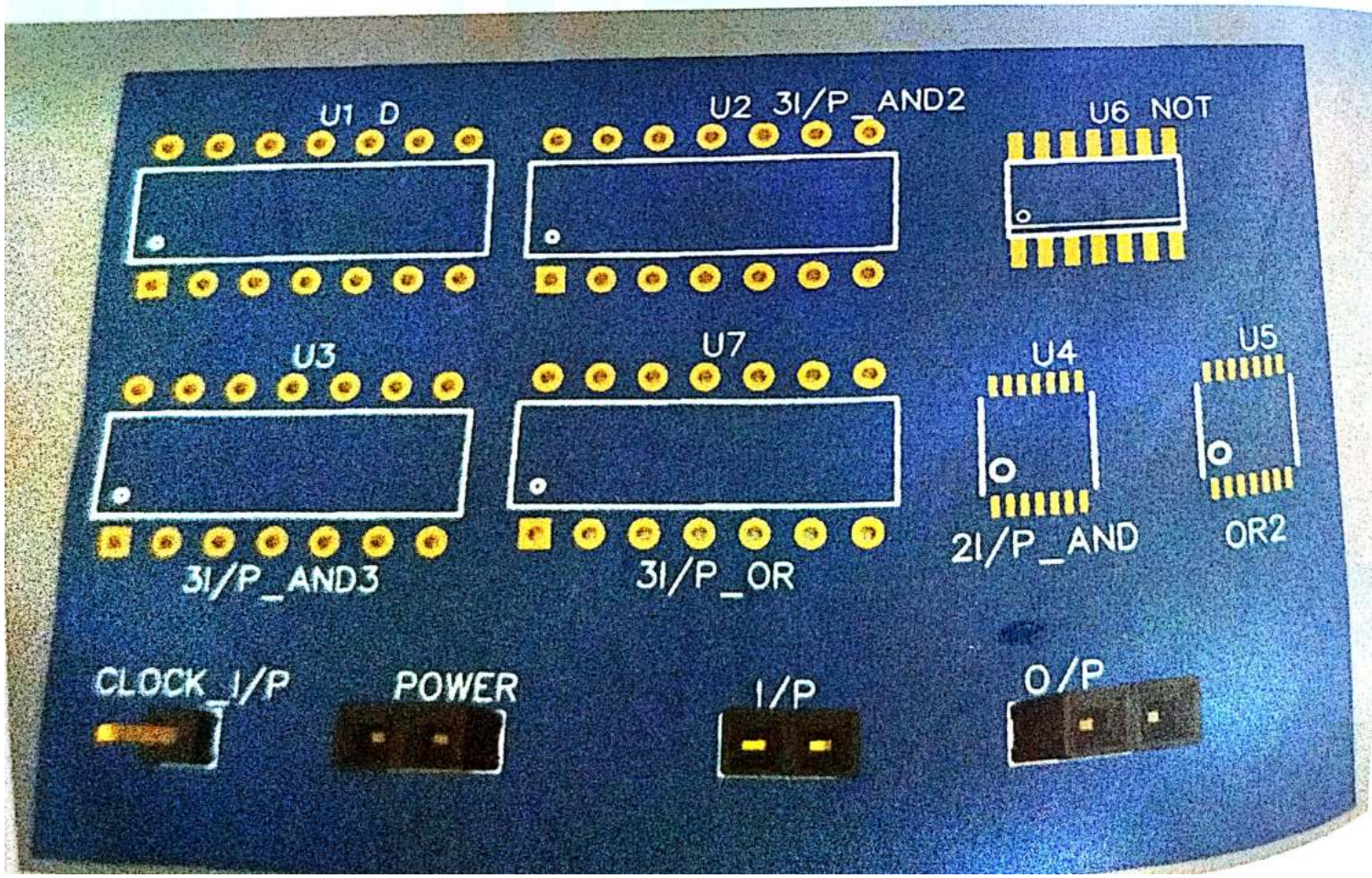
O/P results as shown in live picture.

Input :- 1 1 0 1 1 0 1 0

Output :- 0 0 0 1 0 0 1 0

detected detected





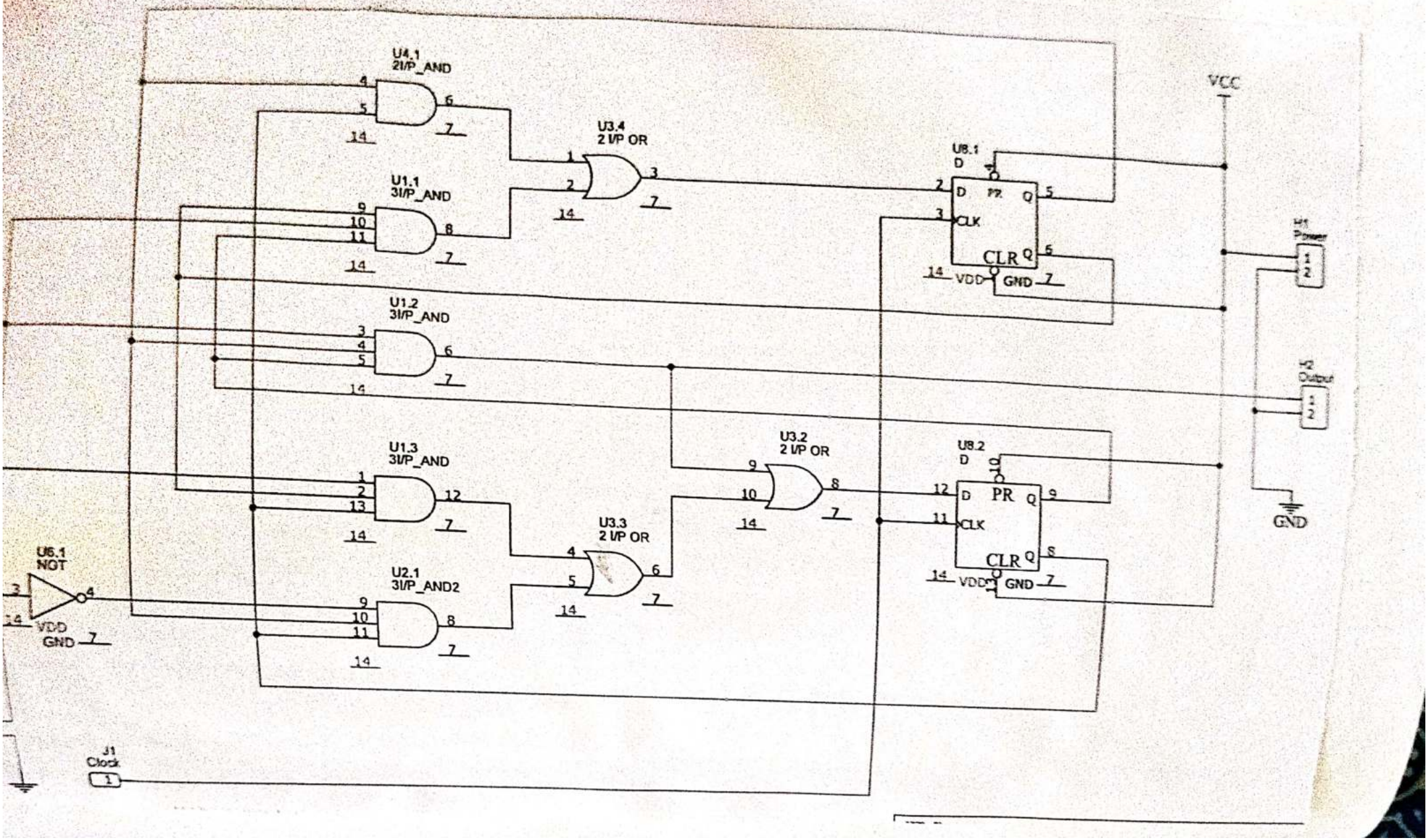
Critical Misuse.

The most important thing which one should keep in mind is that we should never forget to connect the clear and preset terminal of the flipflop IC to the power supply. i.e ~~provide~~ exciting them with binary '1' input. If both clear and preset input has binary 1 as input, it implies that the flipflop will work on its normal stage without any hindrance. Leaving clear and preset ~~ope~~ as open circuit may result in the failure of the project. This is the only critical misuse encountered by us.

Software used for designing circuit ~~for~~ and PCB layout :-

:- Easy EDA.

Circuit layout on Simulation window



Conclusion:-

Through meticulous design, testing and iteration, we've developed a robust system capable of accurately identifying and responding to target sequences. Moving forward, we can leverage the insights gained from this endeavor to tackle even more ambitious projects.