DIGITAL DESIGN

INSTR F215

GROUP 88

Problem Statement:

Design an automated chessboard where the user selects the chess piece and inputs the final position. Assume that your chess board is automated for the queen and rook only.

Members:

1)Yash Jangir - 2019A8PS0526G

2)Shreyas Singh - 2019A8PS0532G

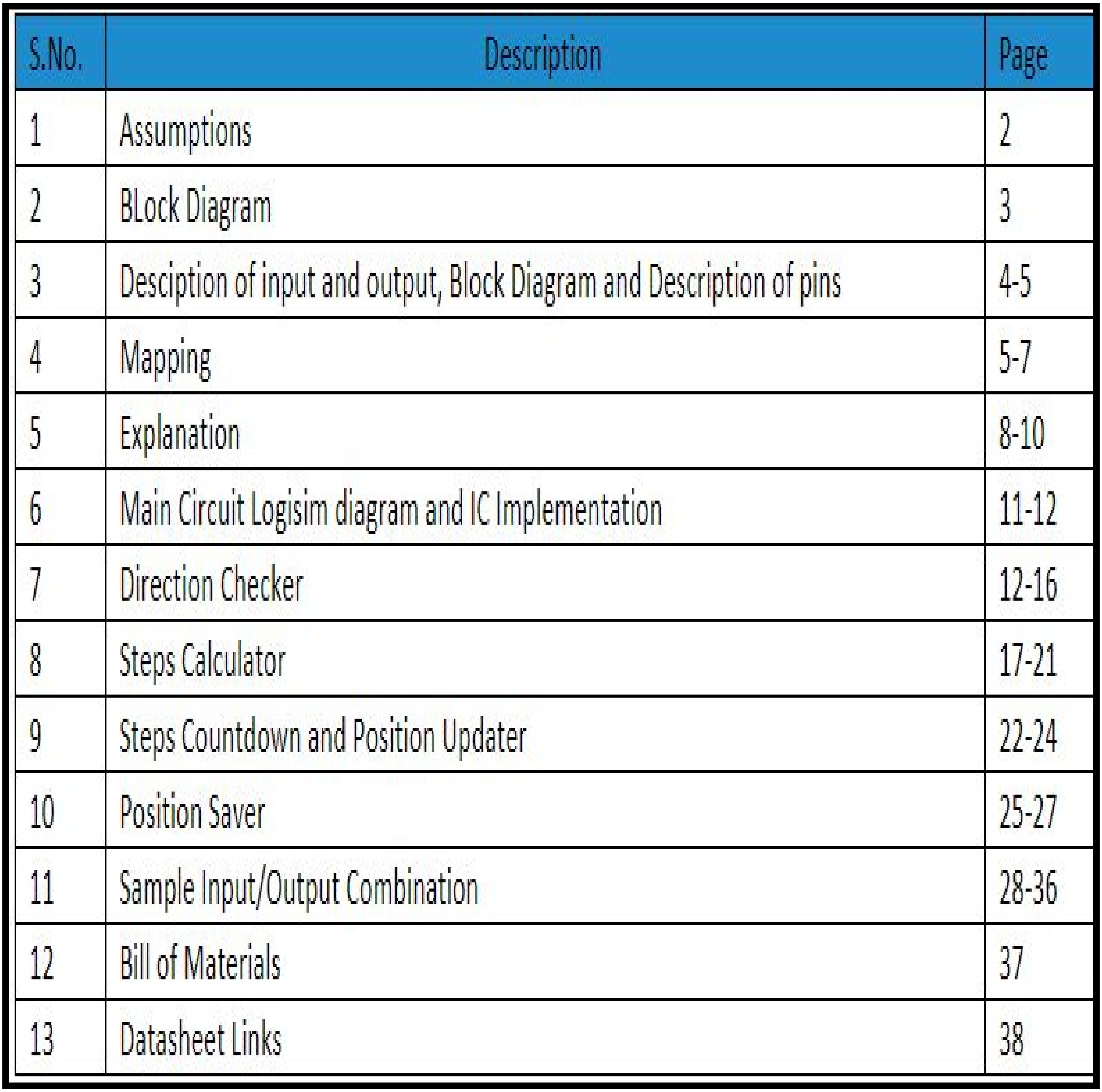
3)Johnson Baby - 2019A8PS0553G

4)Pranav Patil - 2019A8PS1034G

5)Vikramaditya Voleti - 2019A8PS0642G

6)Snigdha Tiwari - 2019A8PS0555G

7)Rushikesh Nikam - 2019A8PS0607G



ASSUMPTIONS :

Certain assumptions have been made while designing the chip to account for the limitation of the design. These assumptions also facilitate proper working of the circuit.

- The circuit gives the output as the direction in which the actuators have to make the piece move every step while moving. It is assumed that the servo motors react and complete the movement per step in less than a clock cycle and take more than half the clock cycle to move per step. For example: if the standard clock cycle is 1 second then the actuators complete the move before 1 second but take more than half-second. The reason for this assumption is as we do not know how much time the actuator will take to move per step. We have provided the clock input which controls the timing of direct output to the actuator. If the motor completes its movement before half a clock cycle, it will again receive the signal to move, since the direction is still available at the output pin.For every negative edge, the output of the direction will be reset to 1001, implying no movement, so that there are no extra moves and on the next positive edge a new move direction will be given. So timing must be between t/2 to t.(t being the time period of the clock).
- It is considered that there are no gate delays in the circuit.
- It is also assumed that the user presses the button correctly to detect the input.
- The user has the option to select the initial position of all 3 pieces of his or her own choice.
- It is assumed that the user will not press the move button for the next movement until the current movement sequence is complete.
- It is assumed that there is no noise in the circuit.
- It has been assumed that the user is aware of the pieces on the chessboard and hence will not give an input that places another piece in either the path or position of the selected piece.

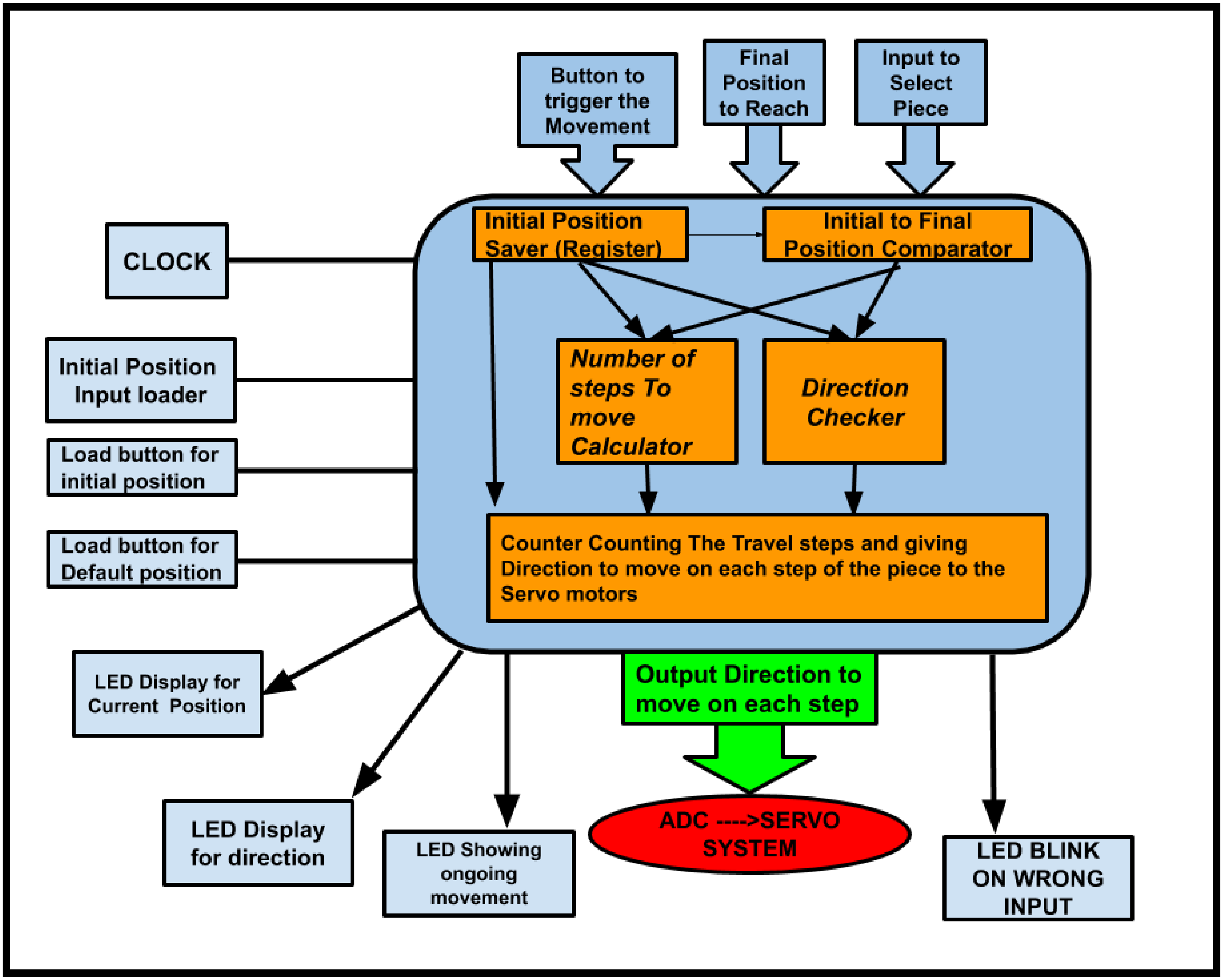

Button to trigger the Movement
Final Position to Reach
Input to Select Piece
CLOCK
Initial Position Saver (Register)
Initial to Final Position Comparator
Initial Position Input loader
Number of steps To move Calculator
Direction Checker
Load button for initial position
Load button for Default position
Counter Counting The Travel steps and giving Direction to move on each step of the piece to the Servo motors
LED Display for Current Position
Output Direction to move on each step
LED Display for direction
LED Showing ongoing movement
ADC ---->SERVO SYSTEM
LED BLINK ON WRONG INPUT

FIG 1.1 A Block Diagram showing the Working of Circuit

All the mapping has been provided in the next section!

Input:-

1. **Final Position :** where the user wants to move the piece in the form of input X coordinate and input Y coordinate.

According to the above chessboard, the mapping is given.

2. **Input Piece Code:** according to the Piece Mapping specified.
3. **Movement Start Button :** Press to start the process of movement.

Output:

1.Direction to move as a 4-bit number according to the Direction mapping.

2. Number of Steps remaining.

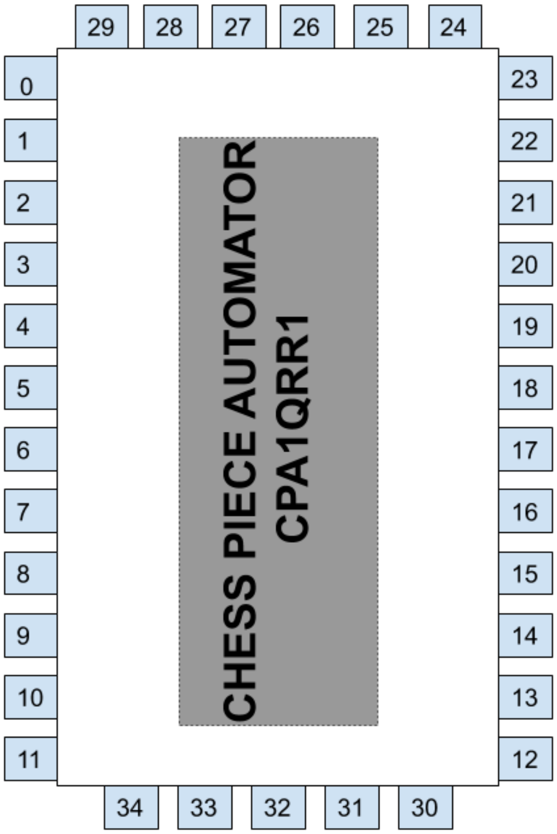
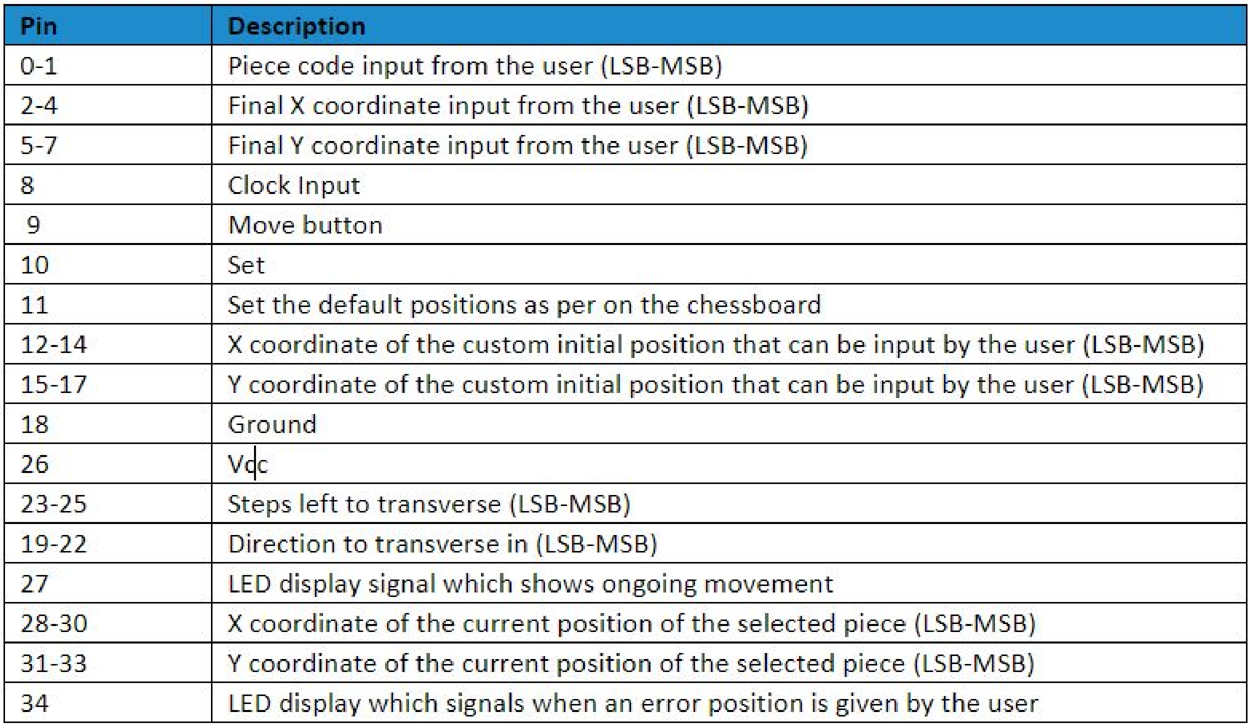


FIG 1.2 Final Circuit Pinout

Description of the pins:

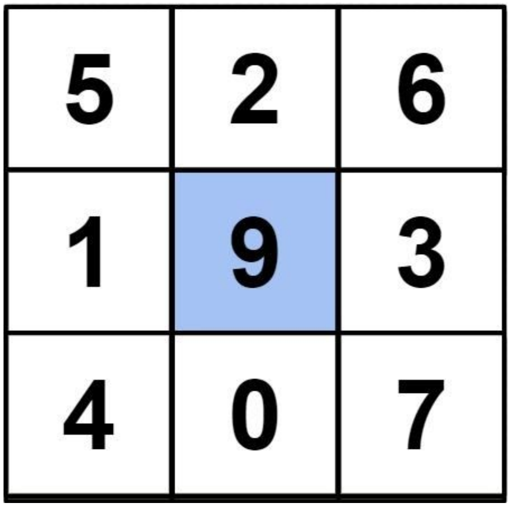


Pin	Description
0-1	Piece code input from the user (LSB-MSB)
2-4	Final X coordinate input from the user (LSB-MSB)
5-7	Final Y coordinate input from the user (LSB-MSB)
8	Clock Input
9	Move button
10	Set
11	Set the default positions as per on the chessboard
12-14	X coordinate of the custom initial position that can be input by the user (LSB-MSB)
15-17	Y coordinate of the custom initial position that can be input by the user (LSB-MSB)
18	Ground
26	Vdc
23-25	Steps left to transverse (LSB-MSB)
19-22	Direction to transverse in (LSB-MSB)
27	LED display signal which shows ongoing movement
28-30	X coordinate of the current position of the selected piece (LSB-MSB)
31-33	Y coordinate of the current position of the selected piece (LSB-MSB)
34	LED display which signals when an error position is given by the user

Mappings:

Direction Mapping:

- Assuming for any general piece which can move in all directions and 9 is coded as the same place/null/no direction.
- Mapping for direction to a 4-bit number is as follows.

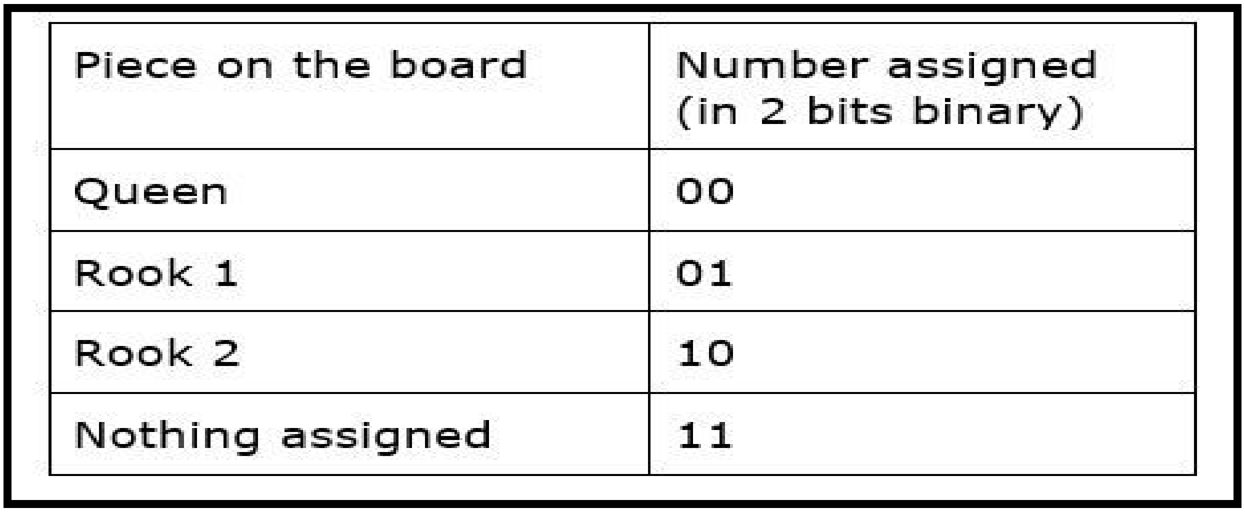


5	2	6
1	9	3
4	0	7

Fig 1.3 Direction Mapping

Piece Mapping:

The chessboard is automated for the queen and both the rooks. Hence to identify and select the pieces on the board, each piece is assigned a 2-bit binary number. Following is the mapping which the chessboard uses to identify each piece :-



Piece on the board	Number assigned (in 2 bits binary)
Queen	00
Rook 1	01
Rook 2	10
Nothing assigned	11

Table 1.1 Piece code

Chess Board :

- Each coordinate is a 3-bit number between 0 and 7.
- The left hand down the corner of the user.
- As there are 64 possible Positions on a 8x8 chess board, the board uses a 2 dimensional coordinate system to identify each state. The x and y axis ranging from 0 to 7 numbers each state from (0,0) to (7,7). The chess board identifies these states in this manner. Following is the board with the axes.

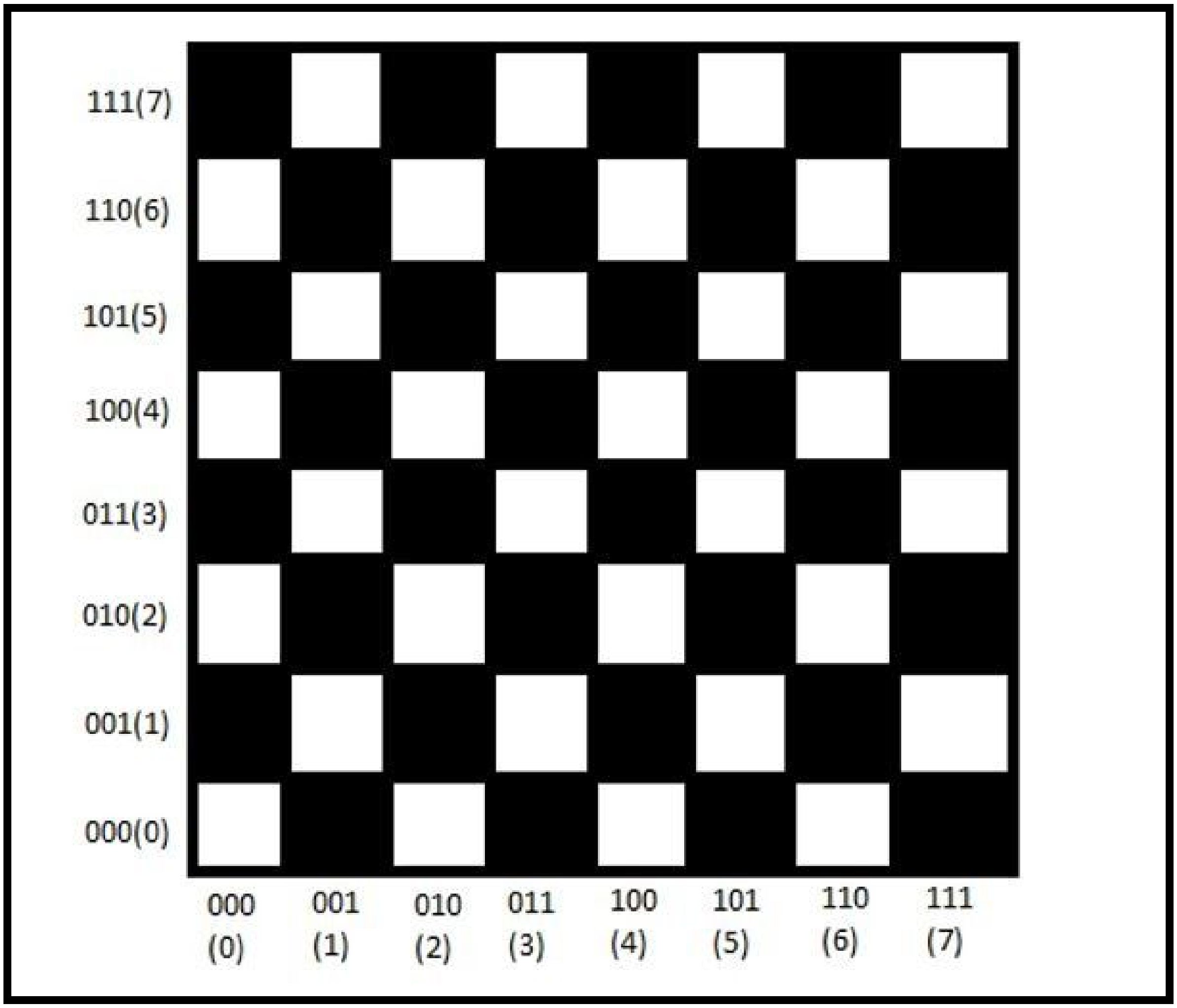


Fig 1.4 Chess Board

- User inputs the portion to transverse keeping this in mind.

EXPLANATION

- The circuit works in stages like an algorithm implemented to calculate the movement before the actual motion.
- For every final position given by the user and the load button/move button by the user, the selected piece will move only if the input position is correct.
- In case the user gives an invalid position, the circuit will give the output as no movement to the actuators as the logic number 9.
- For every piece to move, the user has to select that particular piece. Currently only the Rooks and the Queen have been implemented in this circuit but this logic can even be used for Bishops and kings.
- As the user presses the button, the counter counts down from the number of steps and gives which direction to move on, on each step to the servo motor according to a predefined direction to digital number mapping.
- The initial position of the pieces is saved in a register system after the movement of the piece is completed. The current position is updated in the register system and further movements are made thereon.

So the user has to input the final position and press the button. The rooks have already been initialized to be at corners and Queen in the center. The user has a choice either to preset the initial positions as defined on the chessboard or input an initial position as required at the start.

The circuit is divided into four parts which combine to give a final Output after comparison of the position to move and initial position :

1. Direction Checker
2. Steps Calculator
3. Steps countdown and position updater
4. Position Savers

Algorithm And Logic:-

The best way to make a move is to calculate beforehand what direction to move and how much to move in order to reach a certain position.
We have assumed and the problem statement states that servos make 1 step at a time.
So we can control what direction and how many steps the piece moves.

Direction Output Logic:

- By simply comparing the initial coordinates stored in the position saver and the final coordinates, it gives a 4-bit output according to the direction mapping.

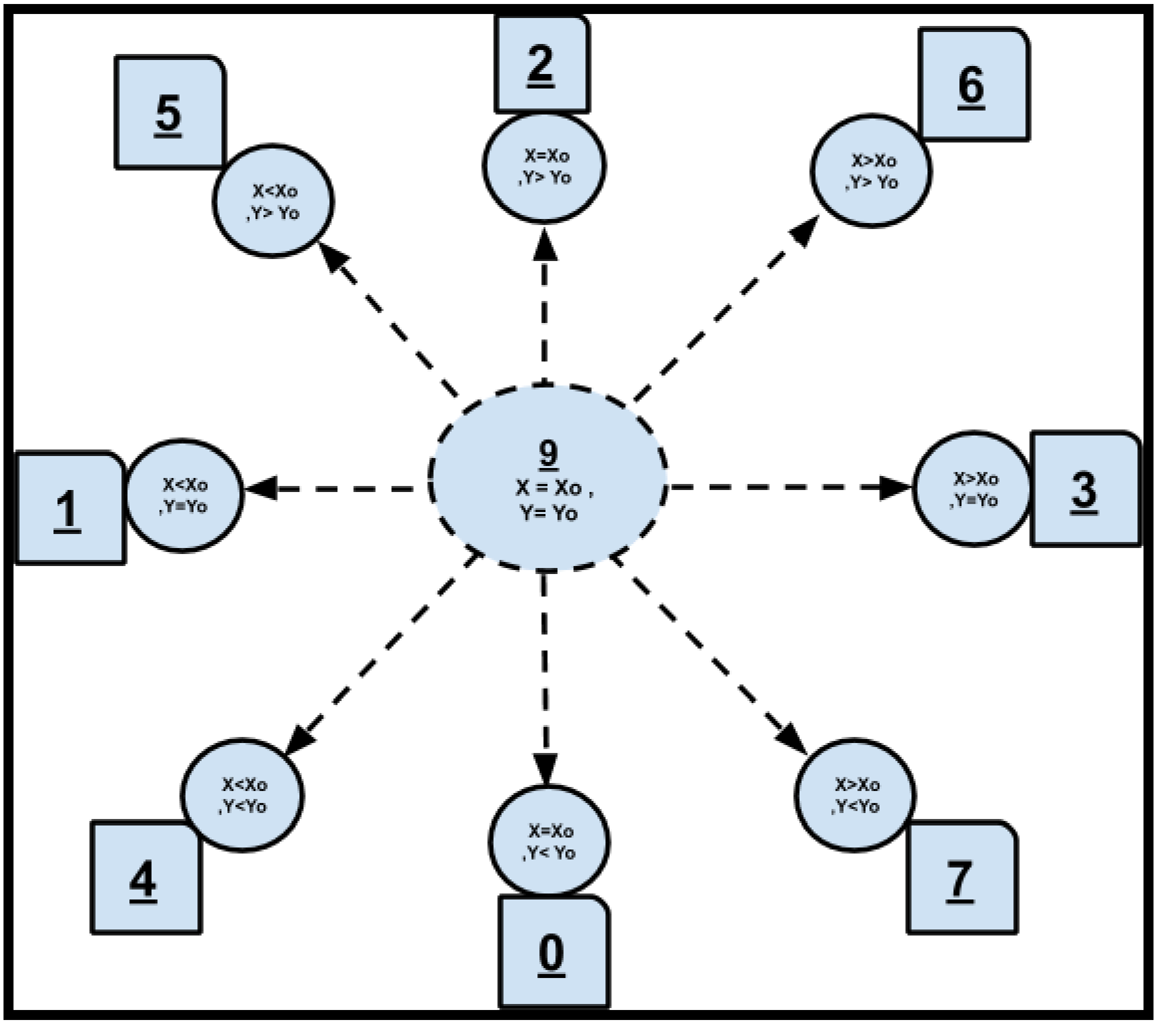


Fig1.5 Finding Direction

<u>**Number Of steps Logic**</u>:

Steps are calculated on a simple feature of chess board, that is, if a general piece that can move in a straight line,the number of steps that are needed to be traveled is the greater of |X-Xo| and |Y-Yo| or either one, if both X and Y coordinates change , given that the piece can travel to the position.
Here |X-X0| and |Y-Yo| can be defined as the mod of the X coordinate difference and Y coordinate difference Respectively.
Representation - (T,S) = (|X-X0|,|Y-Y0|)

Case 1: Piece has to travel Forward or Backward.
Here the coordinate the coordinate differences give values of type :
(0,S)
So the Steps to be travelled will be S.

Case 2 : Piece has to move Rightwards or Leftwards
Here the coordinate the coordinate differences give values of type :
(T,0)
So the Steps to be travelled will be T.

Case 3 : Piece has to move in Diagonal Direction
Here the coordinate the coordinate differences give values of type :
(T,S) but here T and S will be equal and give the steps so T=S.
So the Steps to be travelled will be T or S.

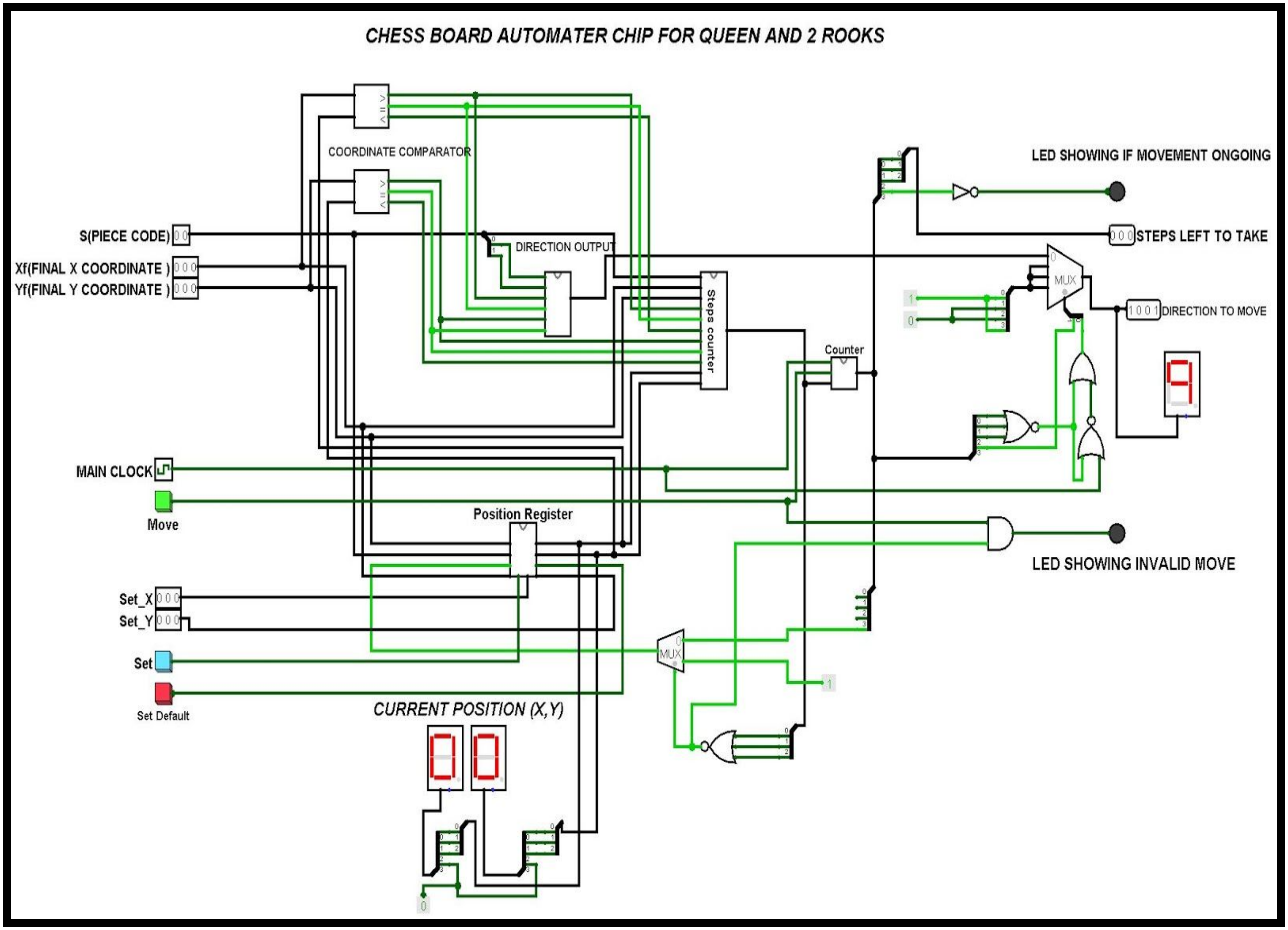
Case 4 : Ensuring the Right move and not taking extra steps.
In this case the Coordinate difference will be of type (T,S) and T =! S.
Where none of S and T is 0.
In this case the point is not a valid move. Therefore the Queen and rook shouldn't move!Hence, steps will be 0.

Now that we know how many steps to move and in which direction we have to move .
We take a countdown that is asynchronous to the number of steps and give output as the direction needed to be travelled on every step.Hence the output

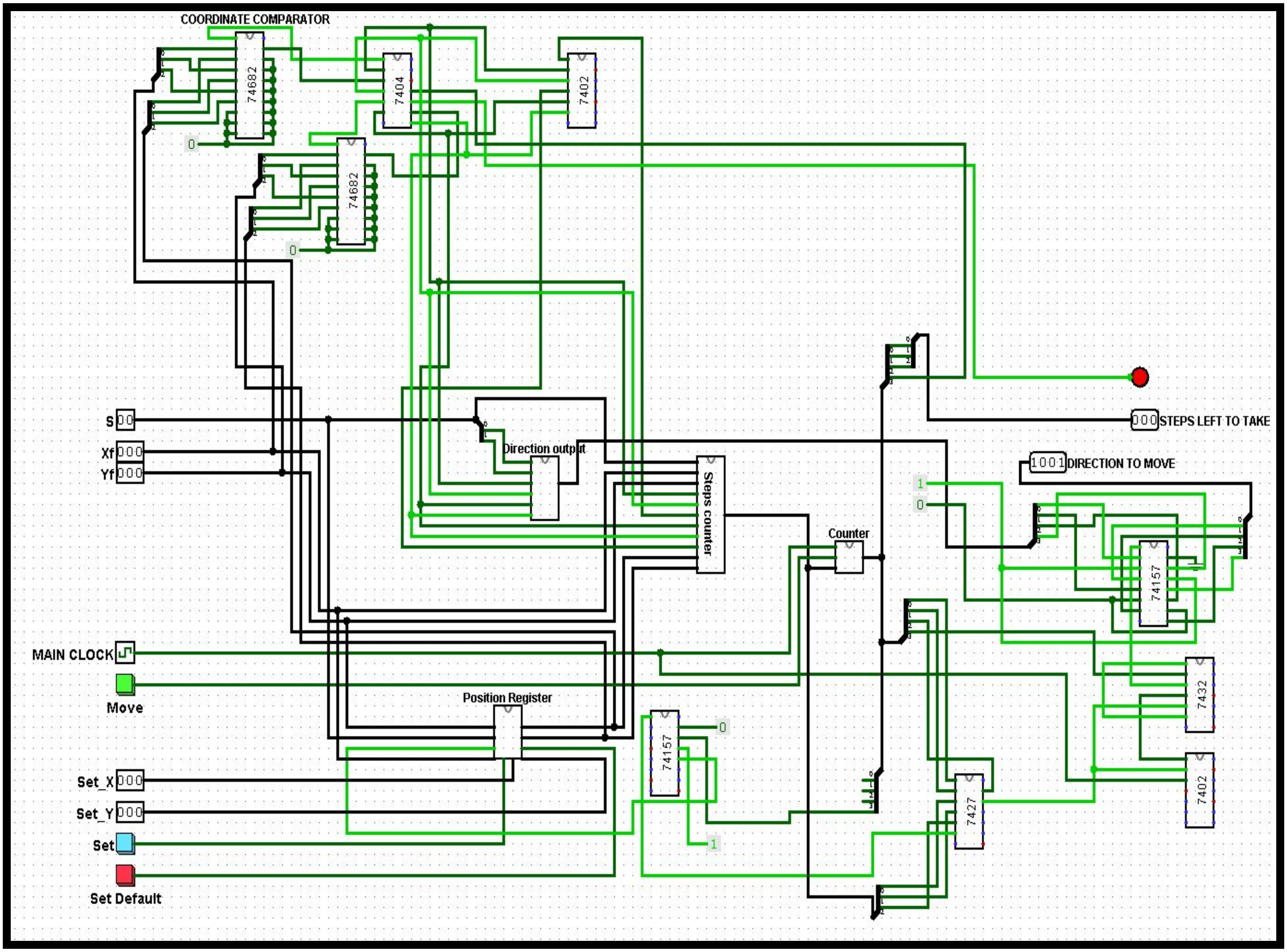
will Fluctuate between don't move and direction to move so that piece moves and knows next where to move.

All the final positions are saved in the registers after movement is complete so that the piece will move from that position next time.

Main Circuit :



Main Circuit’s IC Implementation:



Direction Checker :

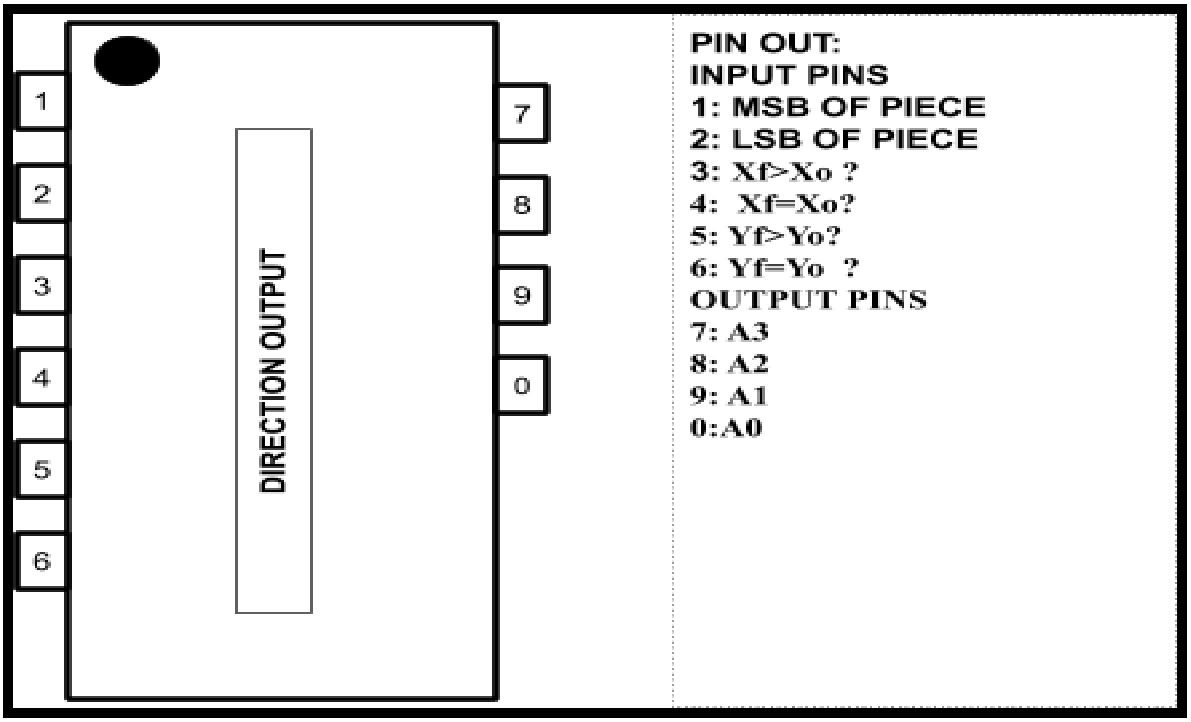
The circuit takes inputs:

1. Piece code.
2. Comparator outputs (from comparison of current and final positions).

The circuit outputs:

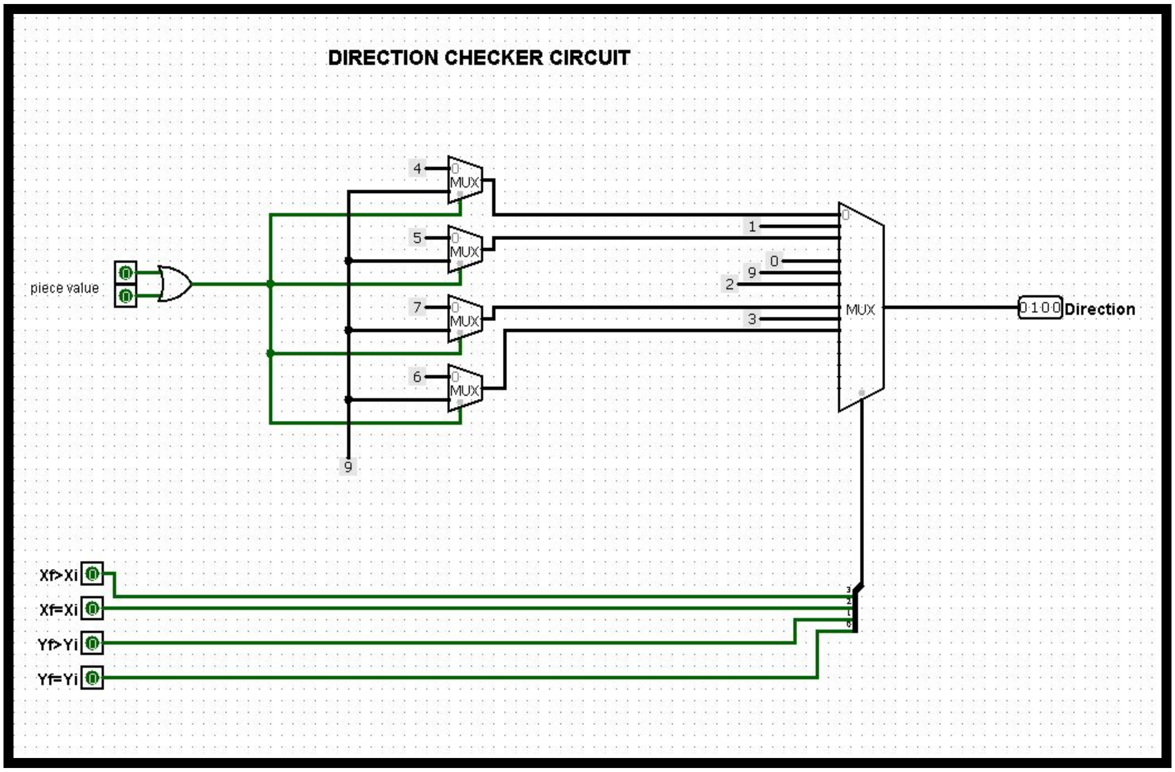
1. The direction of movement as a 4 bit binary number

PINOUT:



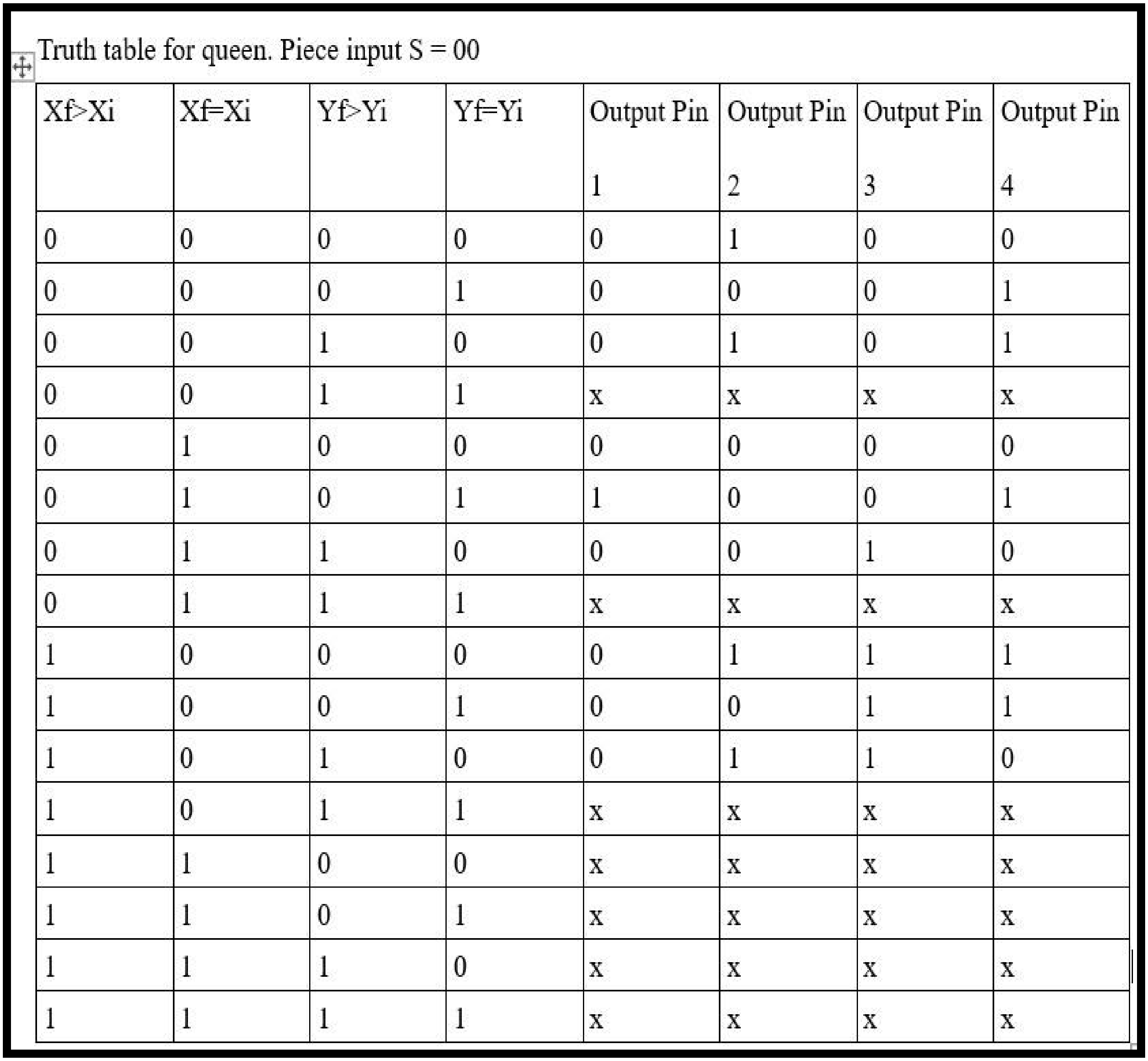
PIN OUT:
INPUT PINS
1: MSB OF PIECE
2: LSB OF PIECE
3: Xf>Xo ?
4: Xf=Xo?
5: Yf>Yo?
6: Yf=Yo ?
OUTPUT PINS
7: A3
8: A2
9: A1
0:A0

LOGISIM CIRCUIT IMPLEMENTATION :



Truth Table :

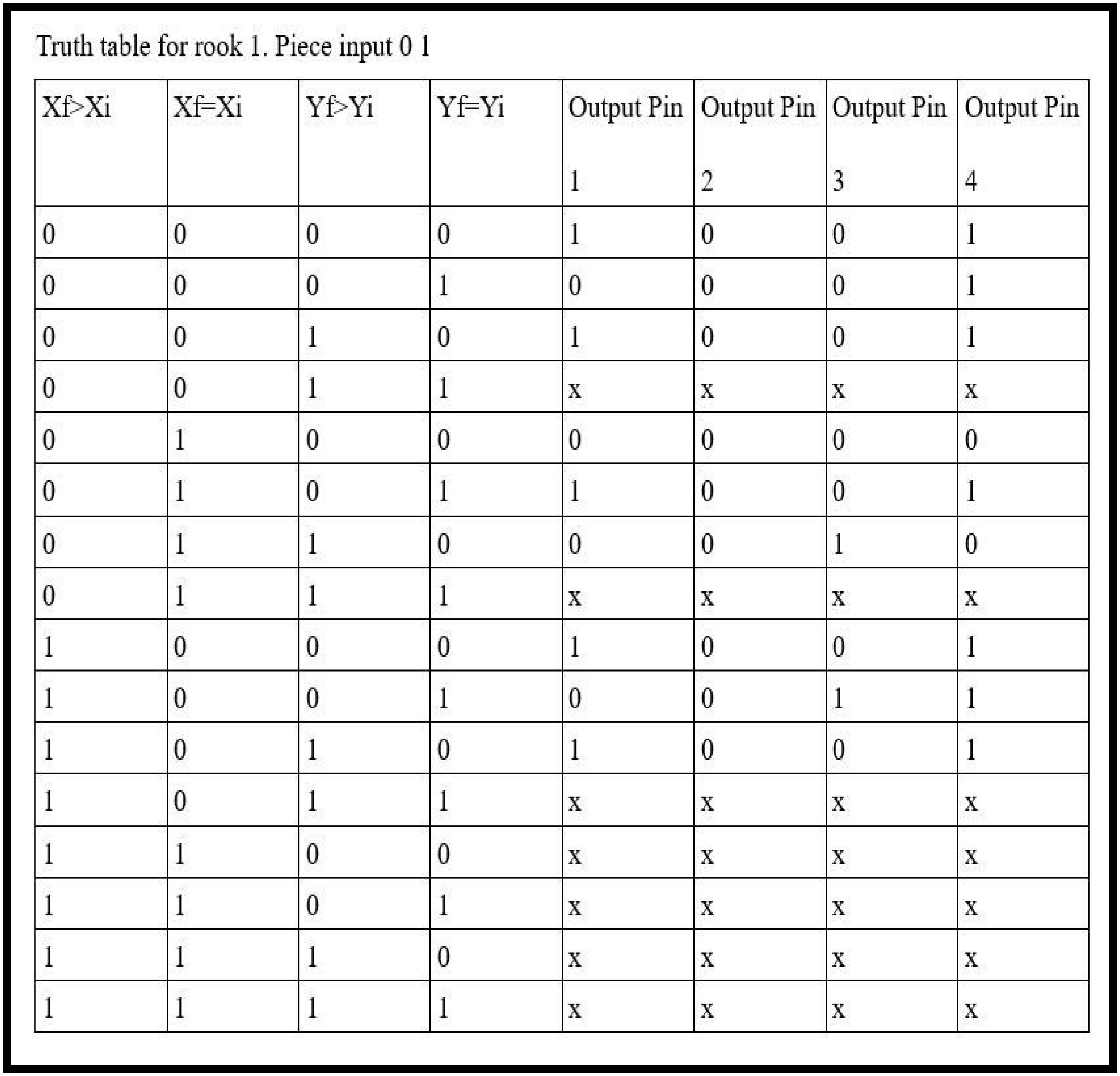
For Queen :



Truth table for queen. Piece input S = 00

Xf>Xi	Xf=Xi	Yf>Yi	Yf=Yi	Output Pin 1	Output Pin 2	Output Pin 3	Output Pin 4
0	0	0	0	0	1	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	1	0	1
0	0	1	1	x	x	x	x
0	1	0	0	0	0	0	0
0	1	0	1	1	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	x	x	x	x
1	0	0	0	0	1	1	1
1	0	0	1	0	0	1	1
1	0	1	0	0	1	1	0
1	0	1	1	x	x	x	x
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x

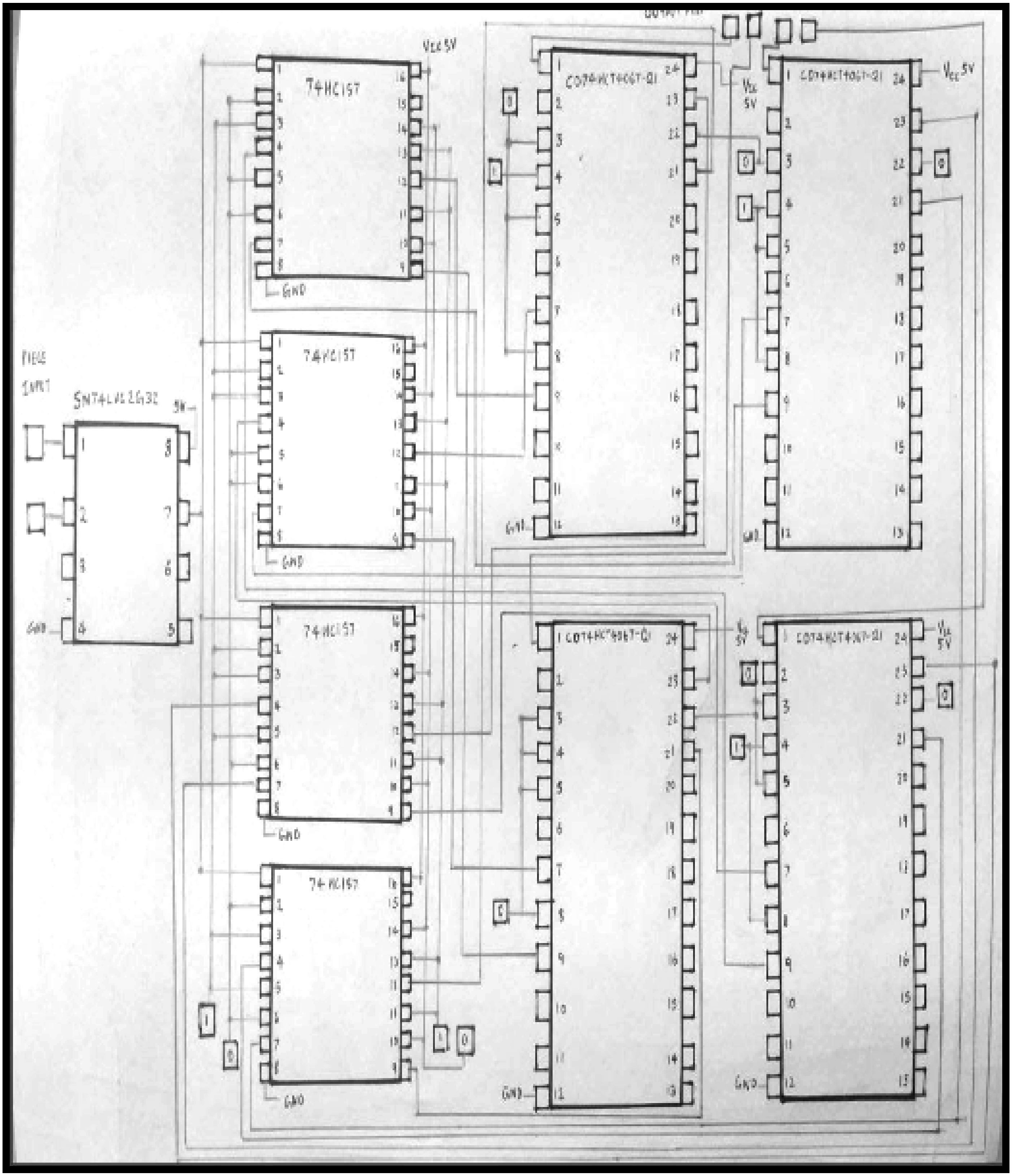
For Rook 1 and Rook 2 :



Truth table for rook 1. Piece input 0 1

Xf>Xi	Xf=Xi	Yf>Yi	Yf=Yi	Output Pin 1	Output Pin 2	Output Pin 3	Output Pin 4
0	0	0	0	1	0	0	1
0	0	0	1	0	0	0	1
0	0	1	0	1	0	0	1
0	0	1	1	x	x	x	x
0	1	0	0	0	0	0	0
0	1	0	1	1	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	x	x	x	x
1	0	0	0	1	0	0	1
1	0	0	1	0	0	1	1
1	0	1	0	1	0	0	1
1	0	1	1	x	x	x	x
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x

IC IMPLEMENTATION:



STEPS CALCULATOR :

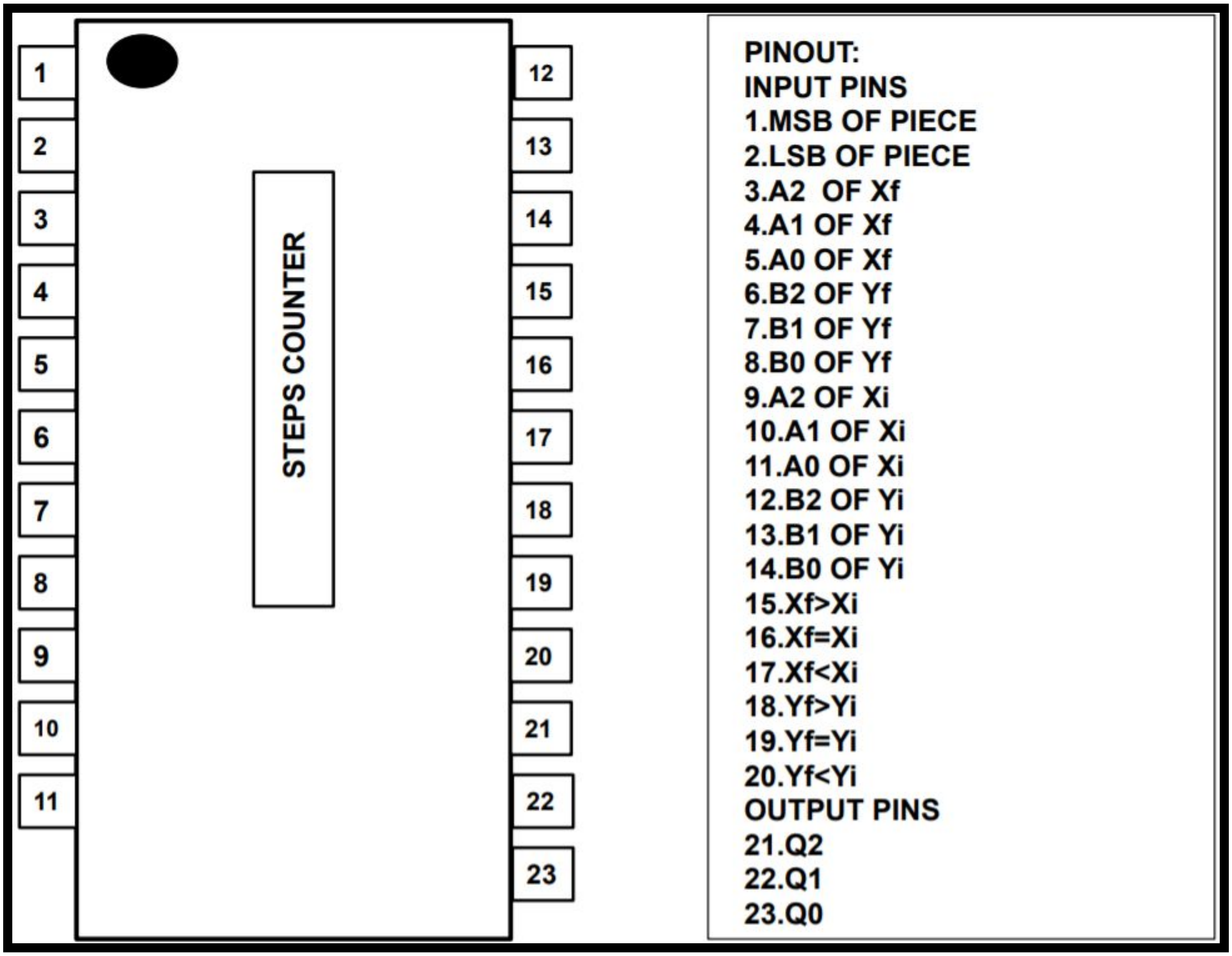
The circuit takes Inputs :

- Final position coordinates to move to.
- Initial position of the piece
- Outputs from the comparator(comparison of the initial and final position)

The circuit gives Outputs:

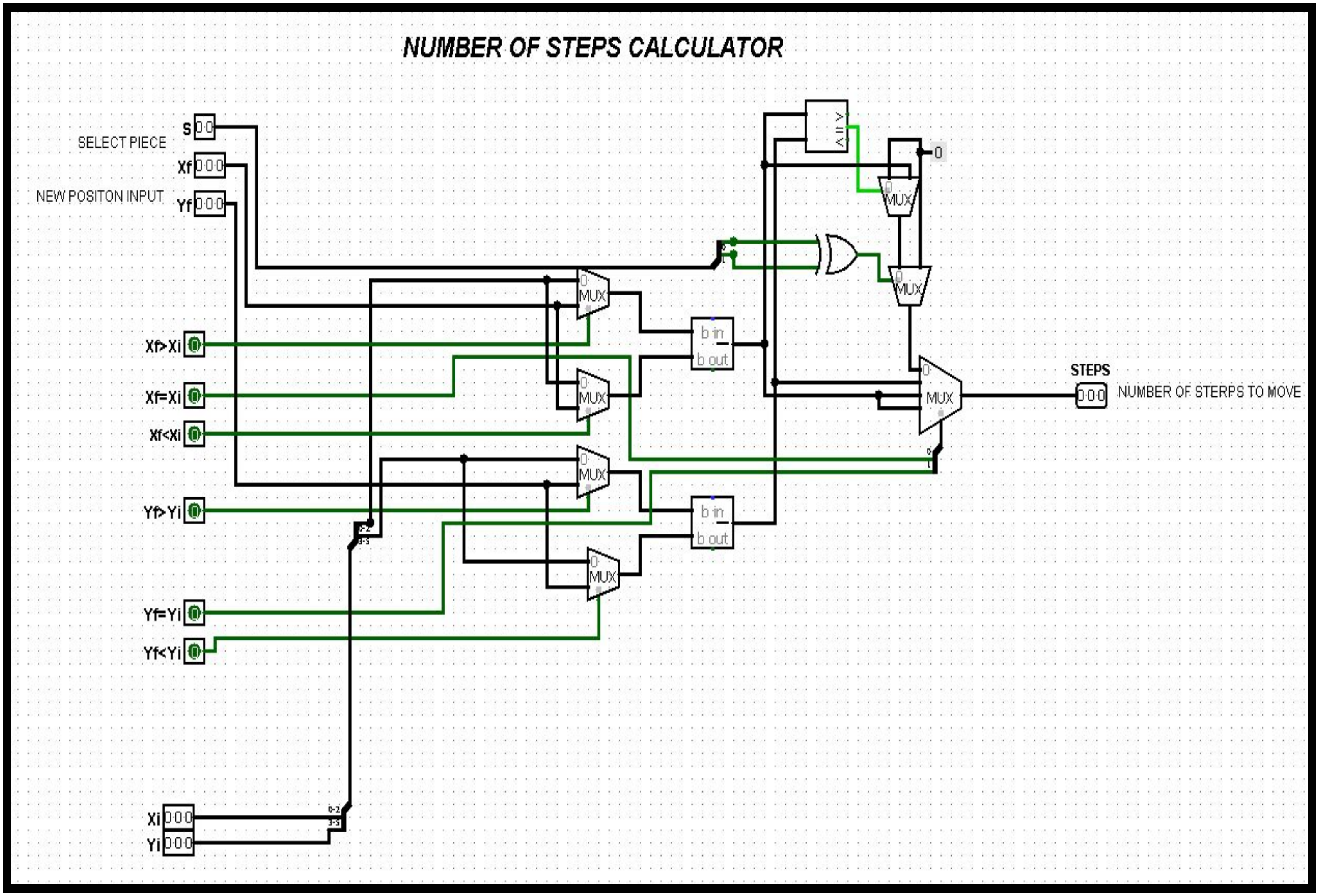
- Number of steps to move to reach final point given

Pinout Diagram :



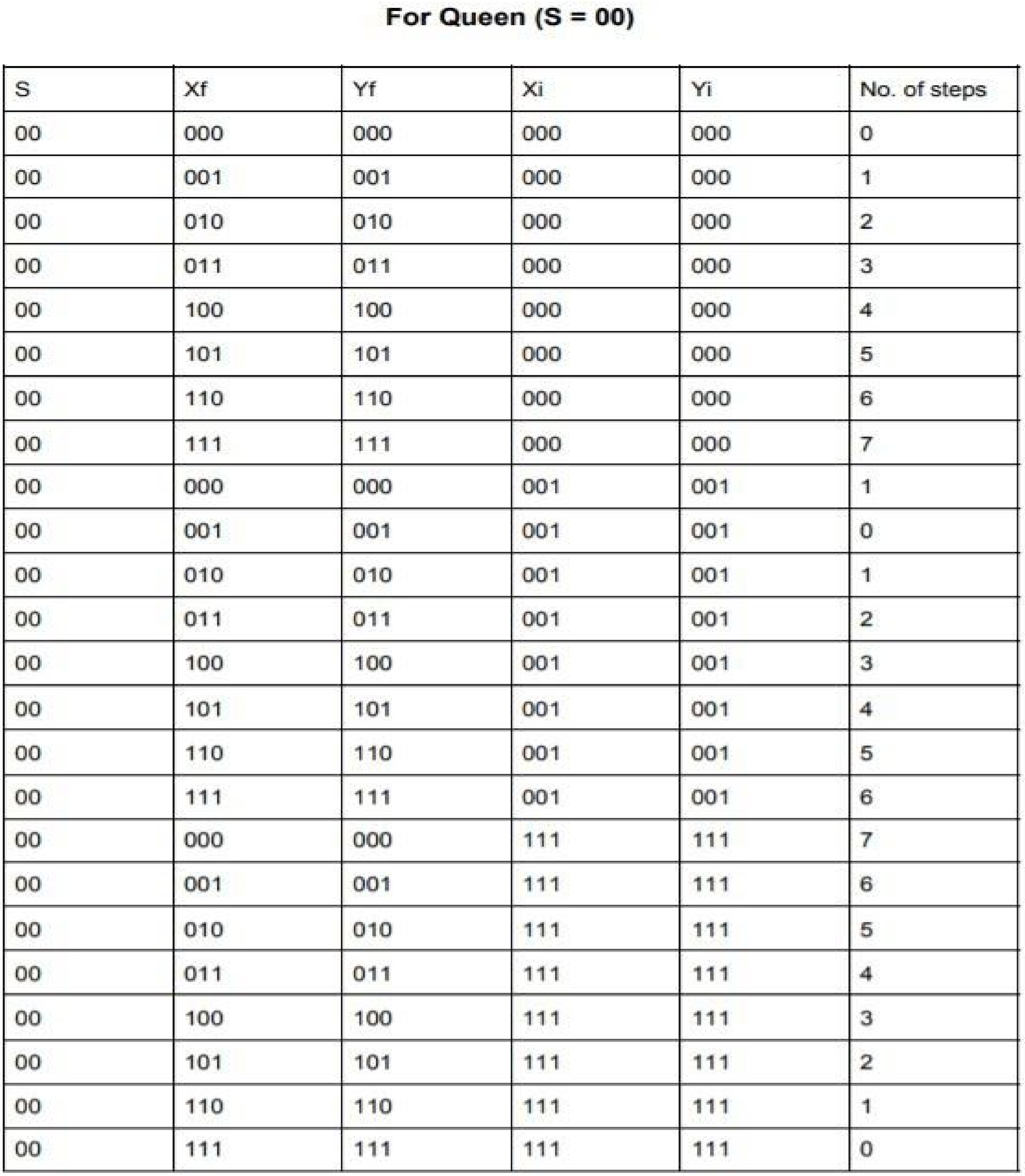
PINOUT:
INPUT PINS
1.MSB OF PIECE
2.LSB OF PIECE
3.A2 OF Xf
4.A1 OF Xf
5.A0 OF Xf
6.B2 OF Yf
7.B1 OF Yf
8.B0 OF Yf
9.A2 OF Xi
10.A1 OF Xi
11.A0 OF Xi
12.B2 OF Yi
13.B1 OF Yi
14.B0 OF Yi
15.Xf>Xi
16.Xf=Xi
17.Xf<Xi
18.Yf>Yi
19.Yf=Yi
20.Yf<Yi
OUTPUT PINS
21.Q2
22.Q1
23.Q0

LOGISIM IMPLEMENTATION OF THE CIRCUIT :



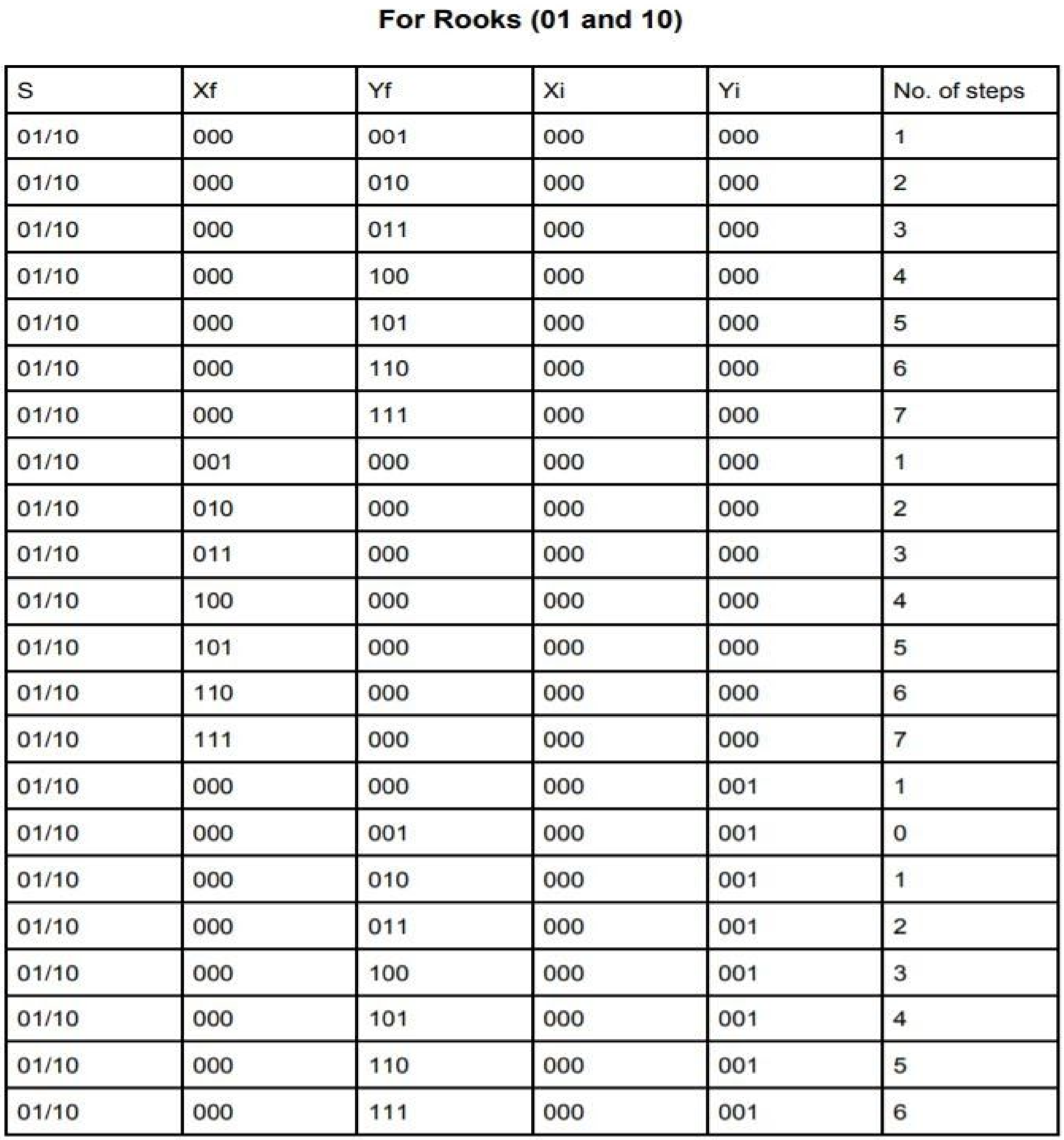
TRUTH TABLE FOR SOME INITIAL AND FINAL POSITION :

Considering the initial position of queen as (0,0) , (1,1) and (7,7).
And sample initial positions for Rook as (0,0) and (1,1).



For Queen (S = 00)

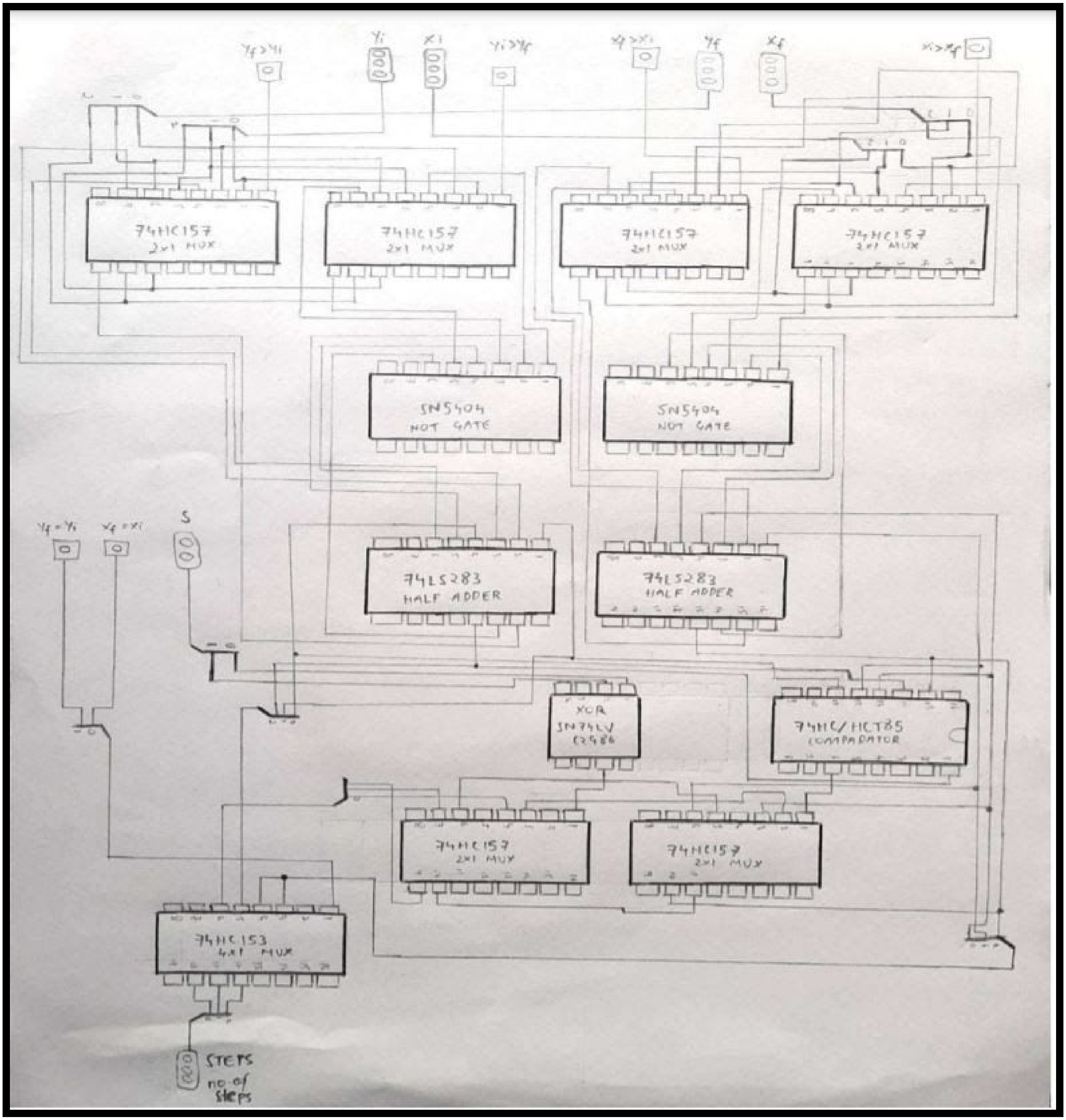
S	Xf	Yf	Xi	Yi	No. of steps
00	000	000	000	000	0
00	001	001	000	000	1
00	010	010	000	000	2
00	011	011	000	000	3
00	100	100	000	000	4
00	101	101	000	000	5
00	110	110	000	000	6
00	111	111	000	000	7
00	000	000	001	001	1
00	001	001	001	001	0
00	010	010	001	001	1
00	011	011	001	001	2
00	100	100	001	001	3
00	101	101	001	001	4
00	110	110	001	001	5
00	111	111	001	001	6
00	000	000	111	111	7
00	001	001	111	111	6
00	010	010	111	111	5
00	011	011	111	111	4
00	100	100	111	111	3
00	101	101	111	111	2
00	110	110	111	111	1
00	111	111	111	111	0



For Rooks (01 and 10)

S	Xf	Yf	Xi	Yi	No. of steps
01/10	000	001	000	000	1
01/10	000	010	000	000	2
01/10	000	011	000	000	3
01/10	000	100	000	000	4
01/10	000	101	000	000	5
01/10	000	110	000	000	6
01/10	000	111	000	000	7
01/10	001	000	000	000	1
01/10	010	000	000	000	2
01/10	011	000	000	000	3
01/10	100	000	000	000	4
01/10	101	000	000	000	5
01/10	110	000	000	000	6
01/10	111	000	000	000	7
01/10	000	000	000	001	1
01/10	000	001	000	001	0
01/10	000	010	000	001	1
01/10	000	011	000	001	2
01/10	000	100	000	001	3
01/10	000	101	000	001	4
01/10	000	110	000	001	5
01/10	000	111	000	001	6

IC IMPLEMENTATION:



<u>Steps countdown and position updater</u> :-

This machine counts down as the piece moves and then goes to a default state when reached.
The move button sets this counter to the respected started and then it counts down

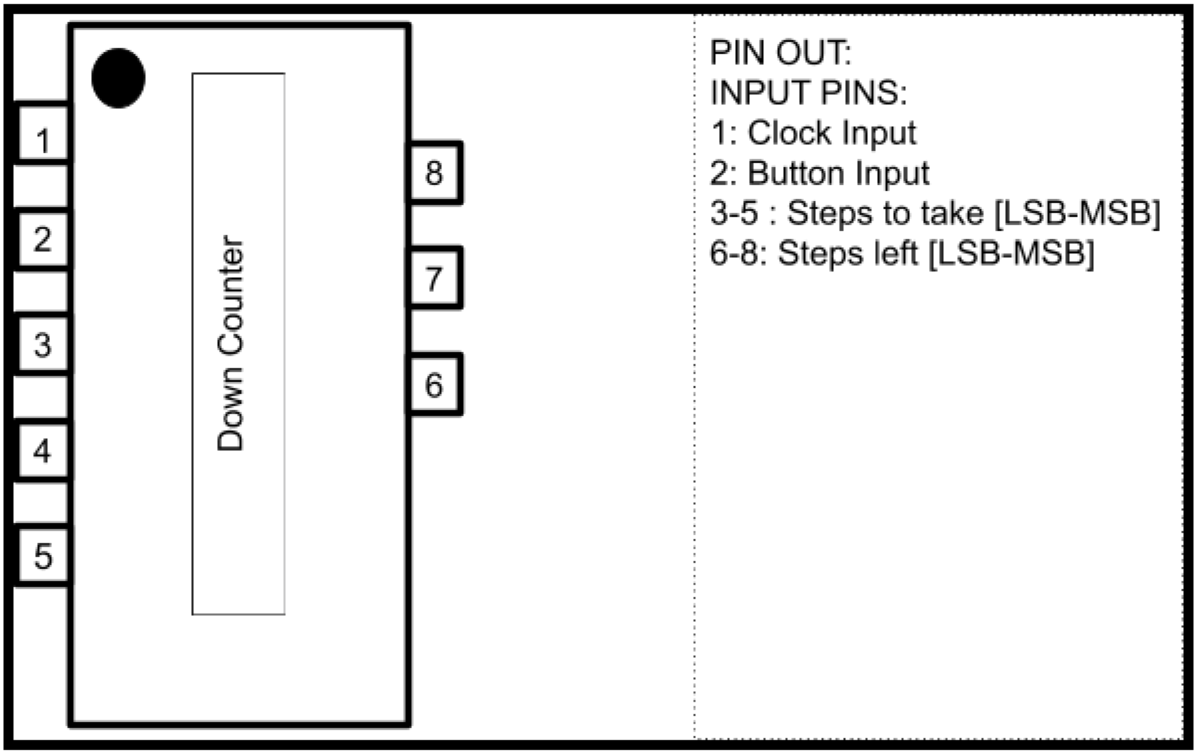
Inputs :

- Button Setting to starting state.
- Clock
- Number of Steps to move

Outputs :

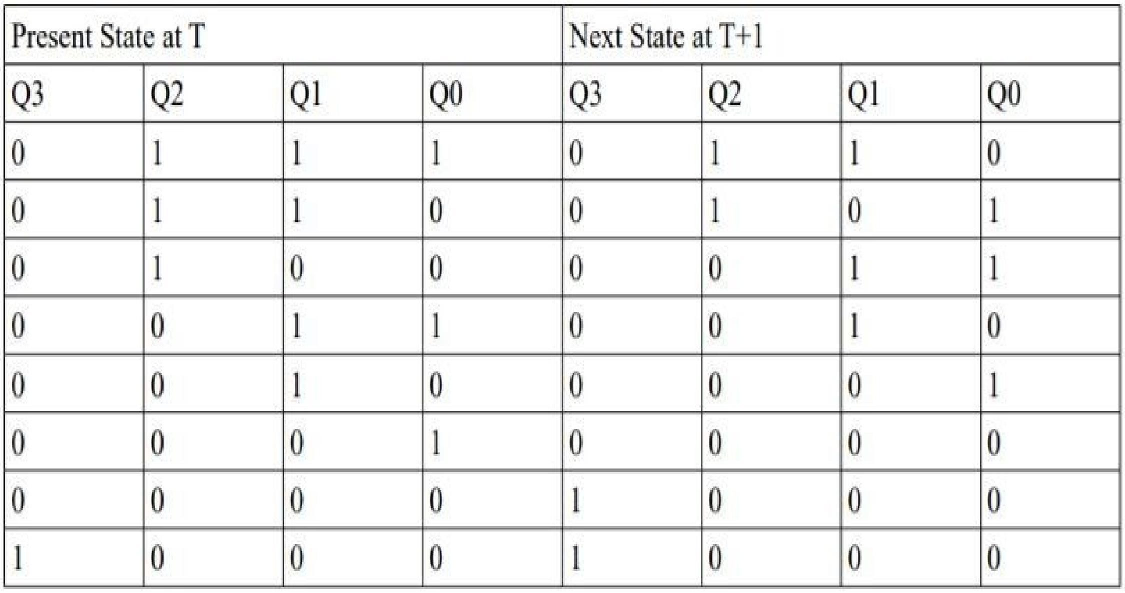
- Steps left to take

Pinout :



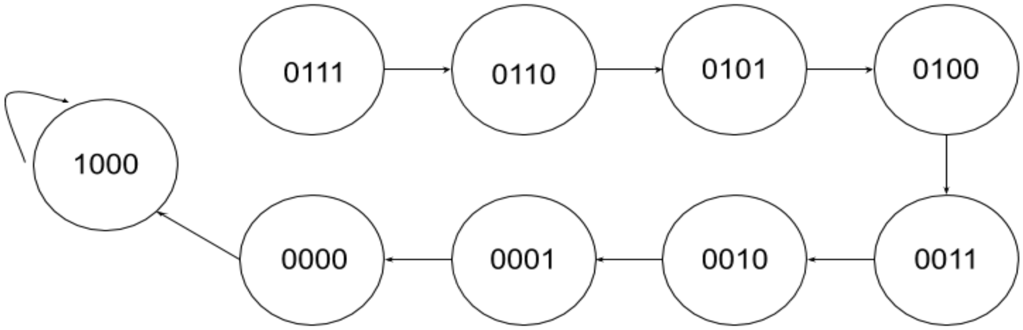
PIN OUT:
INPUT PINS:
1: Clock Input
2: Button Input
3-5 : Steps to take [LSB-MSB]
6-8: Steps left [LSB-MSB]

State Table:

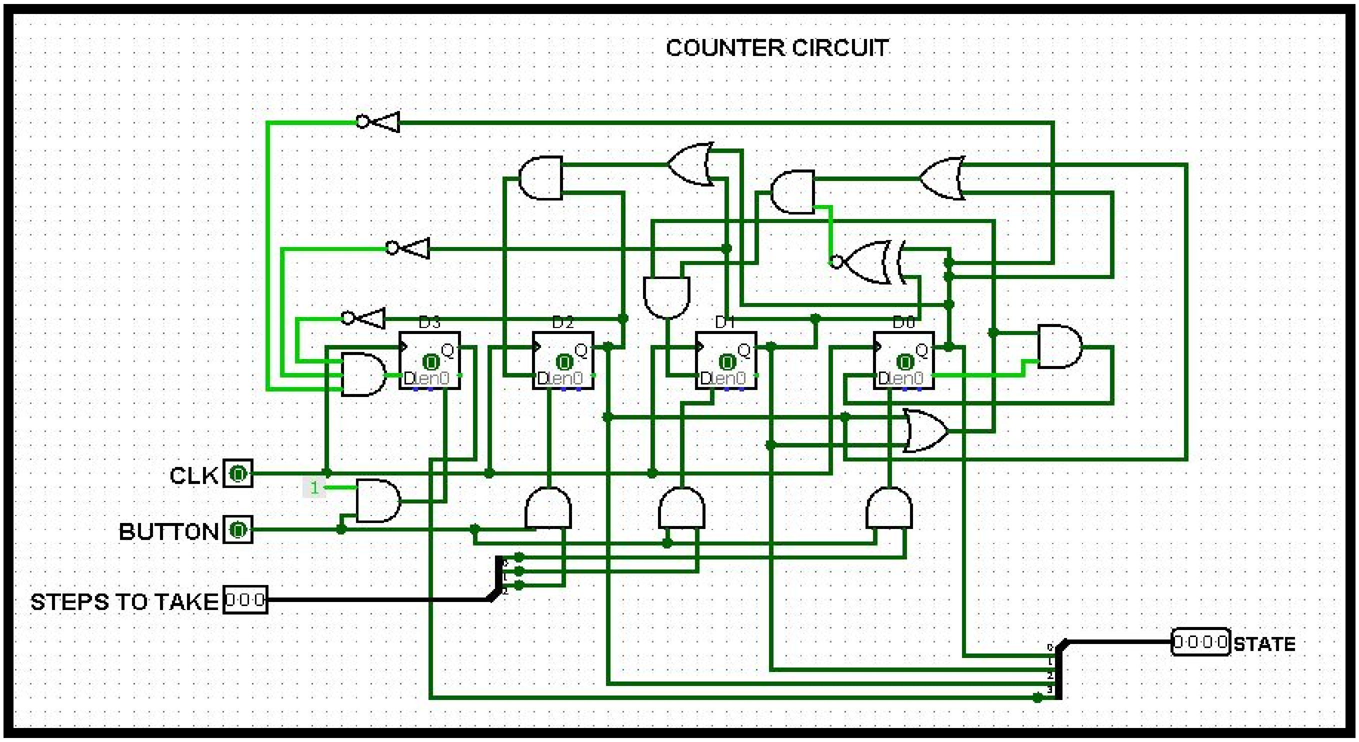


Present State at T				Next State at T+1			
Q3	Q2	Q1	Q0	Q3	Q2	Q1	Q0
0	1	1	1	0	1	1	0
0	1	1	0	0	1	0	1
0	1	0	0	0	0	1	1
0	0	1	1	0	0	1	0
0	0	1	0	0	0	0	1
0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0
1	0	0	0	1	0	0	0

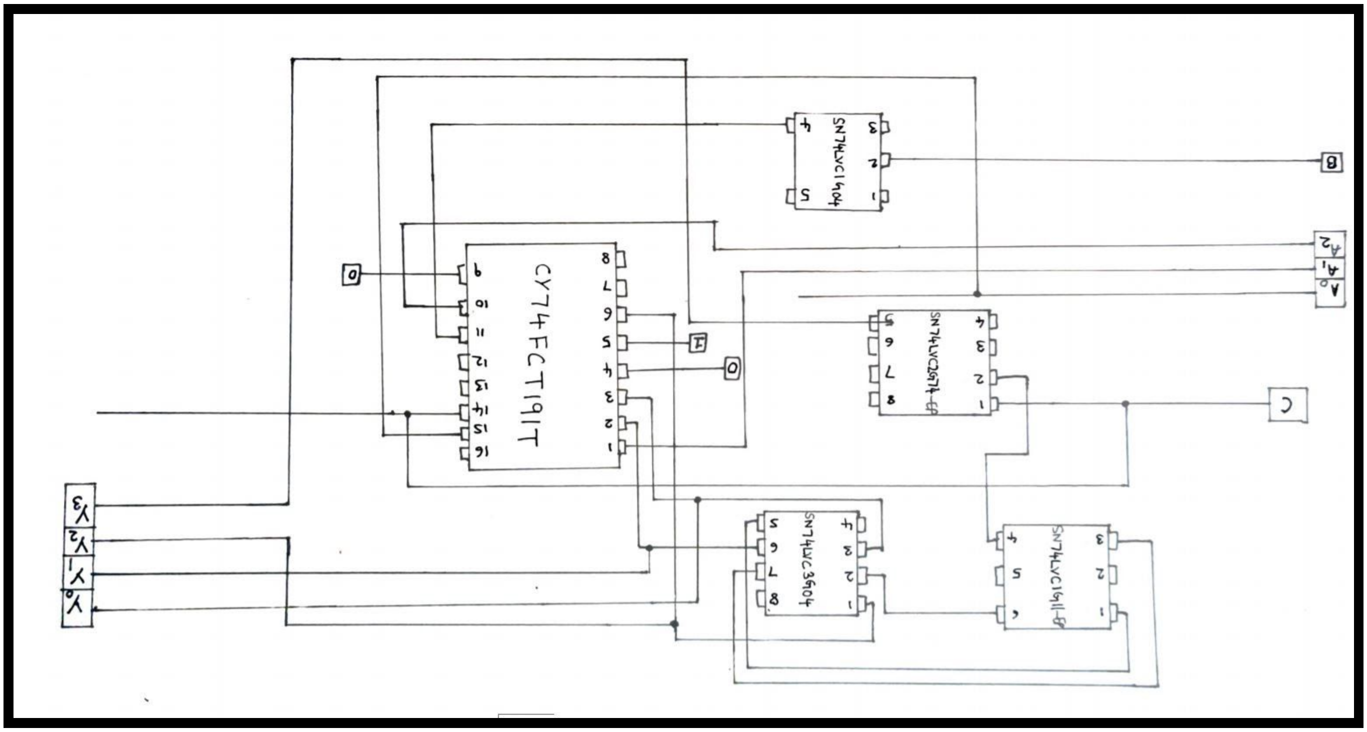
State Diagram :



Logisim Implementation:



IC Implementation :



Position saver :

Saves the position of all the pieces and supplies them for further movement.

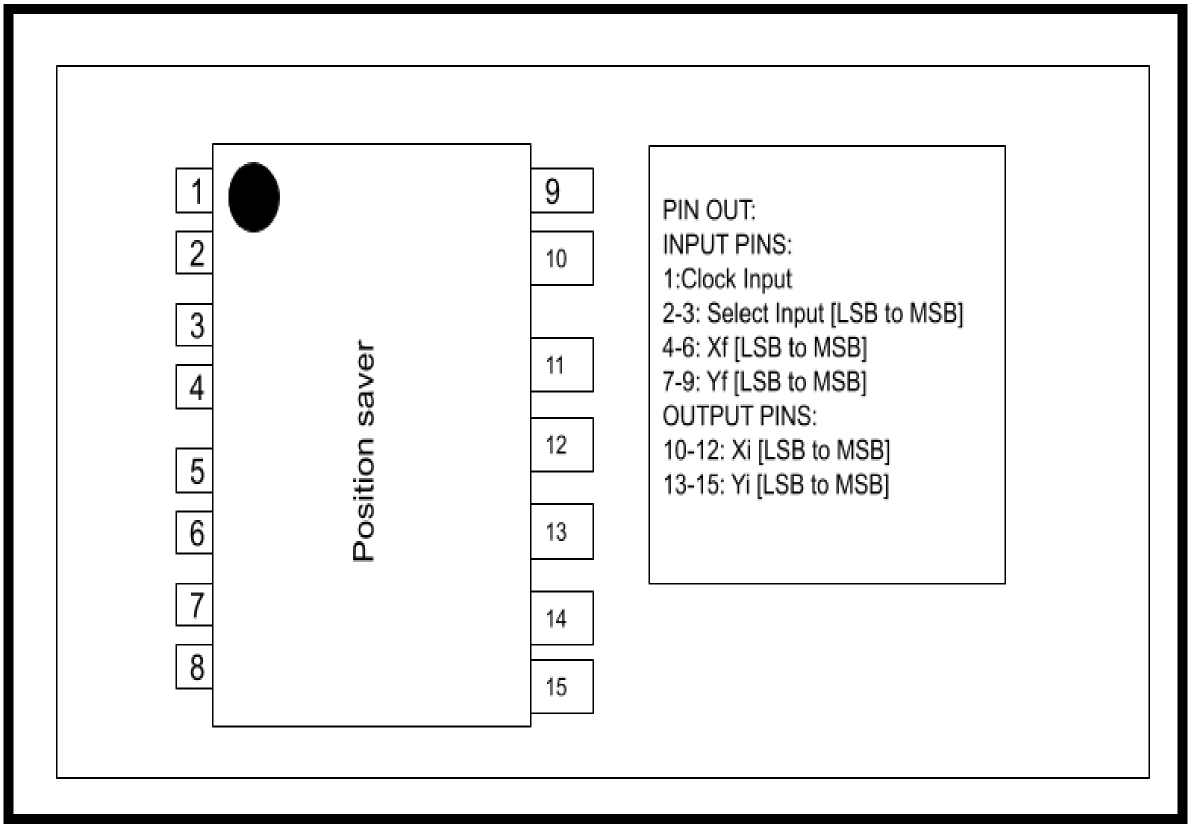
Inputs :

- Final Positions given by the user.
- Clock.
- Piece code.
- Set default button if default positions are needed or set values along with set button if user chooses to set the pieces to his preference.

Outputs :

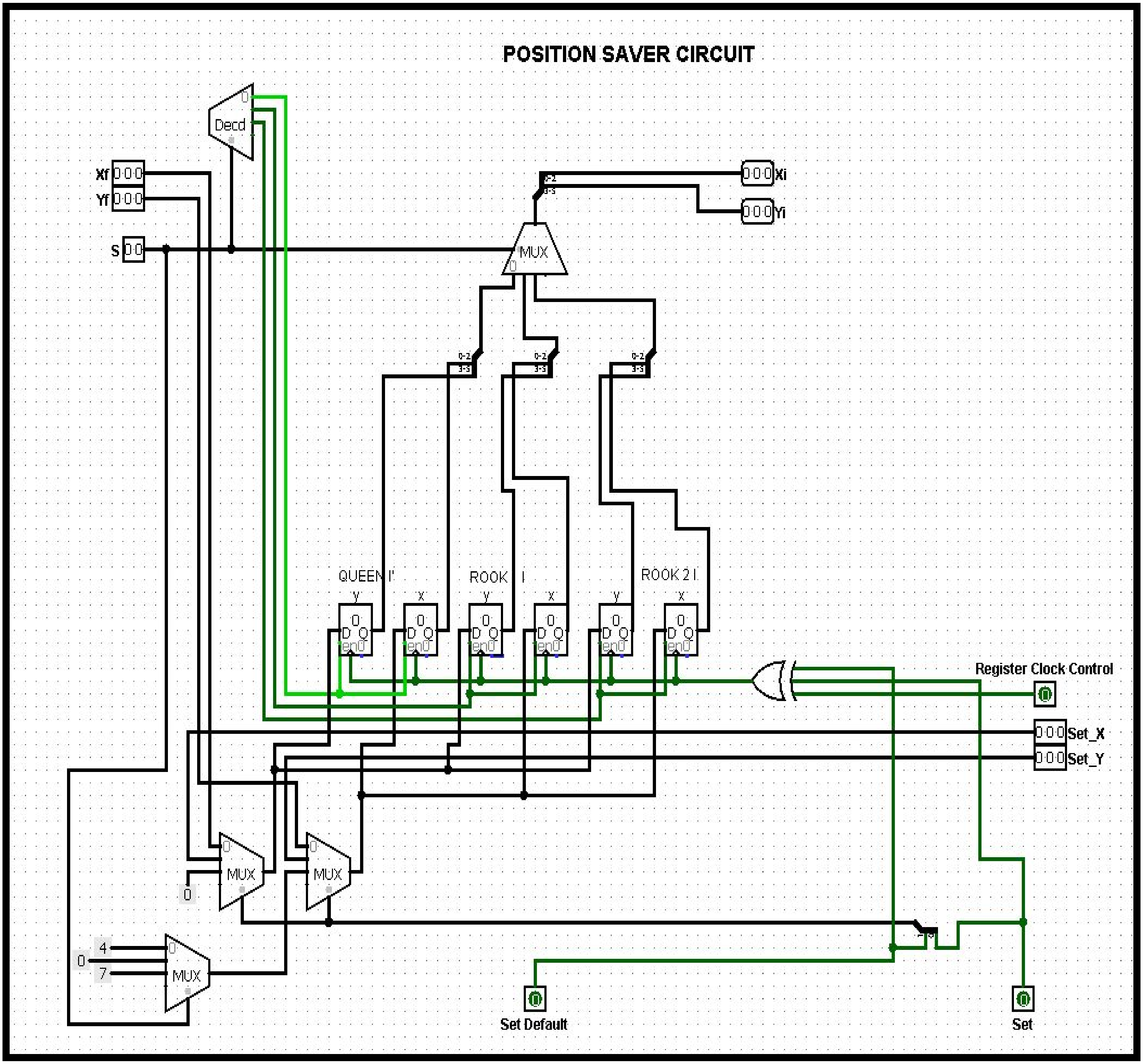
- Initial position

Pinout diagram :

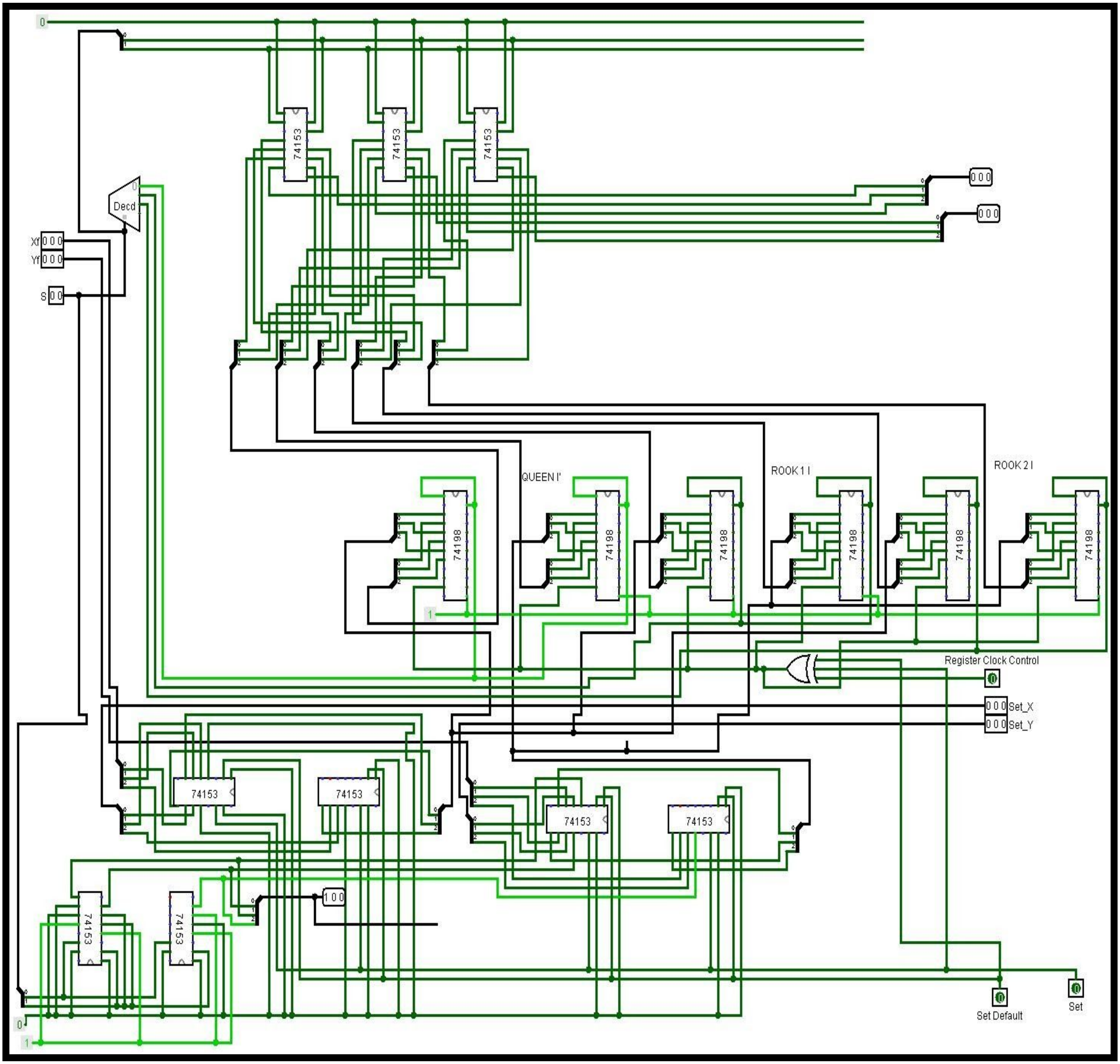


PIN OUT:
INPUT PINS:
1:Clock Input
2-3: Select Input [LSB to MSB]
4-6: Xf [LSB to MSB]
7-9: Yf [LSB to MSB]
OUTPUT PINS:
10-12: Xi [LSB to MSB]
13-15: Yi [LSB to MSB]

Logisim Implementation :

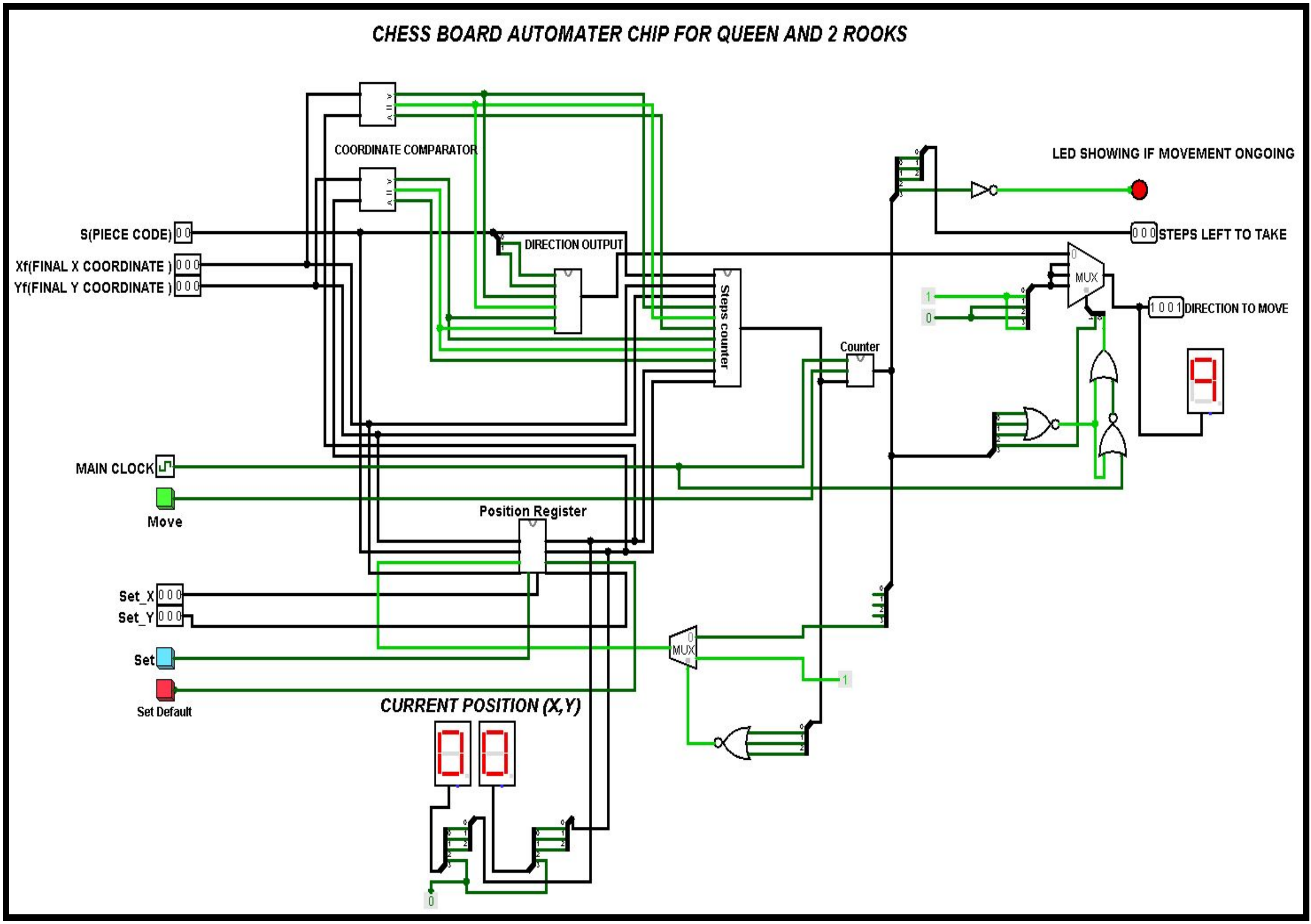


IC Implementation:

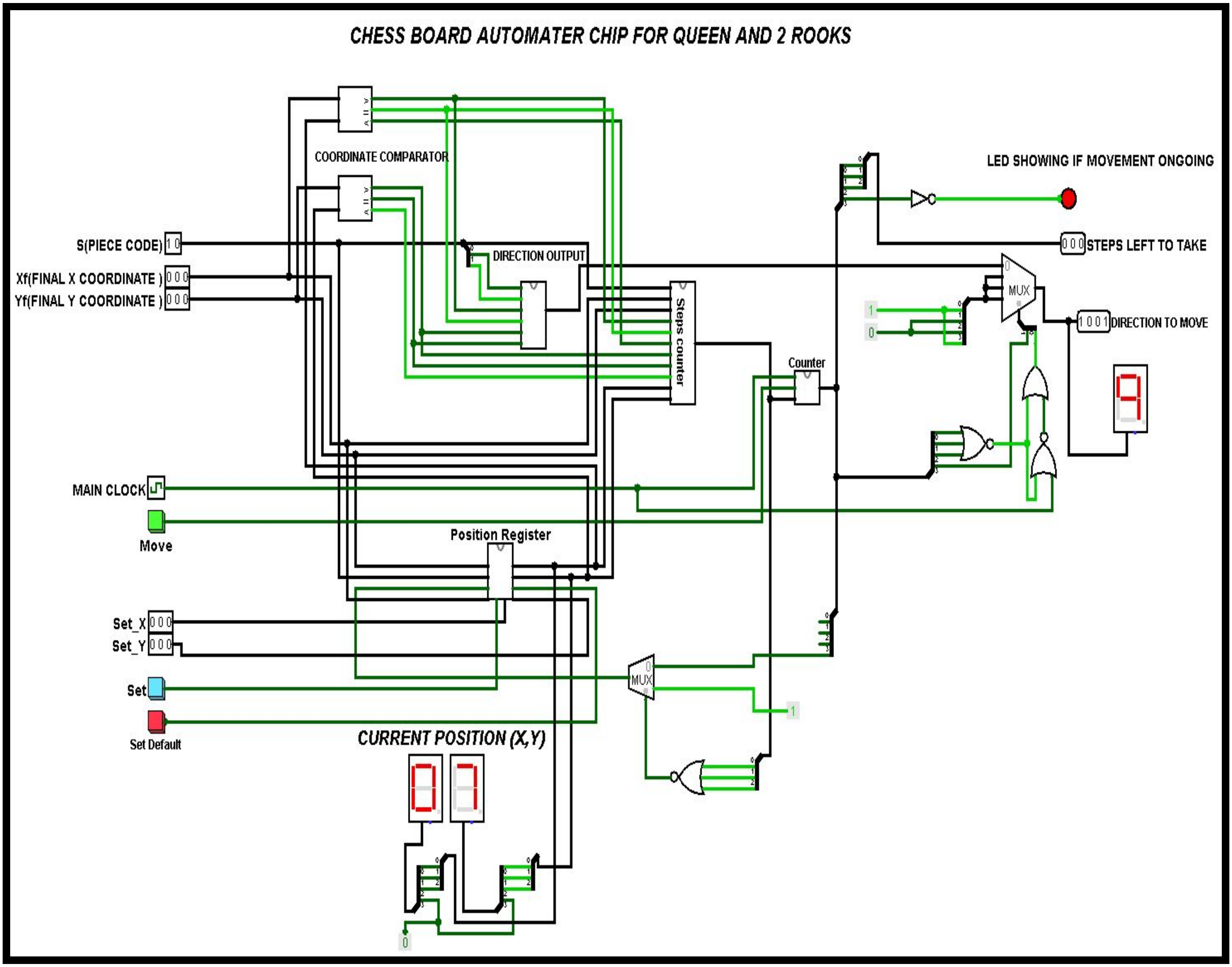


A Sample of Input/Output combination:

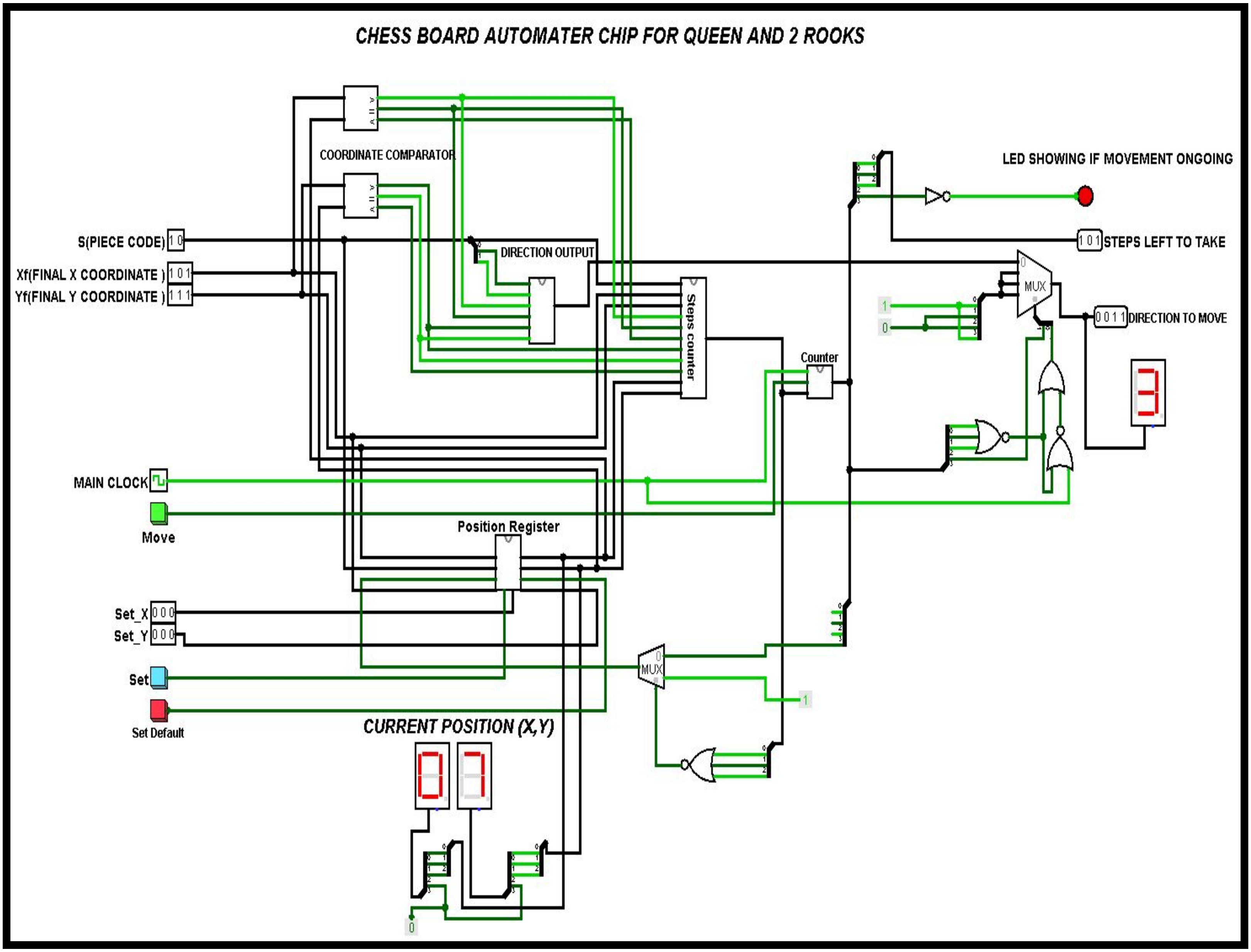
The sample piece that has been selected is the rook at (7,0). The final position given to the machine is (7,5) and the initial position is set using the set_default input:



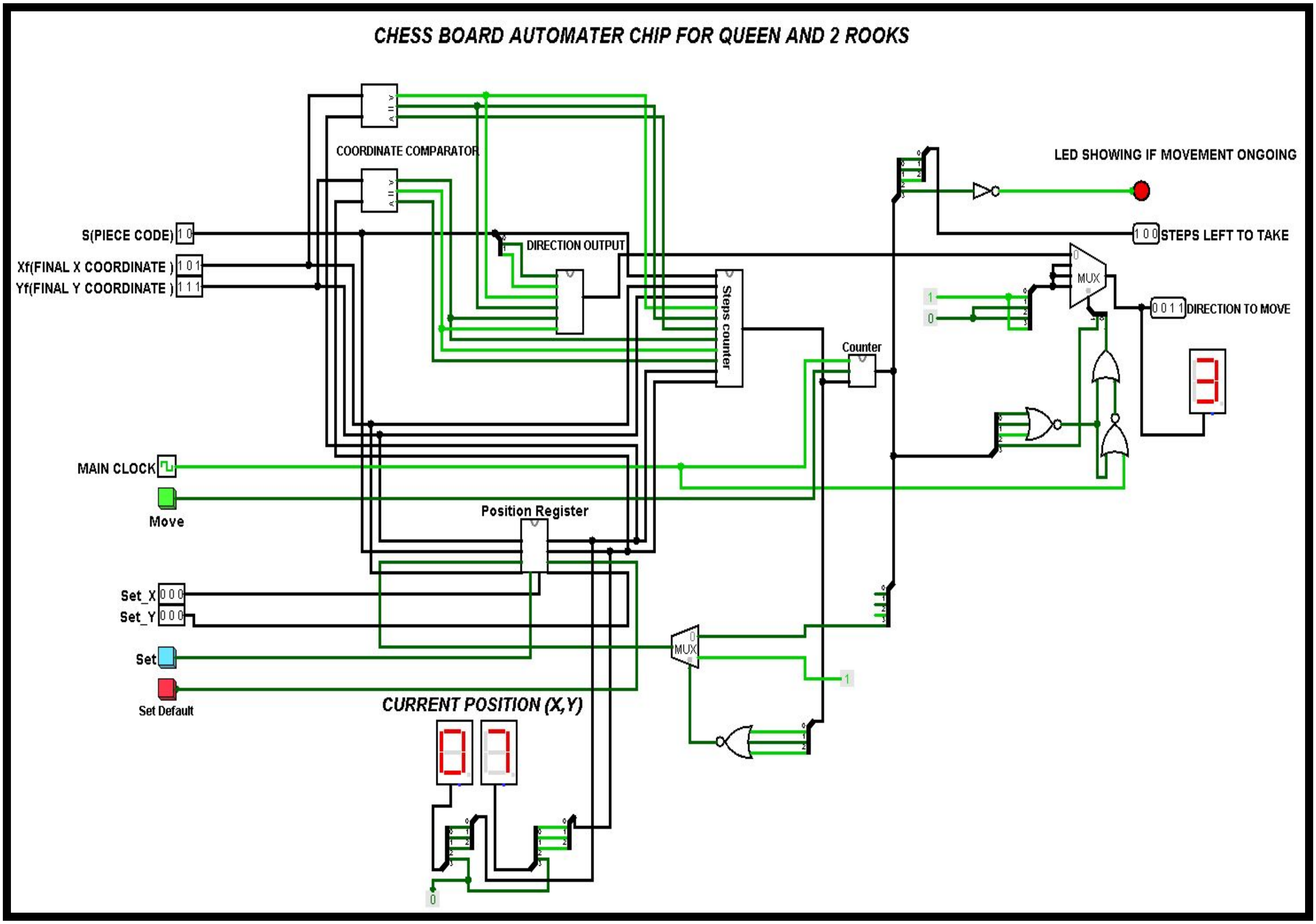
Start of machine. Current positions are (0,0) and the counter is in the stable state and no movement is displayed.



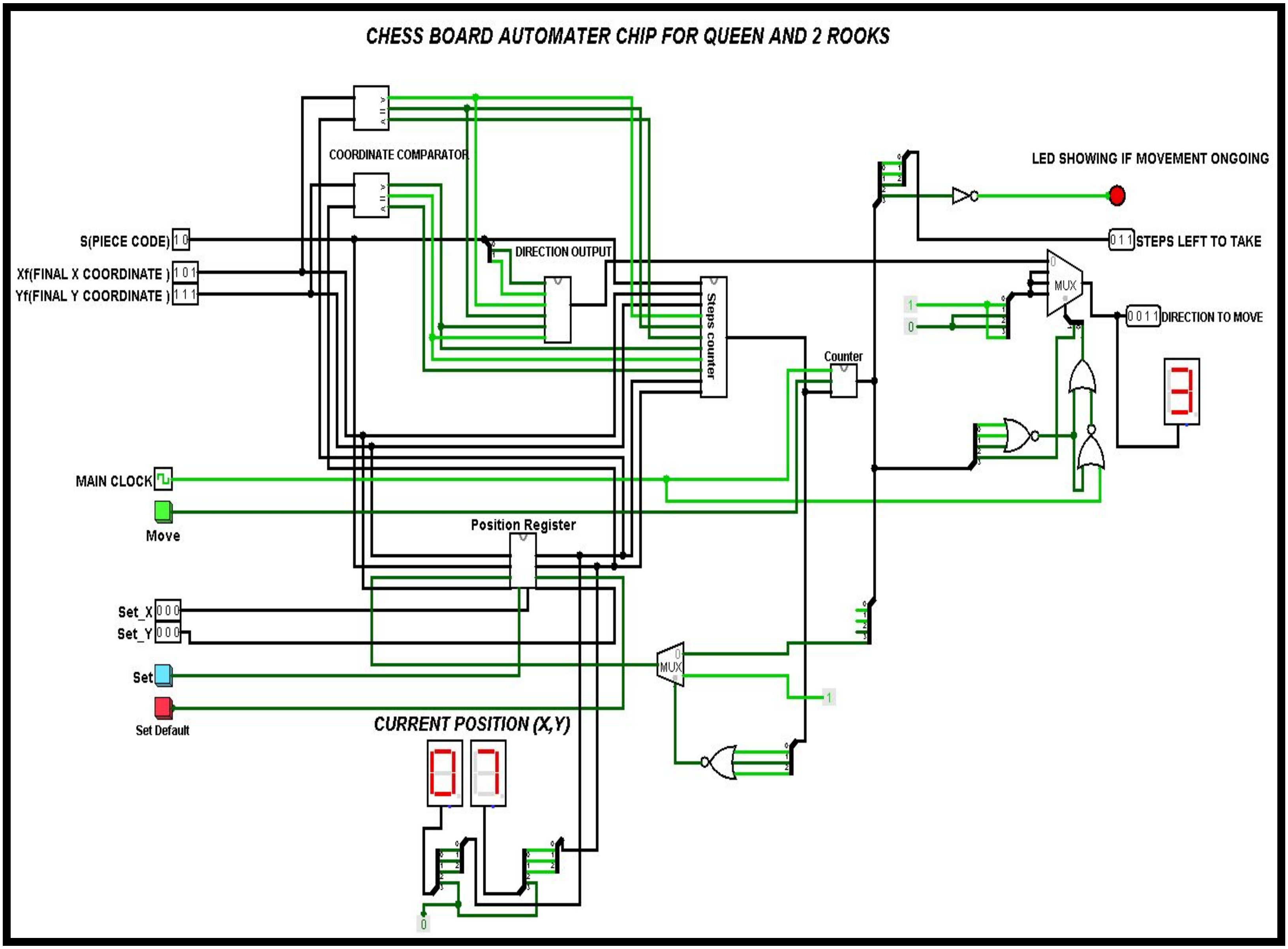
Rook has been selected using S and its initial position is set as (0,7) as a sample, using the Set default button.



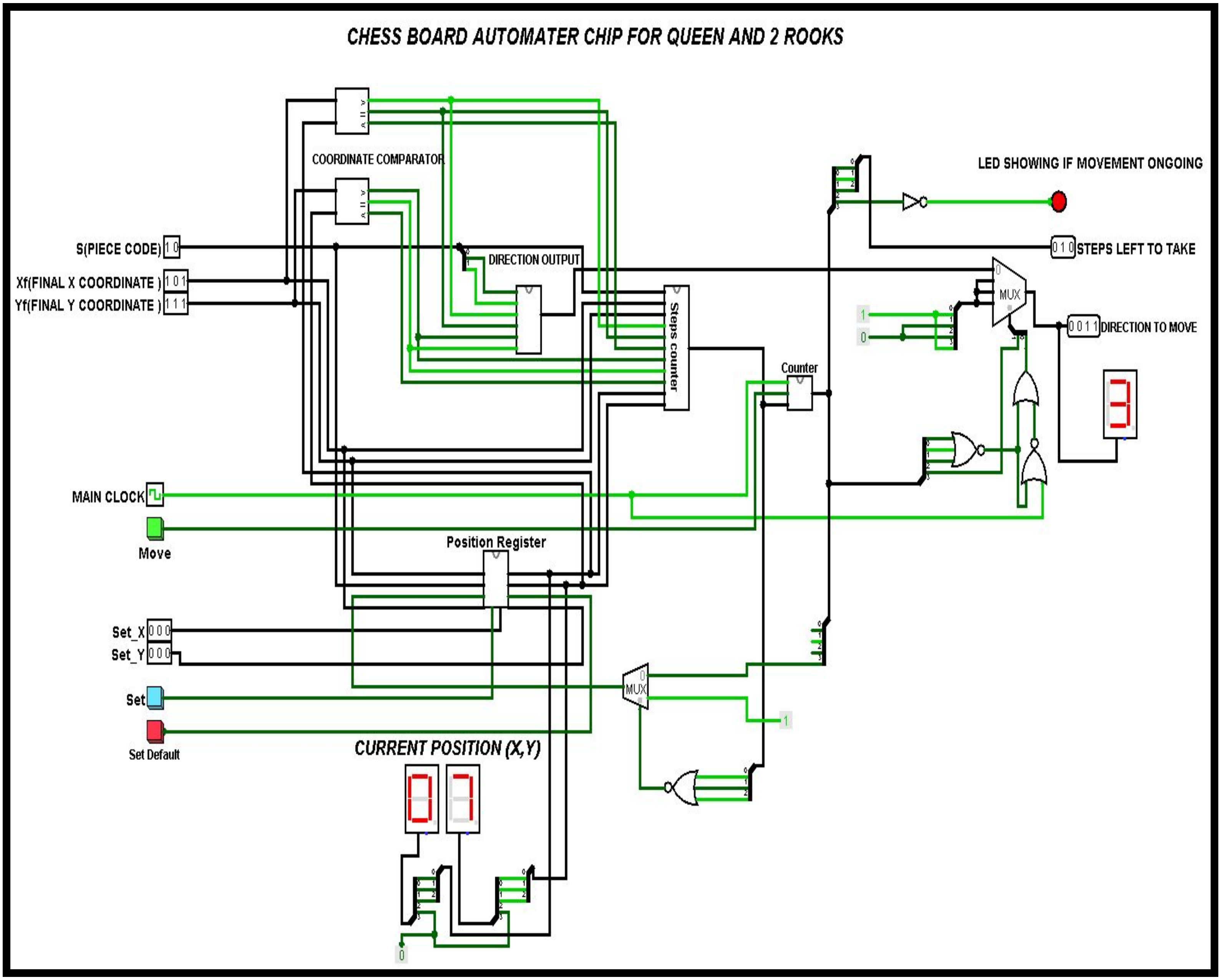
Final position has been entered and direction to be moved in is displayed in output as 3 and number of steps as 5



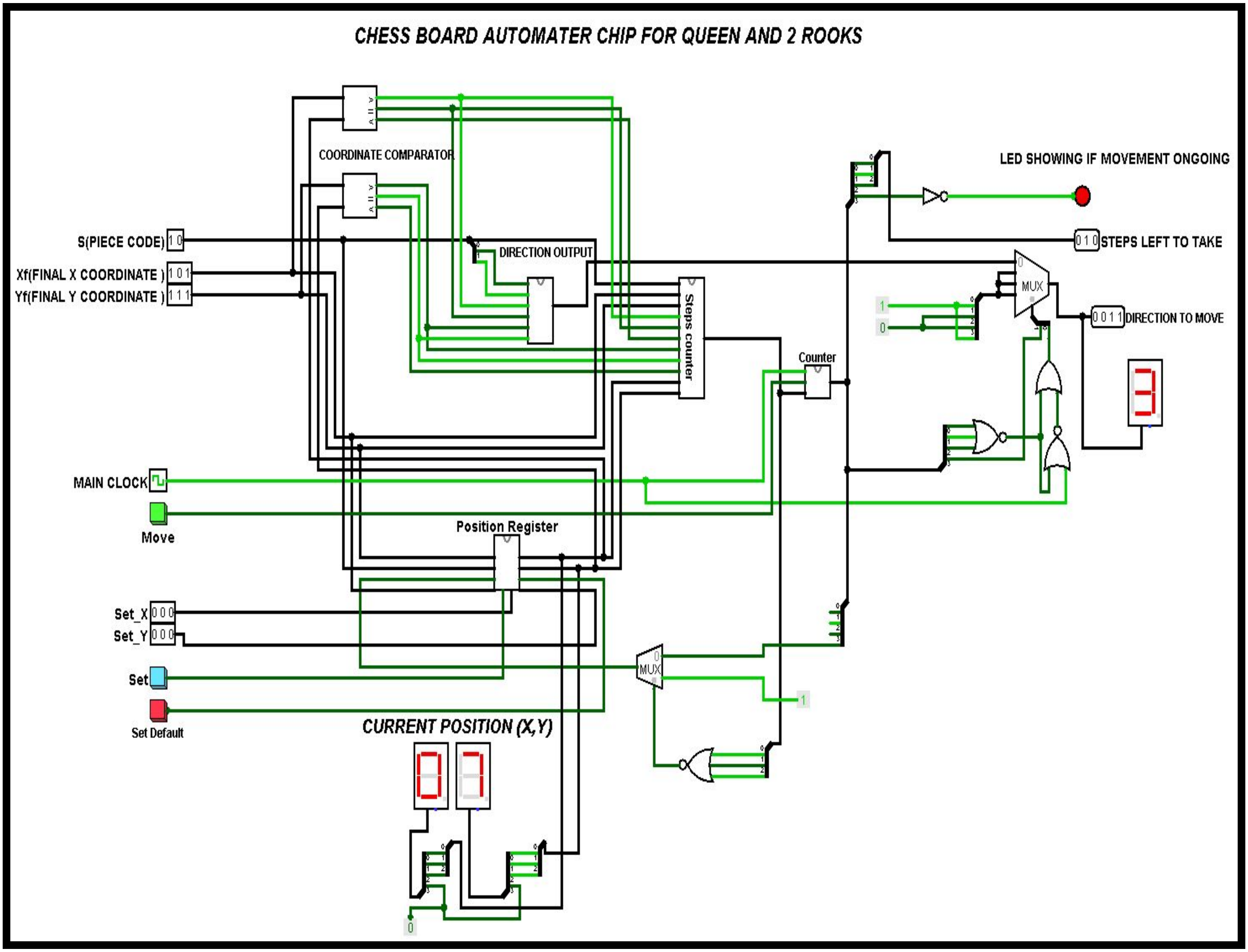
After the first clock cycle, the direction remains as 3 but the steps to take goes down to 4, as can be seen. One the negative edge of the clock cycle, the direction output changes to 9 to account for servo motor functionality.



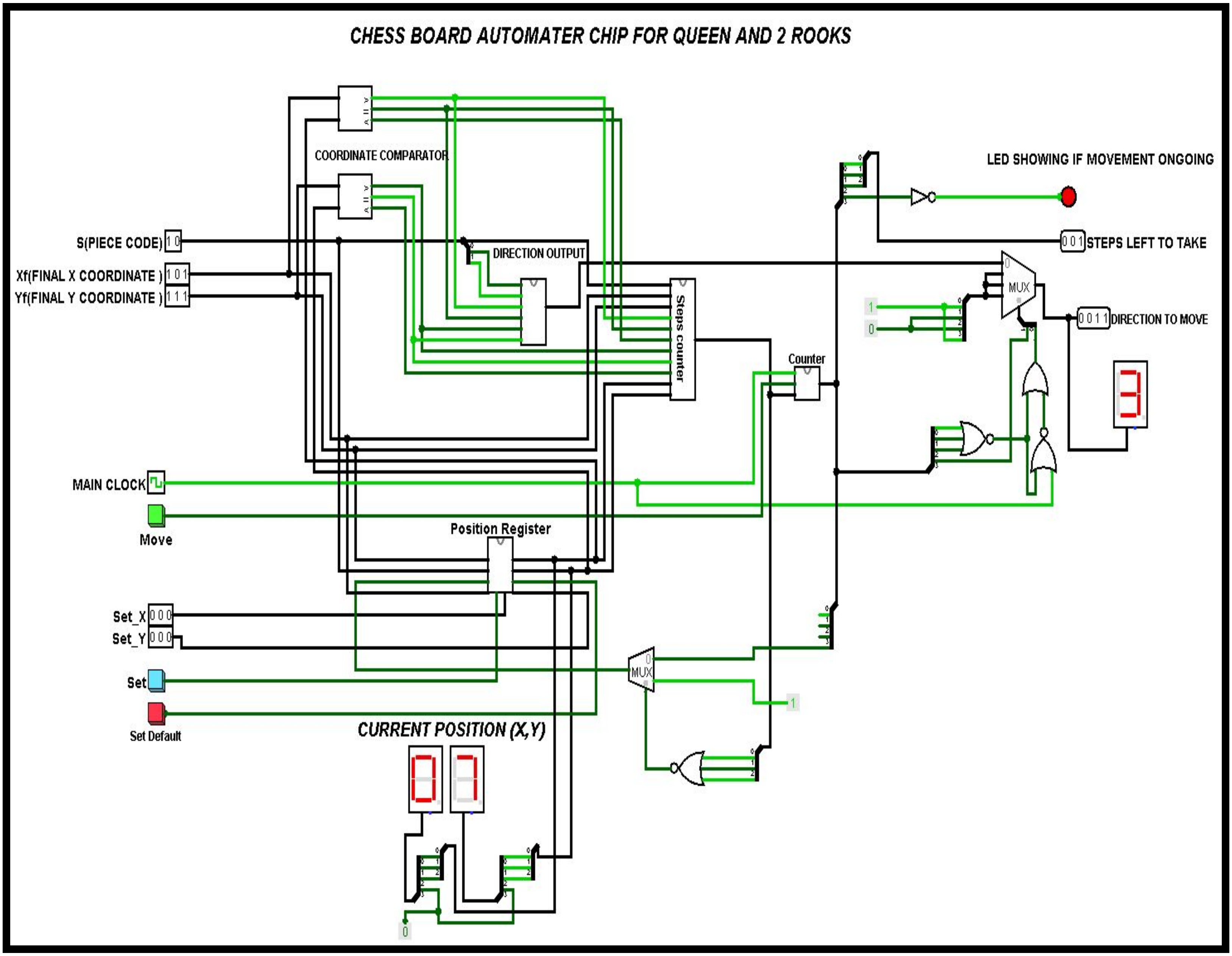
After the second clock cycle, the direction remains as 3 and steps goes down to 3.



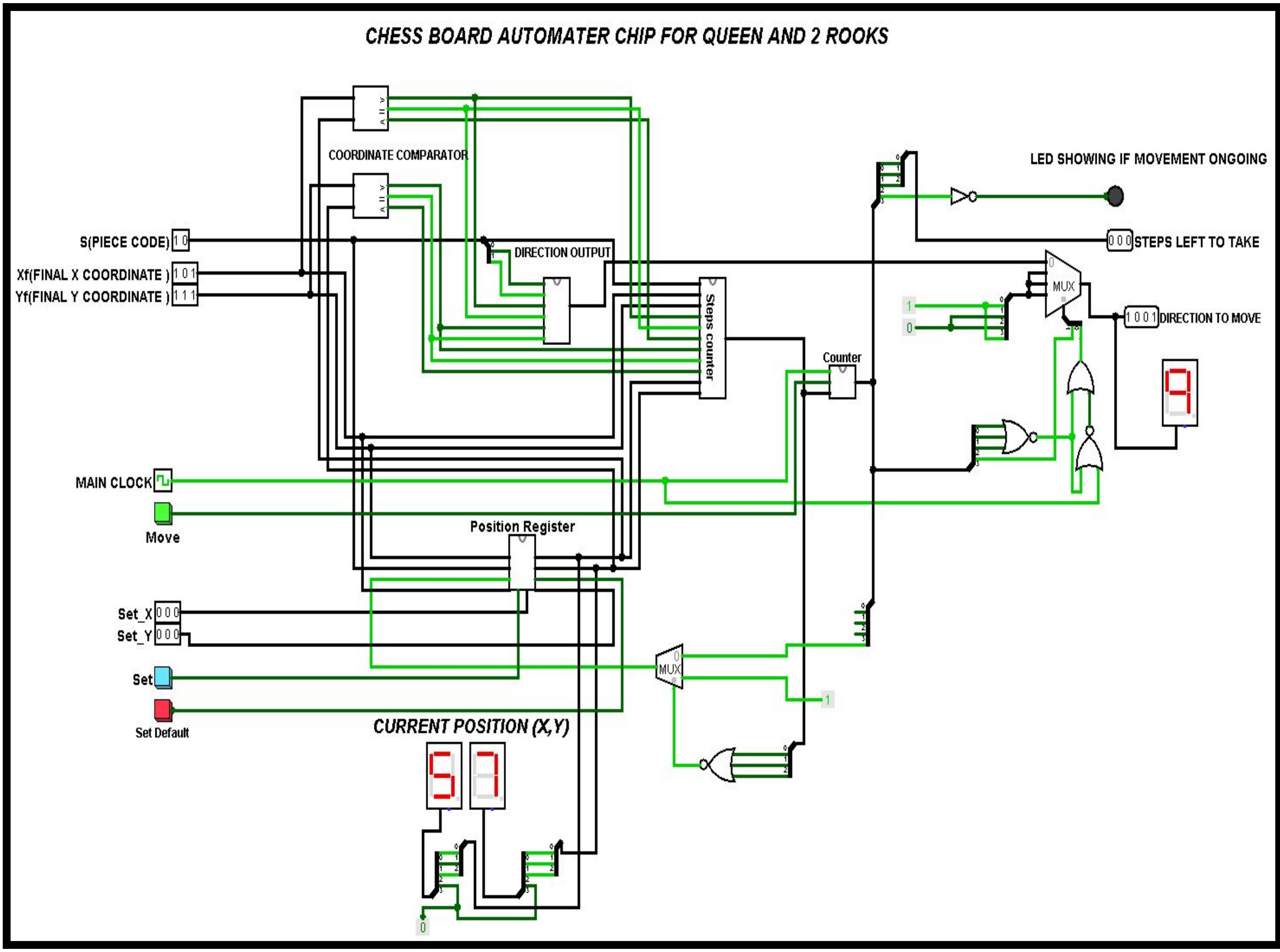
After the third clock cycle, direction remains as 3 and steps goes down to 3.



After the fourth clock cycle, the direction remains as 3 and steps goes down to 2

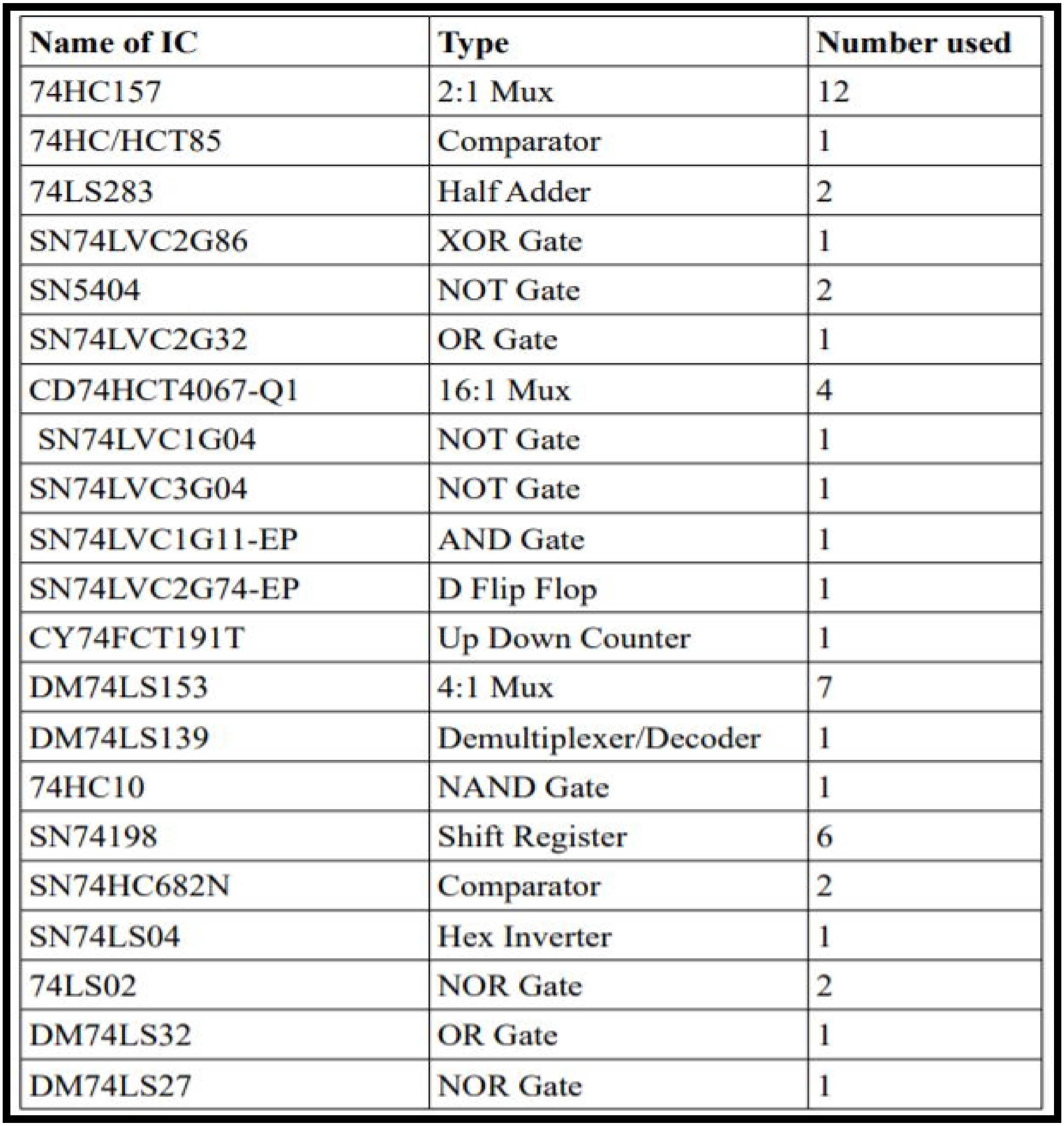


After the fifth clock cycle, the direction remains the same and steps goes down to 1



After all the required steps are completed, the current position has been updated to (5,7) and the direction checker now outputs no movement with the steps counter stable at 000

Bill of Materials :



Name of IC	Type	Number used
74HC157	2:1 Mux	12
74HC/HCT85	Comparator	1
74LS283	Half Adder	2
SN74LVC2G86	XOR Gate	1
SN5404	NOT Gate	2
SN74LVC2G32	OR Gate	1
CD74HCT4067-Q1	16:1 Mux	4
SN74LVC1G04	NOT Gate	1
SN74LVC3G04	NOT Gate	1
SN74LVC1G11-EP	AND Gate	1
SN74LVC2G74-EP	D Flip Flop	1
CY74FCT191T	Up Down Counter	1
DM74LS153	4:1 Mux	7
DM74LS139	Demultiplexer/Decoder	1
74HC10	NAND Gate	1
SN74198	Shift Register	6
SN74HC682N	Comparator	2
SN74LS04	Hex Inverter	1
74LS02	NOR Gate	2
DM74LS32	OR Gate	1
DM74LS27	NOR Gate	1

Datasheet links :

1) 2x1 MUX: 74HC157

2)4x1 MUX:DM74LS153

3)Comparator:74HC/HCT85

4)Half Adder:74LS283

5)XOR gate:SN74LVC2G86

6)Inverter:SN5404

7)OR Gate:SN74LVC2G32

8)16x1 MUX: CD74HCT4067-Q1

9)single NOT:SN74LVC1G04

10)triple NOT:SN74LVC3G04

11)AND gate:SN74LVC1G11

12)D-Flip Flop:SN74LVC2G74

13)Up/Down Counter:CY74FCT191T

14)4x1 MUX:DM74LS153

15)Demultiplexer:DM74LS138

16)NAND gate:74HC10

17)Shift Register:74198