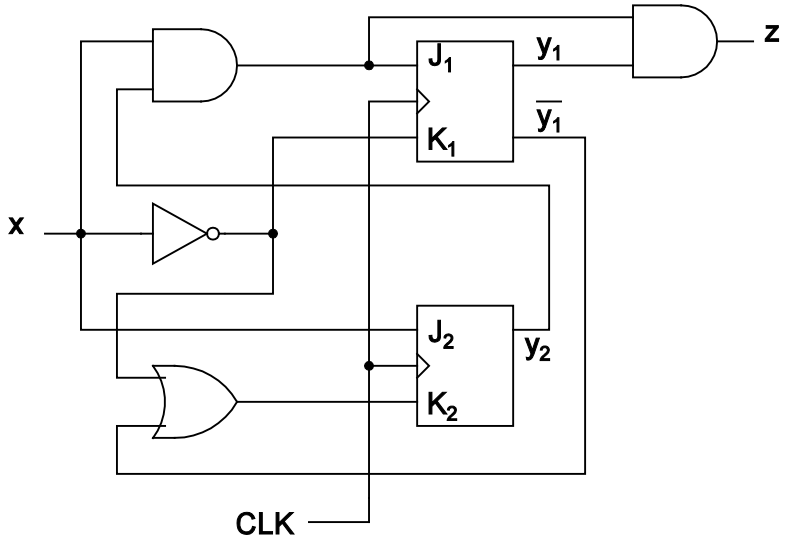
Lab 7: Sequential Circuit, Memory and Register

Objectives

- To analyze, design, construct, and test sequential circuits.
- Construct and test a single bit memory and Read-Only Memory(ROM)
- Test dot-matrix display.

Part I: Sequential Circuit Analysis (TTLs required: 7404, 7408, 7432, 7476)

You are to analyze the circuit below. The following steps will help you to learn how to analyze the sequential circuit.



Step 1: Boolean equations for J1, K1, J2, K2, and Z.
Step 2: Construct truth tables and K-maps.
Step 3: Construct state tables and state diagram.
Step 4: Sketch timing diagram for the input sequence X = 00111100.

Part I: Sequential circuit analysis

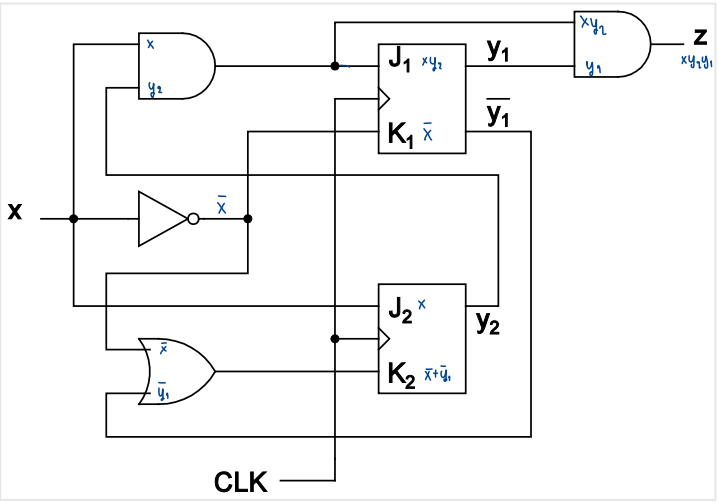
You are to analyze the circuit below. The following steps will help you to learn how to analyze the sequential circuit.

Step 1: Boolean equations for J_1, K_1, J_2, K_2, and Z.

$J_1 =$ xy_2 $K_1 =$ $\bar{x}$

$J_2 =$ x $K_2 =$ $\bar{x}+\bar{y}_1$

$z =$ xy_2y_1



Step 2: Truth table:

x	y_1	y_2	J_1	K_1	J_2	K_2	z
0	0	0	0	1	0	1	0
0	0	1	0	1	0	1	0
0	1	0	0	1	0	1	0
0	1	1	0	1	0	1	0
1	0	0	0	0	1	1	0
1	0	1	1	0	1	0	0
1	1	0	0	0	1	1	0
1	1	1	1	0	1	0	1

K-maps:

y_1y_2 \ x	0	1
00	0	0
01	0	1
11	0	1
10	0	0

J_1

y_1y_2 \ x	0	1
00	1	0
01	1	0
11	1	0
10	1	0

K_1

y_1y_2 \ x	0	1
00	0	1
01	0	1
11	0	1
10	0	1

J_2

y_1y_2 \ x	0	1
00	1	1
01	1	0
11	1	0
10	1	1

K_2

y_1y_2 \ x	0	1
00	0	0
01	0	0
11	0	1
10	0	0

z

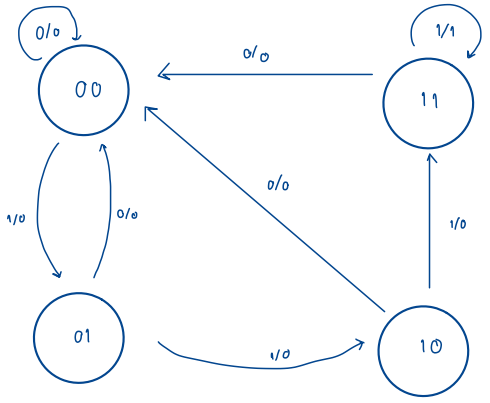
Step 3: State table:

$y_1y_2\backslash x$	0				1			
00	0	1	0	1	0	0	1	1
01	0	1	0	1	1	0	1	0
11	0	1	0	1	1	0	1	0
10	0	1	0	1	0	0	1	1
	J_1	K_1	J_2	K_2	J_1	K_1	J_2	K_2

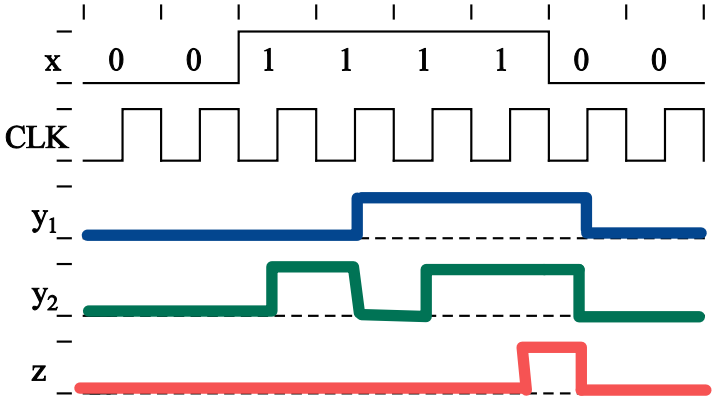
$y_1y_2\backslash x$	0		1	
00	0	0	0	1
01	0	0	1	0
11	0	0	1	1
10	0	0	1	1
	Y_1	Y_2	Y_1	Y_2

$y_1y_2\backslash x$	0	1
00	00/0	01/0
01	00/0	10/0
11	00/0	11/1
10	00/0	11/0
	Y_1Y_2/z	

State diagram



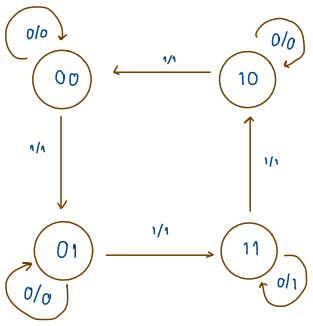
Step 4. Sketch timing diagram for the input sequence x = 00111100.



Part II: Sequential Circuit Design (TTLs required: 7404, 7408, 7432, 7476)

Design a sequential circuit with two JK flip-flops A and B and one input x. When x = 0, the state of the circuit remains the same. When x = 1, the circuit goes through the state transitions from 00 to 01, to 11, to 10, back to 00, and repeats. The output z = 1 when either x = 1 or flip-flop outputs = 11; otherwise z = 0.

Step 1: State diagram:



Step 2: Truth table:

i/p	P.S.		N.S.		A		B		o/p
x	y_A	y_B	y_A	y_B	J_A	K_A	J_B	K_B	z
0	0	0	0	0	0	X	0	X	0
0	0	1	0	1	0	X	X	0	0
0	1	0	1	0	X	0	0	X	0
0	1	1	1	1	X	0	X	0	1
1	0	0	0	1	0	X	1	X	1
1	0	1	1	1	1	X	X	0	1
1	1	0	0	0	X	1	0	X	1
1	1	1	1	0	X	0	X	1	1

K-maps:

y_Ay_B \ x	0	1
00	0	0
01	0	1
11	X	X
10	X	X

$J_A = x\bar{y}_B$

y_Ay_B \ x	0	1
00	X	X
01	X	X
11	0	0
10	0	1

$K_A = x\bar{y}_B$

y_Ay_B \ x	0	1
00	0	1
01	X	X
11	X	X
10	0	0

$J_B = x\bar{y}_A$

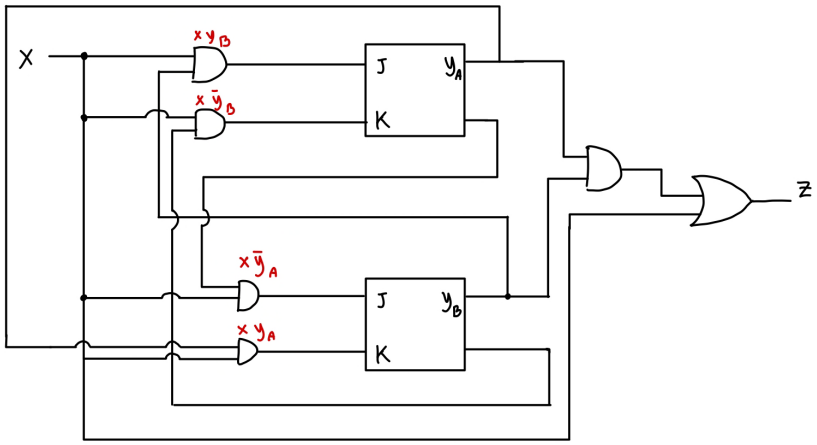
y_Ay_B \ x	0	1
00	X	X
01	0	0
11	0	1
10	X	X

$K_B = x y_A$

y_Ay_B \ x	0	1
00	0	1
01	0	1
11	1	1
10	0	1

$z = x + y_A y_B$

Step 4: Circuit Diagram and Proteus simulation:



Part III: Single-bit memories

RAM

Figure 8.1 (a) shows a single-bit memory or a binary cell (BC) shown in Figure 8.1(b). The select input enables the cell for reading or writing, and the $Read/\overline{Write}$ input determines the operation of the cell when it is selected (or enabled). A "1" in the $Read/\overline{Write}$ input provides the read operation by forming a path from the latch to the output terminal. A "0" in the $Read/\overline{Write}$ input provides the write operation by forming a path from the input terminal to the latch.

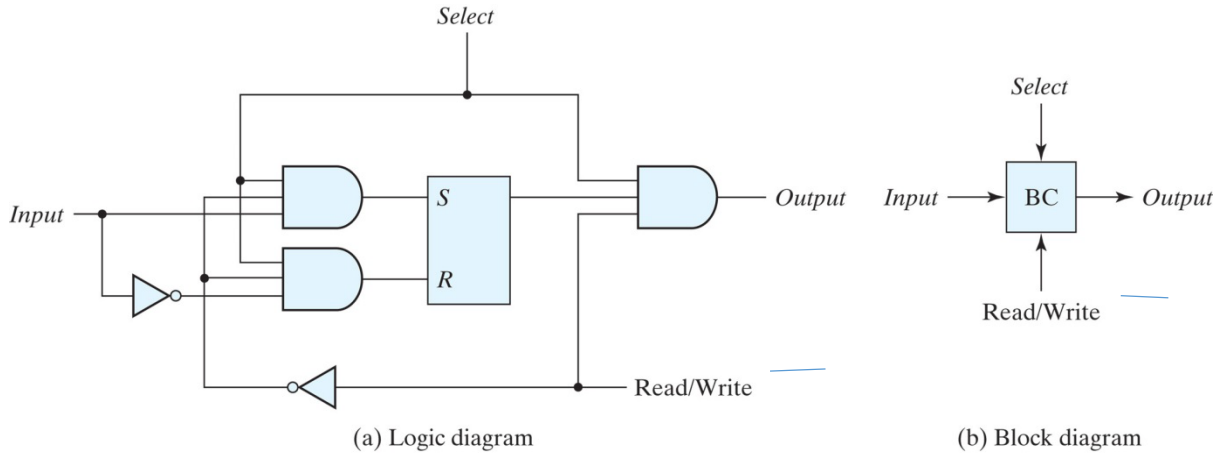


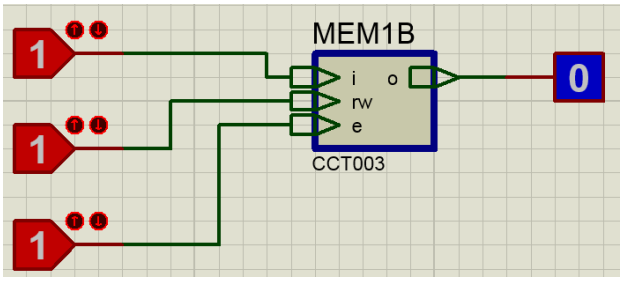
Figure 8.1 Memory cell.

ROM

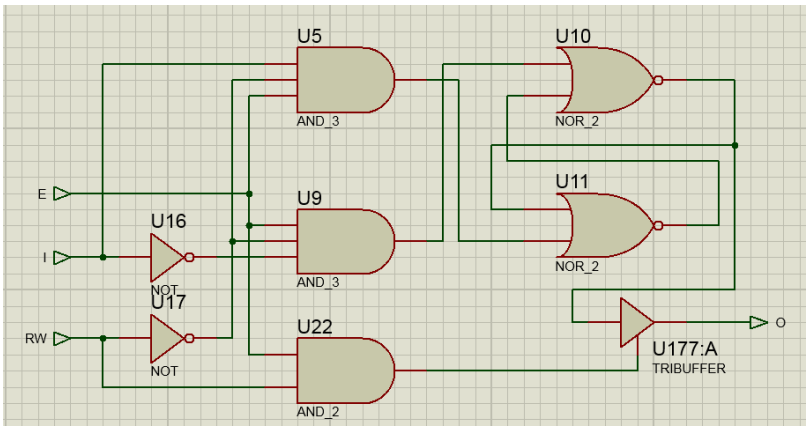
A read-only memory (ROM) is essentially a memory device in which permanent binary information is stored. The binary information must be specified by the designer and is then embedded in the cell.

Procedure

1. Use Proteus program to create a single-bit memory cell.



I is the input; RW= "1" for "read", RW= "0" for "write"; E=1 for "enable"; O is the output.



2. Simulate the single-bit RAM and fill in the table 1.

Table 1: RAM

E	RW	I	O	Operation
0	x	x	?	unenable
1	0	0	?	write
1	0	1	?	write
1	1	0	1	read
1	1	1	1	read

3. Build a single bit ROM as shown below. Simulate the read operation and fill in the table

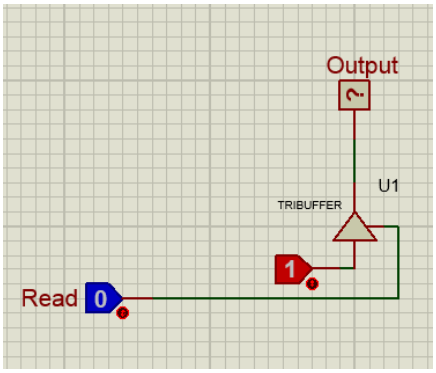
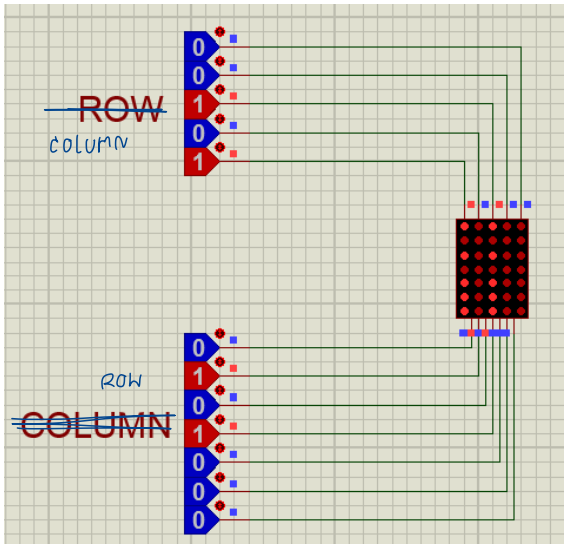


Table 2: ROM

Data	Read	Output	Operation
0	0	?	write
0	1	0	read
1	0	?	write
1	1	1	read

Part IV: Dot matrix display

4. Build and test the dot matrix display. Explain how to light up a particular dot at *row-i* and *column-j*.



Explain how to light up a particular dot at row-i and column-j

column เมื่อให้เลข เป็น 1 ไฟในแถวแนวตั้ง แถวนั้นๆ จะติด และหากให้เป็น 0 ไฟในแถวนั้นจะดับ

ROW เมื่อใส่เลข เป็น 1 ไฟในแถว แนวนอน จะ ดับ แต่ถ้า เป็น 0 ไฟแถวนั้น จะ ติด

Part V: 5x7 ROM

5. Use 2732 ROM chip to build a 5x7 ROM chip driven by the 7493 binary counter. Use input clock B and output QB, QC, and QD as a 3-bit binary counter to count from 0 to 7. Program the ROM so that the dot-matrix display a letter "A". Show the results to TA.

1 1 0 1 1
1 0 1 0 1
0 1 1 1 0
0 1 1 1 0
0 0 0 0 0
0 1 1 1 0
0 1 1 1 0

ROM → 0 | 60 5b 3b 5b | 60 --

TA Signature: .....