# IC TESTER

Submitted by: Group 67


Group Members:

| | |
|---|---|
| Aaranya Prasad | 2019A7PS0107G |
| Aaryan Lad | 2019AAPS0255G |
| Mansi Doshi | 2019A8PS0493G |
| Mitali Doshi | 2019A7PS0064G |
| Sarvesh Garge | 2019AAPS0233G |
| Svadhi Jain | 2019A8PS0573G |


Date: 19/4/2021

# Contents

## Design Problem

Design a Microprocessor based Tester to test the logical functioning of the following chips

1. 7400
2. 7408
3. 7432
4. 7486
5. 747266

The IC to be tested will be inserted in a 14 pin ZIF socket.

The IC number is to be entered via a keyboard.

The keyboard has keys 0-9, backspace, enter and test.

The user places the IC in the ZIF socket, closes it – then enters the IC No, followed by the enter key.

The IC No. is displayed on the 7-segment display.

The testing will start once the user presses the “TEST” key.

After Test the result PASS/FAIL must be displayed on the 7-segment display.

## Assumptions

1. Only the test key can be pressed after the enter key.
2. No key can be pressed while testing the IC.
3. No key can be pressed while the display shows PASS/FAIL

## User Guide

1. Attach the IC in the ZIF Socket properly and lock it.
2. Enter the IC number via the keypad.
3. Press the Enter Key after the IC number is entered via the keypad.
4. The IC Number will be displayed on the 7-segment display.
5. Press the Test key for testing the IC working.
6. If the IC number entered is the same as the IC attached and the IC working properly, PASS shall be displayed.
7. For all the other cases, FAIL shall be displayed.
8. The backspace key can be used in case any wrong digit is typed via the keypad.

## Components used with justification wherever required

| Sr No | COMPONENT | DESCRIPTION | QUANTITY |
|---|---|---|---|
| 1. | 8086 | Microprocessor | 1 unit |
| 2. | 8255A | 1 unit for interfacing with the keyboard input and 7 segment display<br><br>1 unit for interfacing with the ZIF socket | 2 units |
| 3. | 8284A | | 1 unit |
| 4. | 2716 | (2K ROM chips) We would need 2k even bank and 2k odd bank ROM in the starting and then a 2k even bank and 2k odd bank ROM at the end. | 4 units |
| 5. | 6116 | (2K RAM chips) We would need a 2k odd and 2k even bank for RAM. We need RAM for stack and temporary storage of data | 2 units |
| 6. | LS138 | (Active low DMUX 3X8) Selecting between even and odd banks of RAM and ROM. | 2 unis |
| 7. | NL7SZ19 | (Active low DMUX 1X2) Selecting between the 2 8255As | 1 unit |
| 8. | LS245 | System Data bus | 2 unit |
| 9. | LS244 | System Control bus | 1 unit |
| 10. | LS373 | System Address bus | 3 unit |
| 11. | 7 segment anode display | To display digits and PASS/ FAIL | 6 units |
| 12. | 14 pin ZIF Socket | To insert the chip to be tested | 1 unit |
| 13. | 7400 | IC of Quadruple 2-input positive NAND Gates | 1 unit |

| | | | |
|---|---|---|---|
| 14. | 7408 | IC of Quadruple 2-input positive AND Gates | 1 unit |
| 15. | 7432 | IC of Quadruple 2-input positive OR Gates | 1 unit |
| 16. | 7486 | IC of Quadruple 2-input positive XOR Gates | 1 unit |
| 17. | 747266 | IC of Quadruple 2-input positive XNOR Gates | 1 unit |
| 18. | 10k ohm resistor | Keyboard Interface | 4 units |
| 19. | 150 ohm resistor | Display interface for the input | 7 units |
| 20. | 690 ohm resistor | Display Interface for the enable | 6 units |
| 21. | 2N2905 | (PNP BJT)<br>7 Segment Display Interface | 6 units |
| 22. | SPST Push Buttons | Keyboard | 16 units |
| 23. | Switch (SW-SPDT-MOM) | Interactive SPDT Switch with Momentary Action | 1 unit |
| 24. | 2 Input Positive OR gate | For control signal | 4 units |
| 25. | NOT gate | For control signals | 2 units |

# Address Map

## Memory Map

Minimum available size of ROM chip = 2k
Minimum available size of RAM chip = 2k

Since 8086 has a 16-bit data bus and the chips have an 8-bit data bus, we require even and odd banks for memory addressing

ROM1 – $00000_H$ – $00FFF_H$

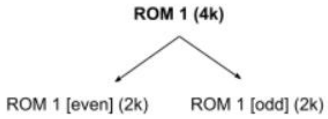
RAM 1– $01000_H$ – $01FFF_H$

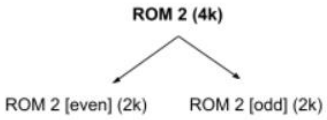
ROM2 – $FF000_H$ – $FFFFF_H$

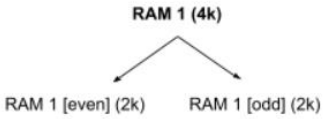
Number of Chips

2K ROM Chips (chip number) = 4

2K RAM Chips (chip number) = 2




Memory Allocation

| | | |
|---|---|---|
| ROM 1 | even | 00000h, 00002h... - 00FFEh |
| | odd | 00001h, 00003h... - 00FFFh |
| RAM 1 | even | 01000h, 01002h... - 01FFEh |
| | odd | 01001h, 01003h... - 01FFFh |
| ROM 2 | even | FF000h, FF002h... - FFFFEh |
| | odd | FF001h, FF003h... - FFFFFh |

ROM1- 00000h-00FFFh

| A19 | A18 | A17 | A16 | A15 | A14 | A13 | A12 | A11 | … | A0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | … | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | … | 1 |

RAM1- 01000h-01FFFh

| A19 | A18 | A17 | A16 | A15 | A14 | A13 | A12 | A11 | … | A0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | … | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | … | 1 |

ROM2- FF000h-FFFFFh

| A19 | A18 | A17 | A16 | A15 | A14 | A13 | A12 | A11 | … | A0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | … | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | … | 1 |

We see that address lines A12, A13 and A14 are unique for every component. Hence, we use these lines as select lines to our decoders for the memory banks

# I/O Map

We have two 8255's being used, for interfacing with the I/O Devices. One of them is used for keyboard and 7 segment display interfacing and the other is used for ZIF Socket interfacing.

The address allocation for the 2 8255A's are as follows:

1. **8255A-1**

- PORT A: 00H
- PORT B: 02H
- PORT C: 04H
- CONTROL REGISTER: 06H

The table below shows the address of the ports of 8255A-1 in binary:

| A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |

**2. 8255A-2**

- PORT A: 10H
- PORT B: 12H
- PORT C: 14H
- CONTROL REGISTER: 16H

The table below shows the address of the ports of 8255A-2 in binary:

| A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |

From the above two tables, we can clearly see that A4 input is different for the different 8255's but remains the same internally. This can be used as input signal to a 1:2 decoder to differentiate the 8255's.

**Port Allocation:**

**8255A-1:**

- PORT A: 8 pins as output pins which are inputs to the 7-segment display
- PORT B: 6 pins as output pins which are the enable inputs for the 7-segment display.
- PORT C:

1. UPPER PORT C: 4 pins as output pins which are input to the keyboard as the column inputs.
2. LOWER PORT C: 4 pins as input pins from the keyboard as the row outputs.

**8255A-2:**

- PORT A: 6 pins as output pins which are inputs to the ZIF Socket pins: P1, P2, P12, P13, P5, and P9.
- PORT B: 2 pins as input pins which are output from P3 and P11 from the ZIF Socket.
- PORT C:
    1. UPPER PORT C: 2 pins: P6 and P8.
    2. LOWER PORT C: 2 pins: P4 and P10

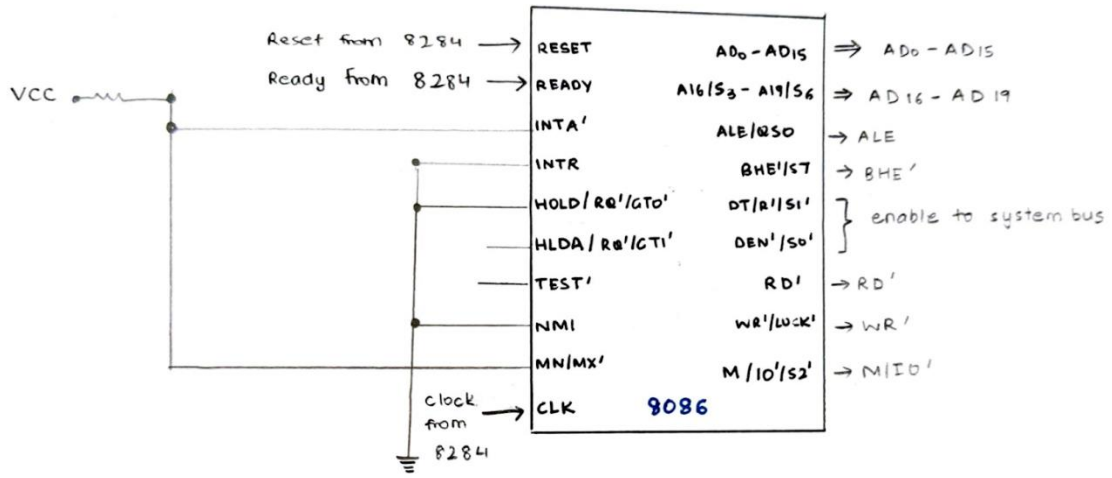
**I/O pins of IC's to be tested:**

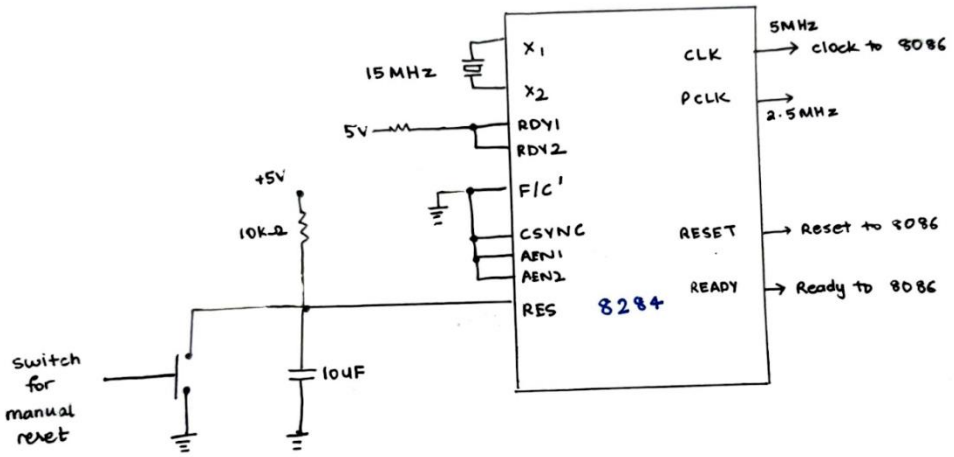
| IC | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 7400 | 1A | 1B | 1Y | 2A | 2B | 2Y | GND | 3Y | 3A | 3B | 4Y | 4A | 4B | VCC |
| 7408 | 1A | 1B | 1Y | 2A | 2B | 2Y | GND | 3Y | 3A | 3B | 4Y | 4A | 4B | VCC |
| 7432 | 1A | 1B | 1Y | 2A | 2B | 2Y | GND | 3Y | 3A | 3B | 4Y | 4A | 4B | VCC |
| 7486 | 1A | 1B | 1Y | 2A | 2B | 2Y | GND | 3Y | 3A | 3B | 4Y | 4A | 4B | VCC |
| 747266 | 1A | 1B | 1Y | 2Y | 2A | 2B | GND | 3A | 3B | 3Y | 4Y | 4A | 4B | VCC |

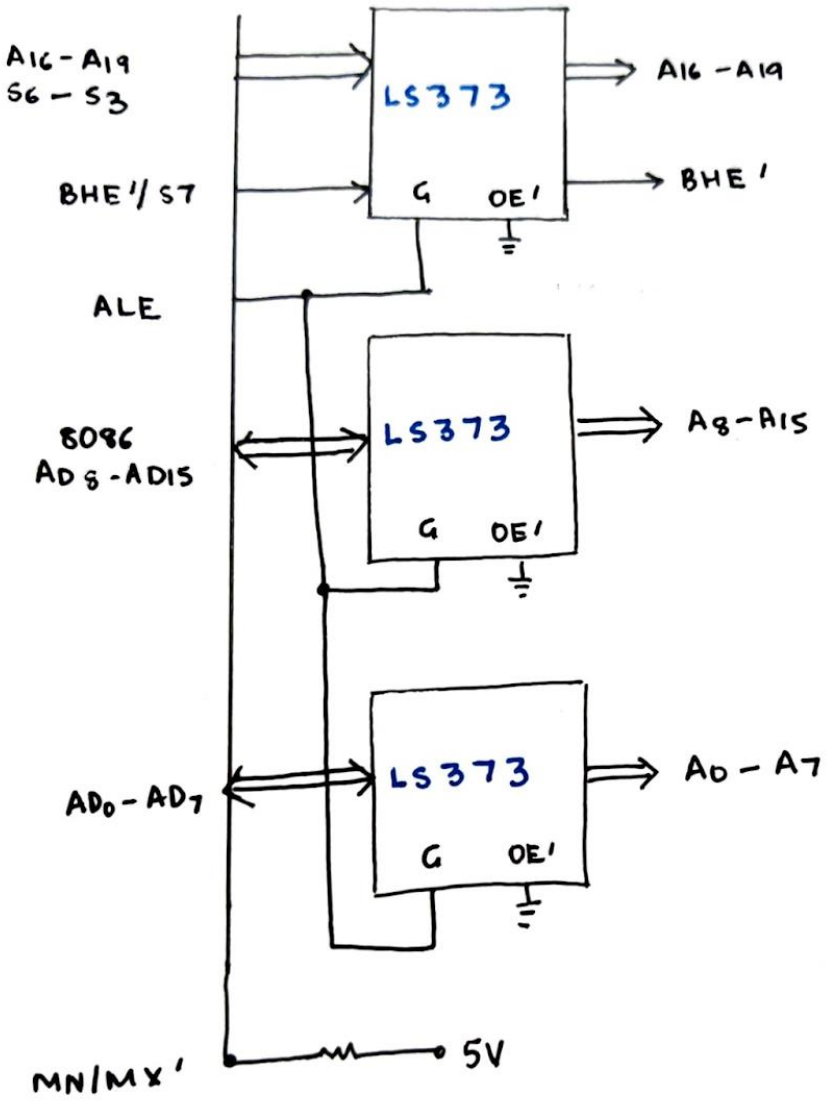
****nA,nB ---> Input pins, nY -----> Output pin**

- From the above table, we can see that pins P1, P2, P9, P5, P12, P13 are always input to the logic gates, and hence these pins are connected to the Port A of the 8255A-2.
- Pins P3 and P11 are always output from the gates, hence they are connected to Port B of 8255A-2.
- The pins P4 and P10 are input pins for the IC chips 7400, 7408, 7432, 7486 and they are output pins for the IC 747266. Hence, these pins are connected to the Upper Port C of the 8255A-2 (As individual bits of port C can be reconfigured via the BSR Mode).
- The pins P6 and P8 are output pins for the IC chips 7400, 7408, 7432, 7486 and they are input pins for the IC 747266. Hence, these pins are connected to the Lower Port C of the 8255A-2 (As individual bits of port C can be reconfigured via the BSR Mode).

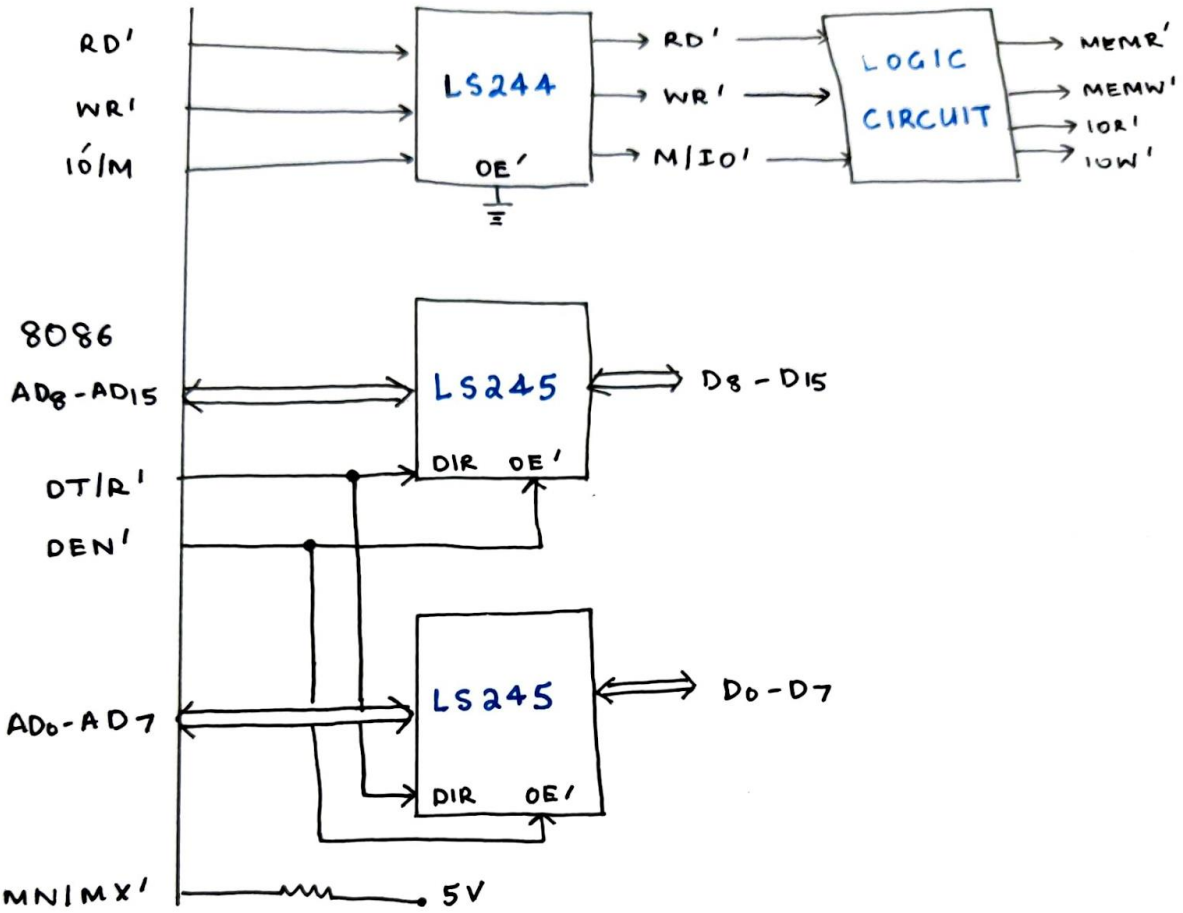
# Design


Reset from 8284
Ready from 8284
VCC
RESET
READY
INTA'
INTR
HOLD/RQ'/GT0'
HLDA/RQ'/GT1'
TEST'
NMI
MN/MX'
CLK
8086
clock from 8284
AD0 - AD15
A16/S3 - A19/S6
ALE/QS0
BHE'/S7
DT/R'/S1'
DEN'/S0'
RD'
WR'/LOCK'
M/IO'/S2'
AD0 - AD15
AD16 - AD19
ALE
BHE'
enable to system bus
RD'
WR'
MIO'

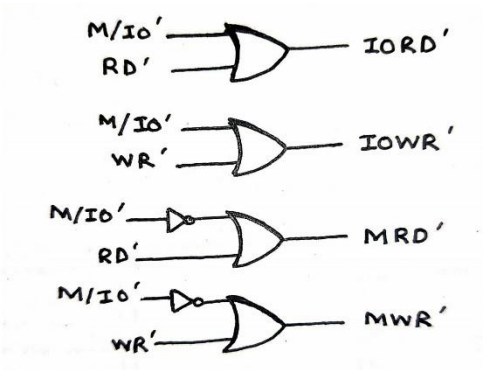

15MHz
X1
X2
5V
RDY1
RDY2
+5V
10KΩ
F/C'
CSYNC
AEN1
AEN2
RES
8284
CLK
PCLK
RESET
READY
5MHz
clock to 8086
2.5MHz
Reset to 8086
Ready to 8086
switch for manual reset
10uF




System Bus of 8086 (Address)



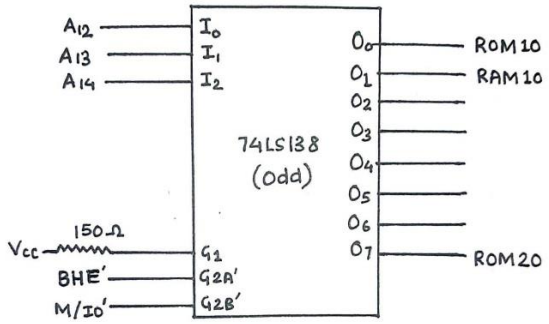

System Bus of 8086 (Data + Control)

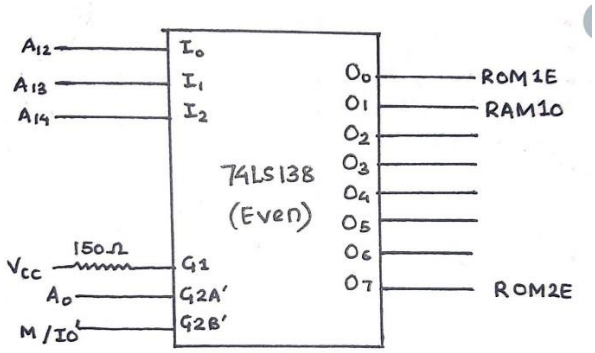


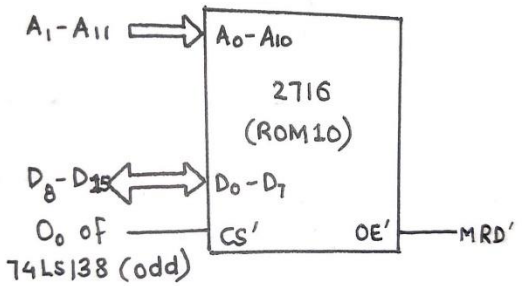

| M/IO' | RD' | WR' | Bus Cycle |
|---|---|---|---|
| 1 | 0 | 1 | MRD' |
| 1 | 1 | 0 | MWR' |
| 0 | 0 | 1 | IORD' |
| 0 | 1 | 0 | IOWR' |

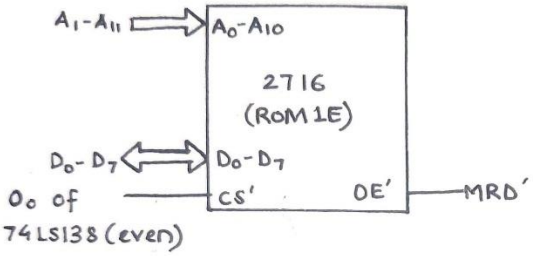
Here, MRD' and MWR' correspond to the memory read (MEMR') and memory write (MEMW') bus cycles, IORD' and IOWR' correspond to I/O read (IOR') and I/O write (IOW') bus cycles respectively

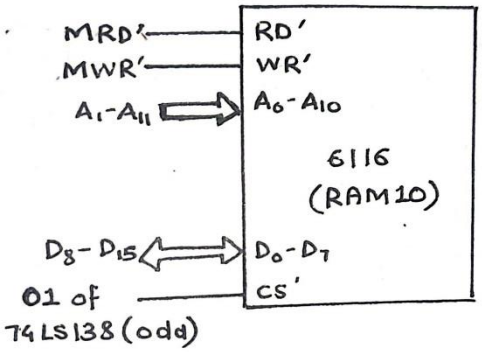
Memory Interfacing Diagrams

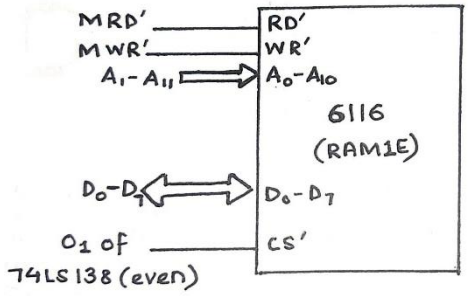


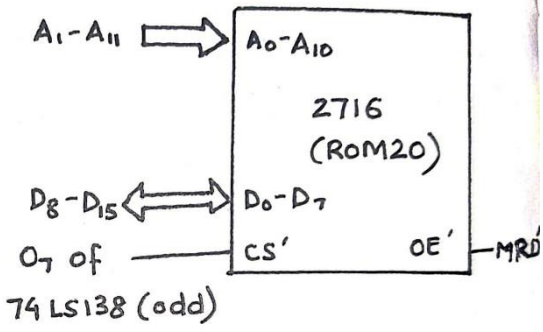



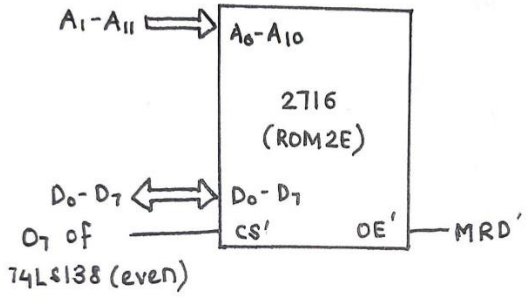



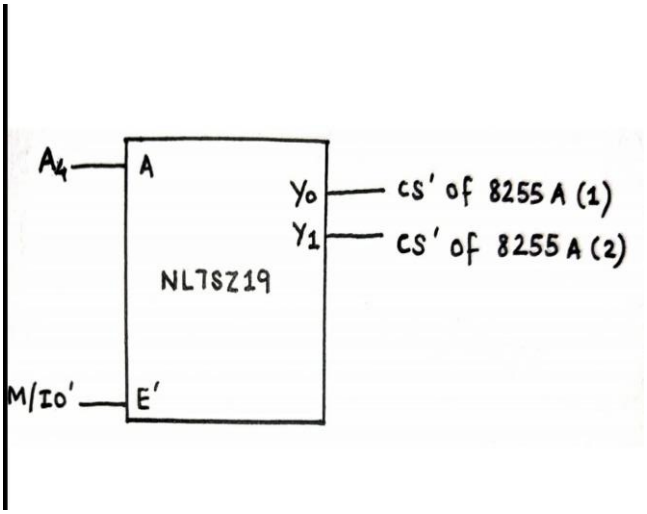



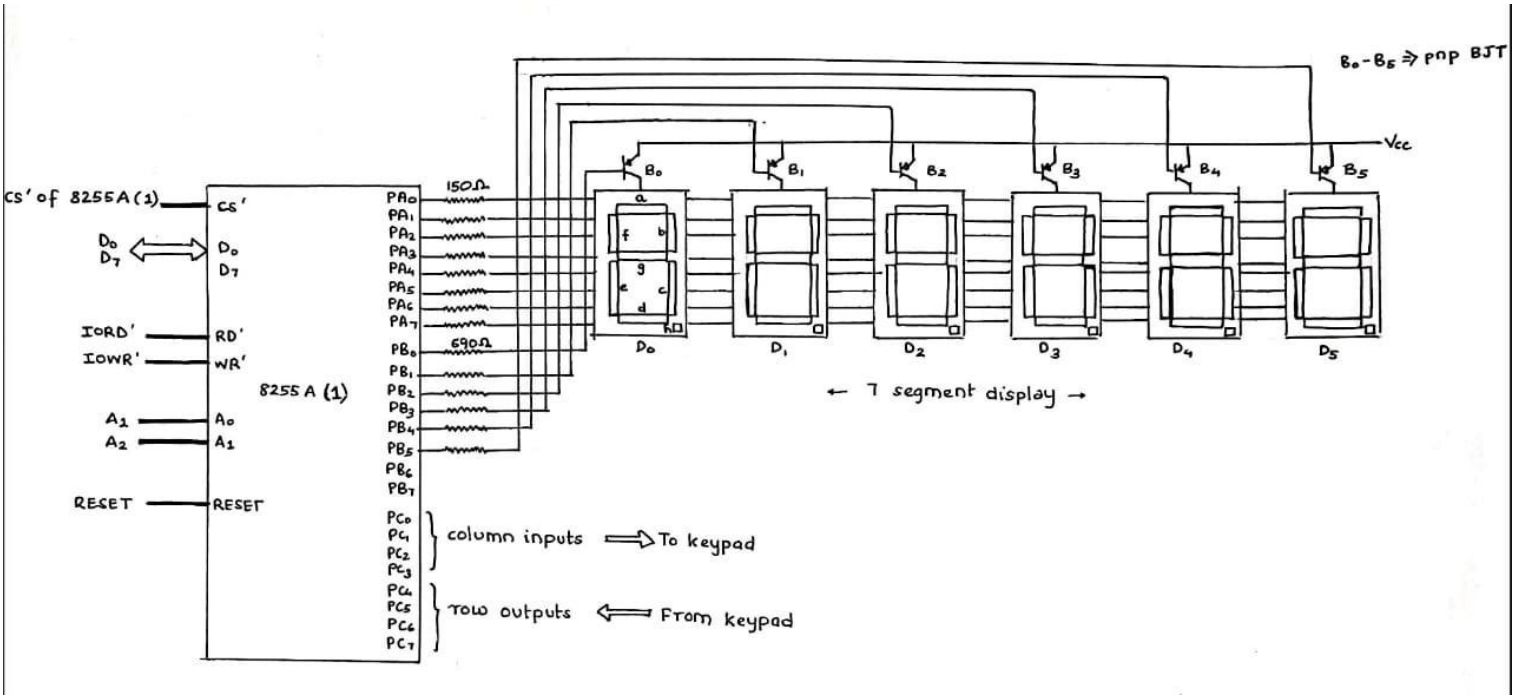



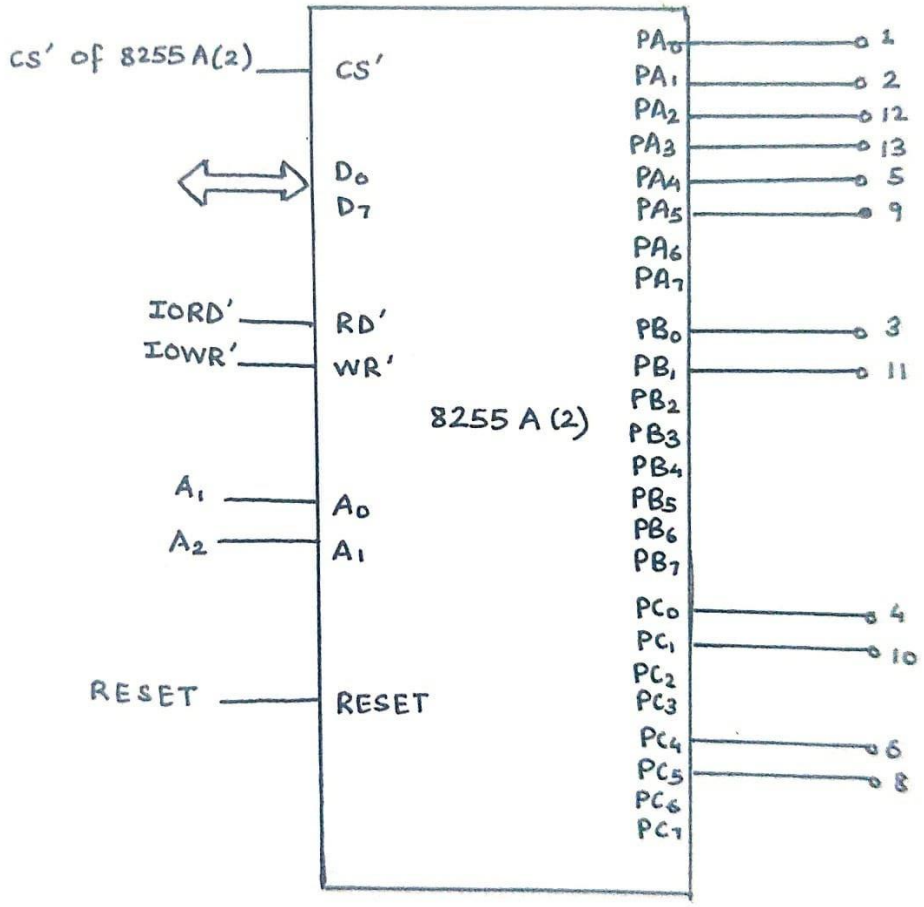



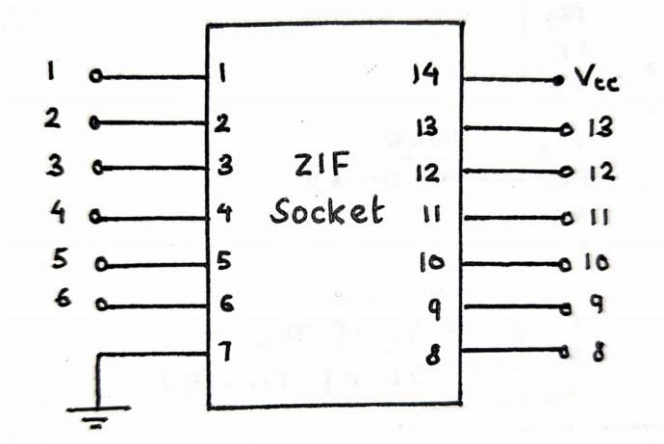



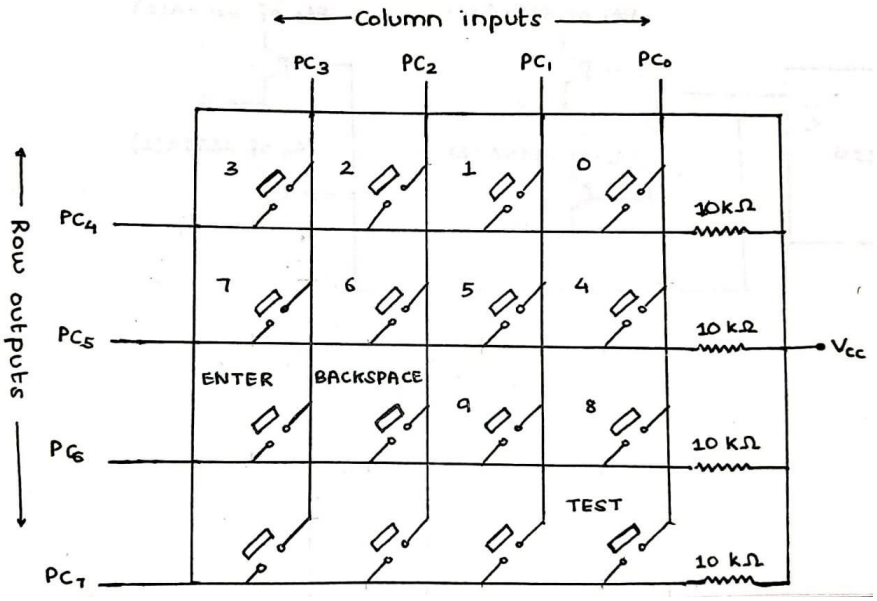
I/O Interfacing Diagrams




CS' of 8255 A(2)
CS'
D0
D7
IORD'
RD'
IOWR'
WR'
8255 A (2)
A1
A0
A2
A1
RESET
RESET
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
1
2
12
13
5
9
3
11
4
10
6
8

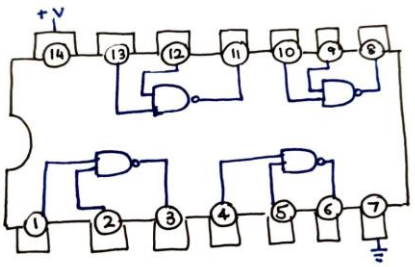

1
2
3
4
5
6
1
2
3
4
5
6
7
ZIF
Socket
14
13
12
11
10
9
8
Vcc
13
12
11
10
9
8




IC's Used

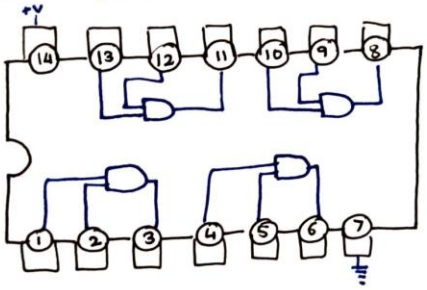
IC-7400

2 input NAND Gate IC



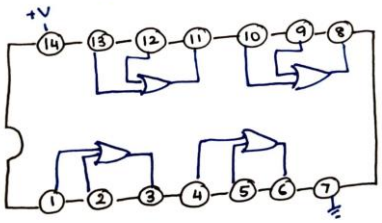

IC-7408

2 input AND gates



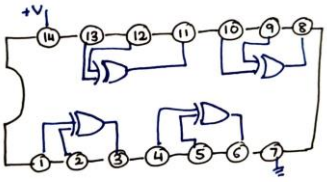

IC 7432

2 input OR gates

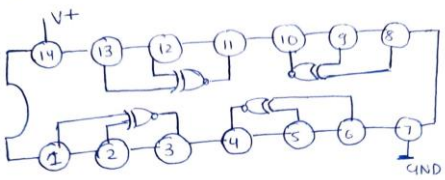



IC 7486

2 input XOR gates

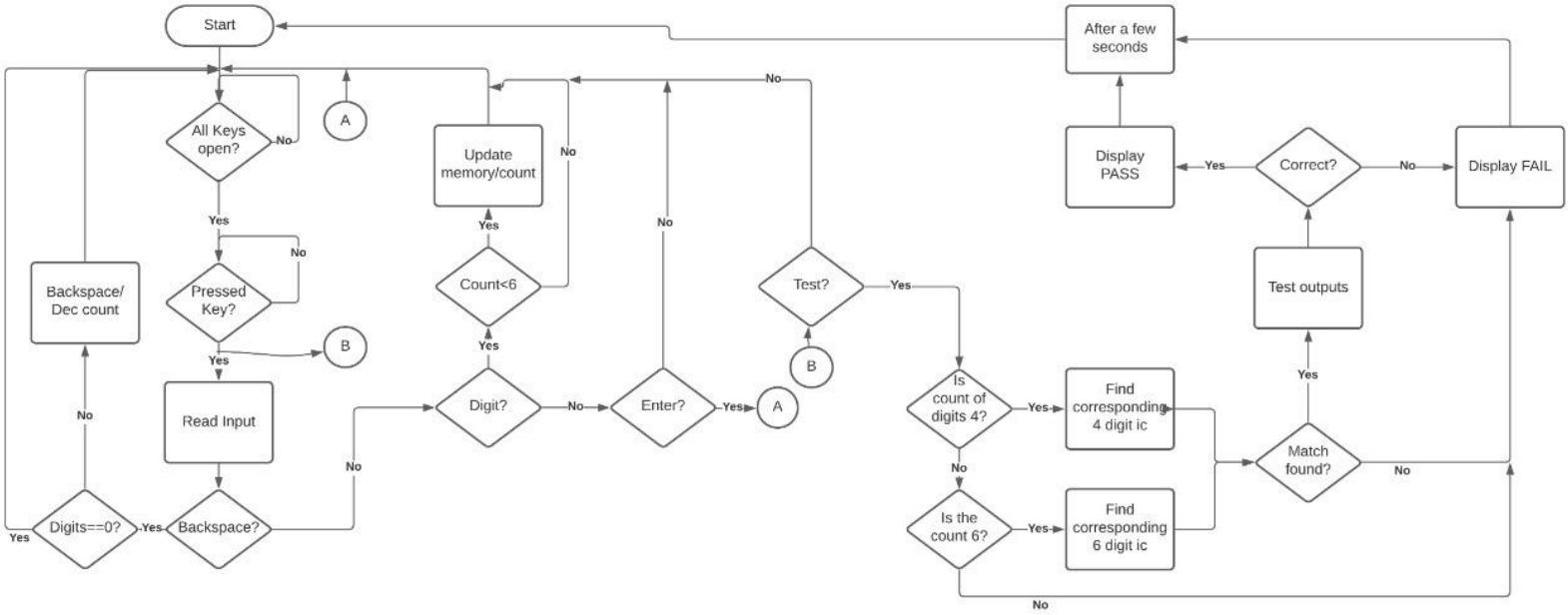



2 INPUT X NOR gates



# Flow Chart

## Main Program



## Variations in Proteus Implementation with Justification

1. Since Proteus does not have working modules of 2K ROM(2716), 4K ROM(2732) modules are used in it. Therefore, it has 16K ROM in total compared to 8 K used in the paper design, with A15, A14 and A13 used accordingly for selection of chip compared to A14, A13 and A12 used in the paper design.
2. Using a gate-based circuit for memory – does the same as LS 138 here.
3. There is no ZIF socket in Proteus, therefore the pins of the IC are directly connected to the corresponding pins in the 8255.

## Firmware

Implemented using emu8086 attached.

## List of Attachments

1. Complete Hardware Real World Design – ictesterdesign.pdf
2. Manuals
    a. IC 7400
    b. IC 7408
    c. IC 7432
    d. IC 7486
    e. IC 747266
    f. 8255A
    g. 8284A
    h. 8086
    i. 2716EPROM
    j. 6716RAM
    k. 2N2905
    l. 7 segment display
    m. LS138
    n. LS244
    o. LS245
    p. LS373
    q. NL7SZ19
    r. Single 2 input positive OR gate
    s. Single Inverter gate
    t. ZIF Socket
3. Proteus File – ictesterdes.dsn
4. EMU8086 ASM File – finalictester_asm.asm
5. Binary File after assembly – finalictester.bin