IC TESTER

Submitted By:

Group 13

Sarthak Gupta (2019AAPS0290G)
Rahul Kumar Singh (2019AAPS0299G)
Chirag Agarwal (2019A3PS0272G)
Taresh Batra (2019A3PS0388G)
Rajath V (2019A7PS0122G)
N Pareekshith Achar (2019A8PS1101G)

Project submitted in partial fulfilment of the requirements of the course

Microprocessors and Interfacing

Contents

1. User Requirements & Technical Specifications 2
2. Assumptions 3
3. Components Used with Justification Wherever Required 4
4. Address Map 6
   - 4.1. Memory Map 6
   - 4.2. I/O Map 8
5. Design 11
6. Flowchart 12
   - 6.1. Input and Display 12
   - 6.2. IC Testing 13
7. Variations in Proteus Implementation with Justification 14
8. Firmware 15
9. List of Attachments 16

User Requirements & Technical Specifications

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 Technical Specifications are as follows:

- 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 and closes it.
- 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.
- The result PASS/FAIL must be displayed on the 7-segment display.
- After testing is complete, PASS/FAIL will be displayed for approximately 5s and then the display will be reset. During this time no input will be taken. After the reset, the user can give another input.

Assumptions

1. The first input cannot be Backspace, Enter or Test key. It Must be a digit.
2. Test key will only be pressed after the Enter key has been pressed.
3. The User cannot give inputs while PASS/FAIL is being displayed.
4. The ALP has been made assuming there is no debounce when a key is pressed in Proteus.

Components Used with Justification Wherever Required

1. **8086** Microprocessor - 1 No
2. **8255-A** (Programmable Peripheral Interface with 24 I/O lines) - 2 Nos; to interface keyboard, display, and the IC.
3. **74HC138** (3:8 Decoder) - 3 Nos
4. **74HC245** (Octal BUS transceivers with tri-state output) - 2 Nos
5. **74HC373** (Octal D-Type transparent latches with tri-state outputs) - 3 Nos
6. **7SEG-MPX1-CA** (7-Segment Anode Display) - 6 Nos
7. **6116** (16K <2Kx8> Static RAM)- 2 Nos; smallest size available, required for temporary storage of data.
8. **2716** (16K <2Kx8> EPROM)- 4 Nos; ROM is required at reset address i.e., FF000H and 00000H where there is the IVT.
9. **BUTTON** (SPST Push Button)- 16 Nos
10. **SW- SPDT-MOM** (Interactive SPDT Switch with Momentary Action)
11. **7400** (Quadruple 2-input positive NAND gates)
12. **7408** (Quadruple 2-input positive AND gates)
13. **7432** (Quadruple 2-input positive OR gates)
14. **7486** (Quadruple 2-input exclusive OR gates)
15. **747266** (Quadruple 2-input exclusive NOR gates)

16. Simple 2-Input OR gate - 4 Nos
17. Simple digital Inverter - 2 Nos
18. **8284** (Clock generator chip implemented in proteus using the internal clock of 8086)
19. **ZIF Socket** (Designed using 'Default' Terminals in Proteus)

Address Map

Memory Map:

The following memory chips have been used:

- 2716 - 16K (2K x 8) = 2KB EPROM each - 2 banks starting at 00000H
- 6116 - 16K (2K x 8) = 2KB Static RAM each - 2 banks starting at 01000H
- 2716 - 16K (2K x 8) = 2KB EPROM each - 2 banks starting at FF000H

Allocation:

- ROM1(Even) - 00000H, 00002H, 00004H ...., 0FFEH

  ROM1(Odd) - 00001H, 00003H, 00005H, ..., 0FFFH

- RAM1(Even) - 01000H, 02002H, 02004H, ..., 01FFEH

  RAM1(Odd) - 01001H, 02003H, 02005H, ..., 01FFFH

- ROM2(Even) - FF000H, FE002H, FE004H, ..., FFFFEH

  ROM2(Odd) - FF001H, FE003H, FE005H, ..., FFFFFH

ROM1 - 00000H -> 00FFFH

A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	....	A0
0	0	0	0	0	0	0	0	0	0	0	...	0
0	0	0	0	0	0	0	0	1	1	1	...	1

RAM1 - 01000H -> 01FFFH

A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	....	A0
0	0	0	0	0	0	0	1	0	0	0	...	0
0	0	0	0	0	0	0	1	1	1	1	...	1

ROM2 - FF000H -> FFFFFH

A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	....	A0
1	1	1	1	1	1	1	1	0	0	0	...	0
1	1	1	1	1	1	1	1	1	1	1	....	1

I/O Map:

We have used two 8255As for our I/O devices. The addresses allocated are:

1. 8255A-1:
    a. Port A - 00H
    b. Port B - 02H
    c. Port C - 04H
    d. Control Register - 06H

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:
    a. Port A - 10H
    b. Port B - 12H
    c. Port C - 14H
    d. Control Register - 16H

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

NOTE: A3, A4 and A5 have different values for the two different 8255s, but they share the same values for the different ports of the same 8255. Hence, we use them as our select lines in our decoder for the I/O devices.

The Port Allocation is done as follows:

1. 8255A-1:

    a. Port A - 8 pins as outputs from the 8255, which are the inputs to the 7-segment display.

    b. Port B - 6 pins as outputs from the 8255, which are the enables to each 7-segment display.

    c. Port C –

        i. Lower: 4 pins as outputs from 8255; inputs to the keyboard.
        ii. Upper: 4 pins as inputs to 8255; outputs from the keyboard

2. 8255A-2:

    a. Port A - 6 pins as outputs from the 8255, which are inputs to the ZIF socket pins: P1, P2, P12, P13, P5 and P9.

    b. Port B - 2 pins as inputs to the 8255, which are outputs from the ZIF socket pins: P3 and P11.

    c. Port C -
        i. Port C-lower - 2 pins: P4 and P10.
        ii. Port C-upper - 2 pins: P6 and P8.

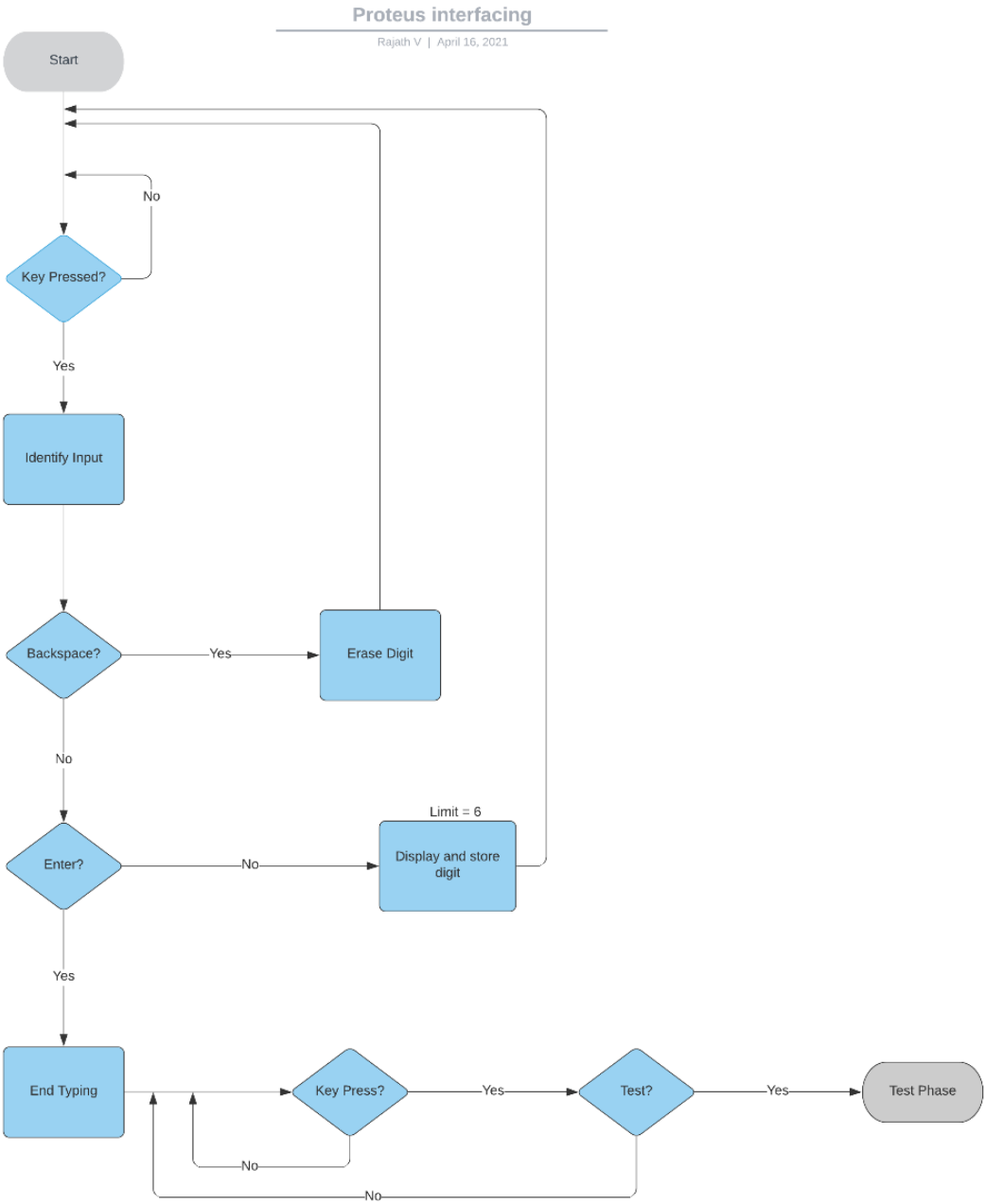
NOTE: The 5 ICs in the database are 7400, 7408, 7432, 7486, and 747266. We can group 7400, 7408, 7432, 7486 into group 1 and 747266 as group 2. As we can see, for group 1, P1, P2, P4, P5, P9, P10, P12, P13 act as input terminals (output from 8255-2), and P3, P6, P8 and P11 act as output terminals (input to 8255-2). But, in group 2 the difference comes in P4, P10 & P6, P8. The former act as output terminals (input to 8255-2) whereas the latter act as input terminals (output from 8255-2). Due to this P1, P2, P12, P13, P5 and P9 have been always connected to Port-A (Output from 8255-2) and P3, and P11 have been always connected to Port-B (Input to 8255-2). Port-C has the property of having its upper and lower ports programmed separately, therefore P4, P10 have been connected to lower half of Port-C and P6, P8 have been connected to upper half of Port-C. These 2 halves of Port-C switch between input/output ports depending on the IC that needs to be tested.

Design

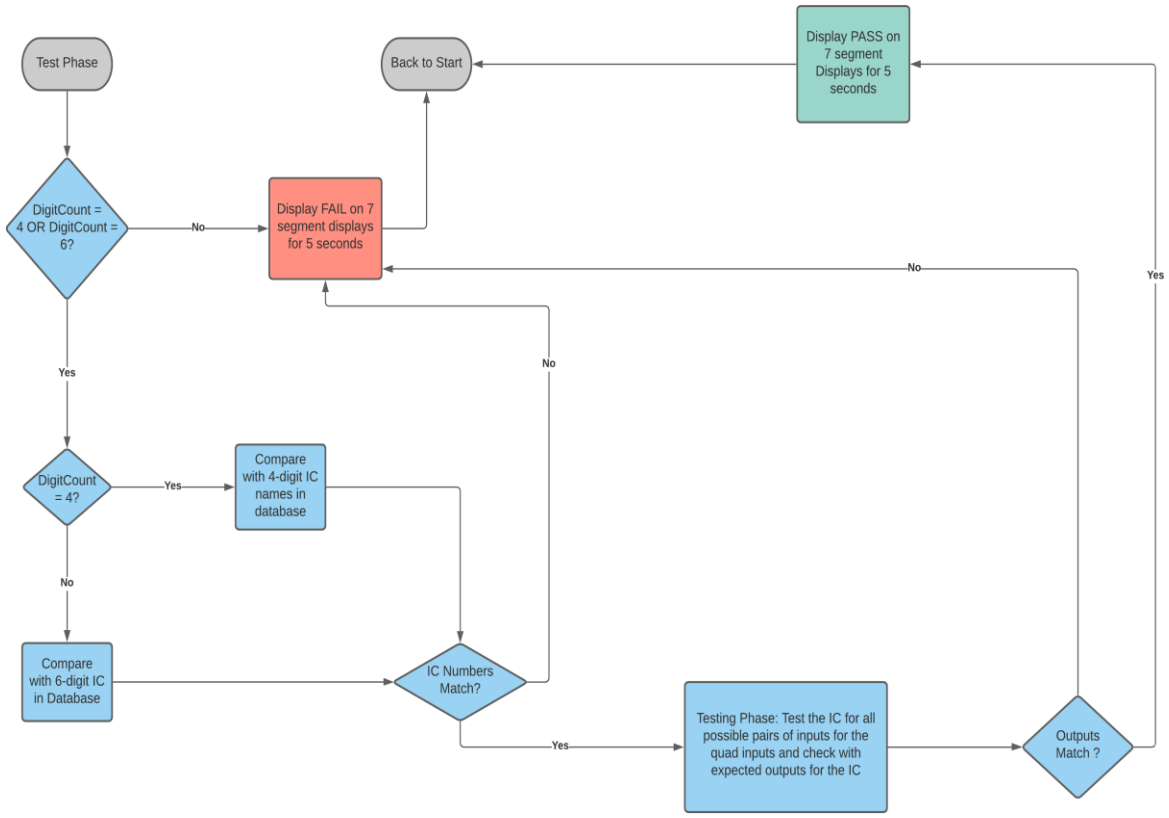
Complete design shown with proper labeling (design attached).

Flowchart

Input and Display:



IC Testing:


Test Phase
Back to Start
Display PASS on 7 segment Displays for 5 seconds
DigitCount = 4 OR DigitCount = 6?
No
Display FAIL on 7 segment displays for 5 seconds
No
Yes
Yes
No
DigitCount = 4?
Yes
Compare with 4-digit IC names in database
No
Compare with 6-digit IC in Database
IC Numbers Match?
Yes
Testing Phase: Test the IC for all possible pairs of inputs for the quad inputs and check with expected outputs for the IC
Outputs Match ?

Variations in Proteus Implementation with Justification

1. 4077 (quad 2 input XNOR chip) has been used in place of 747266 due it being unavailable in Proteus.
2. Since ZIF is a mechanical component, it's not available on Proteus. Thus, it has been designed using the 'Default' Terminals.
3. 2732 is used instead of 2716 as the latter does not work on Proteus.

Firmware

Completed using emu8086 (attached).

List of Attachments

1. Complete hardware real world design: hardware_design.pdf
2. Proteus files (x5, one for each IC):
    a. 7400_NAND.dsn
    b. 7408_AND.dsn
    c. 7432_OR.dsn
    d. 7486_XOR.dsn
    e. 747266_XNOR.dsn
3. EMU8086 ASM file: ictest.asm
4. Binary file after assembly: ictest.bin
5. Manuals-
    a. ZIF 14 Pin Socket: ZIF_14_PIN.pdf
    b. 7400_NAND.pdf
    c. 7408_AND.pdf
    d. 7432_OR.pdf
    e. 7486_XOR.pdf
    f. 747266_XNOR.pdf

Note: Since the IC-Tester needs to test 5 ICs and each .dsn file tests one IC, we have included the .dsn files for all 5 designs as part of our submission for the proteus implementation.