RAM Tester

Submitted by: Group 72

Group Members

Mehul Gupta
Shourya Pathak
Aryan Singh
Dev Goel
Jayant Chaudhary
Abhijit Deo

Date: 16/4/2021

Contents

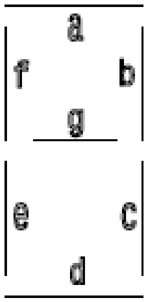
System Description

The RAM tester is so designed , such that the user is provided with one switch for the RAM being tested. Every bit of RAM is first written with 0 and then with 1. After every write operation , Values are read and compared to the values written. If they are equal in both cases, RAM is fully functional, Else, RAM is faulty. Conclusion is displayed on a seven segment display.

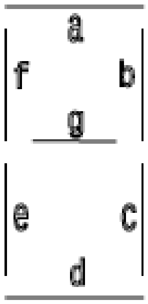
Technical Specification

- The RAM tester is built around a 8086 microprocessor.
- RAM's are connected to it using a 8255 interfaced at 80H because both write and read operation is required.
- Another 8255 interfaced at 50H is used to interact with the input and output devices.
- 4K of ROM have been attached to the microprocessor.
- 8284A clock generator to produce a stable frequency (5 MHz) clock signal which steps the 8086.
- A 7 segment display is used. Its technical specifications are as follows-

It consists of 7 LED's placed as displayed on the next slide:



θ	g	f	e	d	c	b	a



θ	g	f	e	d	c	b	a

The anode is connected to +5V. Hence to see any particular letter, the LEDs that have to glow have to be provided with 0 as input. For example for ‘P’ the code word shall be 9D H. In this way, all the alphabets can be displayed. The display devices have been interfaced through the latches so that display is static

Assumptions & Justifications

1.6264 SRAM model has been used in Proteus implementation due to absence of 616.

2. We have directly generated a DC clock pulse instead of using clock generator chips.

3. The reset switch remains long enough for the 8255 to poll it.

4. 8086 is always RESET before again reusing it to test another RESET.

5.SRAM to be tested is connected before pushing the Test button.

6. At the memory location FFFFOh where the processor returns to after reset the Jmp statement taking it to the start of the code.

Components used with justification wherever required

S.No	Component	Quantity	Description
1.	8086	1	Microprocessor used to interface ROM
2.	74LS373	3	This is an 8 bit address latch used to get address from AD signals (coming from microprocessor) 3 such components have been used because this is an 8 bit latch while the number of AD lines are 20(AD_0-$AD_{19)}$
3.	74HC245	2	This is a bi-directional data buffer. This is used to get data lines from AD lines and amplify them.
4.	7 segment display	4	To display 'PASS' or 'Fail' depending upon the result of the test.
5.	Switches	2	1.TEST SWITCH- To start/Turn off the RAM test 2.RESET SWITCH-To reset the circuit-
6.	8255A	2	It is a programmable peripheral device used to interface system and device side. One 8255A is used to interface RAM and second to interface i/o devices.
7.	LED	1	To generate status of testing
8.	8284	1	To generate 5 MHz clock pulse.
9.	2716	2	Contains the code and the RESET address at FFFF0h
10.	LS138	1	To switch Memory operations between ROM and SRAM
11	6264	2	Used for stack and temporary storage.

Address Map

- ## Memory Map

From 0000h
To 03FFFh

A 19	A 18	A 17	A 16	A 15	A 14	A 13	A 12	A 11	A 10	A 9	A 8	A 7	A 6	A 5	A 4	A 3	A 2	A 1	A 0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1

EEPROM: From
FF000h to FFFFFh

A 19	A 18	A 17	A 16	A 15	A 14	A 13	A 12	A 11	A 10	A 9	A 8	A 7	A 6	A 5	A 4	A 3	A 2	A 1	A 0
1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

• I/O Map Using 8255

1.8255: Interfacing the i/o devices

Base Address:50h

The addresses of the ports are as follows:

NAME OF THE PORTS OF 8255	ADDRESS
Port A	50h
Port B	52h
Port C	54h
Control Register	56h

2. 8255 Interfacing the RAM to be checked

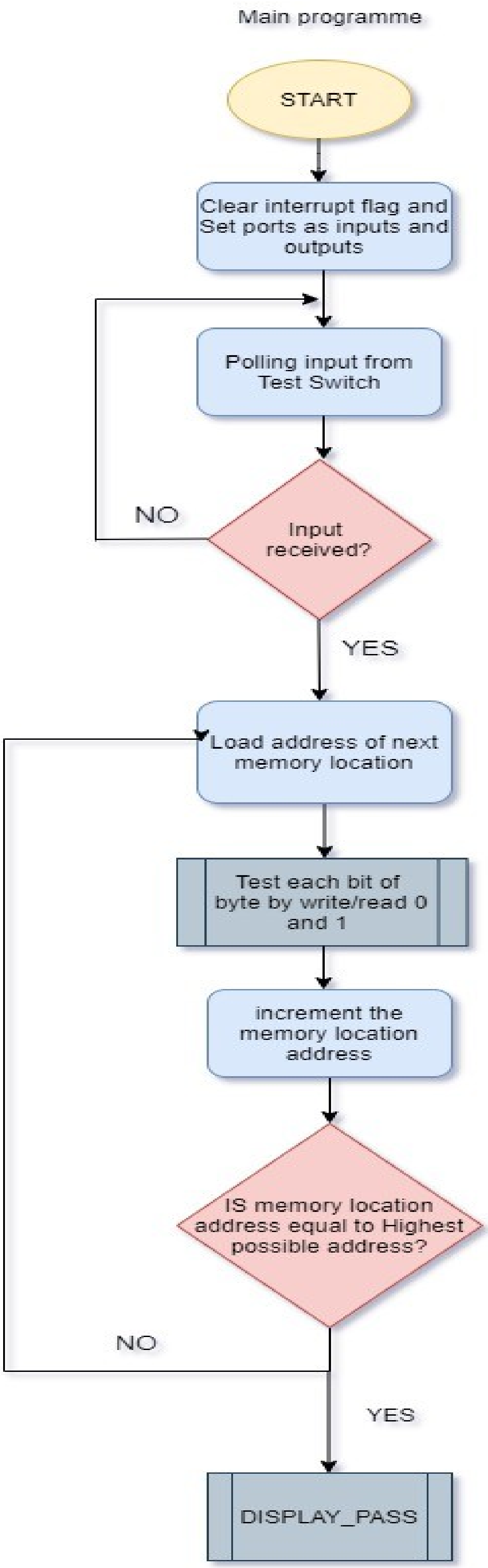
Base Address:80h

The addresses of the ports are as follows:

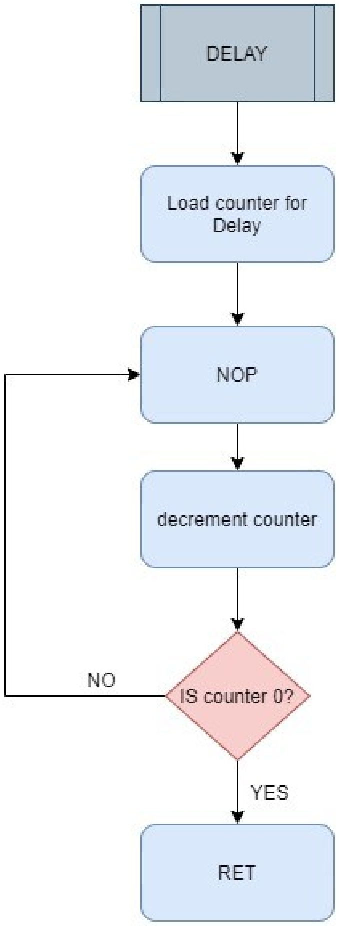
NAME OF THE PORTS OF 8255	ADDRESS
Port A	80h
Port B	82h
Port C	84h
Control Register	86h

FLOW CHART

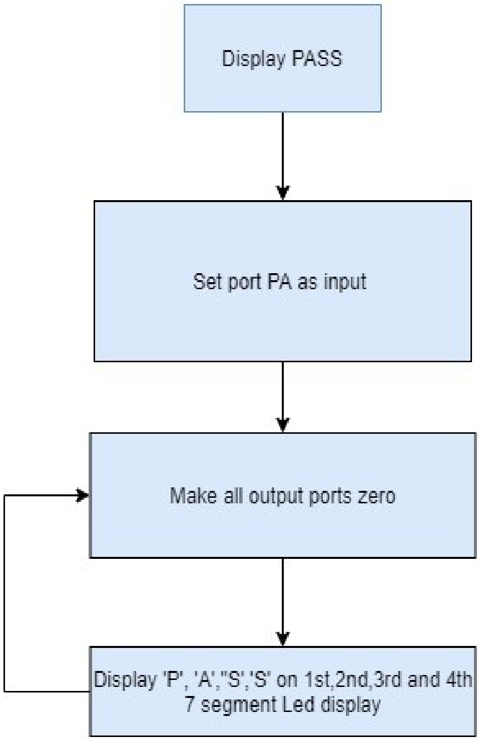
MAIN PROGRAM


Main programme
START
Clear interrupt flag and Set ports as inputs and outputs
Polling input from Test Switch
Input received?
NO
YES
Load address of next memory location
Test each bit of byte by write/read 0 and 1
increment the memory location address
IS memory location address equal to Highest possible address?
NO
YES
DISPLAY_PASS

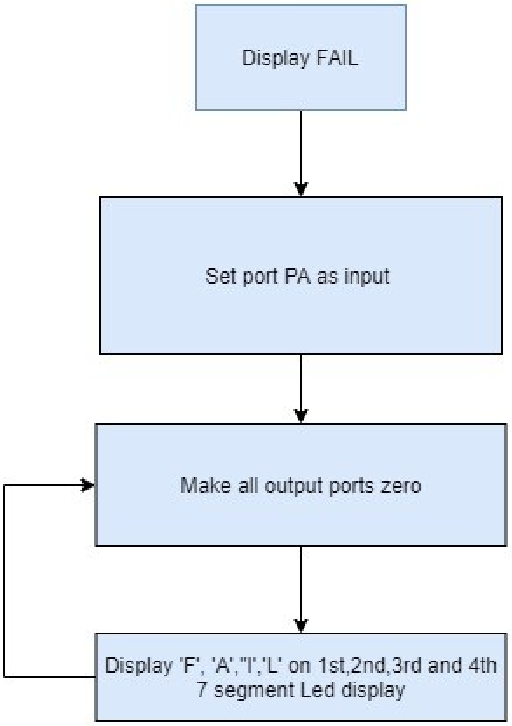
PROCEDURE DELAY



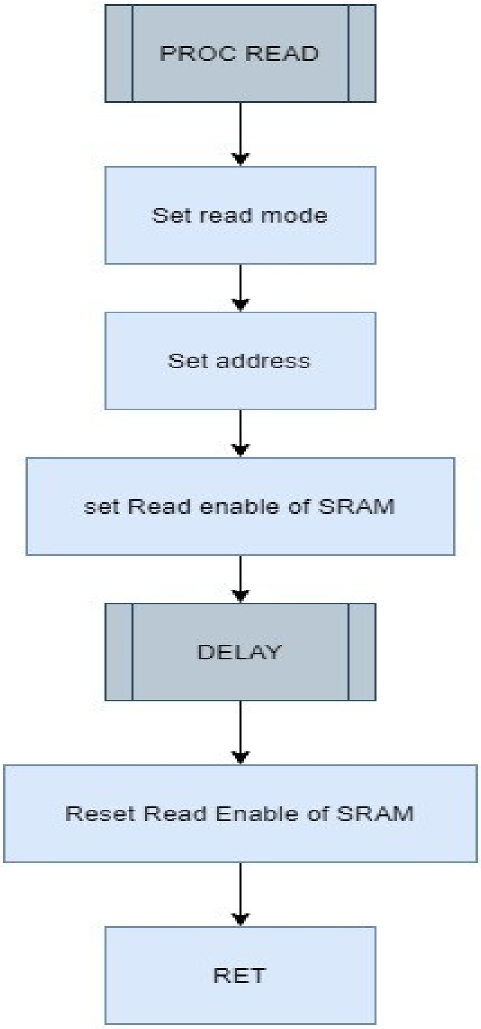
PROCEDURE DISPLAY_PASS


Display PASS
Set port PA as input
Make all output ports zero
Display 'P', 'A',"S','S' on 1st,2nd,3rd and 4th 7 segment Led display

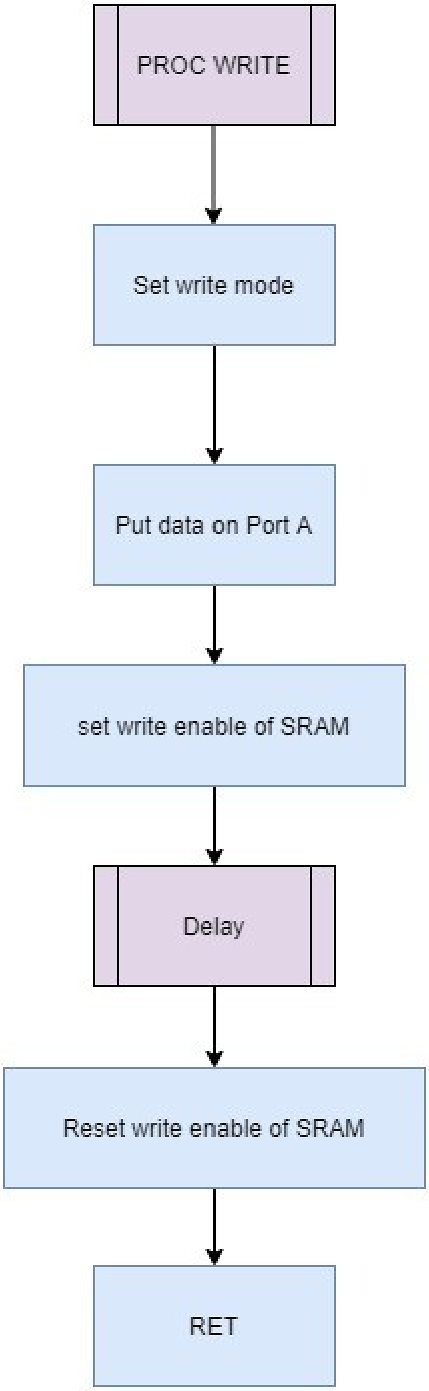
PR CEDURE DISPLAY_FAIL


Display FAIL
Set port PA as input
Make all output ports zero
Display 'F', 'A','I','L' on 1st,2nd,3rd and 4th 7 segment Led display

PROCEDURE READ



PROCEDURE WRITE



PROC WRITE

Set write mode

Put data on Port A

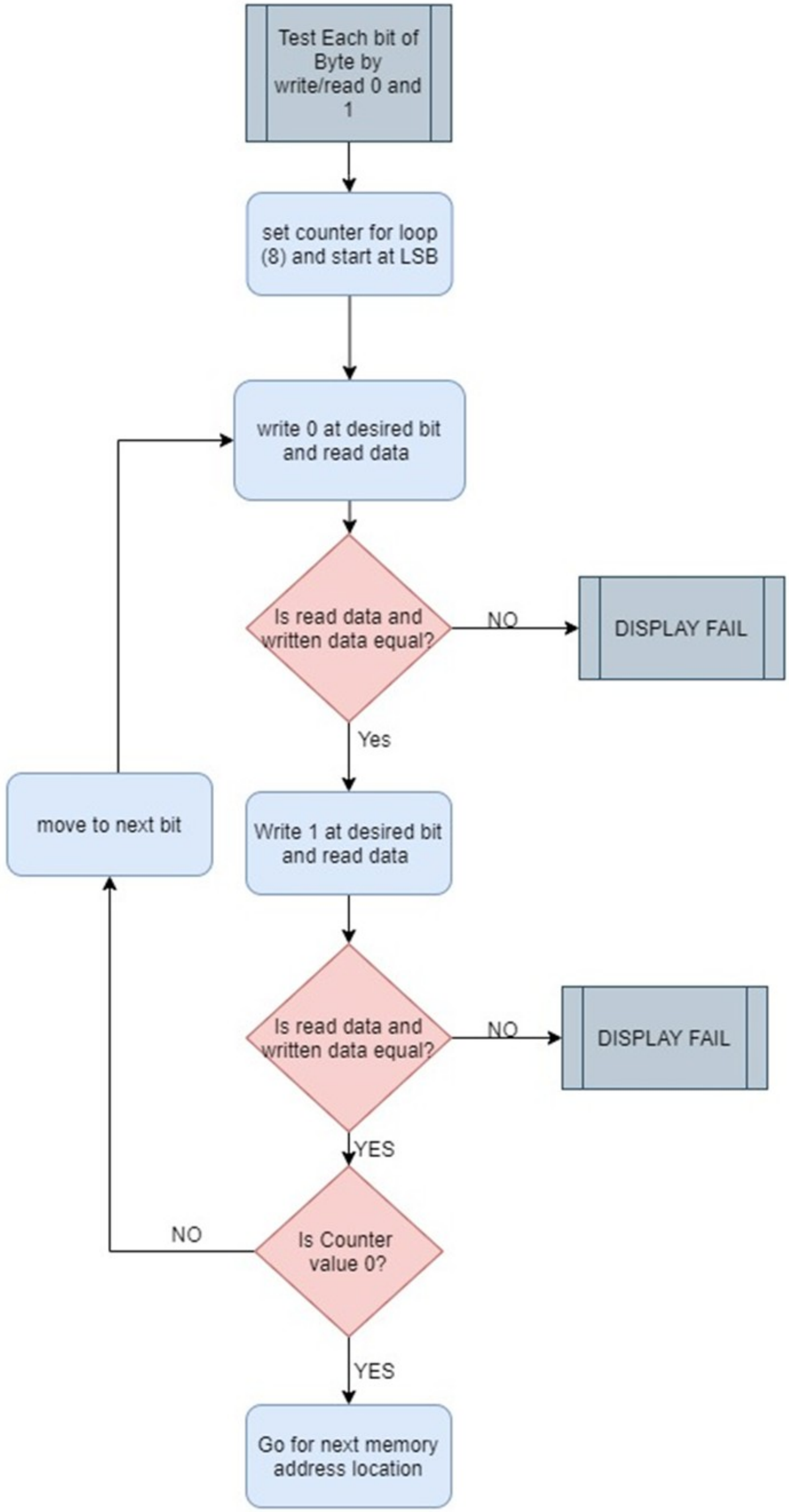
set write enable of SRAM

Delay

Reset write enable of SRAM

RET

TESTING EVERY BYTE BIT BY BIT


Test Each bit of Byte by write/read 0 and 1
set counter for loop (8) and start at LSB
write 0 at desired bit and read data
Is read data and written data equal?
NO
DISPLAY FAIL
Yes
Write 1 at desired bit and read data
move to next bit
Is read data and written data equal?
NO
DISPLAY FAIL
YES
Is Counter value 0?
NO
YES
Go for next memory address location

Variations in Proteus implementation with Justification

1 . We have directly generated a DC clock pulse instead of using 8284.

2. Due to absence of LS138, gate circuitry has been employed.

3. 6164 is not present in proteus, hence 6264 was used.

4. ROM has not been used in proteus as proteus allows to RESET address and run the code without it.

List of Attachments

- Complete Hardware Design showing various components used
- The Proteus file implementing the design for simulation
- The EMU 8086 ASM file
- The BIN file generated after assembly of the code
- Relevant datasheets for components used