PROJECT REPORT
“MICROPROCESSOR BASED RAM TESTER”
Hardware Design And ALP

CS F214
MICROPROCESSOR PROGRAMMING AND INTERFACING

Submitted by:

ADITI MANDLOI	**2017A7PS0160P**
KUSHAGRA RAINA	**2017A7PS0161P**
SHREYAS SRIKRISHNA	**2017A7PS0162P**
JITHIN KALLUKALAM SOJAN	**2017A7PS0163P**

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE
PILANI RAJASTHAN - 333031

P11 :
Design a microprocessor based RAM tester. The tester should be able to test 6116 and 62256 RAM chips. The tester test each bit of the RAM individually. For a byte of RAM, the first bit (D0) is written as zero and read back, now a one is written into the bit and again it is read back. If the two read operations result in bit D0 to contain a zero and one respectively then the bit is inferred as good. Any other result indicates a faulty bit. The test is repeated for all bits of a byte and for all bytes of the RAM. The results of the test along with the RAM IC number are to be displayed on LCD as "PASS" or "FAIL"

Description :
The system is designed such that the user has a choice to select the desired RAM (6116 or 62256). This choice is in the form of a switch. Depending on the choice, the corresponding address ranges of the memory chips are selected. Within the address ranges, every byte of the RAM is written and read 0's and 1's. If the written and read values are not same it is implied that the RAM is faulty and a 'FAIL' message is displayed. If there is no such mismatch in the entire address range, a 'PASS' message is displayed.

Assumptions :
1. It is assumed that at any given instant the two switches decides the SRAM to be tested.
2. At the memory location FFFF0H where the processor returns on reset is provided with a JMP statement taking it to the start of the code.
3. It is assumed that there is no power failure.

Hardware Required:
1. Intel 8086: Microprocessor
2. 8254A: Clock generator 5MHz
3. 6116 SRAM 2K CHIPS (2)
4. 2732 EPROM 4K CHIPS (2)
5. 74154: Decoder (1)
6. 74138: Decoder (1)
7. 74245: 8-bit Bidirectional data buffer (4)
8. 74373: 8-bit address latches (3)
9. 8255A: Programmable Peripheral Interface (1)
10. OR Gates (6)
11. 4078: 8-input NOR Gate (2)
12. NOT Gates (3)
13. Switches (2)
14. LM020L: Liquid Crystal Display (LCD)

15. 6116 SRAM (to be attached when tested)
16. 62256 SRAM (to be attached when tested)

MEMORY MAP :

DEVICE	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
62256																				
32kB SRAM(only even banks)																				
From: 10000h	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
To: 1FFFEh	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6116																				
2kB SRAM(only even banks)																				
From: 20000h	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
To: 20FFEh	0	0	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
6116																				
2kB SRAM(x2)																				
From: 30000h	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
To: 30FFFh	0	0	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
2732																				

4kB EPROM(x2)																				
From: FE000h	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
To: FFFFFh	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

8255 :Interfacing the i/o devices

Base Address: 10H

The addresses of the ports are as follows

PORTS	ADDRESS
Port A	10H
Port B	12H
Port C	14H
Control register	16H

Data lines: D0-D7 data lines of the microprocessor(as it is connected in even bank)

Port Specification:

Group A: Mode 0
Group B: Mode 0

Port A: Output
Port B: Output
Port C: Input

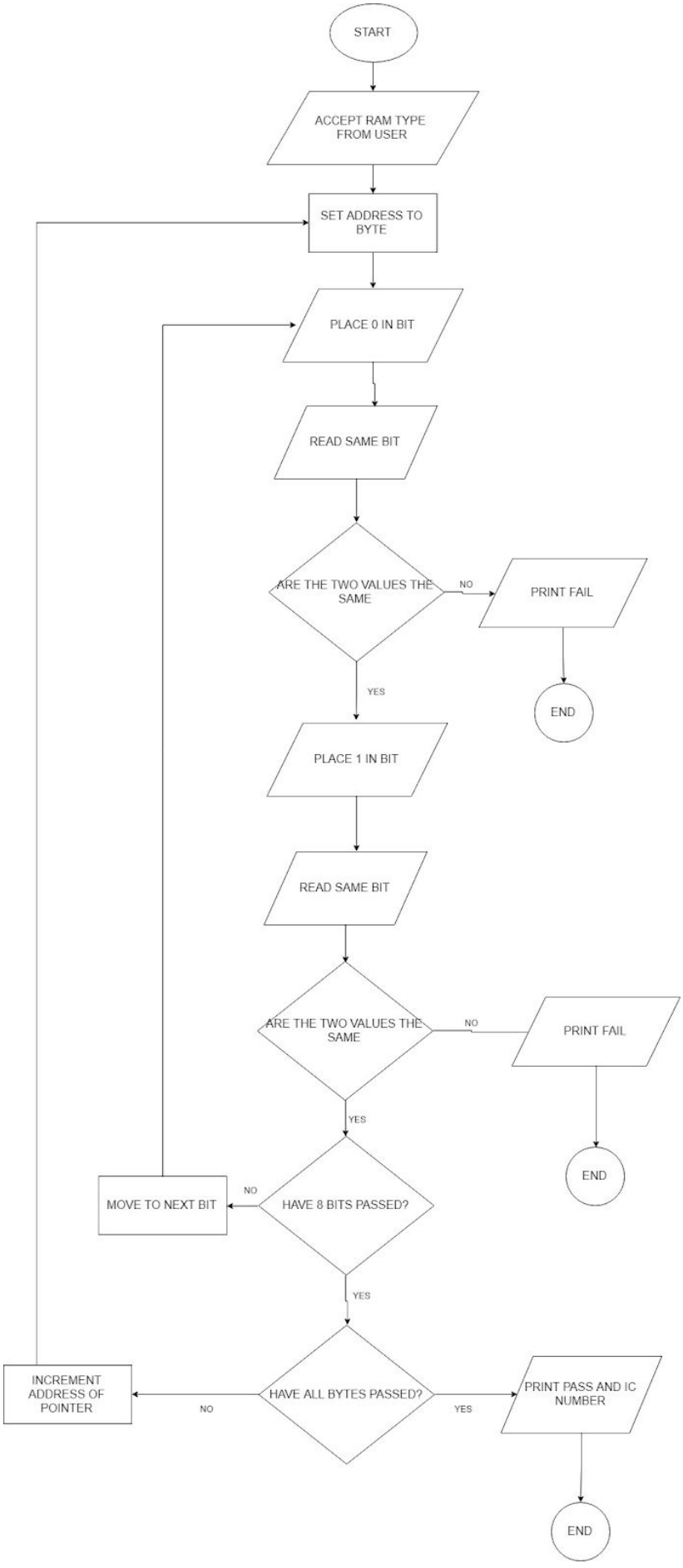
Hence, the control word is

1	0	0	0	1	0	0	1

i.e. 89H which is written to the control register.

Port C is used for the input i.e. getting the information from the user whether 6116 or 62256 SRAM has to be checked on the basis of the switch position(PCE).
Port A is used to give the data of the character to be printed on the LCD screen as well as the control words to the LCD.
Port B is used for controlling the LCD. It sets the pins of the LCD to start glowing and disables the signals as well.

FLOWCHART:


START
ACCEPT RAM TYPE FROM USER
SET ADDRESS TO BYTE
PLACE 0 IN BIT
READ SAME BIT
ARE THE TWO VALUES THE SAME
NO
PRINT FAIL
END
YES
PLACE 1 IN BIT
READ SAME BIT
ARE THE TWO VALUES THE SAME
NO
PRINT FAIL
END
YES
HAVE 8 BITS PASSED?
NO
MOVE TO NEXT BIT
YES
HAVE ALL BYTES PASSED?
NO
INCREMENT ADDRESS OF POINTER
YES
PRINT PASS AND IC NUMBER
END

ASSEMBLY LANGUAGE PROGRAM (ALP) :

```
.model tiny

.data

porta equ 10h ;PORT A of 8255
portb equ 12h ;PORT B of 8255
portc equ 14h ;PORT C of 8255
cw equ 16h    ;Address of control word

.code
.startup

CALL LCD_INIT
CALL START


LCD_INIT PROC NEAR
        push ax
        push cx
        MOV AL, 38H          ;initialise line of LCD
        CALL WRITE_COMND
        CALL DELAY
        CALL DELAY
        CALL DELAY
        MOV AL, 0EH
        CALL WRITE_COMND
        CALL DELAY
        MOV AL,01            ;Clearing the LCD
        CALL WRITE_COMND
        MOV AL,06            ;Pushing the cursor right
        CALL WRITE_COMND
        CALL DELAY
        pop cx
        pop ax
        RET
LCD_INIT ENDP


CLEAR PROC
```

```
        push ax
        push cx
        MOV AL,01
        CALL WRITE_COMND
        CALL DELAY
        CALL DELAY
        pop cx
        pop ax
        RET
CLEAR ENDP

WRITE_COMND PROC
        push dx
        push ax
        push cx
        MOV DX,PORTA
        OUT DX, AL          ;character sent to PORT A
        MOV DX, PORTB
        MOV AL,00000100B  ;RS=0,R/W=0,E=1 for H-To-L pulse
        OUT DX, AL          ;signal enabled on next clock pulse
        NOP
        NOP
        MOV AL,00000000B  ;RS=0,R/W=0,E=0 for H-To-L pulse
        OUT DX, AL          ;signal disbaled on next clock pulse
        pop cx
        pop ax
        pop dx
        RET
WRITE_COMND ENDP

WRITE_IC PROC NEAR
        PUSH AX
        PUSH DI
        MOV AX,DI
        CMP AX,0FFEH
        JNZ RAMT2

        RAMT1:                  ;Printing the IC number 6116
        MOV AX,0000H
        MOV AL,'6'
        CALL WRITE_DATA
        CALL DELAY
        CALL DELAY
```

```
        MOV AL,'1'
        CALL WRITE_DATA
        CALL DELAY
        CALL DELAY
        MOV AL,'1'
        CALL WRITE_DATA
        CALL DELAY
        CALL DELAY
        MOV AL,'6'
        CALL WRITE_DATA
        CALL DELAY
        CALL DELAY
        JMP EOF

        RAMT2:                  ;Printing the IC number 62256
        MOV AX,0000H
        MOV AL,'6'
        CALL WRITE_DATA
        CALL DELAY
        CALL DELAY
        MOV AL,'2'
        CALL WRITE_DATA
        CALL DELAY
        CALL DELAY
        MOV AL,'2'
        CALL WRITE_DATA
        CALL DELAY
        CALL DELAY
        MOV AL,'5'
        CALL WRITE_DATA
        CALL DELAY
        CALL DELAY
        MOV AL,'6'
        CALL WRITE_DATA
        CALL DELAY
        CALL DELAY
        EOF:
        POP DI
        POP AX
        RET
WRITE_IC ENDP
```

```
WRITE_PASS PROC NEAR
        push dx
        push cx
        push ax
        CALL CLEAR
        MOV AL,'P'              ;Printing PASS
        CALL WRITE_DATA
        CALL DELAY
        CALL DELAY
        MOV AL,'A'
        CALL WRITE_DATA
        CALL DELAY
        CALL DELAY
        MOV AL, 'S'
        CALL WRITE_DATA
        CALL DELAY
        CALL DELAY
        MOV AL,'S'
        CALL WRITE_DATA
        CALL DELAY
        CALL DELAY
        pop ax
        pop cx
        pop dx
        RET
WRITE_PASS ENDP


WRITE_FAIL PROC NEAR
        push dx
        PUSH CX
        PUSH AX
        CALL CLEAR
        MOV AL,'F'              ;PRINTING FAIL
        CALL WRITE_DATA
        CALL DELAY
        CALL DELAY
        MOV AL,'A'
        CALL WRITE_DATA
        CALL DELAY
        CALL DELAY
        MOV AL,'I'
        CALL WRITE_DATA
```

```
        CALL DELAY
        CALL DELAY
        MOV AL,'L'
        CALL WRITE_DATA
        CALL DELAY
        CALL DELAY
        POP AX
        POP CX
        POP DX
        RET
WRITE_FAIL ENDP


WRITE_DATA PROC
        PUSH DX
        PUSH AX
        MOV DX, PORTA
        OUT DX, AL
        MOV AL, 00000101B
        MOV DX, PORTB
        OUT DX, AL
        MOV AL, 00000001B
        OUT DX, AL
        POP AX
        POP DX
        RET
WRITE_DATA ENDP


DELAY PROC
        PUSH CX
        PUSH AX
        MOV CX,1325;SETTING COUNTER TO PRODUCE DELAY.
        W1:
        NOP
        NOP
        NOP
        NOP
        NOP
        LOOP W1
        POP AX
        POP CX
        RET
```

```
DELAY ENDP

READ_MEM PROC NEAR
        PUSH DX
        PUSH CX
        PUSH AX
        MOV AX,1000H
        MOV DS,AX
        POP AX
        MOV AL,[SI]   ;READING BYTE FROM MEMORY
        CALL DELAY
        POP CX
        POP DX
        RET

READ_MEM ENDP

LOAD_MEM PROC NEAR
        PUSH DX
        PUSH CX
        MOV AL,CH    ;SETTING BIT TO 0 OR 1
        MOV [SI],AL   ;PUSHING BYTE INTO MEMORY

        CALL DELAY
        MOV CL,AL    ;PRINTING THE BIT THAT HAS BEEN ADDED
        ADD AL,'0'
        CALL WRITE_DATA
        CALL DELAY
        CALL DELAY
        CALL DELAY
        CALL CLEAR
        MOV AL,CL
        POP CX
        POP DX
        RET
LOAD_MEM ENDP


START PROC NEAR

        ;MOV AX,0000H
        ;MOV DS,AX
```

```
        MOV AL,10001001B  ;SETTING PORT C AS INPUT AND PORT A,B AS OUTPUT
        OUT CW, AL
        IN AL, PORTC
        AND AL,01H
        JZ RAM2

        RAM1: MOV DI, 0FFEH       ;IF THE SRAM 6116 IS SELECTED; MAX OFFSET OF
6116 IS 0FFEH
        MOV DX,0004H              ;ACTUAL END IS 0FFEH AS STATED ABOVE; THIS IS
FOR TESTING
        ;MOV DX, 0FFEH
        MOV SI,0000H
        PUSH AX
        MOV AX,2000H
        MOV DS,AX
        POP AX
        JMP TESTING
        RAM2: MOV DI,0FFFEH       ;IF THE SRAM 62256 IS SELECTED; MAX OFFSET OF
62256 IS FFFEH
        MOV DX,0004H              ;ACTUAL END IS 0FFFEH AS STATED ABOVE; THIS IS
FOR TESTING
        ;MOV DX,0FFFEH
        MOV SI,0000H
        PUSH AX
        MOV AX,1000H
        MOV DS,AX
        POP AX
        TESTING:

        MOV BH,00H
        MOV BL,01H

        REPEAT1:
        MOV AH,08H  ;SETTING THE COUNTER FOR EACH BYTE

        REPEATING:
        MOV AL,00H
        MOV CH,BH
        CALL LOAD_MEM
        CALL READ_MEM
        AND AL,BL
        MOV CL,AL    ;PRINTING THE BIT THAT HAS BEEN READ
        ADD AL,'0'     ;OFFSETTING THE BIT WITH RESPECT TO '0'
```

```
        CALL WRITE_DATA
        CALL DELAY
        CALL DELAY
        CALL DELAY
        CALL CLEAR
        MOV AL,CL
        CMP AL,CH    ;COMPARE THE ZERO LOADED AND READ
        JNZ LAST

        MOV CH,BL
        CALL LOAD_MEM
        CALL READ_MEM
        AND AL, BL
        MOV CL,AL            ;PRINTING THE BIT THAT HAS BEEN READ
        ADD AL,'0'
        CALL WRITE_DATA
        CALL DELAY
        CALL DELAY
        CALL DELAY
        CALL CLEAR
        MOV AL,CL
        CMP AL,CH    ;COMPARE THE ONE LOADED AND READ
        JNZ LAST

        ROL BL,01            ;SHIFITNG THE 1 IN BL BY 1 UNIT TOWARDS LEFT
        DEC AH               ;DECREASING COUNTER
        MOV AL,CL
        JNZ REPEATING

        CALL WRITE_PASS  ;PRINTING PASS FOR EVERY BYTE THAT IS CORRECT
        CALL WRITE_IC        ;PRINTING IC NUMBER EVERY BYTE THAT IS CORRECT
        CALL CLEAR

        INC SI
        INC SI          ;MEMORY HAS BEEN ODD/EVEN BANKED
        MOV CX,AX            ;PRINTING THE LAST BYTE OF SI TO SHOW IT IS
INCREASING
        MOV AX,SI
        AND AX, 000EH
        ROR AL,01
        ADD AL,'0'
        CALL WRITE_DATA
        CALL DELAY
```

```
        CALL DELAY
        CALL DELAY
        CALL CLEAR
        MOV AX,CX
        MOV CX,0000H
        CMP SI, DX          ;COMPARING WITH FINAL ADDRESS OF THE SRAM CHIP
        JNZ TESTING
        CALL WRITE_PASS  ;PRINTING PASS WHEN THE MEMORY PASSES THE TEST
        CALL WRITE_IC       ;PRINTING IC NUMBER WHEN THE MEMORY PASSES THE
TEST
        JMP ENP

        LAST: CALL WRITE_FAIL    ;PRINTING FAIL WHEN THE MEMORY FAILS THE TEST
        CALL WRITE_IC            ;PRINTING IC NUMBER WHEN THE MEMORY FAILS
THE TEST
        ENP :                ;PRINTING PASS TO SIGNIFY THE END OF TESTING
        RET

START ENDP

.EXIT
END
```

Datasheet For LCD LM020L :

T-41-39

LM020L
LM020LN (EL Backlit Version)

- 16 character x 1 line
- Controller LSI HD44780 is built-in (See page 115).
- +5V single power supply

MECHANICAL DATA (Nominal dimensions)

Module size	80W x 36H x 12T (max.) mm
Effective display area	64.5W x 13.8H mm
Character size (5 x 7 dots)	3.07W x 5.73H mm
Characrer pitch	3.77 mm
Dot size	0.55W x 0.75H mm
Weight	about 25 g

ABSOLUTE MAXIMUM RATINGS

	min.	max.
Power supply for logic ($V_{DD}-V_{SS}$)	0	7.0 V
Power supply for LCD drive ($V_{DD}-V_O$)	0	13.5 V
Input voltage (Vi)	V_{SS}	V_{DD} V
Operating temperature (Ta)	0	50°C
Storage temperature (Tstg)	−20	70°C
EL Power Supply (when fitted) Voltage (VEL)		AC 150 Vms

INTERNAL PIN CONNECTION

Pin No.	Symbol	Level	Function	
1	V_{SS}	–	0V	Power supply
2	V_{DD}	–	+5V	
3	V_O	–	–	
4	RS	H/L	L: Instruction code input H: Data input	
5	R/W	H/L	H: Data read (LCD module→MPU) L: Data write (LCD module←MPU)	
6	E	H, H→L	Enable signal	
7	DB0	H/L	Data bus line Note (1), (2)	
8	DB1	H/L		
9	DB2	H/L		
10	DB3	H/L		
11	DB4	H/L		
12	DB5	H/L		
13	DB6	H/L		
14	DB7	H/L		

Luminescent output of EL (when fitted) at Ø... 25°C Ø... 0°C ... Cod / m2 typ.