Birla Institute of Technology and Science, Pilani

CS F241 Microprocessors and Interfacing

Design Assignment

By:

Akanksha Dara	**2014A7PS104P**
Ishaan Negi	**2014A7PS105P**
Ameesha Mittal	**2014A7PS107P**
Sharat Chandra	**2014A7PS108P**

Submitted to: Nitin Chaturvedi (Instructor-in-Charge)

Date: 24th April, 2016

System to be Designed: FIRE ALARM SYSTEM

Description:

This system checks for abnormal smoke content in a room every two seconds. Under abnormal conditions it throws open two doors and two windows and opens a valve that releases the gas to put-out the fire. An Alarm is also sounded; this alarm is sounded until the smoke level in the room drops to an acceptable level. The smoke detection system is made up of two smoke sensors placed on the ceiling of the room. When the smoke level comes back below the danger level, the doors, windows and valves are closed.

Specifications of the problem:

1. Two smoke detecting sensors are placed on the ceiling, which detects the smoke content in the room.

2. A fire alarm that is sounded when smoke exceeds normal levels.

3. Two doors, two windows and valves are controlled by motors.

4. The motors and alarms are activated only when **both** the sensors detect an abnormal level in smoke levels.

Assumptions made:

1. Clockwise direction closes all doors, valves and windows and anticlockwise direction opens all doors, valves and windows.

2. We are using MC145010 as the smoke sensor. The IC is used with an infrared photoelectric chamber and detection is accomplished by sensing scattered light from minute smoke particles or other aerosols. The output of the smoke sensor ranges from 0V to 5V. An ADC connected to variable voltage source is used in the design to model the output of the smoke sensor.

3. We have interfaced memory as follows:

RAM – minimum 2k chip - 4k ROM – minimum 4k chip - 8k

ROM1 08000H - 09FFFH

RAM1 0A000H – 0AFFFH

4. When we power on the system, all the doors, windows and valve are closed by passing the appropriate control signals through PPI. After this, interrupts are continuously raised to measure the sensor outputs. If both the sensors exceed a threshold value then only doors, windows and valve are opened. In addition to this, an alarm is also sounded.

5. As long as both the sensors are above threshold value, all the doors, windows and valve will remain open. If any sensor falls below the threshold value then close all the doors, windows and valve and alarm is also turned off.

6. The opening of all doors, windows and valve is controlled by a stepper motor which operates in steps with the use of gears. A stepper motor can be programmed to stop rotating after a certain number of steps, which can be programmed through our code.

Components used:

• **6116 (RAM)** – 2 units used

• **2732 (ROM)** – 2 units used

• **ADC0808** (1 unit):

– 8-bit ADC

– 8 channel

– 1 MHz clock input

• **8253 (Programmable interval timer)**

– 24 pin IC

– 1 counter used with a count value of 18

– Counter operated at a clock speed of 1Hz

• **8255 (Programmable Peripheral Interface (PPI) chip)**

– Contains three 8-bit ports.

– 24 input/output pins.

– Port A and Port C lower are input lines

– Port B and port C upper are output lines

• **8259(Programmable Interrupt Controller)**

– Interrupt generated at interval of 100 micro seconds.

• **8086(Microprocessor)**

– Operating Clock Speed – 5MHz

– 40 pins – 20 de-multiplexed Address Lines and 16 de-multiplexed Data lines

• **74LS373(octal latch)** – 3 Latches used

• **74LS245 (Octal Bus Transmitter/Receiver)** – 2 units used

• **74LS04 (Hex inverter)** – 2 not gates used.

• **74LS32 (OR Gate IC)** – 8 OR Gates used

• **74LS08 (AND Gate)** – 1 AND Gate used

• **L297 (Stepper motor controller):**

Signals from your microprocessor and translates them into stepping signals to send to the L298.

• **L298 (Motor driver)**

– Drives the stepper motor

– 4 output pins connected to one motor

• **Smoke Sensor MC145010**

– Used with an infrared photoelectric chamber

• **Red LEDs**

– Used to represent the doors and windows

Address mapping of memory and I/O devices:

We first initialize all the I/O devices i.e. 8255, 8253 and 8259 with the required values.

- 8259 (0018h to 001Ah) has been programmed as follows:

Control Word	Binary Value	Memory Location
ICW1	00010011	0018h
ICW2	10000000	001Ah
ICW3	Not needed	-
ICW4	00000011	001Ah
OCW1	11111110	001Ah

- 8253 (0010h to 0016h) has been programmed as follows:

Control Word	Binary Value	Memory Location
Cnt0	00010011	0018h
Cnt1	10000000	001Ah
Cnt2	not needed	-
CReg	00000011	001Ah

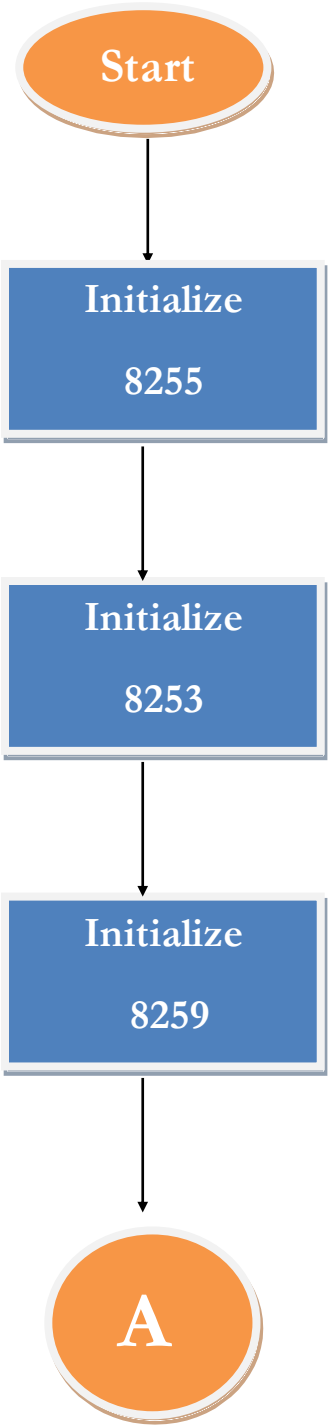
- Programming 8255(PPI chip):

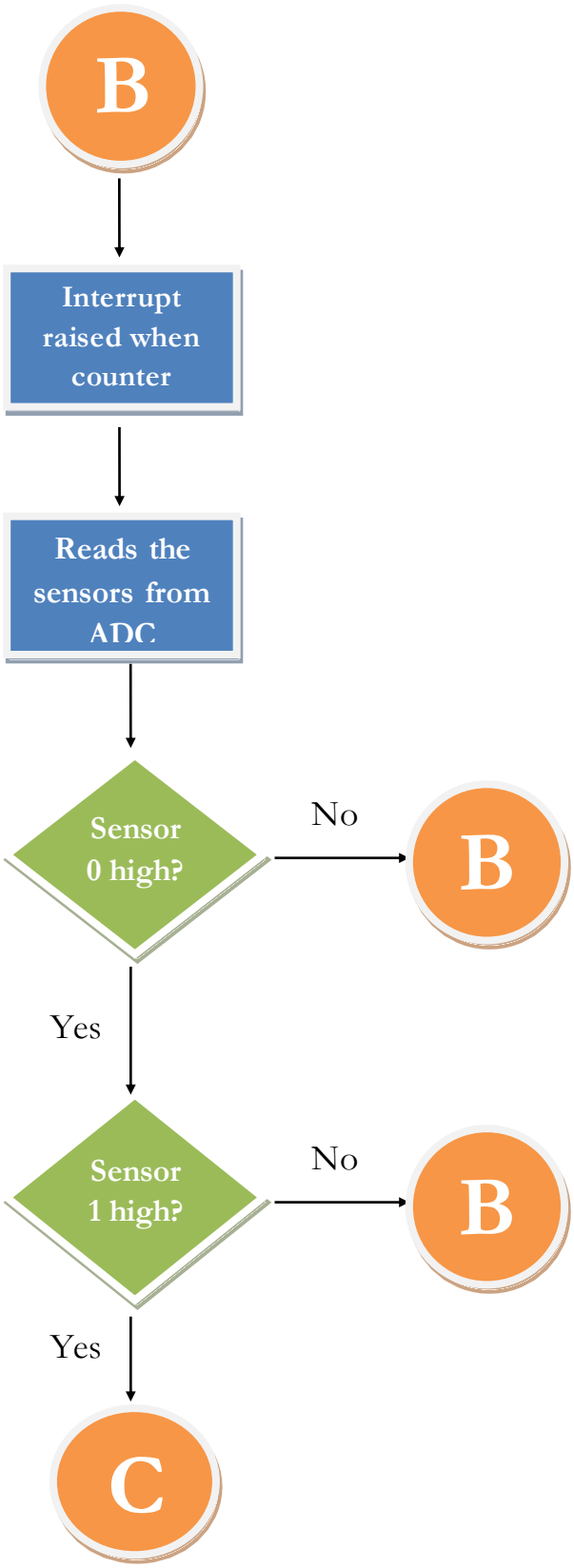
Port	I/O	Address
Port A	Input	0000h
Port B	Output	0002h
Port C – Lower	Input	0004h
Port C – Upper	Output	0006h

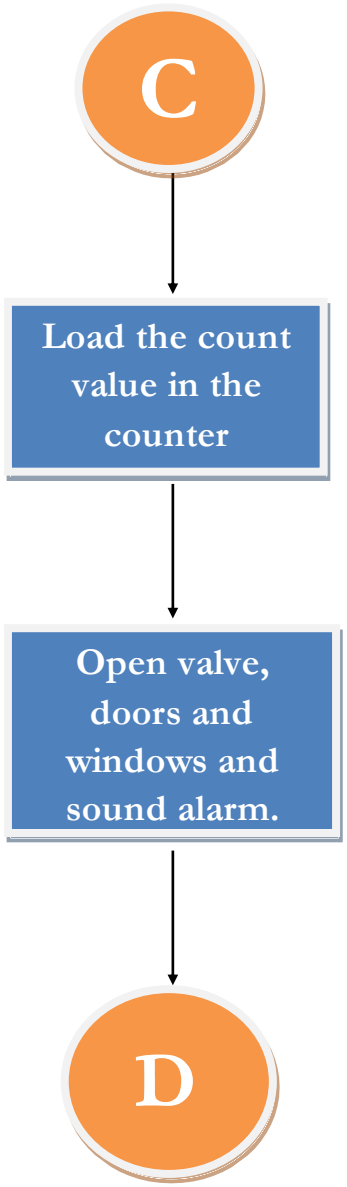
IVT Table:

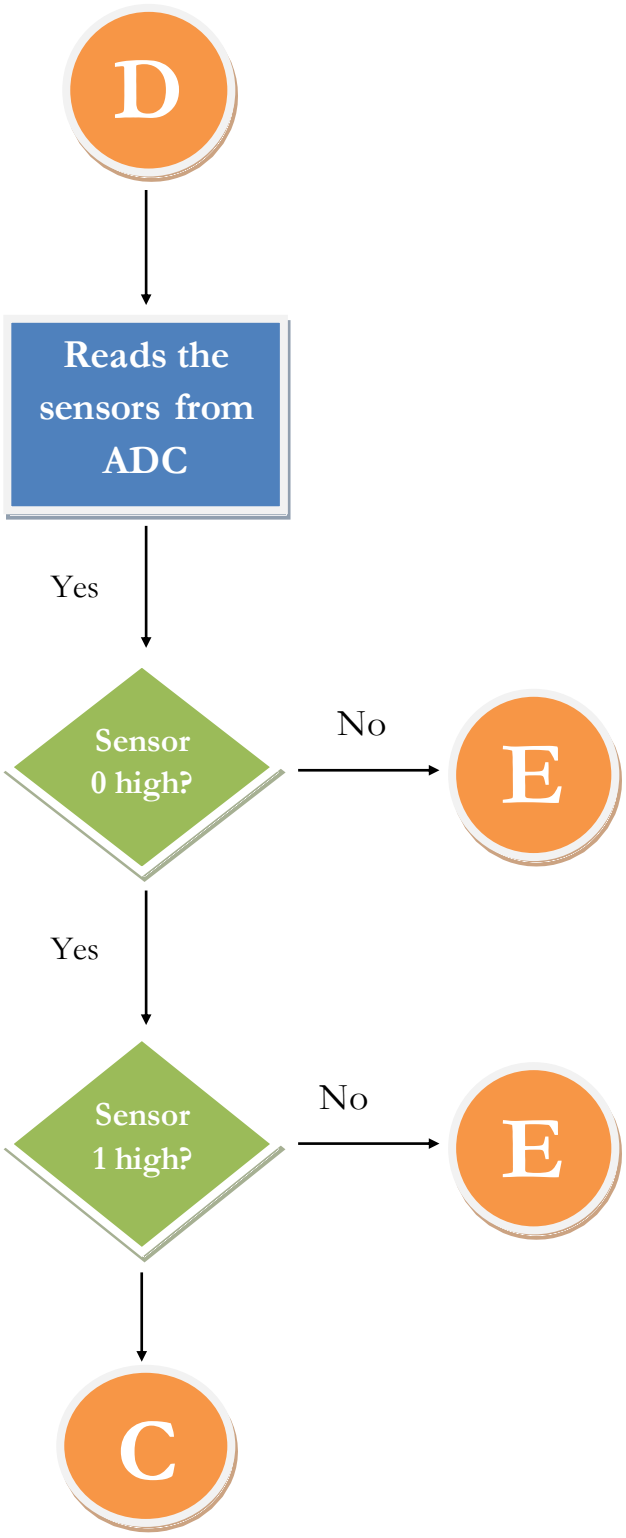
Vector No.	Priority	Offset	Use
80h	1(only one int)	200h	8259 calls this interrupt to check for smoke

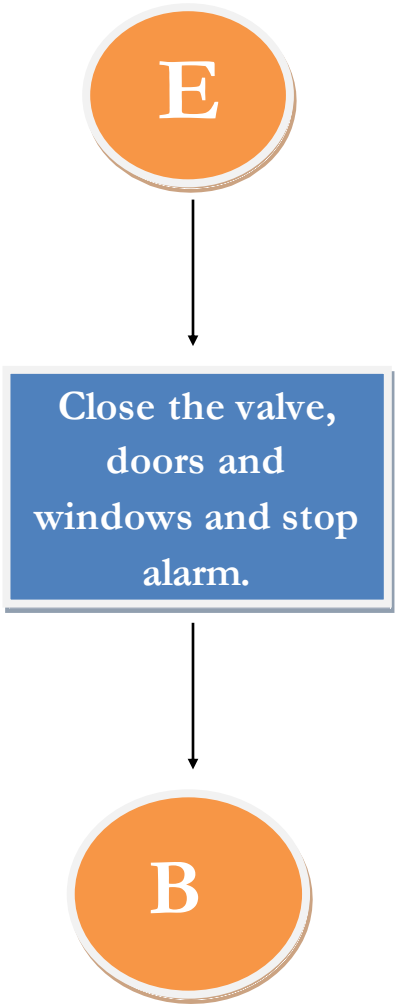
Flowchart for the Software:



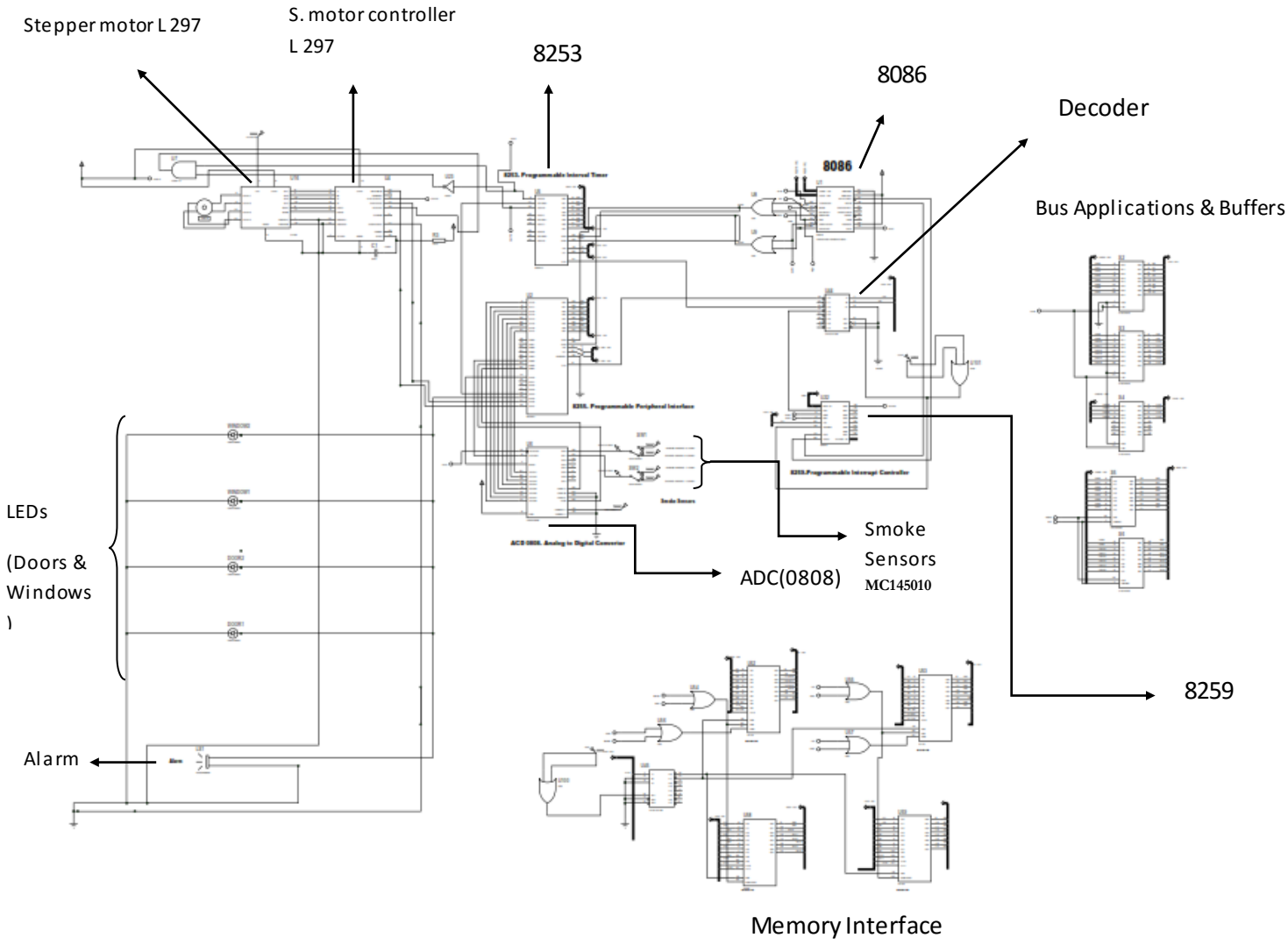

B
Interrupt raised when counter
Reads the sensors from ADC
Sensor 0 high?
No
B
Yes
Sensor 1 high?
No
B
Yes
C


C
Load the count value in the counter
Open valve, doors and windows and sound alarm.
D


D
Reads the sensors from ADC
Yes
Sensor 0 high?
No
E
Yes
Sensor 1 high?
No
E
C

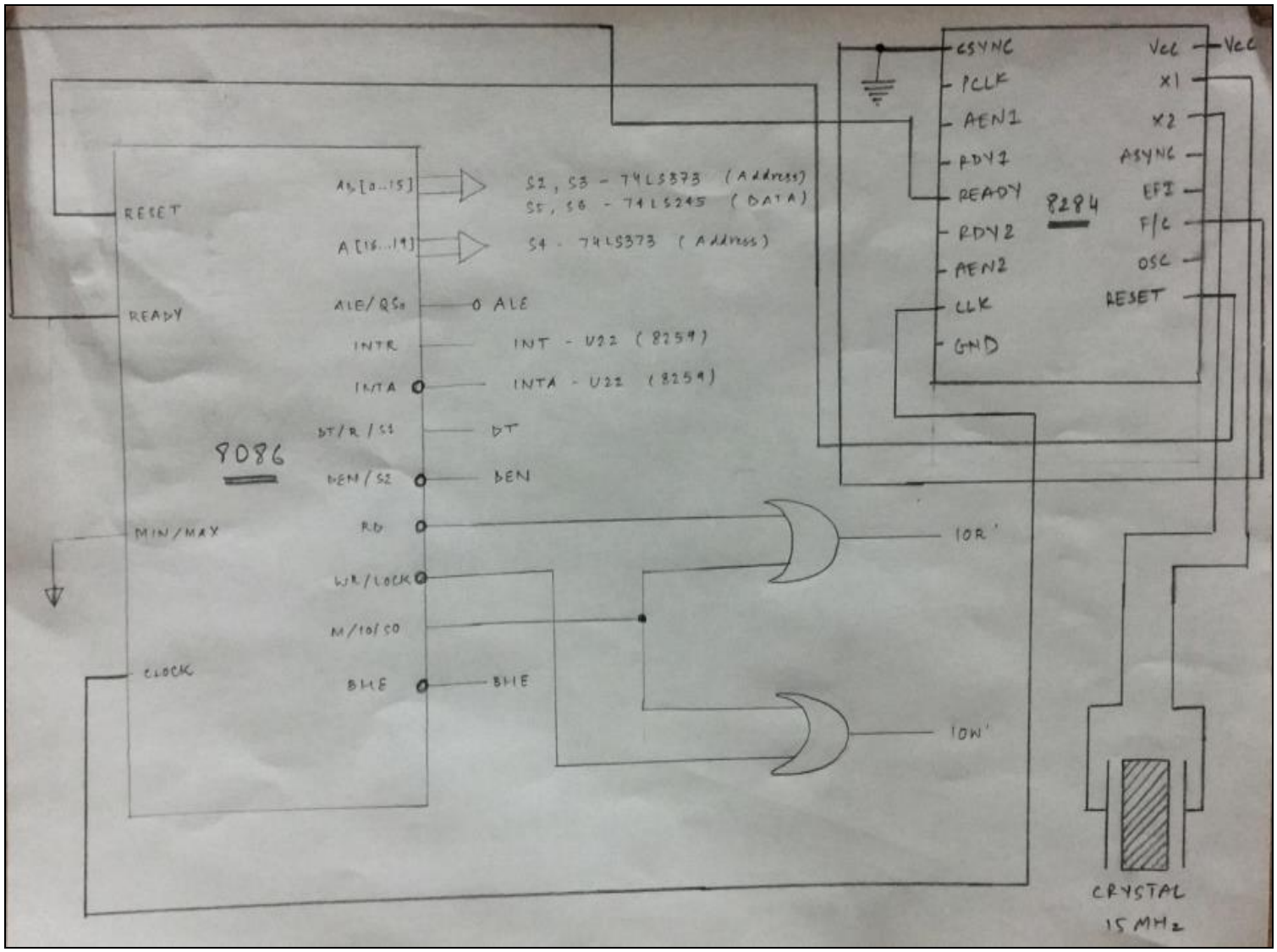

E
Close the valve, doors and windows and stop alarm.
B

Overview of the Design:

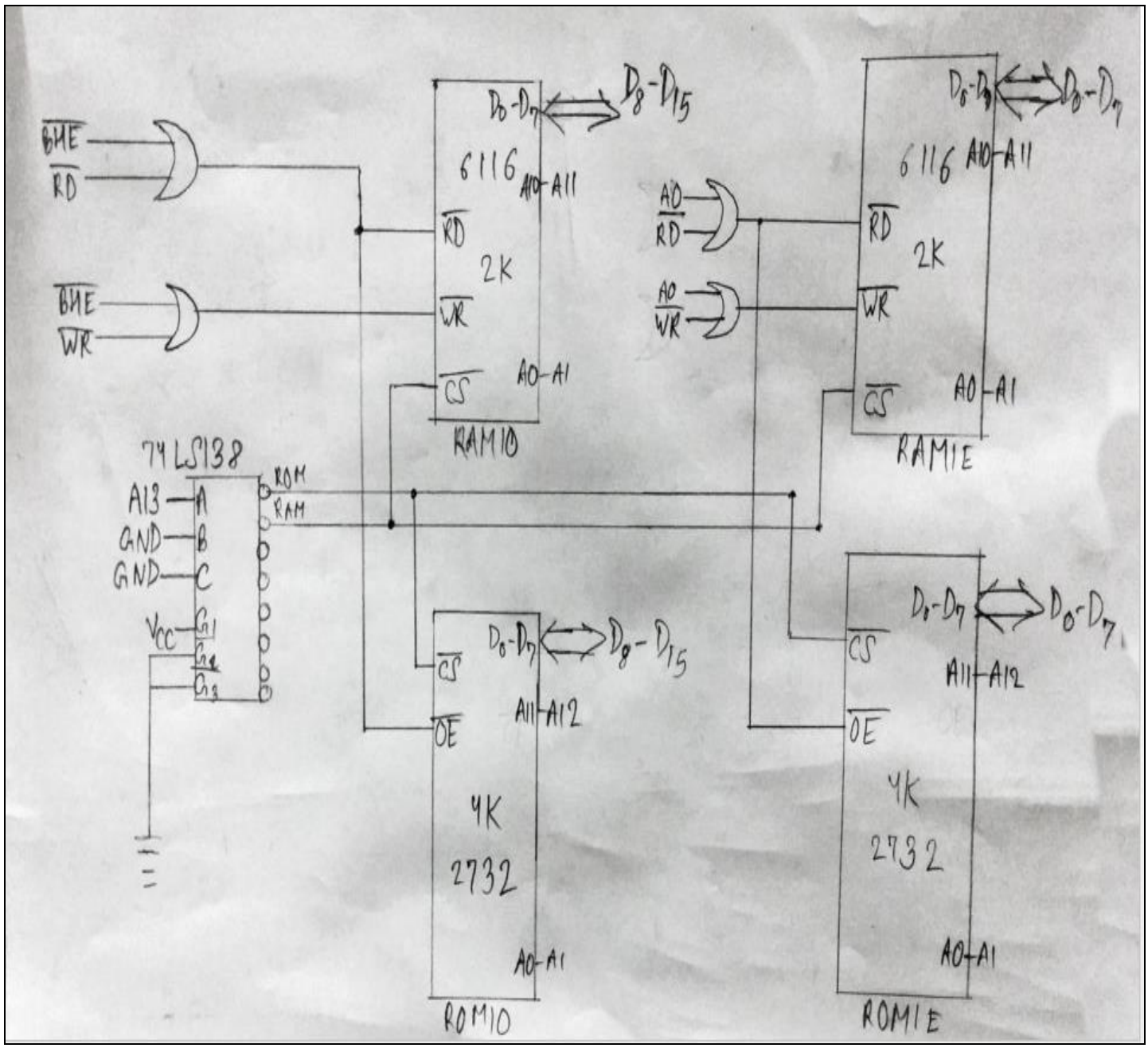


8086 & 8284:

(8284 - used for clock generation)



Memory Interfacing:



ASM Code:

```
#make_bin#
#LOAD_SEGMENT=FFFFh#
#LOAD_OFFSET=0000h#
#CS=0000h#
#IP=0000h#
#DS=0000h#
#ES=0000h#
#SS=0000h#
#SP=FFFEh#
#AX=0000h#
#BX=0000h#
#CX=0000h#
#DX=0000h#
#SI=0000h#
#DI=0000h#
#BP=0000h#
        jmp    st1
        db     509 dup(0)    ;IVT entry for 80H
        dw     t_isr
        dw     0000
        db     508 dup(0)


        st1: cli
        ; intialize ds, es,ss to start of RAM
```

```
 mov      ax,0100h
 mov      ds,ax
mov      es,ax
 mov      ss,ax
mov      sp,0FFFEH
;initialize 8255
;portAequ    0000h
;portBequ 0002h
;portCequ    0004h
;creg   equ    0006h

;portA = input(ADC output)
;portB = output(selection lines for ADC)
;portC lower = input(PC0 connected to EOC of ADC)
;portC upper = output(To valve,windows,doors and Alarm)
mov al, 10010001b
out 0006H, al

;Resets ADC
mov al, 00000000b
out 0004H, al

;8259 - 18H to 1AH
;8259 - enable IRO alone use AEOI
mov      al,00010011b
out      18h,al          ;icw1
mov      al,80H
```

```
        out     1Ah,al          ;icw2
        mov      al,03h
        out     1Ah,al          ;icw4
        mov      al,0FEh
        out     1Ah,al          ;ocw1

x11:
        mov al, 11010000b
        out 0004H, al
        ;initialize 8253
        sti

x12:
        ;cnt0  equ    0010h
        ;cnt1  equ    0012h
        ;cnt2  equ    0014h
        ;creg  equ    0016h

        mov al, 00110000b
        out 0016H, al

        mov al, 0Ah   ;Assuming clock to be 2.5hz
        out 0010H, al

        mov al, 00h
        out 0010H, al
```

```
        mov al,00010000b
        out 0004h,al            ; port C gate enable
        mov al, 00000000b
        out 0002h, al           ;send address 000 to ADC

        mov al, 01000000b ;set ALE
        out 0002h, al

        mov al, 01100000b ;set start
        out 0002h, al

        mov al, 01000000b ;set ALE and clear start
        out 0002h, al

x13:in al, 0004h        ;polls for EOC signal
        and al, 01h
        cmp al, 01h
        jnz x13                 ;if EOC is low, loop back to x13, else proceed
        mov al, 00000000b
        out 0002h, al

        in al, 0000h   ;since EOC is high, take the input from the ADC of smoke sensor 0
        out 0002h, al

        mov cl, 90h             ;CL has threshold smoke value(90h)
        cmp al, cl              ;compare al with the threshold value
        jna x0                  ;if al < danger level, jump to x0
```

```
        mov bl, al
        mov al, 00h

        mov al, 10000000b
        out 0002h, al          ;send address 001 to ADC

        mov al, 11000000b              ;set ALE
        out 0002h, al

        mov al, 11100000b              ;Set Start
        out 0002h, al

        mov al, 11000000b              ;Reset start
        out 0002h, al

x14:    in al, 0004h                   ;polls for EOC signal
        and al, 01h
        cmp al, 01h
        mov al, 00000000b ;reset ADC
        out 0002h, al

        jnz x14                        ;if EOC is low, loop back to x4, else proceed
        mov al, 00h
        in al, 0000h   ;since EOC is high, take input from ADC of smoke sensor 1
        in al, 0000h
        out 0002h, al
```

```
mov cl, 90h          ;CL has threshold smoke value(90h)
cmp al, cl           ;compare al with the threshold value
jna x0

mov al, 00000000b
out 0002h, al

mov al, 10110000b
out 0004h, al        ;rotating the motors in clockwise direction
mov dl, 01h          ;set state of doors, windows, valves as open and raise alarm
; checking if the smoke has reduced
mov al, 00000000b
out 0002h, al        ;send address 000 to ADC
mov al, 01000000b
out 0002h, al
mov al, 01100000b
out 0002h, al
mov al, 01000000b
out 0002h, al


x23:
in al, 0004h             ;polls for EOC signal
and al, 01h
cmp al, 01h
jnz x23                  ;if EOC is low, loop back to x23, else proceed
```

```
mov al, 00000000b
out 0002h, al

in al, 0000h   ;since EOC is high, take the input from the ADC of smoke sensor 0
out 0002h, al

mov cl, 93h
cmp al, cl                ;compare al with an arbitrary value
jna x17                   ;if al <= danger level, jump to x7
mov bl, al
mov al, 00h
mov al, 10000000b
out 0002h, al             ;send address 001 to ADC
mov al, 11000000b
out 0002h, al
mov al, 11100000b
out 0002h, al
mov al, 11000000b
out 0002h, al

x24:
in al, 0004h   ;polls for EOC signal
and al, 01h
cmp al, 01h
mov al, 00000000b
out 0002h, al
```

```
        jnz x24                 ;if EOC is low, loop back to x24, else proceed

        mov al, 00h
        in al, 0000h   ;since EOC is high, take input from ADC of smoke sensor 1
        in al, 0000h
        out 0002h, al

        mov cl, 93h
        cmp al, cl          ;compare al with the arbitrary value
        jna x17             ;if al <= danger level, jump to x7
        jmp x12             ;if al > danger level for both smoke sensors, jump to x12

x17:
        ;close doors, windows, valve and alarm sound



        mov al, 11010000b
        out 0004h, al
        mov dl, 00h                 ;set previous state bit back to 0
        jmp x0

x0: jmp x0                          ; initial run till an interrupt is raised

t_isr:
        ; IVT code
```

```
x2:     ;cnt1   equ     0012h
        ;cnt2   equ     0014h
        ;creg   equ     0016h

        mov al, 00110000b
        out 0016H, al
        mov al, 15h
        out 0010H, al
        mov al, 00h
        out 0010H, al
        mov al,00010000b
        out 0004h,al            ; port C gate enable
        mov al, 00000000b
        out 0002h, al
        mov al, 01000000b
        out 0002h, al

        mov al, 01100000b
        out 0002h, al
        mov al, 01000000b
        out 0002h, al

x3:     in al, 0004h  ;polls for EOC signal
        and al, 01h
        cmp al, 01h
        jnz x3
        ;if EOC is low, loop back to x3, else proceed
```

```
mov al, 00000000b
out 0002h, al

in al, 0000h
;since EOC is high, take the input from the ADC of smoke sensor 0
out 0002h, al

mov cl, 93h
cmp al, cl              ;compare al with an arbitrary value

pushf
pop bx
and bx, 0080h
cmp bx, 0000h
jnz x20                 ;if al < danger level, jump to x20

mov bl, al
mov al, 00h

mov al, 10000000b
out 0002h, al           ;send address 001 to ADC
mov al, 11000000b
out 0002h, al
mov al, 11100000b
out 0002h, al
mov al, 11000000b
out 0002h, al
```

```
x4:   in al, 0004h          ;polls for EOC signal
      and al, 01h
      cmp al, 01h
      mov al, 00000000b
      out 0002h, al

      jnz x4                ;if EOC is low, loop back to x4, else proceed

      mov al, 00h

      in al, 0000h  ;since EOC is high, take input from ADC of smoke sensor 1
      in al, 0000h
      out 0002h, al

      mov cl, 93h
      cmp al, cl            ;compare al with the arbitrary value

      pushf
      pop bx
      and bx, 0080h
      cmp bx, 0000h
      jnz x20                     ;if al < danger level, jump to x20

      ; rotating the motors in clockwise direction and enabling gate 0

      mov al, 00000000b
```

```
        out 0002h, al
        mov al, 10110000b
        out 0004h, al
        mov dl, 01h          ;set state of doors, windows, valves as open
        ; checking if the smoke has reduced
        mov al, 00000000b
        out 0002h, al        ;send address 000 to ADC
        mov al, 01000000b
        out 0002h, al
        mov al, 01100000b
        out 0002h, al
        mov al, 01000000b
        out 0002h, al

x5:     in al, 0004h         ;polls for EOC signal
        and al, 01h
        cmp al, 01h
        jnz x5               ;if EOC is low, loop back to x5, else proceed
        mov al, 00000000b
        out 0002h, al

        in al, 0000h  ;since EOC is high, take the input from the ADC of smoke sensor 0
        out 0002h, al

        mov cl, 93h
        cmp al, cl           ;compare al with the arbitrary smoke threshold value.
```

```
        jle x7

        ;if al <= danger level, jump to x7

        mov bl, al
        mov al, 00h

        mov al, 10000000b
        out 0002h, al          ;send address 001 to ADC
        mov al, 11000000b
        out 0002h, al

        mov al, 11100000b
        out 0002h, al
        mov al, 11000000b
        out 0002h, al

x6:     in al, 0004h  ;polls for EOC signal
        and al, 01h
        cmp al, 01h
        mov al, 00000000b
        out 0002h, al

        jnz x6                 ;if EOC is low, loop back to x6, else proceed

        mov al, 00h
```

```
        in al, 0000h   ;since EOC is high, take input from ADC of smoke sensor 1
        in al, 0000h
        out 0002h, al

        mov cl, 93h
        cmp al, cl              ;compare al with the arbitrary value
        jna x7

        ;if al <= danger level, jump to x7

        jmp x2                  ;if al > danger level for both smoke sensors, jump to x2
(checks again)

x7:
        ;close doors, windows and valve
        mov al, 11010000b
        out 0004h, al
        mov dl, 00h                     ;set previous state bit back to 0

x20:
        iret
```

Specifications of the Smoke Sensor used:

MC145010

1. Operating Voltage Range: 6V to 12V
2. Operating Temperature Range: -10°C to 60°C
3. Average Supply Current: 12 µA

The pin assignment is as follows: