BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

MICROPROCESSORS AND INTERFACING (CS F241)

DESIGN ASSIGNMENT

"FIRE ALARM SYSTEM"

Group Number :- 34 **Project Number :- 20**

Submitted by:

1.) Varun Gupta	**(2016A7PS0087P)**
2.) Padma Sai Dora Babu	**(2016A7PS0084P)**
3.) Nishchay Agrawal	**(2016A7PS0088P)**
4.) Gunraj Singh	**(2016A7PS0085P)**

PROBLEM STATEMENT

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.

ASSUMPTIONS

In order to reduce the design effort as well as to get the simplest design possible using the minimum amount of hardware while meeting all the restrictions specified by our problem statement, we have made the following assumptions:

- The smoke sensors used come with a built in ADC. It gives high output (1) in case smoke is present near it but gives low output (0) when smoke is not present.
- Abnormal conditions are considered when both the smoke sensors give high output.
- Dangerous conditions are considered when only one of the sensors gives high output.
- Below the danger level, both the sensors give low output and the doors, windows and valve are closed.
- We have used LEDs to simulate the 2 doors, 2 windows, 1 valve and 1 alarm.

DESIGN PROCESS

(A)Components Used:

Chip Number	Quantity	Chip	Purpose
8086	1	Microprocessor	Central Processing Unit
8253A	1	Programmable Interval Timer	Used to generate timing signals
8255A	1	Programmable Peripheral Interface	Used to interface ports as input and output for smoke sensors, opening and closing of doors, valves and windows and the sounding of alarm
2732	2	EPROM	Read only Erasable Programmable memory to house code segment
6116	2	RAM	Read Write Memory to house the segment
-	33	2 input OR Gate	Used for Decoding Logic
-	5	NOT Gate	Used for Decoding Logic
-	6	LED (equivalent to mechanical switches relay and alarms in chart)	Simulate doors, windows and valves
74LS373	3	8 bit buffer bidirectional buffers	Buffering address bus
74LS245	2	8 bit buffer bidirectional buffers	Buffering data bus
74LS244	2	Octal Buffers	Creation of Vector number (40h)

(B) Memory Organization:

Total RAM used by system = 4KB

Total ROM used by system = 8KB

RAM chip used = 6116.

So, size of each RAM chip = 16/8 = 2KB.

Hence, number of 6116 RAM chips required = 4KB/2KB = 2.

EPROM chip used = 2732

So, size of each ROM chip = 32/8 = 4KB.

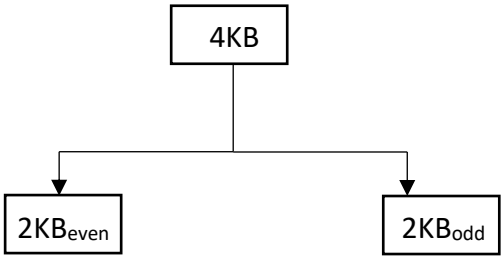
Hence, number of 2732 EPROM chips required = 8KB/4KB = 2.

Let us consider RAM first.

- ✓ **RAM must house the data segment and stack segment.**

Starting address of RAM = 00000_H

System has 4KB RAM. So, Ending address of RAM = $00FFF_H$.



Even Bank of RAM = 00000_H, 00002_H, 00004_H, 00006_H, ... , $00FFE_H$.

Odd Bank of RAM = 00001_H, 00003_H, 00005_H, 00007_H, ... , $00FFF_H$.

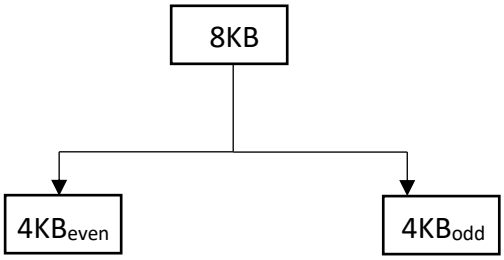
	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
Start Address	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
End Address	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

Now, let us consider ROM.

- ✓ **ROM must house the code segment.**

Starting address of ROM = $FD000_H$

System has 8KB ROM. So, Ending address of ROM = $FEFFF_H$.



Even Bank of ROM = $FD000_H$, $FD002_H$, $FD004_H$, $FD006_H$, ... , $FEFFE_H$.

Odd Bank of ROM = $FD001_H$, $FD003_H$, $FD005_H$, $FD007_H$, ... , $FEFFF_H$.

	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**
Start Address	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
End Address	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1

(C) IO BASED MEMORY MAPPING:

1.) 8253 Programmable Interval Timer

According to our problem statement, we need to check values from sensors every 2 seconds. For this we need a timer and hence we are using 8253. It is organized in the following way:

	Address	A7	A6	A5	A4	A3	A2	A1	A0
Port A	30_H	0	1	0	1	0	0	0	0
Port B	32_H	0	1	0	1	0	0	1	0
Port C	34_H	0	1	0	1	0	1	0	0
Control Reg.	36_H	0	1	0	1	0	1	1	0

2.)8255 Programmable Peripheral Interface

One 8255(Programmable Peripheral Interface) are used to communicate with other input and output devices. It is organized in the following manner.

	Address	A7	A6	A5	A4	A3	A2	A1	A0
Port A	00_H	0	0	0	0	0	0	0	0
Port B	02_H	0	0	0	0	0	0	1	0
Port C	04_H	0	0	0	0	0	1	0	0
Control Reg.	06_H	0	0	0	0	0	1	1	0

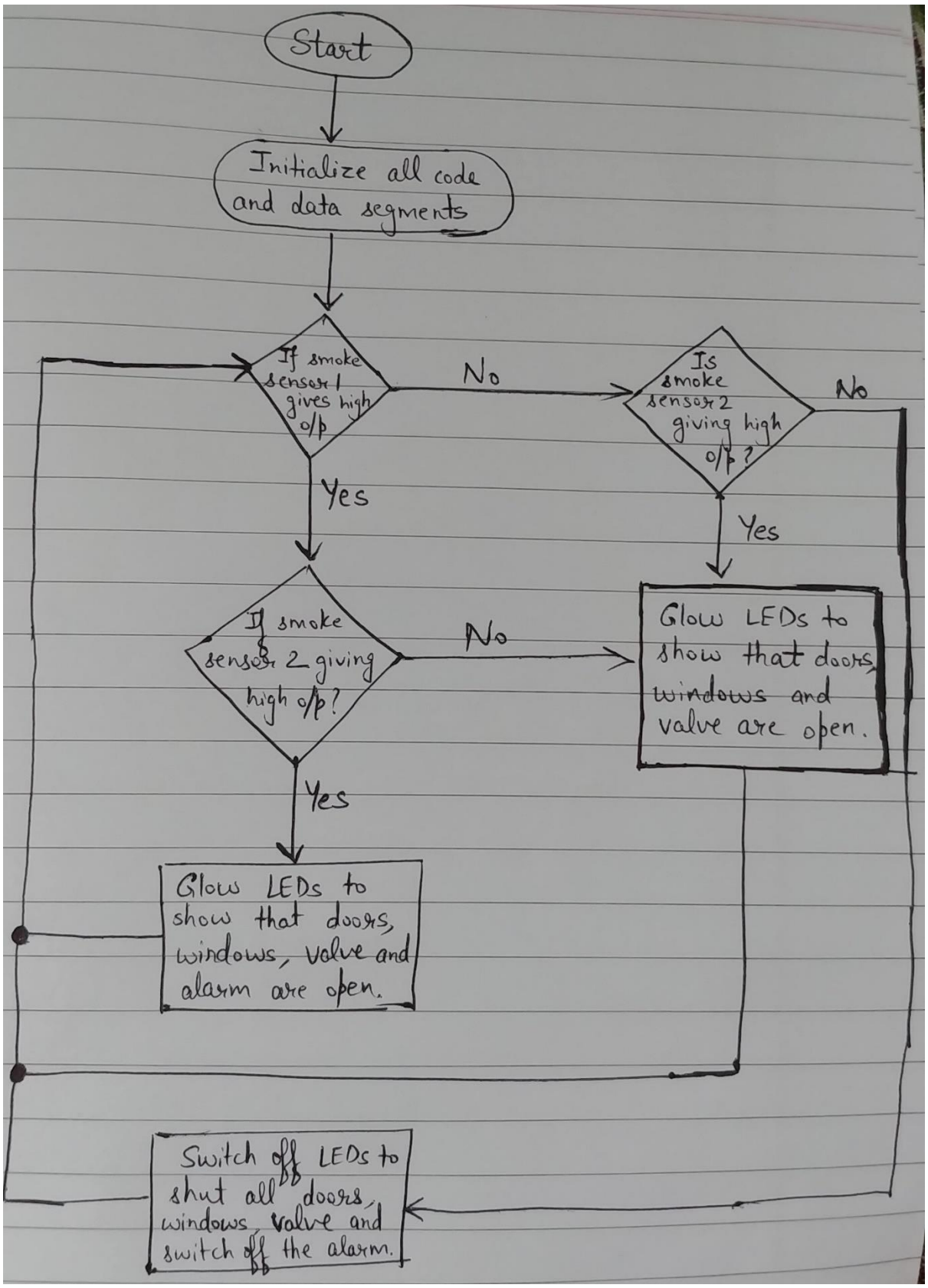
(D)SENSOR:

MQ-2 Smoke Sensor circuit with built in Analog to digital convertor (with logic 1 as output in case of presence of smoke else logic 0 is the output)

The 8253 is used in Mode 2 as a rate counter where an interrupt is given to the 8086 microprocessor every 2 seconds. On the interrupt the 8086 checks for the values at the smoke detectors, that is if it is logic 1 or logic 0. On the basis of the following truth table the actions are taken :-

SENSOR 1	SENSOR 2	ACTION TAKEN
0	0	Close door, valves and windows
0	1	Close door, valves and windows
1	0	Close door, valves and windows
1	1	Open door, valves and windows

FLOWCHART


Start
Initialize all code and data segments
If smoke sensor 1 gives high o/p
No
Is smoke sensor 2 giving high o/p ?
No
Yes
Yes
If smoke sensor 2 giving high o/p?
No
Glow LEDs to show that doors, windows and valve are open.
Yes
Glow LEDs to show that doors, windows, valve and alarm are open.
Switch off LEDs to shut all doors, windows, valve and switch off the alarm.

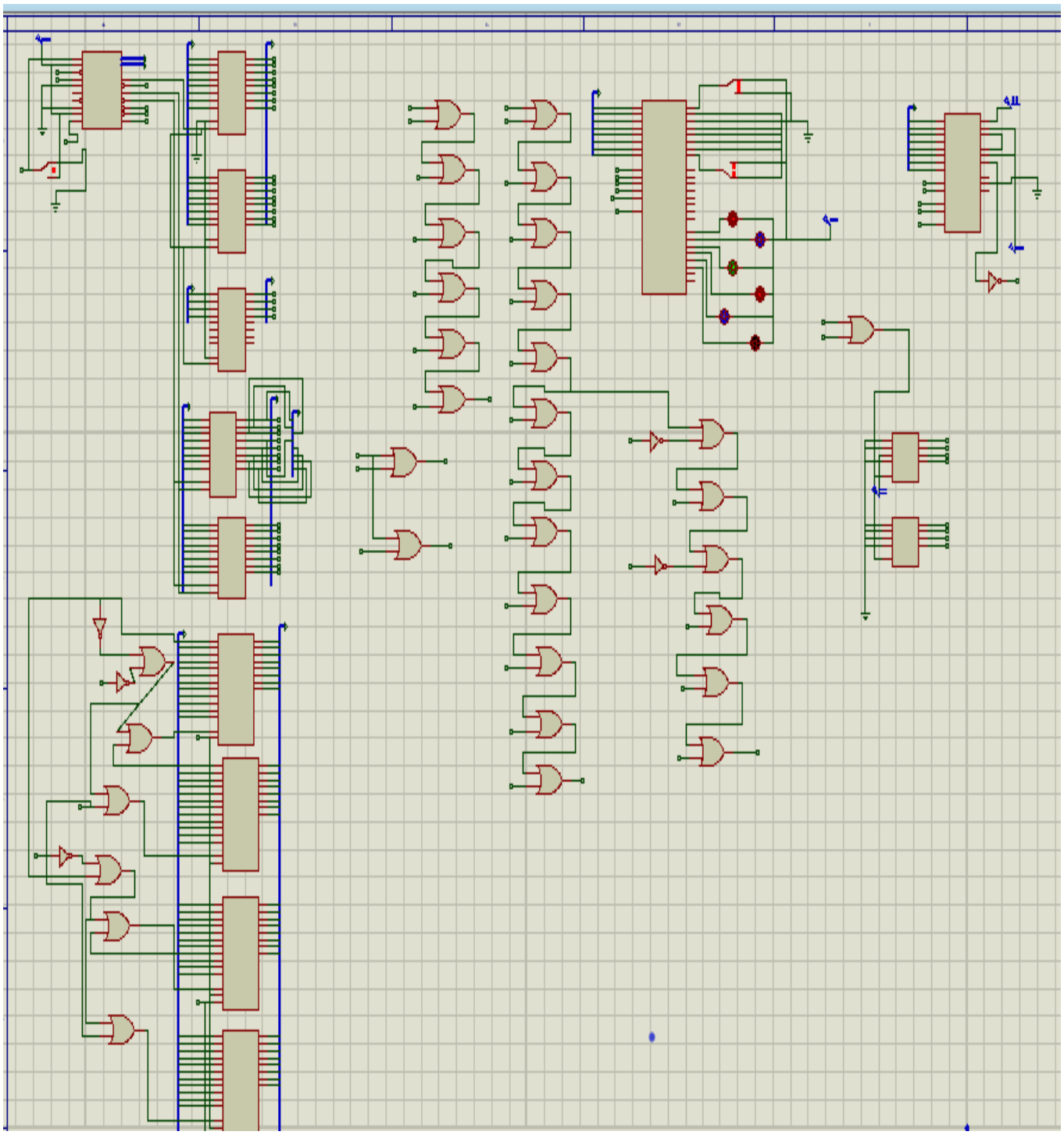
CODE

```
.model tiny
#make_bin#
#LOAD_SEGMENT=FFFFh#
#LOAD_OFFSET=0000h#

.code
.startup
      jmp     st1
      db      1024 dup(0)
st1:
          cli
; initialize port a & b as input & port c as output
      mov     al,92h       ;10010010b
      out     06h, al

; initialize counter mode 2 in 8253
      ;30h = address of count0 i.e. starting address for 8253
      ;count reqd = 5*10^6
      ;count0 stores 2500
      mov     al,34h
      out     36h,al
      mov     al,0C4h
      out     30h,al
      mov     al,09h
      out     30h,al

      ;count1 stores 2000
      mov     al,74h
```

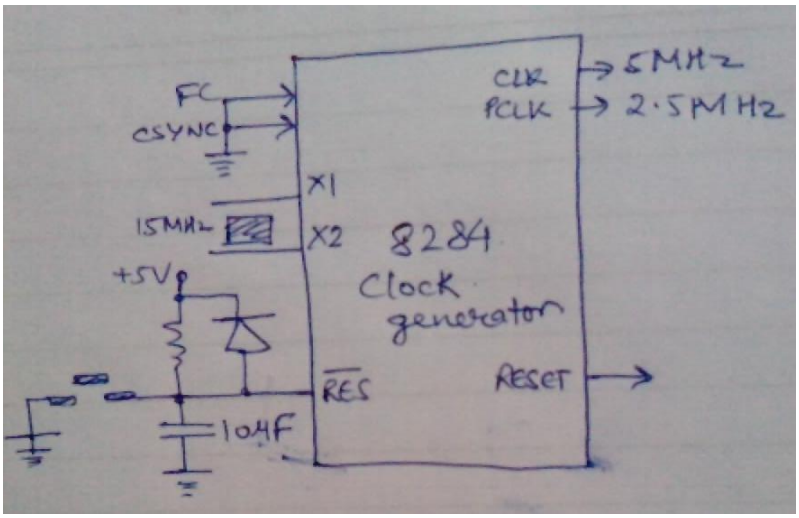
```
        out     36h,al
        mov     al,0D0h
        out     32h,al
        mov     al,07h
        out     32h,al

        ;take input from smoke sensors
next:   in      al,00h
        mov     bl,al
        ;check if both are on or not
        cmp     bl,81h
        jz      glow
        cmp     bl,80h
        jz      offAlarm
                cmp     bl,01h
        jz      offAlarm
              ;close valves, doors and windows
off:      mov         al,0FFh
              out       04h,al
        jmp     over

       ;open valves, doors and windows and sound alarm
glow:   mov     al,0C0h
        out     04h,al
        jmp     over

offAlarm: mov al,0E0h
          out 04h,al
          jmp over
```

```
over:

    ;interupt generation using out of 8253 after every 2 seconds
    lea     si,read
    mov     ds:[256],si  ;vector number = 40h = 64d, so location in IVT = 64*4 = 256
    mov     cx,cs
    mov     ds:[258],cx

    ;set the interrupt flags
    sti

    ;infinite loop, will stop only power is switched off
    jmp     next
.exit
read proc near:
    iret
read endp
end
```

CIRCUIT DIAGRAM



In the above circuit, 8284 was used to generate the required clock at frequencies 5MHz and 2.5 MHz.



REFERENCES TO MANUALS USED

- 8085/86
- x86 processors (Intel processors book by Barey Brey)
- 2716_EPROM
- 6116
- 8255
- SN74LS245
- SN74LS373

All the above manuals were either searched on Internet or available on Edx for student's reference.

For the smoke sensors, the below sites were referred

- http://www.learningaboutelectronics.com/Articles/MQ-2-smoke-sensor-circuit-with-arduino.php
- http://store.fut-electronics.com/products/smoke-sensor-module-mq2-digital-analog
- http://www.nrdcentre.com/MQ2-Smoke-Sensor-Module
- http://www.hw-group.com/products/sensors/index_en.html