

BIRLA INSTITUTE OF TECHNOLOGY AND  
SCIENCE, PILANI



**MICROPROCESSORS AND INTERFACING (CS F241)**  
**DESIGN ASSIGNMENT**

**“FIRE ALARM SYSTEM”**

Group Number :- 76

Project Number :- 16

**Submitted by:**

- |                                   |                 |
|-----------------------------------|-----------------|
| 1.) Rayapati Yaswanth Kumar       | (2017A7PS0038P) |
| 2.) Sikawat Pawankumar Shreegopal | (2017A7PS0039P) |
| 3.) Shaily Bhatt                  | (2017A7PS0040P) |
| 4.) Yaganti Sivakrishna           | (2017A7PS0045P) |

## **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 =  $16/8 = 2\text{KB}$ .

**Hence, number of 6116 RAM chips required =  $4\text{KB}/2\text{KB} = 2$ .**

EPR0M chip used = 2732

So, size of each ROM chip =  $32/8 = 4\text{KB}$ .

Hence, number of 2732 EPROM chips required =  $8\text{KB}/4\text{KB} = 2$ .

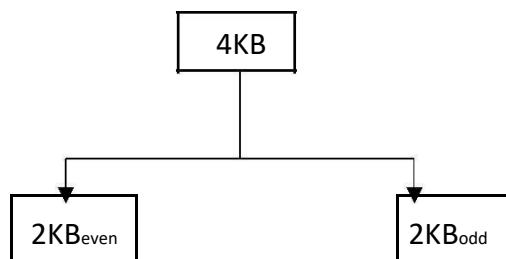
**Let us consider RAM first.**

✓

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

Starting address of RAM = 00000<sub>H</sub>

System has 4KB RAM. So, Ending address of RAM = 00FFF<sub>H</sub>.



**Even Bank of RAM = 00000<sub>H</sub>, 00002<sub>H</sub>, 00004<sub>H</sub>, 00006<sub>H</sub>, . . . . ., 00FFE<sub>H</sub>.**

**Odd Bank of RAM = 00001<sub>H</sub>, 00003<sub>H</sub>, 00005<sub>H</sub>, 00007<sub>H</sub>, . . . . ., 00FFF<sub>H</sub>.**

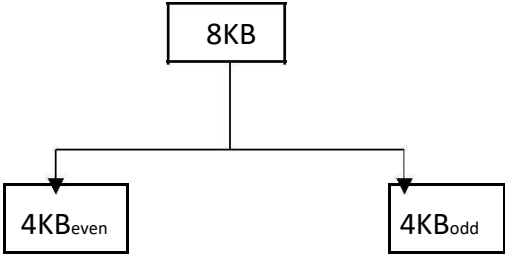
[illegible]



**ROM must house the code segment.**

Starting address of ROM = FD000<sub>H</sub>

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



**Even Bank of ROM** = FD000<sub>H</sub>, FD002<sub>H</sub>, FD004<sub>H</sub>, FD006<sub>H</sub>, . . . . . , FEF<sub>FE</sub><sub>H</sub>.

**Odd Bank of ROM = FD001<sub>H</sub>, FD003<sub>H</sub>, FD005<sub>H</sub>, FD007<sub>H</sub>, . . . . . , FFFFF<sub>H</sub>.**

[illegible]

### **(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 <sub>H</sub>	0	0	1	1	0	0	0	0
Port B	32 <sub>H</sub>	0	0	1	1	0	0	1	0
Port C	34 <sub>H</sub>	0	0	1	1	0	1	0	0
Control Reg.	36 <sub>H</sub>	0	0	1	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 <sub>H</sub>	0	0	0	0	0	0	0	0
Port B	02 <sub>H</sub>	0	0	0	0	0	0	1	0
Port C	04 <sub>H</sub>	0	0	0	0	0	1	0	0
Control Reg.	06 <sub>H</sub>	0	0	0	0	0	1	1	0

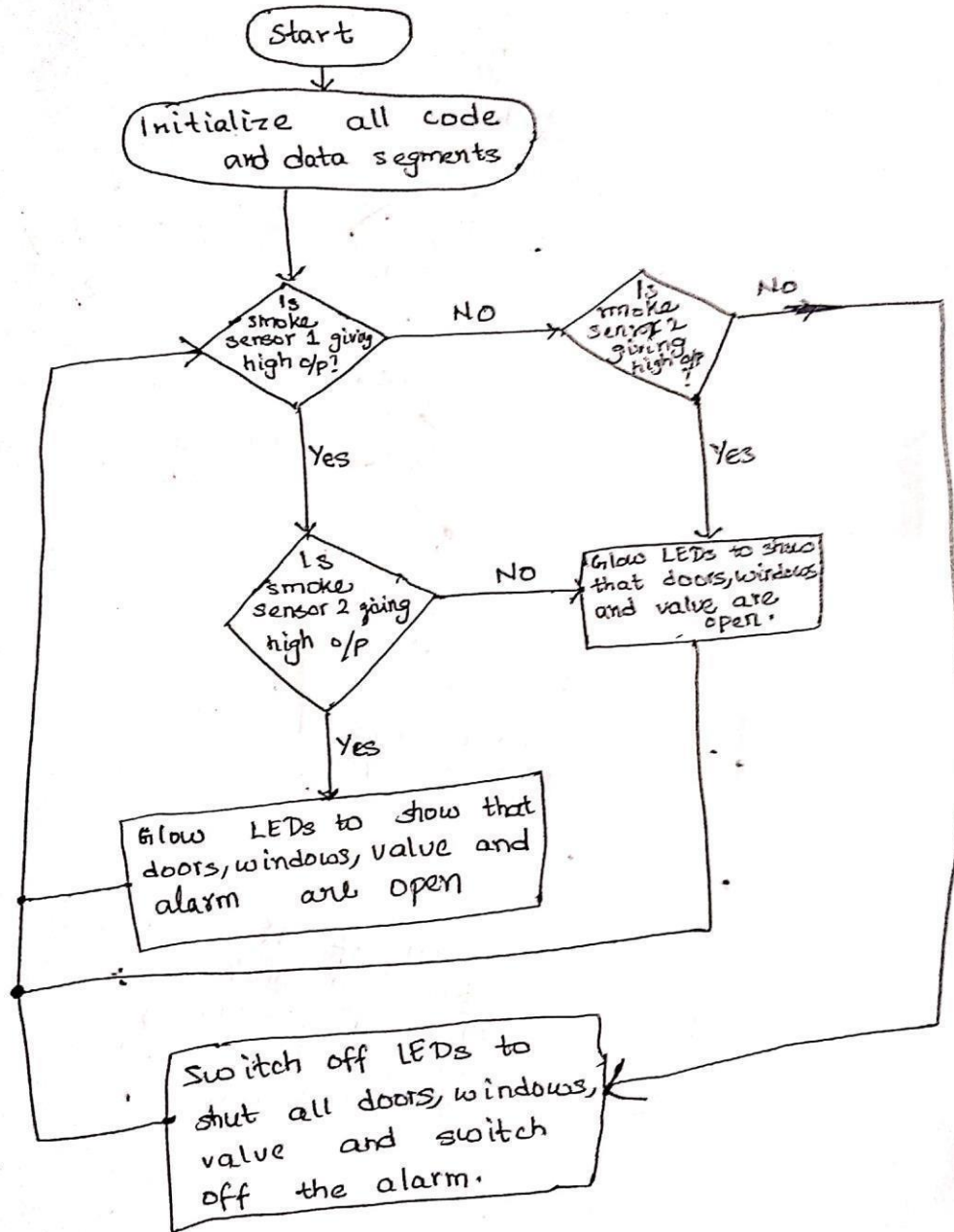
#### **(D)SENSOR:**

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	Open door, valves and windows
1	0	Open door, valves and windows
1	1	Open door, valves and Windows and alarm is sounded

# FLOWCHART





# 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:

;interrupt 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

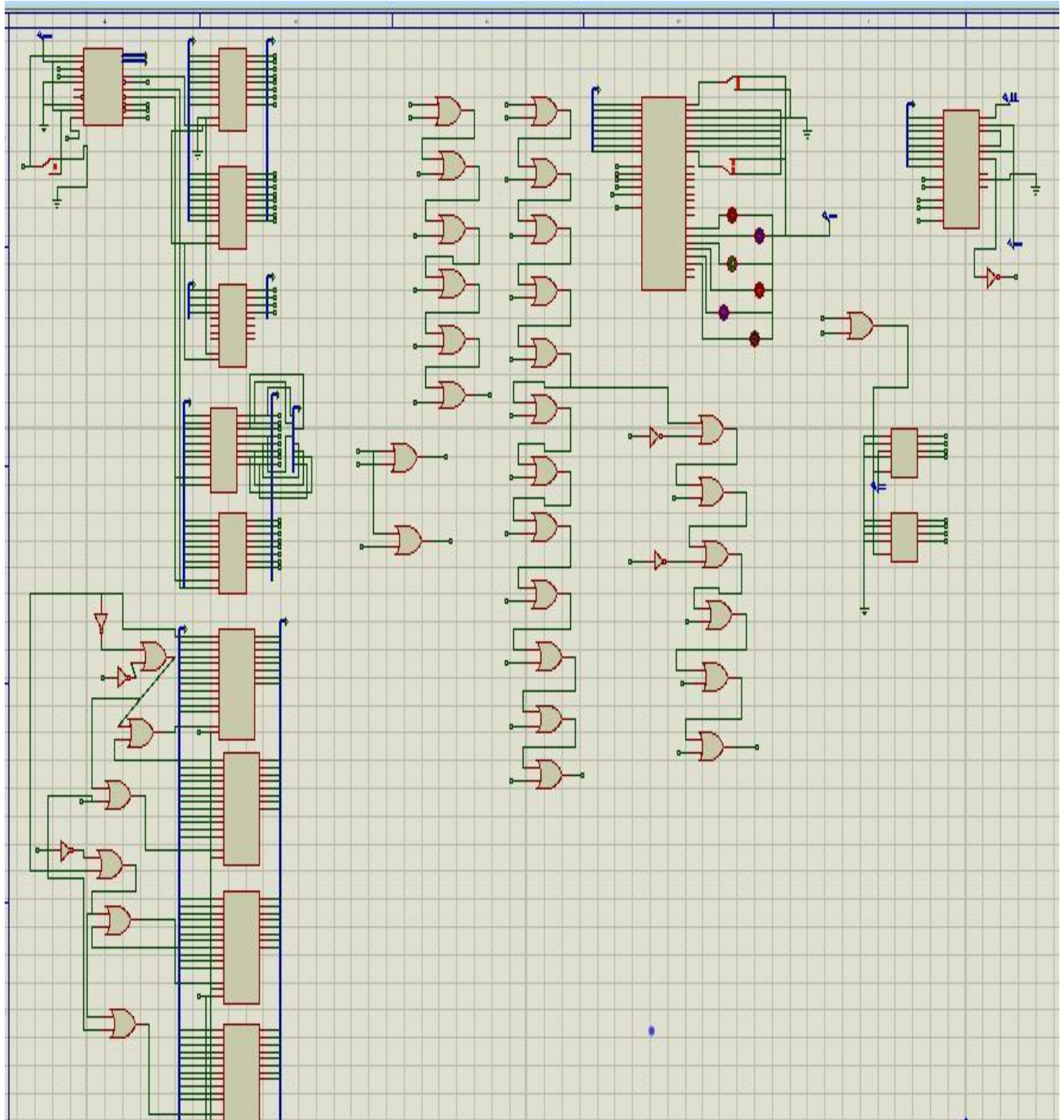
read proc near:

iret

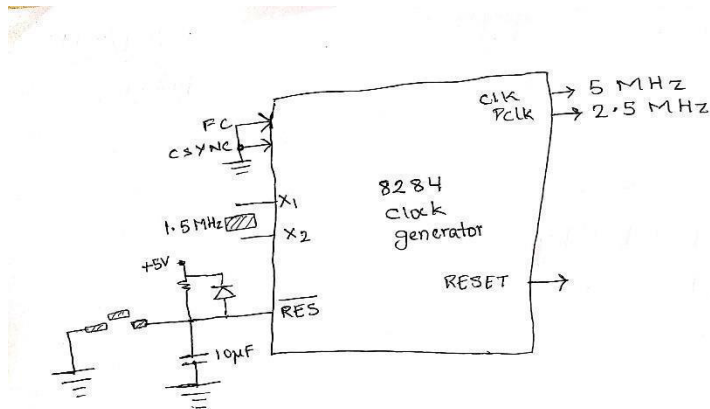
read endp

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\_EEPROM



6116



8255



SN74LS245



SN74LS373

All the above manuals were available on Internet.

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.nrdc-entre.com/MQ2-Smoke-Sensor-Module>
- [http://www.hw-group.com/products/sensors/index\\_en.html](http://www.hw-group.com/products/sensors/index_en.html)