GROUP NUMBER: 36 **PROJECT NUMBER: 20**

MICROPROCESSORS AND INTERFACING DESIGN ASSIGNMENT

"FIRE ALARM SYSTEM"

Submitted by:

1)Saakaar Vashisht (2016A7PS0093P)
2) Saurabh Raj (2016A7PS0094P)
3) Aditi Agarwal (2016A7PS0095P)
4)Gnana Praveen (2016A7PS0096P)

PROBLEM STATEMENT WITH SPECIFICATIONS:

This system checks for abnormal smoke content in a room every 2 seconds. Under abnormal conditions, it throws open 2 doors and 2 windows and opens a valve that releases the gas to put out fire. An alarm is also sounded, this alarm is sounded until smoke level in room drops to an acceptable level. The smoke detection system is made up of 2 smoke sensors placed on the ceiling of the roof. 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:

1) The smoke sensors used come with a built in ADC. Such sensors are actually available in market (MQ-2 smoke sensor). It gives high output (1) in case smoke is present near it but gives low output (0) when smoke is not present.
2) Abnormal conditions are considered when both the smoke sensors give high output.
3) We have used LEDs to simulate the 2 doors, 2 windows, 1 valve and 1 alarm.

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 abnormality in smoke levels.

IVT Table:

Vector No.	Priority	Offset	Use
40h	1(only one int)	200h	8259 calls this interrupt to check for smoke

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 and Write Memory to house thesegment
-	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 bi-directional 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.

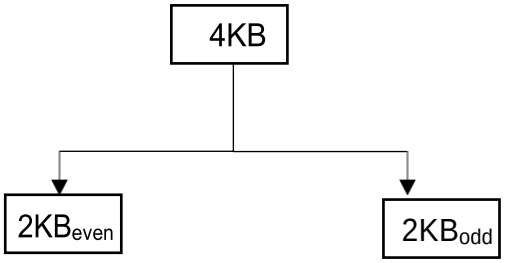
Hence, number of 2732 EPROM chips required = 8KB/4KB = 2.

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 = $00000_H, 00002_H, 00004_H, 00006_H, \ldots\ldots\ldots\ldots\ldots, 00FFE_H$.

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

	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

Data Segment starts at address location 00000_H and is of size 1KB.

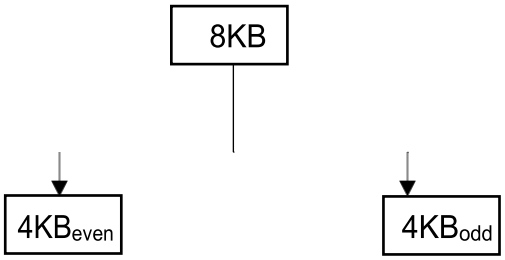
Stack segment starts at address 00800_H and is also of size 1KB.

Now, let us consider ROM.

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

Starting address of ROM = $FE000_H$

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



Even Bank of ROM = $FE000_H, FE002_H, FE004_H, FE006_H, \ldots\ldots\ldots\ldots\ldots, FEFFE_H$.

Odd Bank of ROM = $FE001_H, FE003_H, FE005_H, FE007_H, \ldots\ldots\ldots\ldots\ldots, FEFFF_H$.

	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	1	0	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

The code begins at the address $FF000_H$.

The first address the processor sends out when it is switched on is $FFFF0_H$.

At this location, there is a jump instruction which takes the program control to the beginning of the code.

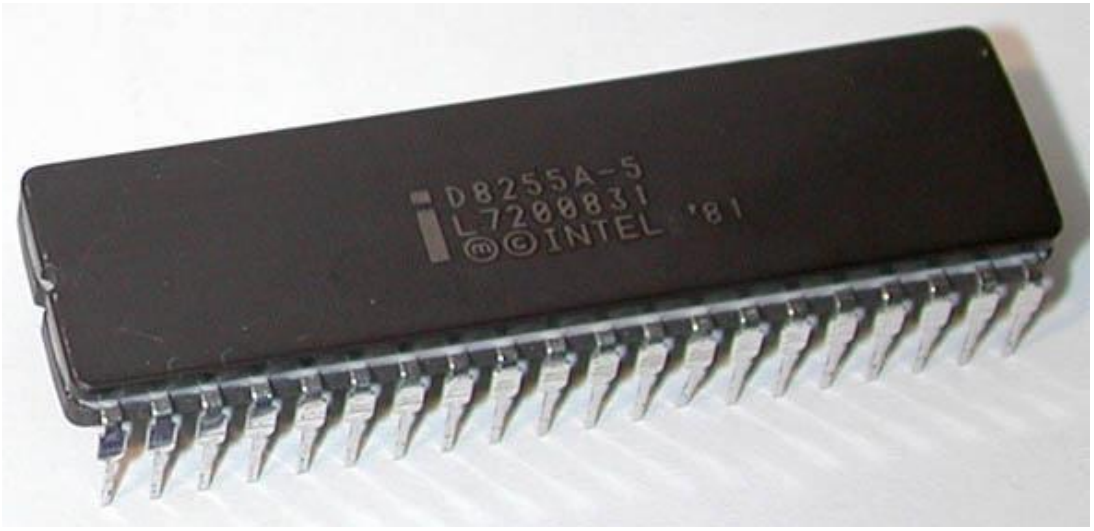
(C) IO BASED MEMORY MAPPING:

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:

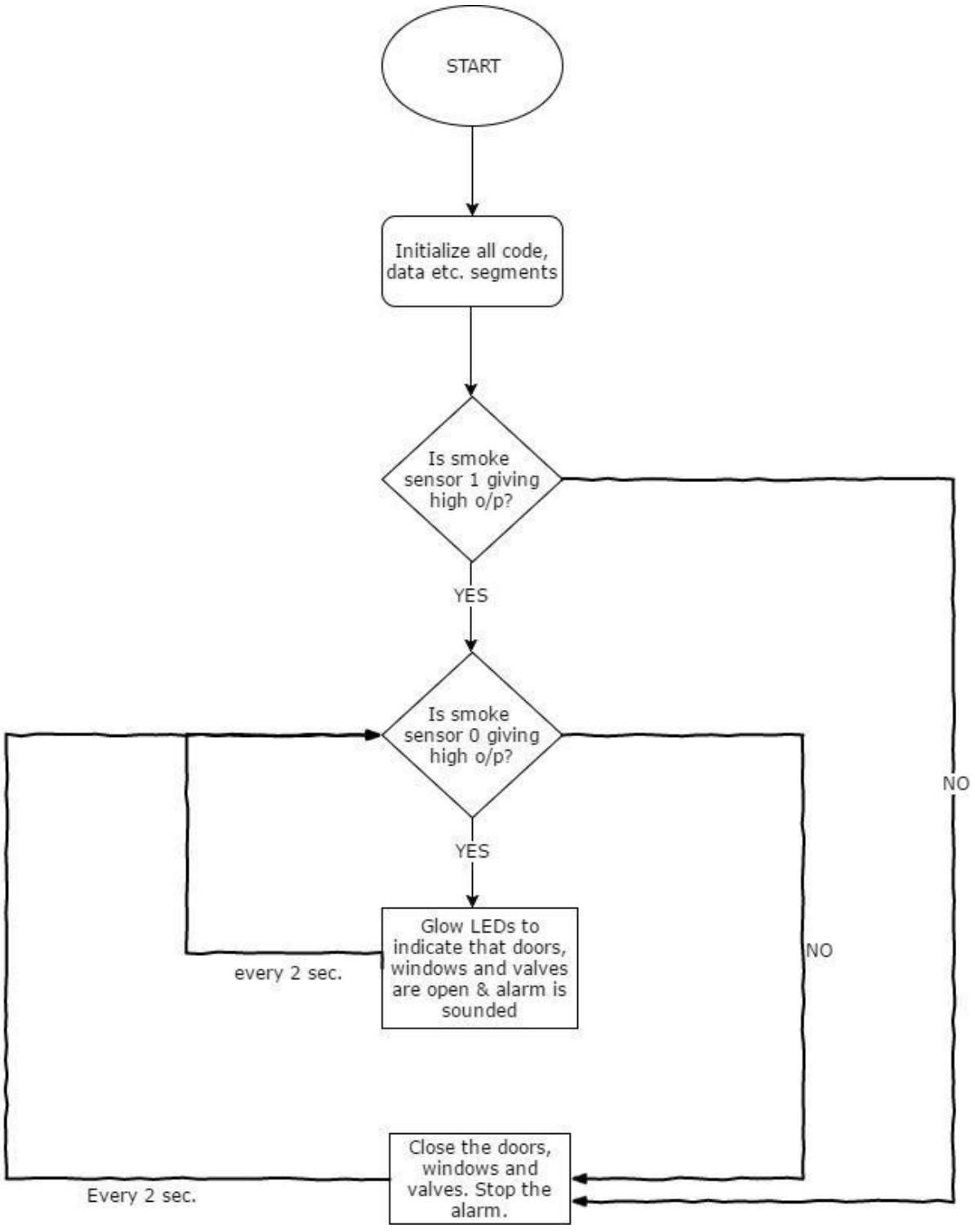
MQ-2 Smoke Sensor circuit with built in Analog to digital convertor (with logic 1 as output in case of presence of smoke else logic 0 is the output)

1 = Output
2 = Vcc (positive voltage)
3 = Gnd

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, data etc. segments
Is smoke sensor 1 giving high o/p?
YES
NO
Is smoke sensor 0 giving high o/p?
YES
NO
Glow LEDs to indicate that doors, windows and valves are open & alarm is sounded
every 2 sec.
Close the doors, windows and valves. Stop the alarm.
Every 2 sec.

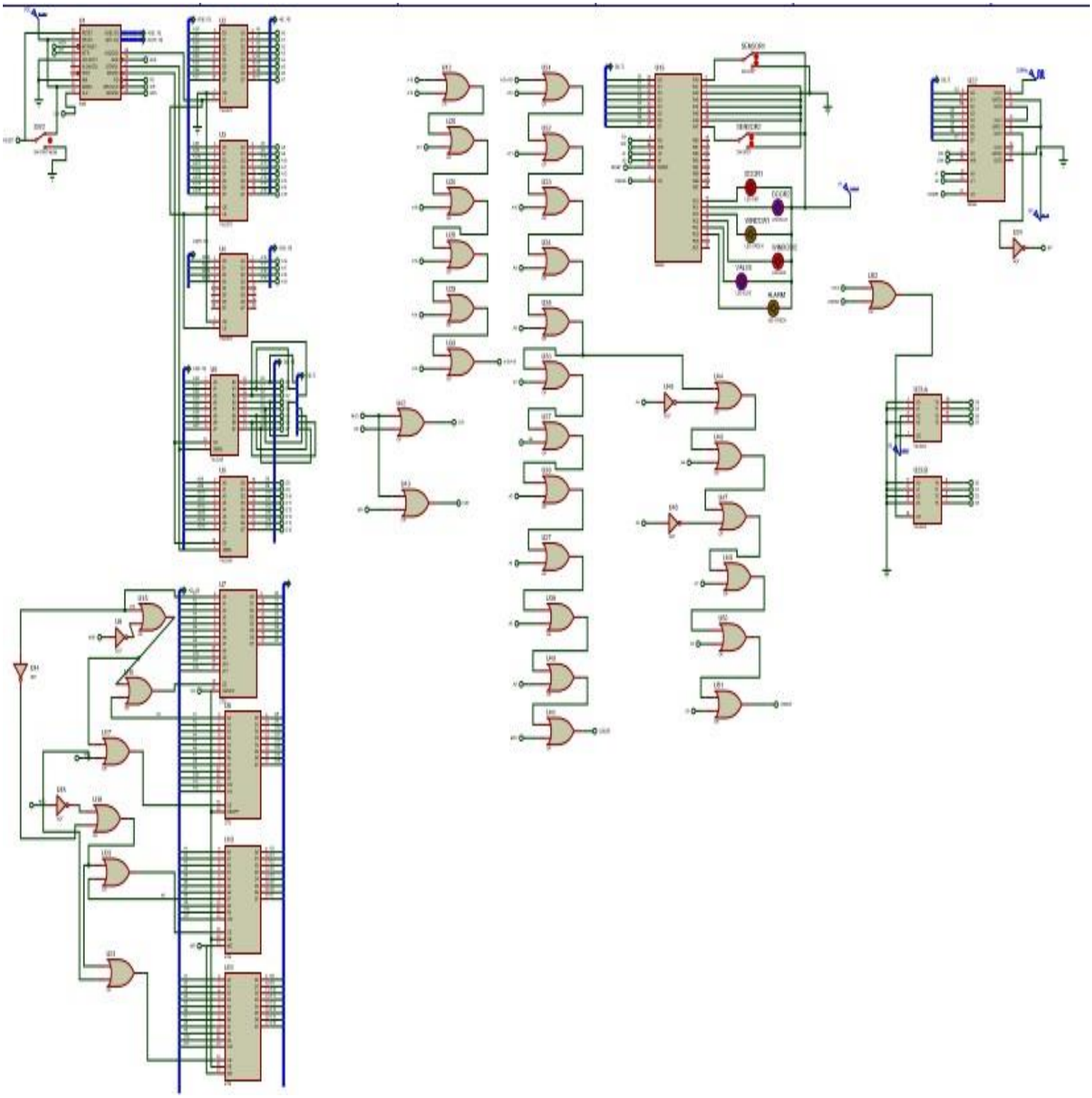
CODE

```
;The code is situated in the location ff000h

#make_bin#

#LOAD_SEGMENT=FFFFh#
#LOAD_OFFSET=0000h#

#CS=0000h#
#IP=0000h#

#DS=0000h#
#ES=0000h#

#SS=0000h#
#SP=0800sh#

#AX=0000h#
#BX=0000h#
#CX=0000h#
#DX=0000h#
#SI=0000h#
#DI=0000h#
#BP=0000h#

; add your code here
      jmp    st1
      db     1024 dup(0)
;main program

st1:
      ;clear interrupt flags
      cli
; intialize ds, es, ss to start of RAM
      mov    ax,0000h
      mov    ds,ax
      mov    es,ax
      mov    ss,ax
      mov    sp,0800H

;intialise porta & b as input & portc as output
      mov    al,92h ;10010010b
```

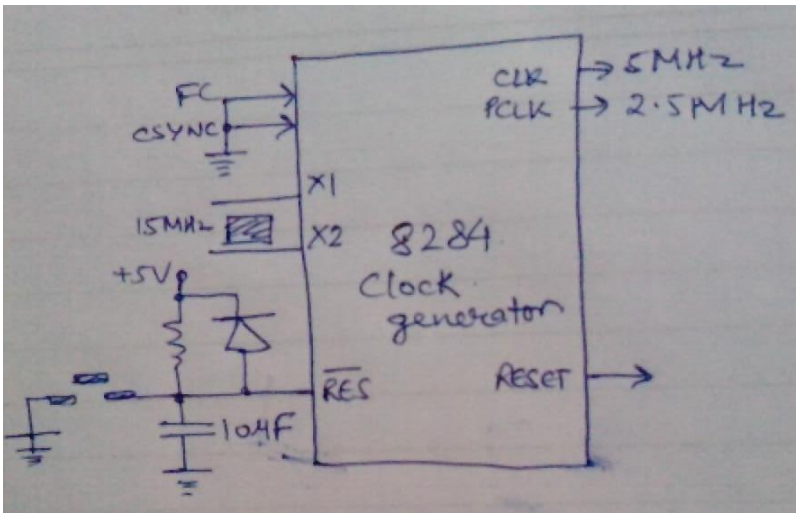
```
out    06h,al

;initialise 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
      jnz    off

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

      ;close valves, doors and windows
off:       mov       al,0FFh
      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 s switched off
        jmp     next

read proc near:
        iret
  read endp
```

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
- adc0808-n (for understanding the concept of ADC, not used in project though)
- 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