**Birla Institute of Technology and**

**Science, Pilani**



**A REPORT**

**ON**

**Smart Lighting System**

**(Group 4 – Problem 13)**

2018A7PS0112G Bhavyam Kamal

2018A7PS0686G Pranav Pateriya

2018A7PS0173G Nitish Silswal

2018A7PS0117G Aditya Mishra

2018A7PS0535G Saubhagya Shukla

**INDEX**

|  |  |  |
| --- | --- | --- |
| Serial Number | Title | Page Number |
| 1 | Problem Statement | 1 |
| 2 | Assumptions | 2 |
| 3 | Hardware Devices | 3 |
| 4 | Complete Address Mapping of Memory And I/O Devices | 4 |
| 5 | Flow Chart of The Software Code with Proper Comments | 5 |
| 6 | Code | 8 |
| 7 | Circuit Diagram | 21 |
| 8 | References | 22 |

**PROBLEM STATEMENT**

**Smart Lighting System**

Description: This is a lighting system for a conference room. As the seats get filled the light should be turned on. The rows are filled from row1 onwards. There are 4 lights per row. As each row begins to get filled the lights get turned on. As each rows empties completely the light gets turned off. You

can assume there are atleast 5 rows. Entry to the auditorium is restricted to a certain point of time. Exit can be at any point of time.

**System Details:**



**HARDWARE DEVICES**

|  |  |  |  |
| --- | --- | --- | --- |
| **CHIP NUMBER** | **CHIP** | **QUANTITY REQUIRED** | **USE** |
| **8086** | Microprocessor | 1 | Central Processing Unit |
| **6116** | RAM 2K | 2 | Random access memory which contains DS,SS |
| **2732** | ROM 4K | 2 | Read only memory which contains entire code (CS) |
| **74LS373** | 8 Bit Latch | 3 | To latch address bus |
| **74LS245** | 8 Bit Buffer | 2 | To buffer data bus (bidirectional) |
| **74LS138** | 3:8 Decoder | 1 | Used for select signals |
| **8255** | Programmable Peripheral Interface | 1 | Input and Output ports |
| **8284** | Clock Timer | 1 | For stable clock signal |
| **LED** | Common Cathode Configuration | 20 | For lighting |

**MAPPING**

**Memory Organization:**

The system uses 4KB of RAM and 8KB of ROM. RAM consists of two 2K chips and ROM

consists of 4K chips. They are organized into odd and even bank to facilitate both byte and word size data transfers.

**Read Only Memory (2732):** Starting Address: 00000h, Ending Address: 01FFFh

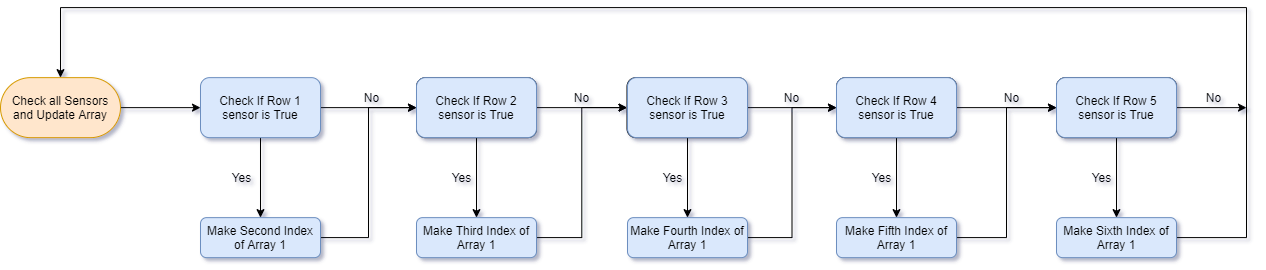
**Random Access Memory (6116):** Starting Address: 02000h, Ending Address: 02FFFh

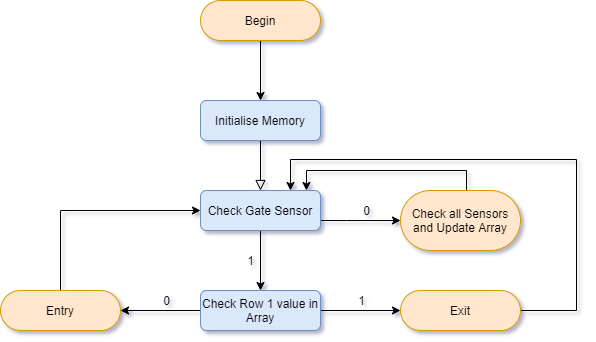


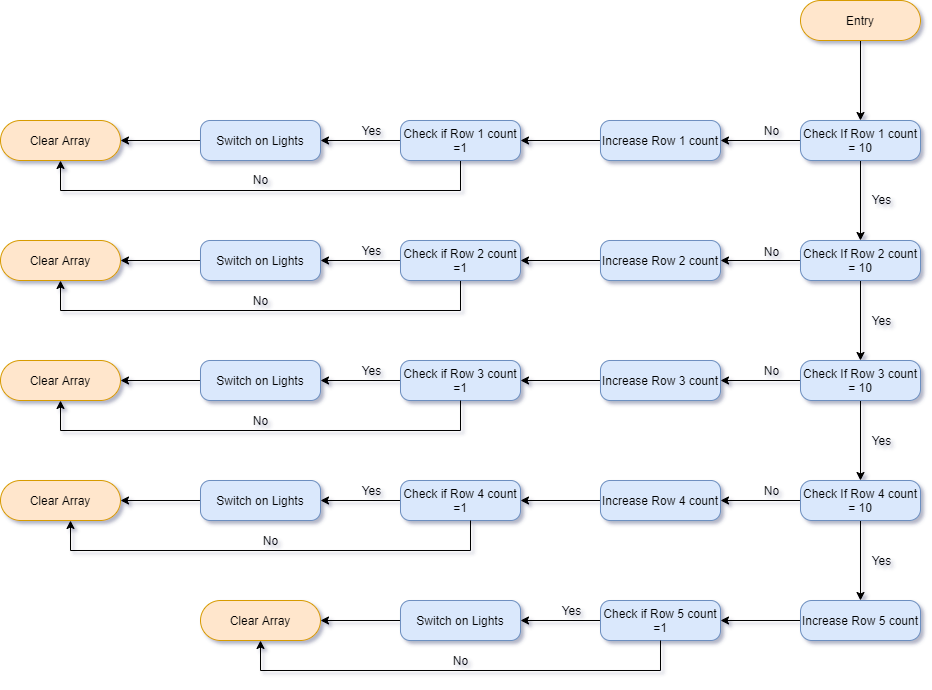
**I/O Mapping:**

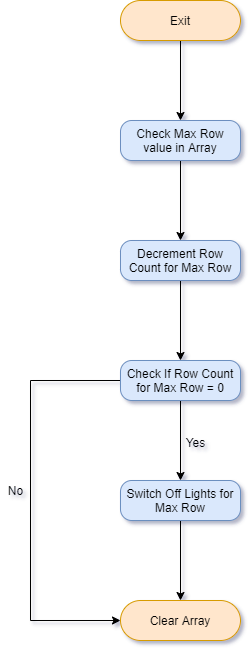
|  |  |  |
| --- | --- | --- |
| **8255-0 Port** | **Address** | **Usage** |
| **Port A** | 00000H | Input from Sensors |
| **Port B** | 00002H | Not Used |
| **Port C** | 00004H | Output to LEDs |
| **CWR** | 00006H | Control Register |

**FLOWCHART**

****





****

**CODE**

;MAIN PROGRAM

COUNTER DW 00H

RCOUNTER DB 00H,00H,00H,00H,00H,00H ;Maintains an array of which sensors have been activated till gate sensor is activated

LSTATUS DB 0 ;Maintains a status of lights

MAXROW DB 00H ;Max # of the row that was pressed during exit

SEATS DB 00H,00H,00H,00H,00H,00H ;Individual Row Count for all rows

;8255-0

PORTA0 EQU 00H

PORTB0 EQU 02H

PORTC0 EQU 04H

COMMAND\_ADDRESS0 EQU 06H

JMP ST1

DB 1001 DUP(0)

ST1:

; INTIALIZE DS, ES,SS TO START OF RAM

MOV AX,02000H

MOV DS,AX

MOV ES,AX

MOV SS,AX

MOV SP,02FFEH

;intialise porta as input & b& c as output

mov al,00110110b

out 0eh,al

mov al,4

out 08h,al

mov al,0

out 08h,al

mov al,90h

out 06h,al

MOV SEATS,00H

MOV SEATS+1,00H

MOV SEATS+2,00H

MOV SEATS+3,00H

MOV SEATS+4,00H

MOV RCOUNTER,00H

MOV RCOUNTER+1,00H

MOV RCOUNTER+2,00H

MOV RCOUNTER+3,00H

MOV RCOUNTER+4,00H

MOV RCOUNTER+5,00H

MOV LSTATUS,00H

MOV MAXROW,00H

;;CHECK FOR ENTRY THROUGH GATE

X1: IN AL,00H

AND AL,80H

CMP AL,80H

JNE X2

JMP X7 ; X7 is the sequence where gate emits code 1 aka it is interrupted

;; this is the code for the Check all sensors and update array part of code

;;CHECK FOR Sensor Interrupt IN ROW1

X2: IN AL,00H

AND AL,40H

CMP AL,40H

JNE X3

MOV CX, 0D000h

W3:

NOP

NOP

NOP

NOP

NOP

LOOP W3

ADD RCOUNTER+1,1

CMP RCOUNTER+1,0

JE X1

CMP RCOUNTER+2,0

JNE X1

MOV MAXROW,1

;;CHECK FOR Sensor Interrupt IN ROW2

X3: IN AL,00H

AND AL,20H

CMP AL,20H

JNE X4

MOV CX, 0D000h

W4:

NOP

NOP

NOP

NOP

NOP

LOOP W4

ADD RCOUNTER+2,1

CMP RCOUNTER+2,0

JE X1

CMP RCOUNTER+3,0

JNE X1

MOV MAXROW,2

;;CHECK FOR Sensor Interrupt IN ROW3

X4: IN AL,00H

AND AL,10H

CMP AL,10H

JNZ X5

MOV CX, 0D000h

W5:

NOP

NOP

NOP

NOP

NOP

LOOP W5

ADD RCOUNTER+3,1

CMP RCOUNTER+3,0

JE X1

CMP RCOUNTER+4,0

JNE X1

MOV MAXROW,3

;;CHECK FOR Sensor Interrupt IN ROW4

X5: IN AL,00H

AND AL,08H

CMP AL,08H

JNE X6

MOV CX, 0D000h

W6:

NOP

NOP

NOP

NOP

NOP

LOOP W6

ADD RCOUNTER+4,1

CMP RCOUNTER+4,0

JE X1

CMP RCOUNTER+5,0

JNE X1

MOV MAXROW,4

;;CHECK FOR Sensor Interrupt IN ROW5

X6: IN AL,00H

AND AL,04H

CMP AL,04H

JNE X1 ; X1 is the sequence that checks the gate

MOV CX, 0D000h

W7:

NOP

NOP

NOP

NOP

NOP

LOOP W7

ADD RCOUNTER+5,1

CMP RCOUNTER+5,0

JE X1

MOV MAXROW,5

JMP X1

;; Check row 1 array value

X7: MOV RCOUNTER,1

CMP RCOUNTER+1,1

JE Y1 ; Y1 is the sequence for exit

JMP Z1 ; Z1 is the sequence for entry

;; Entry Sequence

;; Check if Row1 count is 10

Z1: CMP SEATS,10

JNE Z2

;; Check if Row2 count is 10

Z3: CMP SEATS+1,10

JNE Z4

;; Check if Row3 count is 10

Z5: CMP SEATS+2,10

JNE Z6

;; Check if Row4 count is 10

Z7: CMP SEATS+3,10

JNE Z8

;; Increment Row 5

Z9: SUB RCOUNTER+1,1

SUB RCOUNTER+2,1

SUB RCOUNTER+3,1

SUB RCOUNTER+4,1

ADD SEATS+4,1

CMP SEATS+4,0

JLE C2 ;C2 is the sequence that clears the array RCOUNTER's gate value

MOV AL,LSTATUS ;Load current status of lights into al so they dont get changed

MOV BL,00001000b ;Make sure the light in 5th row is on by or with current status

OR AL,BL

OUT 04H, AL ;Output now condition to port C

MOV LSTATUS,AL ;Update current status of lights

JMP C2

Z2: SUB RCOUNTER+1,1

ADD SEATS,1

CMP SEATS,0

JLE C2 ;C2 is the sequence that clears the array RCOUNTER's gate value

MOV AL,LSTATUS ;Load current status of lights into al so they dont get changed

MOV BL,10000000b ;Make sure the light in 1st row is on by or with current status

OR AL,BL

OUT 04H, AL ;Output now condition to port C

MOV LSTATUS,AL ;Update current status of lights

JMP C2

Z4: SUB RCOUNTER+1,1

SUB RCOUNTER+2,1

ADD SEATS+1,1

CMP SEATS+1,0

JLE C2 ;C2 is the sequence that clears the array RCOUNTER's gate value

MOV AL,LSTATUS ;Load current status of lights into al so they dont get changed

MOV BL,01000000b ;Make sure the light in 2nd row is on by or with current status

OR AL,BL

OUT 04H, AL ;Output now condition to port C

MOV LSTATUS,AL ;Update current status of lights

JMP C2

Z6: SUB RCOUNTER+1,1

SUB RCOUNTER+2,1

SUB RCOUNTER+3,1

ADD SEATS+2,1

CMP SEATS+2,0

JLE C2 ;C2 is the sequence that clears the array RCOUNTER's gate value

MOV AL,LSTATUS ;Load current status of lights into al so they dont get changed

MOV BL,00100000b ;Make sure the light in 3rd row is on by or with current status

OR AL,BL

OUT 04H, AL ;Output now condition to port C

MOV LSTATUS,AL ;Update current status of lights

JMP C2

Z8: SUB RCOUNTER+1,1

SUB RCOUNTER+2,1

SUB RCOUNTER+3,1

SUB RCOUNTER+4,1

ADD SEATS+3,1

CMP SEATS+3,0

JLE C2 ;C2 is the sequence that clears the array RCOUNTER's gate value

MOV AL,LSTATUS ;Load current status of lights into al so they dont get changed

MOV BL,00010000b ;Make sure the light in 4th row is on by or with current status

OR AL,BL

OUT 04H, AL ;Output now condition to port C

MOV LSTATUS,AL ;Update current status of lights

JMP C2

;; Clear Array

C1: MOV RCOUNTER,0

MOV RCOUNTER+1,00h

MOV RCOUNTER+2,00h

MOV RCOUNTER+3,00h

MOV RCOUNTER+4,00h

MOV RCOUNTER+5,00h

MOV MAXROW,00h

MOV CX, 0D000h

W2:

NOP

NOP

NOP

NOP

NOP

LOOP W2

JMP X1

C2: MOV RCOUNTER,0

MOV MAXROW,00h

MOV CX, 0D000h

W1:

NOP

NOP

NOP

NOP

NOP

LOOP W1

JMP X1

;;Exit Sequence

;; Decrement the row count for max row value

Y1: CMP MAXROW,1 ;Check MaxRow Value

JNE Y2

SUB SEATS,1 ;Subtract Row Count of MaxRow

CMP SEATS,0 ;Check If the count has become 0

JNE C1

MOV AL,LSTATUS ;Load Current state of Lights in AL

MOV BL,01111111b ;conserve all values except row LEDs

AND AL,BL

OUT 04H,AL ;Output to port C

MOV LSTATUS,AL ;Update status of Lights

JMP C1

Y2: CMP MAXROW,2

JNE Y3

SUB SEATS+1,1

CMP SEATS+1,0

JNE C1

MOV AL,LSTATUS

MOV BL,10111111b

AND AL,BL

OUT 04H,AL

MOV LSTATUS,AL

JMP C1

Y3: CMP MAXROW,3

JNE Y4

SUB SEATS+2,1

CMP SEATS+2,0

JNE C1

MOV AL,LSTATUS

MOV BL,11011111b

AND AL,BL

OUT 04H,AL

MOV LSTATUS,AL

JMP C1

Y4: CMP MAXROW,4

JNE Y5

SUB SEATS+3,1

CMP SEATS+3,0

JNE C1

MOV AL,LSTATUS

MOV BL,11101111b

AND AL,BL

OUT 04H,AL

MOV LSTATUS,AL

JMP C1

Y5: SUB SEATS+4,1

CMP SEATS+4,0

JNE C1

MOV AL,LSTATUS

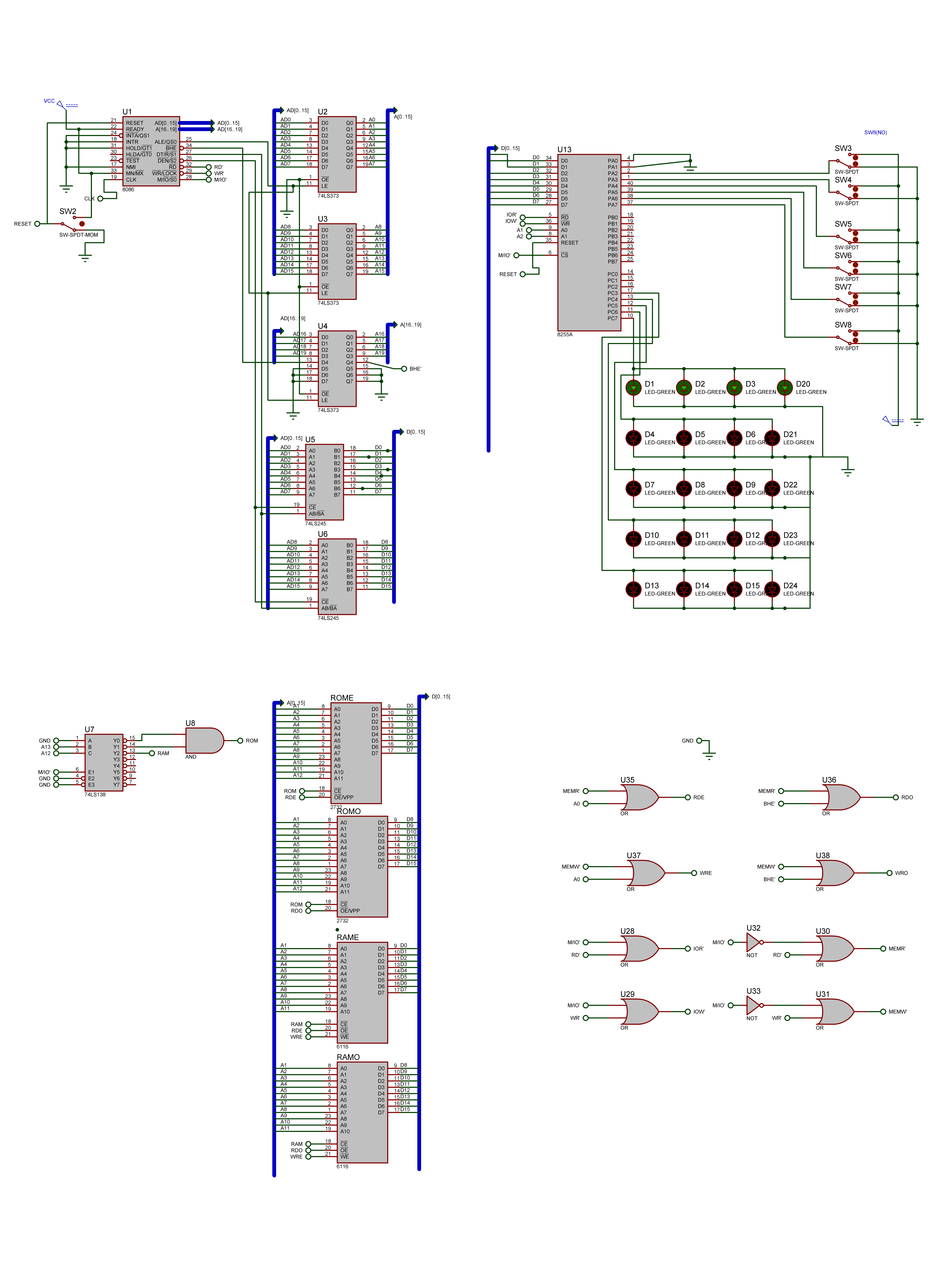
MOV BL,11110111b

AND AL,BL

OUT 04H,AL

MOV LSTATUS,AL

JMP C1

**CIRCUIT DIAGRAM**

**REFERENCES**

****

**PIR SENSOR**

The PIR (Passive Infra-Red) Sensor is a pyroelectric device that detects motion by sensing changes in the infrared (radiant heat) levels emitted by surrounding objects. This motion can be detected by checking for a sudden change in the surrounding IR pattern. When motion is detected the PIR sensor outputs a high signal on its output pin. This logic signal can be read by a microcontroller or used to drive an external load. PDF Documentation Attached