BITS Pilani
Pilani Campus

CS/EEE/INSTR F241

Microprocessor Programming & Interfacing

Design Assignment

Elevator Control

Submitted By:
Group No.25
Assignment No. 22
Mamarde Tanay Vijay
Harshvardhan Agrawal
Pranjal Dave
Rohan Dudeja

Problem Statement

System to be designed: Elevator control

System Requirements

o The elevator operates along 3 floors.
o When not in use the elevator is always on the ground floor.
o The elevator can be called by pressing any one of two buttons available on each floor.
o One button is up and the other is down.

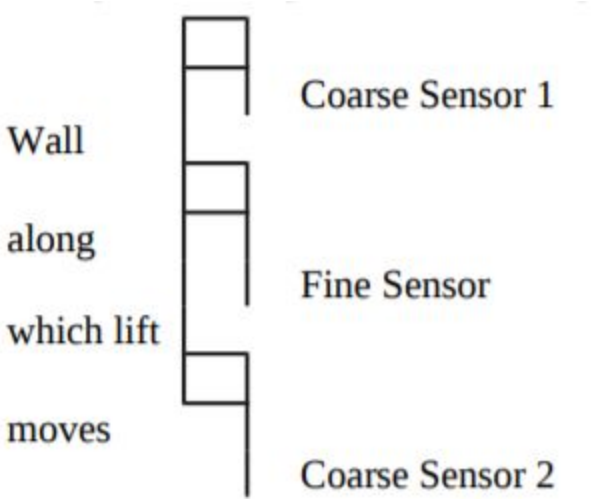
Whether the elevator stops at the floor or not depends on the direction in which the lift moves. For eg. if the lift is moving in upward direction and the person on say the 2nd floor presses the down button; the lift will not stop in the current journey. When the lift reaches the 3rd floor and starts moving down then the lift will stop at the 2nd floor.

o At every floor there is a 7-segement display that indicates the floor in which the lift is right now. The display can be any value from 0 - 3. '0' indicates the ground floor.
o Inside the lift - buttons are available for floor selection.
o The floor towards which the lift is moving is also displayed within the lift.
o Doors to the lift open and close automatically.
o When the lift reaches any floor where it has to stop it opens automatically, and it closes when a button called "Door Close" is pressed. Lift does not move until the door is closed.
o System runs from a standard power inlet.

System Specifications

o An Electro-magnetic system is used for open and close of the door. You just need to provide the on/off control.
o A heavy duty servo motor is used for lift movement. You just need to provide the input to the driver circuit.
o The inputs are direction (up/down) and a PWM input which control the speed at which the lift moves. The duty cycle can vary from 20% to 60%.

o The frequency of the PWM signal is 20 Hz.
o For detecting whether the lift has reached a floor, the system has a set of three sensors – two ‘coarse’ sensors and a ‘fine’ sensor. All the sensors are contact switches (i.e) when the lift reaches the point where the sensors are placed, the contact switch gets pushed in. Output of contact switches are low when closed and high otherwise. The sensor arrangement is represented in the fig below:



o On the ground floor – only Coarse Sensor1 and Fine Sensor will be available. On the 3rd floor only Coarse Sensor 2 and the Fine Sensor will be available.
o When the lift starts at the ground floor it starts at a low speed gradually accelerating to the maximum speed. It should operate at maximum speed when it reaches ‘Coarse Sensor 1”. As the lift moves up if it has to stop at floor ‘1’, when Coarse Sensor 2 is detected at that floor the lift starts moving at a low speed until it can stop when it reaches Fine sensor. When it starts again it moves at low speeds and reaches the maximum possible speed when it reaches the fine sensor. The same is done in the reverse direction with the appropriate sensors.
o Speed at which the lift moves is proportional to the duty cycle. For acceleration, duty cycle has to be gradually increased from 20 % to 60 %. And for deceleration, the duty cycle reduced from 60 % to 20 %. The increase is in steps of 20 %
o A 7447 chip (BCD to seven segment converter) is used for driving the 7-segment displays. o 7447 takes a 4-bit BCD value and converts into the corresponding 7-segment Equivalent.

Assumptions:

- Coarse and fine sensors have been assumed to be SPDT switches and have to be manually pressed during simulation.
- The keypad for selecting the floor in the elevator also consists of SPDT switches.
- Door closes if door close button is pressed or 5 seconds after opening.
- Door control is monitored by using an LED. LED is on for closed door and off for open door.
- The heavy duty motor for lift movement is modelled as an oscilloscope and an LED. LED indicates direction: on for up and off for down and PWM indicates speed.
- 20% duty cycle is full stop, 40% is slow speed, 60% is fast speed.
- Contact with all sensors of a floor is lost before contact with sensors of the next floor is made.

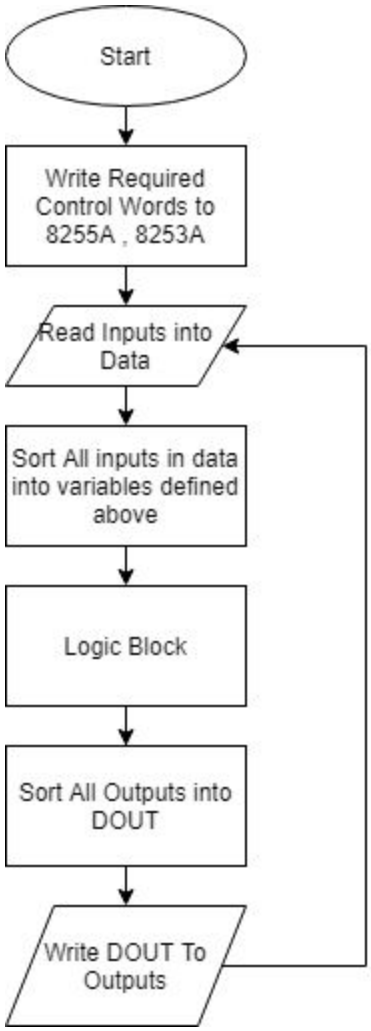
List of Components

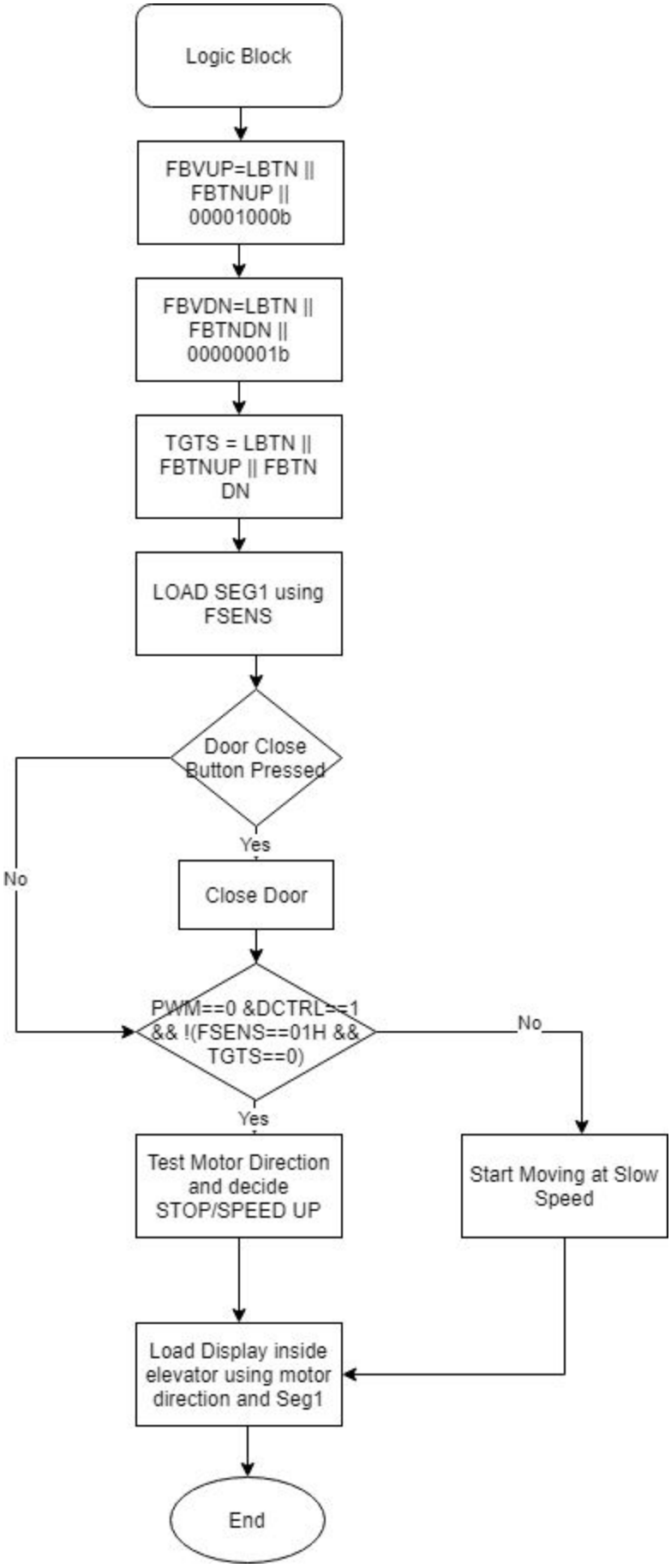
- INTEL 8086 Microprocessor –1
- Octal D-Type latch (74LS373) – 3
- Bidirectional Buffer (74LS245) – 2
- Unidirectional Buffer (74LS244) – 1
- INTEL 8255A Programmable Peripheral Interface chip – 2
- 2732 ROM 4KB – 2
- 6116 RAM 2KB – 2
- 8253A Programmable Interval Timer –1
- 7-Segment Display –5
- 7447 BCD to 7-segment display converter –2
- LEDs – 12
- 74LS138 3-8 Decoder – 1
- Interactive SPDT Switch (Momentary action) – 12
- Interactive SPDT Switch (Latched action) – 10
- 7407 Hex NOT gate – 1
- 7432 Quad 2 input OR Gate – 3
- 7427 Triple 3 input NOR Gate – 1

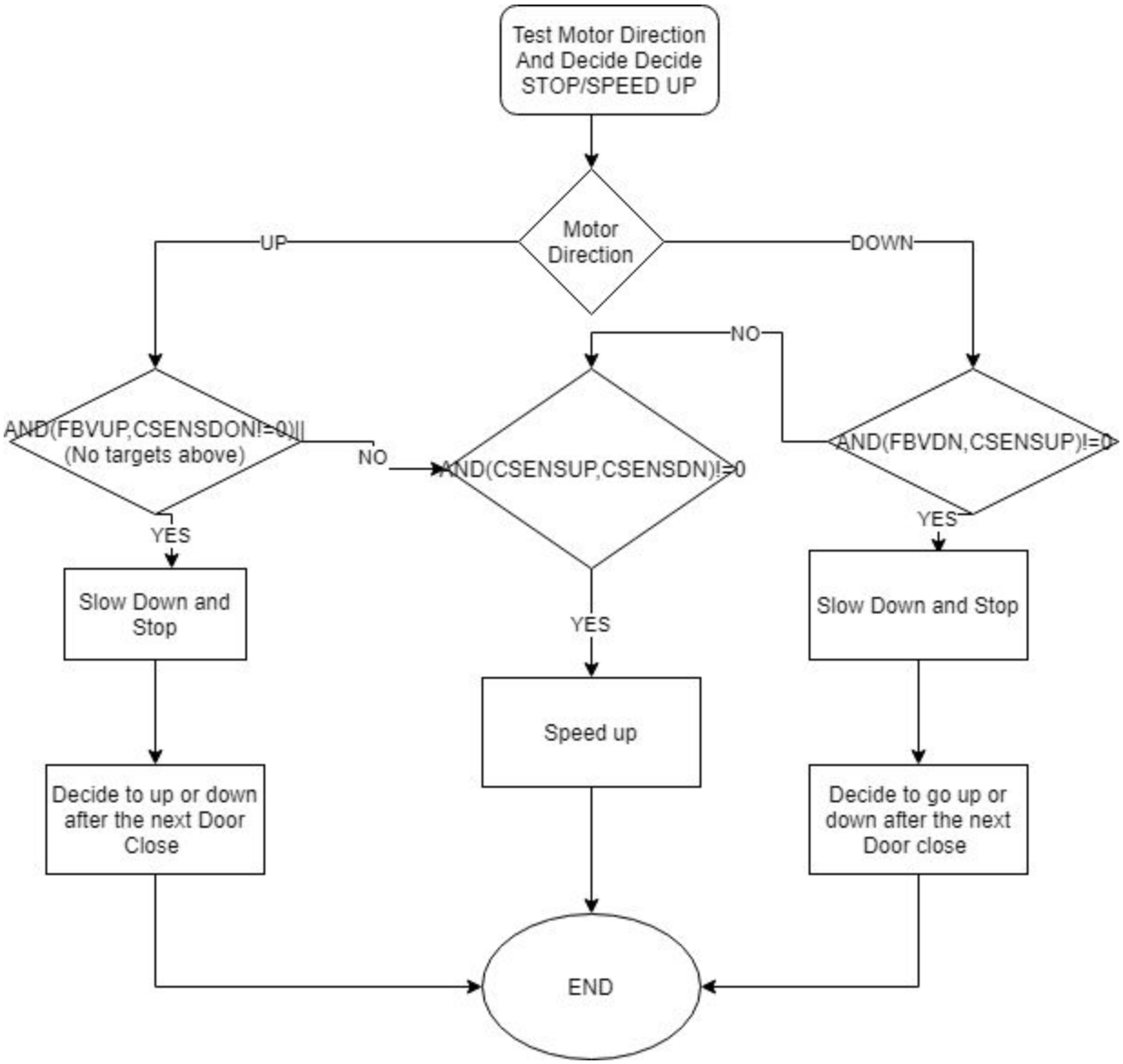
Address Mapping

- ROM : 0x00000 to 0x01FFF (8 kB distributed in even and odd banks)
- RAM : 0x02000 to 0x02FFF (4 kB distributed in even and odd banks)
- 8255A (for inputs): 0x03000, 0x03002, 0x03004, 0x03006
- 8255A (for outputs): 0x04000, 0x04002, 0x04004, 0x04006
- 8253A: 0x05000, 0x05002, 0x05004, 0x05006

Flowcharts


Start
Write Required Control Words to 8255A , 8253A
Read Inputs into Data
Sort All inputs in data into variables defined above
Logic Block
Sort All Outputs into DOUT
Write DOUT To Outputs


Logic Block
FBVUP=LBTN || FBTNUP || 00001000b
FBVDN=LBTN || FBTNDN || 00000001b
TGTS = LBTN || FBTNUP || FBTN DN
LOAD SEG1 using FSENS
Door Close Button Pressed
Yes
No
Close Door
PWM==0 &DCTRL==1 && !(FSENS==01H && TGTS==0)
No
Yes
Test Motor Direction and decide STOP/SPEED UP
Start Moving at Slow Speed
Load Display inside elevator using motor direction and Seg1
End


Test Motor Direction And Decide Decide STOP/SPEED UP
Motor Direction
UP
DOWN
NO
AND(FBVUP,CSENSDON!=0)|| (No targets above)
NO
AND(CSENSUP,CSENSDN)!=0
AND(FBVDN,CSENSUP)!=0
YES
YES
YES
Slow Down and Stop
Speed up
Slow Down and Stop
Decide to up or down after the next Door Close
Decide to go up or down after the next Door close
END

List of Variables

DATA Input to be read from 8255
DOUT Output to be written to 8255
LBTN Internal buttons for lift destinations
FBTNUP Up buttons for all floors
FBTNDN Down buttons for all floors
FSENS Fine sensors
CSENSUP Upper coarse sensors
CSENSDN Lower coarse sensors
DBTN Door close button
FBVUP Floors to stop at while moving up
FBVDN Floors to stop at while moving down
TGTS Floors to visit
MOTOR Motor Direction
DCTRL Door close/open control
PWM PWM pulse to be set (0 – 20%/1 – 40%/2 – 60%)
PWM0 Current PWM pulse
SEG1 Value to load in display on floors
SEG2 Value to load in display in elevator

Code

```
#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#

;DATA
    JMP   START
    DB    1021 DUP(0)

;initiliazing port adresses

        PORTINA         EQU 3000H
            PORTINB             EQU 3002H
            PORTINC             EQU 3004H
            CREGIN      EQU 3006H

            PORTOUTA    EQU 4000H
            PORTOUTB    EQU 4002H
            PORTOUTC    EQU 4004H
            CREGOUT             EQU     4006H
```

```
;initializing clock addresses
                CLKA            EQU 5000H
                CLKB            EQU 5002H
                CLKC            EQU 5004H
                CREGCLK         EQU 5006H

                DATA            DB 3 DUP(0FFH)
                DOUT            DB 3 DUP(00H)

                LBTN            DB 00H
                FBTNUP          DB 00H
                FBTNDN          DB 00H
                FSENS           DB 00H
                CSENSUP                 DB 00H
                CSENSDN                 DB 00H
                DBTN            DB 00H
                DSENS           DB 00H

                FBVUP           DB 00H
                FBVDN           DB 00H
                TGTS            DB 00H

                MOTOR           DB 00H
                DCTRL           DB 00H
                PWM                     DB 00H
                PWM0            DB 00H
                SEG1            DB 00H
                SEG2            DB 00H

;CODE
START:  MOV AX,0200H
        MOV DS,AX
        MOV ES,AX
        MOV SS,AX
        MOV SP,0FFEH

                ;PORT INITIALIZE
                MOV AL, 9BH
```

```
                MOV DX, CREGIN
                OUT DX, AL
                MOV AL, 80H
                MOV DX, CREGOUT
                OUT DX, AL

                ;PWM INITIALIZE
                MOV AL, 00010100B
                MOV DX, CREGCLK
                OUT DX, AL
                MOV DX, CLKA
                MOV AL, 04H
                OUT DX, AL
                MOV AL, 01010010B
                MOV DX, CREGCLK
                OUT DX, AL
                MOV DX, CLKB
                MOV AL, PWM
                INC AL
                OUT DX, AL
                MOV AL, 10011000B
                MOV DX, CREGCLK
                OUT DX, AL

;LOOP
X1: ;INPUT
                MOV DX, PORTINA
                IN      AL, DX
                LEA BX, DATA
                MOV [BX], AL
                INC BX
                MOV DX, PORTINB
                IN      AL, DX
                MOV [BX], AL
                INC BX
                MOV DX, PORTINC
                IN      AL, DX
                MOV [BX], AL
```

```
;SORT OUT THE INPUTS

        ;LOAD DATA + 0
        LEA SI, DATA
        MOV AL, [SI]
        NOT AL

        ;DSENS
        MOV BL, 10000000B
        AND BL, AL
        MOV CL, 1
        ROL BL, CL
        LEA DI, DSENS
        MOV [DI], BL

        ;DBTN
        MOV BL, 01000000B
        AND BL, AL
        MOV CL, 2
        ROL BL, CL
        LEA DI, DBTN
        MOV [DI], BL

        ;(CSENSDN)lower Coarse sensor ofeach floor are stored
        MOV BL, 00111000B
        AND BL, AL
        MOV CL, 2
        ROR BL, CL
        LEA DI, CSENSDN
        MOV [DI], BL

        ;(CSENSUP)upper Coarse sensor ofeach floor are stored
        MOV BL, 00000111B
        AND BL, AL
        LEA DI, CSENSUP
        MOV [DI], BL

        ;Contents of PortInB are loaded
        INC SI
```

```
MOV AL, [SI]
NOT AL

;(LBTN)Buttons inside the lift are stored
MOV BL, 11110000B
AND BL, AL
MOV CL, 4
ROR BL, CL
LEA DI, LBTN
MOV DL, [DI]
OR      BL, DL
MOV [DI], BL

;(FSeNS)Data from fine sensors on each floor is stored
MOV BL, 00001111B
AND BL, AL
LEA DI, FSENS
MOV [DI], BL

;Contents of PortInC are Loaded
INC SI
MOV AL, [SI]
NOT AL

;(FBTNDN)Data from buttons on Floors to Lift To go down is stored
MOV BL, 00111000B
AND BL, AL
MOV CL, 2
ROR BL, CL
LEA DI, FBTNDN
MOV DL, [DI]
OR      BL, DL
MOV [DI], BL

;(FBTNUP)Data from buttons on Floors to Lift To go up is stored
MOV BL, 00000111B
AND BL, AL
LEA DI, FBTNUP
MOV DL, [DI]
```

```
        OR      BL, DL
        MOV DL, [DI]
        OR      BL, DL
        MOV [DI], BL

;LOGIC

        ;FBVUP
        LEA SI, LBTN
        MOV AL, [SI]
        LEA SI, FBTNUP
        MOV AH, [SI]
        OR      AL, AH
        OR      AL, 00001000B
        LEA DI, FBVUP
        MOV [DI], AL

        ;FBVDN
        LEA SI, LBTN
        MOV AL, [SI]
        LEA SI, FBTNDN
        MOV AH, [SI]
        OR      AL, AH
        OR      AL, 00000001B
        LEA DI, FBVDN
        MOV [DI], AL

        ;TGTS(stores the floors where lift has to go)
        MOV BL, TGTS;contains value of TGTS
        LEA SI, LBTN
        MOV AL, [SI]
        LEA SI, FBTNUP
        MOV AH, [SI]
        OR      AL, AH
        LEA SI, FBTNDN
        MOV AH, [SI]
        OR      AL, AH
        LEA DI, TGTS
        MOV [DI], AL;value of TGTS is updated
```

```
        CMP BL, 0
        JNZ SLEEP;if lift is already in use
        CMP AL, 0
        JZ  SLEEP;if lift was not in use before and is not in use now
        MOV AL, FSENS
        CMP AL, 01H
        JNZ SLEEP;if lift is not present on ground floor and is in use
        MOV DX, CLKC
        MOV AL, 32H
        OUT DX, AL

SLEEP:

        ;SEG1(Segment Display 1(Current Position of lift))
        LEA DI, SEG1
        MOV AL, FSENS
        CMP AL, 00H
        JZ DEND;if lift is moving between floors
        CMP AL, 01H
        JZ D0
        CMP AL, 02H
        JZ D1
        CMP AL, 04H
        JZ D2
        CMP AL, 08H
        JZ D3
        D0:     MOV BL, 0;current location is floor 0
                MOV [DI], BL
                JMP DEND
        D1:     MOV BL, 1;current location is floor 1
                MOV [DI], BL
                JMP DEND
        D2:     MOV BL, 2;current location is floor 2
                MOV [DI], BL
                JMP DEND
        D3:     MOV BL, 3;current location is floor 3
                MOV [DI], BL
                JMP DEND
        DEND:
```

```
;DCTRL(Direction of Lift )
LEA DI, DCTRL
MOV AL, [DI]
MOV AH, DBTN
OR      AL, AH
MOV [DI], AL

;STARTUP DECIDE
MOV AL, PWM
MOV AH, 0
CMP AL, AH
JNZ NOSTART
MOV AL, DSENS
MOV AH, 0
CMP AL, AH
JZ NOSTART
MOV AL, DCTRL
MOV AH, 0
CMP AL, AH
JZ NOSTART
MOV AL, FSENS
MOV AH, 01H
CMP AL, AH
JNZ DOSTART
MOV AL, TGTS
MOV AH, 00H
CMP AL, AH
JNZ DOSTART
JMP NOSTART
DOSTART:        CALL DIRECT
                CALL SLOW
                JMP AEND
NOSTART:

;TEST MOTOR DIRECTION AND DECIDE STOP/SPEED UP
;JUST STARTING?

LEA SI, MOTOR
```

```
MOV AL, [SI]
MOV AH, 0
CMP AH, AL
JNZ A1
JZ  A2

A1:     ;STOP?
        MOV AL, FBVUP
        MOV AH, CSENSDN
        AND AL, AH
        MOV AH, 0
        CMP AL, AH
        JZ A3
        A3X:
        CALL STOP
        JMP AEND
        A3:             ;NO TARGETS ABOVE?
        MOV AL, CSENSDN
        MOV CL, 1
        ROL AL, CL
        DEC AL
        MOV AH, TGTS
        CMP AH, AL
        JLE A3X
        ;SPEED UP?
        MOV AL, CSENSUP
        MOV AH, CSENSDN
        MOV BL, FSENS
        AND BL, 00000001B
        OR      AH, BL
        AND AL, AH
        MOV AH, 0
        CMP AL, AH
        JNZ A5
        CALL FAST
        A5:
        JMP AEND

A2:     ;STOP?
```

```
        MOV AL, FBVDN
        MOV AH, CSENSUP
        AND AL, AH
        MOV AH, 0
        CMP AL, AH
        JZ A4
        CALL STOP
        JMP AEND
        A4:
        ;SPEED UP?
        MOV AL, CSENSDN
        MOV AH, CSENSUP
        MOV BL, FSENS
        AND BL, 00001000B
        OR      AH, BL
        AND AL, AH
        MOV AH, 0
        CMP AL, AH
        JNZ A6
        CALL FAST
        A6:
        JMP AEND
AEND:

;SEG2
LEA SI, MOTOR
LEA DI, SEG2
MOV BL, SEG1
MOV AL, [SI]
MOV AH, 0
CMP AH, AL
JNZ S1
JZ  S2
S1:     CMP BL, 3
        JZ S3
        INC BL
        S3:
        JMP SEND
S2:     CMP BL, 0
```

```
                JZ S4
                DEC BL
                S4:
                JMP SEND
        SEND:
        MOV [DI], BL

;SORT OUT THE OUTPUTS

        ;LOAD DOUT + 0
        LEA DI, DOUT
        MOV AL, FBTNUP
        MOV BL, FBTNDN
        MOV CL, 2
        ROL BL, CL
        OR      AL, BL
        MOV BL, DCTRL
        MOV CL, 2
        ROR BL, CL
        OR      AL, BL
        MOV BL, MOTOR
        MOV CL, 1
        ROR BL, CL
        OR      AL, BL
        MOV [DI], AL

        ;LOAD DOUT + 1
        INC DI
        MOV AL, LBTN
        MOV BL, SEG1
        MOV CL, 4
        ROL BL, CL
        OR      AL, BL
        MOV BL, SEG2
        MOV CL, 2
        ROR BL, CL
        OR      AL, BL
        MOV [DI], AL
```

```
;OUTPUT
        LEA BX, DOUT
        MOV AL, [BX]
        MOV DX, PORTOUTA
        OUT DX, AL
        INC BX
        MOV AL, [BX]
        MOV DX, PORTOUTB
        OUT DX, AL
        INC BX
        MOV AL, [BX]
        MOV DX, PORTOUTC
        OUT DX, AL
        JMP X1

;SUBROUTINES
        STOP    PROC NEAR
                MOV AL, CSENSUP
                MOV BL, FSENS
                AND BL, 00001000B
                OR  AL, BL
                MOV AH, CSENSDN
                MOV BL, FSENS
                AND BL, 00000001B
                OR  AH, BL
                AND AL, AH
                MOV AH, 0
                CMP AL, AH
                JNZ B1
                JZ B2
                B1:     CALL FSTOP
                        JMP BEND
                B2:     CALL SLOW
                        JMP BEND
                BEND:
                RET
                ENDP

        SLOW    PROC NEAR
```

```
        LEA DI, PWM
        MOV AL, 1
        MOV [DI], AL
        CALL PWMSET
        RET
        ENDP

FAST    PROC NEAR
        LEA DI, PWM
        MOV AL, 2
        MOV [DI], AL
        CALL PWMSET
        RET
        ENDP

FSTOP   PROC NEAR
        MOV AL, PWM
        CMP AL, 0
        JZ      NOTIMER
        MOV DX, CLKC
        MOV AL, 32H
        OUT DX, AL
        NOTIMER:
        LEA DI, PWM
        MOV AL, 0
        MOV [DI], AL
        CALL PWMSET
        LEA DI, DCTRL
        MOV AL, 0
        MOV [DI], AL
        MOV AH, FSENS
        NOT AH
        LEA DI, FBTNUP
        MOV AL, [DI]
        AND AL, AH
        MOV [DI], AL
        LEA DI, FBTNDN
        MOV AL, [DI]
        AND AL, AH
```

```
                MOV [DI], AL
                LEA DI, LBTN
                MOV AL, [DI]
                AND AL, AH
                MOV [DI], AL
                RET
                ENDP

DIRECT  PROC NEAR
                MOV AL, FSENS
                MOV AH,01H
                CMP AL, AH
                JNZ C1
                LEA DI, MOTOR
                MOV BL, 01H
                MOV [DI], BL
                C1:
                MOV AL, FSENS
                MOV AH,08H
                CMP AL, AH
                JNZ C2
                LEA DI, MOTOR
                MOV BL, 00H
                MOV [DI], BL
                C2:
                MOV AL, FSENS
                MOV AH, TGTS
                CMP AH, AL
                JG C3
                LEA DI, MOTOR
                MOV BL, 00H
                MOV [DI],BL
                C3:
                RET
                ENDP

PWMSET          PROC NEAR
                MOV AL, PWM0
                MOV AH, PWM
```

```
CMP AL, AH
JZ ENDPWM
LEA DI, PWM0
MOV [DI], AH
MOV DX, CLKB
MOV AL, PWM
INC AL
OUT DX, AL
ENDPWM:
RET
ENDP
```

Circuit Diagram