Microprocessors, Programming and Interfacing

ELEVATOR CONTROL SYSTEM

Submitted by: Group No.: 74

Group Members

Chetan Gupta	2018A7PS0225G
Kunjir Rohan Sharad	2018A8PS0511G
Manas Agarwal	2018A8PS0470G
Wahib Sabir Kapdi	2018A3PS0247G
Aakash Jagdish Khosla	2018A8PS0813G

Date:19/4/2020

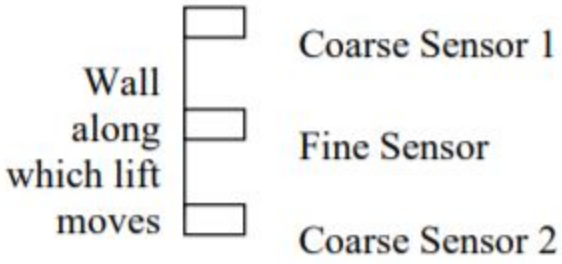
Contents:

System Requirements:

- The elevator operates along 4 floors.
- When not in use the elevator is always on the ground floor.
- The elevator can be called by pressing any one of two buttons available on each floor.
- 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.
- At every floor there is a 7-segment 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.
- Inside the lift - buttons are available for floor selection.
- The floor towards which the lift is moving is also displayed within the lift.
- Doors to the lift open and close automatically.
- 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.
- System runs from a standard power inlet.

System Specifications:

- An Electro-magnetic system is used for open and close of the door. You just need to provide the on/off control. A heavy duty servo motor is used for lift movement. You just need to provide the input to the driver circuit.
- 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 50%.
- The frequency of the PWM signal is 10 Hz.
- 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



- On the ground floor – only Coarse Sensor q and Fine Sensor will be available. On the 3rd floor only Coarse Sensor 2 and the Fine Sensor will be available.
- 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 Coarse sensor 2. The same is done in the reverse direction with the appropriate sensors.

- Speed at which the lift moves is proportional to the duty cycle. For acceleration, duty cycle has to be gradually increased from 20 % to 50 %. And for deceleration, the duty cycle reduced from 50 % to 20 %. The increase is in steps of 10 %.

Assumptions with Justifications:

- 3 sensors –Coarse 1,Course 2,Fine -are assumed to be protruding out of the wall and get pushed in when the lift moves past them.Hence, they are modeled as push buttons. (1 each per floor)
- System runs at a speed of 2.5 MHz.
- The door closes as soon as the door close button is pressed or automatically when the lift is called at a different floor.
- Two buttons are not pressed at the exact same moment.

Hardware Components:

- INTEL 8086 Microprocessor – 1
- 8255 Programmable Peripheral Interface – 2
- 8253 Programmable Interval Timer – 1
- 7-Segment Display – 5
- 7447 BCD to 7-segment display convertor – 1
- Servo motor – 1
- L293D Motor Driver – 1
- 74LS373 Octal Latch – 3
- 74LS245 Bidirectional Buffer – 2
- 2732 ROM 4KB – 2
- 6116 RAM 2KB – 2
- 8284 Clock Generator – 2
- 74LS138 3-8 Decoder – 1
- 21 Push Buttons
- 7404 Hex Converter – 1
- 7432 2 input or gate – 2
- 10K pull up registers - 8

Justification of Components Used:

8253

The 8253 is used for driving the servo motor.

The two clocks are used for the pwm outputs to the motor driver.

Using the given clock of 100 Hz, pwm output of 10Hz is produced.

Initializing 8253

```
mov al,00010110b      ;square wave generator with mode3
out cr_8253,al
mov al,0Ah
out cnt0,al

mov al,01010010b      ;mode1 to adjust duty cycle
out cr_8253,al
mov al,0Ah
out cnt1,ah
```

DUTY CYCLE	COUNT
0%	0Ah
20%	08h
30%	07h
40%	06h
50%	05h

Initializing 8255

```
mov al,10000010b
out 06h,al
mov al,10000000b
out 16h,al
```

Interface to Push Buttons

- The 14 push buttons are connected to Port A and Port B of 8255.
- PA0-PA3 are row inputs while PB0-PB3 are column inputs.
- They are connected in a manner equivalent to the 4X4 matrix.
- So, constant polling keeps occurring, to check for the key presses, and using the table provided, the key press is decoded.

Table on the Next Page

R3	R2	R1	R0	C3	C2	C1	C0	Hex	Button
0	1	1	1	0	1	1	1	77	Cs_1
0	1	1	1	1	0	1	1	7B	Fine
0	1	1	1	1	1	0	1	7D	Cs_2
0	1	1	1	1	1	1	0	7E	Ground_up
1	0	1	1	1	0	1	1	BB	First_up
1	0	1	1	1	1	0	1	BD	First_in
1	0	1	1	1	1	1	0	BE	First_down
1	1	0	1	0	1	1	1	D7	2_UP
1	1	0	1	1	0	1	1	DB	2_IN
1	1	0	1	1	1	0	1	DD	2_DOWN
1	1	0	1	1	1	1	0	DE	3_IN
1	1	1	0	0	1	1	1	E7	3_DOWN
1	1	1	0	1	0	1	1	EB	CLOSE
1	1	1	0	1	1	0	1	ED	Ground_in

7 Segment Display:

- 5-Seven Segment display having the same output.
- 1-7447
- Requires 4-bit

Keyboard Layout:

	C0	C1	C2	C3
R0	X	GROUND_in	CLOSE	3_DOWN
R1	3_IN	2_DOWN	2_IN	2_UP
R2	1_DOWN	1_IN	1_UP	X
R3	GROUND_UP	FINE	CS_2	CS_1

Address Mapping

I/O Mapping:

Addresses:

- 8255_1: 00h-06h
- 8253: 08h-0Eh
- 8255_2: 10h-16h

Memory Mapping:

Addresses:

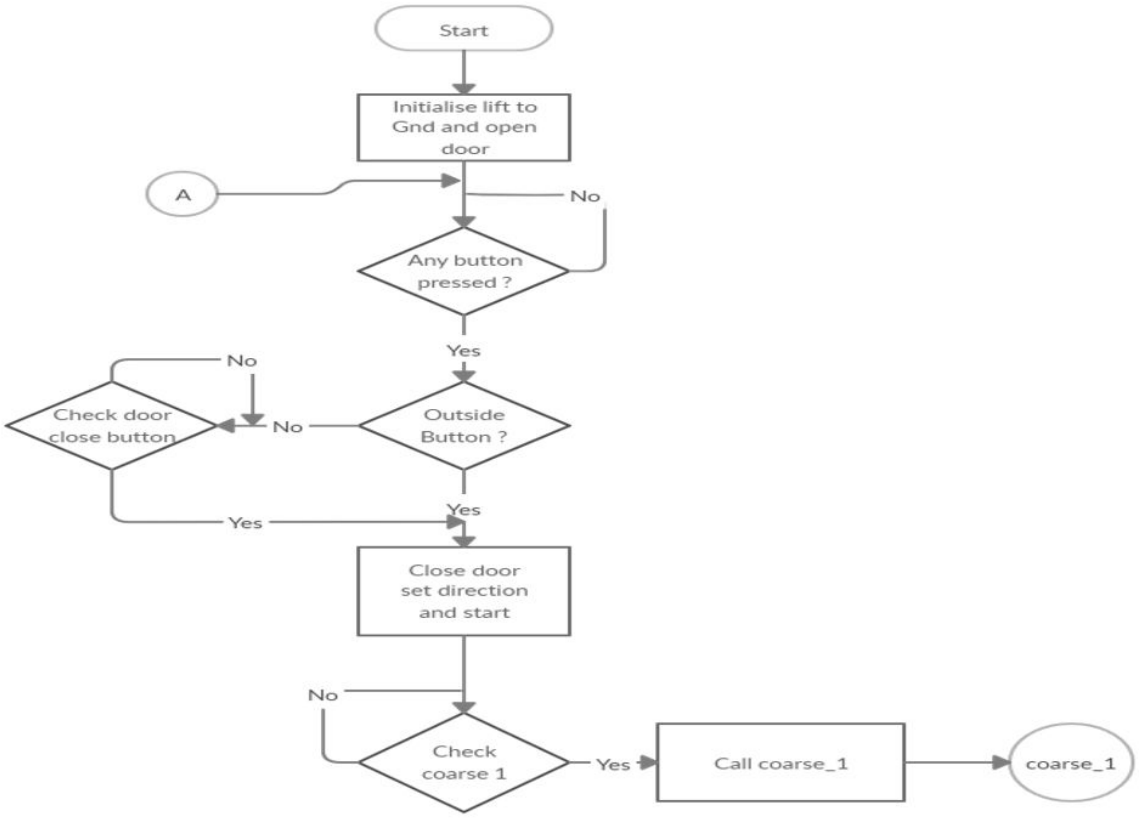
- 2 nos. - 2732 ROM: 00000h - 01FFFh
- 2 nos. - 6116 ROM: 02000h - 02FFFh

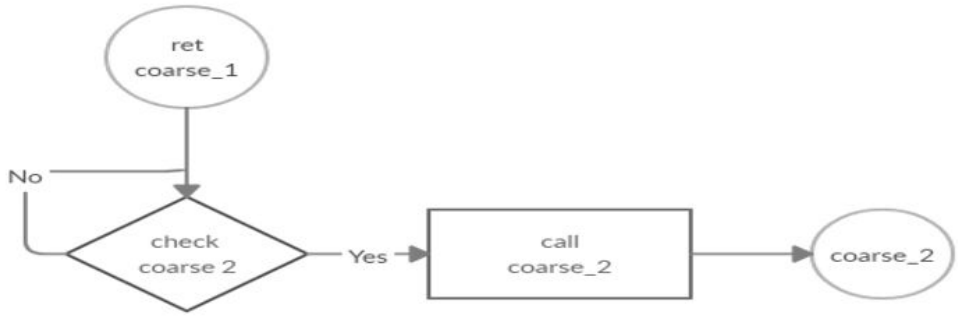
Design

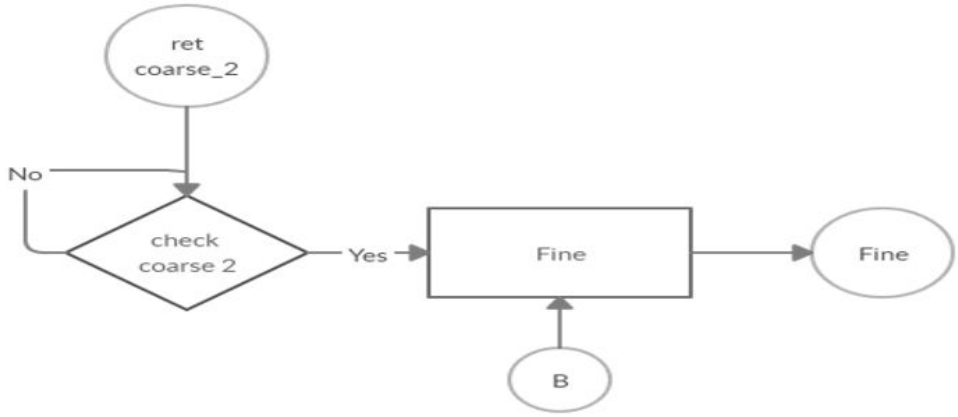
Complete design shown with proper labelling (design attached)

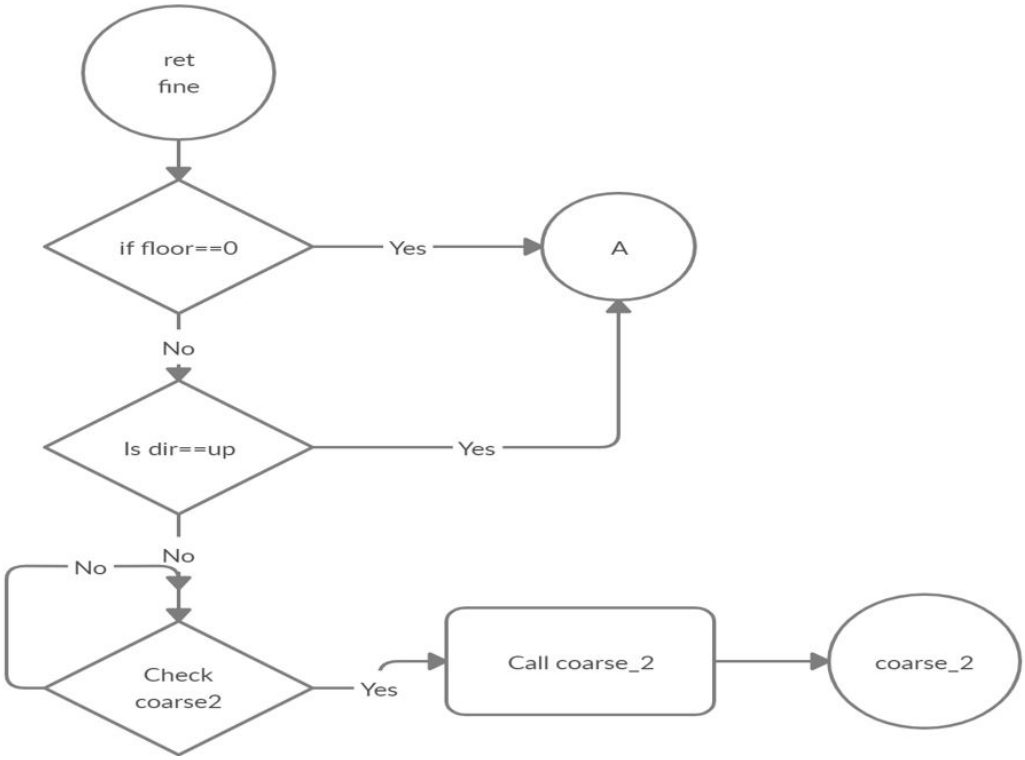
Flowchart

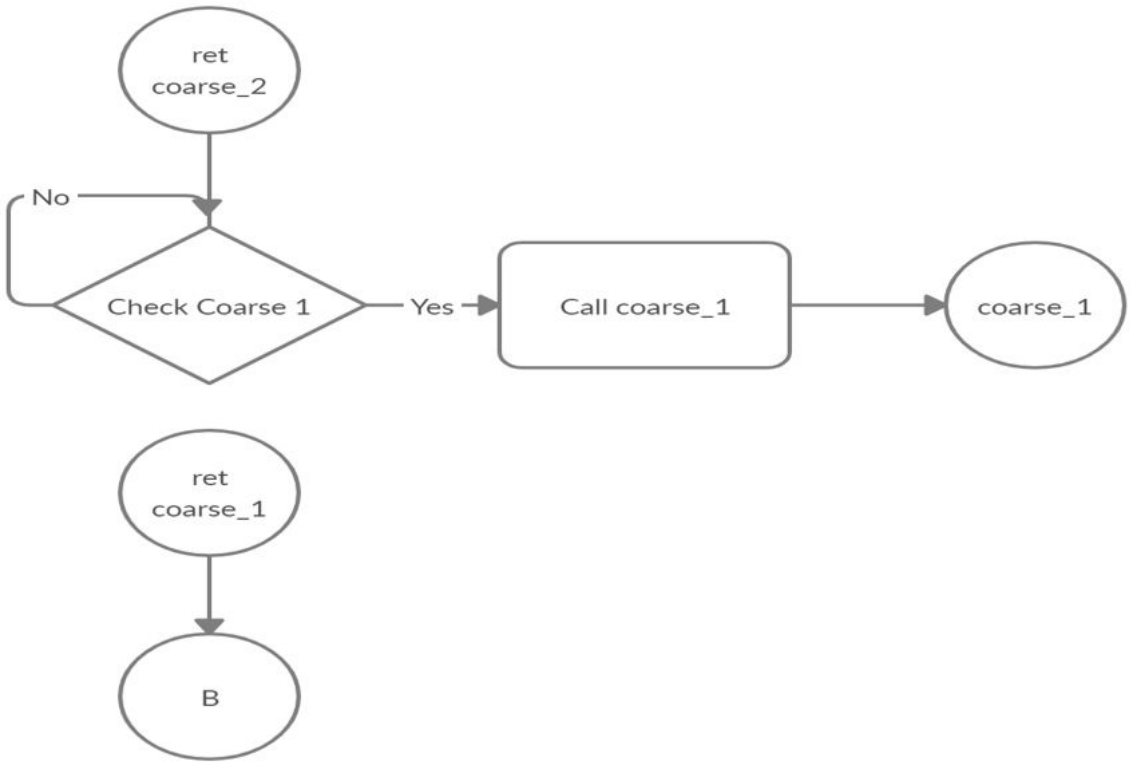
Main Program

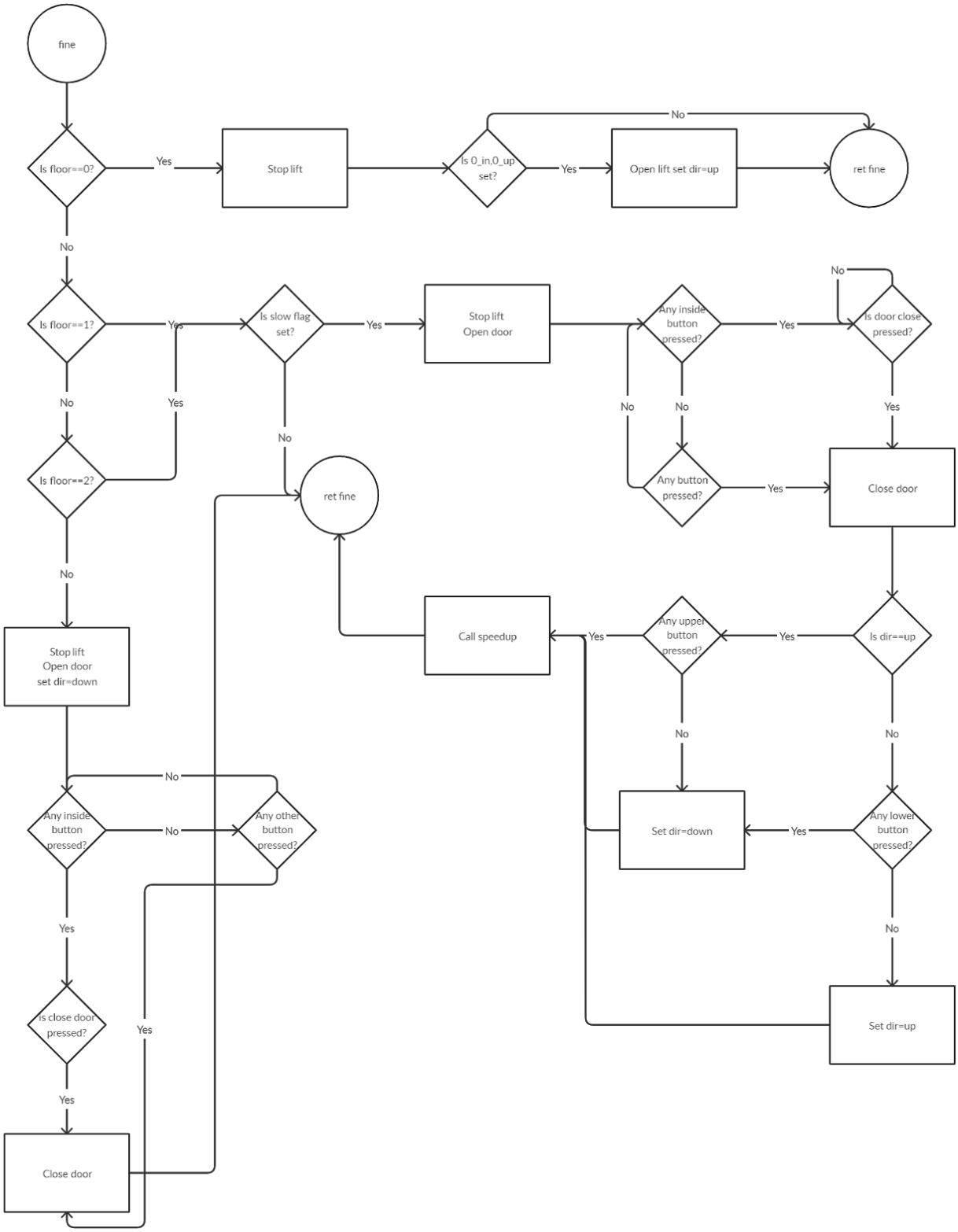


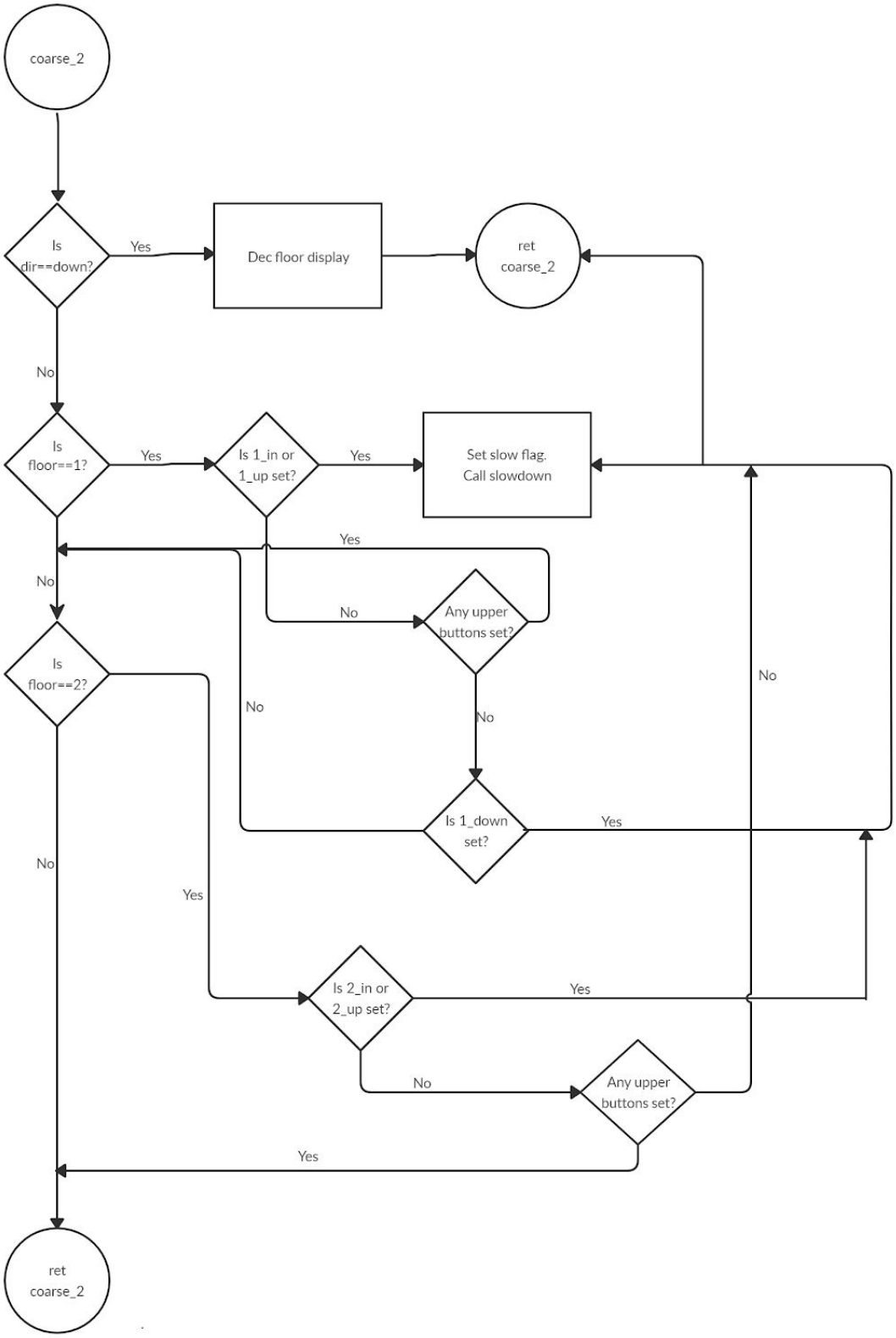

ret
coarse_1
No
check
coarse 2
Yes
call
coarse_2
coarse_2

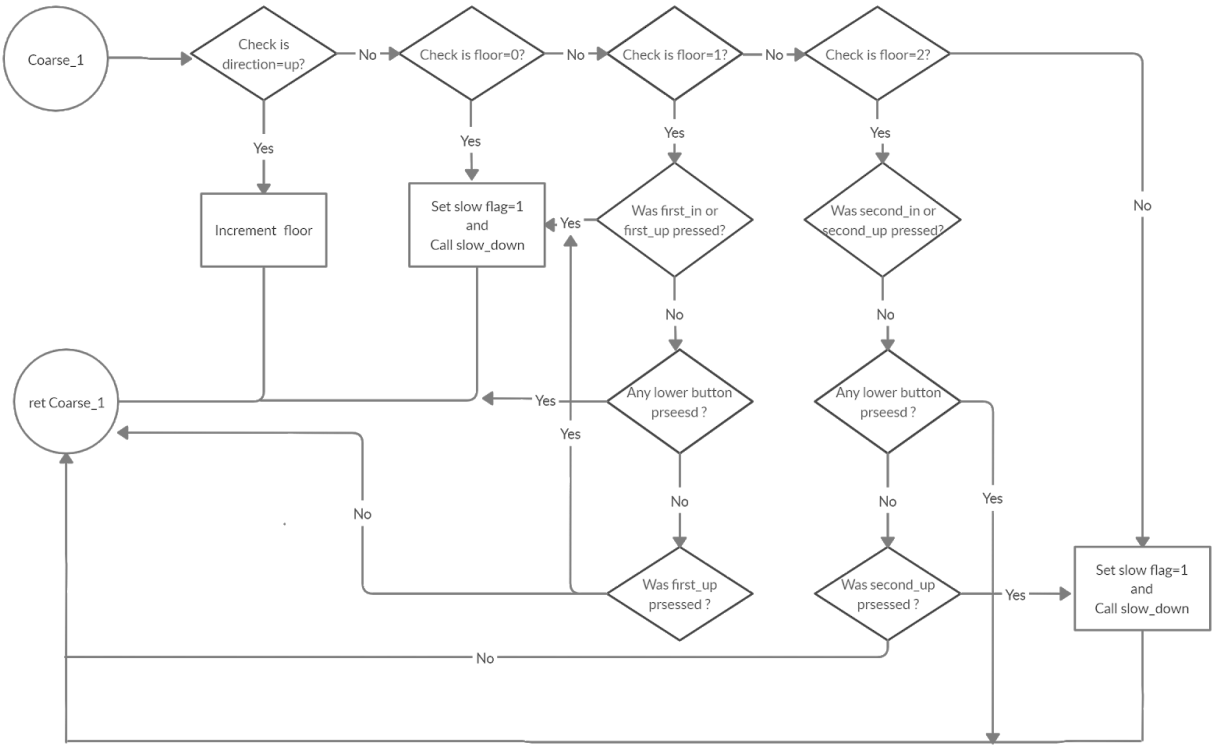

ret
coarse_2
No
check
coarse 2
Yes
Fine
Fine
B


ret
fine
if floor==0
Yes
A
No
Is dir==up
Yes
No
No
Check
coarse2
Yes
Call coarse_2
coarse_2


ret
coarse_2
No
Check Coarse 1
Yes
Call coarse_1
coarse_1
ret
coarse_1
B


fine
Is floor==0?
Yes
Stop lift
Is 0_in,0_up set?
No
Yes
Open lift set dir=up
ret fine
No
Is floor==1?
Yes
Is slow flag set?
Yes
Stop lift
Open door
Any inside button pressed?
Yes
No
Is door close pressed?
No
Yes
No
Yes
No
No
Yes
Is floor==2?
No
ret fine
Any button pressed?
Yes
Close door
No
Stop lift
Open door
set dir=down
Call speedup
Yes
Any upper button pressed?
Yes
Is dir==up
No
No
No
Any inside button pressed?
No
Any other button pressed?
Set dir=down
Yes
Any lower button pressed?
Yes
No
Yes
is close door pressed?
Yes
Set dir=up
Yes
Close door


coarse_2
Is dir==down?
Yes
Dec floor display
ret coarse_2
No
Is floor==1?
Yes
Is 1_in or 1_up set?
Yes
Set slow flag. Call slowdown
Yes
No
No
Any upper buttons set?
Is floor==2?
No
No
No
Is 1_down set?
Yes
No
Yes
Is 2_in or 2_up set?
Yes
No
Any upper buttons set?
Yes
ret coarse_2


Coarse_1
Check is direction=up?
No
Check is floor=0?
No
Check is floor=1?
No
Check is floor=2?
Yes
Yes
Yes
Yes
No
Increment floor
Set slow flag=1 and Call slow_down
Was first_in or first_up pressed?
Was second_in or second_up pressed?
Yes
No
No
ret Coarse_1
Yes
Any lower button prseesd ?
Any lower button prseesd ?
Yes
Yes
No
No
No
Yes
Was first_up prsessed ?
Was second_up prsessed ?
Yes
Set slow flag=1 and Call slow_down
No

Variations in Proteus implementation with Justification

1. 2732 is used as 2716 - not available in Proteus.
2. Clock generator not used for 8086 - Can be edited in 8086 properties.
3. Using 8253 - as 8254 is not available in Proteus
4. Using gate based circuit for memory - does the same as LS 138 here

Firmware

Implemented using emu8086 attached.

List of Attachments

1. Complete hardware Real world design - Design.pdf
2. Manuals
    a. L293D
    b. 74LS447
3. Proteus file - elevator control.dsn
4. emu8086 asm file -elevator control.asm
5. Binary file after assembly- elevator control.bin