BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

SECOND SEMESTER 2018-2019

COURSE NUMBER: CS F241

COURSE TITLE: Micro Processors and Interfacing

DESIGN ASSIGNMENT- ***"SMART OVERHEAD TANK"***

Submitted by:

ARYAN MEHRA – 2017A7PS0077P

KATTA SIVA KUMAR- 2017A7PS0078P

ABHISHEK BHARADWAJ – 2017A7PS0079P

VISHAL MITTAL – 2017A7PS0080P

[**Group – 82, Problem Statement 19**]

TABLE OF CONTENTS

Topic	Page Number
1. Problem Statement	3
2. Project Specifications	4
3. Assumptions Made	5
4. Components Used	6
5. Address Mapping	7
6. Flowchart	8
7. Code	11
8. Circuit Diagram	18
9. References	19

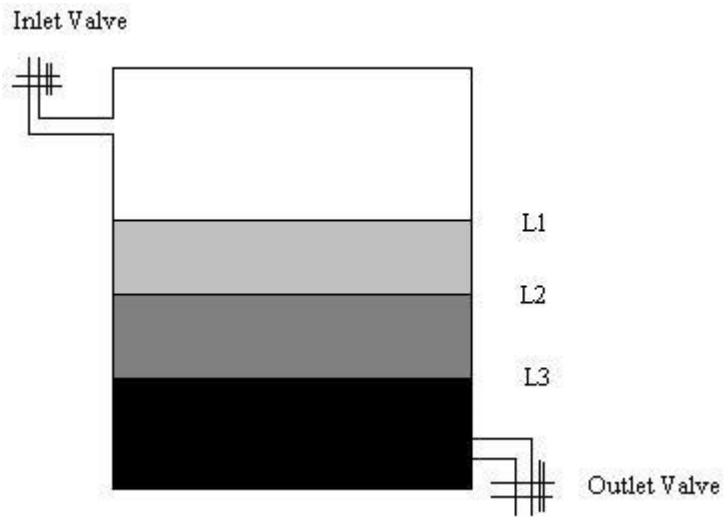
PROBLEM STATEMENT:

P19: System to be Designed : Smart Overhead Tank

Description: This is a tank system in which the water level is maintained according to the time of the day. The water level should be maintained at three different values according to the time of the day.

Peak Hours: Maximum Level of Tank Peak Hours is between 5:00 AM to 9:00 AM in the Morning and 4:00- 6:00 PM in the evening

Low Hours: Minimum level. The rest of the time it is maintained at a nominal level. Low hours is between 12:00 Midnight and 5:00 AM in the morning



The inlet valve draws water from the main-tank system and the outlet valve sends the surplus water back to the main tank. The water in the main tank must be maintained at a constant value, if the level drops the motor must be turned on.

The water tank is used for supplying water to bathrooms and kitchen – sensors used must be non-contact.

SPECIFICATIONS:

- The water level in the smart overhead tank is controlled using a microprocessor system.

- The water level is controlled at the specified level according to the time of the day. INTEL microprocessor 8086 is used as the processing unit for the project.

- An 8KB ROM (2732) and a 4KB RAM (6116) which have been further divided into even and odd parts are interfaced with the microprocessor.

- IC8253, a Programmable Interfacing Timer, is set to mode 3. The counter 0 of the timer has been used. After 60 seconds of interval, which is the analogy of one hour in the model for faster presentation process, the voltage level of out pin goes high which then used as a input for IC8259 [Which interrupts the microprocessor every 60 seconds] (60 Seconds = 1 hour).

- The resistors are added before the switches to make the circuit as realistic as possible.

- De Multiplexing of address-lines has been done using 74LS373. 74LS245 is the bi-directional buffer used at data lines. 74LS138 is used for interfacing 8255, 8253, 8259 to address-bus and data-bus.

- The frequency of the clock used is 4.25Hz.

- Three switches have been used to indicate the sensors placed in the water-tank. And two motors have been used so as to control the inlet valve and outlet valve.

- L293D [Motor Driver IC] is used to drive the unipolar motor which has its inputs from IC 8255. Port B controls the motors.

ASSUMPTIONS MADE:

1. The initial time, when the system is first turned on is assumed to be 00:00AM.

2. We have used the analogy of 60 minutes or 1 hour being equated to 60 seconds on the proteus clock. The clock that is generated is at a frequency of 4.25Hz and so around 255 cycles are calculated to be sufficient to simulate 60 seconds accurately.

3. The sensors are represented as switches in the design. Switch is opened to represent the logic one i.e., the switch is connected to high voltage. The three switches can be imagined as binary representations of the level of water. So when the three pins are logic one (open) then it represents water level seven (highest).

4. NMI has been grounded in this design.

5. Power supply is continuous, once the system is turned on, it is assumed to run continuously.

6. The 3:8 decoder (74LS138) is used to select one out of three (8255 , 8253 and 8259)

COMPONENTS USED:

Components	Model	Number
Microprocessor	INTEL 8086	1
RAM	6116	2
ROM	2732	2
Latch	74LS373	3
Bi-Directional Buffer	74LS245	1
De Mux	74LS138	1
Programmable peripheral Interface	INTEL 8255	1
Programmable Interface Timer	INTEL 8253	1
Programmable Interrupt Controller	INTEL 8259	1
Motor Driver	L293D	2
Unipolar Motor		2
Logic Gates		8
Resistors		3
LED’s		1 (optional)
Switches		3
DPDT		1

ADDRESS MAPPING OF MEMORY AND I/O DEVICES:

1. Initializations:
   Base Address of 8255 – 00 h
   ROM address – 00000h-01FFFh
   RAM address – 02000h-02FFFh

2. Memory Mapping:

   For ROM, 4Kb plus 4KB

   ROM1E – 00000h, 00002h……...……, 01FFCh, 01FFEh

   ROM1O – 00001h, 00003h……, 01FFDh, 01FFFh

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	1	X	X	X	X	X	X	X	X	X	X	X	X

For RAM, 2KB plus 2KB

RAM1E – 02000h, 02002h……...……, 02FFCh, 02FFEh

RAM1O – 02001h, 02003h……, 02FFDh, 02FFFh

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	1	0	X	X	X	X	X	X	X	X	X	X	X	X

A0 and BHE’ are used to select between even and odd chips respectively. A13 is used to differentiate between RAM and ROM.

3. Address Mapping Values

Device	A5	A4	A3
8255	0	0	0
8253	0	0	1
8259	0	1	0

8255:

PORTA - 00H

PORTB - 02H

PORTC - 04H

CREG - 06H

8253:

Counter0 - 08H

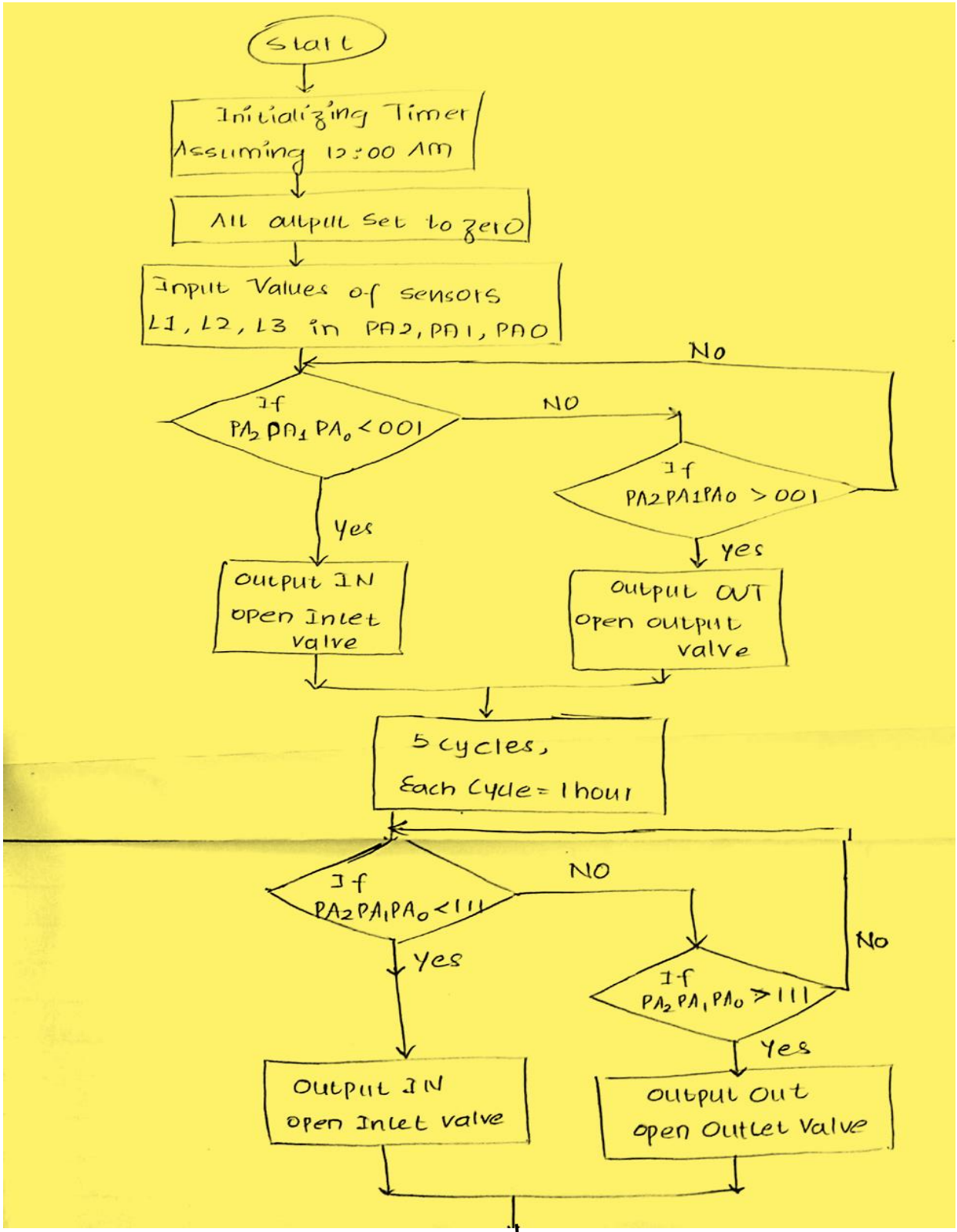
Counter1 – 0AH

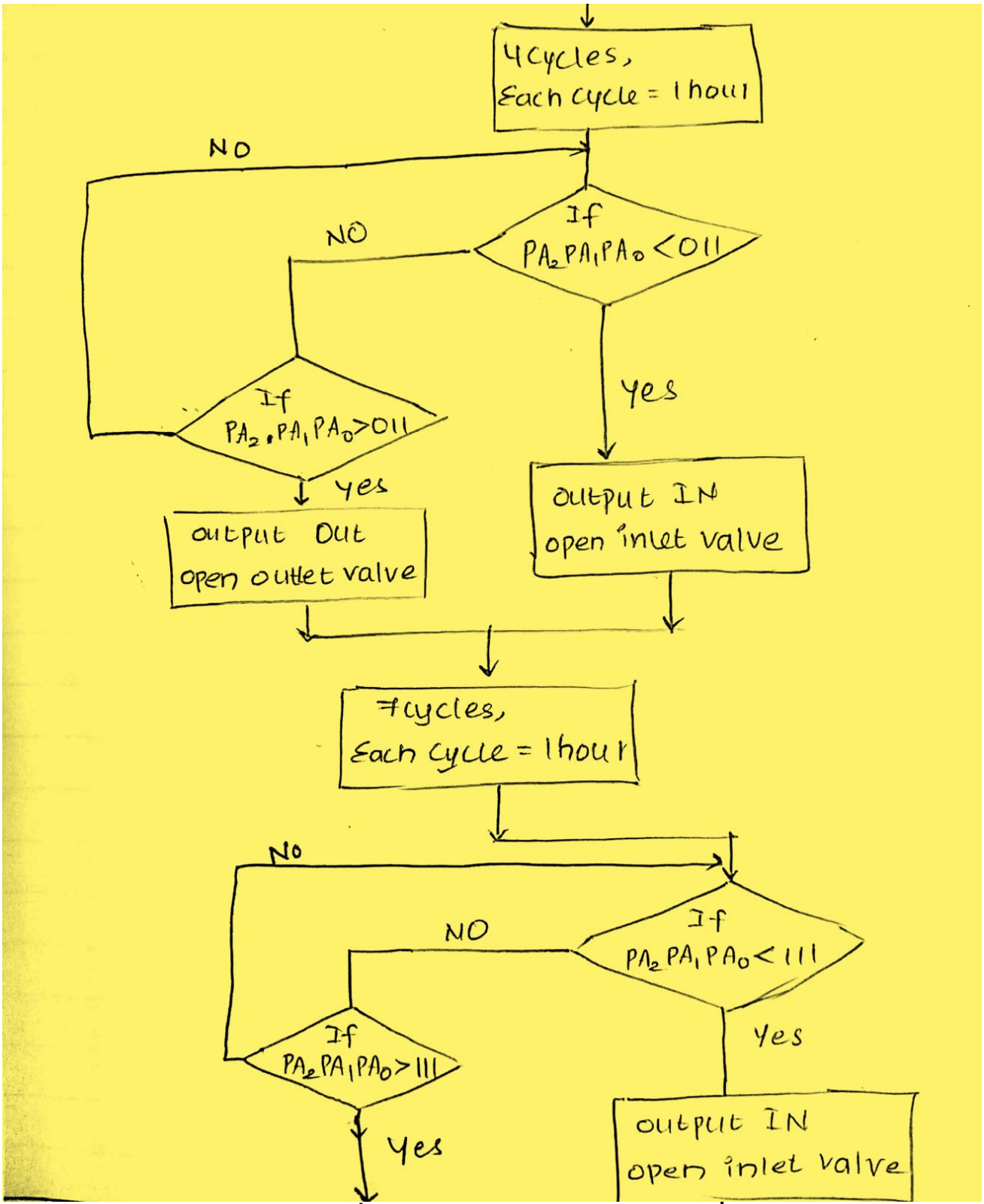
Counter2 - 0CH

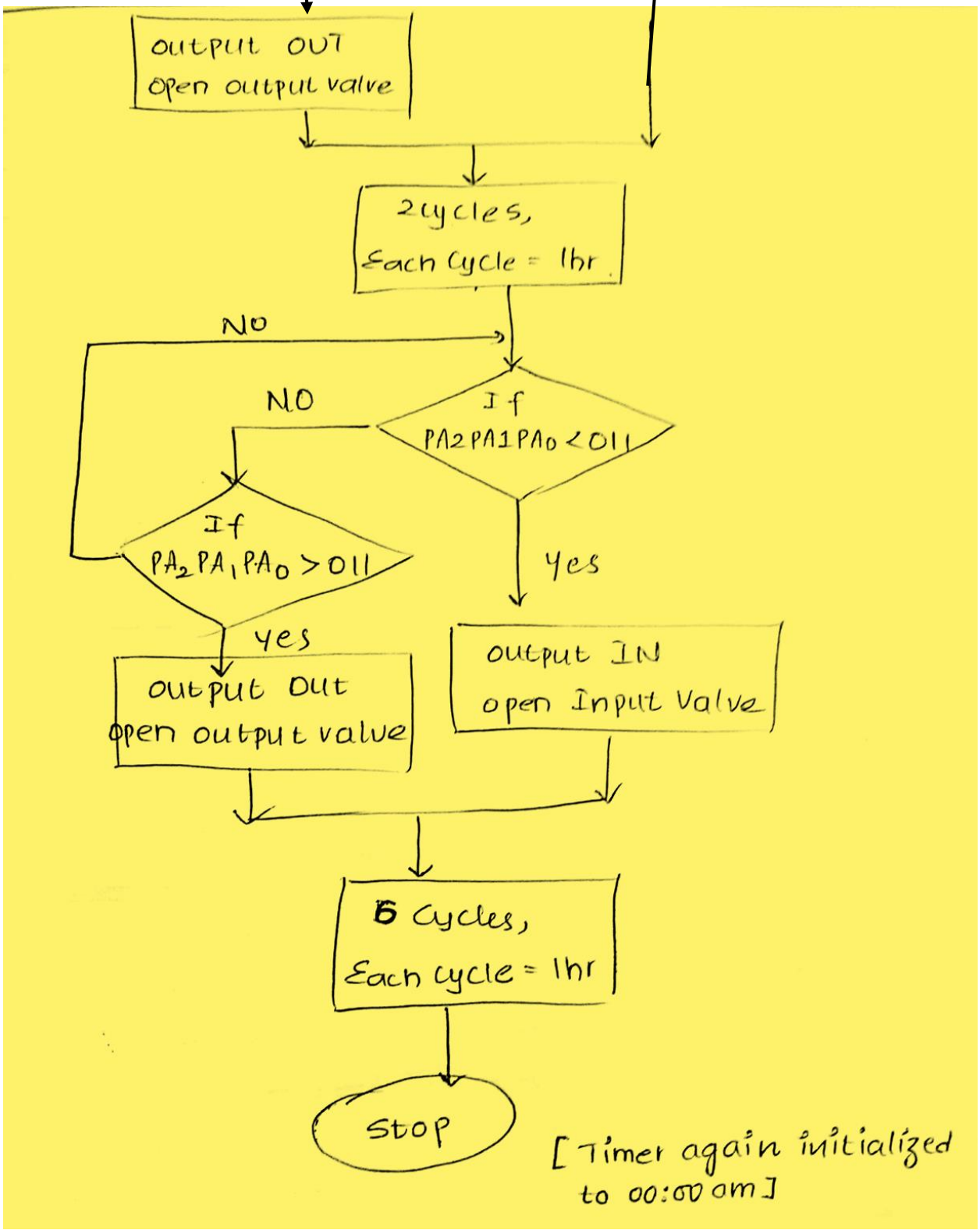
CREG – 0EH

Interrupt Information: Only tenth interrupt INT10H has been used where the IP and CS of the ISR named ‘ac_isr’ is stored.

FLOW CHART:


Start
Initializing Timer
Assuming 12:00 AM
All output set to zero
Input Values of sensors
L1, L2, L3 in PA2, PA1, PA0
If
PA2 PA1 PA0 < 001
NO
No
If
PA2 PA1 PA0 > 001
Yes
Yes
Output IN
open Inlet
valve
Output OUT
open output
valve
5 cycles,
Each Cycle = 1 hour
If
PA2 PA1 PA0 < 111
NO
No
Yes
If
PA2 PA1 PA0 > 111
Yes
Output IN
open Inlet valve
Output Out
open Outlet Valve


4 cycles,
Each cycle = 1 hour
NO
If
$PA_2PA_1PA_0 < 011$
NO
If
$PA_2PA_1PA_0 > 011$
yes
output Out
open outlet valve
yes
output IN
open inlet valve
7 cycles,
Each cycle = 1 hour
NO
If
$PA_2PA_1PA_0 < 111$
NO
If
$PA_2PA_1PA_0 > 111$
yes
yes
output IN
open inlet valve


output OUT
open output valve
2 cycles,
Each cycle = 1hr
NO
If
PA2 PA1 PA0 < 011
NO
If
PA2 PA1 PA0 > 011
yes
yes
output Out
open output valve
output IN
open Input Valve
5 cycles,
Each cycle = 1hr
Stop
[Timer again initialized
to 00:00 am]

CODE FROM ASM FILE:

```
#make_bin#


; BIN is plain binary format similar to .com format, but not
limited to 1 segment;

; All values between # are directives, these values are saved
into a separate .binf file.

; Before loading .bin file emulator reads .binf file with the
same file name.

; All directives are optional, if you don't need them, delete
them.

; set loading address, .bin file will be loaded to this address:


#LOAD_SEGMENT=0000h#

#LOAD_OFFSET=0000h#


; set entry point:

#CS=0000h#      ; same as loading segment

#IP=0000h#      ; same as loading offset


; set segment registers

#DS=0000h#      ; same as loading segment

#ES=0000h#      ; same as loading segment


; set stack

#SS=0000h#      ; same as loading segment

#SP=0000h#      ; set to top of loading segment


; set general registers (optional)

#AX=0000h#
```

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

; BIN file directives are over, my code begins from here

          db 512 dup(0)


          mov ax, 0                 ; initialize the segment
          mov es,ax
          mov al,10h
          mov bl,4h
          mul bl                    ; the result is now '40'
          mov bx,ax
          mov si,offset[ac_isr]     ; we thus store the IP
          mov es:[bx], si           ; desired location for INT10H
          add bx, 2
          mov ax, 0000
          mov es:[bx], ax           ; we then store the segment

          ; var is indicating the initial value of water level
          var  db   01h
          creg equ  06h             ; control register for 8255
          porta equ 00h
```

```
portb     equ  02h
portc     equ  04h
cnt0      equ  08h
creg2     equ  0Eh        ; control register for 8253

cli                       ; disable the interrupt

; initializing port A as input for the switches and port B
; as output for the motors to be controlled
mov  al,90h
out  creg,al
mov  al,00h
out  portb,al                  ;output low at port B

; Control word for 8253 for initializing the time in mode3
mov       al,00010110b
out       0Eh,al
mov       al, 0feh        ; initializing timer for 254
out       cnt0, al

; initialize 8259a
; ICW1 initialized
mov al, 00010111b
out 10h, al
; ICW2
mov al, 00010000b
out 12h, al
```

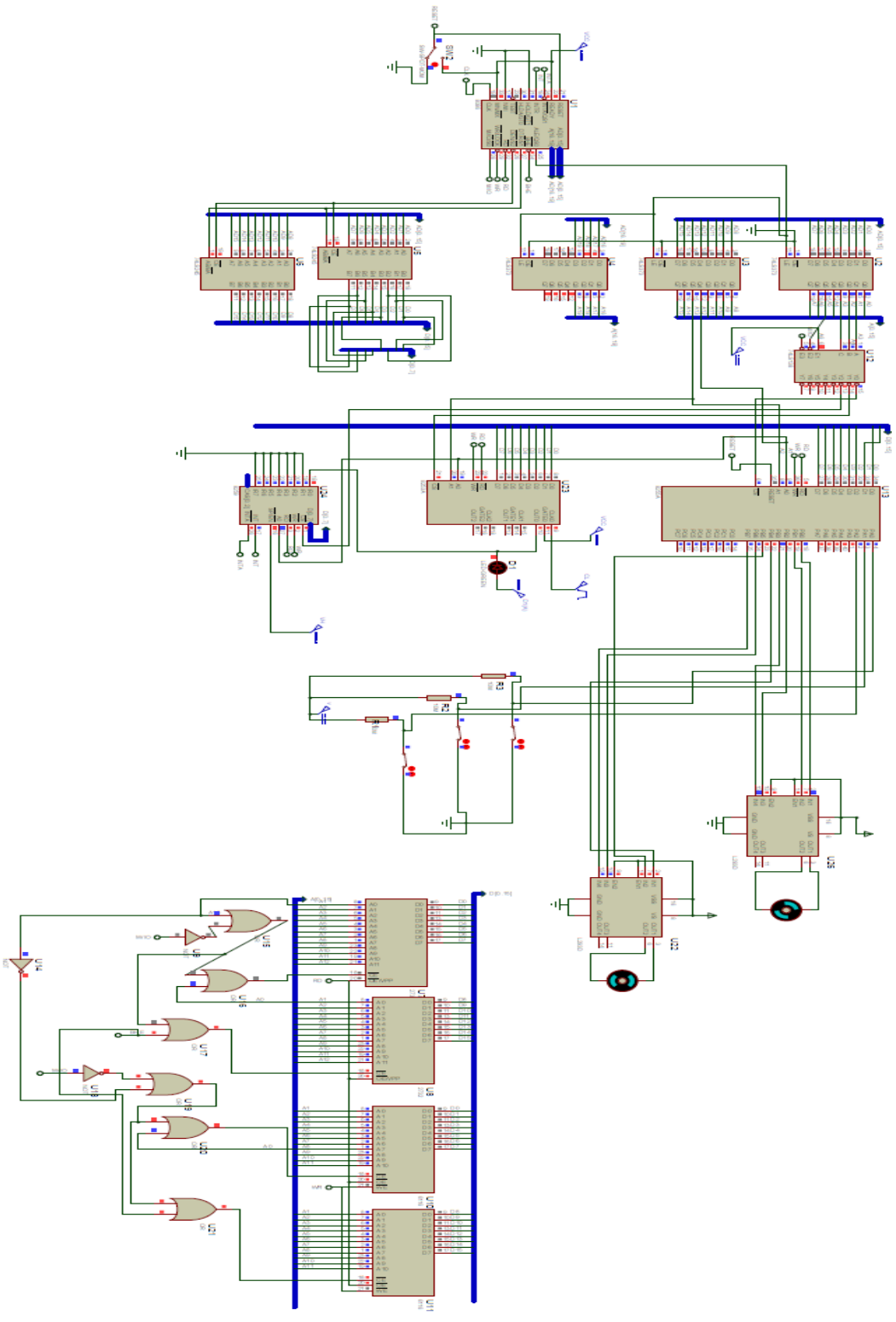
```
; ICW4
          mov al, 00000001b
          out 12h, al
; OCW1 (unmask all interrupt bits)
          mov al, 00h
          out 12h, al

; enable interrupts
          sti
                                    ; reset port c
          mov al, 0
          out 04h, al

; initializing the counter to check the time of day
          mov  cx,0
          mov  bl,10h
; extra register for letting the motors be in same state till a
change occurs in the input

next:     in   al, porta            ; Reading value of porta
          and  al,07h          ; masking other inputs from porta

;comparing with the existing output needed
          cmp  al,var
          jz   x4
          ja   x2
          jb   x3
```

```
; segment if the water level is more than desired

     x2:  cmp  bl,04h
          jz   x5
; switching inlet valve off and outlet valve on
          mov  al,01100011b
          out  portb,al
          mov  bl,04h
          jmp  x5

; segment if the water level is less than the desired level
     x3:  cmp  bl,01h
          jz   x5
; switching inlet valve on and outlet valve off
          mov  al,00110110b
          out  portb, al
          mov  bl,01h
          jmp  x5

; segment if the water level is equal to desired level
     x4:  cmp  bl,00h
          jz   x5

          mov  al,00110011b        ; switching the valves off
          mov  bl,00h
          out  portb, al

x5:       jmp  next
```

```
                ; loop here till a interrupt occurs

ac_isr:

; OCW2 (non-specific EOI command) for resetting ISR
          mov al, 00100000b
          out 10h, al

; increment counter register and check for the time of day
          inc cx

          cmp cx,5
          jz   x8
          cmp cx,9
          jz   x7
          cmp cx,16
          jz   x8
          cmp cx,18
          jz   x7
          cmp cx,24
          jz   x6
          jmp x9

; low level of water in tank
x6:       mov var,01h
          mov cx,00h

; reset the 24 hours clock
          jmp x9
```

```
; medium level of water in tank
x7:       mov var,03h
          jmp x9

; peak level of water in tank
x8:       mov var,05h
x9:       iret                    ; return


.EXIT
END
HLT
; halt!
```

DESIGN: (PDF IS ALSO ATTACHED)



REFERENCES:

https://www.geeksforgeeks.org/pin-diagram-8086-microprocessor/

http://www.ti.com/lit/ds/symlink/sn54ls373-sp.pdf

http://www.sycelectronica.com.ar/semiconductores/74LS138-9.pdf

http://www.ti.com/lit/gpn/sn54ls245-sp

https://www.geeksforgeeks.org/programmable-peripheral-interface-8255/

https://nptel.ac.in/courses/108107029/module10/lecture54.pdf

https://pdos.csail.mit.edu/6.828/2005/readings/hardware/8259A.pdf

http://www.eeeguide.com/programming-8259/

https://www.rakeshmondal.info/L293D-Motor-Driver