

# Microprocessor Programming and Interfacing

## DESIGN ASSIGNMENT

P12- Automatic Washing Machine

SUBMITTED BY

Anirudh S Chakravarthy (2017A7PS1195P)

Raj Sanjay Shah (2017A7PS1181P)

Divyam Goel (2017A7PS1196P)

Mihir A Chavarkar (2017A8PS0026P)



**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI**

**May 2019**

# Index

<b>Problem Statement</b>	<b>2</b>
<b>Components Used</b>	<b>3</b>
<b>Memory Mapping</b>	<b>4</b>
ROM	4
RAM	4
Table 1: Memory interfacing for ROM and RAM	4
<b>I/O Mapping</b>	<b>5</b>
Table 2: Port address for PPI	5
Table 3: Port interfacing with PPI	5
<b>Assumptions</b>	<b>6</b>
<b>Interrupt Vector Table</b>	<b>6</b>
Table 4: Address for Interrupt Service Routine	6
<b>Flow Chart</b>	<b>7</b>
<b>Code</b>	<b>11</b>
<b>Circuit Diagram</b>	<b>17</b>

## Problem Statement

**Description:** An Automatic washing machine with Dryer.

The Washing Machine can handle three different types of load: Light, Medium and Heavy. The Washing Machine has three different cycles: Rinse, Wash and Dry. Depending on the load the number of times a cycle is done and the duration of the cycle varies.

**Light Load:** Rinse- 2 mins, Wash- 3 mins, Rinse – 2 mins, Dry Cycle –2 mins

**Medium Load:** Rinse- 3 mins, Wash- 5 mins and Rinse – 3 mins Dry Cycle –4 mins

**Heavy load:** Rinse - 3 mins, Wash- 5 mins and Rinse – 3 mins, Wash- 5 mins and Rinse – 3 mins, Dry Cycle – 4 mins

- The Washing Machine is a single tub machine.
- The Washing machine is made of a Revolving Tub and an Agitator. The Agitator is activated during the Rinse and Wash cycle; revolving tub is active only during the Dry cycle. The door of the washtub should remain closed as long as the agitator is active.
- Before each cycle the water, level is sensed. At the beginning of the cycle the water level should be at the maximum possible level, the water should be completely drained during dry cycle. The cycle should begin only when the water level is correct.
- At the end of each cycle a buzzer is activated. The user should drain the water at the end of the rinse/wash cycle and refill the water for the next cycle; once this has been completed the user can press the resume button.
- At the beginning of the wash cycle the user should add the detergent.
- At the end of the complete wash process the Buzzer is sounded.
- User can turn off system by pressing STOP Button.
- Different sounds are used for different events.
- Display the load selected using a seven-segment display.

### User Interface:

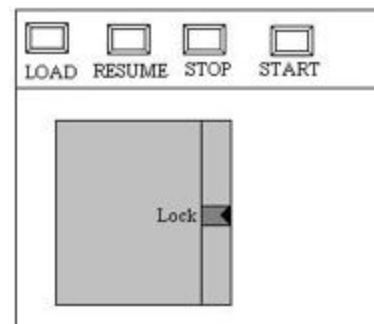
User Interface is shown in fig.

The number of times the load button is pressed determines load:

1 press - light; 2 presses - medium and 3 presses – heavy.

To begin washing process START is pressed.

Pressing STOP can stop the process.



## Components Used

1. 74LS138: 3-line to 8-line Decoder (x1)
2. 74LS245: Bidirectional Octal Buffer (x2)
3. 74LS373: Octal Latch (x3)
4. 2732: 4K EPROM (x2)
5. 6116: 2K SRAM (x2)
6. 7404: NOT gate
7. 7432: 2 input OR gate
8. 4072: 4 input OR gate
9. 4078: 8 input NOR gate
10. 8255: Programmable Peripheral Interface (x1)
11. 8086: Intel Microprocessor (x1)
12. LED (x3)
13. BUZZER (x3)
14. BUTTON (x4)
15. RESISTOR (x7)
16. DC MOTOR (x2)
17. SW-SPST: Latched Action Switch (x3)
18. SW-SPDT-MOM: Momentary Action Switch (x1)
19. RELAY (x3)
20. 7SEG-COM-AN-GRN: 7 Segment Display Green (x1)



## I/O Mapping

### 8255 (Programmable Peripheral Interface)

**Table 2: Port address for PPI**

Port	Address	State
Port A	00h	Input Port
Port B	02h	Output Port
Port C	04h	Output Port (both upper and lower)
Control Word Register	06h	N/A

**Table 3: Port interfacing with PPI**

Port A	Port B	Port C
PA0- Start Button	PB0 - Agitator	PC0-PC3- Input to decoder (BCD to 7 segment)
PA1 - Stop Button	PB1 - Revolving Tub	
PA2 - Load Button	PB2 - Buzzer (dry)	
PA3 - Resume Button	PB3 - Buzzer (wash)	
PA4 - Door lock switch	PB4 - Buzzer (rinse)	
PA5 - Water Max switch		
PA6 - Water min switch		

## Assumptions

1. Water level min and max are modelled as switches. In real life scenario, they will be pressure sensitive (pressed automatically if water level is min/max) and not require manual intervention.
2. Agitators and revolving tub are modelled using DC motors.
3. The door is assumed to be locked when the DC motor (agitator or revolving tub) is running. Although there is a check to ensure door is locked before the start of the operation, it is assumed that the user cannot open the door during this operation.
4. Before a wash cycle, the user is given 1 minute to put detergent.

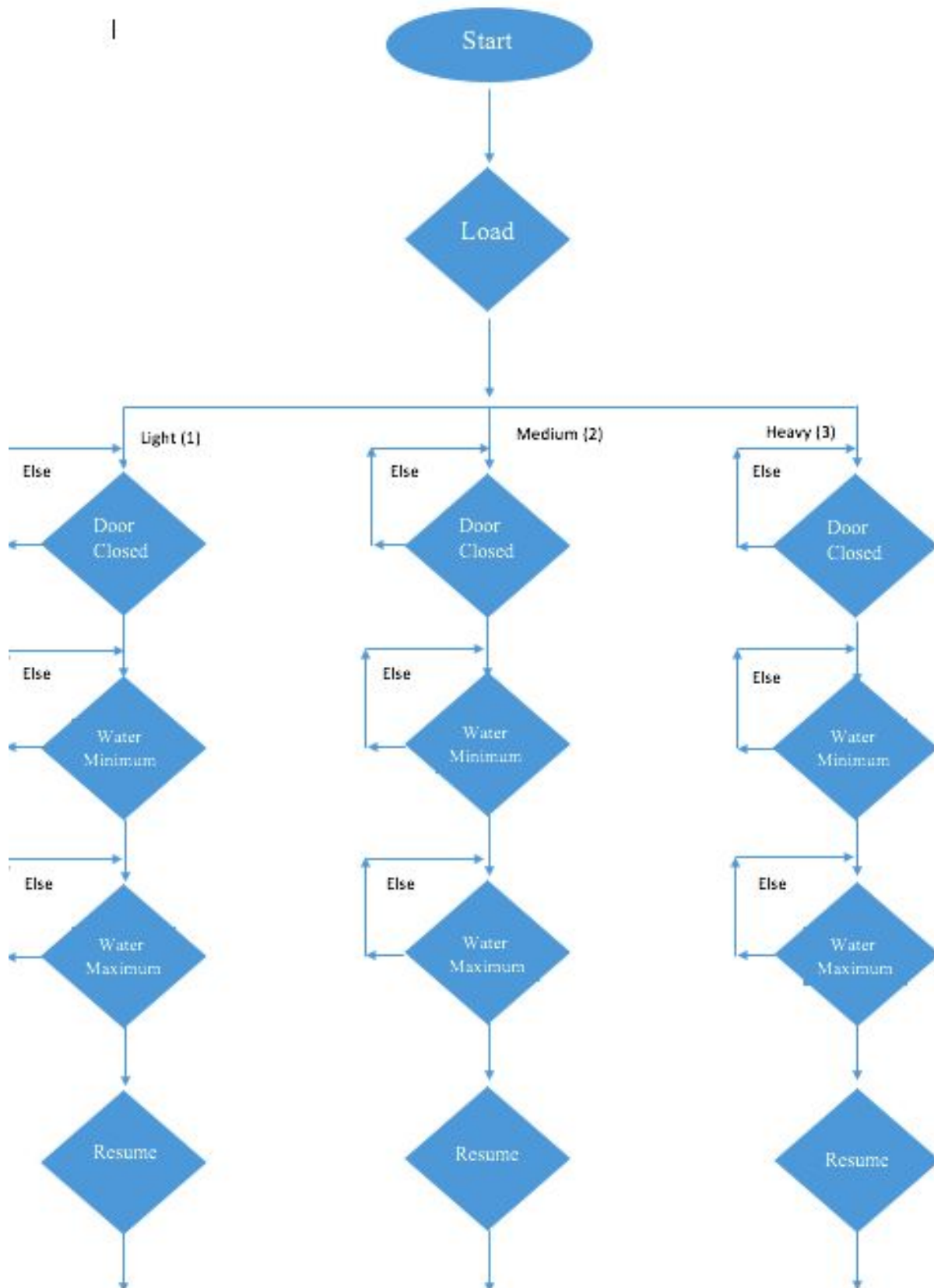
## Interrupt Vector Table

Non maskable Interrupt (Vector Number 2) is used to model STOP button operation.

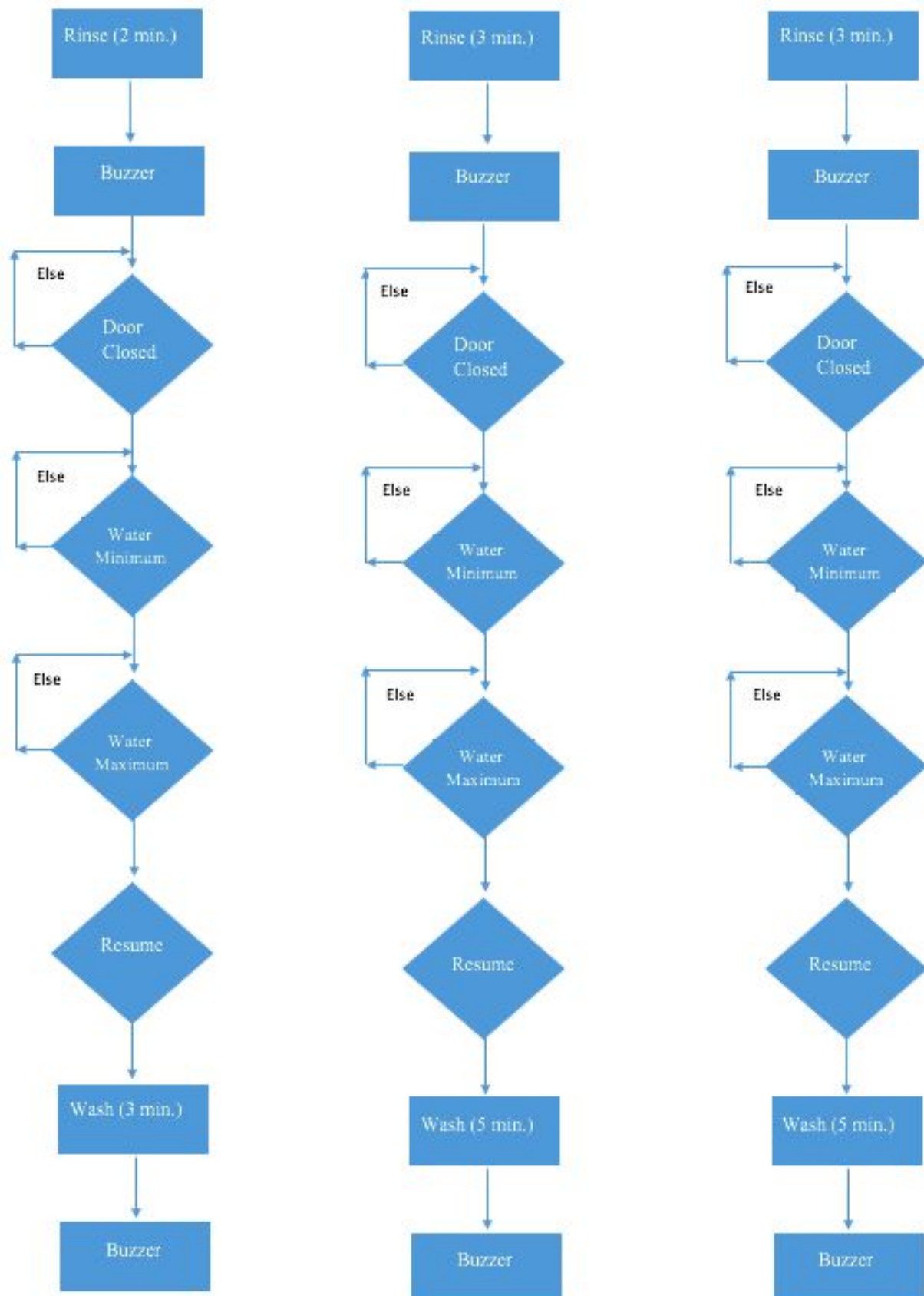
**Table 4: Address for Interrupt Service Routine**

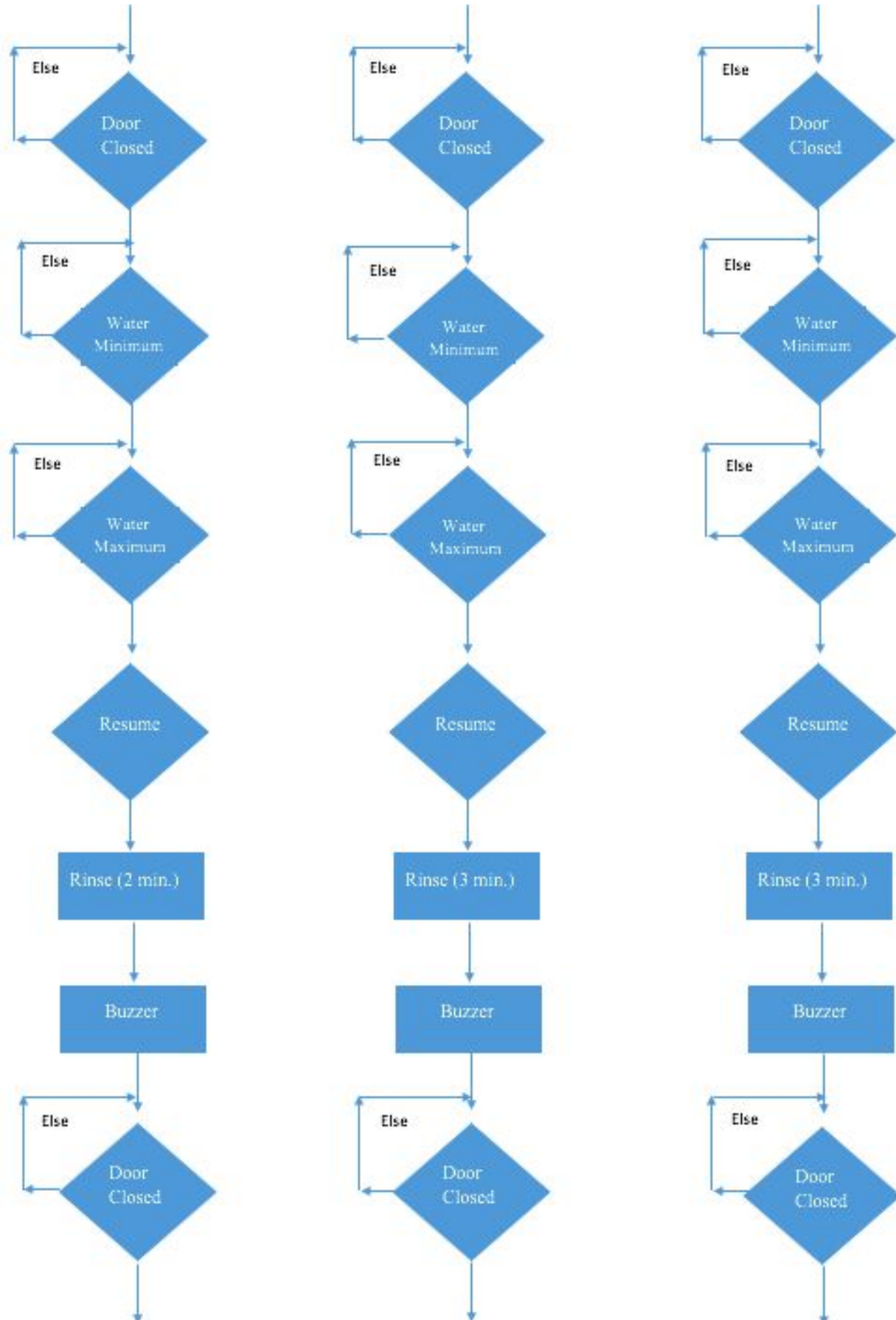
Address	Value
00008h	IP (LSB)
00009h	IP (MSB)
0000Ah	CS (LSB)
0000Bh	CS (MSB)

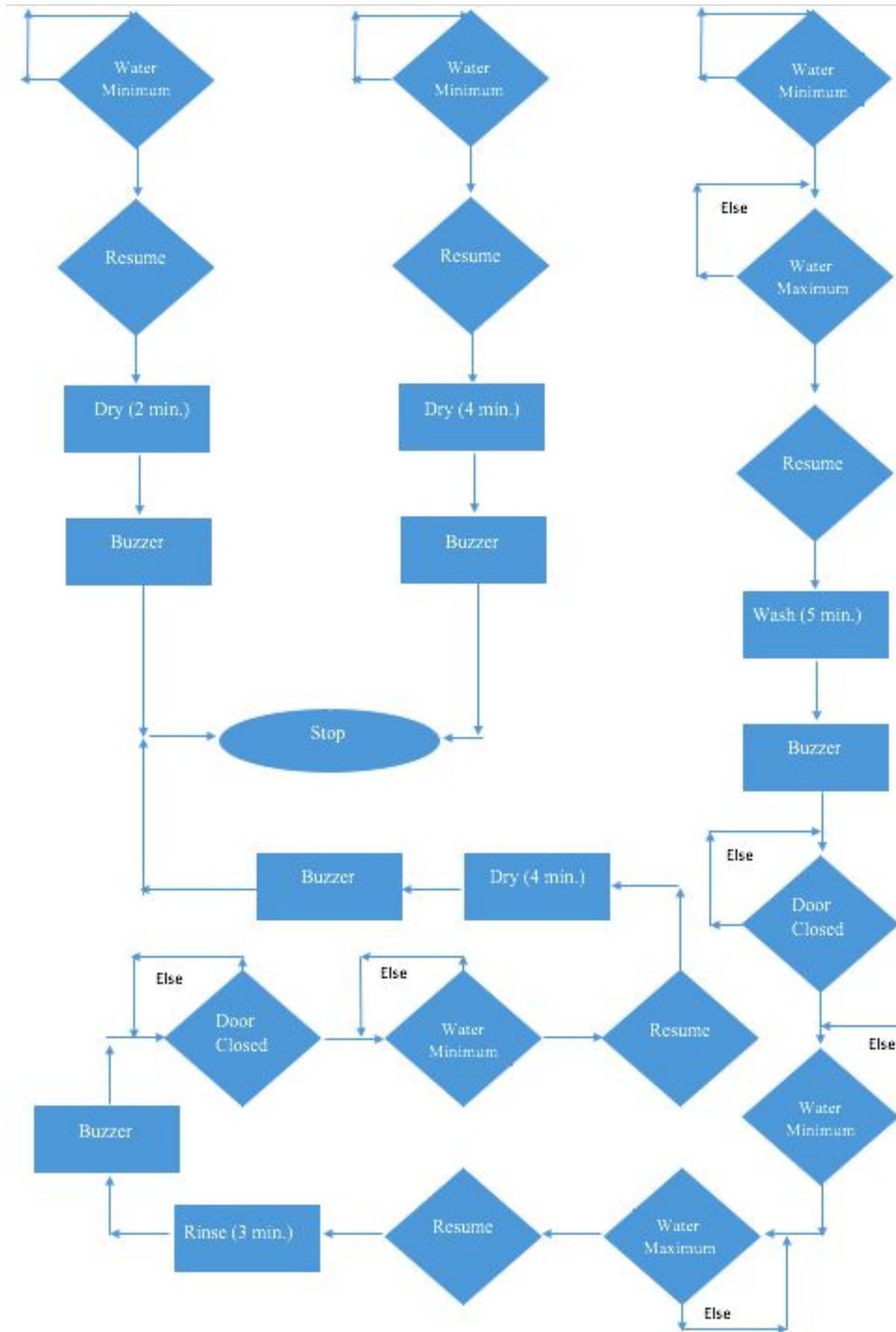
## Flow Chart











## Code

```
.model tiny

; --- MACROS --- ;

; macro for rinse cycle
RINSE_CYCLE MACRO DURATION
    MOV AL, 00000001b
    OUT PORTB, AL ; turn on agitator
    MOV CX, DURATION
    CALL DELAY
    CALL RINSED
ENDM

; macro for wash cycle
WASH_CYCLE MACRO DURATION
    MOV AL, 00000001b
    OUT PORTB, AL ; turn on agitator
    MOV CX, DURATION
    CALL DELAY
    CALL WASHED
ENDM

; macro for dry cycle
DRY_CYCLE MACRO DURATION
    MOV AL, 00000010b
    OUT PORTB, AL ; turn on revolving tub
    MOV CX, DURATION
    CALL DELAY
    CALL DRIED
ENDM

; macro for consecutive rinse and wash cycles
RINSE_WASH MACRO RINSE_TIME, WASH_TIME
    CALL WATER_LEVEL_MIN
    CALL WATER_LEVEL_MAX
    CALL RESUMED
    CALL DEBOUNCE_DELAY
    RINSE_CYCLE RINSE_TIME ; RINSE cycle
```

```
    CALL WATER_LEVEL_MIN
    CALL WATER_LEVEL_MAX
    MOV CX, 1
    CALL DELAY ; user enters detergent during this delay period
    CALL RESUMED
    CALL DEBOUNCE_DELAY
    WASH_CYCLE WASH_TIME ; WASH cycle
ENDM
```

; macro for consecutive rinse and dry cycles

```
RINSE_DRY MACRO RINSE_TIME, DRY_TIME
    CALL WATER_LEVEL_MIN
    CALL WATER_LEVEL_MAX
    CALL RESUMED
    CALL DEBOUNCE_DELAY
    RINSE_CYCLE RINSE_TIME ; RINSE cycle
```

```
    CALL WATER_LEVEL_MIN
    CALL RESUMED
    CALL DEBOUNCE_DELAY
    DRY_CYCLE DRY_TIME ; DRY cycle
ENDM
```

```
BUZZER MACRO BUZZER_NUM
    MOV AL, BUZZER_NUM
    OUT PORTB, AL
    MOV CX, 1
    CALL DELAY ; turn on buzzer for 1 minute
    MOV AL, 00h
    OUT PORTB, AL ; turn off buzzer
ENDM
```

; --- CODE --- ;

.data

```
    PORTA EQU 00h
    PORTB EQU 02h
    PORTC EQU 04h
    CREG EQU 06h
    MODE DB 00h
```

.code

.startup

; storing appropriate CS and IP values for interrupt handling

MOV AX, 0

MOV ES, AX

MOV BX, 0008h ; address for NMI

MOV SI, OFFSET [STOP]

MOV ES:[BX], SI ; IP address

ADD BX, 2

MOV AX, 0000h

MOV ES:[BX], AX ; CS address

; initializing 8255 using control word reg.

MOV AL, 10010000b

OUT CREG, AL

; check if start button is ON(Active Low)

START:

MOV AL, 00h

OUT PORTB, AL

MOV MODE, 00h

IN AL, PORTA

CMP AL, 11111110b

JNZ START

CALL DEBOUNCE\_DELAY

MOV AL, 00h

OUT PORTC, AL

; check for number of load presses

LOAD:

MOV AL, MODE

OUT PORTC, AL ; display mode number on 7 seg display

IN AL, PORTA

CMP AL, 11101111b ; check if door is closed

JZ DOOR\_CLOSED

CMP AL, 11111011b ; else if load button pressed

JNZ LOAD

INC BYTE PTR MODE

CALL DEBOUNCE\_DELAY

CMP MODE, 03h

JLE LOAD

MOV MODE, 00h ; reset to 0 if 4 presses

JMP LOAD

```
; door is now closed
DOOR_CLOSED:
    MOV AH, MODE
    CMP AH, 00h
    JE START ; reset the machine if 0 load presses

; jump to the valid mode
CMP MODE, 01h
JZ LIGHT
CMP MODE, 02h
JZ MEDIUM
JMP HEAVY

LIGHT:
    RINSE_WASH 2, 3
    RINSE_DRY 2, 2
    JMP START

MEDIUM:
    RINSE_WASH 3, 5
    RINSE_DRY 3, 4
    JMP START

HEAVY:
    RINSE_WASH 3, 5
    RINSE_WASH 3, 5
    RINSE_DRY 3, 4
    JMP START

; interrupt service routine for pressing STOP button
STOP:
    MOV AL, 00h
    OUT PORTB, AL
    OUT PORTC, AL
    POP AX ; pop previous IP address location
    POP AX ; pop previous CS address location
    PUSH CS ; push CS address
    MOV DX, OFFSET [START]
    PUSH DX ; moves IP address to START label
    IRET
```

.exit

; --- PROCEDURES --- ;

; introduce delay in the system - DURATION held in CX register

DELAY PROC NEAR USES BX DX

```
L0:      MOV BX, 00E0h
L1:      MOV DX, 0FFFFh
L2:      NOP
          DEC DX
          JNZ L2
          DEC BX
          JNZ L1
```

LOOP L0

RET

DELAY ENDP

; ensure no button is pressed

DEBOUNCE\_DELAY PROC NEAR

```
DEBOUNCE:
          IN AL, PORTA
          OR AL, 11110000b
          CMP AL, 11111111b
          JNE DEBOUNCE
```

RET

DEBOUNCE\_DELAY ENDP

; check if water level is maximum and door is closed

WATER\_LEVEL\_MAX PROC NEAR

```
MAX:      IN AL, PORTA
          CMP AL, 11001111b
          JNE MAX
```

RET

WATER\_LEVEL\_MAX ENDP

; check if water level is minimum and door is closed

WATER\_LEVEL\_MIN PROC NEAR

```
MIN:      IN AL, PORTA
          CMP AL, 10101111b
          JNE MIN
```



```
    RET
WATER_LEVEL_MIN ENDP
```

```
; check if resume button is pressed and door is closed
```

```
RESUMED PROC NEAR
```

```
    RESUMEOFF:
```

```
        IN AL, PORTA
```

```
        OR AL,11100111b
```

```
    CMP AL,11100111b
```

```
    JNE RESUMEOFF
```

```
    RET
```

```
RESUMED ENDP
```

```
; rinse cycle completed
```

```
RINSED PROC NEAR
```

```
    MOV AL, 00h
```

```
    OUT PORTB, AL ; turn off agitator
```

```
    BUZZER 00010000b
```

```
    RET
```

```
RINSED ENDP
```

```
; wash cycle completed
```

```
WASHED PROC NEAR
```

```
    MOV AL, 00h
```

```
    OUT PORTB, AL ; turn off agitator
```

```
    BUZZER 00001000b
```

```
    RET
```

```
WASHED ENDP
```

```
; dry cycle completed
```

```
DRIED PROC NEAR
```

```
    MOV AL, 00h
```

```
    OUT PORTB, AL ;turn off revolving tub
```

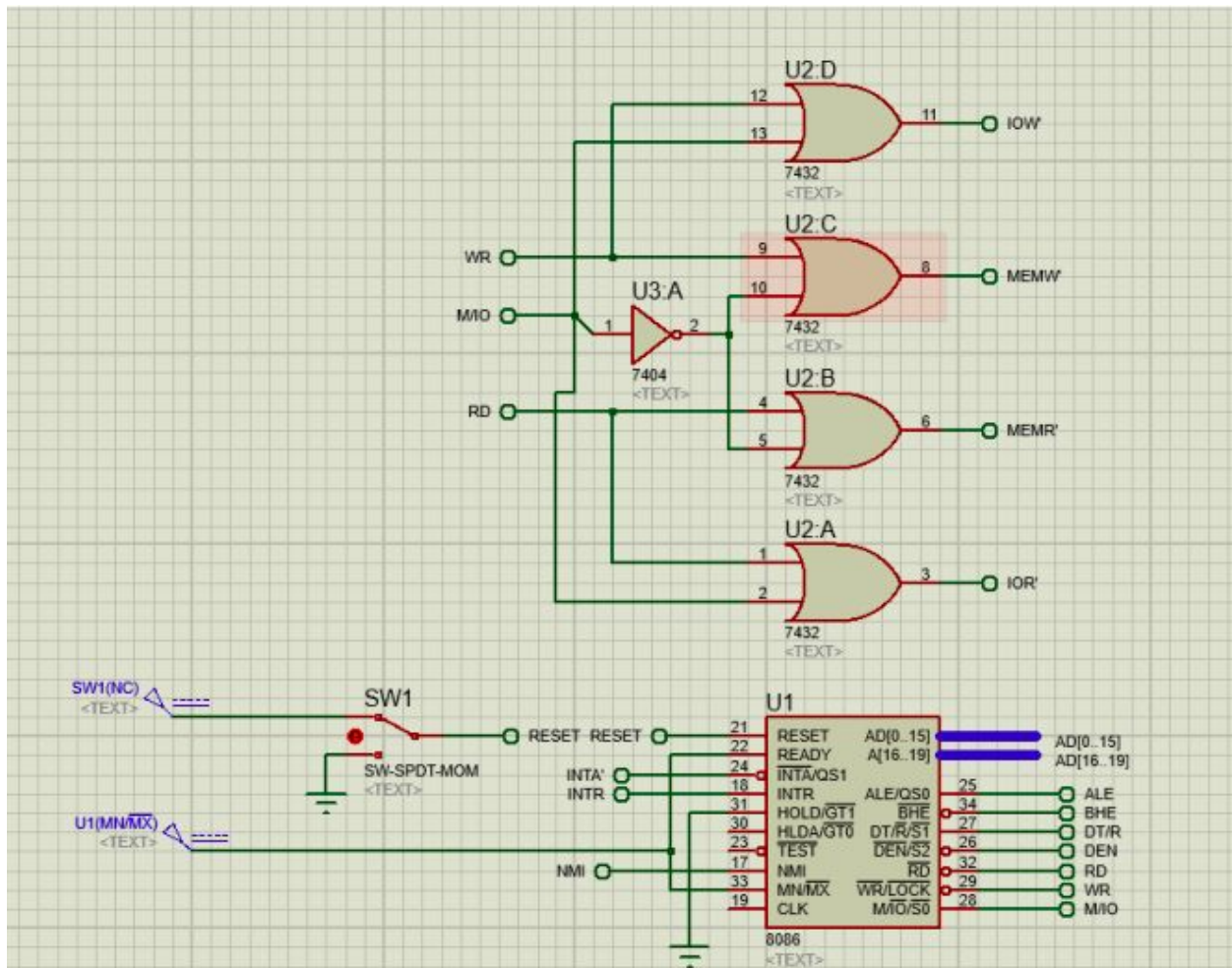
```
    BUZZER 00000100b
```

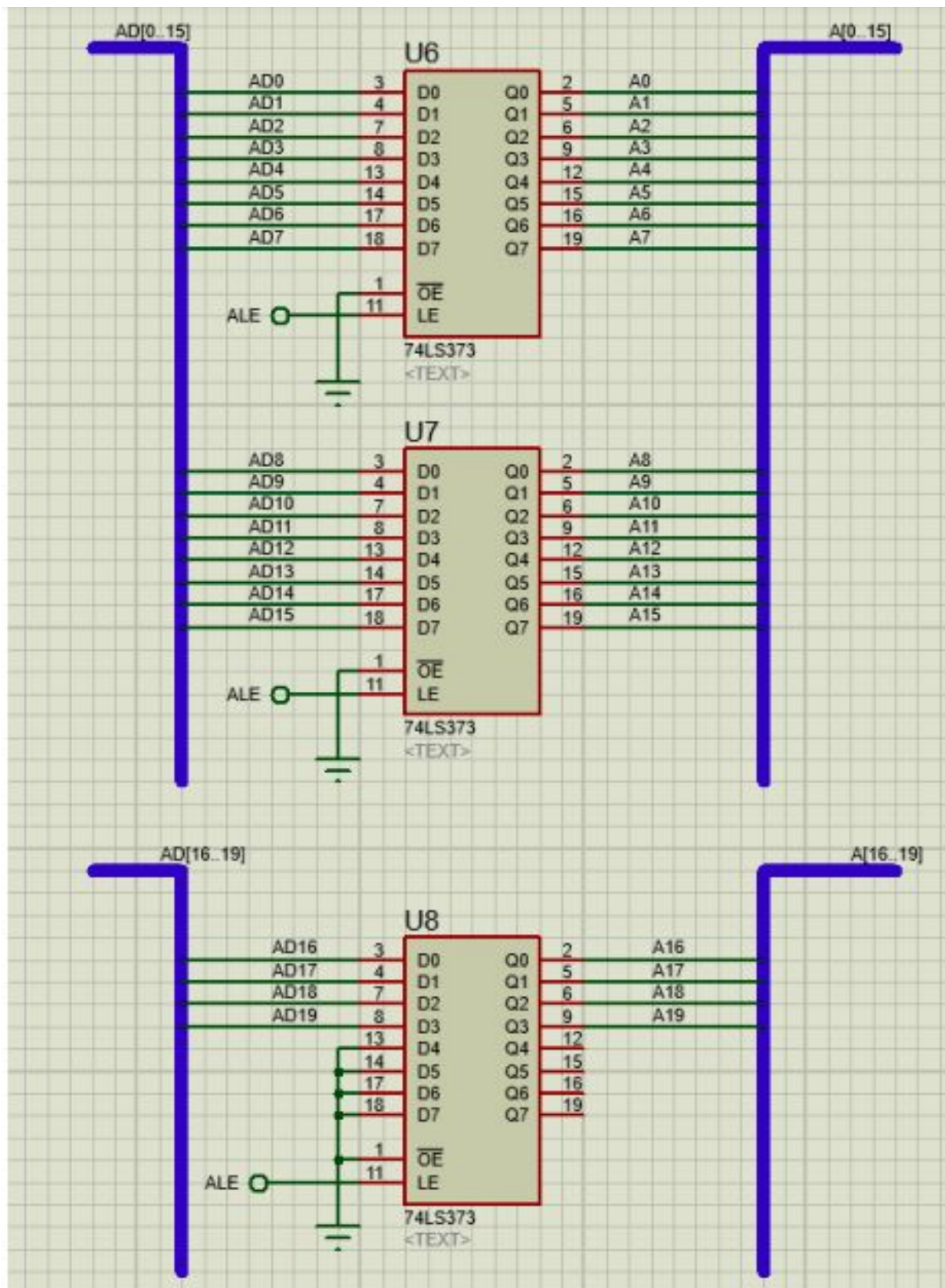
```
    RET
```

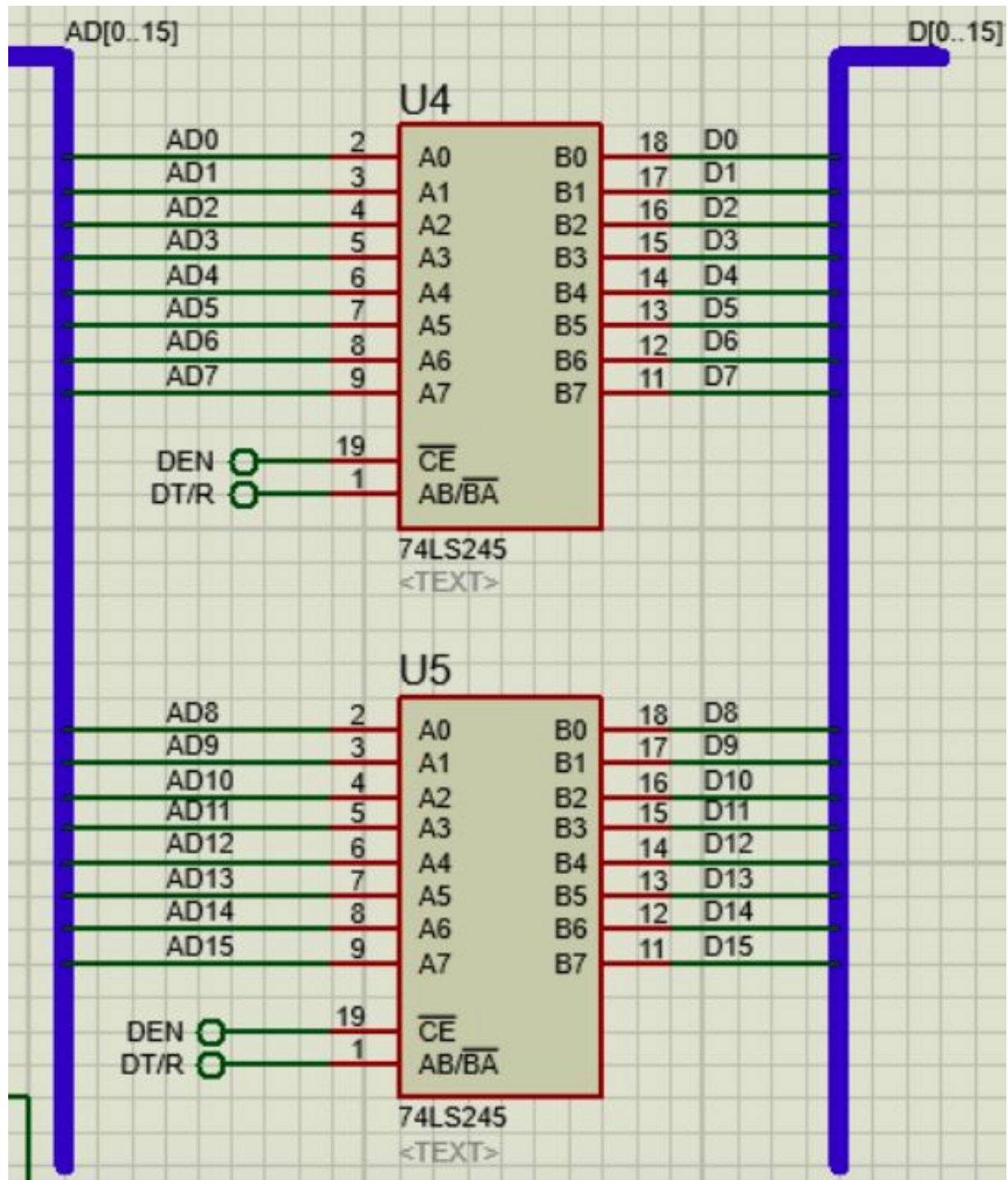
```
DRIED ENDP
```

```
END
```

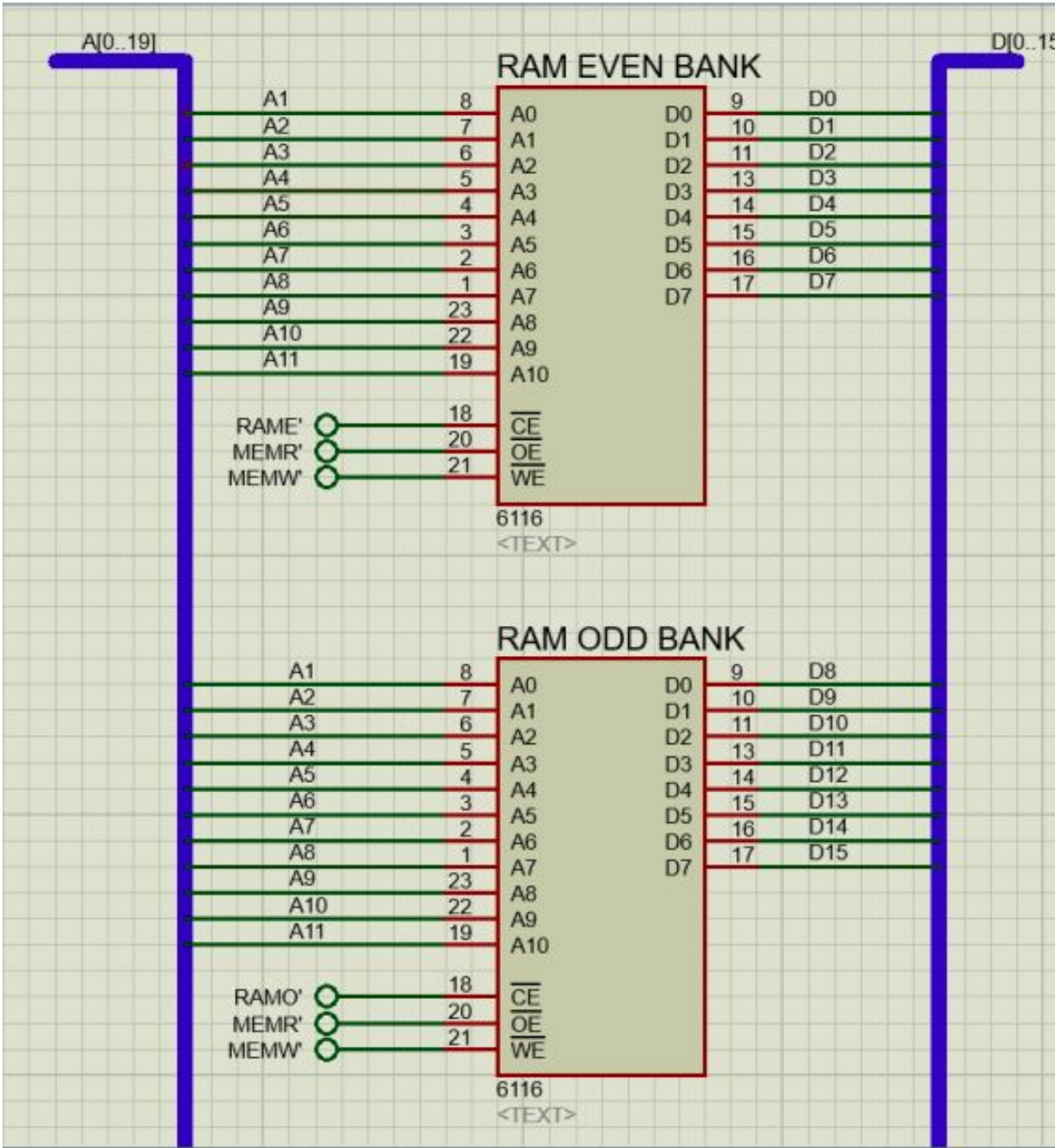
## Circuit Diagram

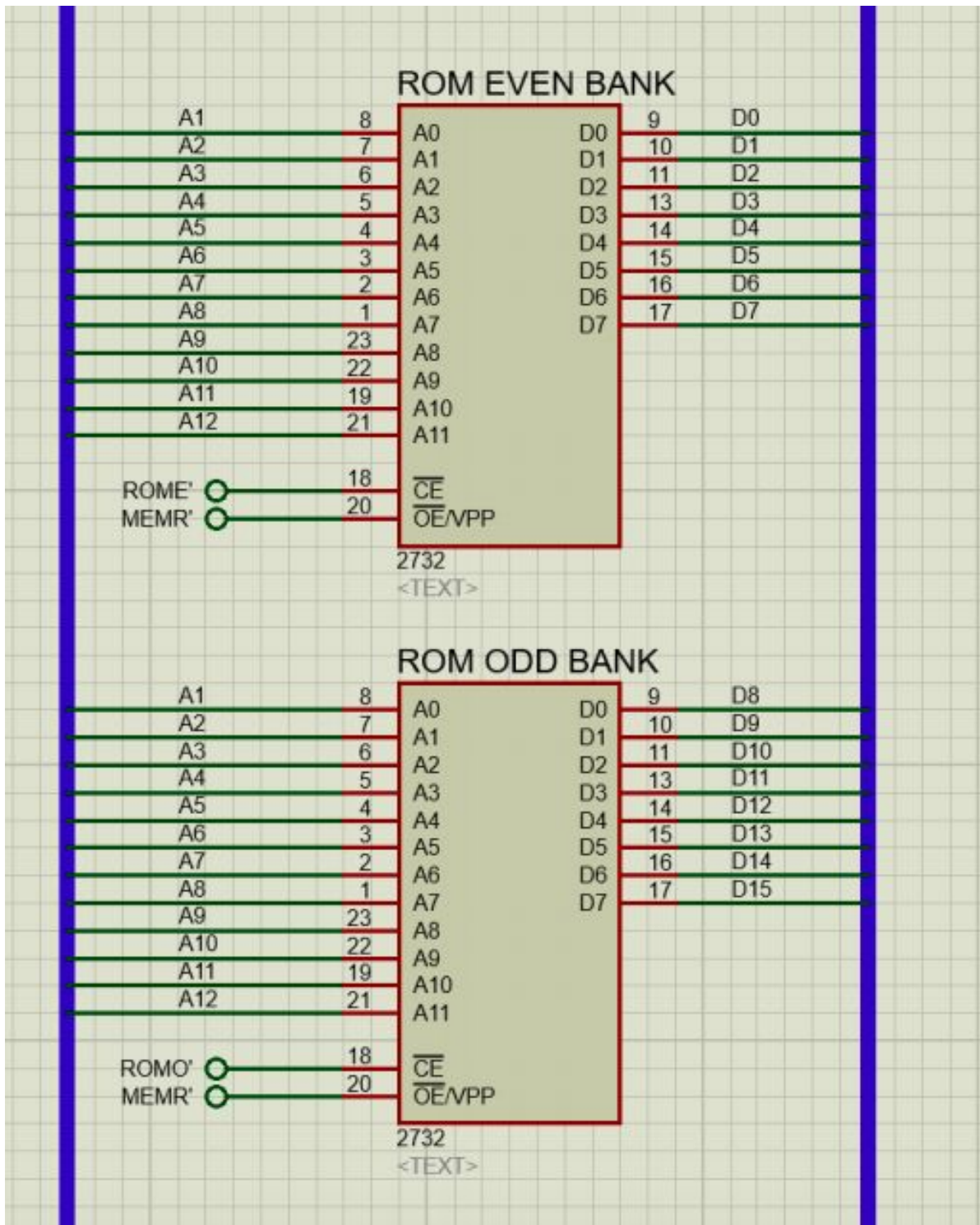


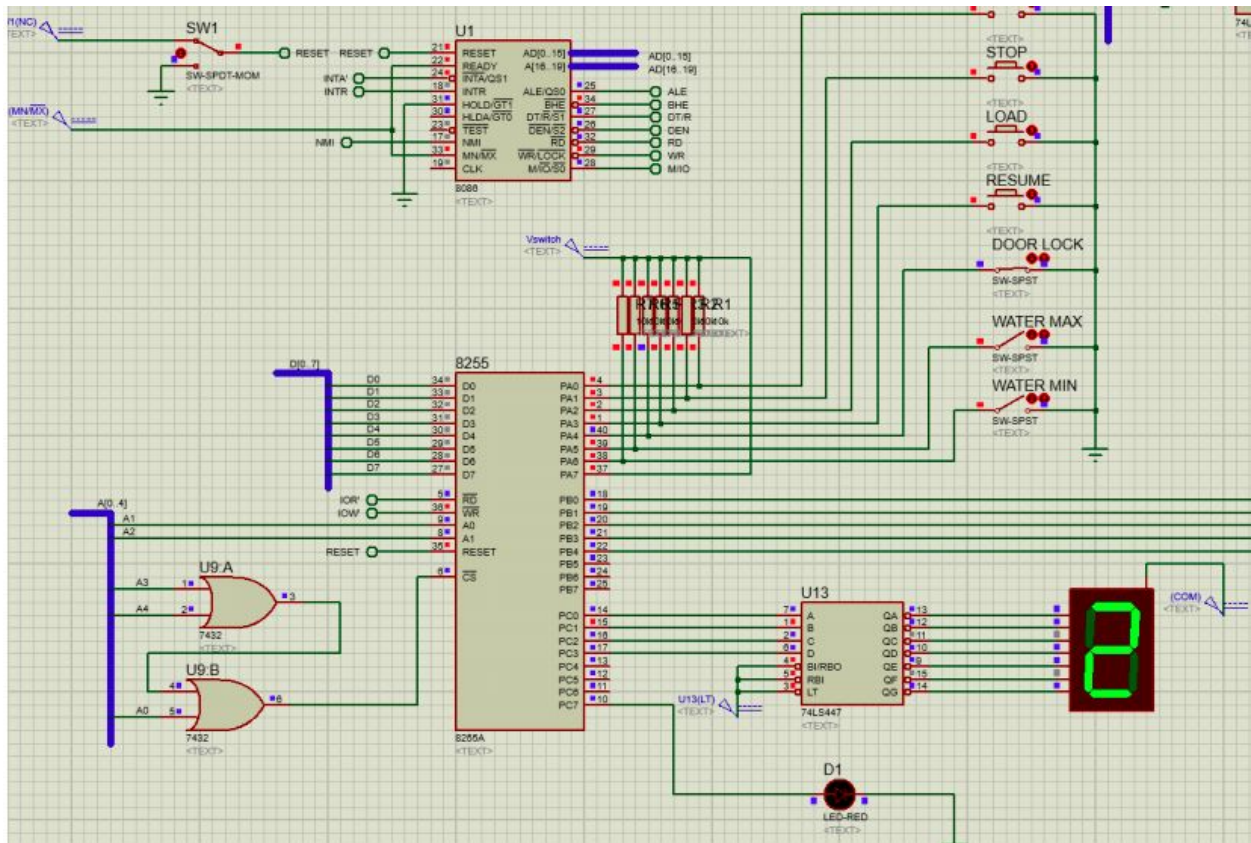
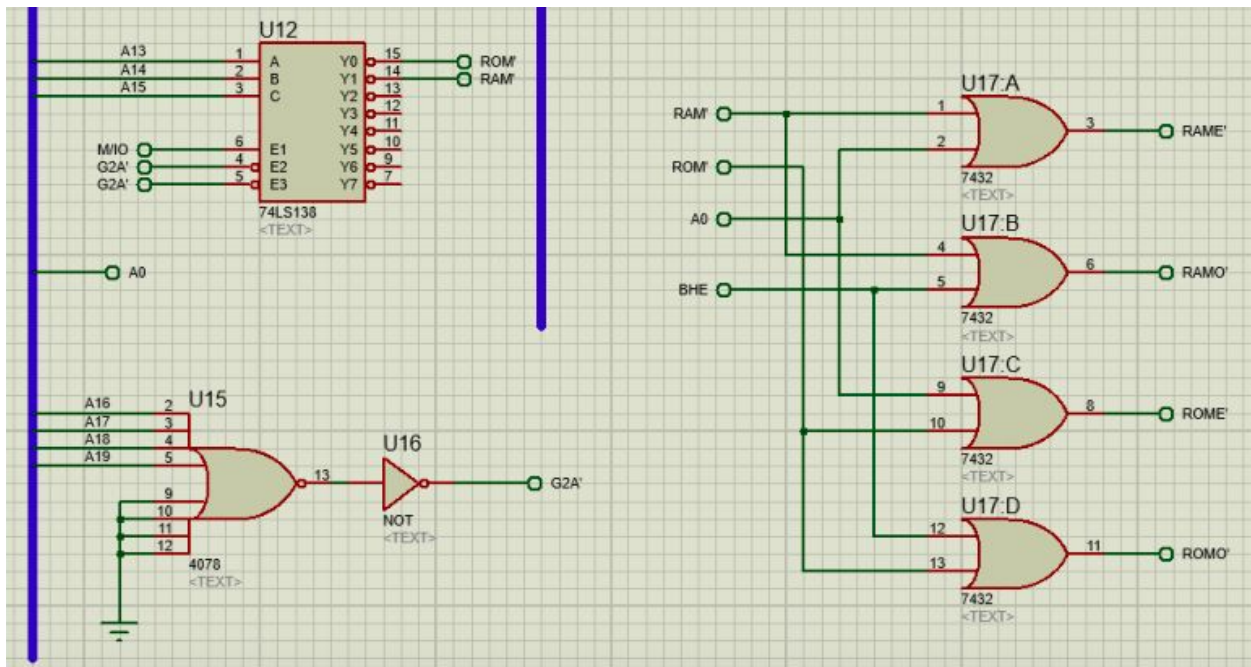




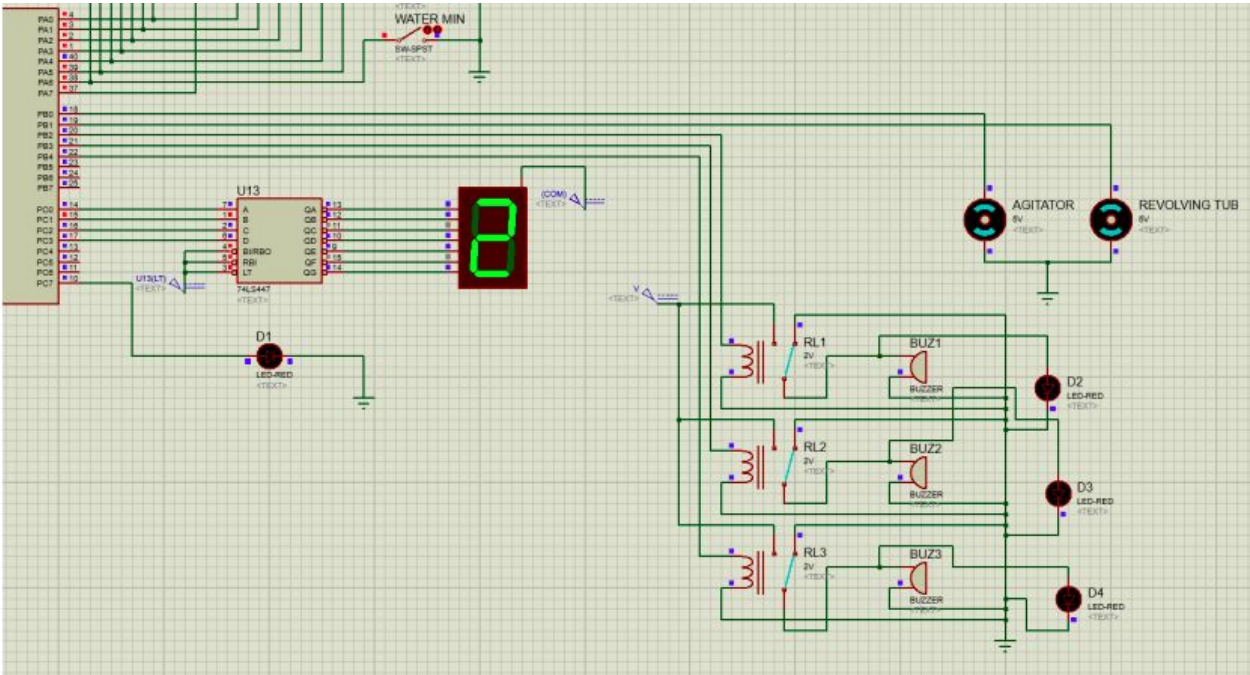














# Thank You