SAFE-LOCKER

Helwan University

MARCH 22, 2021

FACULTY OF ENGINEERING

Computer Department

أشرف محمد محمد عبد المنعم	**Sec. (1)**
عبد الرحمن ياسر أحمد محمد	**Sec. (4)**
عبد الله محمود أحمد محمود	**Sec. (4)**
عبد الله مصطفى محمد الكيلاني	**Sec. (4)**
محمد ناصر السيد أحمد	**Sec. (5)**

TABLED OF CONTENT:

CONTENTS

Tabled of content: 1

Contents 1

1. Introduction: 2

2. Scope: 2

3. Overview description: 3

4. Constraints and Assumptions: 4

5. Tools: 4

6. Functional Requirements: 5

   Enumerated Functional Requirements: 5

   Enumerated Nonfunctional Requirements: 6

      Performance Requirements: 6

      Safety and Security Requirements: 6

7. Flow Chart (Safe-Locker): 7

8. Safe-Lock (Proteus Design): 8

9. Hardware Description: 9

   a. 16*2 LCD (Liquid Crystal Oscillator): 9

   b. Programmable peripheral interface [PPI] (8255): 15

   c. Microprocessor 8086: 17

   d. 4*4 keypad: 20

      debouncing: 21

   e. Buffer 74HC245(Data Buffering): 22

   f. Chip 74HCT373(Address buffering): 23

10. Nomenclature: 25

1. INTRODUCTION:

First of all, we will introduce an overview description of our safe lock application. Many and many companies, large valuable shops, banks and even business men …etc. seeks to place their money in a portion that is safe that's why safe lock application is very important now days so from that point we start to establish a well-defined document explains how software can be created for safe lock applications and also constructing a software simulation that simulates our project.

2. SCOPE:

By using 8086 processor to program the safe locker, you can set and change password which is displayed on LCD. if user enter a wrong password, Else the Blue led is glowed where this led is simulate the opening of cash safe locker there is keypad to enter the password and the cash safe locker supports changing the password for many times where we will use non-volatile memory which is EEPROM in which the password is saved.

3. OVERVIEW DESCRIPTION:

Our safe-lock application is developed with a default password which is (1234) and that's password actually can be changed by the owners easily when they enter it at the first time correctly (default password).

When the owner enters the right password actually, we simulate the safe is opened by simple led (light Emitting Diode).

Also, when the owner wants to close the safe after using it, we simulate that by turning off the led.

Any incorrect attempt an error message will be displayed on the LCD and asks the user to try again, if the attempt is correct also a confirmation message will be displayed and the user will have only three trials if it exceeded system will crash. We constructed our safe-lock Based on Three Main Modes the First one is Closed Mode that Mode represent the state that the user enters Password if the password is wrong an error message appear and the user has only three trials if user's number of trials exceeds 3 trials system will go to infinite loop and user will not be able to access the Safe Locker until be reset from the manufacturing Factory

The second Mode is Open mode (Admin Mode) if the user enters correct password a confirmation message will be displayed and the safe will be opened that simulated by a green LED, the user start in a new mode which is open mode in that mode the owner will be accessible either exit from open mode to the closed mode or to go to another mode which is Reset Mode at that mode the owner can change password, when the user enters new password system will return back owner to the open mode.

That modes help in to organize system and to define well interfacing between LCD and KEYPAD where some buttons has no meaning to use in certain modes for instance in the open mode user is prevented to write digits or using delete or show

passwords buttons only will be accessed in closed or reset mode and for the exit button cannot be accessed in closed mode only will be accessed in open and reset mode. the button for reset which return from reset mode to open mode that button cannot be accessed in either open or closed mode but can be used in reset mode

4. CONSTRAINTS AND ASSUMPTIONS:

- We assumed that the password number must be only Four numbers.
- We assumed the lighting of the LED is an indication of the safe-lock is opened.
- We assumed that if the LED is not lighting up that's mean the safe lock is closed.
- We have mentioned number of trials is 3 if it is exceeded the system will crash.
- We assume base address of the port registers of 00h.

5. TOOLS:

-Emulator 8086(EMU8086).

-Proteus simulation:

- 16*2 LCD.
- 4*4 keypad.
- 64 kilo bytes EEPROM.

- 8086 intel MP.
- 8255 PPI.
- 74HC245 bi direction Buffer for data buffering.
- 74hc343 unidirectional buffer for address latching and buffer.
- Variable resistor for LCD contrasts.
- Clock generator (Frequency oscillator) 1Mhz.
- LED.

6. FUNCTIONAL REQUIREMENTS:

ENUMERATED FUNCTIONAL REQUIREMENTS:

Enter Password	**-System should user to access to the system by entering password**
Change password	-System should provide the owner to change password
Open safe-locker	-System should open the safe locker if the password is correct (simulated by a turning on LED)
Close safe-locker	- System should close the safe locker if owner close the safe locker (simulated by a turning off LED)

ENUMERATED NONFUNCTIONAL REQUIREMENTS:

PERFORMANCE REQUIREMENTS:

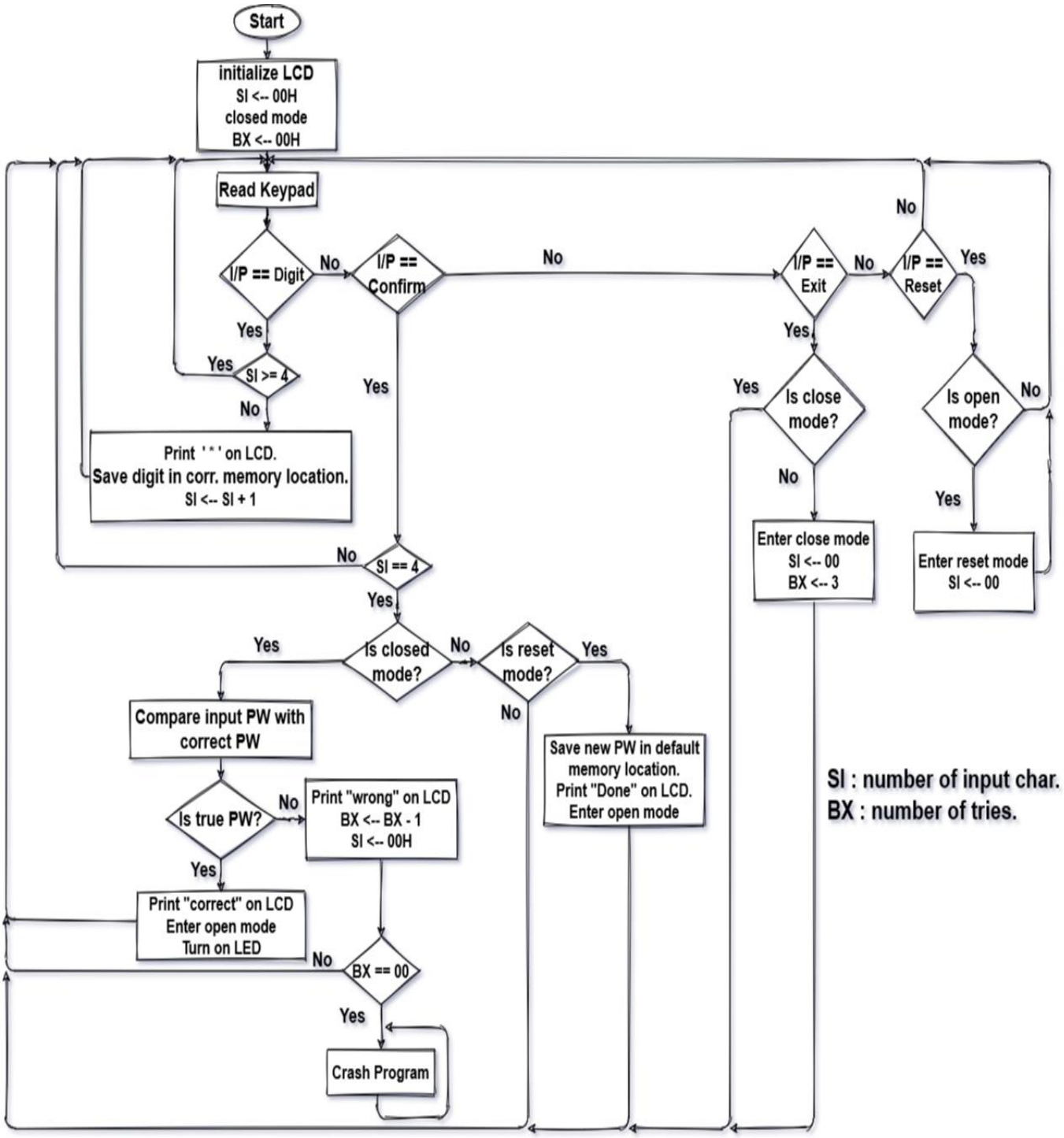
Requirement ID	Requirement Description
1-PR	The system must be respond the business operation in less than 4 seconds for user.
4-PR	The system should response the operation messages to the users within 2 seconds.

SAFETY AND SECURITY REQUIREMENTS:

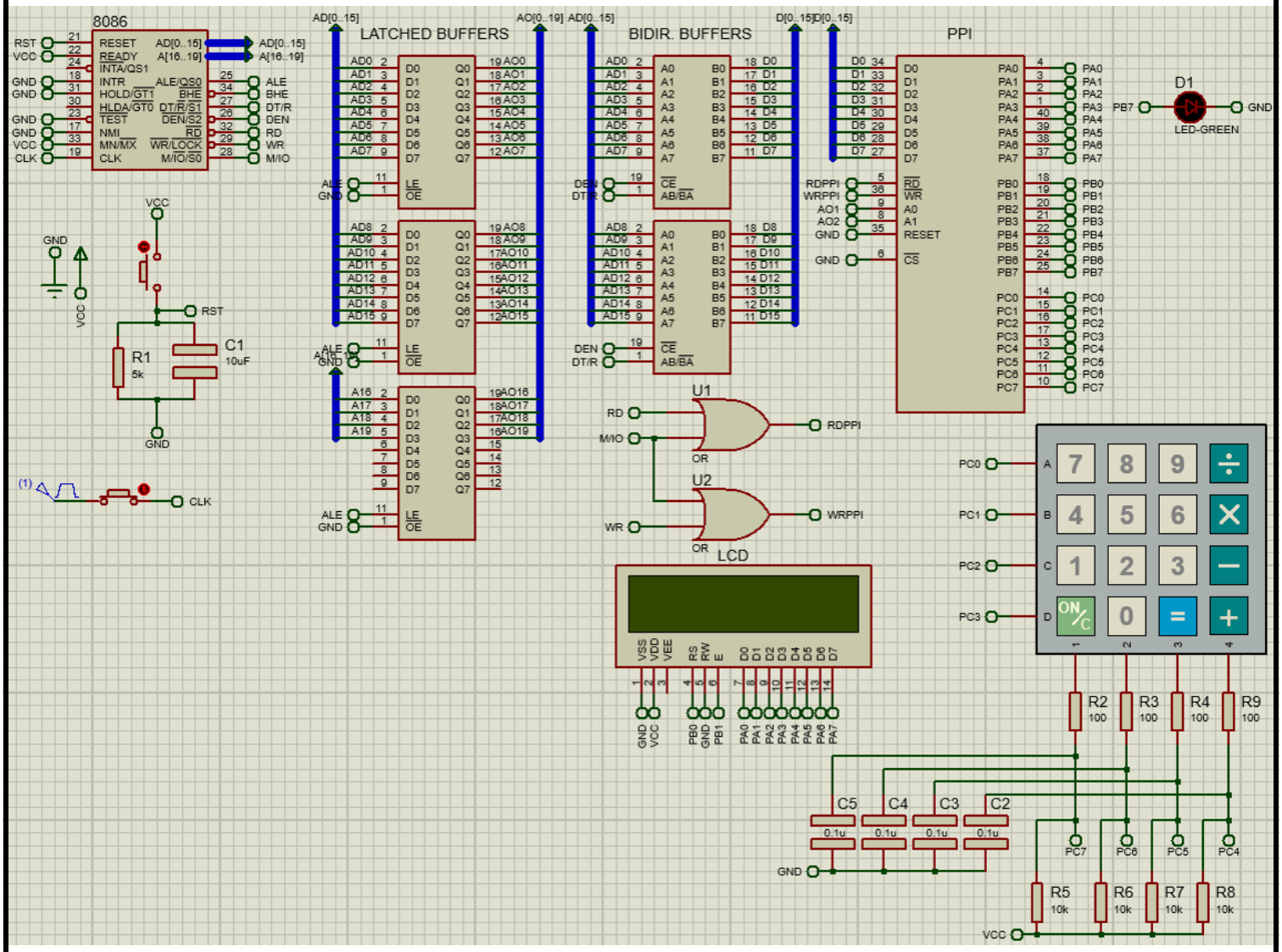
System cannot affect, harm, damage to user. It also cannot damage user's device while accessing the system over a network.

Requirement ID	Requirement Description
1-SSR	The system must handle safe login of the registered user.
2-SSR	The database should be secured to prevent leak or loss of information.

7. FLOW CHART (SAFE-LOCKER):

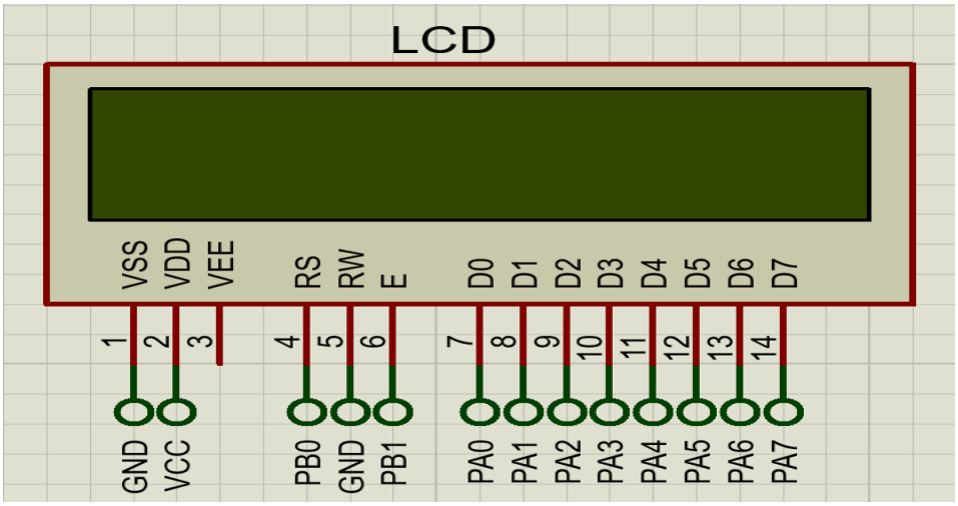


8. SAFE-LOCK (PROTEUS DESIGN):



9. HARDWARE DESCRIPTION:

16*2 LCD (LIQUID CRYSTAL DISPLAY):



16*2 LCD or alphanumeric display we have used is of type 16 * 2 which means 16 character can be written per line and contains two lines.

Actually, LCD is very important in our project as it show the outputs to the user so let's explain some detail about it.

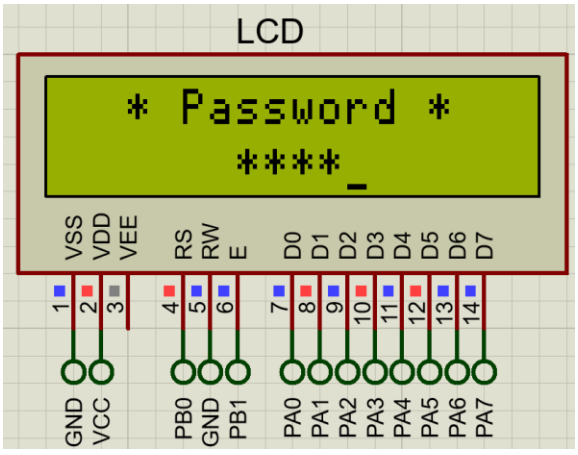
First of all, LCD inside it microcontroller based on the HD44780 microcontroller (Hitachi) and can display two lines with 16 characters each as explained.

The instruction sets of that controller is very useful to access LCD and perform different operations like clear screen and display character and strings, set cursor, show cursor, hide cursor ….and so on

To interface with the LCD we have to connect 16 lines to it VCC (5 volt) , ground the power supply of our LCD , three control lines R/W to read from LCD so connect that line to VCC to write on the LCD we connect it to the Ground that line allow

users to read or write from or to LCD , and the second control line is RS (Register select) that responsible for determine the data on the pins D0 to D7 is it command or data if it was command so it will be loaded on the command register else it will be loaded on data register if RS is grounded means commands held on data bins if it is connected to VCC so data is held and the last control line is Enable which synchronize operation where any operation to be held either writing or reading that control line must be used to write data whatever it is command data or data information that line must be passed to it.

Pin No	Function	Name
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	Vcc
3	Contrast adjustment; through a variable resistor	V_{EE}
4	Selects command register when low; and data register when high	Register Select
5	Low to write to the register; High to read from the register	Read/write
6	Sends data to data pins when a high to low pulse is given	Enable
7	8-bit data pins	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	Backlight V_{CC} (5V)	Led+
16	Backlight Ground (0V)	Led-



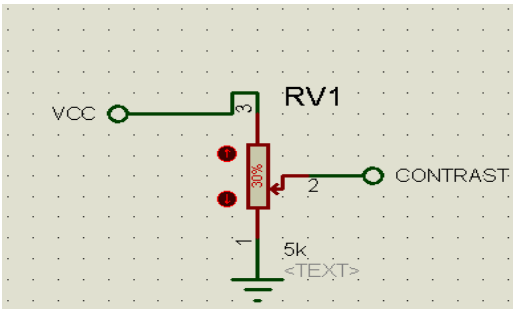
a pulse with duration from 1-5 millisecond so operation will be done accurately to write on the LCD: write operation will be done if enable line is stated from high level to low level with duration as mentioned from 1 – 5 millisecond and to read transition must be from low level to high level.

```
;-------------------------------------
;To oprate LCD in 8-BIT and 2-LINE mode
;-------------------------------------
MOV AL,38H
OUT PORTA,AL

CALL ENABLEOFCOMMAND
;-------------------------
;To turn on cursor of LCD
;-------------------------
MOV AL,0EH
OUT PORTA,AL

CALL ENABLEOFCOMMAND
;-----------------------------
;To print "* PASSWORD *" on LCD.
;-----------------------------
;STR0 DB 'PASSWORD'
CALL PRINTPASSWORD
;-----------------------------------------------
;To set cursor in second line and seventh place.
;-----------------------------------------------
MOV AL,0C6H
OUT PORTA,AL
CALL ENABLEOFCOMMAND
```

There is also VEE pin for the contrast of the screen connected to the slider of the LCD.



From D0 to D7 is connected to an output port from the ports of the PPI used we used in our project port A as output port for those lines and we used port B also output port to control the 3 control lines of the LCD

Now we have described almost the LCD hardware interface now we can talk briefly

About the instruction sets we have used in our project:

-We used 8-bit mode in our project:

So we passes to data pins 38h to set the 8-bit mode instruction

And also we used clear screen instruction by passing 01h to data pins, we used entry mode which means cursor will move from right to left by passing 06h do data bins and we have been used write character to print proper sentences according to the case on the screen.

-And also we used function LCD_SET_CUR which is responsible for setting the position of the cursor according the case and the situation.

And also we have used instructions like hide cursor and clear screen instructions and more and more instructions to achieve our successful software interfacing with the LCD we have introduced most of them.

Note Function LCD_CMD responsible for adjusting enable pin to write data and output on the output ports the instruction code has been sent.

-And the following photos describe all instruction sets in the LCD.

Instruction	Hex	Decimal
Function Set: 8-bit, 1 Line, 5x7 Dots	0x30	48
Function Set: 8-bit, 2 Line, 5x7 Dots	0x38	56
Function Set: 4-bit, 1 Line, 5x7 Dots	0x20	32
Function Set: 4-bit, 2 Line, 5x7 Dots	0x28	40
Entry Mode	0x06	6
Display off Cursor off (clearing display without clearing DDRAM content)	0x08	8
Display on Cursor on	0x0E	14
Display on Cursor off	0x0C	12
Display on Cursor blinking	0x0F	15
Shift entire display left	0x18	24
Shift entire display right	0x1C	30
Move cursor left by one character	0x10	16
Move cursor right by one character	0x14	20
Clear Display (also clear DDRAM content)	0x01	1

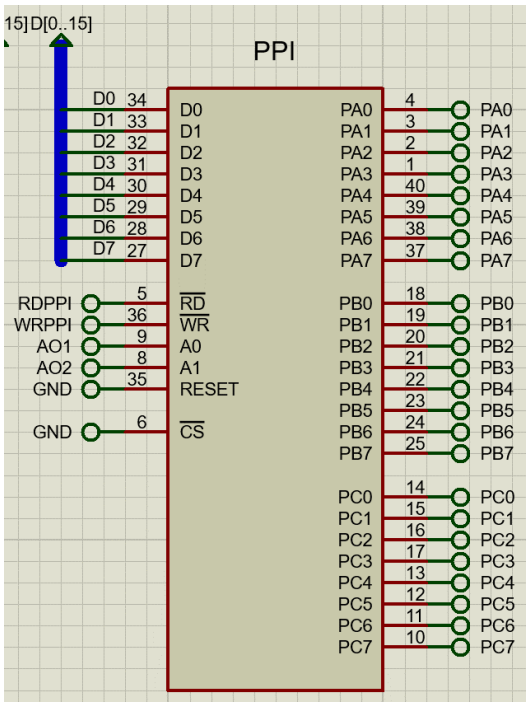
-We constructed our LCD Code Based on Three Main Modes in our LCD the First one is Closed Mode that Mode represent the state that the user enters Password if the password is wrong an error message appear and the user has only three trials if user's number of trials exceeds 3 trials system will go to infinite loop and user will not be able to access the Safe Locker until be reset from the manufacturing Factory

The second Mode is Open mode (Admin Mode) if the user enters correct password a confirmation message will be displayed and the safe will be opened that simulated by a green LED, the user start in a new mode which is open mode in that mode the owner will be accessible either exit from open mode to the closed mode or to go to

another mode which is Reset Mode at that mode the owner can change password, when the user enters new password system will return back owner to the open mode.

That modes help in to organize system and to define well interfacing between LCD and KEYPAD where some buttons has no meaning to use in certain modes for instance in the open mode user is prevented to write digits or using delete or show passwords buttons only will be accessed in closed or reset mode and for the exit button cannot be accessed in closed mode only will be accessed in open and reset mode. the button for reset which return from reset mode to open mode that button cannot be accessed in either open or closed mode but can be used in reset mode.

PROGRAMMABLE PERIPHERAL INTERFACE [PPI] (8255):



To make microprocessor to interface with the external world we have two choices the first one is to use buffers as a ports to be fixed ports either output or input and the second one is to use PPI which is programmable which means any ports in it can be converted from input to output or vice versa.

In our project we selected the second choice which is PPI device that device includes on three main ports which are Porta A and Port B and Port C we can adjust each of them input or output by passing the suitable control word to the control register here we used only one PPI in our project , we passes the control word 88h to the control register to adjust port C the lower part of it from PC0 to PC3 is as an output port and PC4 to PC7 as input port and actually we used port C for interfacing with the Keypad and we adjust port A as an output port and also port B as output port to interface with external device LCD

As we mentioned in LCD section we have used 8-bit mode so we need 8 bits to deal with data pins of LCD so we specialize port A for that purpose and also we mentioned in LCD section we have three control lines which are RS, R/W, E needs to be controlled so we used port B for that purpose to control the three lines moreover we used in our project a LED to simulate opening and closing the safe-lock so we have to interface it with an external port so we specify port B for that purpose also.

The pins of D0 to D7 is output of the data buffering which means the data bus of the microprocessor is connected to those pins

For the pins A0, A1 we apply on them the address lines A1, A2 of microprocessor respectively to them as A0 of microprocessor is used for controlling even Bank of the EEPROM memory from that point we conclude that the step between each port register is 2 since we have assumed that the base address of the port registers is 00h so the base address of the port A is 00H, port B is 02H, port C is 04H, CR is 06H

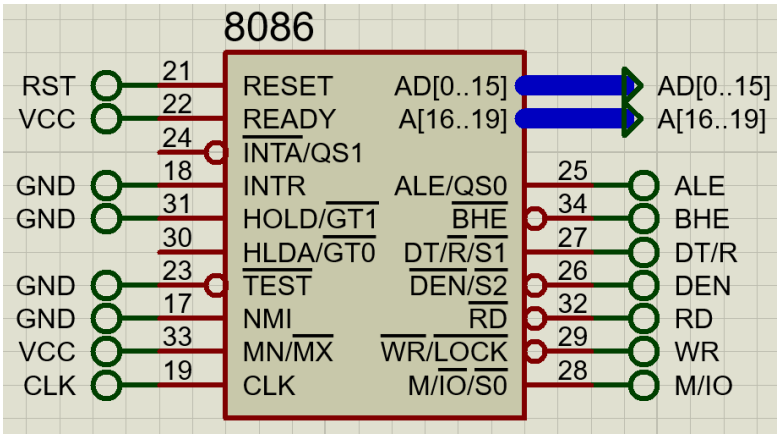
We controlled the read line by an OR gate where the input of the OR gate is M/IO line and Read line of processor

We controlled the write line by an OR gate where the input of the OR gate is M/IO line and write line of processor

Reset pin to reset PPI actually we have grounded it as we did not need to reset it in our project.

The last pin which is CS chip select which is active low will be grounded to be used.

MICROPROCESSOR 8086:

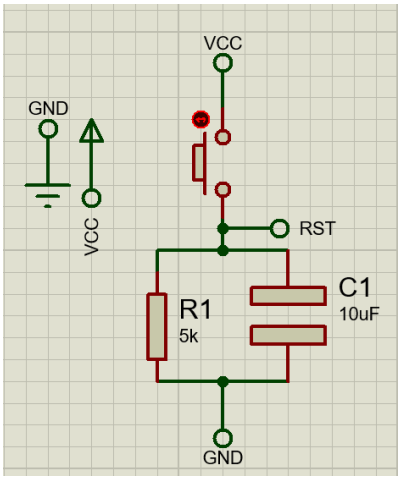


We have used 8086 microprocessor to work on the processor we have to connect several fields like MN/MX we connected that bin to VCC as we want our processor to operate as a single processor without its coprocessor and also we have connected ready bin to VCC.

Interrupt and non-maskable interrupt have connected to ground with Hold and TEST pin.

There are two pins we construct a circuit for each like Reset pin we have connected it to push button with parallel resistor and capacitor so when the push button is pressed the RESET bin is active high so will be activated, MP will be restarted. The second bin is CLK bin that is responsible for synchronizing the processor as MP is a digital System so we need clock for that purpose we construct a simple circuit to generate clocks with a frequency 5 MHz so we have used clock generator with push button and pull up resistor so when the push button is clicked the clock signals will be passed to the processor for that reason we make push button closed and there is another push button used if required to restart CLK bin at any time. We used the pull up resistor to prevent floating signals to be passed.

By just connecting those bins mentioned MP in proteus will be ready for operations



there are another bins used from MP like BHE which is bus high enable which responsible for controlling even Bank in memories and used in general to indicate if there are a valid data on the most significant 8 bits or not.

As address and data lines are multiplexed together so to determine if the bits on that lines are data or address there is ALE address latched Enable which is responsible for telling processer on that multiplexed lines consider address lines is represented and another control is active low DEN which represent if it is activated by low level voltage the data bus is considered on the multiplexed lines

M/IO: this line to indicate which memory mode working on is it isolated memory map or I/O memory map. We used in our project the I/O memory map.

DT/R: this data transfer/Receive line indicates if MP transfer or receive data

R, WR: Read and Write line are indication if Processor read or write

MP includes multiplexed address and data lines from AD0 to AD15

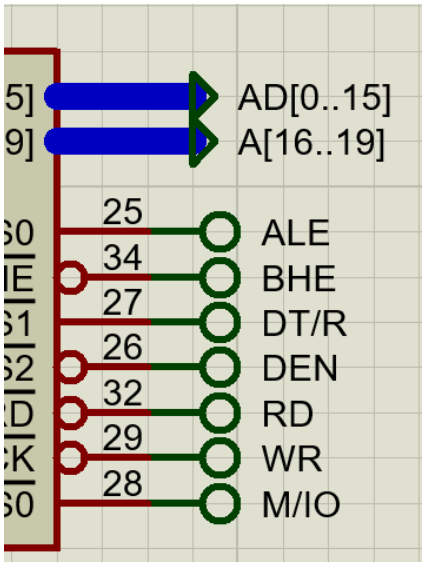
Then three lines multiplexing address and three control lines A16/S3 to A19 /S5.

S3, S4: indication which memory segment is used

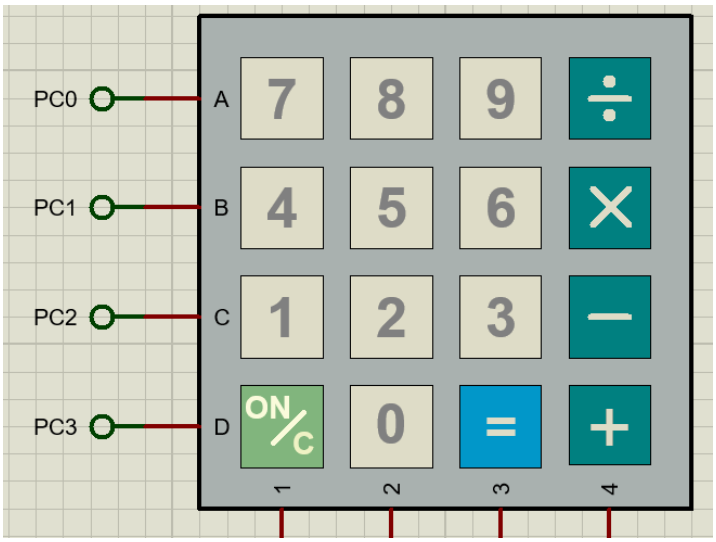
S5: indication of interrupt service

ALE: which is used to activate Address Buffers as we have

Note: we have used the attached Memory that is internal in the MP with size 64 kilo Byte.



4*4 KEYPAD:



we have used 4 * 4 keypad to allow user to enter password we have specified number of buttons for certain purpose like '+' button is for confirming password and 'x' for exit, we have nine digits used. We make MP interface with keypad by the aid of PPI using lower bits of Port C for output and higher bits for input and we have used debouncing concept as keypad is a mechanical switch.

Idea of Working we are shifting zero to the high voltage outputs and if the one of the keys is pressed we can determine the pressed key by the zero voltage at certain row and certain column, take the proper action according to the key pressed now let's talk about debouching circuit in more details.

Buttons Function:

ON/C Button	To show password.
Minus Button	To delete one char.
Multiplication Button	To exit from open mode or reset mode.
Division Button	To confirm the written password.
Plus Button	To reset password.

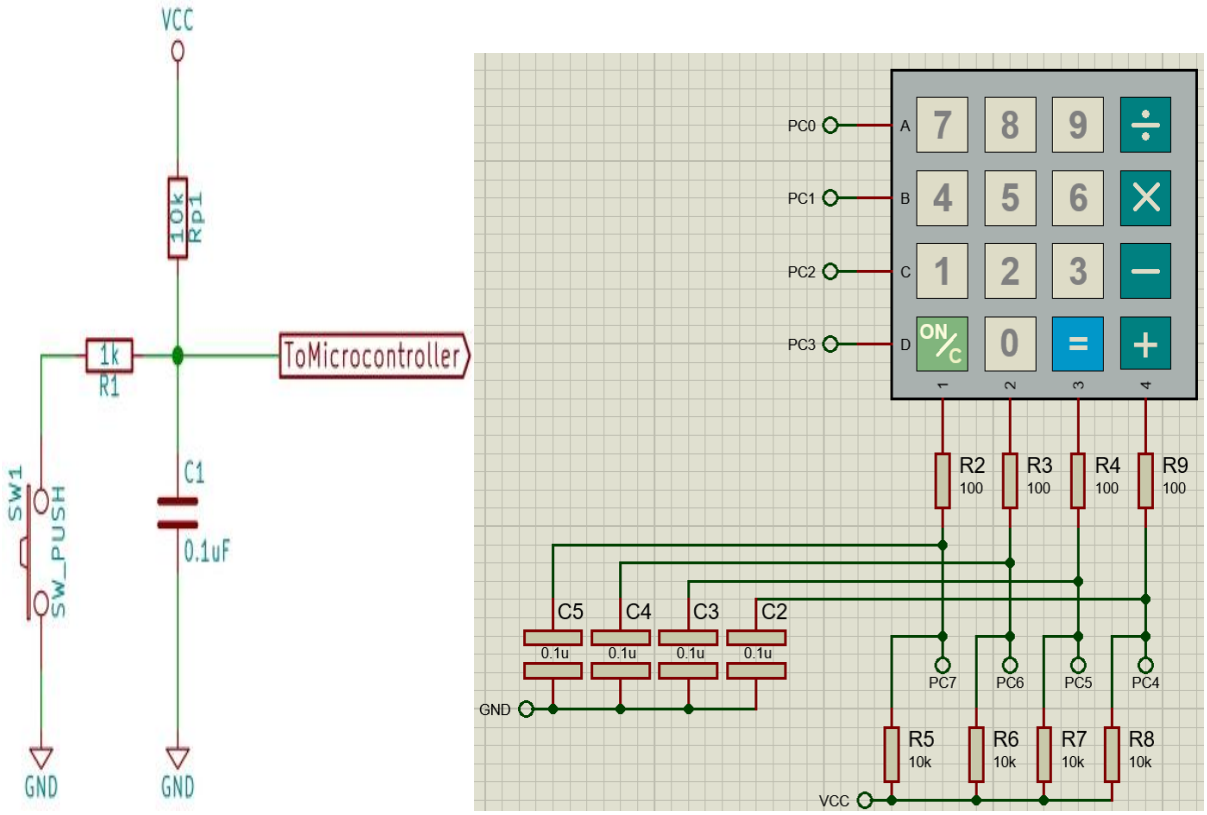
DEBOUNCING:

When you press a pushbutton, it goes from an open circuit to a closed one, and that single transition defines the button "press". But the reason that it's so clear-cut to us is because we perceive things pretty slowly. To a microprocessor, which is able to test a button's state as often as a few thousand times per second, it's not on-off at all.

When you physically press a normal pushbutton, two pieces of metal come into contact with each other. If these two tiny sheets of metal aren't perfectly flat or perfectly aligned (and they're not) then they can make and break contact a few times before being firmly enough squished together that they're always conducting. To a microprocessor the button appears to be pressed many times for extremely short durations, when you think you've just pressed it once. Debouncing is all about making sure that you and the microprocessor agree about when a button push or release event happened.

This problem is easy solved using a capacitor and a resistor sure you should choose the values for both to make the time delay you want t=1/RC the bull up resistor

should be relatively greater than the resistor of the switch, NOW if you press the button before reading directly low the circuit waits until the capacitor discharge.



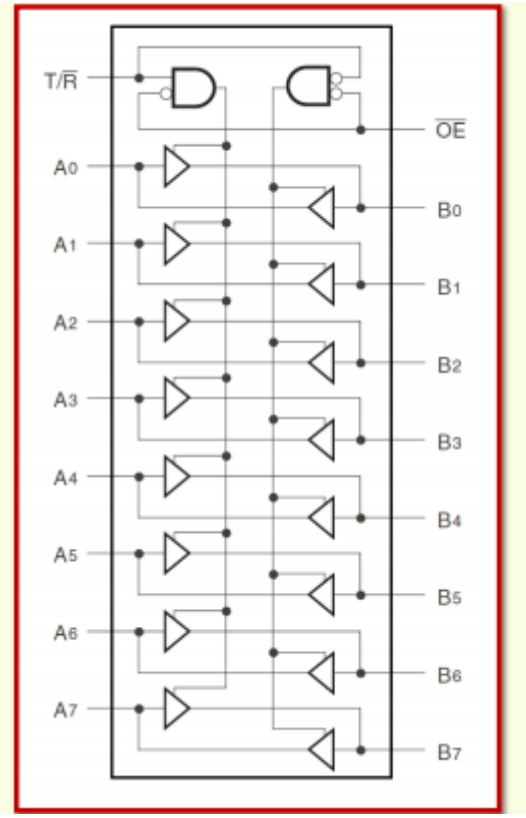
BUFFER 74HC245(DATA BUFFERING):

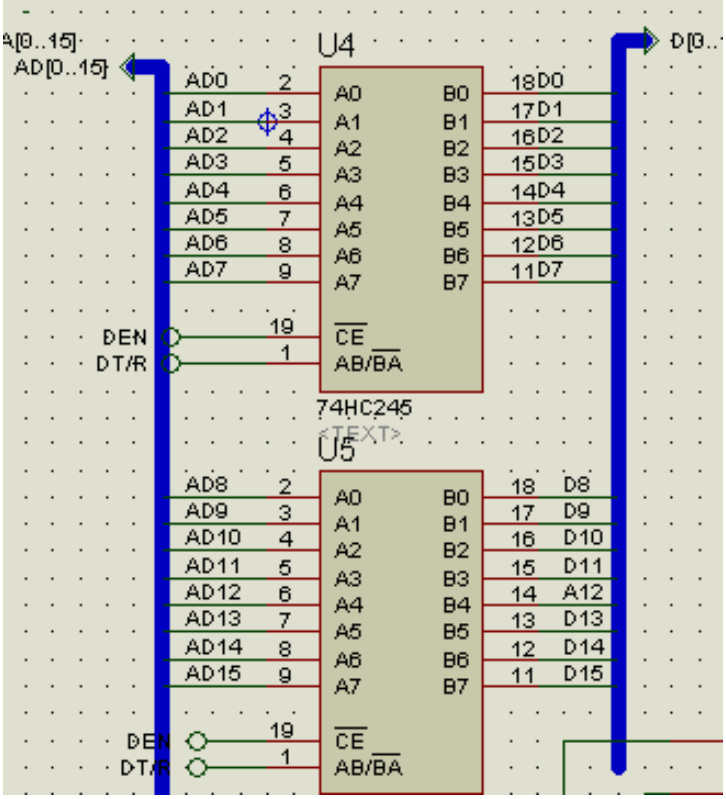
Octal bidirectional bus interface. Used to buffer data bus as data bus is bidirectional and needs to be buffered by bidirectional buffer

Wide power2 to 6V supply range.

Low quiescent maximum (74 HC) current: 80 µA.

When we put low voltage on OE (output enable) and a high voltage on T/R data can propagate from (A to b) and it propagates in the other direction by making T/R pin low.





CHIP 74HCT373(ADDRESS BUFFERING):

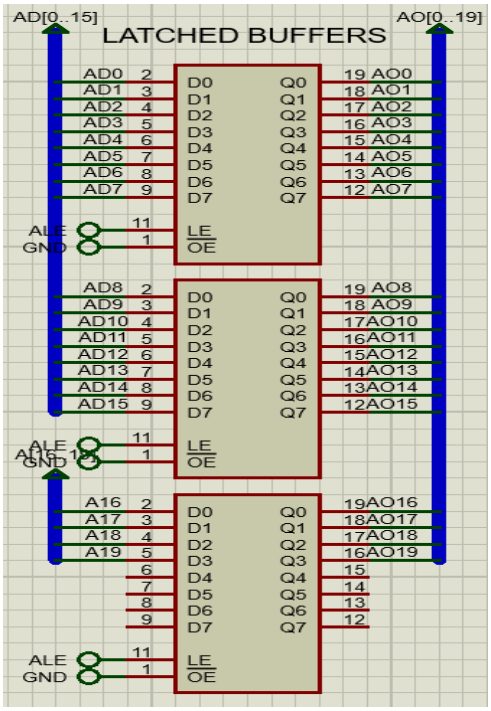
74HCT373 is one directional latch buffer it consists of an octal D-type transparent latch with 3-state outputs which has: latch enable (LE) and output enable (OE) inputs, VCC and GND inputs to bias the chip, 8 input pins and 8 output pins.

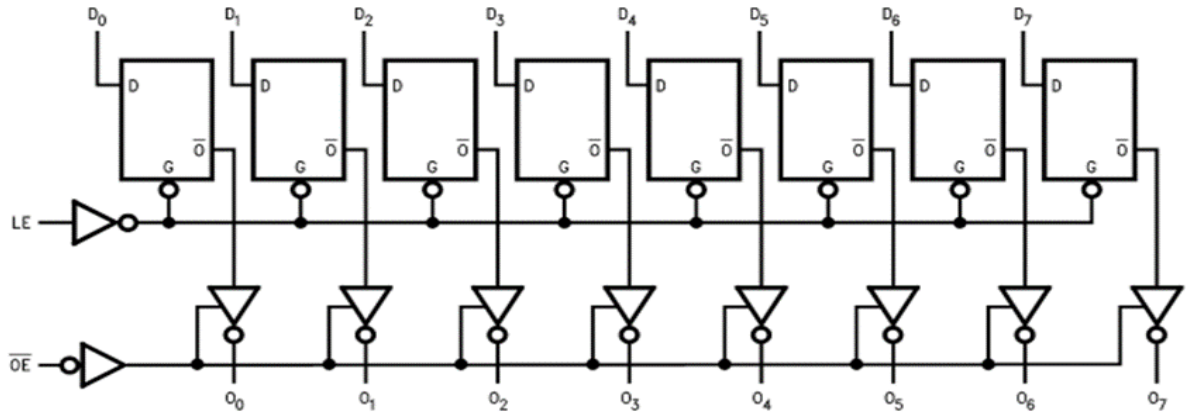
When LE is HIGH, data at the inputs enter the latches. In this condition the latches are transparent, a latch output will change each time its corresponding D-input changes. When LE is LOW the latches store the information that was present at the inputs a set-up time preceding the HIGH-to-LOW transition of LE.

When OE is LOW the information stored in the D latches is passed through out the output pins, a HIGH on OE causes the outputs to assume a high-impedance OFF-state.

It's used to buffer 8086 address bus using 3 chips of the 74HCT373 latches to buffer all 8086's 20 address lines:

- 8086 (ALE) connected to 74HCT373 (LE).
- 74HCT373 (OE) is grounded.
- 8086 (AD_0…AD_{15}, A_{16}...A_{19}) connected to 47HCT373 (input pins).
- 74HCT373 (output pins) (AO_1, AO_2) connected to PPI (A_0, A_1).





10. NOMENCLATURE:

This subsection presents definitions for the terms and acronyms used throughout this SRS as they relate to the subject SLP

Term	Description
SLP	Safe lock Project
SRS	Software Requirement Specification.
CR	Control Register
MP	Microprocessor
PPI	Programmable peripheral interface
LCD	Liquid Crystal Display
LED	Light Emitting Diode.
EEPROM	Electrical Erasable Programmable Read Only Memory
ALE	Address Latch Enable
BHE	Bus High Enable