İSTANBUL OKAN UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

CENG482 Embedded Systems

Final Project

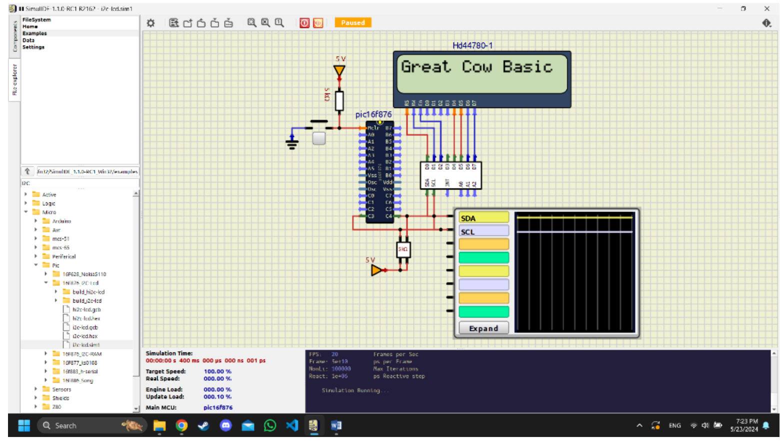
Spring 2023-2024

Prepared By:

Yazid Omran

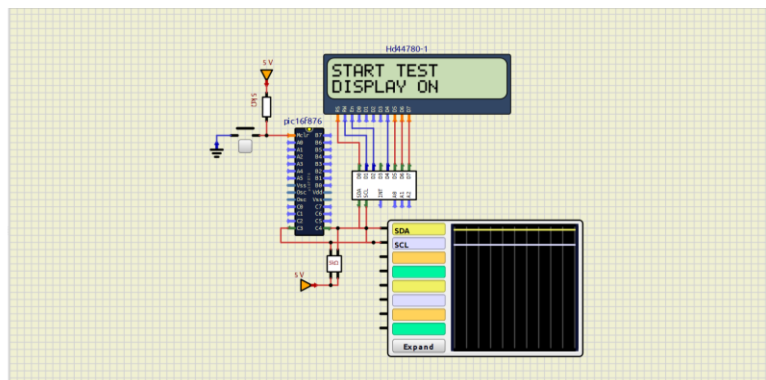
20021315

Q5. When the simulation starts, the program will control an I2C-connected LCD to demonstrate multiple display functionalities. Here's what happens step-by-step:



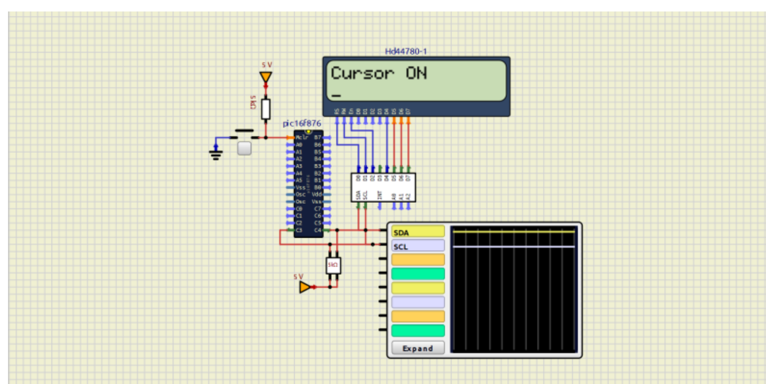
1. Initial Message:

- The LCD displays the message "Great Cow Basic" for 1 second.



2. Start Test:

- The LCD clears its screen (CLS) and waits for 3 seconds.
- The message "START TEST" is displayed.
- The cursor moves to the beginning of the second line, and "DISPLAY ON" is displayed. The display waits for 3 seconds.



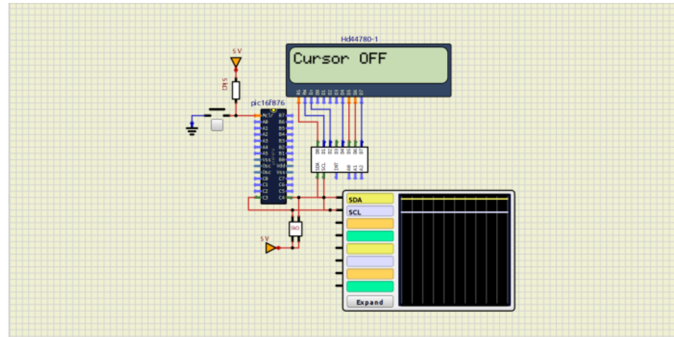
3. Cursor and Flash Demonstrations:

- Cursor ON:

- The screen is cleared, and "Cursor ON" is displayed.
- The cursor is turned on and displayed for 3 seconds.

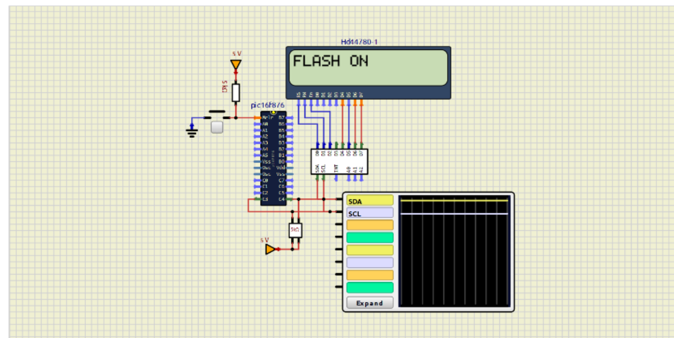
- Cursor OFF:

- The screen is cleared, and "Cursor OFF" is displayed.
- The cursor is turned off and displayed for 3 seconds.



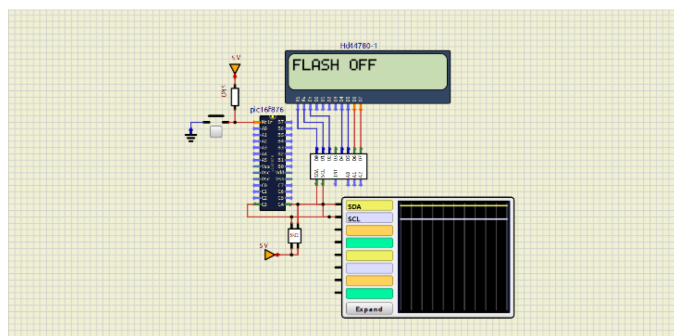
- Flash ON:

- The screen is cleared, and "FLASH ON" is displayed.
- Cursor flashing is turned on and displayed for 3 seconds.



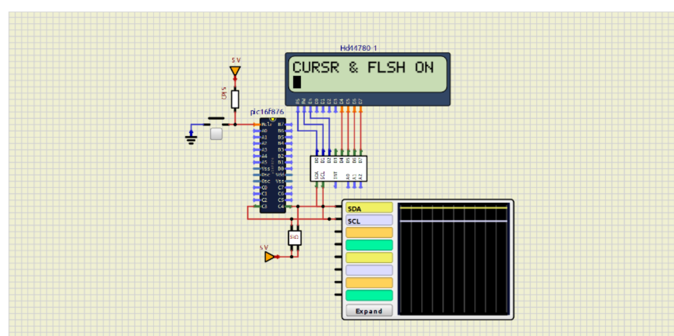
- Flash OFF:

- The screen is cleared, and "FLASH OFF" is displayed.
- Cursor flashing is turned off and displayed for 3 seconds.



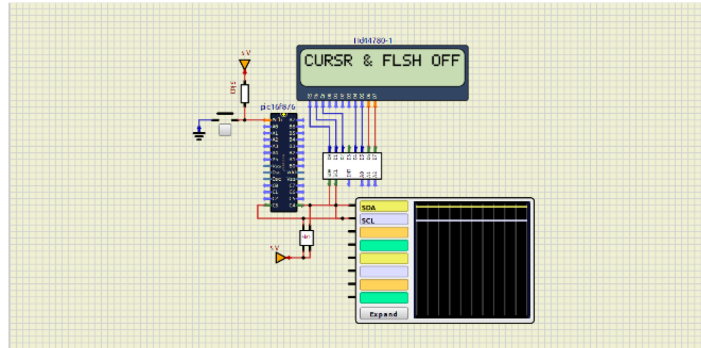
- Cursor & Flash ON:

- The screen is cleared, and "CURSR & FLSH ON" is displayed.
- Both cursor and flashing are turned on and displayed for 3 seconds.



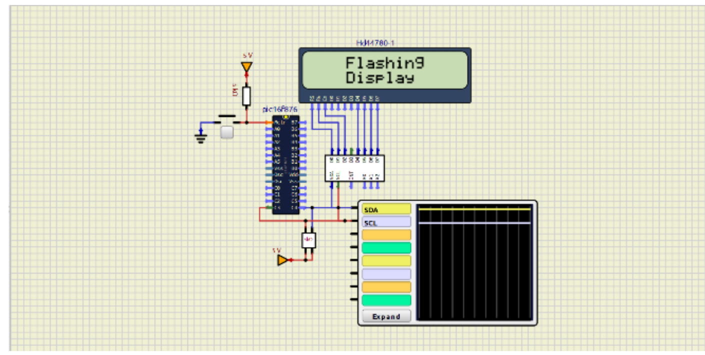
- Cursor & Flash OFF:

- The screen is cleared, and "CURSR & FLSH OFF" is displayed.
- Both cursor and flashing are turned off and displayed for 3 seconds.



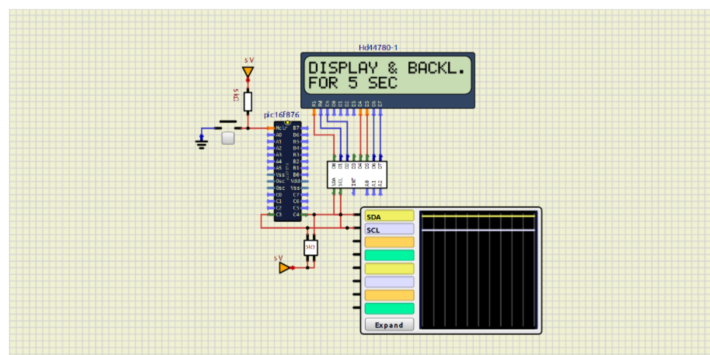
4. Flashing Display:

- The screen is cleared, and "Flashing" is displayed on the first line.
- "Display" is displayed on the second line.
- The display flashes 10 times, alternating every 500 milliseconds.



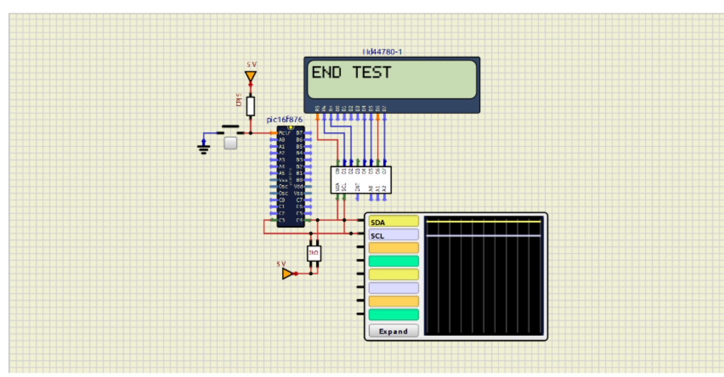
5. Display and Backlight Control:

- The screen is cleared, and "DISPLAY & BACKL." is displayed on the first line.
- "FOR 5 SEC" is displayed on the second line.



- After 2 seconds, the display and backlight are turned off for 5 seconds.

- The backlight is turned back on, the screen is cleared, and the display is turned on.
- "END TEST" is displayed for 3 seconds.



This sequence of actions repeats indefinitely, demonstrating how to control various features of the LCD using I2C communication.

Q6-1. Assembly Commands Functions

BCF: Bit Clear f Syntax: [label] BCF f,b

(Bit 'b' in register 'f' is cleared.)

MOVLW: Move Literal to W Syntax: [label] MOVLW k

(The eight bit literal 'k' is loaded into W register. The don't cares will assemble as 0's.)

MOVWF: Move W to f Syntax: [label] MOVWF f

(Move data from W register to register 'f'.)

CALL: Call subroutine Syntax: [label] CALL k

(Call Subroutine. First, return address (PC+1) is pushed onto the stack. The eleven-bit immediate address is loaded into PC bits. The upper bits of the PC are loaded from PCLATH. CALL is a two-cycle instruction.)

CLRF: Clear f Syntax: [label] CLRF f

(The contents of register 'f' are cleared and the Z bit is set.)

DECFSZ: Decrement f, Skip if 0 Syntax: [label] DECFSZ f,d

(The contents of register 'f' are decremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'. If the result is 1, the next instruction is executed. If the result is 0, then a NOP is executed instead making it a 2TCY instruction.

GOTO: Go to address Syntax: [label] GOTO k

(GOTO is an unconditional branch. The eleven-bit immediate value is loaded into PC bits . The upper bits of PC are loaded from PCLATH. GOTO is a twocycle instruction.)

NOP: No Operation Syntax: [label] NOP

(does nothing and serves as a placeholder.)

Q6-2. Code Explanation

Analyzing the Code Segment

1. Clear Bit in Register:

bcf SYSLCDTEMP,1 - Clears bit 1 of the `SYSLCDTEMP` register.

2. Load Literal and Write to LCD:

movlw 1 - Loads the value 1 into the WREG.

movwf LCDBYTE - Moves the value from WREG to the `LCDBYTE` register.

call LCDNORMALWRITEBYTE - Calls the `LCDNORMALWRITEBYTE` subroutine to write the byte to the LCD.

3. Set Up and Call Delay:

movlw 4 - Loads the value 4 into the WREG.

movwf SysWaitTempMS - Moves the value from WREG to the `SysWaitTempMS` register.

clrf SysWaitTempMS_H - Clears the `SysWaitTempMS_H` register.

call Delay_MS - Calls the `Delay_MS` subroutine to create a delay.

4. Load Another Literal and Write to LCD:

movlw 128 - Loads the value 128 into the WREG.

movwf LCDBYTE - Moves the value from WREG to the `LCDBYTE` register.

call LCDNORMALWRITEBYTE - Calls the `LCDNORMALWRITEBYTE` subroutine to write the byte to the LCD.

5. Microsecond Delay Loop:

movlw 66 - Loads the value 66 into the WREG.

movwf DELAYTEMP - Moves the value from WREG to the `DELAYTEMP` register.

DelayUS1 - Enters a loop where `DELAYTEMP` is decremented. If `DELAYTEMP` is not zero,

decfsz DELAYTEMP,F the loop repeats. If `DELAYTEMP` reaches zero, the next instruction

goto DelayUS1 (`nop`) is executed, effectively creating a delay.

nop - The `nop` instruction does nothing and serves as a placeholder to ensure proper timing.

return - Returns from the subroutine, ending the current function.

Overall Functionality

This code initializes the LCD, writes specific values to it, and incorporates delays to ensure proper timing and data integrity. Here is a summary of what it does:

1. Clears a specific bit in the `SYSLCDTEMP` register, possibly setting or resetting a particular mode or status.
2. Writes the value 1 to the LCD.
3. Sets up a delay using the `Delay_MS` subroutine to wait for 4 milliseconds.
4. Writes the value 128 to the LCD.
5. Creates a precise microsecond delay loop using a decrementing counter.
6. Returns from the subroutine.

The combination of writing values to the LCD and including delays suggests this code is part of a routine to initialize the LCD, send commands or data, and manage timing to ensure the LCD processes each instruction correctly.

Q7.

I2C (Inter-Integrated Circuit) is a synchronous, multi-master, multi-slave serial communication protocol developed by Philips Semiconductor (now NXP Semiconductors) in the 1980s. It is designed for efficient short-distance communication between integrated circuits on the same board. I2C uses two bidirectional lines, the Serial Data Line (SDA) and the Serial Clock Line (SCL), to transmit information between devices. Each device on the bus is addressable by a unique address, allowing multiple devices to coexist without conflict. This simplicity and versatility make I2C a popular choice for connecting peripherals such as sensors, displays, and EEPROMs in various electronic devices.

Two real-world examples of I2C usage include:

1. **Smartphones:** In smartphones, I2C is commonly used to connect components such as touchscreens, camera modules, and gyroscope sensors to the main processor, enabling efficient data transfer and communication between different parts of the device.
2. **Consumer Appliances:** In household appliances like washing machines and microwaves, I2C connects microcontrollers with other components such as temperature sensors, displays, and keypads, facilitating smooth control and monitoring of the appliance's functions.