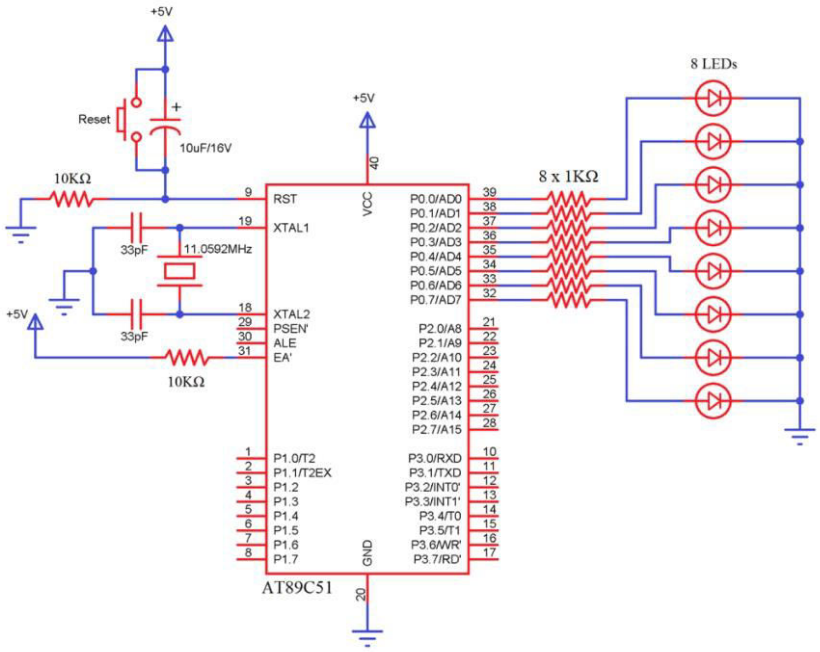
Date: 21/02/2021

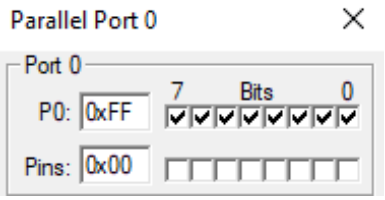
Student's name	Anuj Shah
Roll Number	18104B0024
Name of Professor	Professor Mohit Gujar

Experiment number	3
Experiment title	Interfacing of LED's with 8051 ports
Hardware requirement	-
Software requirement	Keil uVision5

Aim	To write a program that will cause 8 LEDs connected to port-0 of our 8051 microcontroller to blink on-and-off, in an infinite loop.
Theory	

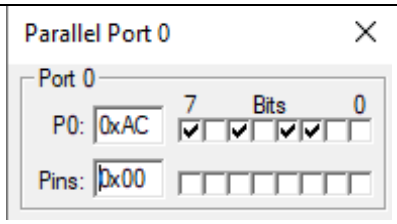
Controlling the voltage of pins

You can control if a pin is high (+5V) or low (0V) using the "MOV P0, #number" command. For example, "MOV P0, #0FFH" will cause all the pins of port 0 to become high:



This is because 0xFF in binary is 1111111. Notice how each high pin corresponds to a 1.

The command "MOV P0,#ACH" will cause:



This is because 0xAC in binary is 10101100.

Subroutines

```
        ORG 0000H
MAIN:   ……..
        ……..
        CALL SUB1       ;subroutine call (store PC on stack)
        CALL SUB2
        ……..
        JMP MAIN

SUB1:   ……..            ;label defines start of subroutine
        ……..
        RET             ;return to line after CALL

SUB2:   ……..
        ……..
        RET
        END
```

A subroutine is always executed with the ACALL instruction

- When the ACALL instruction is executed, the PC register contains the address of the next instruction to be executed
- The PC is saved onto the stack low byte first
- The PC is then loaded with the address of the subroutine
- The subroutine is then executed

The last line of a subroutine is always the RET instruction

- The RET instruction will cause the return address to be popped off the stack and loaded into the PC
- The instruction at the return address is then executed

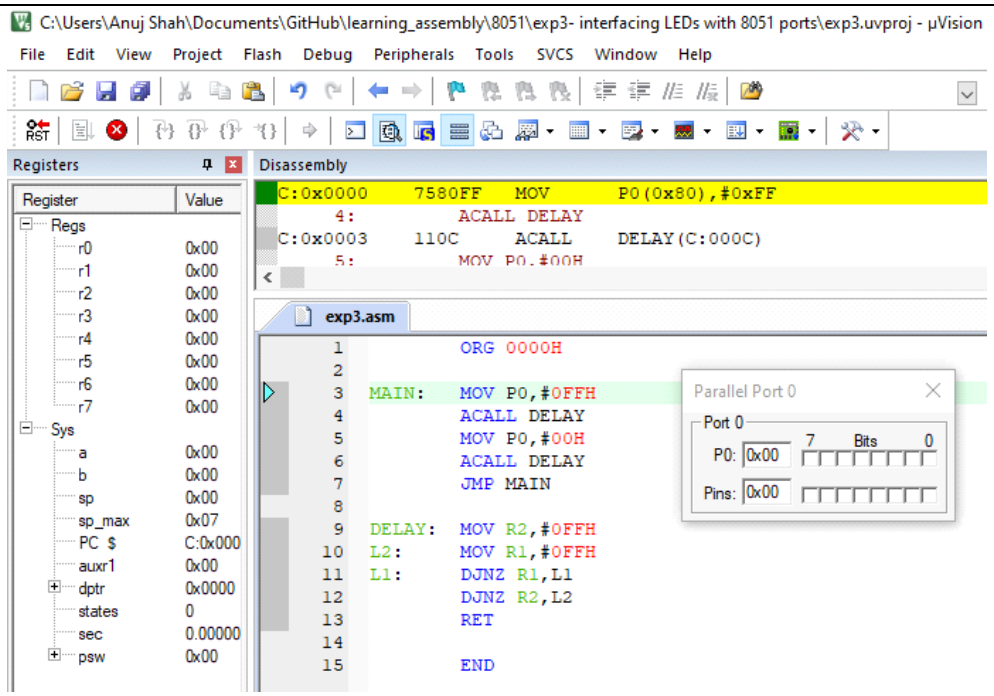
Source:
http://www.polyengineeringtutor.com/8051%20Assembly%20Programming.pdf

Algorithm / Flowchart

MAIN routine:

1. The instruction "MOV P0,#0FFH" causes all the pins of port 0 to become activated. Physically, this means that all the LEDs will turn on.
2. "ACALL DELAY" calls the DELAY subroutine, producing a delay of around 1/10th of a second.
3. The instruction "MOV P0,#00H" causes all the pins of port 0 to become disactivated. Physically, this means that all the LEDs will turn off.
4. "ACALL DELAY" calls the DELAY subroutine, again producing a delay of

	around 1/10th of a second. 5. "JMP MAIN" causes the whole MAIN routine to be executed again. Thus, we see that the MAIN routine produces an infinite loop, in which all the LEDs are on for 0.1 seconds, and then they are off for 0.1 seconds. We thus get a blinking effect. ***DELAY subroutine:*** 1. The instruction "MOV R2,#0FFH" causes the hex number 0xFF (decimal = 255) to be loaded into register R2. 2. The instruction "MOV R1,#0FFH" causes the hex number 0xFF (decimal = 255) to be loaded into register R1. 3. The instruction "DJNZ R1,L1" decrements register R1 (eg. 0xFF to 0xFE), and as long as R1 doesn't become 0x00, it will keep jumping back to this step. 4. The instruction "DJNZ R2,L2" decrements register R2 (eg. 0xFF to 0xFE), and as long as R2 doesn't become 0x00, it will jump to step 2. 5. The "RET" statement let's us leave the DELAY subroutine, and go back to the MAIN routine. The delay subroutine is a nested loop. Both L1 (the inner loop) and L2 (the outer loop) are executed 255 (FF) times. Because each machine cycle is approximately 1 microsecond, and because a DJNZ instruction requires 2 machine cycles, we thus get a total delay of approximately: (2*255*255)*1 microsecond = 130050 micrseconds ~ 130 milliseconds = 0.13 seconds ~ 1/10th of a second
Program	exp3.asm 1 ORG 0000H 2 3 MAIN: MOV P0,#0FFH 4 ACALL DELAY 5 MOV P0,#00H 6 ACALL DELAY 7 JMP MAIN 8 9 DELAY: MOV R2,#0FFH 10 L2: MOV R1,#0FFH 11 L1: DJNZ R1,L1 12 DJNZ R2,L2 13 RET 14 15 END

Results / Output	
Conclusion	We learned a lot of important features of ASM programming in this experiment: • Controlling the voltages of the pins of a port • Calling subroutines using the ACALL instruction, and returning to the MAIN routine using the RET statement • Generating a ~0.1 second delay using a nested loop

Faculty Sign

Grade received