LAB 3-B

I/O PORTS

OBJECTIVES:

- □ To examine the I/O port operation using a simulator.
- □ To trace through a CALL subroutine using a simulator.

MATERIAL:

- □ Atmel Studio
- □ https://lcgamboa.github.io/js/picsimlab.html?../picsimlab_examples/ (Simulator)

WEB SITES:

- □ www.microchip.com for Atmel Studio Software

ACTIVITY 1

Write and assemble a program to toggle all the bits of PORTB, PORTC, and PORTD continuously by sending $55 and $AA to these ports. Put a time delay (between the "on" and "off" states. Then using the simulator, single-step through the program and examine the ports. **Do not single-step through the time delay call.**

```
.equ DELAY_INNER = 100      ; Inner loop count
.equ DELAY_OUTER = 800      ; Outer loop count

delay:
    LDI R18, DELAY_OUTER    ; Load outer loop counter (800)
L1:
    LDI R19, DELAY_INNER    ; Load inner loop counter (100)
L2:
    NOP                     ; Each NOP takes 1 cycle
    DEC R19                 ; Decrement inner loop counter
    BRNE L2                  ; Branch if not zero, takes 2 cycles if branch
                               taken, 1 cycle if not
    DEC R18                 ; Decrement outer loop counter
    BRNE L1                  ; Branch if not zero
    RET                     ; Return from subroutine
```

.INCLUDE "m328pdef.inc"

.EQU DELAY_INNER = 100

.EQU DELAY_OUTER = 800

LAB 3-B

I/O PORTS

```
.ORG 0x00
    ; 1. Setup: Configure PORTB, PORTC, and PORTD as outputs
    LDI R16, 0xFF
    OUT DDRB, R16    ; Set Port B as output
    OUT DDRC, R16    ; Set Port C as output
    OUT DDRD, R16    ; Set Port D as output

START:
    ; 2. Send $55 (01010101) to ports
    LDI R17, 0x55
    OUT PORTB, R17
    OUT PORTC, R17
    OUT PORTD, R17
    RCALL delay      ; Wait

    ; 3. Send $AA (10101010) to ports
    LDI R17, 0xAA
    OUT PORTB, R17
    OUT PORTC, R17
    OUT PORTD, R17
    RCALL delay      ; Wait

    RJMP START       ; Repeat forever

; --- Delay Subroutine ---
delay:
    LDI R18, DELAY_OUTER   ; Load outer loop counter
L1:
    LDI R19, DELAY_INNER   ; Load inner loop counter
L2:
    NOP                ; 1 cycle
    DEC R19              ; 1 cycle
    BRNE L2              ; 2 cycles (taken)
    DEC R18              ; 1 cycle
    BRNE L1              ; 2 cycles (taken)
    RET                ; Return to main program
```

ACTIVITY 2

Examine the registers of the delay subroutine and make the delay shorter or longer by changing the DELAY_INNER or DELAY_OUTTER value.

To reduce delay, decrease the values of DELAY_INNER or DELAY_OUTER.

To increase delay, increase the values instead.

ACTIVITY 3

Using a simulator, write a program to get a byte of data from PORTD (Change the value of PORTD during debugging when getting data from it) and send it to PORTB. Also, give a copy of it to registers R20, R21, and R22. Single-step the program and examine the ports and registers.

```
.INCLUDE "m328pdef.inc"


LDI R16, 0xFF      ; All 1's
OUT DDRB, R16      ; PORTB as output
LDI R16, 0x00      ; All 0's
```

```
OUT DDRD, R16      ; PORTD as input


MAIN_LOOP:

  IN R17, PIND    ; Read data from PORTD pins

  OUT PORTB, R17  ; Send to PORTB


  ; Make copies in R20, R21, R22

  MOV R20, R17

  MOV R21, R17

  MOV R22, R17


   RJMP MAIN_LOOP  ; Repeat forever
```

a) Upon reset, all the ports of the AVR are configured as ___input___ (input, output).

b) To make all the bits of a port an input port we must write _0x00___ hex to DDRx.

c) Write a program to monitor port B.0 continuously. When it becomes low, it sends $55 to PORTB.

```
.INCLUDE "m328pdef.inc"


.ORG 0x0000

 CBI DDRB, 0 ; Set PORTB.0 as Input

 ; Note: Assuming rest of PORTB is output to display $55

 LDI R16, 0xFE ; 11111110 (Bit 0 input, others output)

 OUT DDRB, R16


CHECK_PIN:
```

```
SBIC PINB, 0 ; Skip next line if PINB bit 0 is Cleared (Low)

RJMP CHECK_PIN ; Jump back if bit 0 is Set (High) - Keep waiting


LDI R16, 0x55

OUT PORTB, R16


HERE: RJMP HERE ; Stop or loop here
```

ACTIVITY 4

Test the AVR's ports by using picsimlab for input operation as follows.

a) Connect the pins of PORTx.4-PORTx.7 (PORTD for example) of the AVR to DIP switches. Also connect the pins of PORTy.4-PORTy.7 (e.g. PORTB) to LEDs.

b) Then, write and run a program to get data from PORTx.4-PORTx.7 and send it to PORTy.4-PORTy.7, respectively. Any change of status of the switches connected to PORTx will be instantly reflected on LEDs which are connected to PORTy.

```
.INCLUDE "m328Pdef.inc"


; Setup

LDI R16, 0xF0      ; PORTB: pins 4-7 output, 0-3 input

OUT DDRB, R16

LDI R16, 0x00      ; PORTD: all inputs

OUT DDRD, R16


MAIN_LOOP:

    ; Read PORTD, mask upper nibble

    IN R17, PIND

    ANDI R17, 0xF0  ; Keep only bits 4-7
```

```
; Read current PORTB

IN R18, PORTB

ANDI R18, 0x0F  ; Keep only bits 0-3


; Combine

OR R18, R17


; Output

OUT PORTB, R18


RJMP MAIN_LOOP
```

Note: The main program functions must be in the infinite loop to keep the controller working