LAB 4-A

LOGIC INSTRUCTIONS AND BIT MANIPULATIONS

OBJECTIVES:

- To write a program to perform bit manipulations.

MATERIAL:

- Atmel Studio

WEB SITES:

- www.microchip.com for Atmel Studio Software

ACTIVITY 1

Write a program that copies bits 1, 2, and 3 of PIND to bits 0, 1, and 2 of port B.

```
.INCLUDE "m328pdef.inc"
.ORG 0x0000

    LDI R16, 0x00    ; Set R16 for input
    OUT DDRD, R16    ; Port D as input
    LDI R16, 0x07    ; Pins 0, 1, and 2 as output (binary 00000111)
    OUT DDRB, R16    ; Port B bits 0-2 as output

MAIN:
    IN R20, PIND     ; Read bits from Port D
    ANDI R20, 0x0E
    LSR R20
    OUT PORTB, R20      ; Output to Port B
    RJMP MAIN        ; Repeat continuously
```

LAB 4-A

LOGIC INSTRUCTIONS AND BIT MANIPULATIONS

ACTIVITY 2

Write a program that copies bits 1 and 6 of PINC to bits 0, and 2 of port B, respectively.

```
.INCLUDE "m328pdef.inc"
.ORG 0x0000

    LDI R16, 0x00    ; Prepare for input
    OUT DDRC, R16    ; Set Port C as input
    LDI R16, 0x05    ; Binary 00000101 (Bits 0 and 2)
    OUT DDRB, R16    ; Set Port B pins 0 and 2 as output

MAIN:
    IN  R16, PINC    ; Read Port C
    BST R16, 1       ; Store Bit 1 of R16 into the T-flag
    IN  R17, PORTB   ; Read current Port B state to preserve other bits
    BLD R17, 0       ; Load the T-flag into Bit 0 of R17

    BST R16, 6       ; Store Bit 6 of R16 into the T-flag
    BLD R17, 2       ; Load the T-flag into Bit 2 of R17
    OUT PORTB, R17   ; Send the modified values to Port B

    RJMP MAIN        ; Repeat continuously
```

LAB 4-A

LOGIC INSTRUCTIONS AND BIT MANIPULATIONS

ACTIVITY 3

1) Write a program that inverts bit 3 of port C and sends it to bit 5 of port B.

```
.INCLUDE "m328pdef.inc"
.ORG 0x0000

    LDI R16, 0x00    ; Prepare for input
    OUT DDRC, R16    ; Set Port C as input
    LDI R16, 0x20    ; Binary 00100000 (Bit 5)
    OUT DDRB, R16    ; Set Port B pin 5 as output

MAIN:
    IN  R16, PINC    ; Read physical state of Port C

    ; We use XORI to flip bit 3 (mask 0x08 = 00001000)
    LDI R17, 0x08
    EOR R16, R17     ; Bit 3 is now inverted inside R16

    BST R16, 3       ; Store the inverted bit 3 into the T-flag
    IN  R18, PORTB   ; Read Port B to preserve other pins
    BLD R18, 5       ; Load T-flag into bit 5 of R18
    OUT PORTB, R18   ; Output the result to Port B

    RJMP MAIN        ; Repeat continuously
```

LAB 4-A

LOGIC INSTRUCTIONS AND BIT MANIPULATIONS

2) Find the value in R16 after the following code.

```
LDI   R16, $45
ROR   R16
ROR   R16
ROR   R16
```

R16 = $48 in hex

• Start: $45 (0100 0101)

• ROR 1: 0010 0010 ($22), C=1 (Bit 0 shifted into C)

• ROR 2: 1001 0001 ($91), C=0 (C shifted into Bit 7)

• ROR 3: 0100 1000 ($48), C=1

3) Find the value in R16 after the following code.

```
LDI   R16, $45
ROL   R16
ROL   R16
ROL   R16
```

R16 = $29 in hex

• Start: $45 (0100 0101)

• ROL 1: 1000 1010 ($8A), C=0

• ROL 2: 0001 0100 ($14), C=1

• ROL 3: 0010 1001 ($29), C=0

4) In the absence of the "SWAP Rn" instruction, how does the operation perform?

Create a duplicate of the original register to preserve the data for both operations

-Copy the register

-Mask the lower nibble of one copy , shift left 4 times

-Mask the upper nibble of one copy , shift right 4 times, then OR both together

5) Can the SWAP instruction work on any register?

Yes, SWAP works on any of the 32 registers.

6) Find the value in R2 after the following code.

```
CLR    R2
LDI    R21, $FF
EOR    R2, R21
```

R2 = $FF in hex

• CLR R2 → R2 = $00

• LDI R21, $FF → R21 = $FF

• EOR R2, R21 → $00 XOR $FF = $FF

7) Find the value in A after the following code.

```
CLR    R10
COM    R10
LDI    R16, $AA
EOR    R10, R16
```

R10 = $55 in hex

CLR R10 → R10 = $00

• COM R10 → R10 = $FF (One's Complement of 0 is all 1s)

• LDI R16, $AA → R16 = $AA (1010 1010)

• EOR R10, R16 → $FF XOR $AA

  ◦ 1111 1111 XOR 1010 1010 = 0101 0101 ($55)