

# LAB 4-A

## LOGIC INSTRUCTIONS AND BIT MANIPULATIONS

### OBJECTIVES:

- ② To write a program to perform bit manipulations.

### MATERIAL:

- ② Atmel Studio

### WEB SITES:

- ② [www.microchip.com](http://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
OUT DDRD, R16    ; Set PORTD as Input
LDI R16, 0xFF
OUT DDRB, R16    ; Set PORTB as Output
```

#### MAIN:

```
IN R16, PIND    ; Read from PORTD
ANDI R16, 0x0E  ; Mask bits 1, 2, 3 (Binary 0000 1110)
LSR R16        ; Logical Shift Right (Bit 1->0, 2->1, 3->2)
OUT PORTB, R16  ; Write result to PORTB
RJMP MAIN
```

### 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
OUT DDRC, R16    ; Set PORTC as Input
LDI R16, 0xFF
OUT DDRB, R16    ; Set PORTB as Output
```

#### MAIN:

```
IN R16, PINC    ; Read PORTC
```

# LAB 4-A

## LOGIC INSTRUCTIONS AND BIT MANIPULATIONS

```
; Process Bit 1 -> Bit 0
MOV R17, R16      ; Make copy for bit 1
ANDI R17, 0x02    ; Keep only Bit 1
LSR R17         ; Shift right 1 (Bit 1 -> 0)

; Process Bit 6 -> Bit 2
MOV R18, R16      ; Make copy for bit 6
ANDI R18, 0x40    ; Keep only Bit 6
LSR R18
LSR R18
LSR R18
LSR R18         ; Shift right 4 times (Bit 6 -> 2)

; Combine and Output
OR R17, R18      ; Combine the two bits
OUT PORTB, R17   ; Send to PORTB
RJMP MAIN
```

### 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
; Configure Ports
LDI R16, 0x00
OUT DDRC, R16    ; Set PORTC as Input
LDI R16, 0xFF
OUT DDRB, R16    ; Set PORTB as Output

MAIN:
IN R16, PINC     ; Read Input from PORTC

; 1. Invert Bit 3
LDI R17, 0x08    ; Mask for Bit 3 (0000 1000)
EOR R16, R17    ; Exclusive OR toggles the bit where mask is 1

; 2. Isolate Bit 3
ANDI R16, 0x08   ; Keep only Bit 3, clear others

; 3. Shift to position 5 (3 -> 4 -> 5)
LSL R16         ; Logical Shift Left (Bit 3 moves to 4)
LSL R16         ; Logical Shift Left (Bit 4 moves to 5)

; 4. Output to PORTB
OUT PORTB, R16

RJMP MAIN
```

# 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?

Without SWAP, you must manually exchange the upper and lower nibbles using shifts and logic operations.

- Procedure:
  1. Make a copy of the register.
  2. Mask the lower nibble of one copy (\$0F) and Shift Left 4 times (to move it to the upper half).
  3. Mask the upper nibble of the other copy (\$F0) and Shift Right 4 times (to move it to the lower half).
  4. OR the two results together.

5) Can the SWAP instruction work on any register?

Yes, the SWAP instruction works on any of the 32 General Purpose Registers (R0 – R31).

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

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

## LAB 4-A

### LOGIC INSTRUCTIONS AND BIT MANIPULATIONS

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)