LAB 4-B

DATA TRANSFER

OBJECTIVES:

- To code a program to transfer data from program memory into RAM locations.
- To code a program to transfer data from RAM locations to other RAM locations.
- To experiment with a look-up table.

MATERIAL:

- Atmel Studio

WEB SITES:

- www.microchip.com for Atmel Studio Software

ACTIVITY 1

Write a program to transfer a string of data from **program memory** starting at address $200 to RAM locations starting at $140. Using the simulator, single-step through the program and examine the data transfer and registers.

```
.ORG $0000
    RJMP RESET

; Define Data in Program Memory at $200
.ORG $0200
MY_STRING: .DB "HELLO AVR"   ; 9 characters of data
END_MARKER: .DB 0            ; Null terminator (optional, but good practice)

RESET:
    ; Initialize Pointers
    ; Z Pointer for Source (Program Memory $200)
    LDI ZL, LOW(MY_STRING << 1)
    LDI ZH, HIGH(MY_STRING << 1)

    ; X Pointer for Destination (RAM $140)
    LDI XL, $40
    LDI XH, $01

    ; Counter for how many bytes to move (9 bytes for "HELLO AVR")
    LDI R20, 9

COPY_LOOP:
    ; Transfer Cycle
    LPM R16, Z+      ; Load byte from Program Memory into R16, increment Z
    ST X+, R16       ; Store byte into RAM at X, increment X

    DEC R20          ; Decrement counter
    BRNE COPY_LOOP   ; Repeat until R20 = 0

DONE:
    RJMP DONE        ; Infinite loop
```

LAB 4-B

DATA TRANSFER

ACTIVITY 2

Add the subroutine to the program in Activity 1. After data has been transferred from program memory into RAM, the subroutine function should copy the data from **RAM** locations starting at $140 to **RAM** locations starting at $160. Use single-step through the subroutine and examine the operations.

```
;Define Data in Program Memory at $200
.ORG $0200
MY_STRING: .DB "HELLO AVR"   ; 9 bytes
END_MARKER: .DB 0

RESET:
    ;  Program Memory -> RAM $140
    LDI ZL, LOW(MY_STRING << 1)
    LDI ZH, HIGH(MY_STRING << 1)

    LDI XL, $40     ; X points to RAM $140 (Destination 1)
    LDI XH, $01

    LDI R20, 9      ; Counter for 9 bytes

COPY_PM_TO_RAM:
    LPM R16, Z+     ; Read from Program Memory
    ST X+, R16      ; Store to RAM $140
    DEC R20
    BRNE COPY_PM_TO_RAM

    ; Call Subroutine to Copy RAM -> RAM
    RCALL COPY_RAM_TO_RAM

DONE:
    RJMP DONE       ; End of main program

; Copy RAM $140 -> RAM $160
COPY_RAM_TO_RAM:
    ;Setup Pointers
    LDI YL, $40     ; Y points to Source RAM $140
    LDI YH, $01

    LDI XL, $60     ; X points to Destination RAM $160
    LDI XH, $01

    LDI R20, 9      ; Reset counter for 9 bytes

SUB_LOOP:
    LD R16, Y+      ; Load from Source (RAM $140)
    ST X+, R16      ; Store to Destination (RAM $160)
    DEC R20
    BRNE SUB_LOOP

    RET             ; Return from Subroutine
```

LAB 4-B

DATA TRANSFER

ACTIVITY 3

1. Write a program to calculate y where $y = x^2 + 2x + 9$. Where x is the number between 0 and 9 and the look-up table for x^2 is located at the address $200 of **program memory**. Register R20 keeps the value of x, and at the end of the program R21 should contain the value of y. Use the simulator to change the x value and single-step through the program, examining the changes.

```
.ORG $0000
    RJMP RESET

;  Look-Up Table for x^2 at $200
.ORG $0200
SQ_TABLE: .DB $00, $01, $04, $09, $10, $19, $24, $31, $40, $51
         ;     0^2  1^2  2^2  3^2  4^2  5^2  6^2  7^2  8^2  9^2

RESET:
    ;  Set Input x
    LDI R20, 5      ;  test x = 5 (Target y = 25 + 10 + 9 = 44 or 0x2C)

    ; Initialize Z Pointer to Start of Table
    LDI ZL, LOW(SQ_TABLE << 1)
    LDI ZH, HIGH(SQ_TABLE << 1)

    ;  Add offset x to Z Pointer
    ADD ZL, R20     ; ZL = ZL + x
    CLR R16         ; Use R16 as 0 for carry
    ADC ZH, R16     ; ZH = ZH + 0 + Carry

    ; Load x^2 from Table
    LPM R16, Z      ; R16 now holds x^2 (If x=5, R16 = 25 or 0x19)

    ; Calculate 2x
    MOV R17, R20    ; Copy x to R17
    LSL R17         ; Left Shift = Multiply by 2 (R17 = 2x)

    ;. Calculate y = x^2 + 2x + 9
    ADD R16, R17    ; R16 = x^2 + 2x
    LDI R17, 9      ; Load constant 9
    ADD R16, R17    ; R16 = x^2 + 2x + 9

    ;  Store Result in R21
    MOV R21, R16    ; R21 = y

DONE:
    RJMP DONE
```

LAB 4-B

DATA TRANSFER

2. Explain the difference between the following two instructions:
    a. LPM R16, Z

(Load Program Memory): Loads a byte from Flash (Program) Memory into a register. It is used for reading constants or look-up tables stored in the code space.

    b. LD R16, Z

(Load Indirect): Loads a byte from **SRAM (Data) Memory** into a register. It is used for reading variables or data stored in RAM.

3. Circle the invalid instructions.
    a. LDS R20, 60

    b. LD R30, Z Invalid cannot load data into R30 using Z as the pointer.

    c. LD R25, Z+

    d. LPM R25, Z+4 **Invalid**. LPM instructions cannot have an immediate offset like

4. Explain the difference between the following two instructions:
    a. LDS R20, $40

(Load Direct from Data Space): Go to **SRAM address $40**, find the value stored there, and copy it into R20

    b. LDI R20, $40

(Load Immediate): Simply put the **number $40** (hex for 64) directly into R20. It does not access memory.