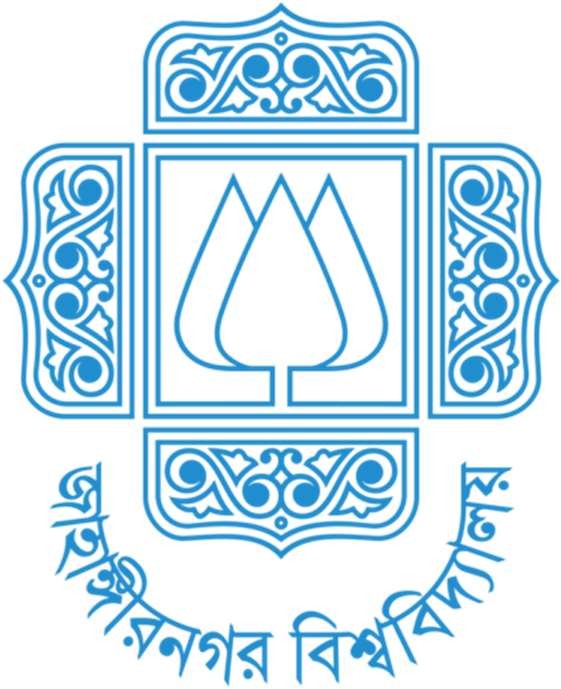
Jahangirnagar University (JU)



**Institute of Information Technology**

**Lab Report-3**

Assembly Language

**Name:** Md. Shakil Hossain **Roll:** 2023

# Experiment 1:

Shift Left Operation (Using an assembly language program).

# Algorithm:

1. Load the binary number into the shift register.
2. Set the shift direction to left.
3. Set the clock signal to start shifting the bits.
4. When the desired number of shifts has been reached, stop the clock signal.
5. The resulting binary number will be stored in the shift register.

**Program Source Code:**

.MODEL SMALL

.STACK

.DATA

A DB 'ENTER NUMBER: $'

C DB 10,13,'ANSWER: $'

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

MOV AH,09

LEA DX,A

INT 21H

MOV AH,1

INT 21H

CALL CONV

MOV BL,AL

ROR BL,4

MOV AH,1

INT 21H

CALL CONV

ADD BL,AL

MOV CL,3

SHL BL,CL

MOV CL,BL

AND BL,0F0H

ROL BL,4

CALL CONV2

MOV AH,09

LEA DX,C

INT 21H

MOV DL,BL

MOV AH,02

INT 21H

MOV BL,CL

AND BL,0FH

CALL CONV2

MOV DL,BL

MOV AH,02

INT 21H

MOV AH,4CH

INT 21H

CONV2 PROC

CMP BL,0aH

JC E

ADD BL,07H

E: ADD BL,30H

RET

ENDP

CONV PROC

CMP AL,40H

JC D

SUB AL,07H

D: SUB AL,30H

RET

ENDP

**Sample Input:** 8A

**Sample Output:** 50



# Experiment 2:

Shift Right Operation (Using an assembly language program)[input as hex]

# Algorithm:

1. Load the binary number into the shift register.
2. Set the shift direction to right.
3. Set the clock signal to start shifting the bits.
4. When the desired number of shifts has been reached, stop the clock signal.
5. The resulting binary number will be stored in the shift register.

**Program Source Code:**

.MODEL SMALL

.STACK

.DATA

A DB 'ENTER NUMBER: $'

C DB 10,13,'ANSWER: $'

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

MOV AH,09

LEA DX,A

INT 21H

MOV AH,1

INT 21H

CALL CONV

MOV BL,AL

ROR BL,4

MOV AH,1

INT 21H

CALL CONV

ADD BL,AL

MOV CL,2

SHR BL,CL

MOV CL,BL

AND BL,0F0H

ROL BL,4

CALL CONV2

MOV AH,09

LEA DX,C

INT 21H

MOV DL,BL

MOV AH,02

INT 21H

MOV BL,CL

AND BL,0FH

CALL CONV2

MOV DL,BL

MOV AH,02

INT 21H

MOV AH,4CH

INT 21H

CONV2 PROC

CMP BL,0aH

JC E

ADD BL,07H

E: ADD BL,30H

RET

ENDP

CONV PROC

CMP AL,40H

JC D

SUB AL,07H

D: SUB AL,30H

RET

ENDP

**Sample Input:** 8A

**Sample Output:** 22



**Experiment 3:**

Rotate Left Operation (Using an assembly language program) [input as hex]

# Algorithm:

1. Load the binary number into the shift register.
2. Set the shift direction to left.
3. Set the clock signal to start shifting the bits.
4. When the desired number of shifts has been reached, stop the clock signal.
5. Save the bit that was shifted out of the left end of the number.
6. Shift the number one more position to the left.
7. Insert the saved bit at the right end of the number.
8. The resulting binary number will be stored in the shift register.

**Program Source Code:**

.MODEL SMALL

.STACK

.DATA

A DB 'ENTER NUMBER: $'

C DB 10,13,'ANSWER: $'

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

MOV AH,09

LEA DX,A

INT 21H

MOV AH,1

INT 21H

CALL CONV

MOV BL,AL

ROR BL,4

MOV AH,1

INT 21H

CALL CONV

ADD BL,AL

MOV CL,3

ROL BL,CL

MOV CL,BL

AND BL,0F0H

ROL BL,4

CALL CONV2

MOV AH,09

LEA DX,C

INT 21H

MOV DL,BL

MOV AH,02

INT 21H

MOV BL,CL

AND BL,0FH

CALL CONV2

MOV DL,BL

MOV AH,02

INT 21H

MOV AH,4CH

INT 21H

CONV2 PROC

CMP BL,0aH

JC E

ADD BL,07H

E: ADD BL,30H

RET

ENDP

CONV PROC

CMP AL,40H

JC D

SUB AL,07H

D: SUB AL,30H

RET

ENDP

**Sample Input:** 8A

# Sample Output: 54



**Experiment 4:**

Rotate Right Operation (Using an assembly language program) [input as hex]

# Algorithm:

1. Load the binary number into the shift register.
2. Set the shift direction to right.
3. Set the clock signal to start shifting the bits.
4. When the desired number of shifts has been reached, stop the clock signal.
5. Save the bit that was shifted out of the right end of the number.
6. Shift the number one more position to the right.
7. Insert the saved bit at the left end of the number.
8. The resulting binary number will be stored in the shift register.

**Program Source Code:**

.MODEL SMALL

.STACK

.DATA

A DB 'ENTER NUMBER: $'

C DB 10,13,'ANSWER: $'

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

MOV AH,09

LEA DX,A

INT 21H

MOV AH,1

INT 21H

CALL CONV

MOV BL,AL

ROR BL,4

MOV AH,1

INT 21H

CALL CONV

ADD BL,AL

MOV CL,3

ROR BL,CL

MOV CL,BL

AND BL,0F0H

ROR BL,4

CALL CONV2

MOV AH,09

LEA DX,C

INT 21H

MOV DL,BL

MOV AH,02

INT 21H

MOV BL,CL

AND BL,0FH

CALL CONV2

MOV DL,BL

MOV AH,02

INT 21H

MOV AH,4CH

INT 21H

CONV2 PROC

CMP BL,0aH

JC E

ADD BL,07H

E: ADD BL,30H

RET

ENDP

CONV PROC

CMP AL,40H

JC D

SUB AL,07H

D: SUB AL,30H

RET

ENDP

**Sample Input:** 8A

# Sample Output: 51

# 