Date:

# EXPERIMENT NO. 6

#### AIM: Perform following operations with basic assembly language programming:

|  |  |  |  |
| --- | --- | --- | --- |
| addition | and | logical left shift | rotate left with carry |
| subtraction | or | logical right shift | rotate left without carry |
| multiplication | xor | arithmetic left shift | rotate right with carry |
| division | not | arithmetic right shift | rotate right without carry |

**Objective 1:**

**Perform addition of two 32-bit numbers i. 1xxx1B64h and ii. 9135F13Ah and store result at memory location 30020h. (consider xxx is last three digits of your enrolment number)**

**Code:**

org 100h

;add 10811B46h and 9135F13Ah

mov ax,1B64h

mov bx,0F13Ah

add ax,bx

mov cx,3000h

mov ds,cx

mov di,0020h

mov [di],ax

mov ax,1081h

mov bx,9135h

adc ax,bx

inc di

inc di

mov [di],ax

ret

**STEP-BY-STEP EXECUTION:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation executed by instruction** | **(Before execution of instruction) Content of to be affected Registers/Memory locations and flags** | **(After execution of instruction) Content of affected Registers/Memory locations and flags** |
| mov ax,1B64h |  |  | Move operand to AX | Ax: 0000h | Ax: 1B64h |
| mov bx,0F13Ah |  |  | Move second operand to BX | Bx: 0000h | Bx: F13Ah |
| add ax,bx |  |  | Add ax and bx and store in ax | AX: 1B64h  CF: 0 | AX: 0C9Eh  CF: 1 |
| mov cx,3000h |  |  | Move to  CX | CX: 001f | CX: 3000 |
| mov ds,cx |  |  | Move cx to ds | Ds: 0700 | DS: 3000 |
| mov di,0020h |  |  | Move value to di | DI: 0000 | DI: 0020 |
| mov [di],ax |  |  | Move ax to [di] | [di]: 0000 |  |
| mov ax,1081h |  |  | Move 1081h to AX | Ax: 0C9E | AX: 1081h |
| mov bx,9135h |  |  | Move 9135h to BX | BX: F13A | BX: 9135 |
| adc ax,bx |  |  | Add with carry ax and bx | AX: 1081h | AX: A1B7 |
| inc di  inc di |  |  | Two times increment di | DI: 20  DI:21 | DI: 21  DI: 22 |
| mov [di],ax |  |  | Move answer to [di] | 30022: 00  30023: 00 |  |

**Objective 2:**

**Perform Subtraction of two 32-bit numbers i. 1xxx1B64h and ii. 9135F13Ah and store result at memory location 30040h. (consider xxx is last three digits of your enrolment number)**

**Code:**

org 100h

;sub 10811B64 AND 9135F13A

mov ax,1B64h

mov bx,0F13Ah

sub ax,bx

mov cx,3000h

mov ds,cx

mov di,0040h

mov [di],ax

mov ax,1081h

mov bx,9135h

sub ax,bx

inc di

inc di

mov [di],ax

ret

**STEP-BY-STEP EXECUTION:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation executed by instruction** | **(Before execution of instruction) Content of to be affected Registers/Memory locations and flags** | **(After execution of instruction) Content of affected Registers/Memory locations and flags** |
| mov ax,1B64h |  |  | 1B64 in AX | AX: 0000 | AX: 1B64 |
| mov bx,0F13Ah |  |  | F13A in BX | BX: 0000 | BX: F13A |
| sub ax,bx |  |  | Subtract ax from bx and store in ax | AX: 1B64 | AX: 2A2A |
| mov cx,3000h |  |  | Move 3000 to CX | CX: 001F | CX: 3000 |
| mov ds,cx |  |  | Move cx to ds | Ds: 0700 | Ds:3000 |
| mov di,0040h |  |  | Give value to di | Di: 0000 | Di: 0040 |
| mov [di],ax |  |  | Move ax to di |  |  |
| mov ax,1081h |  |  | Move 1081 to AX | AX: 2A2A | AX: 1081 |
| mov bx,9135h |  |  | Move 9135 to BX | BX: F13A | BX: 9135 |
| sbb ax,bx |  |  | Subtract with borrow ax from bx | AX: 1081 | AX: 7F4B |
| inc di |  |  | Increment DI |  |  |
| inc di |  |  | Increment DI |  |  |
| mov [di],ax |  |  | Move answer at address of DI |  |  |

**Objective 3:**

**Perform Multiplication of two 16-bit numbers i. Exxxh and ii. A2B3h and store result at memory location 30060h. (consider xxx is last three digits of your enrolment number)**

**Code:**

org 100h

; Multiply E081h with A2b3h

mov ax,0E042h

mov dx,0A2B3h

mul dx

mov cx,3000h

mov ds,cx

mov di,0060h

mov [di],ax

inc di

inc di

mov [di],dx

ret

**STEP-BY-STEP EXECUTION:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation executed by instruction** | **(Before execution of instruction) Content of to be affected Registers/Memory locations and flags** | **(After execution of instruction) Content of affected Registers/Memory locations and flags** |
| mov ax,0E081h |  |  | Move operand to AX | AX: 0000 | AX: E081 |
| mov dx,0A2B3h |  |  | Move op to DX: | DX: 0000 | DX: A2B3 |
| mul dx |  |  | Multiply both values | Dx: A2B3  AX: E081 | DX: 8EAE  AC: 9C33 |
| mov cx,3000h |  |  | Move 3000 to CX | CX: 0017 | Cx: 3000 |
| mov ds,cx |  |  | Move cx to ds | DS: 0700 | DS: 3000 |
| mov di,0060h |  |  | Move 0060 to di | DI: 0000 | DI: 0060 |
| mov [di],ax |  |  | Move ax at the address of di |  |  |
| inc di |  |  | Increment di | Di: 0060 | Di: 0061 |
| inc di |  |  | Increment di | DI: 0061 | Di: 0062 |
| mov [di],dx |  |  | Move the answer at address of DI |  |  |

**Objective 4:**

**Perform Division on Exxxh by 0777h and store result at memory location 30070h. (consider xxx is last three digits of your enrolment number)**

**Code:**

org 100h

;DIV E081h AND 0777h

mov ax,0E081h

mov bx,0777h

div bx

mov cx,3000h

mov ds,cx

mov di,0070h

mov [di],ax

ret

**STEP-BY-STEP EXECUTION:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation executed by instruction** | **(Before execution of instruction) Content of to be affected Registers/Memory locations and flags** | **(After execution of instruction) Content of affected Registers/Memory locations and flags** |
| mov ax,0E081h |  |  | Value E081 move to ax | AX: 0000 | AX: E081 |
| mov bx,0777h |  |  | Move 0777 to bx | BX: 0000 | BX: 0777 |
| div bx |  |  | Division | AX: E081 | AX: 001E |
| mov cx,3000h |  |  | Move 3000 to cx | CX: 0013 | CX: 3000 |
| mov ds,cx |  |  | Move cx to Ds | DS: 0700 | DS: 3000 |
| mov di,0070h |  |  | Move 0070 to di | DI: 0000 | DI: 0070 |
| mov [di],ax |  |  | Move answer to address of di |  |  |

**Objective 5:**

**Perform ANDing on Cxxxh by 00FFh and store result at memory location 30080h. (consider xxx is last three digits of your enrolment number)**

**Code:**

org 100h

;and C081h AND 00FFh

mov ax,0C081h

mov bx,00FFh

and ax,bx

mov cx,3000h

mov ds,cx

mov di,0080h

mov [di],ax

ret

**STEP-BY-STEP EXECUTION:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation executed by instruction** | **(Before execution of instruction) Content of to be affected Registers/Memory locations and flags** | **(After execution of instruction) Content of affected Registers/Memory locations and flags** |
| mov ax,0C081h |  |  | Move C081 to ax | AX: 0000 | AX: C081 |
| mov bx,00FFh |  |  | Move 00FF to bx | BX: 0000 | BX: 00FF |
| and ax,bx |  |  | ANDing | AX: C081 | AX: 0081 |
| mov cx,3000h |  |  | Move 3000 to cx | CX: 0013 | CX: 3000 |
| mov ds,cx |  |  | Mov cx to ds | DS: 0700 | DS: 3000 |
| mov di,0080h |  |  | Move 0080 to ds | DI: 0000 | DI: 0080 |
| mov [di],ax |  |  | Move answer to address of di |  |  |

**Objective 6:**

**Perform ORing on Cxxxh by 00FFh and store result at memory location 300A0h. (consider xxx is last three digits of your enrolment number)**

**Code:**

org 100h

; OR C081h or 00FFh

mov ax,0C081h

mov bx,00FFh

or ax,bx

mov cx,3000h

mov ds,cx

mov di,00A0h

mov [di],ax

ret

**STEP-BY-STEP EXECUTION:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation executed by instruction** | **(Before execution of instruction) Content of to be affected Registers/Memory locations and flags** | **(After execution of instruction) Content of affected Registers/Memory locations and flags** |
| mov ax,0C081h |  |  | Move C081 to ax | AX: 0000 | AX: C081 |
| mov bx,00FFh |  |  | Move 00FF to bs | BX: 0000 | BX: 00FF |
| or ax,bx |  |  | ORing | AX:C081 | AX:C081 |
| mov cx,3000h |  |  | Move 3000 to cx | CX: 0013 | CX: 3000 |
| mov ds,cx |  |  | Move cx to ds | DS: 0700 | DS: 3000 |
| mov di,00A0h |  |  | Move 00A0 to di | DI:0000 | DI:00A0 |
| mov [di],ax |  |  | Move answer to ds |  |  |

**Objective 7:**

**Perform XORing on Cxxxh by 00FFh and store result at memory location 300B0h. (consider xxx is last three digits of your enrolment number)**

**Code:**

org 100h

;C081h XOR 00FFH

mov ax,0C081h

mov bx,00FFh

xor ax,bx

mov cx,3000h

mov ds,dx

mov di,00B0h

mov [di],ax

ret

**STEP-BY-STEP EXECUTION:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation executed by instruction** | **(Before execution of instruction) Content of to be affected Registers/Memory locations and flags** | **(After execution of instruction) Content of affected Registers/Memory locations and flags** |
| mov ax,0C081h |  |  | Move C081 to ax | AX: 0000 | AX:C081 |
| mov bx,00FFh |  |  | Move 00FF to bx | BX: 0000 | BX: 00FF |
| xor ax,bx |  |  | XORing | AX:C081 | AX: C07E |
| mov cx,3000h |  |  | Move 3000 to cx | CX: 0013 | CX: 3000 |
| mov ds,dx |  |  | Move dx to ds | DS: 0700 | DS: 3000 |
| mov di,00B0h |  |  | Move 00B0 to di | DI:0000 | DI: 00B0 |
| mov [di],ax |  |  | Move answer at address |  | 300B0: C07E |

**Objective 8:**

**Perform NOT operation on Cxxxh and store result at memory location 300C0h. (consider xxx is last three digits of your enrolment number)**

**Code:**

org 100h

;NOT OF C081h

mov ax,0C081h

not ax

mov cx,3000h

mov ds,cx

mov di,00C0h

mov [di],ax

ret

**STEP-BY-STEP EXECUTION:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation executed by instruction** | **(Before execution of instruction) Content of to be affected Registers/Memory locations and flags** | **(After execution of instruction) Content of affected Registers/Memory locations and flags** |
| mov ax,0C081h |  |  | Move C081 to ax | AX: 0000 | AX:C081 |
| not ax |  |  | NOT ax | AX: C081 | AX: 3F7E |
| mov cx,3000h |  |  | Move 3000 to cx | CX: 0013 | CX: 3000 |
| mov ds,cx |  |  | Move dx to ds | DS: 0700 | DS: 3000 |
| mov di,00C0h |  |  | Move 00C0 to di | Di: 0000 | DI: 00C0 |
| mov [di],ax |  |  | Move answer at address |  |  |

**Objective 9:**

**Perform logical left shift on Cxxxh and store result at memory location 300D0h. (consider xxx is last three digits of your enrolment number)**

**Code:**

org 100h

;shl C081h

mov ax,0C081h

shl ax ,1

mov cx,3000h

mov ds,cx

mov di,00D0h

mov [di],ax

ret

**STEP-BY-STEP EXECUTION:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation executed by instruction** | **(Before execution of instruction) Content of to be affected Registers/Memory locations and flags** | **(After execution of instruction) Content of affected Registers/Memory locations and flags** |
| mov ax,0C081h |  |  | Move C081 to ax | AX: 0000 | AX: C081 |
| shl ax ,1 |  |  | Shift left AX and 1 | AX: C081 | AX: 8102 |
| mov cx,3000h |  |  | Move 3000 to cx | CX: 0013 | CX: 3000 |
| mov ds,cx |  |  | Move dx to ds | DS: 0700 | DS: 3000 |
| mov di,00D0h |  |  | Move 00D0 to di | DI: 0000 | DI: 00D0 |
| mov [di],ax |  |  | Move answer at address |  |  |

**Objective 10:**

**Perform logical right shift on Cxxxh and store result at memory location 300D0h. (consider xxx is last three digits of your enrolment number)**

**Code:**

org 100h

;SHR C081

mov ax,0C081h

shr ax,1

mov cx,3000h

mov ds,cx

mov di,00D0h

mov [di],ax

ret

**STEP-BY-STEP EXECUTION:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation executed by instruction** | **(Before execution of instruction) Content of to be affected Registers/Memory locations and flags** | **(After execution of instruction) Content of affected Registers/Memory locations and flags** |
| mov ax,0C081h |  |  | Move C081 to ax | AX: 0000 | AX: C081 |
| shr ax,1 |  |  | Shift right AX and 1 | AX:C081 | AX: 6040 |
| mov cx,3000h |  |  | Move 3000 to cx | CX: 0013 | CX: 3000 |
| mov ds,cx |  |  | Move dx to ds | DS: 0700 | DS: 3000 |
| mov di,00D0h |  |  | Move 00D0 to di | Di:0000 | DI: 00D0 |
| mov [di],ax |  |  | Move answer at address |  |  |

**Objective 11:**

**Perform arithmetic left shift on Cxxxh and store result at memory location 300E0h. (consider xxx is last three digits of your enrolment number)**

**Code:**

org 100h

;ashl C081h

mov ax,0C081h

sal ax,1

mov cx,3000h

mov ds,cx

mov di,00E0h

mov [di],ax

ret

**STEP-BY-STEP EXECUTION:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation executed by instruction** | **(Before execution of instruction) Content of to be affected Registers/Memory locations and flags** | **(After execution of instruction) Content of affected Registers/Memory locations and flags** |
| mov ax,0C081h |  |  | Move C081 to ax | AX: 0000 | AX: C081 |
| sal ax,1 |  |  | Arithmetic Shift left AX and 1 | AX: C081 | AX:8102 |
| mov cx,3000h |  |  | Move 3000 to cx | CX: 0013 | CX: 3000 |
| mov ds,cx |  |  | Move dx to ds | DS: 0700 | DS: 3000 |
| mov di,00E0h |  |  | Move 00E0 to di | DI: 0000 | DI: 00E0 |
| mov [di],ax |  |  | Move answer at address |  |  |

**Objective 12:**

**Perform arithmetic right shift on Cxxxh and store result at memory location 300F0h. (consider xxx is last three digits of your enrolment number)**

**Code:**

org 100h

;ashl C081h

mov ax,0C081h

sar ax,1

mov cx,3000h

mov ds,cx

mov di,00F0h

mov [di],ax

ret

**STEP-BY-STEP EXECUTION:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation executed by instruction** | **(Before execution of instruction) Content of to be affected Registers/Memory locations and flags** | **(After execution of instruction) Content of affected Registers/Memory locations and flags** |
| mov ax,0C081h |  |  | Move C081 to ax | AX: 0000 | AX: C081 |
| sar ax,1 |  |  | Shift right AX and 1 | AX: C081 | AX: E040 |
| mov cx,3000h |  |  | Move 3000 to cx | CX: 0013 | CX: 3000 |
| mov ds,cx |  |  | Move dx to ds | DS: 0700 | DS: 3000 |
| mov di,00F0h |  |  | Move 00F0 to di | DI: 0000 | DI: 00F0 |
| mov [di],ax |  |  | Move answer at address |  |  |

**Objective 13:**

**Perform rotate left shift with carry on Cxxxh and store result at memory location 30100h. (consider xxx is last three digits of your enrolment number)**

**Code:**

org 100h

;ashl C081h

mov ax,0C081h

rcl ax,1

mov cx,3000h

mov ds,cx

mov di,0100h

mov [di],ax

ret

**STEP-BY-STEP EXECUTION:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation executed by instruction** | **(Before execution of instruction) Content of to be affected Registers/Memory locations and flags** | **(After execution of instruction) Content of affected Registers/Memory locations and flags** |
| mov ax,0C081h |  |  | Move C081 to ax | AX: 0000 | AX: C081 |
| rcl ax,1 |  |  | left shift with carryAX and 1 | AX:C081 | AX:6040 |
| mov cx,3000h |  |  | Move 3000 to cx | CX: 0013 | CX: 3000 |
| mov ds,cx |  |  | Move dx to ds | DS: 0700 | DS: 3000 |
| mov di,0100h |  |  | Move 00D0 to di | Di: 0000 | DI: 0100 |
| mov [di],ax |  |  | Move answer at address |  |  |

**Objective 14:**

**Perform rotate left shift without carry on Cxxxh and store result at memory location 30110h. (consider xxx is last three digits of your enrolment number)**

**Code:**

org 100h

;ashl C081h

mov ax,0C081h

rol ax,1

mov cx,3000h

mov ds,cx

mov di,0110h

mov [di],ax

ret

**STEP-BY-STEP EXECUTION:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation executed by instruction** | **(Before execution of instruction) Content of to be affected Registers/Memory locations and flags** | **(After execution of instruction) Content of affected Registers/Memory locations and flags** |
| mov ax,0C081h |  |  | Move C081 to ax | AX: 0000 | AX: C081 |
| rol ax,1 |  |  | left shift with carryAX and 1 | AX: C081 | AX:8103 |
| mov cx,3000h |  |  | Move 3000 to cx | CX: 0013 | CX: 3000 |
| mov ds,cx |  |  | Move dx to ds | DS: 0700 | DS: 3000 |
| mov di,0110h |  |  | Move 0110 to di | DI: 0000 | Di: 0110 |
| mov [di],ax |  |  | Move answer at address |  |  |

**Objective 15:**

**Perform rotate right shift with carry on Cxxxh and store result at memory location 30120h. (consider xxx is last three digits of your enrolment number)**

**Code:**

org 100h

;ashl C081h

mov ax,0C081h

ror ax,1

mov cx,3000h

mov ds,cx

mov di,0120h

mov [di],ax

ret

**STEP-BY-STEP EXECUTION:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation executed by instruction** | **(Before execution of instruction) Content of to be affected Registers/Memory locations and flags** | **(After execution of instruction) Content of affected Registers/Memory locations and flags** |
| mov ax,0C081h |  |  | Move C081 to ax | AX: 0000 | AX: C081 |
| ror ax,1 |  |  | right shift with carryAX and 1 | AX:C081 | AX: E040 |
| mov cx,3000h |  |  | Move 3000 to cx | CX: 0013 | CX: 3000 |
| mov ds,cx |  |  | Move dx to ds | DS: 0700 | DS: 3000 |
| mov di,0120h |  |  | Move 0120 to di | DI: 0000 | DI:0120 |
| mov [di],ax |  |  | Move answer at address |  |  |

**Objective 16:**

**Perform rotate right shift without carry on Cxxxh and store result at memory location 30130h. (consider xxx is last three digits of your enrolment number)**

**Code:**

org 100h

;ashl C081h

mov ax,0C081h

rcr ax,1

mov cx,3000h

mov ds,cx

mov di,0130h

mov [di],ax

ret

**STEP-BY-STEP EXECUTION:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation executed by instruction** | **(Before execution of instruction) Content of to be affected Registers/Memory locations and flags** | **(After execution of instruction) Content of affected Registers/Memory locations and flags** |
| mov ax,0C081h |  |  | Move C081 to ax | AX: 0000 | AX: C081 |
| rcr ax,1 |  |  | right shift without carryAX | AX: C081 | AX: 6040 |
| mov cx,3000h |  |  | Move 3000 to cx | CX: 0013 | CX: 3000 |
| mov ds,cx |  |  | Move dx to ds | DS: 0700 | DS: 3000 |
| mov di,0130h |  |  | Move 0130 to di | Di:0000 | Di:0130 |
| mov [di],ax |  |  | Move answer at address |  |  |

**CONCLUSION:**