EXPERIMENT NO. 6

Date: 16/9/24

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 30080h. (consider xxx is last three digits of your enrolment number)

Code:

org 100h

mov ax,3000H mov ds,ax mov ax,1004H mov bx,1b64H mov cx,9135H

mov dx,0f13aH add bx,dx

adc ax,cx mov [80h],ax mov [82h],bx ret

STEP-BY-STEP EXECUTION:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation xecuted by nstruction** | **Before execution f instruction) ontent of to be ffected egisters/Memory ocations and flag** | **After execution of instruction) Content of affected Registers/Memor y locations and lags** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| **mov ax,3 000 H** |  |  | **Loa d 300**  **0H**  **into AX** |  |  |
| **mov ds,a x** |  |  | **Mo ve AX**  **to DS** |  |  |
| **mov ax,1 004 H** |  |  | **Loa d 103**  **3H**  **into AX** |  |  |
| **mov bx,1 b64 H** |  |  | **Loa d 1B6**  **4H**  **into BX** |  |  |
| **mov cx,9 135 H** |  |  | **Loa d 913**  **5H**  **into CX** |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **mov dx,0 f13a H** |  |  | **Loa d 0F1**  **3A H**  **into DX** |  |  |
| **add bx,d x** |  |  | **Add DX**  **to BX** |  |  |
| **adc ax,c x** |  |  | **Add CX**  **with carr y to AX** |  |  |
| **mov [80h**  **],ax** |  |  | **Stor e AX**  **at me mor y [80**  **H]** |  |  |
| **mov [82h**  **],bx** |  |  | **Stor e BX**  **at me mor y [82**  **H]** |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ret** |  |  | **Ret urn fro m pro ced ur** |  |  |

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

mov ax,3000H mov ds,ax mov ax,1004H mov bx,1b64H mov cx,9135H

mov dx,0f13aH sub bx,dx

sbb ax,cx mov [40h],ax mov [42h],bx ret

STEP-BY-STEP EXECUTION:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation xecuted by nstruction** | **Before execution f instruction) ontent of to be ffected egisters/Memory ocations and flag** | **After execution of instruction) Content of affected Registers/Memor y locations and lags** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **mov ax,3 000 H** |  |  | **Loa d 300**  **0H**  **into AX** |  |  |
| **mov ds,a x** |  |  | **Mo ve AX**  **to DS** |  |  |
| **mov ax,1 004 H** |  |  | **Loa d 103**  **3H**  **into AX** |  |  |
| **mov bx,1 b64 H** |  |  | **Loa d 913**  **5H**  **into CX** |  |  |
| **mov cx,9 135 H** |  |  | **Loa d 0F1**  **3A H**  **into DX** |  |  |
| **mov dx,0 f13a H** |  |  | **Loa d 0F1**  **3A H**  **into DX** |  |  |



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **sub bx,d x** |  |  | **Sub trac t DX**  **fro m BX**  **sbb ax,c**  **x** |  |  |
| **sbb ax,c x** |  |  | **Sub trac t CX**  **with carr y fro m**  **AX** |  |  |
| **mov [40h**  **],ax** |  |  | **Stor e AX**  **at me mor y [40**  **H]** |  |  |
| **mov [42h**  **],bx** |  |  | **Stor e BX**  **at me mor y [42 H** |  |  |
| **ret** |  |  | **Ret urn fro m pro ced ure** |  |  |

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

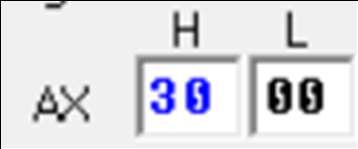
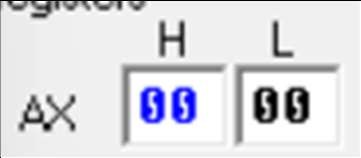
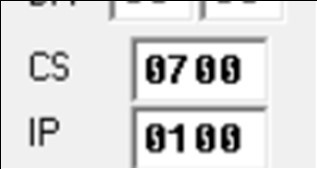
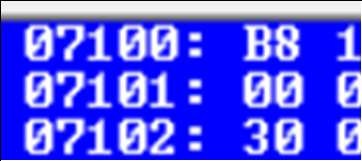
mov ax,3000H mov ds,ax

mov ax,0E004H mov bx,0A2B3H mul ax

mov [60h],bx ret

STEP-BY-STEP EXECUTION:

{Remove this comment before printing: edit following section as per requirement}



**Operation xecuted by nstruction**

**Before execution After execution f instruction) of instruction) ontent of to be Content of ffected affected**

**egisters/MemoryRegisters/Memor ocations and flagy locations and**

**lags**

**Loa d 300**

**0H**

**into AX**

**mov ax,300 0H**

**Value in Instruction Pointer**

**Screenshot of the Instruction in program memory**

**Instruction in Assembly Language**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **mov ds,ax** |  |  | **Mo ve AX**  **to DS** |  |  |
| **mov ax,0E0 04H** |  |  | **Loa d 0E0**  **04H**  **into AX** |  |  |
| **mov bx,0A2 B3 H** |  |  | **Loa d 0A2 B3 H**  **into BX** |  |  |
| **Mul ax** |  |  | **Mul tipl y AX**  **by BX** |  |  |
| **mov [60h],a**  **x** |  |  | **Stor e BX**  **at me mor y [60**  **H]** |  |  |
| **ret** |  |  | **Ret urn fro m pro ced ure** |  |  |



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

mov ax,3000H mov ds,ax

mov ax,0E004H mov bx,0777H mul ax

mov [70h],bx ret

STEP-BY-STEP EXECUTION:

{Remove this comment before printing: edit following section as per requirement}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation xecuted by nstruction** | **Before execution f instruction) ontent of to be ffected egisters/Memory ocations and flag** | **After execution of instruction) Content of affected Registers/Memor y locations and lags** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **mov ax,3 000 H** |  |  | **Loa d 300**  **0 H**  **into AX** |  |  |
| **mov ds,a x** |  |  | **Mo ve AX**  **to DS** |  |  |
| **mov ax,0 E004H** |  |  | **Loa d 0E0**  **04H**  **into AX** |  |  |
| **mov bx,0 777 H** |  |  | **Loa d 077**  **7H**  **into BX** |  |  |
| **Div ax** |  |  | **Divi de AX**  **by BX** |  |  |
| **mov [70h**  **],ax** |  |  | **Stor e BX**  **at me mor y [70**  **H]** |  |  |



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ret** |  |  | **Ret urn fro m pro ced ure** |  |  |

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

mov ax,3000H mov ds,ax

mov ax,0C004H mov bx,00FFH and ax,bx

mov [80h],ax ret

STEP-BY-STEP EXECUTION:

{Remove this comment before printing: edit following section as per requirement}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation xecuted by nstruction** | **Before execution f instruction) ontent of to be ffected egisters/Memory ocations and flag** | **After execution of instruction) Content of affected Registers/Memor y locations and lags** |
| **mov ax,3 000 H** |  |  | **Loa d 300**  **0 H**  **into AX** |  |  |
| **mov ds,a x** |  |  | **Mo ve AX**  **to DS** |  |  |
| **mov ax,0 C03 3H** |  |  | **Loa d 0C0**  **04H**  **into AX** |  |  |
| **mov bx,0 0FF H** |  |  | **Loa d 00F FH**  **into BX** |  |  |
| **And ax,b x** |  |  | **Perf orm AN D**  **bet wee n AX**  **and bx** |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **mov [80h**  **],ax** |  |  | **Stor e AX**  **at me mor y [80**  **H]** |  |  |
| **ret** |  |  | Ret urn fro m pro ced ure |  |  |

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

mov ax,3000H mov ds,ax

mov ax,0C004H mov bx,00FFH or ax,bx

mov [0A0h],ax ret

STEP-BY-STEP EXECUTION:

{Remove this comment before printing: edit following section as per requirement}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation xecuted by nstruction** | **Before execution f instruction) ontent of to be ffected egisters/Memory ocations and flag** | **After execution of instruction) Content of affected Registers/Memor y locations and lags** |
| **mov ax,3 000 H** |  |  | **Loa d 300**  **0H**  **into AX** |  |  |
| **mov ds,a x** |  |  | **Mo ve AX**  **to DS** |  |  |
| **mov ax,0 C03 3H** |  |  | **Loa d 0C0**  **04H**  **into AX** |  |  |
| **mov bx,0 0FF H** |  |  | **Loa d 00F FH**  **into BX** |  |  |
| **ora x,bx** |  |  | **Perf orm OR**  **bet wee n AX**  **and BX** |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **mov [0A 0h],**  **ax** |  |  | **Stor e AX**  **at me mor y [0A**  **0H]** |  |  |
| **ret** |  |  | **Ret urn fro m pro ced ure** |  |  |

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

mov ax,3000H mov ds,ax

mov ax,0C004H mov bx,00FFH

xor ax,bx

mov [0B0h],ax ret

STEP-BY-STEP EXECUTION:

{Remove this comment before printing: edit following section as per requirement}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation xecuted by nstruction** | **Before execution f instruction) ontent of to be ffected egisters/Memory ocations and flag** | **After execution of instruction) Content of affected Registers/Memor y locations and lags** |
| **mov ax,3 000 H** |  |  | **Loa d 300**  **0 H**  **into AX** |  |  |
| **mov ds,a x** |  |  | **Mo ve AX**  **to DS** |  |  |
| **mov ax,0 C004H** |  |  | **Loa d 0C0**  **04H**  **into AX** |  |  |
| **mov bx,0 0FF H** |  |  | **Loa d 00F FH**  **into BX** |  |  |
| **xor ax,b x** |  |  | **Perf orm XO R**  **bet wee n** |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | **AX**  **and BX** |  |  |
| **mov [0B 0h],**  **ax** |  |  | Stor e AX at me mor y [0B 0H] |  |  |
| **ret** |  |  | **Ret urn fro m pro ced ure** |  |  |

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

mov ax,3000H mov ds,ax

mov ax,0C004H

not ax

mov [0C0h],ax ret

STEP-BY-STEP EXECUTION:

{Remove this comment before printing: edit following section as per requirement}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation xecuted by nstruction** | **Before execution f instruction) ontent of to be ffected egisters/Memory ocations and flag** | **After execution of instruction) Content of affected Registers/Memor y locations and lags** |
| **mov ax,3 000 H** |  |  | **Loa d 300**  **0H**  **into AX** |  |  |
| **mov ds,a x** |  |  | **Mo ve AX**  **to DS** |  |  |
| **mov ax,0 C004H** |  |  | **Loa d 0C0**  **04H**  **into AX** |  |  |
| **Not ax** |  |  | **Perf orm NO T**  **ope rati on on AX** |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **mov [0C 0h],**  **ax** |  |  | **Stor e AX**  **at me mor y [0C**  **0H** |  |  |
| **ret** |  |  | **Ret urn fro m pro ced ure** |  |  |

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

mov ax,3000H mov ds,ax

mov ax,0C004H shl ax,1

mov [0D0h],ax ret

STEP-BY-STEP EXECUTION:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation xecuted by nstruction** | **Before execution f instruction) ontent of to be ffected egisters/Memory ocations and flag** | **After execution of instruction) Content of affected Registers/Memor y locations and lags** |
| **mov** |  |  | **Loa** |  |  |
| **ax,3** | **d** |
| **000** | **300** |
| **H** | **0H** |
|  | **into** |
|  | **AX** |
| **mov** |  |  | **Mov** |  |  |
| **ds,a** | **e** |
| **x** | **AX** |
|  | **to** |
|  | **DS** |
| **mov** |  |  | **Loa** |  |  |
| **ax,0** | **d** |
| **C04** | **0C0** |
| **5H** | **04H** |
|  | **into** |
|  | **AX** |
| **shl** |  |  | **Shif** |  |  |
| **ax,1** | **t** |
|  | **AX** |
|  | **left** |
|  | **by 1** |
|  | **bit** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **mov [0D 0h],**  **ax** |  |  | **Stor e AX**  **at me mor y [0D**  **0H** |  |  |
| **ret** |  |  | **Ret urn fro m proc edu re** |  |  |

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

mov ax,3000H mov ds,ax

mov ax,0C004H shr ax,1

mov [0D0h],ax ret

STEP-BY-STEP EXECUTION:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation xecuted by nstruction** | **Before execution f instruction) ontent of to be ffected egisters/Memory ocations and flag** | **After execution of instruction) Content of affected Registers/Memor y locations and lags** |
| **mov ax,3 000 H** |  |  | **Loa d 300**  **0H**  **into AX** |  |  |
| **mov ds,a x** |  |  | **Mov e AX**  **to DS** |  |  |
| **mov ax,0 C03 3H** |  |  | **Loa d 0C0**  **04H**  **into AX** |  |  |
| **shr ax,1** |  |  | **Shif t AX**  **righ t by 1 bit** |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **mov [0D 0h],**  **ax** |  |  | **Stor e AX**  **at me mor y [0D**  **0H** |  |  |
| **ret** |  |  | **Ret urn fro m proc edu re** |  |  |

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

mov ax,3000H mov ds,ax

mov ax,0C004H sal ax,1

mov [0E0h],ax ret

STEP-BY-STEP EXECUTION:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation xecuted by nstruction** | **Before execution f instruction) ontent of to be ffected egisters/Memory ocations and flag** | **After execution of instruction) Content of affected Registers/Memor y locations and lags** |
| **mov ax,3 000 H** |  |  | **Loa d 300**  **0H**  **into AX** |  |  |
| **mov ds,a x** |  |  | **Mov e AX**  **to DS** |  |  |
| **mov ax,0 C03 3H** |  |  | **Loa d 0C0**  **04H**  **into AX** |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **shl ax,1** |  |  | **Shif t AX**  **left by 1 bit (arit hme tic**  **shift left)** |  |  |
| **mov [0E 0h],**  **ax** |  |  | **Stor e AX**  **at me mor y [0E**  **0H]** |  |  |
| **ret** |  |  | **Ret urn fro m proc edu re** |  |  |

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

mov ax,3000H mov ds,ax

mov ax,0C004H sar ax,1

mov [0F0h],ax ret

STEP-BY-STEP EXECUTION:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation xecuted by nstruction** | **Before execution f instruction) ontent of to be ffected egisters/Memory ocations and flag** | **After execution of instruction) Content of affected Registers/Memor y locations and lags** |
| **mov ax,3 000 H** |  |  | **Loa d 300**  **0H**  **into AX** |  |  |
| **mov ds,a x** |  |  | **Mov e AX**  **to DS** |  |  |
| **mov ax,0 C03 3H** |  |  | **Loa d 0C0**  **04H**  **into AX** |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **sar ax,1** |  |  | **Shif t AX**  **righ t by 1 bit (arit hme tic shift righ**  **t)** |  |  |
| **mov [0F0**  **h],a x** |  |  | **Stor e AX**  **at me mor y [0F0 H]** |  |  |
| **ret** |  |  | **Ret urn fro m proc edu re** |  |  |

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

mov ax,3010H mov ds,ax

mov ax,0C004H rcl ax,1

mov [00h],ax ret

STEP-BY-STEP EXECUTION:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation xecuted by nstruction** | **Before execution f instruction) ontent of to be ffected egisters/Memory ocations and flag** | **After execution of instruction) Content of affected Registers/Memor y locations and lags** |
| **mov ax,3 000 H** |  |  | **Loa d 301**  **0H**  **into AX** |  |  |
| **mov ds,a x** |  |  | **Mov e AX**  **to Ds** |  |  |
| **mov ax,0 C03 3H** |  |  | **Loa d 0C0**  **04H**  **into AX** |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **rcl ax,1** |  |  | **Rot ate AX**  **left by 1 bit thro ugh carr y mov [00h**  **],ax** |  |  |
| **mov [00h**  **],ax** |  |  | **Stor e AX**  **at me mor y [00**  **H]** |  |  |
| **ret** |  |  | **Ret urn fro m proc edu re** |  |  |

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

mov ax,3010H mov ds,ax

mov ax,0C004H rol ax,1

mov [10h],ax ret

STEP-BY-STEP EXECUTION:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation xecuted by nstruction** | **Before execution f instruction) ontent of to be ffected egisters/Memory ocations and flag** | **After execution of instruction) Content of affected Registers/Memor y locations and lags** |
| **mov ax,3 000 H** |  |  | **Loa d 301**  **0H**  **into AX** |  |  |
| **mov ds,a x** |  |  | **Mov e AX**  **to DS** |  |  |
| **mov ax,0 C004H** |  |  | **Loa d 0C0**  **04H**  **into AX** |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **rol ax,1** |  |  | **Rot ate AX**  **left by 1 bit (wit hout carr**  **y)** |  |  |
| **mov [10h**  **],ax** |  |  | **Stor e AX**  **at me mor y [10**  **H]** |  |  |
| **ret** |  |  | **Ret urn fro m proc edu re** |  |  |

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

mov ax,3010H mov ds,ax

mov ax,0C004H rcr ax,1

mov [20h],ax ret

STEP-BY-STEP EXECUTION:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation xecuted by nstruction** | **Before execution f instruction) ontent of to be ffected egisters/Memory ocations and flag** | **After execution of instruction) Content of affected Registers/Memor y locations and lags** |
| **mov ax,3 000 H** |  |  | **Mov e AX**  **to DS** |  |  |
| **mov ds,a x** |  |  | **Loa d 0C0**  **04H**  **into AX** |  |  |
| **mov ax,0 C03 3H** |  |  | **Rot ate AX**  **righ t by 1 bit thro ugh carr** |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | **y** |  |  |
| **rcr ax,1** |  |  | **Stor e AX**  **at me mor y [20**  **H** |  |  |
| **mov [20h**  **],ax** |  |  | **Rot ate AX**  **righ t by 1 bit thro ugh carr**  **y** |  |  |
| **ret** |  |  | **Ret urn fro m proc edu re** |  |  |

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

mov ax,3010H mov ds,ax

mov ax,0C004H ror ax,1

mov [30h],ax ret

STEP-BY-STEP EXECUTION:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction in Assembly Language** | **Screenshot of the Instruction in program memory** | **Value in Instruction Pointer** | **Operation xecuted by nstruction** | **Before execution f instruction) ontent of to be ffected egisters/Memory ocations and flag** | **After execution of instruction) Content of affected Registers/Memor y locations and lags** |
| **mov ax,3 000 H** |  |  | **Loa d 301**  **0H**  **into AX** |  |  |
| **mov ds,a x** |  |  | **Mov e AX**  **to DS** |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **mov ax,0 C03 3H** |  |  | **Loa d 0C0**  **04H**  **into AX** |  |  |
| **ror ax,1** |  |  | **Rot ate AX**  **righ t by 1 bit (wit hout carr**  **y)** |  |  |
| **mov [20h**  **],ax** |  |  | **Stor e AX**  **at me mor y [30**  **H** |  |  |
| **ret** |  |  | **Ret urn fro m proc edu re** |  |  |

CONCLUSION:

