**MICROPROCESSOR & EMBEDDED SYSTEM**

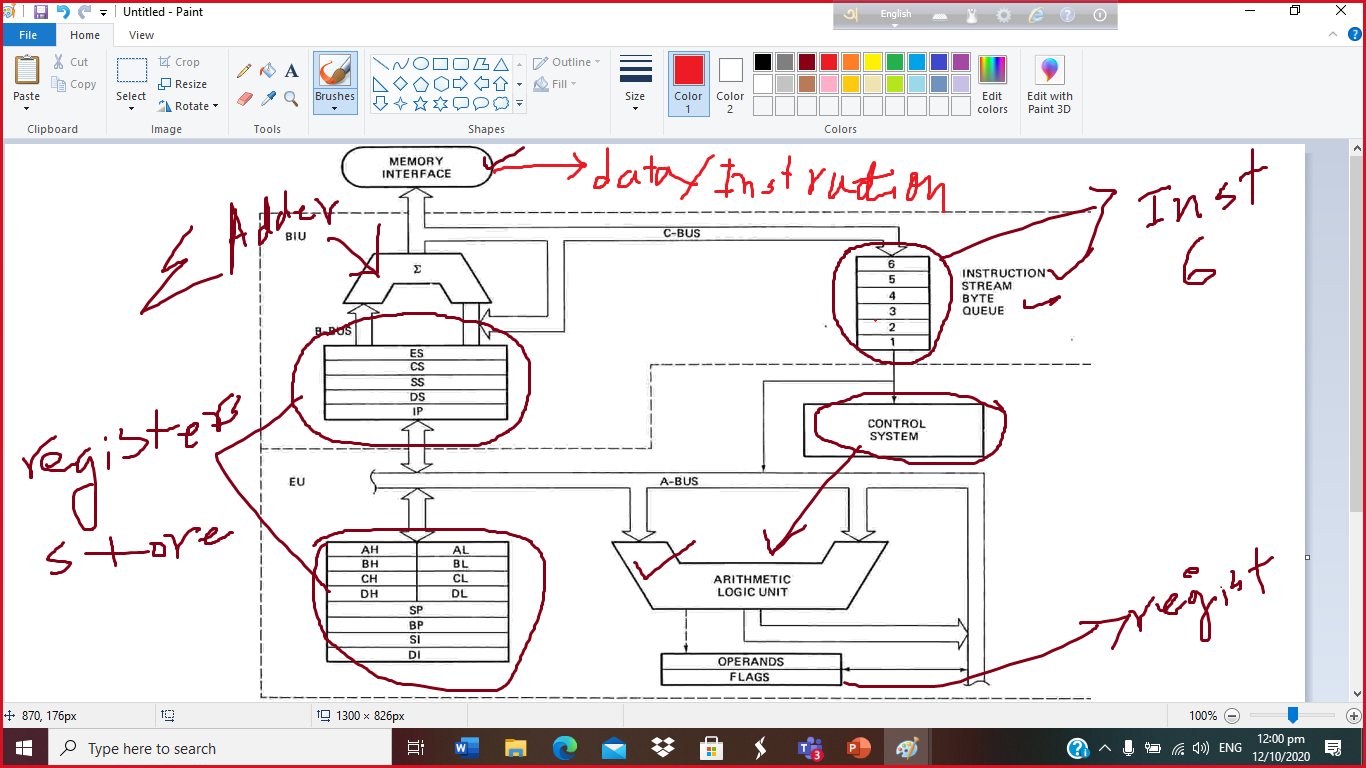
**Sec- A, C, H**

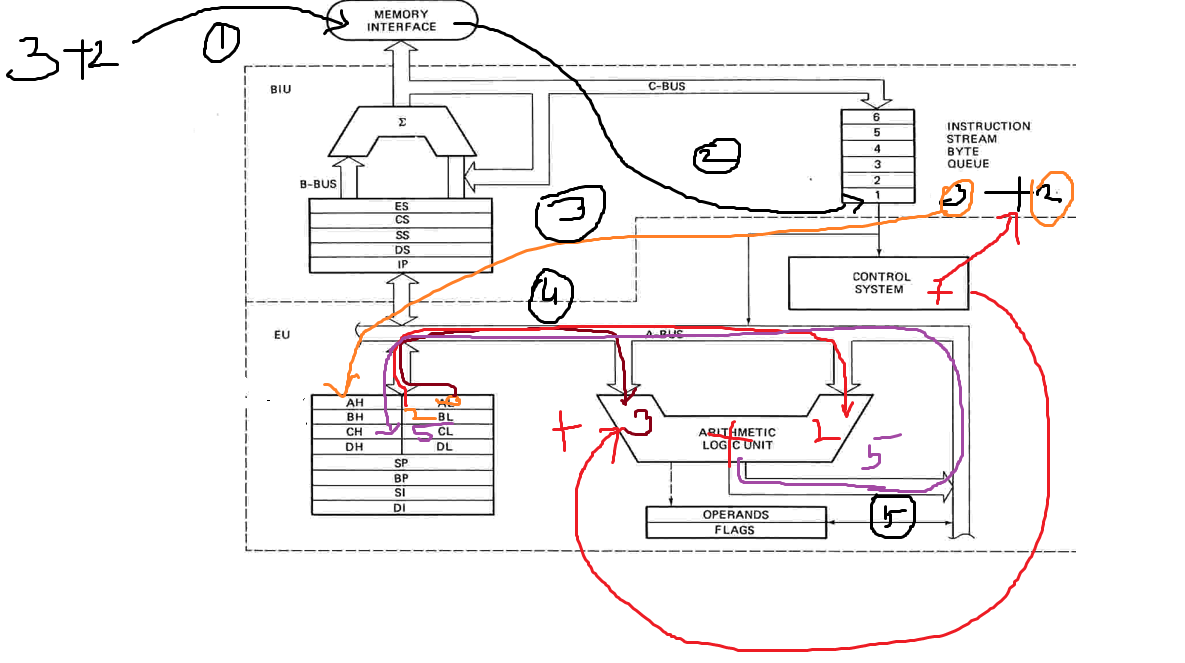
**COURSE TEACHER- NADIA NOWSHIN**

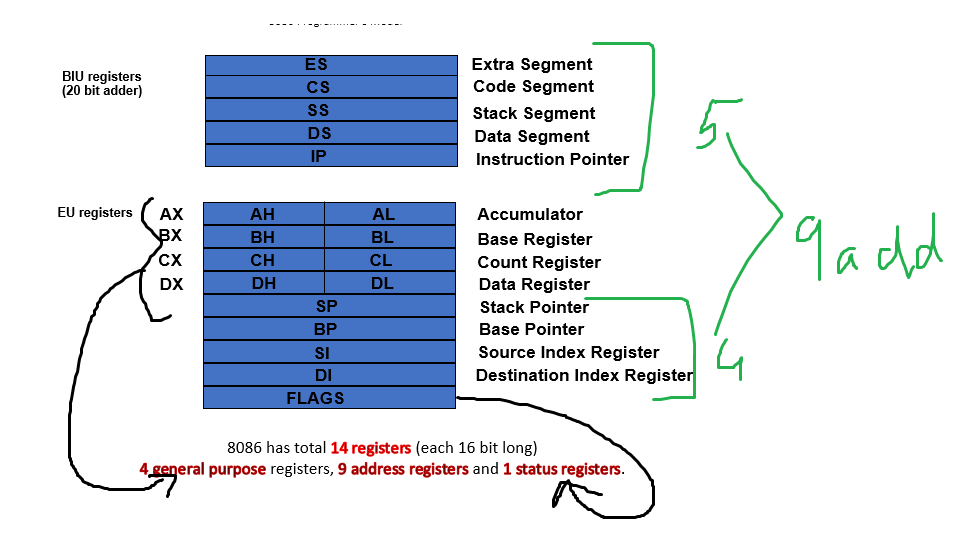
**Assistant Professor, Dept of EEE & COE**

**email-** [**nowshin@aiub.edu**](mailto:nowshin@aiub.edu)

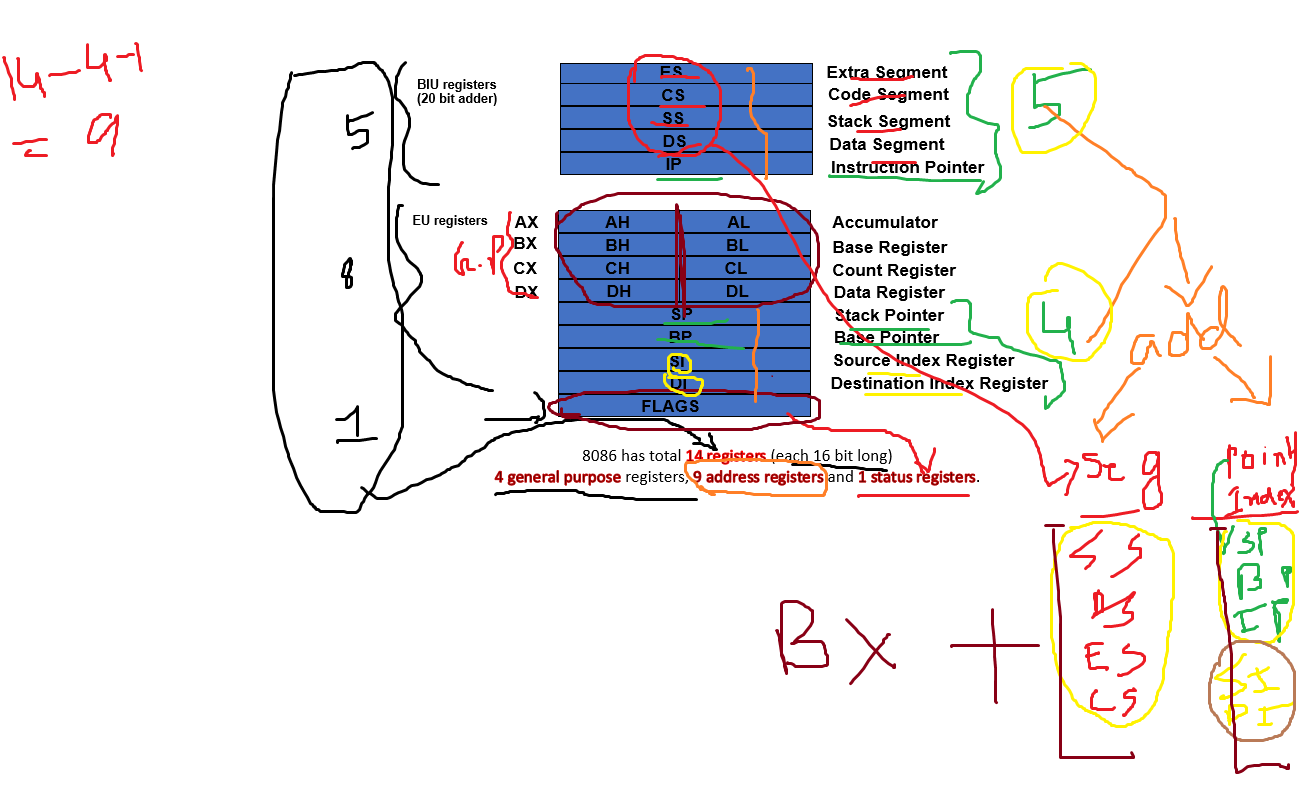
**Class will start sharp at \*\*:\*\* am.**



****

****

**Mid\_Assign\_Sec\_\*\_Roll\_\***

****

**Lab\_5\_Sec\_A\_Roll\_\***

**0000B= 0 4 bit one address total 16 address**

**0001B=1**

**0010B=2**

**.**

**0101B=5**

**.**

**1111B=15**

**Binary to Decimal Conversion:-**

**1111 = 15**

**1 0 0 1**

**Power of =**

**= 8 + 0 + 0+ 1 =9**

**8 4 2 1**

**1 0 1 0 = 10**

**1 0 1 1 = 11**

**1 1 1**

**1011**

**1111**

**11010 8 4 2 1**

**Binary to hexa conversion:-**

**1011 1111 0111 0001 B**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **8** | **4** | **2** | **1** | **8** | **4** | **2** | **1** | **8** | **4** | **2** | **1** | **8** | **4** | **2** | **1** |
| **Binary** | **1** | **0** | **1** | **1** | **1** | **1** | **1** | **1** | **0** | **1** | **1** | **1** | **0** | **0** | **0** | **1** |
| **Dec** | **11** | | | | **15** | | | | **7** | | | | **1** | | | |
| **Hex** | **B** | | | | **F** | | | | **7** | | | | **1** | | | |

**1011 1111 0111 0001 B= BF71 H**

**8 4 2 1**

**1 0 1 0 0101 0001 1110 B =**

**10 5 1 14**

**A 5 1 E H**

**20 bit Physical addressing: -**

**0000 0000 0000 0000 0000 B = 00000 H**

**0000 0000 0000 0000 0001 B = 00001 H**

**0000 0000 0000 0000 0010 B = 00002 H**

**.**

**.**

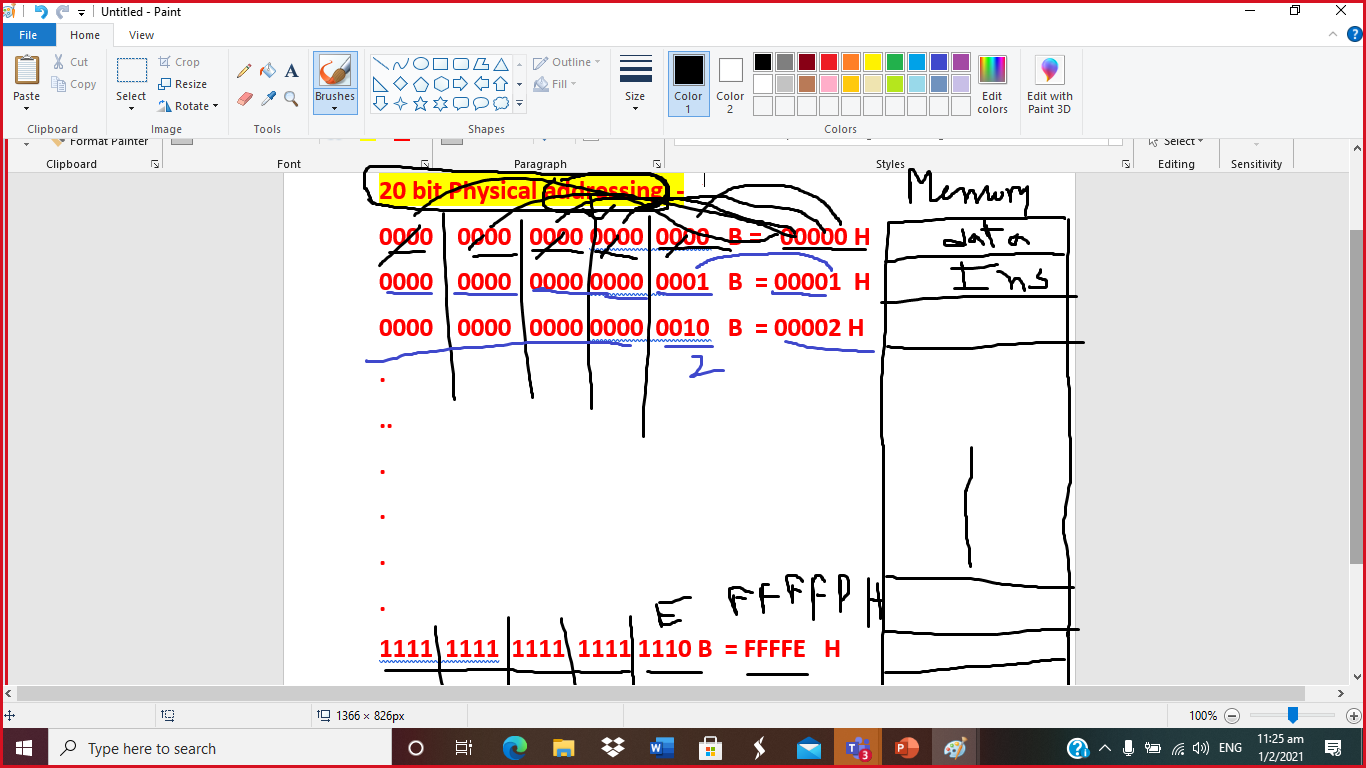
**.**

**.**

**.**

**1111 1111 1111 1111 1110 B = FFFFE H**

**1111 1111 1111 1111 1111 B = FFFFF H**



**FFFFFh**

**Segment: Offset**

**FFFF : 0000 ….**

**.**

**.**

**FFFF: FFFF**

|  |  |
| --- | --- |
| **Decimal** | **Hexa** |
| **0** | **0** |
| **1** | **1** |
| **2** | **2** |
| **3** | **3** |
| **4** | **4** |
| **5** | **5** |
| **6** | **6** |
| **7** | **7** |
| **8** | **8** |
| **9** | **9** |
| **10** | **A** |
| **11** | **B** |
| **12** | **C** |
| **13** | **D** |
| **14** | **E** |
| **15** | **F** |

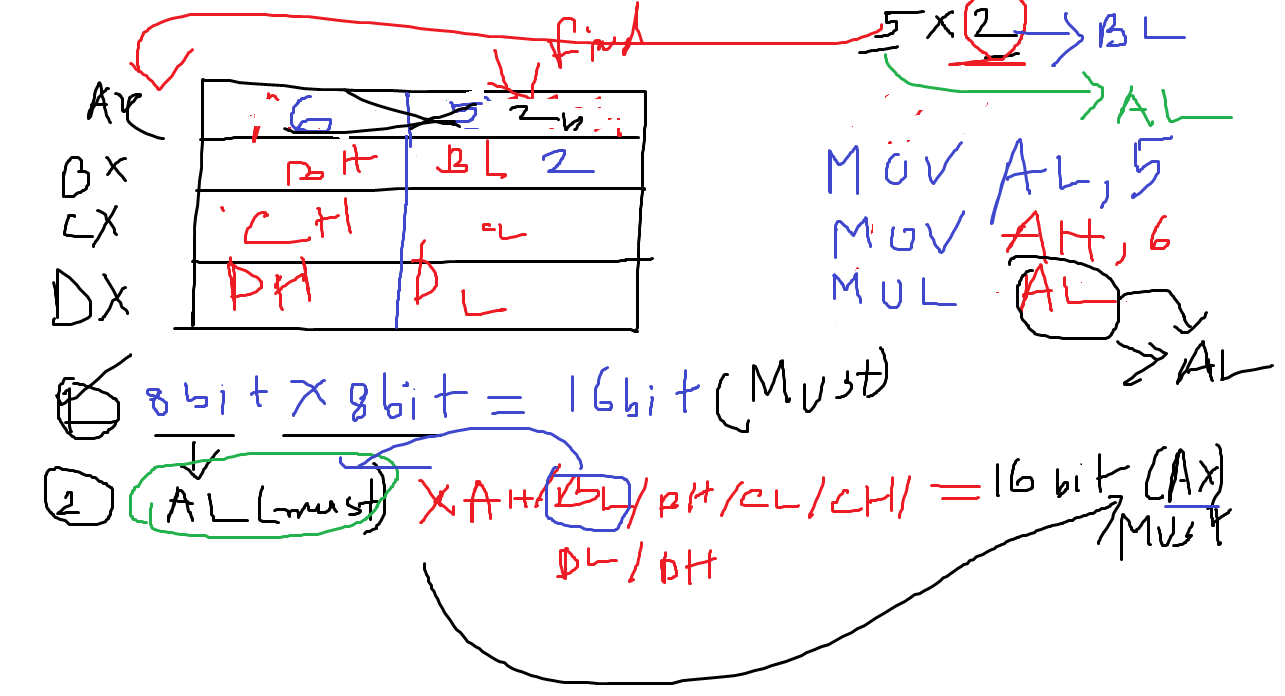
**cin>> s; cin>> y;**

**2+3**

**Mov AX, 2**

**Mov BX, 3**

**ADD AX, BX----🡪 Ax= Ax+BX= 2+3=5**

****

**Mov AL, 10**

**Mov DL, 10**

**Mul DL ----🡪Ax(16bit)= DL(8 bit) \*AL (8bit)=10\*10=100**

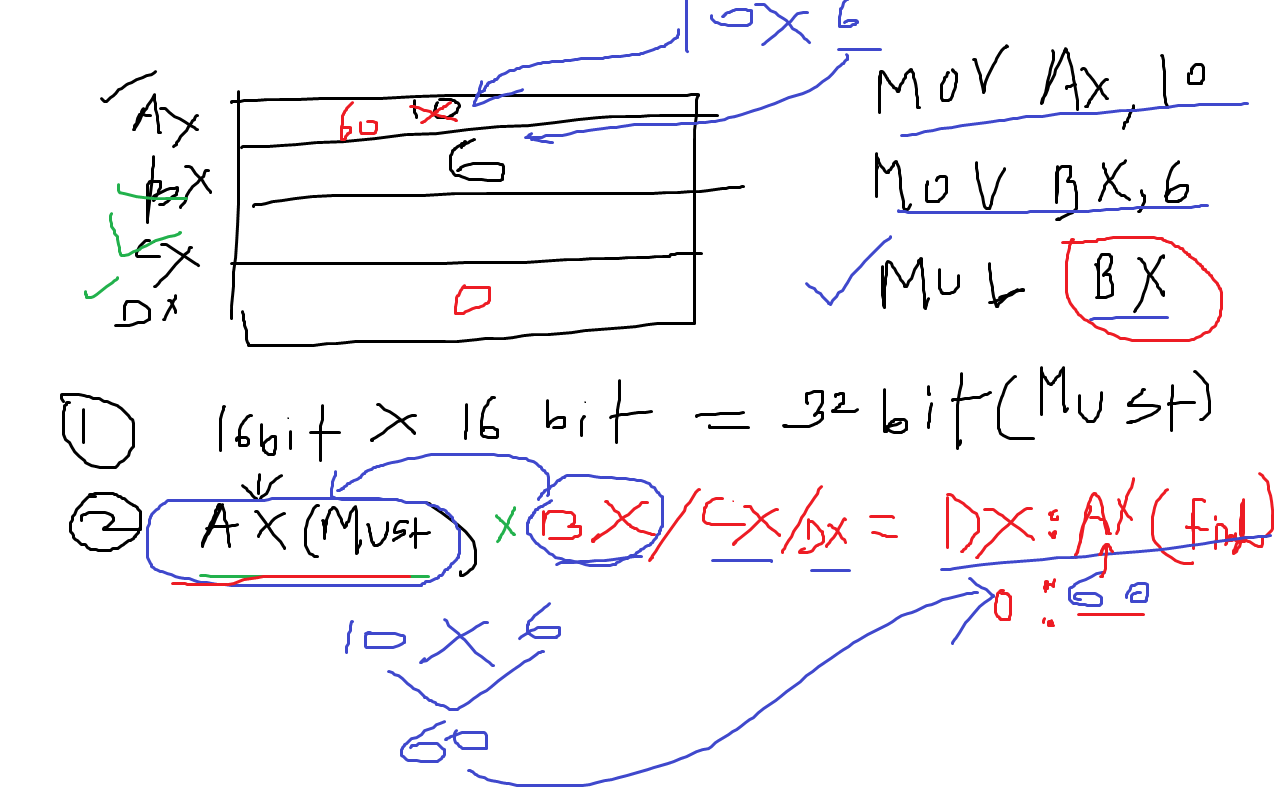
**Perform 10 - 3 using 8086 Assembly instruction set:-**

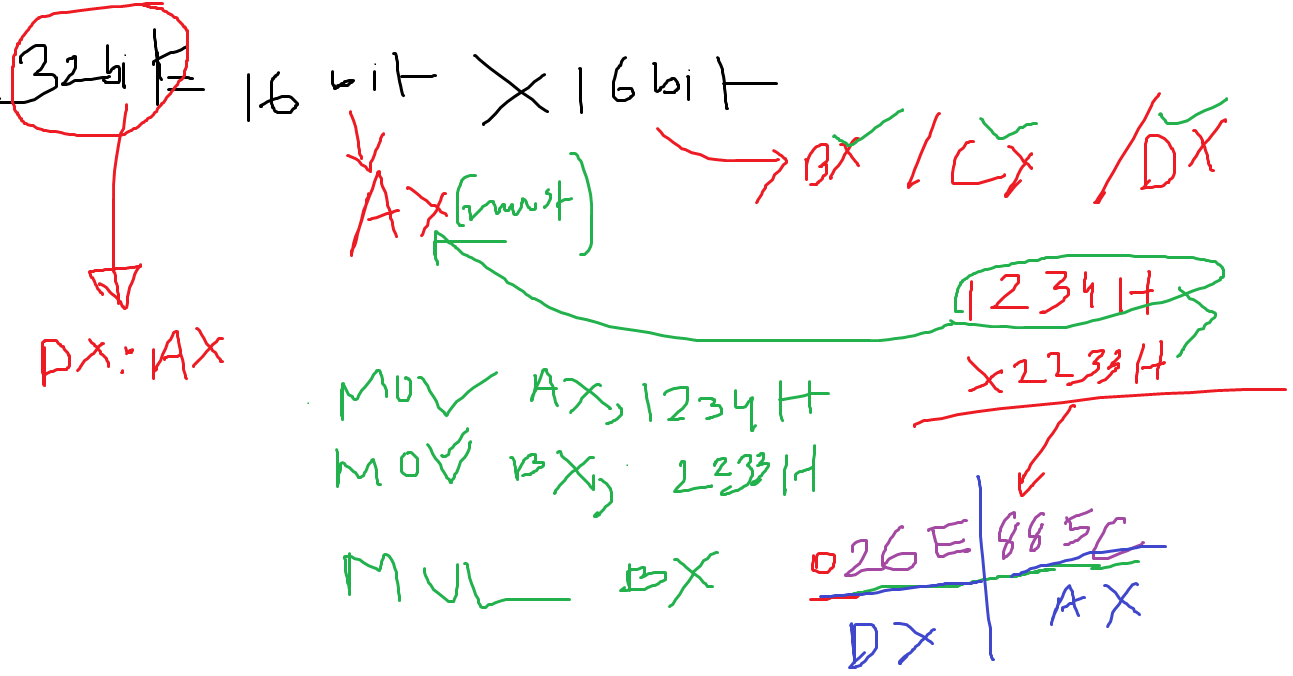
**Mov CL, 10----🡪 CL=10**

**mov DL, 3-----🡪 DL=3**

**SUB CL,DL ---🡪 CL = 10-3**

**= 7 , DL=3**

****

****

**Perform 2 \* 3 using 8086 instruction set:-**

**MOV AX, 2----🡪 AX=2(00000000000000010 B)**

**MOV BX, 3-----🡪 BX=3(00000000000000011 B)**

**MUL BX ---🡪 DX:AX = AX(16bit)\*BX (16 bit)**

**16bit \*16 bit= 32 bit**

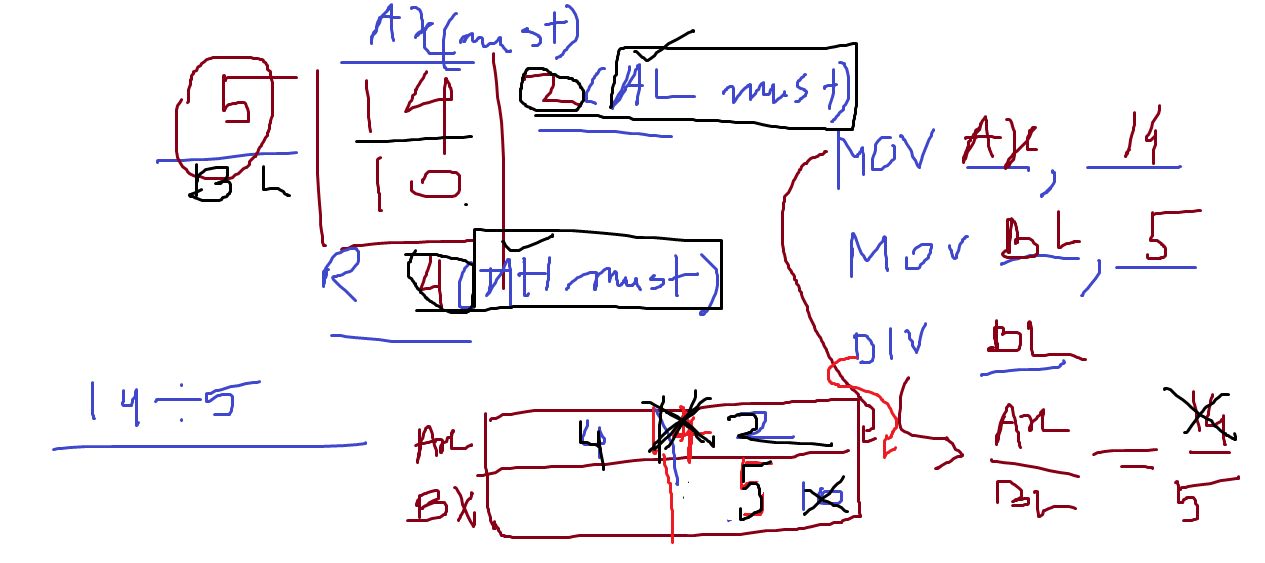
**6= 110**

**0000000000000000 0000000000000110**

**DX (higher 16 bit result) : Ax ((lower 16 bit result)**

**0 6**

**DIVISION:-**

****

**010**

**011 101 01 0 0 10**

**101**

**For (i=1; i<=4;i++)**

**{ ppp }**

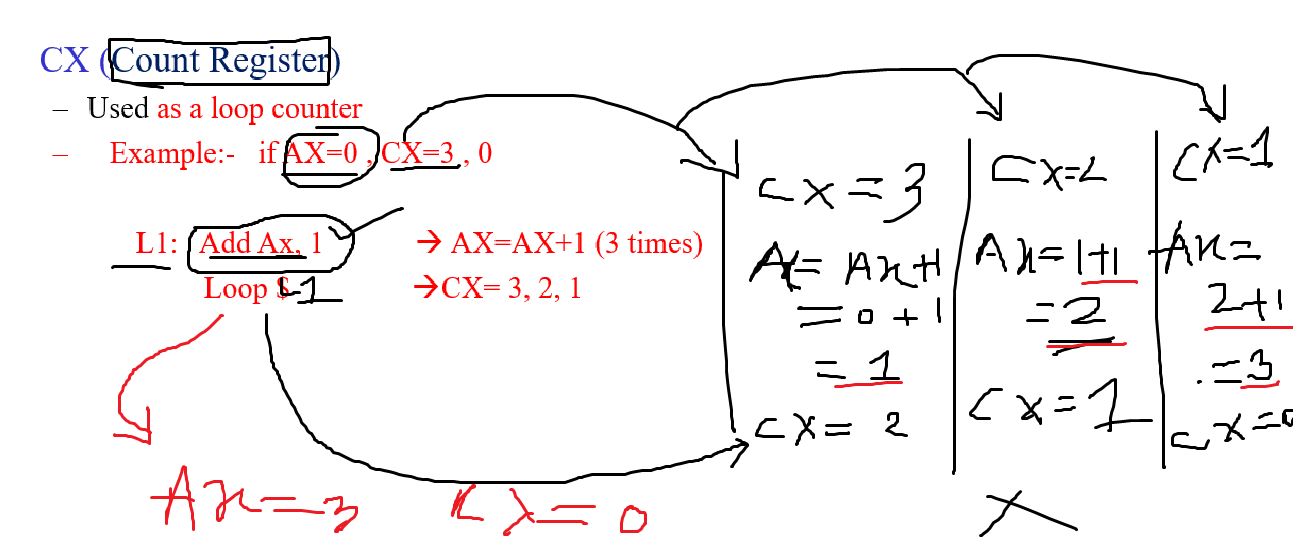
**1**

**2**

**3**

**4**

**LOOP Operation:-**

****

**CX= 5, Ax=2**

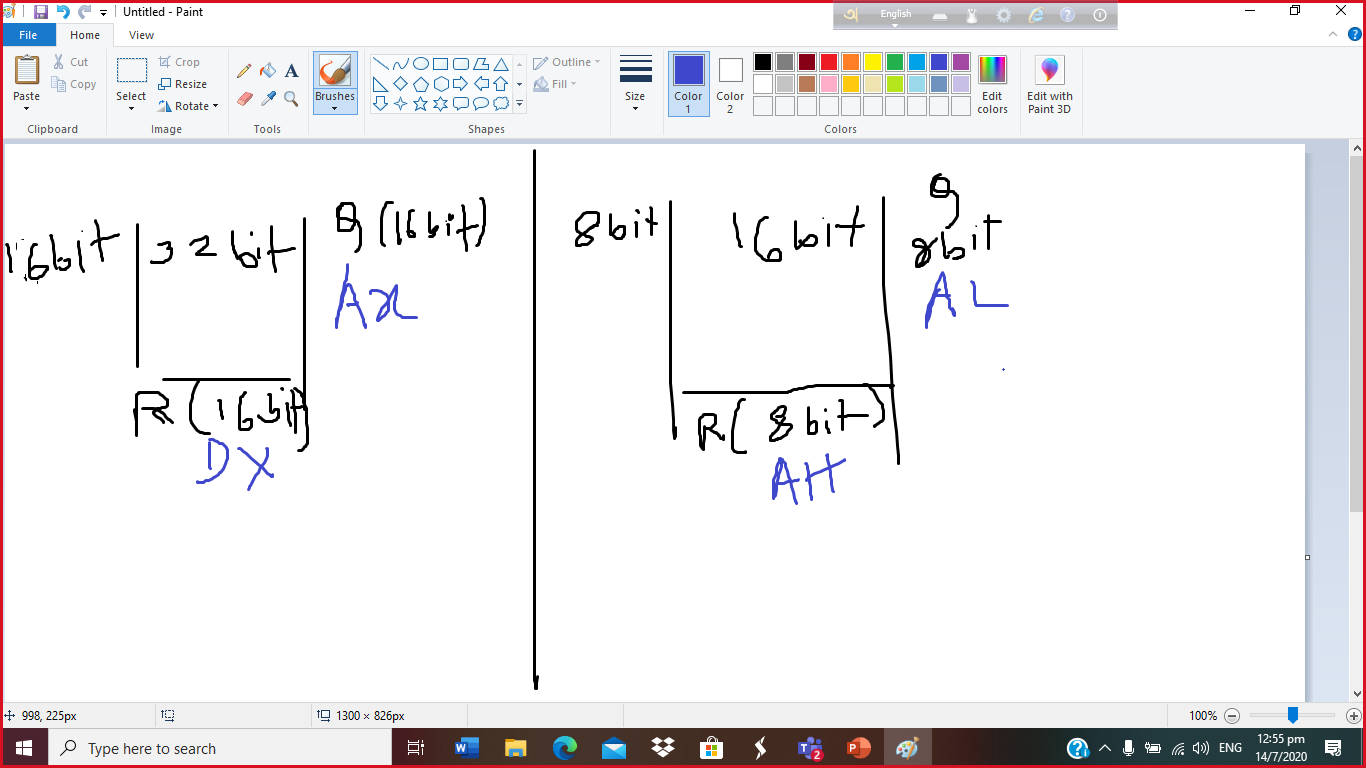
**Mov CX, 5**

**Mov AX,2**

**L1: Add Ax, 2 -----------🡪 Ax= AX+2**

**Loop L1 --🡪 Cx= CX-1, 5……….0**

**CX=0 ? , Ax= 12 ?**



**Segment associated with Offset registers: -**

|  |  |
| --- | --- |
| **Segment register** | **Offset/ pointer & Index register** |
| **CS** | **IP/ BP/BX** |
| **SS** | **SP/ BP/BX** |
| **DS** | **SI/DI/ BP/BX** |
| **ES** | **SI/DI/ BP/BX** |

* **BX can be used with all the segments.**

**For 20 bit address :-**

**First address will be :-**

**0000 0000 0000 0000 0000 B =00000 h**

**0000 0000 0000 0000 0001 B =00001 h**

**0000 0000 0000 0000 0010 B = h.**

**,**

**.**

**.**

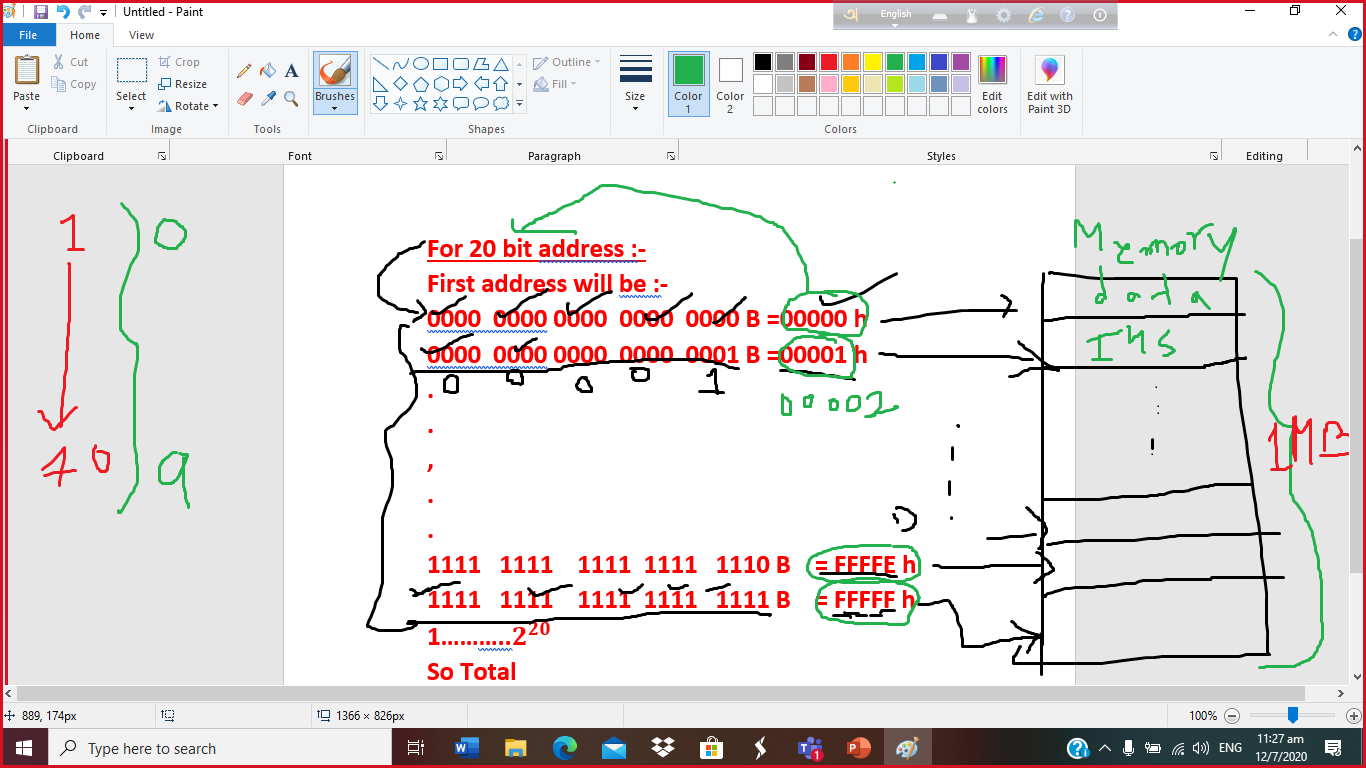
**1111 1111 1111 1111 1110 B = FFFFE h**

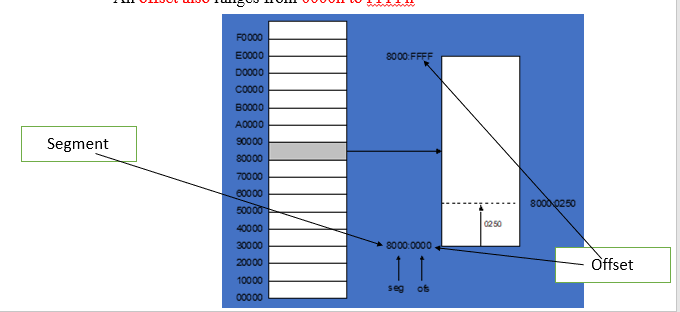
**1111 1111 1111 1111 1111 B = FFFFF h**

**1………..**

**So Total**

**0………..() memory locations .**



****

|  |
| --- |
|  |
|  |
| **.**  **.**  **.** |
|  |

|  |
| --- |
| **FFFF: FFFF** |
| **FFFF:FFFE** |
| **.**  **.**  **.** |
| **FFFF:0000** |

Let, x=5 , 5x +2=27

X EQU 5

MOV CX, X

ADD CX, CX

int a[3]= { 1, 2, 3}

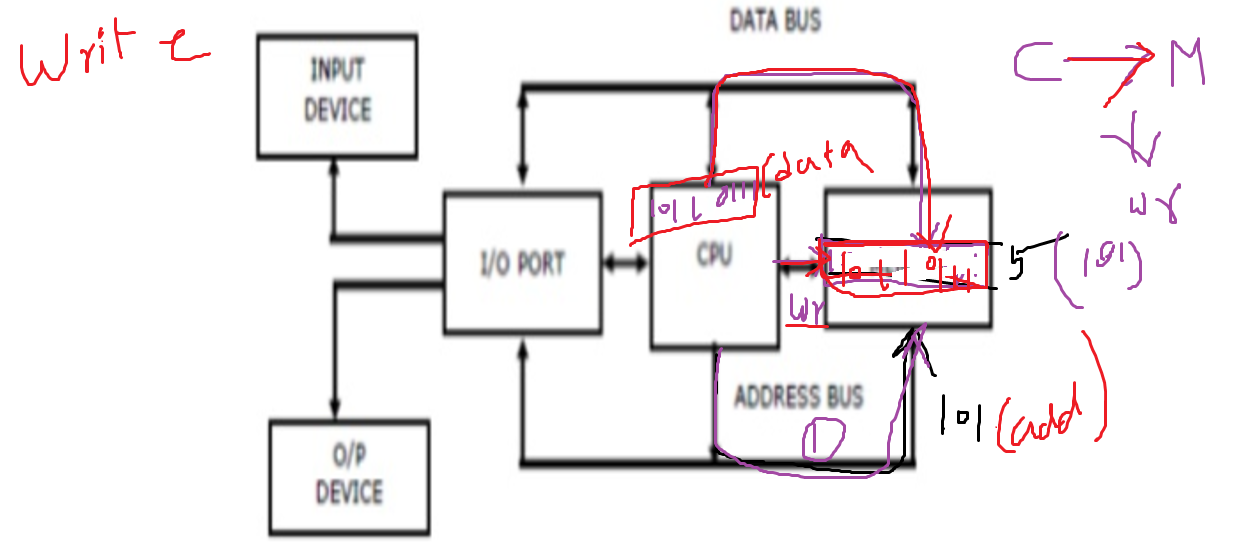
a db 1, 2, 3

a dw 1, 2, 3

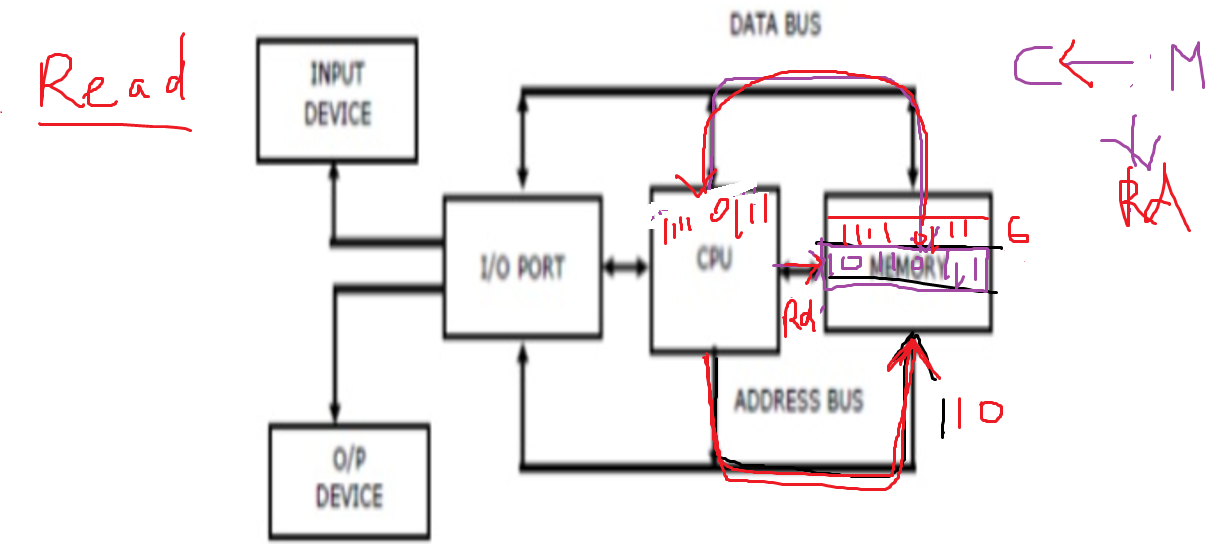
Var DW 1234 h

**LECTURE-1**

**Write Operation: -**

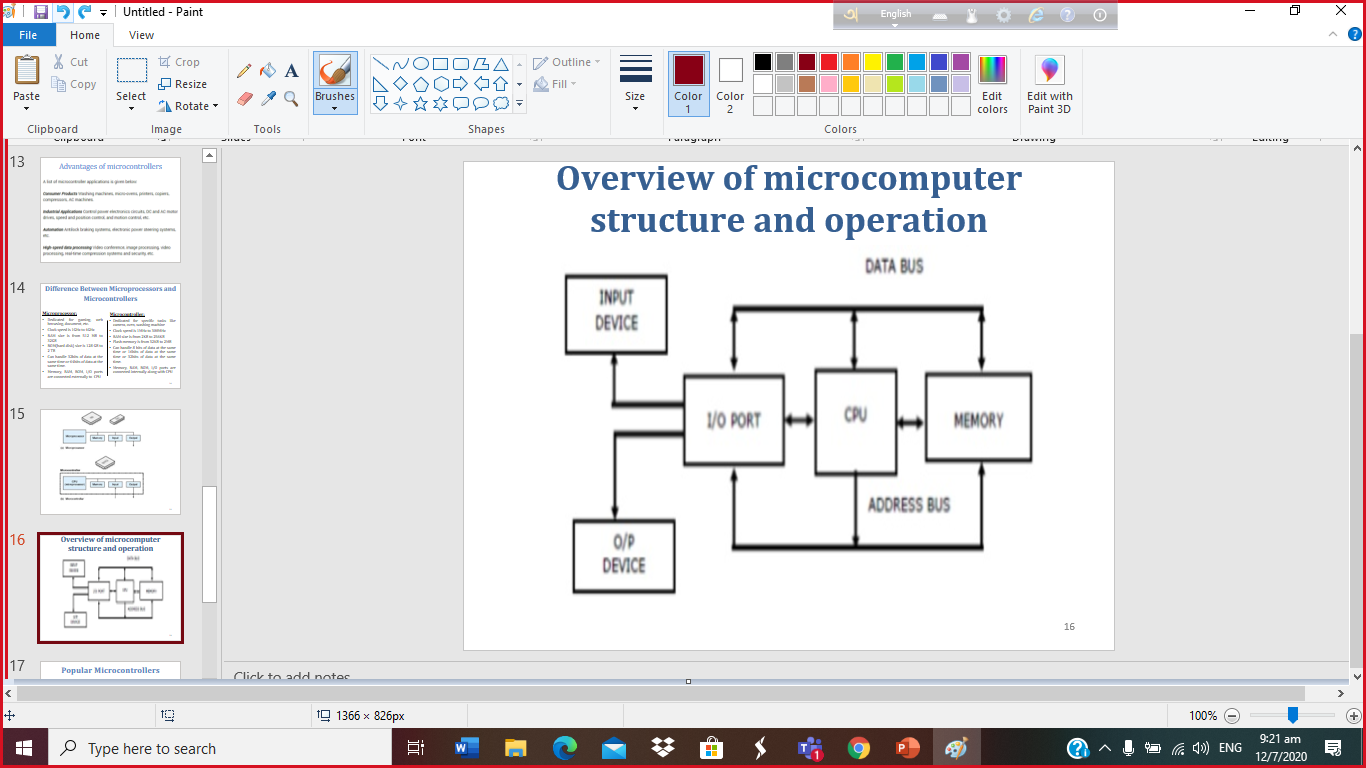


**Read Operation: -**



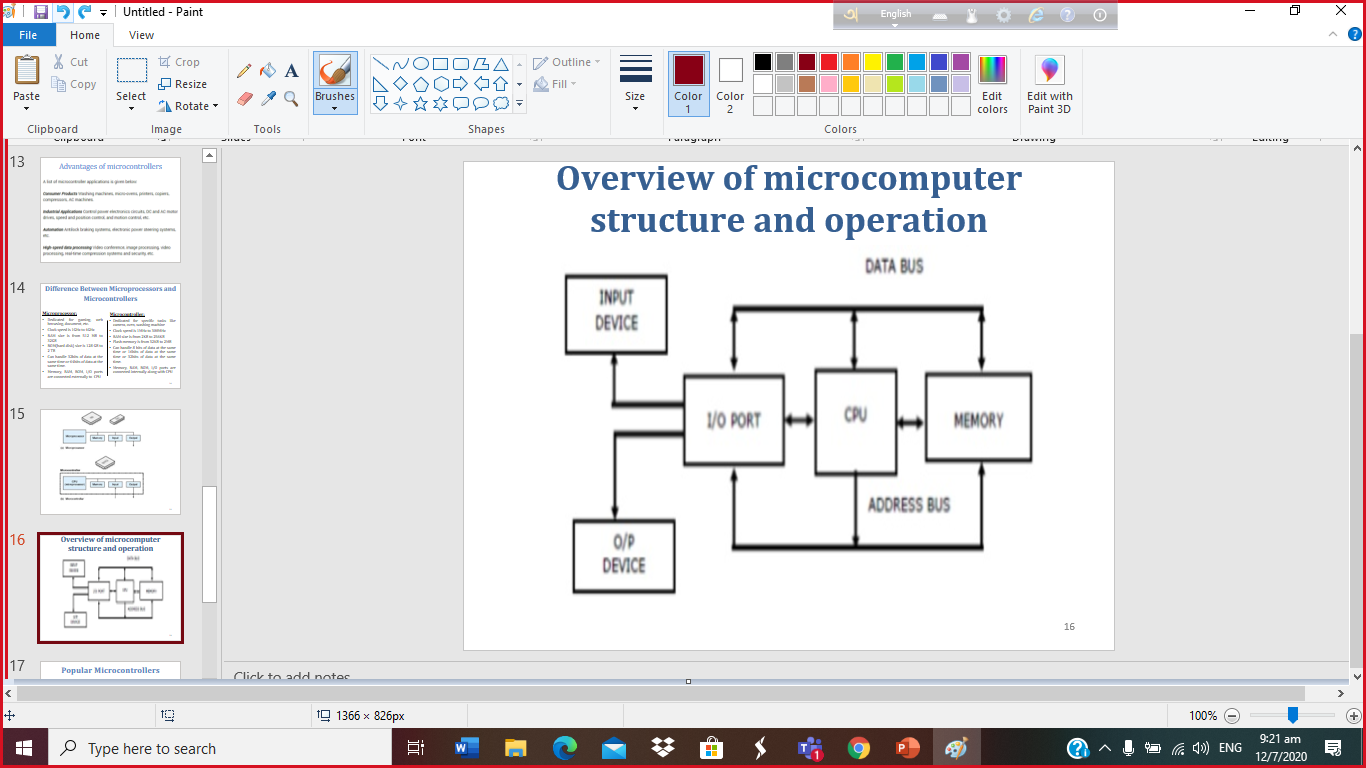
Io/ = 1 perform Input/ Output operation

= 0 perform memory operation



**DDR**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | **1** | **0** | **1** | **0** | **1** | **1** | **0** |



**How to record your screen?**

<https://www.youtube.com/watch?v=KCi7z-bJJXw>

|  |  |
| --- | --- |
| **Address** | **Instruction/Data(8 bit)** |
| **0h/ 0000 B** | **LDA Fh /** |
| **1h/ 0001 B** | **ADD 8h/** |
| **2h/ 0010 B** | **SUB Ch/** |
| **3h/ 0011 B** | **OUT X h/** |
| **4h/ 0100 B** | **HLT X h/** |
| **5h/ 0101 B** |  |
| **6h/ 0110 B** |  |
| **7h/ 0111 B** |  |
| **8h/ 1000 B** | **02 h/ 0000 0010 B** |
| **9h/ 1001 B** | **06 h/ 0000 0101 B** |
| **Ah/ 1010 B** |  |
| **Bh/ 1011 B** | **04 h/ 0000 0100 B** |
| **Ch/ 1100 B** | **03 h/ 0000 0011 B** |
| **Dh/ 1101 B** |  |
| **Eh/ 1110 B** |  |
| **Fh/ 1111 B** | **03 h/ 0000 0100 B** |

|  |  |
| --- | --- |
| **Address** | **Instruction/Data(8 bit)** |
| **0h/ 0000 B** | **LDA 9h /** |
| **1h/ 0001 B** | **ADD Ah/** |
| **2h/ 0010 B** | **SUB Fh/** |
| **3h/ 0011 B** | **OUT X h/** |
| **4h/ 0100 B** | **HLT X h/** |
| **5h/ 0101 B** |  |
| **6h/ 0110 B** |  |
| **07h/ 0111 B** |  |
| **8h/ 1000 B** | **02 h/ 0000 0010 B** |
| **9h/ 1001 B** | **05 h/ 0000 0101 B** |
| **Ah/ 1010 B** | **04 h/ 0000 0100B** |
| **Bh/ 1011 B** |  |
| **Ch/ 1100 B** | **03 h/ 0000 0011 B** |
| **Dh/ 1101 B** |  |
| **Eh/ 1110 B** | **0** |
| **Fh/ 1111 B** | **06 h/ 0000 0110 B** |