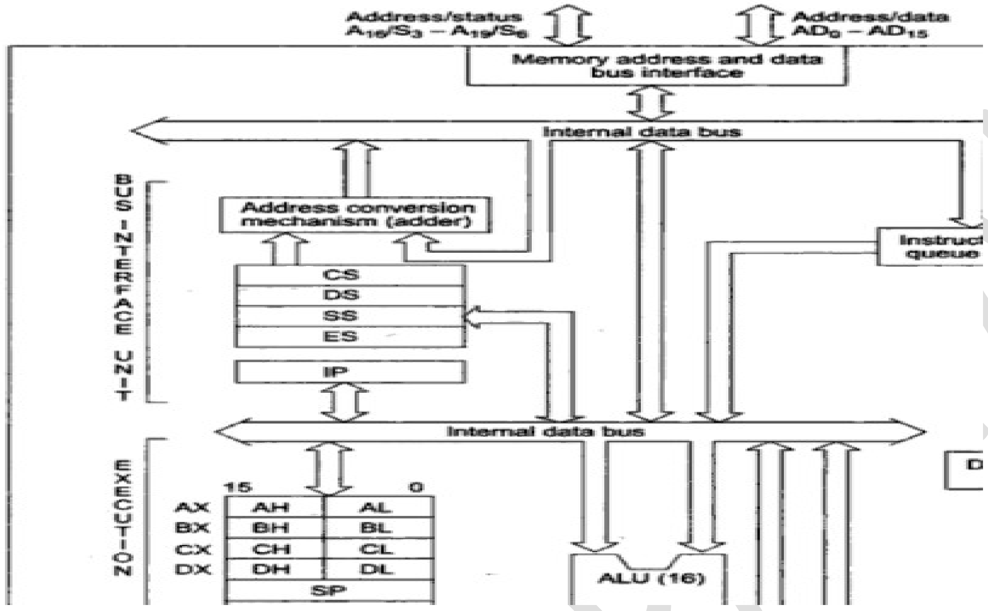
## 1. Introduction to MASM

==FOLLOW THE NOTES PROVIDED==

## 2. 8086 Architecture (Left side page of the record )


Address/status $A_{16}/S_3 - A_{19}/S_6$
Address/data $AD_0 - AD_{15}$
Memory address and data bus interface
Internal data bus
BUS INTERFACE UNIT
Address conversion mechanism (adder)
Instruct queue
CS
DS
SS
ES
IP
Internal data bus
EXECUTION
15
0
AX
BX
CX
DX
AH
AL
BH
BL
CH
CL
DH
DL
SP
ALU (16)

## 3. 8086 instructions classifications and List

1. DATA TRANSFER INSTRUCTIONS
================================

| | | |
|---|---|---|
| MOV | PUSH | POP |
| XCHG | IN | OUT |
| XLAT | LEA | LDS/LES |
| LAHF | SAHF | PUSHF |
| POPF | | |

2. ARITHMETIC INSTRUCTIONS
================================

| | | | |
|---|---|---|---|
| ADD | ADC | | DEC |
| SUB | SBB | | CMP |
| MUL | IMUL | | |
| DIV | IDIV | | NEG |
| AAA | AAS | AAM | AAD |
| DAA | DAS | CBW | CWD |

3. LOGICAL INSTRUCTIONS
================================

| | | |
|---|---|---|
| AND | OR | NOT |
| XOR | TEST | SHL/SAL |
| SHR | SAR | ROR |
| ROL | RCR | RCL |

4. STRING MANIPULATION INSTRUCTIONS
================================

| | | |
|---|---|---|
| REP | MOVSB | CMPS |
| SCAS | LODS | STOS |

5. UNCONDITIONAL BRANCH INSTRUCTIONS

    ================================

| | | |
|---|---|---|
| CALL | RET | INT N |
| INTO | JMP | IRET |
| LOOP | | |

6. CONDITIONAL BRANCH INSTRUCTIONS

    =================================

    JN/JE
    JNZ/JNE
    JS
    JNS
    JO
    JNO
    JP/JPE
    JNP
    JB/JNAE/JC
    JNB/JAE/JNC
    JBE/JNA
    JNBE/JA
    JL/JNGE
    JNL/JGE
    JLE/JNC
    JNLE/JE

7. FLAG MANIPULATION INSTRUCTIONS

    ===================================

| | | |
|---|---|---|
| CLC | CMC | STC |
| CLD | STD | CLI |
| STI | | |

8. MACHINE CONTRO INSTRUCTIONS

    =====================================

    WAIT HLT NOP ESC LOCK

    =====================================

## 4. 8086 Assembler Directives

**1. Write and Assembly Language Program for Addition of two 8-bit numbers for 8086 µP.**

==========================================================================================

```
ASSUME CS: CODE, DS:DATA
DATA SEGMENT
A   DB        H
B   DB        H
RES DB?
DATA ENDS

CODE SEGMENT
START:  MOV AX, DATA
         MOV DS,AX

      MOV AL, A
      MOV BL, B
      ADD AL, BL

      MOV RES, AL
INT 03H
CODE ENDS
END START
```

------------------------------------------------------------------------------------------

**2. Write and Assembly Language Program for Subtraction of two 8-bit numbers for 8086 µP.**

==========================================================================================

```
ASSUME CS: CODE, DS:DATA
DATA SEGMENT
A   DB        H
B   DB        H
RES DB ?
DATA ENDS
CODE SEGMENT
START:  MOV AX, DATA
         MOV DS,AX
      MOV AL, A
      MOV BL, B

      ____________
      MOV RES, AL
INT 03H
CODE ENDS
END START
```

**3. Write and Assembly Language Program for Multiplication of two 8-bit numbers for 8086 µP.**

======================================================================

```
ASSUME CS: CODE, DS:DATA
DATA SEGMENT
A   DB        H
B   DB        H
RES DB   ?
DATA ENDS

CODE SEGMENT
START:  MOV AX, DATA
        MOV DS,AX
      MOV AL, A
      MOV BL, B
      ___________
      MOV RES, AL
INT 03H
CODE ENDS
END START
```

======================================================================

**4. Write and Assembly Language Program for Division of two 8-bit numbers for 8086 µP.**

======================================================================

```
ASSUME CS: CODE, DS:DATA
DATA SEGMENT
A   DB        H
B   DB        H
RES DB ?
DATA ENDS
CODE SEGMENT
START:  MOV AX, DATA
        MOV DS,AX
      MOV AX,0000H
      MOV AL, A
      MOV BL, B
      ___________
      MOV RES, AL
INT 03H
CODE ENDS
END START
```

==============================================================================

**5. Write and Assembly Language Program for Addition of two 16-bit numbers for 8086 µP.**

==============================================================================

```
ASSUME CS: CODE, DS:DATA
  DATA SEGMENT
        A   DW 5555H
        B   DW 4444H
        RES  DW ?
 DATA ENDS
CODE SEGMENT
START:  MOV AX, DATA
        MOV DS,AX
        MOV AX, A
        MOV BX, B
        ADD AX, BX
        MOV RES, AX
INT 03H
CODE ENDS
END START
```

==============================================================================

**6. Write and Assembly Language Program for Subtraction of two 16-bit numbers for 8086 µP.**

==============================================================================AS

```
SUME CS:CODE, DS:DATA
DATA SEGMENT
A   DW 55H
B   DW 44H
RES DW ?
DATA ENDS

CODE SEGMENT
START:  MOV AX, DATA
        MOV DS,AX
        MOV AX, A
        MOV BX, B
        SUB AX, BX
MOV RES, AX
INT 03H
CODE ENDS
END START
```

==========================================================================

**7. Write and Assembly Language Program for Multiplication of two 16-bit numbers for 8086 µP.**

==========================================================================

```
ASSUME CS:CODE, DS:DATA
DATA SEGMENT
A   DW 5555H ; 0101 0101 0101 0101
B   DW 2244H ; 0010 0010 0100 0100
RES DW ?
DATA ENDS
CODE SEGMENT
START:  MOV AX, DATA
        MOV DS,AX
        MOV AX,0000H
        MOV AX, A
        MOV BX, B
        MUL BX
        MOV RES, AX

INT 03H
CODE ENDS
END START
```

==========================================================================

**8. Write and Assembly Language Program for Division of two 16-bit numbers for 8086 µP.**

==========================================================================

```
ASSUME CS:CODE, DS:DATA
DATA SEGMENT
A   DW 5555H ; 0101 0101 0101 0101
B   DW 2244H ; 0010 0010 0100 0100
RES DW ?
DATA ENDS

CODE SEGMENT
START:  MOV AX, DATA
        MOV DS,AX
        MOV AX,0000H
        MOV AX, A
        MOV BX, B
        DIV BX
        MOV RES, AX
INT 03H
CODE ENDS
END START
```

==========================================================================

## 9. Write and Assembly Language Program for Addition of two 32-bit numbers for 8086 µP.

==========================================================================

```
DATA SEGMENT
A DD 66664444H
B DD 44442222H
C DW ?
DATA ENDS

CODE SEGMENT
ASSUME CS:CODE, DS:DATA
START:  MOV AX,DATA
        MOV DS,AX
        MOV DL,00H
        MOV AX, WORD PTR A
        MOV BX, WORD PTR B
        ADD AX,BX
        MOV WORD PTR C,AX
        MOV AX, WORD PTR A+2
        MOV BX, WORD PTR B+2
        ADC AX,BX
        MOV WORD PTR C+2,AX
INT 03H
CODE ENDS
END START
```

==========================================================================

## 10. Write and Assembly Language Program for Subtraction of two 32-bit numbers for 8086 µP.

==========================================================================

```
DATA SEGMENT
A DD 66664444H
B DD 44442222H
C DW ?
DATA ENDS
 CODE SEGMENT
ASSUME CS:CODE, DS:DATA
START:  MOV AX,DATA
        MOV DS,AX
        MOV DL,00H

        MOV AX, WORD PTR A
        MOV BX, WORD PTR B
        SUB AX,BX
        MOV WORD PTR C,AX

        MOV AX, WORD PTR A+2
        MOV BX, WORD PTR B+2
        SBB AX,BX
        MOV WORD PTR C+2,AX
INT 3
CODE ENDS
END START
```

===============================================================

## 11. Write and Assembly Language Program for Multiplication of two 32-bit numbers for 8086 µP.

===============================================================

```
DATA SEGMENT
A DD 33333333H
B DD 22222222H
C DQ ?
DATA ENDS

CODE SEGMENT
ASSUME CS:CODE, DS:DATA
START:
MOV AX, DATA
MOV DS, AX

MOV AX, WORD PTR A
MUL WORD PTR B
MOV WORD PTR C, AX
MOV CX, DX

MOV AX, WORD PTR A+2
MUL WORD PTR B
ADD CX, AX
MOV BX, DX

JNC MOVE
ADD BX,0001H

MOVE: MOV AX,WORD PTR A
MUL WORD PTR B+2
ADD CX, AX
MOV WORD PTR C+2, CX
MOV CX,DX

JNC MA
ADD BX, 0001H
MA: MOV AX, WORD PTR A+2
MUL WORD PTR B+2
ADD CX, AX

JNC MB
ADD DX, 0001H
MB: ADD CX, BX
MOV WORD PTR C+4, CX

JNC MC
ADD DX, 0001H
MC: MOV WORD PTR C+6, DX
INT 3
CODE ENDS
END START
```

## 12. Write and Assembly Language Program for Division of two 32-bit numbers for 8086 µP.

```
ASSUME CS:CODE, DS:DATA
DATA SEGMENT
A    DD      99999999H
B    DW      2222H
QUOT    DD      ?
RMDR    DW      ?
DATA ENDS

CODE SEGMENT
START:  MOV AX, DATA
              MOV DS,AX
              MOV     CX,B                  ;CX = DVSR
              XOR       DX,DX                 ;DX = 0

MOV   AX,WORD PTR [A+2]           ;AX = HIGH ORDER NUMERATOR
DIV     CX                          ;DX = REM, AX = HIGH ORDER QUOTIENT
MOV   WORD PTR [QUOT+2],AX     ;STORE HIGH ORDER QUOTIENT
MOV   AX,WORD PTR [A]              ;AX = LOW  ORDER NUMERATOR
DIV     CX                          ;DX = REM, AX = LOW  ORDER QUOTIENT
MOV   WORD PTR [QUOT], AX        ;STORE LOW  ORDER QUOTIENT
MOV   WORD PTR [RMDR], DX                         ;STORE REMAINDER

INT 03H
CODE ENDS
END START
```

===================================================================

## 13. Write and Assembly Language Program for Factorial of a number for 8086 µP.

===================================================================

```
ASSUME CS:CODE, DS:DATA
DATA SEGMENT
        A     DB 05H
        RES DB  ?
DATA ENDS

CODE SEGMENT
START:  MOV AX, DATA
            MOV DS,AX
               MOV AX,0001H
               MOV BL, A
 BACK:  MUL BL

        DEC BL
        JNZ BACK
MOV RES, AL
INT 03H
CODE ENDS
END START
```

================================================================================

**13. Write and Assembly Language Program for Addition of two arrays for 8086 µP.**

================================================================================

```
ASSUME CS:CODE, DS:DATA
DATA SEGMENT
ARY1 DB
ARY2 DB 02H,02H,02H,02H,02H
RES  DB 05 DUP(0)
COUNT EQU 05H
DATA ENDS

CODE SEGMENT
START:  MOV AX, DATA
        MOV DS,AX
        MOV AX,0000H
        MOV SI, 0000H
        MOV CL, COUNT

                MOV SI, OFFSET ARY1
BACK:           MOV AL, ARY1[SI]
                ADD AL, ARY2[SI]
                MOV RES[SI], AL
                INC SI
                LOOP BACK


INT 03H

CODE ENDS
END START
```

**14. Write and Assembly Language Program for Subtraction of two arrays for 8086 µP.**

================================================================================

```
ASSUME CS:CODE, DS:DATA
DATA SEGMENT
ARY1 DB
ARY2 DB 02H,02H,02H,02H,02H
RES  DB 05 DUP(0)
COUNT EQU 05H
DATA ENDS

CODE SEGMENT
START:  MOV AX, DATA
      MOV DS,AX
      MOV AX,0000H
      MOV SI, 0000H
      MOV CL, COUNT

              MOV SI, OFFSET ARY1 ;______________________________
BACK:         MOV AL, ARY1[SI]     ;______________________________
              SUB AL, ARY2[SI]     ;______________________________
              MOV RES[SI], AL      ;______________________________
              INC SI
              LOOP BACK


INT 03H

CODE ENDS
END START
```

## 15. Write and Assembly Language Program for Multiplication of two arrays for 8086 µP.

================================================================================================

```
ASSUME CS:CODE, DS:DATA
DATA SEGMENT
ARY1 DB
ARY2 DB 02H,02H,02H,02H,02H
RES  DB 05 DUP(0)
COUNT EQU 05H
DATA ENDS

CODE SEGMENT
START:  MOV AX, DATA
      MOV DS,AX
      MOV AX,0000H
      MOV SI, 0000H
      MOV CL, COUNT

            MOV SI, OFFSET ARY1
BACK:       MOV AL, ARY1[SI]
            MUL, ARY2[SI]
            MOV RES[SI], AL
            INC SI
            LOOP BACK


INT 03H

CODE ENDS
END START
```

**16. Write and Assembly Language Program for Division of two arrays for 8086 µP.**

================================================================================

```
ASSUME CS:CODE, DS:DATA
DATA SEGMENT
ARY1 DB 09H,04H,02H,05H,22H
ARY2 DB 02H,02H,02H,02H,02H
RES  DB 05 DUP(0)
COUNT EQU 05H
DATA ENDS

CODE SEGMENT
START:  MOV AX, DATA
        MOV DS,AX
        MOV AX,0000H
        MOV SI, 0000H
        MOV CL, COUNT

                MOV SI, OFFSET ARY1         ;______________________________
BACK:           MOV AL, ARY1[SI]       ;______________________________
                DIV, ARY2[SI]               ;______________________________
                MOV RES[SI], AL         ;______________________________
                INC SI                      ;______________________________
                LOOP BACK


INT 03H

CODE ENDS
END START
```

**17. Write and Assembly Language Program for 16 bit LOGICATL OPERATIONS for 8086 µP.**

==========================================================================================

**18. Write and Assembly Language Program for Smallest Number in an Array for 8086 µP.**

==================================================================================================

```
ASSUME CS:CODE, DS:DATA
DATA SEGMENT
ARY1 DB 09H,04H,02H,05H,22H
RES  DB ?
COUNT EQU 04H
DATA ENDS

CODE SEGMENT
START:  MOV AX, DATA
       MOV DS,AX
       MOV AX,0000H
       MOV SI, 0000H
       MOV CL, COUNT

         MOV SI, OFFSET ARY1

           MOV AL,[SI]
BACK:  INC SI
           CMP AL, [SI]         ;_______________________________________
           JL NEXT
           MOV AL,[SI]

NEXT: DEC CL
          JNZ BACK

          MOV RES, AL

INT 03H

CODE ENDS
END START
```

**19. Write and Assembly Language Program for Largest Number in an Array for 8086 µP.**

=====================================================================================

```
ASSUME CS:CODE, DS:DATA
DATA SEGMENT
ARY1 DB 09H,04H,02H,05H,22H
RES  DB ?
COUNT EQU 04H
DATA ENDS

CODE SEGMENT
START:  MOV AX, DATA
      MOV DS,AX
      MOV AX,0000H
      MOV SI, 0000H
      MOV CL, COUNT

        MOV SI, OFFSET ARY1

           MOV AL,[SI]
BACK:  INC SI
          CMP AL, [SI]         ;________________________________________
           JL NEXT
          MOV AL,[SI]

NEXT: DEC CL
         JNZ BACK

         MOV RES, AL

INT 03H

CODE ENDS
END START
```

## 20. Write and Assembly Language Program for Ascending order in an Array for 8086 µP.

```
ASSUME CS: CODE, DS:DATA

DATA SEGMENT
ARY1 DB 79H,44H,22H,55H,35H
COUNT EQU 04H
DATA ENDS

CODE SEGMENT
START:  MOV AX, DATA
      MOV DS, AX
      MOV AX,0000H
      MOV SI, 0000H
      MOV CL, COUNT
 UP1:       MOV SI, OFFSET ARY1
            MOV DL, CL

UP2: MOV AL, [SI]
        CMP AL, [SI+1]

        JL DOWN              ;_______________________________

        MOV BL , [SI+1]      ;_______________________________
        MOV [SI+1], AL       ;_______________________________
        MOV [SI], BL         ;_______________________________

DOWN:       INC SI
        DEC DL
        JNZ UP2
DEC CL
JNZ UP1

INT 03H

CODE ENDS
END START
```

**21. Write and Assembly Language Program for <u>Descending order in an Array</u> for 8086 µP.**

===================================================================================

```
ASSUME CS: CODE, DS:DATA

DATA SEGMENT
ARY1 DB 79H,44H,22H,55H,35H
COUNT EQU 04H
DATA ENDS

CODE SEGMENT
START:  MOV AX, DATA
      MOV DS, AX
      MOV AX,0000H
      MOV SI, 0000H
      MOV CL, COUNT
 UP1:        MOV SI, OFFSET ARY1
             MOV DL, CL

UP2:  MOV AL, [SI]
        CMP AL, [SI+1]

        JNL DOWN                ;________________________________________

        MOV BL , [SI+1]         ;________________________________________
        MOV [SI+1] , AL         ;________________________________________
        MOV [SI] , BL           ;________________________________________

DOWN:       INC SI
        DEC DL
        JNZ UP2
DEC CL
JNZ UP1

INT 03H

CODE ENDS
END START
```

## 22. Write and Assembly Language Program for Moving String from one memory to another memory for 8086 µP

==========================================================================================

```
ASSUME CS: CODE, DS:DATA, ES:EXTRA
DATA SEGMENT
STR1 DB 'MECHATRONICS$'
DATA ENDS

EXTRA SEGMENT
STR2 DB ?
EXTRA ENDS

CODE SEGMENT
START:     MOV AX, DATA
        MOV DS, AX

        MOV AX, EXTRA
        MOV ES, AX
        MOV CL, 0CH          ;____________________________________________

MOV SI, OFFSET STR1   ;____________________________________________
MOV DI, OFFSET STR2   ;____________________________________________

CLD

REP MOVSB    ;_____________________________________________________


INT 03H            ;_____________________________________________________
CODE ENDS
END START
```

**23. Write and Assembly Language Program for Reversing String for 8086 µP**

==============================================================================================

```
ASSUME CS: CODE, DS:DATA, ES:EXTRA

DATA SEGMENT
STR1 DB 'MECHATRONICS$'  ;         SCINORTAHCEM
DATA ENDS

EXTRA SEGMENT
STR2 DB ?
EXTRA ENDS

CODE SEGMENT
START:      MOV AX, DATA
            MOV DS, AX        ;_______________________________________

        MOV AX, EXTRA
        MOV ES, AX                ;_______________________________________
MOV CL, 0CH

MOV SI, OFFSET STR1
MOV DI, OFFSET STR2

MOV DI, 20H


BACK:       MOV AL  , [SI]  ;_______________________________________
            MOV [DI],  AL   ;_______________________________________

INC SI
DEC DI

DEC CL
JNZ BACK
INT 03H
CODE ENDS
END START
```

## 24. Write and Assembly Language Program for Inserting Character in a String for 8086 µP

```
======================================================================================

ASSUME CS: CODE, DS:DATA, ES:EXTRA

DATA SEGMENT
STR1 DB 'MECHANICS'
STR2 DB 'TRO'
DATA ENDS

EXTRA SEGMENT
STR3 DB   ?
EXTRA ENDS

CODE SEGMENT
START:      MOV AX, DATA
            MOV DS, AX

        MOV AX, EXTRA
        MOV ES, AX


MOV SI, OFFSET STR1
MOV DI, OFFSET STR3

MOV SI,00H              ;_______________________________
MOV CL,05H              ;_______________________________
REP MOVSB               ;_______________________________

MOV SI, 09H             ;_______________________________
MOV CL,03H              ;_______________________________
REP MOVSB               ;_______________________________

MOV SI, 05H             ;_______________________________
MOV CL, 04H             ;_______________________________
REP MOVSB               ;_______________________________

INT 03H
CODE ENDS
END START
```

**25. Write and Assembly Language Program for Deleting Character from a String for 8086 µP**

==========================================================================

```
ASSUME CS: CODE, DS:DATA, ES:EXTRA

DATA SEGMENT
STR1 DB 'MECHATRONICS'
DATA ENDS

EXTRA SEGMENT
STR2 DB   ?
EXTRA ENDS

CODE SEGMENT
START:     MOV AX, DATA
           MOV DS, AX

       MOV AX, EXTRA
       MOV ES, AX


MOV SI, OFFSET STR1
MOV DI, OFFSET STR2

MOV SI,00H             ;______________________________
MOV CL,05H             ;______________________________
REP MOVSB              ;______________________________

MOV SI, 08H            ;______________________________
MOV CL, 04H            ;______________________________
REP MOVSB              ;______________________________

INT 03H
CODE ENDS
END START
```

## 26. Write and Assembly Language Program for Finding a Character Length in a String for 8086 µP

==========================================================================================

```
ASSUME CS: CODE, DS:DATA

DATA SEGMENT
STR1 DB 'MECHATRONICS$'
MSG DB 'i found N in string$'
LEN DW   ?
DATA ENDS

CODE SEGMENT
START:      MOV AX, DATA
            MOV DS, AX
            MOV SI, OFFSET STR1
            MOV AL,'N'      ;____________________________________
BACK:       CMP AL, [SI]    ;____________________________________
        JZ NEXT
        INC SI
        JMP BACK
NEXT: MOV LEN , SI          ;____________________________________

LEA DX,MSG                  ;____________________________________
MOV AH,09H                  ;____________________________________
INT 21H                     ;____________________________________

INT 03H
CODE ENDS
END START
```

**27. Write and Assembly Language Program for Finding Length of a String for 8086 µP**

========================================================================================

```
ASSUME CS: CODE, DS:DATA

DATA SEGMENT
STR1 DB 'MECHATRONICS$'
LEN DW   ?
DATA ENDS

CODE SEGMENT
START:      MOV AX, DATA
            MOV DS, AX
            MOV SI, OFFSET STR1
            MOV AL,'$'         ;________________________________________
BACK:       CMP AL, [SI]       ;________________________________________
        JZ NEXT
        INC SI
        JMP BACK
NEXT: MOV LEN , SI            ;________________________________________

LEA DX,MSG                     ;________________________________________
MOV AH,09H                     ;________________________________________
INT 21H                        ;________________________________________

INT 03H
CODE ENDS
END START
```

## 28. Write and Assembly Language Program for Finding number of +ve and -ve numbers in an array for 8086 µP

==========================================================================================

```
ASSUME CS: CODE, DS:DATA

DATA SEGMENT
ARY1 DB 23H, 34H, 66H, 10H, 09H, 01H, 45H
DATA ENDS

CODE SEGMENT
START:      MOV AX, DATA
            MOV DS, AX
MOV CL, 07H
MOV SI, OFFSET ARY1
BACK:       MOV AL, [SI]
            SHL AL, 01
      JC NEXT
      INC DL
      JMP AGAIN

NEXT: INC BL

AGAIN:      INC SI
      DEC CL
      JNZ BACK

INT 03H
CODE ENDS
END START
```

**29. Write and Assembly Language Program for** Finding number of **EVEN**and **ODD** numbers in an array **for 8086 µP**

**==============================================================================================**

```
ASSUME CS: CODE, DS:DATA

DATA SEGMENT
ARY1 DB 23H, 34H, 66H, 10H, 09H, 01H, 45H
DATA ENDS

CODE SEGMENT
START:      MOV AX, DATA
            MOV DS, AX
MOV CL, 07H
MOV SI, OFFSET ARY1
BACK:       MOV AL, [SI]
            SHR AL, 01
      JC NEXT
      INC DL
      JMP AGAIN

NEXT: INC BL

AGAIN:      INC SI
      DEC CL
      JNZ BACK

INT 03H
CODE ENDS
END START
```

**30. Write and Assembly Language Program for DISPLAYING System Time for 8086 µP**

====================================================================================================================