MODEL MP LAB

Name: Rahul Ram M

Roll No. : 185001121

Date: 09/11/2020

1. Write an ALP using 8086 to print the system date and time.

Aim:

To write an ALP using 8086 to print the system date and time.

Algorithm:

- Move 2A to AH.
- INT 21 with AH == 2A will get the system date. A -> day, DH -> month and CX -> year(word).
- Move the values in DH (day), DL (month) and CX (year) to corresponding variables declared using MOV instruction.
- Move 2C to AH.
- INT 21 with AH == 2A will get the system time. CH -> hour, CL -> minute and DH -> second.
- Move the values in CH (hour), CL (minute) and DH (second) to corresponding variables declared using MOV instruction.
- INT 21H with AH == 4CH will terminate the program.

CODE:

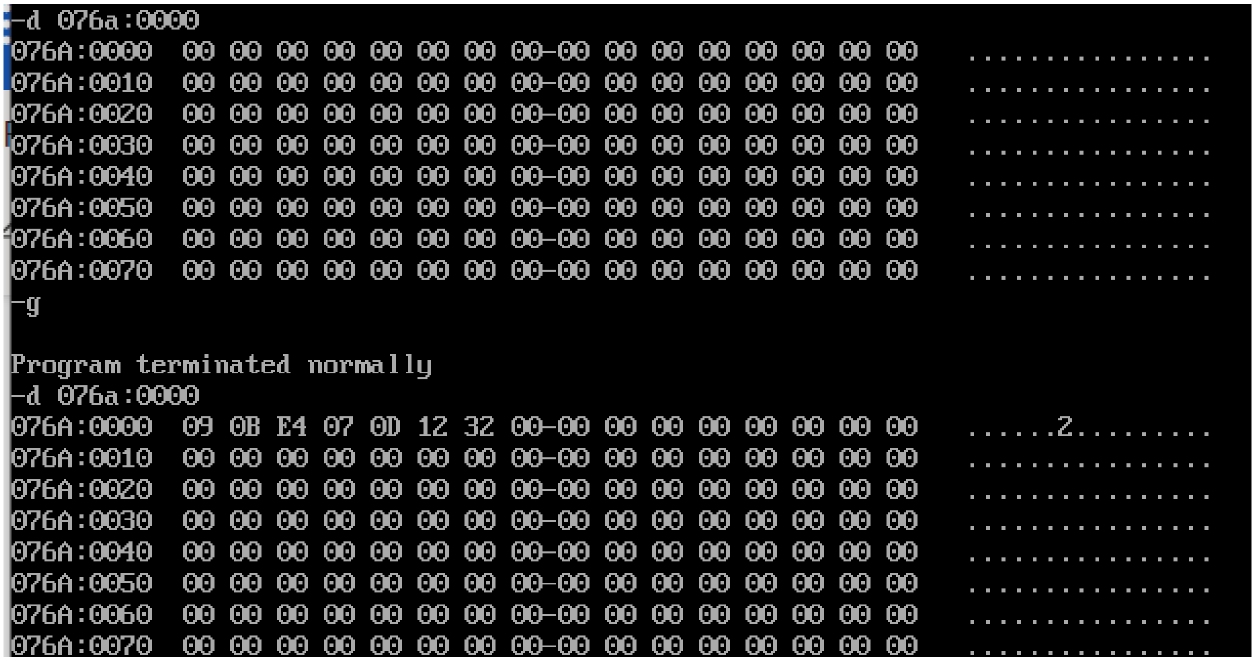
```
assume cs:code,ds:data

data segment

   day db 01 dup(?)

   month db 01 dup(?)

   year db 02 dup(?)

   hour db 01 dup(?)

   minute db 01 dup(?)

   second db 01 dup(?)

data ends
```

```
code segment
        org 0100h
start:    mov ax,data
     mov ds,ax

    ; AH = 2A with INT 21 will give system date
    mov ah,2ah
     int 21q
    mov si, offset day
    mov [si], dl    ; DL = day
    mov si, offset month
    mov [si], dh    ; DH = month
    mov si,offset year
     mov [si],cx    ; CX = year

    ; AH = 2C with INT 21 will give system time
    mov ah, 2ch
    int 21h
    mov si,offset hour
     mov [si],ch    ; CH = hour
    mov si,offset minute
     mov [si],cl    ; Cl = minute
    mov si,offset second
     mov [si],dh    ; DH = second
    ; Ah = 4c with INT 21 - terminates the program.
    mov ah,4ch
    int 21h
```

code ends

end start

OUTPUT:



```
-d 076a:0000
076A:0000  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
-g

Program terminated normally
-d 076a:0000
076A:0000  09 0B E4 07 0D 12 32 00-00 00 00 00 00 00 00 00   ......2.........
076A:0010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
```

Result:

ALP using 8086 to print system date and time is executed successfully.

2.Write an ALP using 8051-microprocessor to convert BCD to ASCII.

AIM:

To write an ALP using 8051-microprocessor to convert BCD to ASCII.

Algorithm:

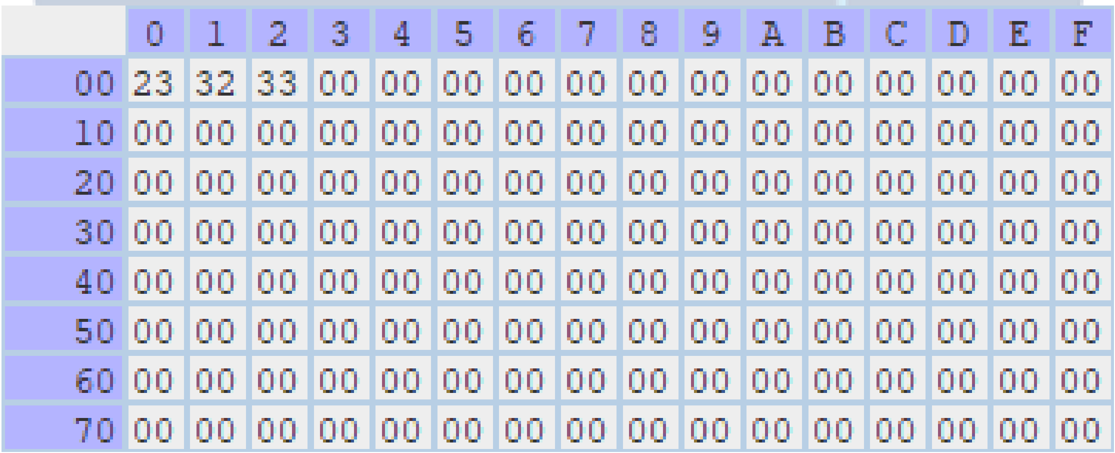
- R0 contains the BCD value (XY).
- Move the value in R0 to register A.
- ANL A, #0F0H will perform bitwise AND operation on A with respect to F0H.
- SWAP A -> will swap the higher and lower order 4 bits of A. (X0H -> 0XH).
- Adding 30H to A using ADD instruction will give the ASCII equivalent of X.
- Store the result in R1 using MOV instruction.
- Move the value in R0 to register A.
- ANL A, #0FH will perform bitwise AND operation on A with respect to 0FH.

- Adding 30H to A using ADD instruction will give the ASCII equivalent of Y.
- Store the result in R2 using MOV instruction.
- INT 21H with AH == 4CH will terminate the program.

CODE:

```
    MOV A, R0    ; ASCII equivalent of first digit.
   ANL A, #0F0H
   SWAP A
   ADD A, #30H
   MOV R1, A
   MOV A, R0    ; ASCII equivalent of second digit.
   ANL A, #0FH
   ADD A, #30H
   MOV R2, A
HERE:     SJMP HERE
```

OUTPUT:



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	23	32	33	00	00	00	00	00	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Result:

ALP using 8051 for BCD to ASCII is executed successfully.