UCS1512 – Microprocessors Lab

Case Conversion

Exp no : 8 Name: Sreedhar V

Date : 25-10-2020 Reg no: 185001161

AIM:

To program and execute the program for inverting the case of the letter on the fly in 8086 using an emulator.

Case Conversion:

Algorithm:

- Program is set to run from any specified memory position.
- Move the address of data segment to register DS .
- Set the value of the count to be 10 using equ directive(for 10 counts).
- Transfer the value of count to CX register.
- Read the input by setting AH to 1 and executing INT 21h
- Check whether the read character is lowercase or uppercase.
- If it's in uppercase, convert to lowercase by adding 20h
- If it's in lowercase, convert to uppercase by subtracting 20h
- Repeat these steps till CX becomes 0
- Terminate the program.

Program:

CODE	COMMENT
Program for Case conversion:	
assume cs:code,ds:data data segment count equ 10h data ends	Data segment is initialized count is initialized to 10h
code segment start : mov ax,data mov ds,ax	Code segment begins Address of the data is transferred to AX , from AX transferred to DS.
mov cx,count l1 : mov ah,1 int 21h	Move count to CX register Move 1 to AH , when AH = 1, a character is read from standard input and echoed back to the standard output.

cmp al,60h
jnc upper
add al,20h
jmp skip
	Compare AL with 96(60h) to decide whether it is lower case or upper case
	
	Jump to “Upper” if no carry is generated
	Add AL by 20h will convert to lower case
	Jump to Skip

upper : sub al,20h skip : mov ah,2h	Sub AL by 20h will convert to upper case AH = 2 will give the output
mov dl,al int 21h loop l1	Move AL to DL , loop from l1 till CX becomes 0
mov ah,4ch int 21h code ends end start	Store the result Program terminates

Unassembled code:

```
DOSBox 0.74-3, Cpu speed:    3000 cycles, Frameskip  0, Progra...
D:\>debug 8.exe
-u
076A:0000 B86A07        MOV     AX,076A
076A:0003 8ED8          MOV     DS,AX
076A:0005 B91000        MOV     CX,0010
076A:0008 B401          MOV     AH,01
076A:000A CD21          INT     21
076A:000C 3C60          CMP     AL,60
076A:000E 7304          JNB     0014
076A:0010 0420          ADD     AL,20
076A:0012 EB02          JMP     0016
076A:0014 2C20          SUB     AL,20
076A:0016 B402          MOV     AH,02
076A:0018 8AD0          MOV     DL,AL
076A:001A CD21          INT     21
076A:001C E2EA          LOOP    0008
076A:001E B44C          MOV     AH,4C
-
```

Execution:

```
DOSBox 0.74-3, Cpu speed:    3000 cycles, Frameskip  0, Progra...
-g
gGhH
-jJkKaA
-aA
-bB
-Bb
-nN
-aA
Program terminated normally
-
```

Result:

Case conversion is executed and verified using an emulator.

UCS1512 – Microprocessors Lab

Floating point operations

Exp no : 9 Name: Sreedhar V

Date : 25-10-2020 Reg no: 185001161

AIM:

To program and execute the code for floating point operations like addition and subtraction in 8086 using an emulator.

Floating point addition:

Algorithm:

- Program is set to run from any specified memory position.
- Move the address of data segment to register DS .
- Use define double word(dd) directive to declare the variable -x,y and sum and initialize them.
- Initialize the floating point unit of 8087 ‘s stack using FINIT
- Load the contents x,y to stack using FLD
- Add the contents of the stack using FADD
- Transfer the result from the stack to the variable sum.
- Terminate the program.

Program:

CODE	COMMENT
Program Floating point addition:	
assume cs:code,ds:data	
data segment	
org 00h	Data segment is initialized
x dd 20.4375	x and y are declared and initialized with values 20.4375 and 20.4375 respectively
org 10h	
y dd 20.4375	
org 20h	
sum dd ?	sum is declared
data ends	
code segment	Code segment begins
start : mov ax,data	Address of the data is transferred to AX , from AX transferred to DS.
mov ds,ax	
finit	Initialize the floating point unit stack of 8087.
fld x	loading x to ST(0) //stack
fld y	loading y to ST(0)
fadd st(0),st(1)	Adding the stack contents
fst sum	Storing ST(0) to sum.
mov ah,4ch	Program terminates
int 21h	
code ends	
end start	

Unassembled code:

```
DOSBox 0.74-3, Cpu speed:    3000 cycles, Frameskip  0, Progra...
D:\>debug 9a.exe
-u
076D:0000 B86A07        MOV     AX,076A
076D:0003 8ED8          MOV     DS,AX
076D:0005 9B            WAIT
076D:0006 DBE3                  FINIT
076D:0008 9B            WAIT
076D:0009 D9060000              FLD     DWORD PTR [0000]
076D:000D 9B            WAIT
076D:000E D9061000              FLD     DWORD PTR [0010]
076D:0012 9B            WAIT
076D:0013 D8C1                  FADD    ST,ST(1)
076D:0015 9B            WAIT
076D:0016 D9162000              FST     DWORD PTR [0020]
076D:001A B44C          MOV     AH,4C
076D:001C CD21          INT     21
076D:001E 0000          ADD     [BX+SI],AL
-
-
-
-
-
-
-
-
```

Execution:

```
DOSBox 0.74-3, Cpu speed:    3000 cycles, Frameskip  0, Progra...
-d 076a:0000
076A:0000  00 80 A3 41 00 00 00 00-00 00 00 00 00 00 00 00   ...A............
076A:0010  00 80 A3 41 00 00 00 00-00 00 00 00 00 00 00 00   ...A............
076A:0020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0030  B8 6A 07 8E D8 9B DB E3-9B D9 06 00 00 9B D9 06   .j..............
076A:0040  10 00 9B D8 C1 9B D9 16-20 00 B4 4C CD 21 00 00   ........ ..L.!..
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  00 80 A3 41 00 00 00 00-00 00 00 00 00 00 00 00   ...A............
076A:0010  00 80 A3 41 00 00 00 00-00 00 00 00 00 00 00 00   ...A............
076A:0020  00 80 23 42 00 00 00 00-00 00 00 00 00 00 00 00   ..#B............
076A:0030  B8 6A 07 8E D8 9B DB E3-9B D9 06 00 00 9B D9 06   .j..............
076A:0040  10 00 9B D8 C1 9B D9 16-20 00 B4 4C CD 21 00 00   ........ ..L.!..
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:

Floating point addition is executed and verified using an emulator.

Floating point subtraction:

Algorithm:

- Program is set to run from any specified memory position.
- Move the address of data segment to register DS .
- Use define double word(dd) directive to declare the variable -x,y and sum and initialize them.
- Initialize the floating point unit of 8087 ‘s stack using FINIT
- Load the contents x,y to stack using FLD
- Add the contents of the stack using FSUB
- Transfer the result from the stack to the variable diff.
- Terminate the program.

Program:

CODE	COMMENT
Program Floating point subtraction:	
assume cs:code,ds:data data segment org 00h x dd 20.4375 org 10h y dd 0.125 org 20h diff dd ? data ends	Data and code segment is initialized x and y are declared and initialized with values 20.4375 and 20.4375 respectively diff is declared
code segment start : mov ax,data mov ds,ax	Code segment begins Address of the data is transferred to AX , from AX transferred to DS.
finit fld x fld y	Initialize the floating point unit stack of 8087. loading x to ST(0) //stack loading y to ST(0)
fsub st(0),st(1)	Subtracting the stack contents
fst diff	Storing ST(0) to diff.
mov ah,4ch int 21h code ends end start	Program terminates

Unassembled code:

```
DOSBox 0.74-3, Cpu speed:    3000 cycles, Frameskip  0, Progra...
D:\>debug 9b.exe
-u
076D:0000 B86A07        MOV     AX,076A
076D:0003 8ED8          MOV     DS,AX
076D:0005 9B            WAIT
076D:0006 DBE3                  FINIT
076D:0008 9B            WAIT
076D:0009 D9060000              FLD     DWORD PTR [0000]
076D:000D 9B            WAIT
076D:000E D9061000              FLD     DWORD PTR [0010]
076D:0012 9B            WAIT
076D:0013 D8E1                  FSUB    ST,ST(1)
076D:0015 9B            WAIT
076D:0016 D9162000              FST     DWORD PTR [0020]
076D:001A B44C          MOV     AH,4C
076D:001C CD21          INT     21
076D:001E 0000          ADD     [BX+SI],AL
-
-
-
-
-
-
-
-
```

Execution:

```
DOSBox 0.74-3, Cpu speed:    3000 cycles, Frameskip  0, Progra...
-d 076a:0000
076A:0000  00 80 A3 41 00 00 00 00-00 00 00 00 00 00 00 00   ...A............
076A:0010  00 00 00 3E 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  B8 6A 07 8E D8 9B DB E3-9B D9 06 00 00 9B D9 06   .j..............
076A:0040  10 00 9B D8 E1 9B D9 16-20 00 B4 4C CD 21 00 00   ........ ..L.!..
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  00 80 A3 41 00 00 00 00-00 00 00 00 00 00 00 00   ...A............
076A:0010  00 00 00 3E 00 00 00 00-00 00 00 00 00 00 00 00   ...>............
076A:0020  00 80 A2 C1 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0030  B8 6A 07 8E D8 9B DB E3-9B D9 06 00 00 9B D9 06   .j..............
076A:0040  10 00 9B D8 E1 9B D9 16-20 00 B4 4C CD 21 00 00   ........ ..L.!..
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:

Floating point subtraction is executed and verified using an emulator.

UCS1512 – Microprocessors Lab

Display a String

Exp no : 1 0 Name: Sreedhar V

Date : 25-10-2020 Reg no: 185001161

AIM:

To program and execute the code for displaying a string in the standard output in 8086 using an emulator.

Display a string:

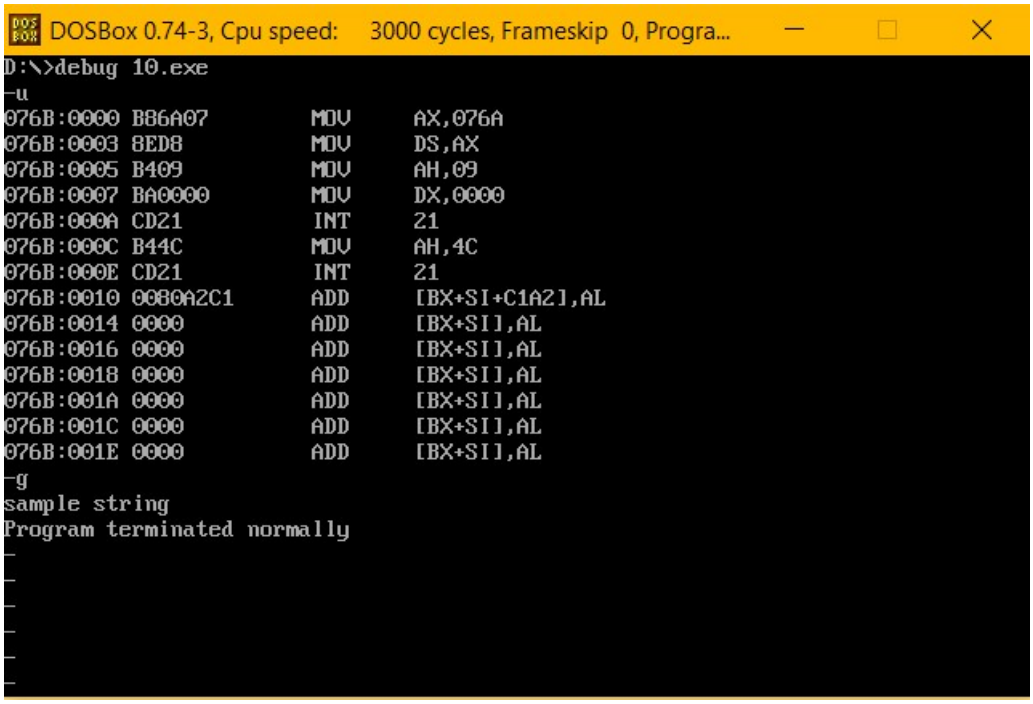
Algorithm:

- Program is set to run from any specified memory position.
- Move 09h into the AH register and move the offset address of the variable which stores the string into the DX register
- Use Interrupt 21h to display the string into standard output stream
- Terminate the program

Program:

CODE	COMMENT
;Program for displaying a string.	
assume cs:code,ds:data	Data and code segment initialized
data segment message db "sample string$" data ends	message is declared and initialized to “sample string”
code segment start : mov ax,data mov ds,ax	Code segment begins Address of data segment moved to AX , from AX transferred to DS.
mov ah,9h mov dx,offset message	Move 9H to AH and offset value of the message to the DX register
int 21h	Call int 21h to display the message
mov ah,4ch int 21h code ends end start	Program terminates

Unassembled code and Execution:



Result:

Displaying a string is executed and verified using an emulator.

UCS1512 – Microprocessors Lab

Display system date and time

Exp no : 1 1 Name: Sreedhar V

Date : 25-10-2020 Reg no: 185001161

AIM:

To program and execute the code for displaying the system date and time in 8086 using an emulator.

Display system date:

Algorithm:

- Program is set to run from any specified memory position.
- Declare variables for day, month and year.
- Move 2Ah into AH register and execute int 21h to get the system date.
- After executing this function,move the value of day available in DL register to day, value of month available in DH to month and the value of year available in CX to year.
- Terminate the program.

Program:

CODE	COMMENT
Program Displaying the date:	
assume cs:code,ds:data data segment day db 01 dup(?) month db 01 dup(?) year db 02 dup(?) data ends	Data and Code segment is initialized Declare 1 byte to day Declare 1 byte to month Declare 2 byte to year //Uninitialized
code segment start: mov ax,data mov ds,ax	Code segment begins Address of the data is transferred to AX , from AX transferred to DS.
mov ah,2ah int 21h	Move 2ah to AH and execute int 21 to get the system date

mov si,offset day
mov [si],dl
mov si,offset month
mov [si],dh
mov si,offset year
mov [si],cx
	Value of the day , month and year are stored in DL ,DH and CX registers respectively
	
	Move the contents of DL,DH,CX registers to the offset value of variables declared to store them.(DL to day , DH to month ,CX to year).

mov ah,4ch int 21h code ends end start	Program terminates

Unassembled code:

```
DOSBox 0.74-3, Cpu speed:    3000 cycles, Frameskip  0, Progra...
D:\>debug 11a.exe
-u
076B:0000 B86A07        MOV     AX,076A
076B:0003 8ED8          MOV     DS,AX
076B:0005 B42A          MOV     AH,2A
076B:0007 CD21          INT     21
076B:0009 BE0000        MOV     SI,0000
076B:000C 8814          MOV     [SI],DL
076B:000E BE0100        MOV     SI,0001
076B:0011 8834          MOV     [SI],DH
076B:0013 BE0200        MOV     SI,0002
076B:0016 890C          MOV     [SI],CX
076B:0018 B44C          MOV     AH,4C
076B:001A CD21          INT     21
076B:001C 0000          ADD     [BX+SI],AL
076B:001E 0000          ADD     [BX+SI],AL
```

Execution:

```
DOSBox 0.74-3, Cpu speed:    3000 cycles, Frameskip  0, Progra...
-d 076a:0000
076A:0000  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0010  B8 6A 07 8E D8 B4 2A CD-21 BE 00 00 88 14 BE 01   .j....*.!.......
076A:0020  00 88 34 BE 02 00 89 0C-B4 4C CD 21 00 00 00 00   ..4......L.!....
076A:0030  B8 6A 07 8E D8 9B DB E3-9B D9 06 00 00 9B D9 06   .j..............
076A:0040  10 00 9B D8 E1 9B D9 16-20 00 B4 4C CD 21 00 00   ........ ..L.!..
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  1B 0A E4 07 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0010  B8 6A 07 8E D8 B4 2A CD-21 BE 00 00 88 14 BE 01   .j....*.!.......
076A:0020  00 88 34 BE 02 00 89 0C-B4 4C CD 21 00 00 00 00   ..4......L.!....
076A:0030  B8 6A 07 8E D8 9B DB E3-9B D9 06 00 00 9B D9 06   .j..............
076A:0040  10 00 9B D8 E1 9B D9 16-20 00 B4 4C CD 21 00 00   ........ ..L.!..
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:

Displaying system date is executed and verified using an emulator.

Display system time:

Algorithm:

- Program is set to run from any specified memory position.
- Declare variables for day, month and year.
- Move 2Ch into AH register and execute int 21h to get the system time.
- After executing this function,move the value of hour available in CH register to day, value of minute available in CL register to month and the value of second available in DH register to year.
- Terminate the program.

Program:

CODE	COMMENT
Program Displaying the time:	
assume cs:code,ds:data data segment hour db ? minute db ? second db ? data ends	Data and Code segment is initialized Declare 1 byte to hour Declare 1 byte to minute Declare 1 byte to second //Uninitialized
code segment org 0100h start: mov ax,data mov ds,ax	Code segment begins Address of the data is transferred to AX , from AX transferred to DS.
mov ah,2ch int 21h	Move 2ch to AH and execute int 21 to get the system time

mov si,offset hour
mov [si],ch
mov si,offset minute
mov [si],cl
mov si,offset second
mov [si],dh
	Value of the hour , minute and second are stored in CH ,CL and DH registers respectively
	
	Move the contents of CH,CL and DH registers to the offset value of variables declared to store them.(CH to hour, CL to minute , DH to second).

mov ah,4ch int 21h code ends end start	Program terminates

Unassembled code:

```
DOSBox 0.74-3, Cpu speed:    3000 cycles, Frameskip  0, Progra...
D:\>debug 11b.exe
-u
076B:0100 B86A07        MOV     AX,076A
076B:0103 8ED8          MOV     DS,AX
076B:0105 B42C          MOV     AH,2C
076B:0107 CD21          INT     21
076B:0109 BE0000        MOV     SI,0000
076B:010C 882C          MOV     [SI],CH
076B:010E BE0100        MOV     SI,0001
076B:0111 880C          MOV     [SI],CL
076B:0113 BE0200        MOV     SI,0002
076B:0116 8834          MOV     [SI],DH
076B:0118 B44C          MOV     AH,4C
076B:011A CD21          INT     21
076B:011C 0000          ADD     [BX+SI],AL
076B:011E 0000          ADD     [BX+SI],AL
-
-
-
-
-
-
-
-
-
```

Execution:

```
DOSBox 0.74-3, Cpu speed:    3000 cycles, Frameskip  0, Progra...
-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  14 38 36 00 00 00 00 00-00 00 00 00 00 00 00 00   .86.............
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:

Displaying system time is executed and verified using an emulator.