MICROPROCESSOR ASSIGNMENTS

NAME : SHRIRANG. R. MHALGI

CLASS : S.E.

DIVISION : B

ROLL NO 222006

PROBLEM STATEMENT :

Write X86/64 ALP to convert 4-digit Hex number into its equivalent BCD number and 5-digit BCD number into its equivalent HEX number. Make your program user friendly to accept the choice from user for: (a) HEX to BCD b) BCD to HEX (c) EXIT. Display proper strings to prompt the user while accepting the input and displaying the result. (wherever necessary, use 64-bit registers).

CODE :

%macro scall 4

mov rax, %1

mov rdi, %2

mov rsi, %3

mov rdx, %4

syscall

%endmacro

section .data

menu\_msg db 10, "1 : HEX TO BCD "

db 10, "2 : BCD TO HEX "

db 10, "0 : EXIT "

db 10, ">> "

menu\_msgl equ $-menu\_msg

msg\_ip db 10, "ENTER A 4 DIGIT HEX NO "

msg\_ipl equ $-msg\_ip

msg\_ip1 db 10, "ENTER A 5 DIGIT BCD NO "

msg\_ip1l equ $-msg\_ip1

msg\_op db 10, "EQUIVALENT BCD NO IS :- "

msg\_opl equ $-msg\_op

msg\_op1 db 10, "EQUIVALENT HEX NO IS :- "

msg\_op1l equ $-msg\_op1

o db "ok"

ol equ $-o

section .bss

choice resb 10

buff resb 10

buff\_l equ $-buff

digitcnt resb 10

hex\_no resb 10

bcd\_no resb 10

fact resw 2

ans resb 4

section .txt

global \_start

\_start :

scall 1, 1, menu\_msg, menu\_msgl

scall 0, 0, choice, 10

cmp byte[choice], 31h

je case1

cmp byte[choice], 32h

je case2

cmp byte[choice], 30h

je case3

case1 : call h2b

call exit

case2 : call b2h

call exit

case3 : call exit

exit : mov rax, 60

mov rdi, 0

syscall

h2b :

mov byte[digitcnt], 0

scall 1, 1, msg\_ip, msg\_ipl

scall 0, 0, hex\_no, 10

mov rsi, hex\_no

call accept\_proc

mov [hex\_no], rdx

mov bx, [hex\_no]

mov rsi, buff

mov ax, bx

mov rbx, 10

back : xor rdx, rdx

div rbx

push dx

inc byte[digitcnt]

cmp rax, 0

jnz back

popins : pop dx

add dl, 30h

mov [rsi], dl

inc rsi

dec byte[digitcnt]

jnz popins

scall 1, 1, msg\_op, msg\_opl

scall 1, 1, buff, 10

ret

;--------------------------------------------------

b2h : scall 1, 1, msg\_ip1, msg\_ip1l

scall 0, 0, bcd\_no, 6

mov rsi, bcd\_no+4

mov byte[digitcnt], 5

mov word[fact], 1

mov bx,00

bck : mov ax,0

mov al, [rsi]

sub al, 30h

mul word[fact]

add bx, ax

mov ax,10

mul word[fact]

mov word[fact],ax

dec rsi

dec byte[digitcnt]

jnz bck

call display\_proc

scall 1, 1, ans, 4

ret

accept\_proc :

mov rcx, 04h

mov rdx, 0

mov rbx, 0

up : rol rdx, 4

mov bl, [rsi]

cmp bl, 39h

jbe l1

sub bl, 07h

l1 : sub bl, 30h

add rdx, rbx

inc rsi

dec rcx

jnz up

ret

display\_proc :

scall 1, 1, msg\_op1, msg\_op1l

mov rsi, ans

mov rcx, 4

;mov rbx, 0

mov rax, 0

upp : rol bx, 4

mov al,bl

and al,0fh

cmp al, 09h

jbe l2

add al, 07h

l2 : add al, 30h

mov [rsi], al

inc rsi

dec rcx

jnz upp

ret

OUTPUT :

