EXPERIMENT No.6

Name: Sadhya Kashid

Roll No.: 12131

Subject : Microprocessor

section .data

menumsg db 10,10,'###### Menu for Code Conversion ######'

db 10,'1: Hex to BCD'

db 10,'2: BCD to Hex'

db 10,'3: Exit'

db 10,10,'Please Enter Choice::'

menumsg\_len equ $-menumsg

hexinmsg db 10,10,'Please enter 4 digit hex number::'

hexinmsg\_len equ $-hexinmsg

bcdopmsg db 10,10,'BCD Equivalent::'

bcdopmsg\_len equ $-bcdopmsg

bcdinmsg db 10,10,'Please enter 5 digit BCD number::'

bcdinmsg\_len equ $-bcdinmsg

hexopmsg db 10,10,'Hex Equivalent::'

hexopmsg\_len equ $-hexopmsg

section .bss

numascii resb 06 ; Buffer for choice, hex, and bcd input

outputbuff resb 08 ; Output buffer for BCD or Hex

dispbuff resb 08 ; Display buffer

%macro display 2

mov rax, 1

mov rdi, 1

mov rsi, %1

mov rdx, %2

syscall

%endmacro

%macro accept 2

mov rax, 0

mov rdi, 0

mov rsi, %1

mov rdx, %2

syscall

%endmacro

section .text

global \_start

\_start:

menu:

display menumsg, menumsg\_len

accept numascii, 2 ; Accept choice input (1, 2, or 3)

cmp byte [numascii], '1'

je hex2bcd\_proc

cmp byte [numascii], '2'

je bcd2hex\_proc

cmp byte [numascii], '3'

je exit

jmp \_start

exit:

mov rax, 60 ; syscall number for exit

mov rbx, 0 ; exit status

syscall

hex2bcd\_proc:

display hexinmsg, hexinmsg\_len

accept numascii, 5 ; Accept 4 digit hex number

call packnum ; Convert hex to packed BCD

mov ax, bx ; Move result to ax

mov rcx, 0

mov bx, 10 ; Base of Decimal Number system

h2bup1:

mov dx, 0

div bx

push rdx

inc rcx

cmp ax, 0

jne h2bup1

mov rdi, outputbuff

h2bup2:

pop rdx

add dl, 30h

mov [rdi], dl

inc rdi

loop h2bup2

display bcdopmsg, bcdopmsg\_len

display outputbuff, 5

jmp menu

bcd2hex\_proc:

display bcdinmsg, bcdinmsg\_len

accept numascii, 6 ; Accept 5 digit BCD number

display hexopmsg, hexopmsg\_len

mov rsi, numascii

mov rcx, 5 ; 5 digits in BCD

mov rax, 0

mov ebx, 0ah ; Base for BCD

b2hup1:

mov dx, 0

mul ebx

mov dl, [rsi]

sub dl, 30h ; Convert ASCII to integer

add rax, rdx

inc rsi

loop b2hup1

mov ebx, eax ; Store result in ebx

call disp32\_num

jmp menu

packnum:

mov bx, 0

mov ecx, 4

mov esi, numascii

up1:

rol bx, 4

mov al, [esi]

cmp al, 39h

jbe skip1

sub al, 07h

skip1:

sub al, 30h

add bl, al

inc esi

loop up1

ret

disp32\_num:

mov rdi, dispbuff ; Point to display buffer

mov rcx, 8 ; Number of digits to display

dispup1:

rol ebx, 4 ; Rotate left by 4 bits

mov dl, bl ; Move lower byte in dl

and dl, 0fh ; Mask upper digit of byte in dl

add dl, 30h ; Convert to ASCII

cmp dl, 39h

jbe dispskip1

add dl, 07h

dispskip1:

mov [rdi], dl

inc rdi ; Move to the next byte

loop dispup1 ; Repeat until rcx reaches 0

display dispbuff + 3, 5 ; Display only lower 5 digits (upper 3 are 0)

ret

OUTPUT :

##### Menu for Code Conversion ######

1: Hex to BCD

2: BCD to Hex

3: Exit

Please Enter Choice::1

Please enter 4 digit hex number::000A

BCD Equivalent::10