



Digital Design (22COMP05C)

16-bit Decimal Calculator

Submitted by:

Sara Hesham 207792

Ezzat Mazhar 218404

Mariem Ashraf 214761

Seifeldin Amr 218359

Abdelrahman Hassan 208558

Year Two, Computer Engineering Department

Submitted to:

Dr. Mervat. A. Mahmoud

Eng. Eman Ahmed

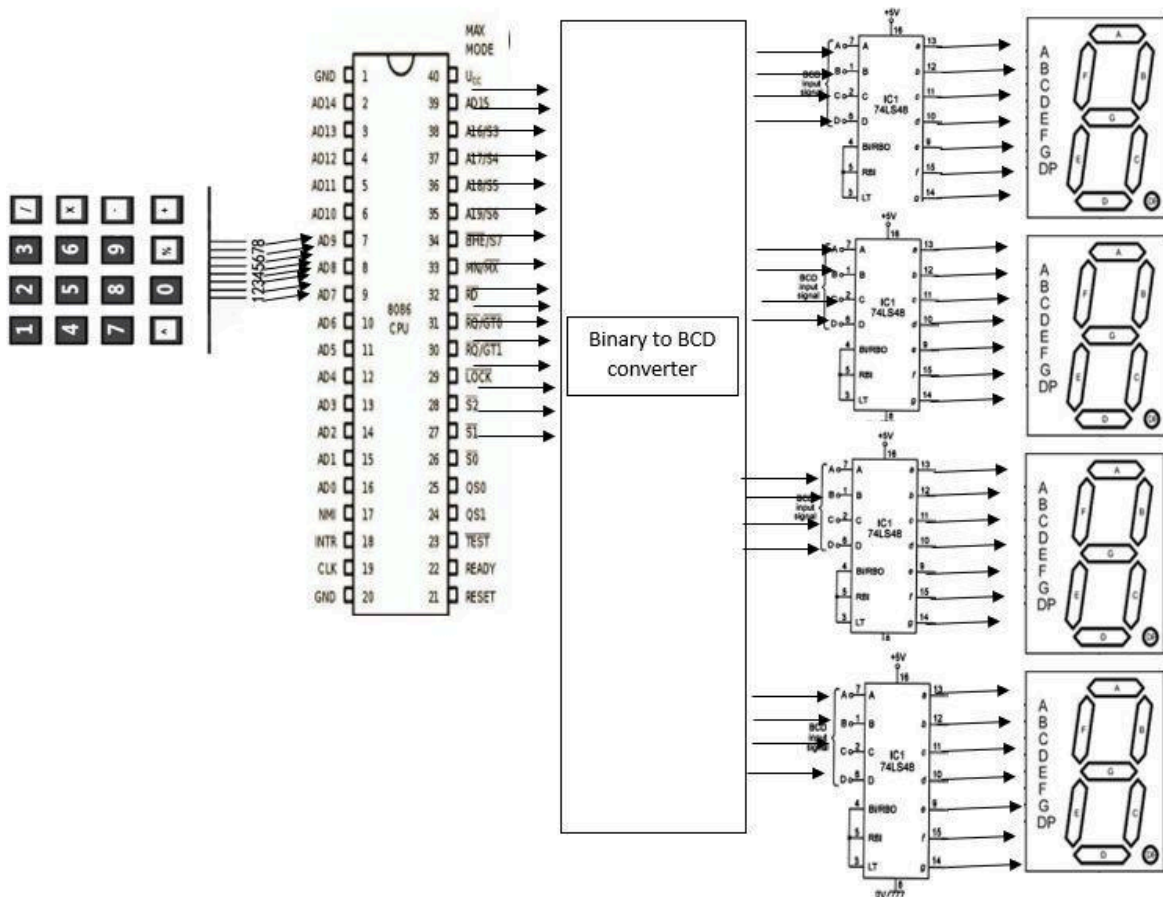
Computer Engineer Department

DEC 2, 2022

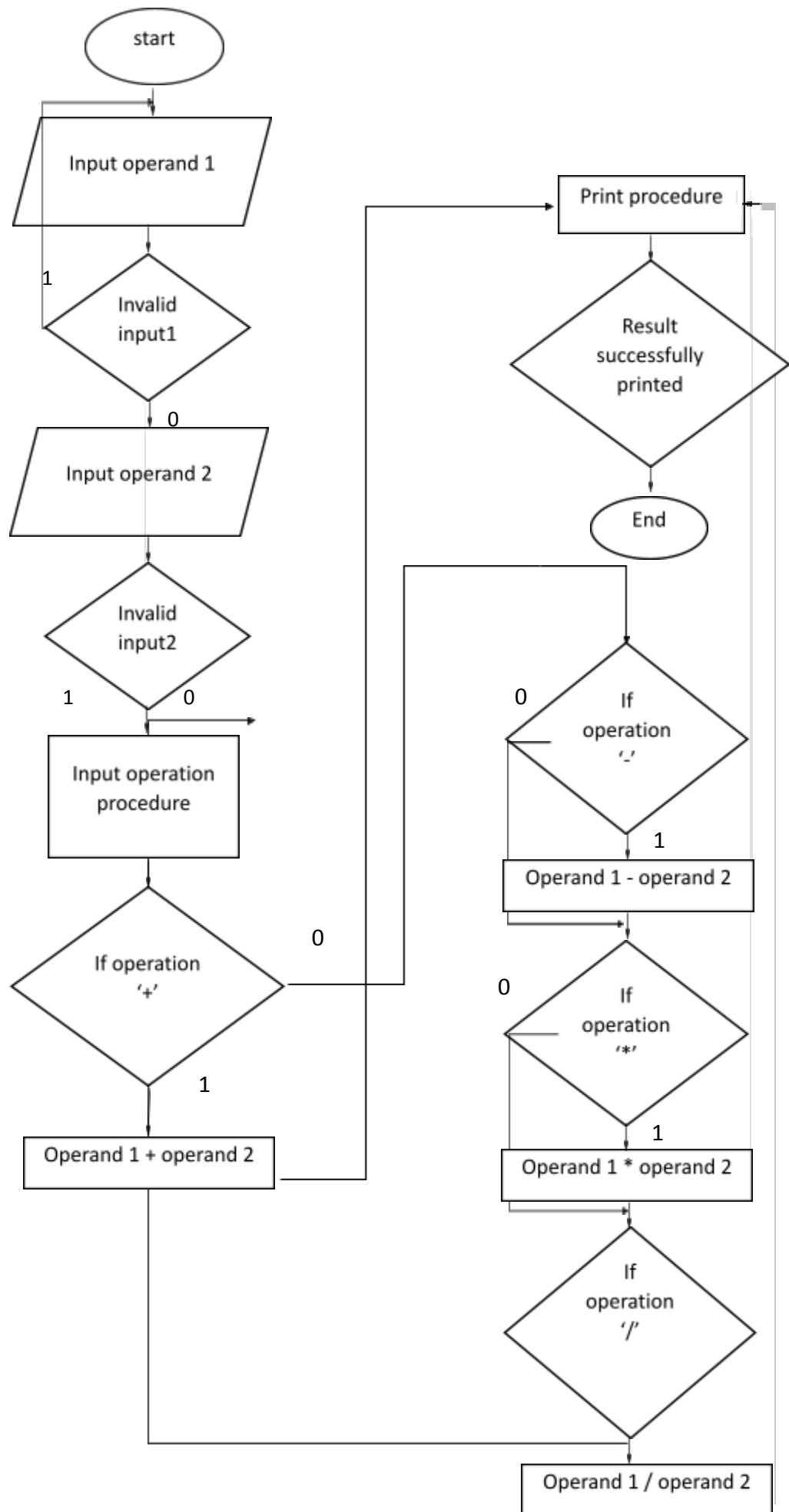
● Introduction

This program is designed to perform some useful functions such as addition, subtraction, division, and multiplication. It acts as a 16-bit decimal calculator. The user proceeds by inputting two inputs and an operator, and the program works to determine whether those three inputs are valid before delivering the output after the data processing is completed. There is also a first input (ranging from 0 to 65535) and a second input (from 0 to 65535). The user uses a keypad to enter the 3 inputs, and the outputs are displayed on the 7-segment screen.

• Block diagram



1. The user proceeds to enter 2 inputs and a 3rd input (operator) from the keypad.
2. The inputs are then transferred to the CPU (Which in this case tis the x8086).
3. The CPU then verifies that the inputs are valid and then it processes the data and delivers the output.
4. The data passes to the Binary to BCD converter to convert a binary (Base-2) number to a BCD (Binary-coded decimal).
5. Every 4 bits of bcd are sent to a decoder, which illuminates one or more of the 7 segments that make up the numerical digit to be displayed (on the 7 segment display).



● Flow chart

Each number input should be equal to 16 bit, else it will not work it. It would be an invalid number and the entry procedure of the invalid number will be repeated (that is user is given a chance to enter the first number again if it was invalid and second number again if it was invalid).

Then the operator will be checked to be sure it is one of the four valid operators (+, -, *, /).

if it is valid the operation will proceed, else it would output invalid operator and repeat the operator entry step. The valid and invalid check of these inputs is shown in the flow chart as 1 and 0 .

- Input procedure:
Reading the number
stores digits in memory.
- Operation procedures;
Read operator and perform the appropriate operation
storing result in memory .
- Print Procedure:
Prints the results on screen.

Code:

```
TITLE MASM Template                                (main.asm)

INCLUDE Irvine32.inc
.data

Msg1 byte "please enter the first number ",0
Msg2 byte "please enter your desired operator ",0
Msg3 byte "please enter the second number ",0
Msg4 byte "invalid operator ",0
Msg5 byte "Invalid number 1",0
Msg6 byte "Invalid number 2 ",0

Num1 Dword ?
Num2 Dword ?
Char byte ?

.code
main PROC
call Clrscr

Beginning:
Mov edx, offset Msg1
Call WriteString
Call Crlf
Call Readint
Mov Num1 , eax
Cmp eax,65536
Ja invalid1
Jmp step2
    Invalid1:
Mov edx,OFFSET Msg5
Call WriteString
Call crlf
Jmp Beginning

    Step2:
Mov edx, offset Msg3
Call Writestring
call Crlf
call Readint
Mov Num2, eax
Cmp eax,65536
Ja Invalid2
Jmp step3

Invalid2:
Mov edx,OFFSET Msg6
Call WriteString
```

```
Call crlf
Jmp Step2
```

```
Step3:
Mov edx, offset Msg2
Call Writestring
call crlf
```

```
Call Readchar
Call writechar
Call crlf
Mov char, al
Cmp al, "+"
Jz addition
Cmp al, "-"
Jz subtraction
Cmp al, "*"
Jz multiplication
Cmp al, "/"
Jz divison
jmp invalidoperator
```

```
invalidoperator:
mov edx , OFFSET Msg4
call writestring
call crlf
jmp step3
```

```
addition:
Mov eax,0
mov eax,num1
add eax,num2
call writeint
call crlf
jmp ending
```

```
subtraction:
Mov eax,0
mov eax,num1
sub eax,num2
call writeint
call crlf
jmp ending
```

```
multiplication:
Mov edx ,0
Mov eax,0
mov eax,num1
mul num2
call writeint
call crlf
jmp ending
```

```
divison :
mov edx,0
```



```
mov eax,num1
mov ebx,0
mov ebx,num2
div num2
call writeint
call crlf
mov eax,edx
call writeint
call crlf
```

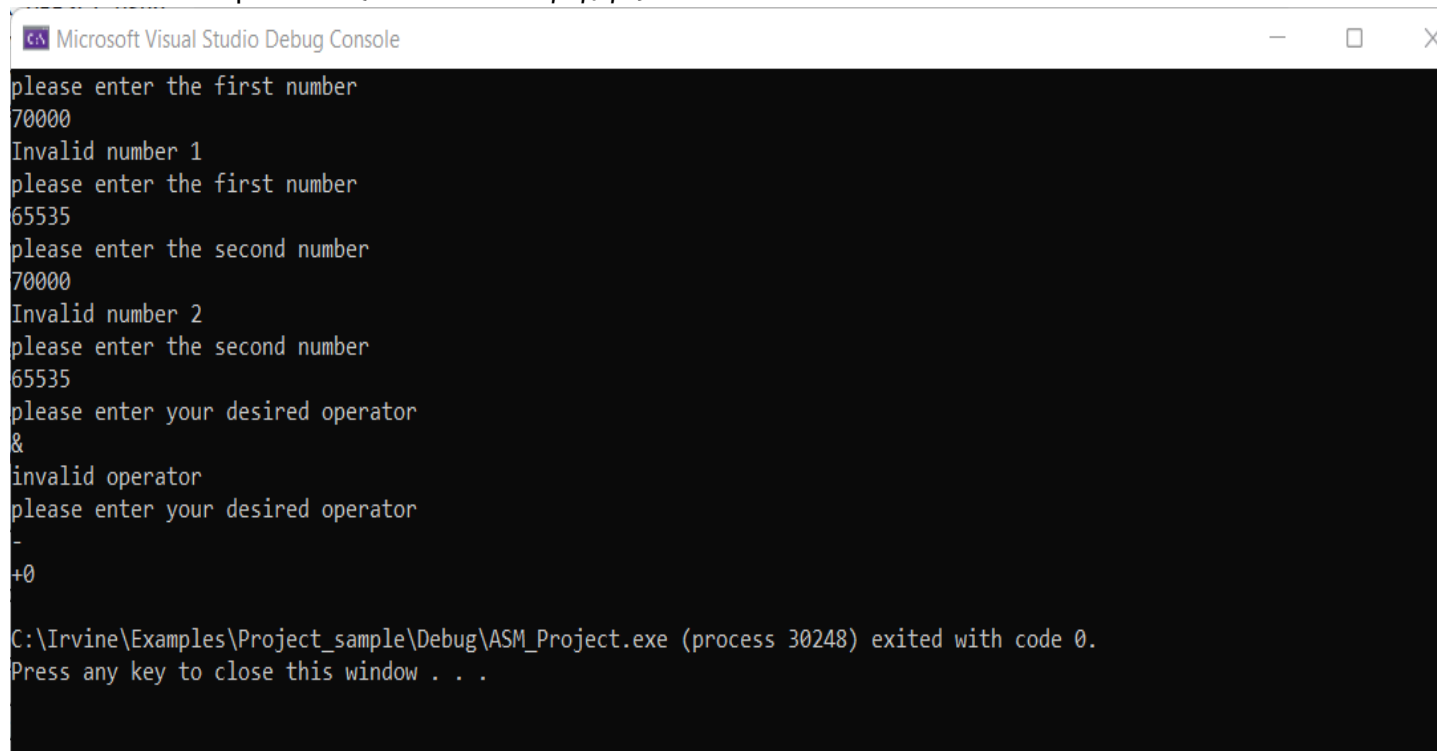
```
ending:
    exit
main ENDP

END main
```

OUTPUT :

Testing

- Invalid number entry (numbers greater than the 16 bit=65535 range of calculator)
- Invalid operator (other than -,+/,*)



```
Microsoft Visual Studio Debug Console

please enter the first number
70000
Invalid number 1
please enter the first number
65535
please enter the second number
70000
Invalid number 2
please enter the second number
65535
please enter your desired operator
&
invalid operator
please enter your desired operator
-
+0

C:\Irvine\Examples\Project_sample\Debug\ASM_Project.exe (process 30248) exited with code 0.
Press any key to close this window . . .
```

With Addition :

```
Microsoft Visual Studio Debug Console
please enter the first number
8
please enter the second number
2
please enter your desired operator
+
+10

C:\Irvine\Examples\Project_sample\Debug\ASM_Project.exe (process 5832) exited with code 0.
Press any key to close this window . . .
```

With subtraction :

```
Microsoft Visual Studio Debug Console
please enter the first number
8
please enter the second number
2
please enter your desired operator
-
-6

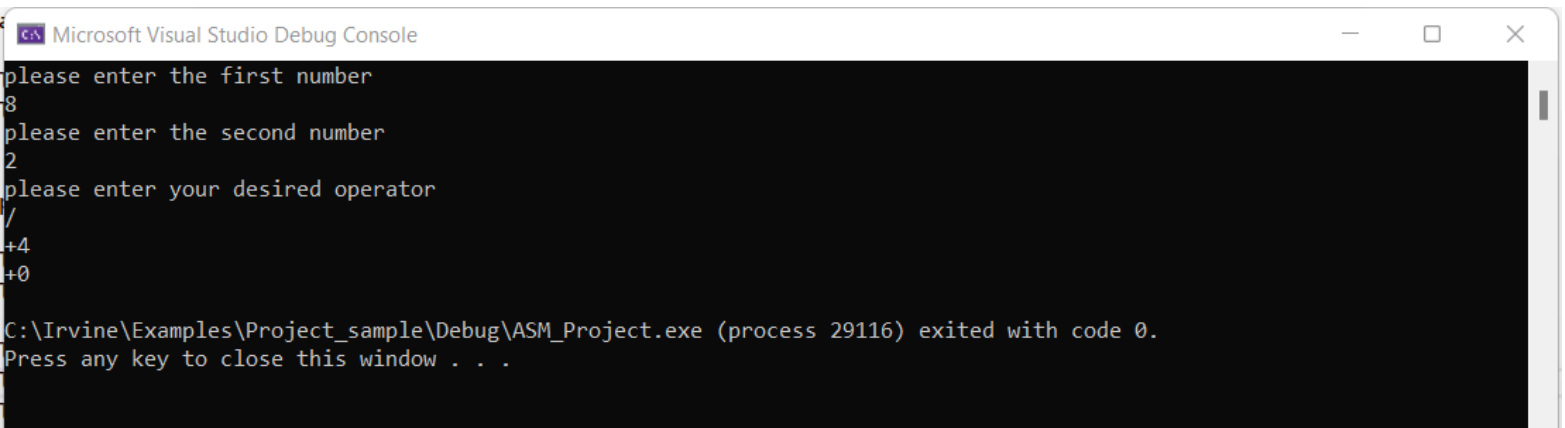
C:\Irvine\Examples\Project_sample\Debug\ASM_Project.exe (process 32460) exited with code 0.
Press any key to close this window . . .
```

With multiplication :

```
Microsoft Visual Studio Debug Console
please enter the first number
8
please enter the second number
2
please enter your desired operator
*
+16

C:\Irvine\Examples\Project_sample\Debug\ASM_Project.exe (process 18004) exited with code 0.
Press any key to close this window . . .
```

With Division : (no remainder example)

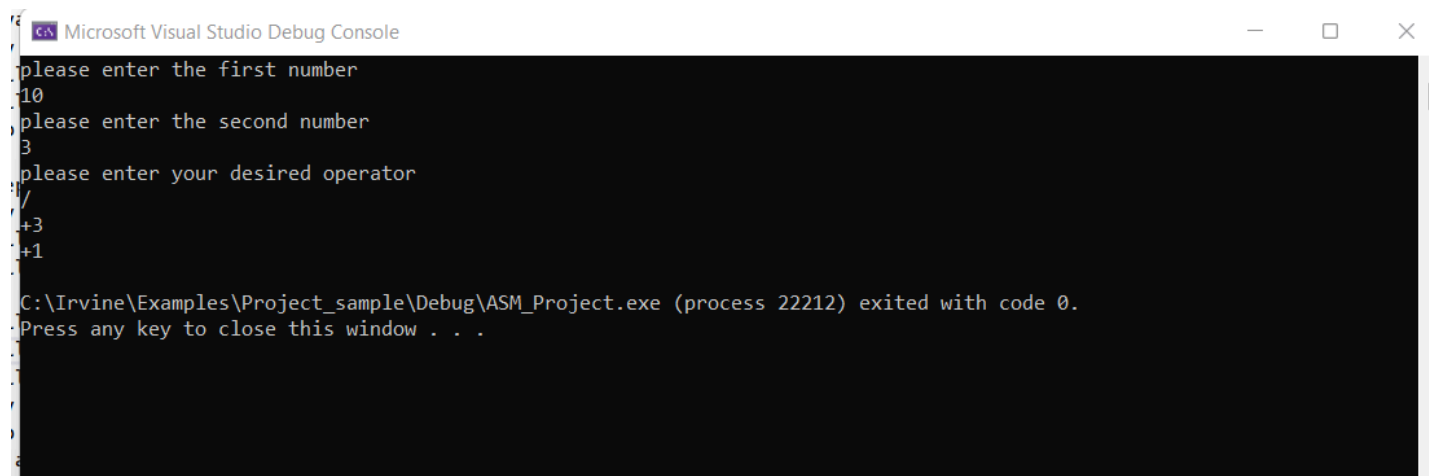


```
Microsoft Visual Studio Debug Console

please enter the first number
8
please enter the second number
2
please enter your desired operator
/
+4
+0

C:\Irvine\Examples\Project_sample\Debug\ASM_Project.exe (process 29116) exited with code 0.
Press any key to close this window . . .
```

With Division : (with remainder example)



```
Microsoft Visual Studio Debug Console

please enter the first number
10
please enter the second number
3
please enter your desired operator
/
+3
+1

C:\Irvine\Examples\Project_sample\Debug\ASM_Project.exe (process 22212) exited with code 0.
Press any key to close this window . . .
```