International Islamic University Islamabad
Faculty of Engineering and Technology
Department of Electrical and Computer Engineering

Computer Architecture and Organization Lab (CO 202 L)

Lab 12: Using Subroutines

Name:	**Shujat_Ali_Shah**
Reg. No:	**90-FET/BSCE/F23**
Date of Experiment:	**13/05/025**

OBE Rubrics Evaluation

a) PSYCHOMOTOR (To be judged in the field/lab during experiment)

Sr. No.	Criteria	Level 1 (0%)	Level 2 (25%)	Level 3 (50%)	Level 4 (75%)	Level 5 (100%)	Marks Obtained
1	Practical Implementation	0	1.25	2.5	3.75	5	
		Absent	With several critical errors and incomplete	With few errors, and incomplete	With some errors and complete	Without errors and complete	
2	Use of Equipment or Simulation/ Programming Tool	0	0.5	1	1.5	2	
		Absent	Limited competence	Some competence	Considerable competence	Competence	

(b) COGNITIVE (To be judged on the copy of experiment submitted)

Sr. No.	Criteria	Level 1 (0%)	Level 2 (25%)	Level 3 (50%)	Level 4 (75%)	Level 5 (100%)	Marks Obtained
3	Level of Participation & Attitude to Achieve Individual/Group Goals	0	0.25	0.5	0.75	1	
		Absent	Bad Attitude	Decent Attitude	Good Attitude	Proactive Attitude	

(c) AFFECTIVE (To be judged in the field/lab during experiment)

Sr. No.	Criteria	Level 1 (0%)	Level 2 (25%)	Level 3 (50%)	Level 4 (75%)	Level 5 (100%)	Marks Obtained
4	Level of Participation & Attitude to Achieve Individual/Group Goals	0	0.5	1	1.5	2	
		Absent	Rare sensible interaction	Some sensible interaction	Good sensible interaction	Encouraging sensible interaction	

5	**TOTAL OBTAINED MARKS (Out of 10)**	


```
      ORG  100   / Main program
      LDA  X     / Load X
      BSA  SH4   / Branch to subroutine
      STA  X     / Store shifted number
      LDA  Y     / Load Y
      BSA  SH4   / Branch to subroutine again
      STA  Y     / Store shifted number
      HLT        / halt
X,    HEX  1234  / shift left this number
Y,    HEX  4321  / shift left this number too


/ This is the Subroutine to shift left a number 4 times
SH4,  HEX  0      / Store return address here
      CIL         / Circulate left 1st time
      CIL         / Circulate left 2nd time
      CIL         / Circulate left 3rd time
      CIL         / Circulate left 4th time
      AND  MSK    / Set AC(0-4) to zero
      BUN  SH4  I / Return to main program

MSK,  HEX  FFF0   / Mask operand
      END         / End of Program
```

Program 12.1: Using Subroutine

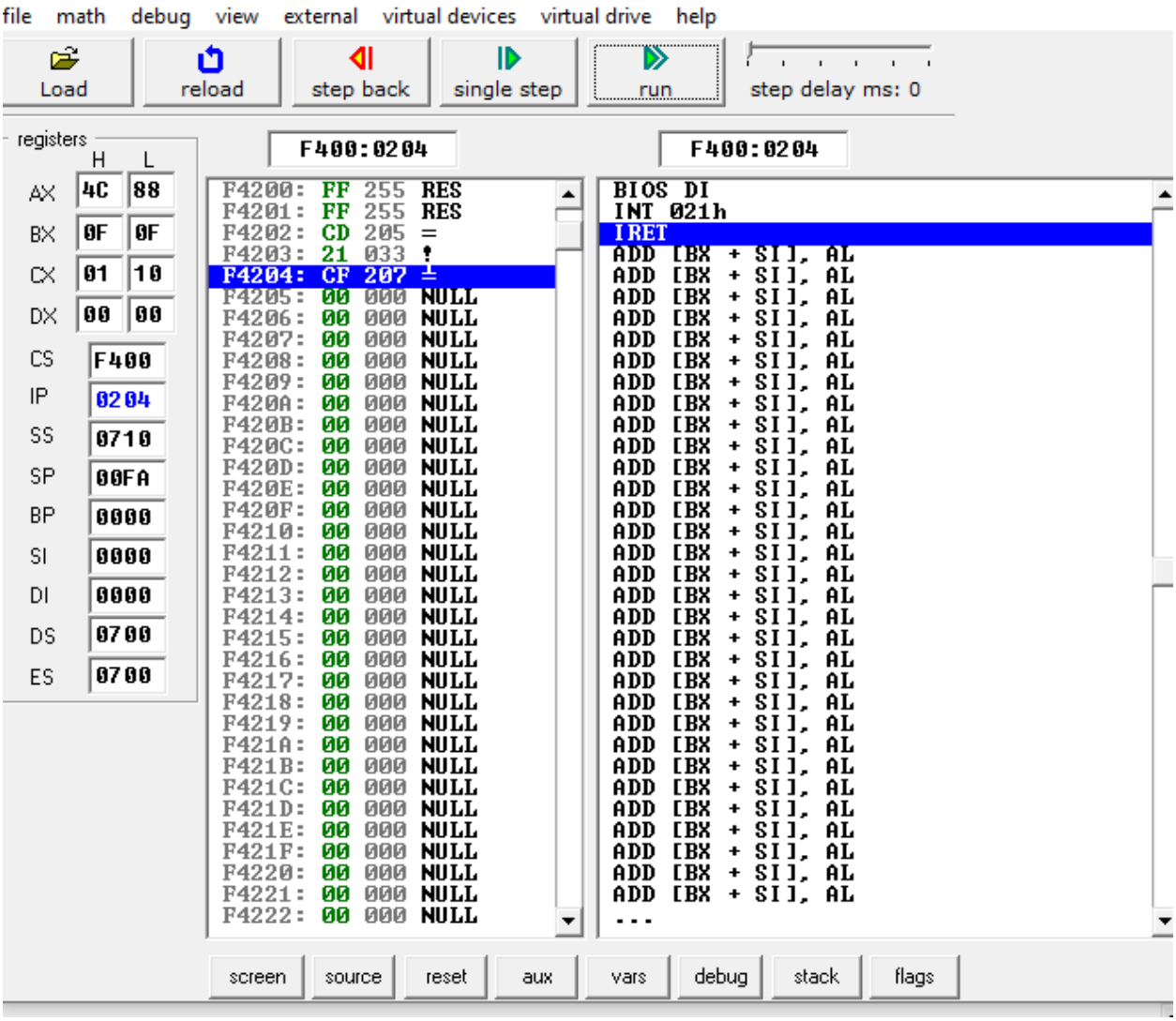
```
      ORG  200   / Main program
      LDA  X     / Load X
      BSA  OR    / Branch to subroutine OR
      HEX 3AF6   / Second Operand Stored Here
      STA  Y     / Subroutine returns here
      HLT        / halt computer
X,    HEX  7B95  / First operand stored here
Y,    HEX  0     / Result is stored here
OR,   HEX  0     / Subroutine OR
      CMA        / Complement first operand
      STA TMP    / Store in temporary location
      LDA OR I   / Load second operand
      CMA        / Complement second operand
      AND TMP    / AND complemented first operand
      CMA        / Complement again to get OR
      ISZ OR     / Increment return address
      BUN OR I   / Return to main program
TMP,  HEX 0      / Temporary Storage
      END
```

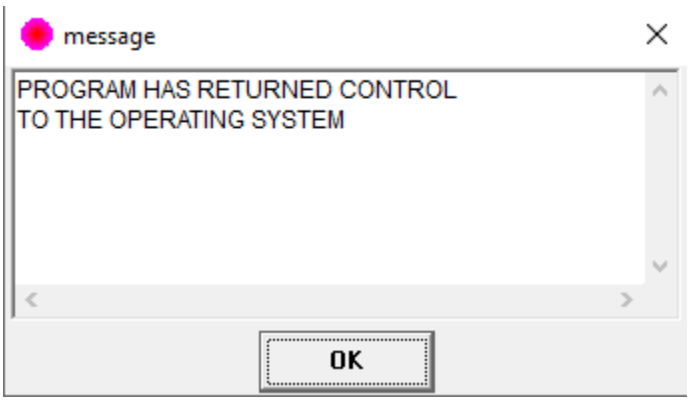
Program 12.2: Passing Parameters to a Subroutine

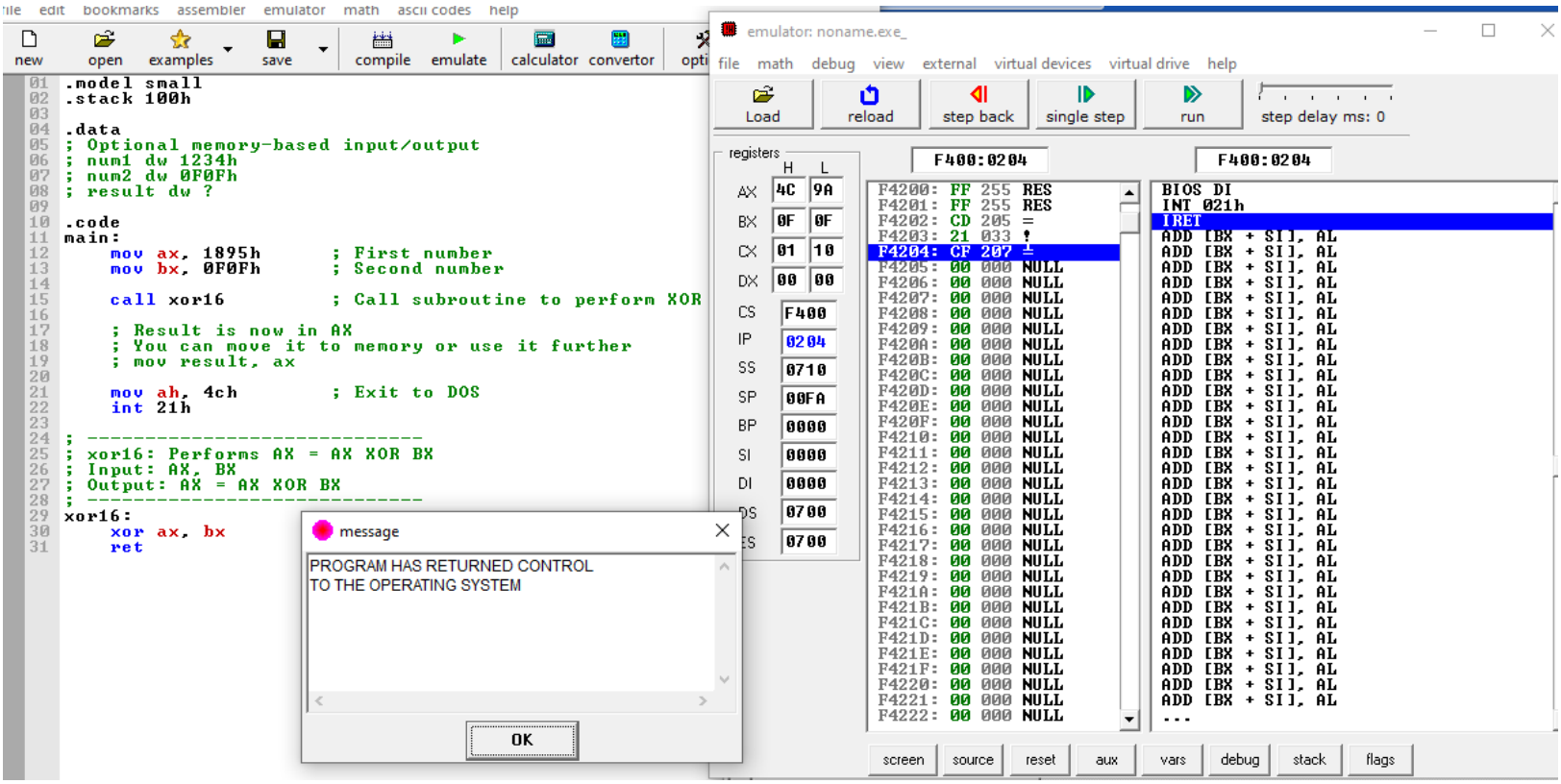
Lab 12 Task: Write a subroutine to perform bitwise XOR of two 16-bit numbers

Solution:

```
.model small
.stack 100h

.data
; Optional memory-based input/output
; num1 dw 1234h
; num2 dw 0F0Fh
; result dw ?

.code
main:
    mov ax, 1287h       ; First number
    mov bx, 0F0Fh       ; Second number

    call xor16          ; Call subroutine to perform XOR

    ; Result is now in AX
    ; You can move it to memory or use it further
    ; mov result, ax

    mov ah, 4ch         ; Exit to DOS
    int 21h

; ----------------------------
; xor16: Performs AX = AX XOR BX
; Input: AX, BX
; Output: AX = AX XOR BX
; ----------------------------
xor16:
    xor ax, bx
    ret
```


file math debug view external virtual devices virtual drive help
Load
reload
step back
single step
run
step delay ms: 0
registers
H L
AX 4C 88
BX 0F 0F
CX 01 10
DX 00 00
CS F400
IP 0204
SS 0710
SP 00FA
BP 0000
SI 0000
DI 0000
DS 0700
ES 0700
F400:0204
F400:0204
F4200: FF 255 RES
F4201: FF 255 RES
F4202: CD 205 =
F4203: 21 033 !
F4204: CF 207 =
F4205: 00 000 NULL
F4206: 00 000 NULL
F4207: 00 000 NULL
F4208: 00 000 NULL
F4209: 00 000 NULL
F420A: 00 000 NULL
F420B: 00 000 NULL
F420C: 00 000 NULL
F420D: 00 000 NULL
F420E: 00 000 NULL
F420F: 00 000 NULL
F4210: 00 000 NULL
F4211: 00 000 NULL
F4212: 00 000 NULL
F4213: 00 000 NULL
F4214: 00 000 NULL
F4215: 00 000 NULL
F4216: 00 000 NULL
F4217: 00 000 NULL
F4218: 00 000 NULL
F4219: 00 000 NULL
F421A: 00 000 NULL
F421B: 00 000 NULL
F421C: 00 000 NULL
F421D: 00 000 NULL
F421E: 00 000 NULL
F421F: 00 000 NULL
F4220: 00 000 NULL
F4221: 00 000 NULL
F4222: 00 000 NULL
BIOS DI
INT 021h
IRET
ADD [BX + SI], AL
...
screen
source
reset
aux
vars
debug
stack
flags


message
PROGRAM HAS RETURNED CONTROL
TO THE OPERATING SYSTEM
OK


file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor
01 .model small
02 .stack 100h
03
04 .data
05 ; Optional memory-based input/output
06 ; num1 dw 1234h
07 ; num2 dw 0F0Fh
08 ; result dw ?
09
10 .code
11 main:
12 mov ax, 1895h ; First number
13 mov bx, 0F0Fh ; Second number
14
15 call xor16 ; Call subroutine to perform XOR
16
17 ; Result is now in AX
18 ; You can move it to memory or use it further
19 ; mov result, ax
20
21 mov ah, 4ch ; Exit to DOS
22 int 21h
23
24 ; ------------------------------
25 ; xor16: Performs AX = AX XOR BX
26 ; Input: AX, BX
27 ; Output: AX = AX XOR BX
28 ; ------------------------------
29 xor16:
30 xor ax, bx
31 ret
message
PROGRAM HAS RETURNED CONTROL
TO THE OPERATING SYSTEM
OK
emulator: noname.exe_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
registers
H L
AX 4C 9A
BX 0F 0F
CX 01 10
DX 00 00
CS F400
IP 0204
SS 0710
SP 00FA
BP 0000
SI 0000
DI 0000
DS 0700
ES 0700
F400:0204
F4200: FF 255 RES
F4201: FF 255 RES
F4202: CD 205 =
F4203: 21 033 !
F4204: CF 207 ╧
F4205: 00 000 NULL
F4206: 00 000 NULL
F4207: 00 000 NULL
F4208: 00 000 NULL
F4209: 00 000 NULL
F420A: 00 000 NULL
F420B: 00 000 NULL
F420C: 00 000 NULL
F420D: 00 000 NULL
F420E: 00 000 NULL
F420F: 00 000 NULL
F4210: 00 000 NULL
F4211: 00 000 NULL
F4212: 00 000 NULL
F4213: 00 000 NULL
F4214: 00 000 NULL
F4215: 00 000 NULL
F4216: 00 000 NULL
F4217: 00 000 NULL
F4218: 00 000 NULL
F4219: 00 000 NULL
F421A: 00 000 NULL
F421B: 00 000 NULL
F421C: 00 000 NULL
F421D: 00 000 NULL
F421E: 00 000 NULL
F421F: 00 000 NULL
F4220: 00 000 NULL
F4221: 00 000 NULL
F4222: 00 000 NULL
F400:0204
BIOS DI
INT 021h
IRET
ADD [BX + SI], AL
...
screen source reset aux vars debug stack flags