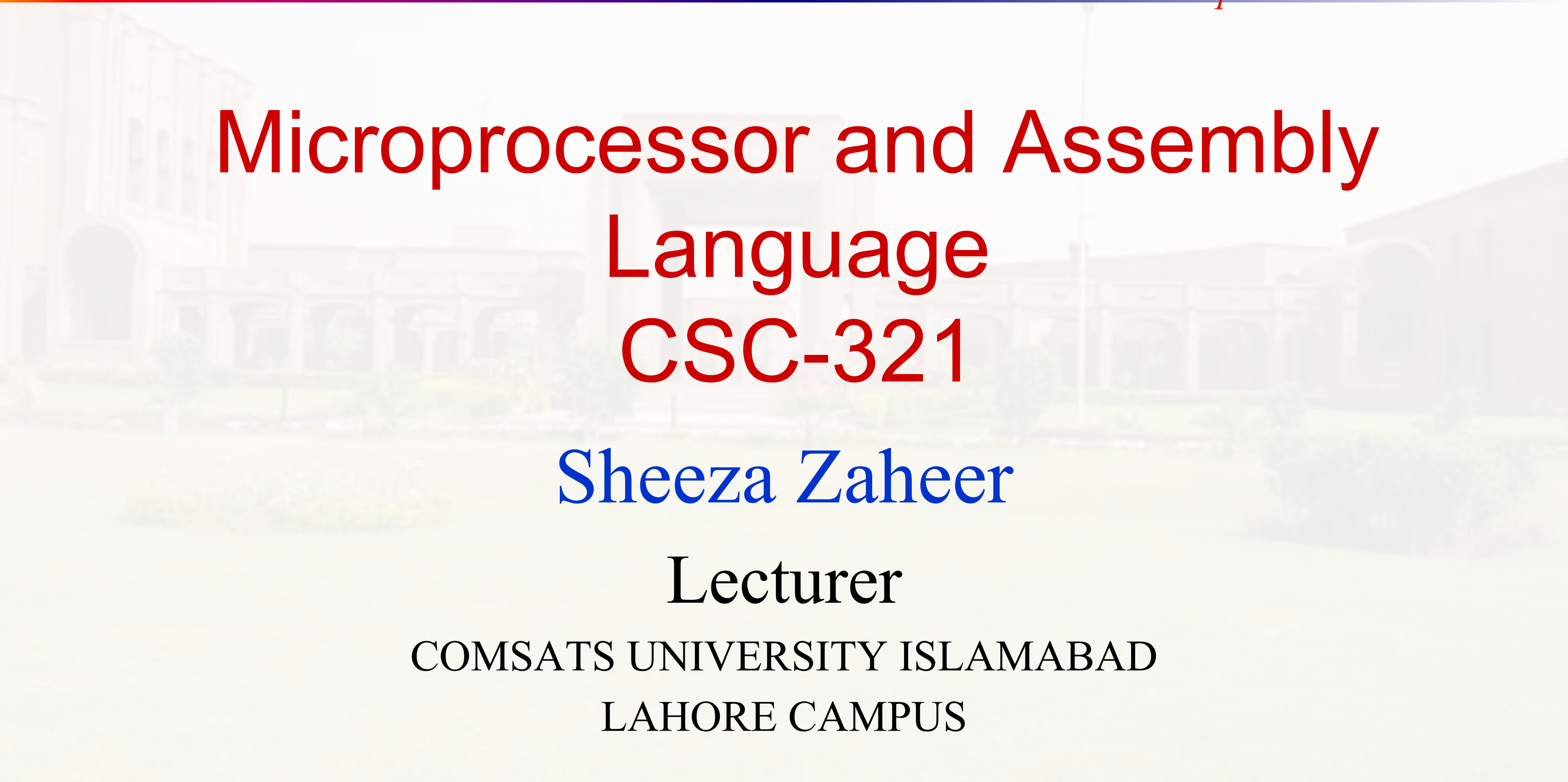


Microprocessor and Assembly Language CSC-321

Sheeza Zaheer

Lecturer

COMSATS UNIVERSITY ISLAMABAD

LAHORE CAMPUS

The Procedures

OUTLINE

- **Procedure**
  - Introduction
  - Syntax
  - CALL and RET instructions
  - Example

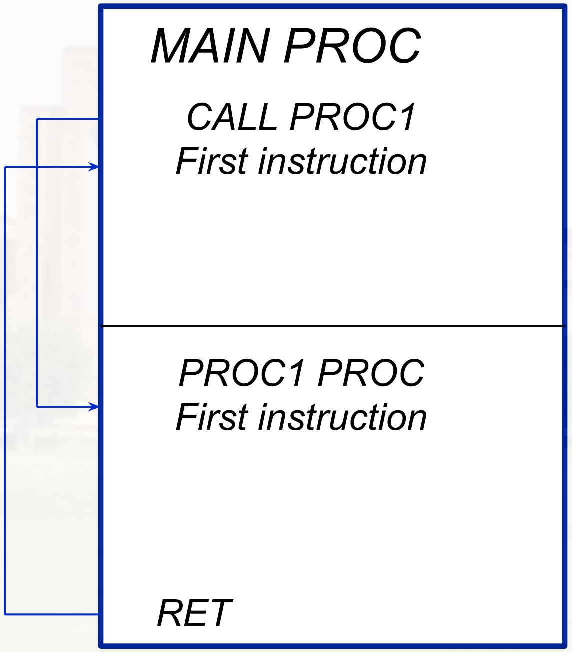
- **References**
  - **Chapter 8, Section 8.3, 8.4 & 8.5,** Ytha Yu and Charles Marut, “Assembly Language Programming and Organization of IBM PC

Creating Procedures

- Large problems can be divided into smaller tasks to make them more manageable
- A procedure is the assembly equivalent of a Java or C function.
- Following is an assembly language procedure named **sample**:

```
sample PROC
   .
   .
   ret
sample ENDP
```

Procedure call and return



MAIN PROC

CALL PROC1
First instruction

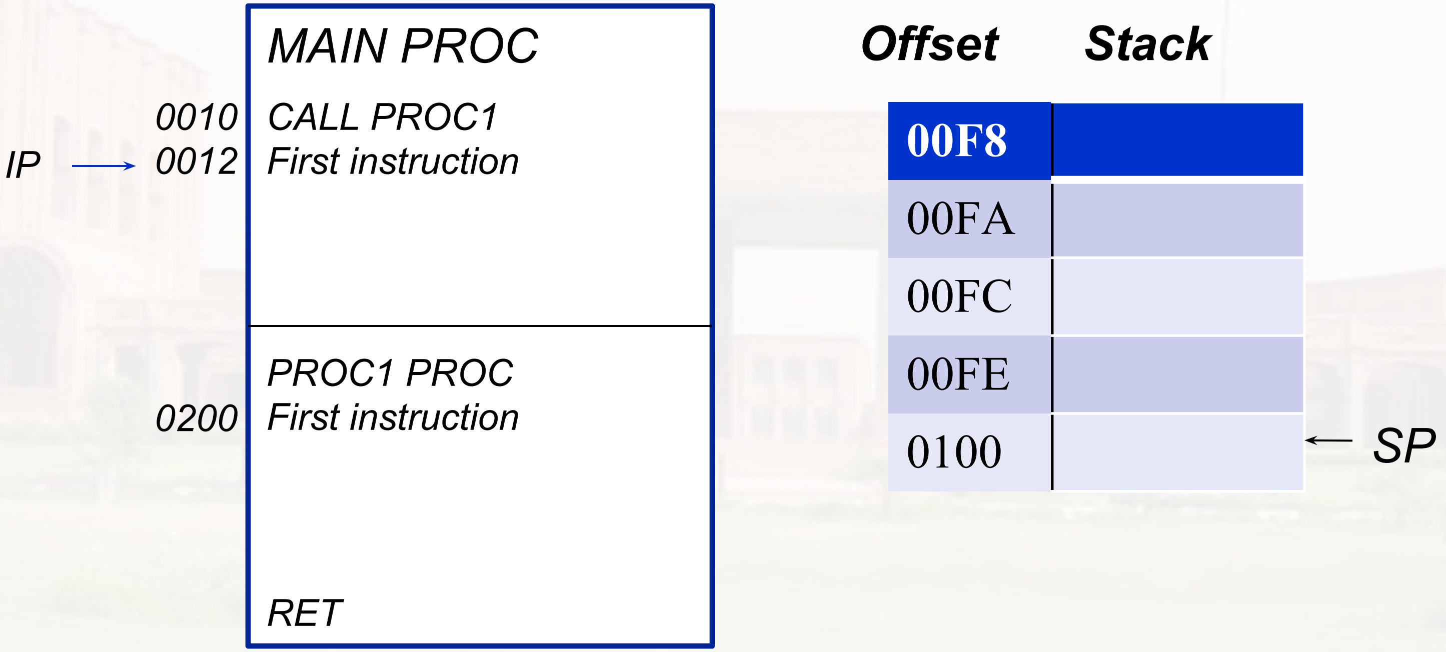
PROC1 PROC
First instruction

RET

CALL and RET Instructions

- The CALL instruction calls a procedure
  - pushes offset of next instruction on the stack
  - copies the address of the called procedure into IP (Note: IP=Instruction Pointer)
- The RET instruction returns from a procedure
  - pops top of stack into IP

Before Call



MAIN PROC

0010 CALL PROC1

IP → 0012 First instruction

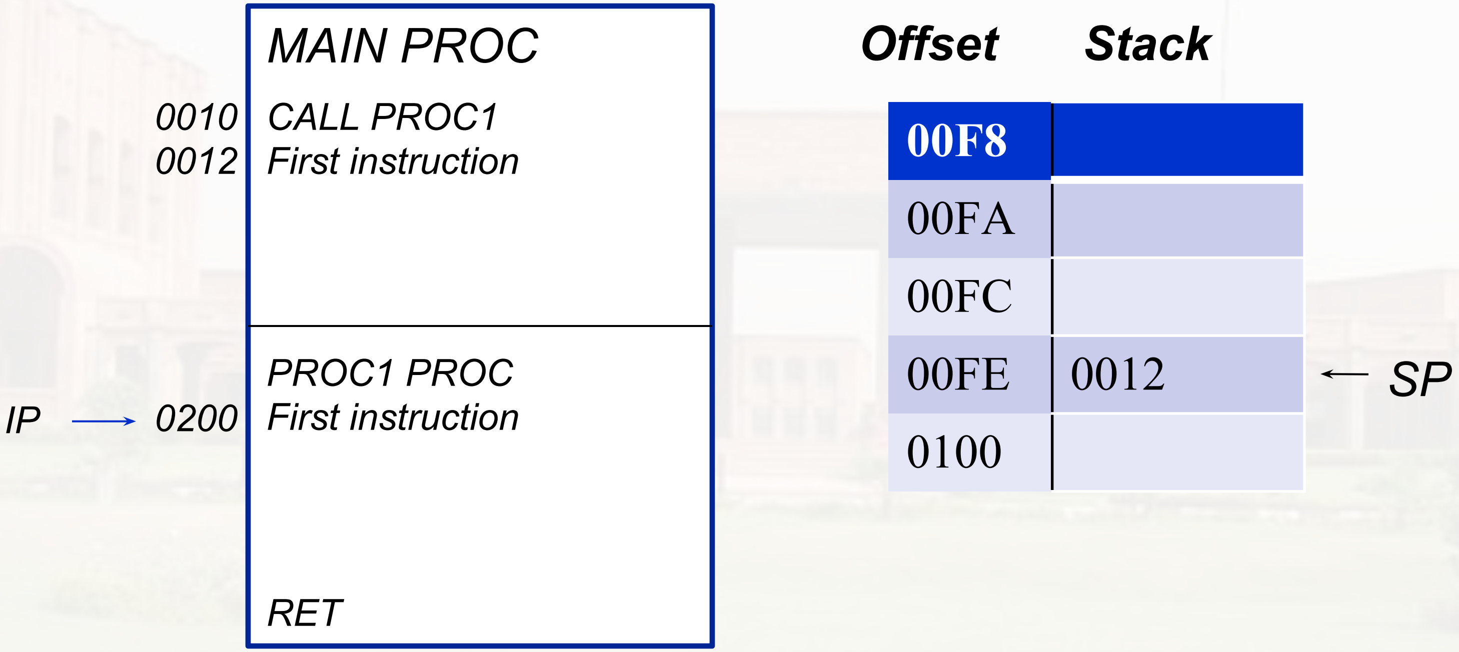
PROC1 PROC

0200 First instruction

RET

Offset	Stack	
00F8		
00FA		
00FC		
00FE		
0100		← SP

After Call



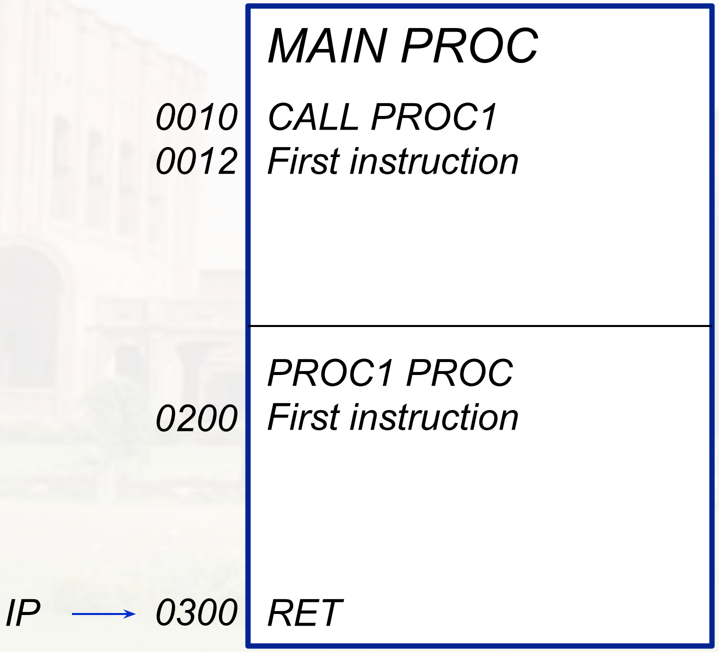
```
          MAIN PROC
     0010 CALL PROC1
     0012 First instruction

          PROC1 PROC
IP → 0200 First instruction

          RET
```

Offset	Stack	
00F8		
00FA		
00FC		
00FE	0012	← SP
0100		

Before RET



MAIN PROC

0010 CALL PROC1

0012 First instruction

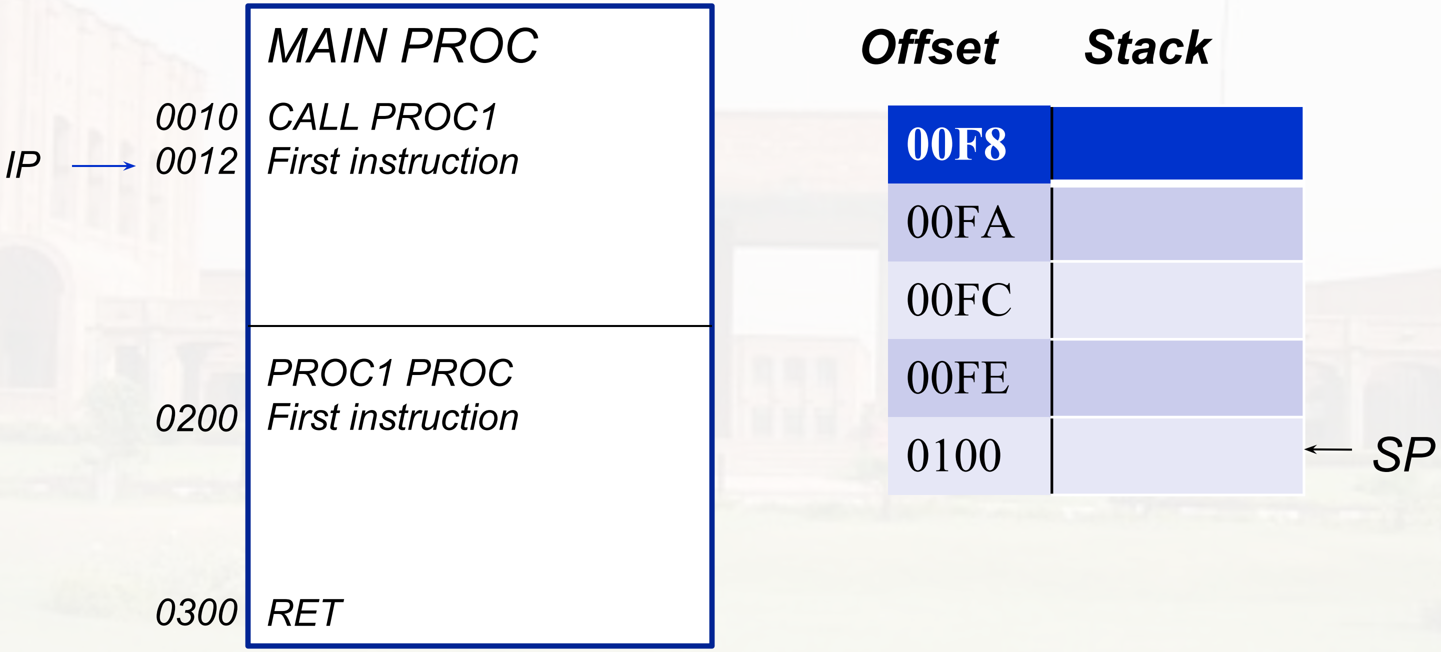
PROC1 PROC

0200 First instruction

IP → 0300 RET

Offset	Stack	
00F8		
00FA		
00FC		
00FE	0012	← SP
0100		

After RET



MAIN PROC

0010 CALL PROC1

IP → 0012 First instruction

PROC1 PROC

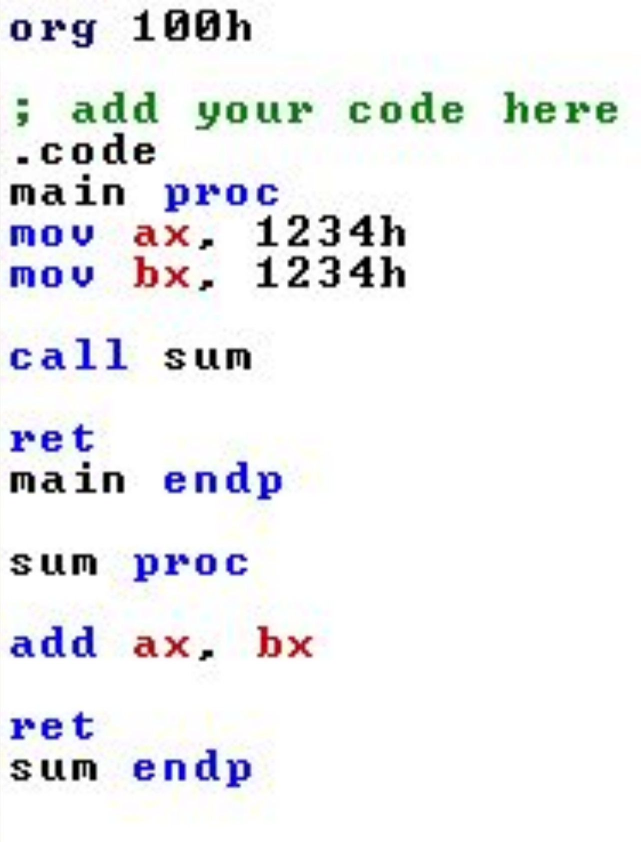
0200 First instruction

0300 RET

Offset	Stack	
00F8		
00FA		
00FC		
00FE		
0100		← SP

Example

- Write a procedure that adds numbers in AX and BX and save their answer in AX.



```
org 100h

; add your code here
.code
main proc
mov ax, 1234h
mov bx, 1234h

call sum

ret
main endp

sum proc

add ax, bx

ret
sum endp
```

For Practice

- Exercise Ch#8: Q9 and Q10.