

Tutorial 3

Addresses and Manual Tracing

CO 2103 Assembly Language

Addressing Modes

- **Task 1:** Determine the addressing mode(s) for each of the following instructions:

```
main proc
    mov ax,8
    push ax
    call Factorial
    mov ax,4C00h
    int 21h
main endp
```

```
Factorial proc
    push bp
    mov bp,sp
    mov ax,[bp+4]
    cmp ax,1
    ja L1
    mov ax,1
    jmp L2
L1:    dec ax
        push ax
        call Factorial
        mov bx,[bp+4]
        mul bx
L2:    pop bp
        ret 2
Factorial endp
```

Physical vs Logical Addresses

- **Task 2:** Determine the physical address (in hex) for each of the following logical addresses (in hex):

1785 : 1234

A000 : 2341

FE01 : 5678

B000 : 1A3C

1FA0 : 9ABC

294B : 000C

0034 : DEF0

0123 : 4000

5000 : 6789

FFFF : FFFF

- **Task 3:** Referring to above answers, or otherwise, answer the following questions:
 - What is the size of the address bus for **Intel 8086**?
 - What is the highest memory address accessible by **Intel 8086**?
 - Are all the above physical addresses accessible by **Intel 8086**?
 - Can you identify **overlapping** and **wrapping**?

Effective Addresses

- **Task 4:** Determine the effective address (logical) for each of the following instructions, except `int 20h` and `ret`:
 - assuming `DS=17D3`, `CS=17E3`

```
        .code
0000 BE 0000 R      main:  mov si, offset num
0003 8B 04          mov ax,[si]
0005 8B 5C 02      mov bx,[si+2]
0008 E8 000D R      call proc1
000B CD 20         int 20h
000D BF 0004 R      proc1: mov di, offset sum
0010 C7 05 0000    mov word ptr [di],0
0014 01 05         add [di],ax
0016 01 1D         add [di],bx
0018 81 3D        FF00 cmp [di],0ff00h
001C 74 03        je here1
001E C2 0001      ret 1
0021 C3          here1: ret 0

        .data
0000 1122 3344    num dw 1122h,3344h
0004 0000        sum dw ?
```

Predicting Memory/Register Contents

- **Task 5:** Predict the content of the affected memory/register(s)/flag(s) after execution of each of the following instructions in sequence, except the `int 20h` (all addresses in hex):

– assuming *num* is at `DS:000A`, *proc1* is at `CS:000D`, *here1* at `CS:0028`

```
main:  mov si, offset num          proc1: mov di, offset sum
      mov ax,[si]              mov word ptr [di],0
      mov bx,[si+2]           add [di],ax
      call proc1              add [di],bx
      ;display sum           cmp [di],0ff00h
      int 20h                 je here1
                                mov cx,[di]
                                and ch,0ffh
                                not cl
                                ret 1
                                here1: ret 0

                                .data
                                num dw 1122h,3344h
                                sum dw ?
```