CS253 Architectures II

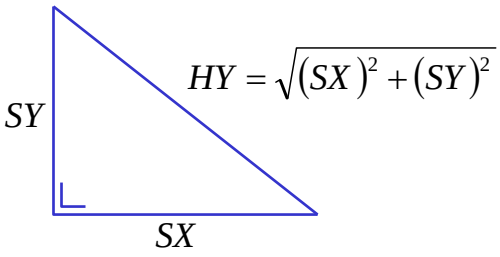
Lecture 6

Assembly Language

(Floating Point, 8087)

Charles Markham

A Pythagorean Problem



$SX = 5$

$SY = 12$

$HY = \sqrt{(5)^2 + (12)^2} = 13$

Use the FP processor (8087) to carry out the calculation shown above.

Use short real numbers in the calculation.

Start of FP Calculation

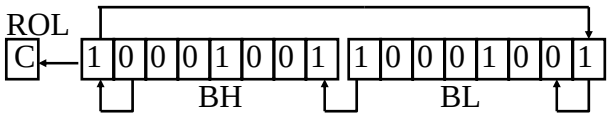
```
        .8087                   ; Tell MASM co-processor is present
        .STACK 100h
        .DATA

        SX      dd      5.0     ; short real 4 bytes
        SY      dd      12.0    ; short real
        HY      dd      0.0     ; Result will be stored here
        cntrl   dw      03FFh   ; Control word for 8087
        stat    dw      0       ; Status after calculation

        .CODE
```

Note that DD is short for define *double word*, the compiler converts floating numbers to their short real format (also 4 bytes).

```
        .STARTUP
        FINIT                   ; Set FPU to default state
        FLDCW   cntrl           ; Round even, Mask Interrupts
        FLD     SX              ; Push SX onto FP stack
        FMUL    ST,ST(0)        ; Multiply ST*ST result on ST
        FLD     SY              ; Push SY onto FP stack
        FMUL    ST,ST(0)        ; Multiply ST*ST
        FADD    ST,ST(1)        ; ADD top two numbers on stack
        FSQRT                   ; Square root number on stack
        FSTSW   stat            ; Load FPU status into [stat]
        mov     ax,stat         ; Copy [stat] into ax
        and al,0BFh             ; Check all 6 status bits
        jnz pass                ; If any bit set then jump
        FSTP HY                 ; Copy result from stack into HY
pass:   nop
```

Now print the value in HY

Printing out the binary value

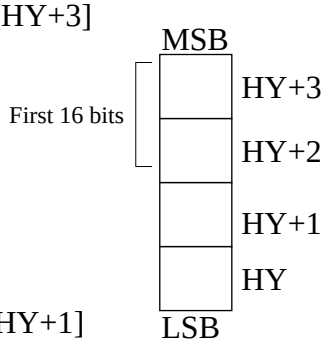



```
        mov   cx,16
back:   rol   bx,1    ; Rotate bx
        jc    set     ; Check MSB first
        mov   dl,'0'  ; If carry set dl='0'
        jmp   over
set:    mov   dl,'1'  ; If carry not set dl='1'
over:   mov   ah,02h  ; Print ASCII of dl
        int   021h
skip:   loop  back    ; Repeat 16 times
```

Rotate all 16 bits in BX into the carry

Getting at the value in HY

```
mov bx,OFFSET HY      ;bx=[HY+2]+256*[HY+3]
inc bx
inc bx
mov ax,[bx]
mov bx,ax

print binary of bx bits 31 to 16

mov bx,OFFSET HY      ;bx=[HY+0]+256*[HY+1]
mov ax,[bx]
mov bx,ax

print binary of bx bits 15 to 0
```



DD : Define double word

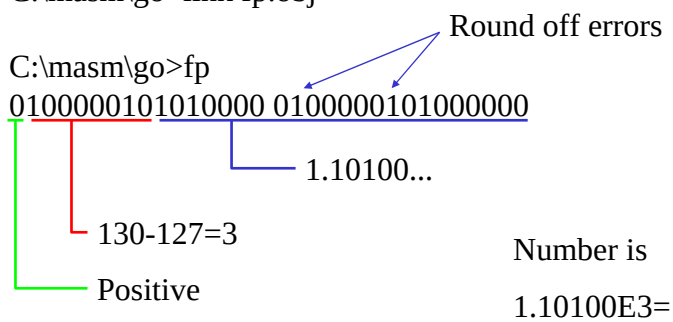
If data in format x.yz then stored in short real format

Testing the FP Program

C:\masm\go>masm /l fp.asm

C:\masm\go>link fp.obj

C:\masm\go>fp



Number is

$1.10100E3=1.101x2^3$

1101.000 = 13 decimal

Note earlier change 10^3 to 2^3

A final look !

```
0000                                    .DATA
0000    40A00000                        SX      dd      5.0     ; short real
0004    41400000                        SY      dd      12.0    ; short real
0008    00000000                        HY      dd      0.0     ; short real
000C    03FF                            cntrl   dw      03FFh
000E    0000                            stat    dw      0

0000                                    .CODE
                                        .STARTUP
0017    9B DB E3                        FINIT                   ; Set FPU to default state
001A    9B D9 2E 000C R                 FLDCW   cntrl           ;
001F    9B D9 06 0000 R                 FLD     SX              ; Push SX onto FP stack
0024    9B D8 C8                        FMUL    ST,ST(0)        ; Multiply ST*ST result on ST
0027    9B D9 06 0004 R                 FLD     SY              ; Push SY onto FP stack
002C    9B D8 C8                        FMUL    ST,ST(0)        ; Multiply ST*ST
002F    9B D8 C1                        FADD    ST,ST(1)        ; ADD top two numbers on stack
0032    9B D9 FA                        FSQRT                   ; Square root number on stack
0035    9B DD 3E 000E R                 FSTSW   stat            ; Load FPU status into ax
003A    A1 000E R                       mov     ax,stat
003D    24 BF                           and     al,0BFh         ; Check all 6 status bits
003F    75 05                           jnz     pass
0041    9B D9 1E 0008 R                 FSTP    HY              ; Copy result from stack into HY
0046    90                      pass:   nop
```