

EE424 Computer Architecture

Lab 6: Sine computation using Taylor Series

Spring 2003

You are to compute Sine X using the Taylor series which has the following form:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

For these computations use floating point (FP) numbers and operations.

The maximum error (ϵ) that is allowed for this computation is 1×10^{-5} . No more terms need to be computed when the maximum error is reached. If you use a loop; this will be the condition to stop the loop.

Your program will read from the console a value (X) and will display the result (Sin X).

A program that converts C to F (using floating point) is included here. This program may help you to understand how FP instructions and data can be used.

REPORT

Please include the following items in your report.

1. Explain how your program works. You may use a flow chart, pseudo C program, or other way to explain the program.
2. Show examples that test your code
3. Conclusion. Explain what you learned here and what was difficult about this lab
4. A print out of your program.
5. Please include your program code along with SPIM translation to assembly (addresses, instruction code –hexadecimal, instruction,...) Please show how some instructions (pseudo instructions) are translated. Remember that we are interested on the real instructions. These are the instructions that are executed by the processor. Please identify the format (R, I, J) of each instruction. An example is shown below

[0x00400018]	0x3c011001	I	lui \$1, 4097 [data1]	← ; 28: l.s \$f0,data1 #load data
[0x0040001c]	0xc4200044	I	lwc1 \$f0, 68(\$1) [data1]	← ; 29: mul.s \$f0,\$f0,\$f2 #
[0x00400028]	0x46020002	R	mul.s \$f0, \$f0, \$f2	; 30: add.s \$f12,\$f0,\$f4 # add 32
[0x0040002c]	0x46040300	R	add.s \$f12, \$f0, \$f4	

pseudo instruction

Using simulator setting (toolbar: simulator then setting) display the floating point in hexadecimal. After executing your program please pick two floating point registers that have been modified (i.e. they are not set to zero) and get their binary value. With the binary value figure out their decimal equivalent (MIPS uses the IEEE 754 FP Standard).

REPORT IS DUE: April 1 (in class or 3-5pm session). {late reports will get 15 points off}

SHOW AND TELL. Please give a demonstration of your program to your TA.

```
##### TEMP_CONV.S  FILE #####
## this program asks user for temperature in Celsius, ##
## converts to Fahrenheit, and prints the result.      ##
##                                                    ##
##          f0 - reads in Celsius                    ##
##          f12 - holds Fahrenheit result             ##
##          a0 - points to output strings             ##
#####
```

```
        .text
        .globl __start
__start:
        la $a0,prompt      # print prompt on terminal
        li $v0,4
        syscall

        li $v0,6           # syscall 6 reads a FP number in $f0
        syscall

        l.s $f2, conv      # load single (f2<--1.8)
        l.s $f4, diff      # load single (f4<-- 32)

        mul.s $f0,$f0,$f2   # to convert,multiply by 1.8,
        add.s $f12,$f0,$f4  # add 32

        la $a0,ans1        # print string before result
        li $v0,4
        syscall

        li $v0,2           # print result (fp reg 12)
        syscall

        la $a0,end1        # system call to print
        li $v0,4           # out a newline
        syscall

        li $v0,10
        syscall             # BYE
```

```
#####
#                                                    #
#          data segment                            #
#                                                    #
#####
```

```
        .data
prompt:  .asciiz "Enter temperature (Celsius): "
ans1:    .asciiz "The temperature in Fahrenheit is "
endl:    .asciiz "\n"
conv:    .float 1.8
diff:    .float 32.0
```

```
## END OF  TEMP_CONV.S  FILE
```