CMPE 12L Lab 4 - Winter 2015

Prof. HP Dommel
Due: Feburary 23, 2015 5pm
75 Points (20 Report, 55 Work)

Prerequisites

- Read through this **entire** lab assignment.

Tutor/TA Review

- Overview of LDR and STR instructions
- Overview of floating point precision
- What's required

Overview

You have two objectives in this lab. First you must convert two two–digit base 10 numbers into half–precision floating–point (16 bits) numbers (note: numbers less than ten can be entered with a preceeding 0). Second you must multiply these floating point numbers. The two operands and the product must be stored in memory. Don't forget to handle negative numbers and make a case for inputs of zero.

Additional Requirements

- Starting location of your code is 0x3000
- Store the product as a half–precision floating–point at address 0x3200
- Sign calculation, mantissa calculation, exponent calculation, multiplication, and division (for bit–shifting) are to be implemented as subroutines. Use proper subroutine convention (JSR and RET). You may use the multiplication and division subroutines you wrote in Lab 3.
- It is mandatory that you work with a partner
- Ensure that your program can handle not just positive operands, but negative and zero inputs as well.

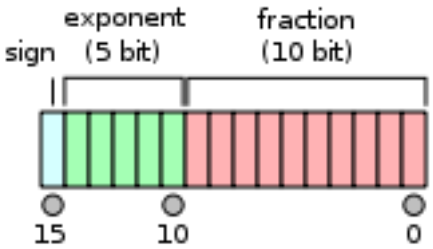

Figure 1: Half–Precision Floating–Point

Pre-Lab Requirements

Before lab begins, you must have flowcharts that detail:

- Taking a two–digit sequence of characters (0–9) and converting them to an integer
- Converting the resulting integer (between 0 and 99) to a halfprecision floatingpoint number

These will be reviewed by your Tutor/TA. Once these flowcharts have been checked off, you may begin the programming.

Tutor/TA Review

At the beginning of the lab, your lab tutor/TA will answer questions concerning what is required of the lab. After they answer questions, they will review and comment on individual flowcharts. Please write, debug and demonstrate that these routines are correctly functioning before starting the multiplication portion of the lab. These must be checked off by your TA. On the second day of lab, the TA will present an overview of the multiplication algorithm (in the attached pdf) and flowchart.

Half-Precision Floating-Point Format

Because the LC–3 has a 16–bit word size, we will be using Half–Precision Floating–Point Format for this lab, as opposed to Full–Precision (which has 32 bits).

The format of this number format is as follows:

- 1 bit: Sign bit (bit 15)
- 5 bits: Exponent (bits 14-10)
- 10 bits: Mantissa (bits 9-0)

Lab Submission

Your lab will be submitted via your eCommons account. Please log in to eCommons using your UCSC account and attach the following files to your "Lab4" assignment submission:

- lab4_[username].asm
- lab4_report_[username].pdf

Note that the final report must be submitted in PDF format. Make sure to confirm that your assignment is SAVED and SUBMITTED before the deadline. You may resubmit your assignment an unlimited number of times up until the due date.

Check-off

For this lab, as with most labs, you will need to demonstrate your lab when it is finished to a TA/tutor and get it signed off. You must get your flowcharts checked–off and then your program once it is completed.

Grading template

Requirements

☐ (10 pts) Flowcharts

- Flowchart for two–digit conversion of ASCII to integer (5 pts)
- Flowchart for conversion of integer to floating–point (5 pts)

☐ (35 pts) Conversion Subroutine

- Able to convert two–digit sequence of ASCII characters to integer (5 pts)
- Able to convert integer to half–precision floating–point (5 pts)
- Recognizing negative numbers (3 pts)
- Correct mantissa calculation (10 pts)
- Correct exponent value (7 pts)
- Subroutines for sign calculation, mantissa calculation, exponent calculation, multiplication, and division (5 pts)

☐ (10 pts) Commenting and Style

Lab Write-up Requirements

In the lab write–up, we will be looking for the following things. The lap report is out of 20 points. Along with the usual items and formatting, you should answer the following questions:

- The algorithm you use in your program in English and/or pseudocode with flowcharts

- Enumerate every subroutine you implemented along with its function, the registers that hold its arguments, and the registers that hold its return values
- What works? What doesn't?
- Sample output

Lab Report Questions

- What's the largest number that can be represented in half–precision floating–point format?
- What's the smallest positive number that can be represented in half–precision floating–point format?
- How do JSR and RET work in LC3?