Study of an FPU

Yu-wen Pwu
Department of Computer Science, National Chiao Tung University
ywpu@cs.nctu.edu.tw

Abstract

This study examines how the floating point unit (FPU) is designed in modern computers. My study methods are including, but not limited to, source code analysis, behavioral simulation of the circuit and implement it onto a Xilinx Spartan-3E Starter Kit development board featuring Spartan-3 XC3S500E-4FG320C FPGA.

1. Introduction

The FPU core I am going to study is "*fpu*" by Rudolf Usselmann [1]. It complies the IEEE 754 standard, and is able to do addition, subtraction, multiplication and division operations for two single precision floating point numbers. I uses an IEEE paper format template for OpenOffice and LibreOffice to write this paper. The template is designed by fullmetalsoa [2].

Because I am so busy recently, I have little time to work on this project. I feel so sorry about that, but I have no choice… I only have 86,400 seconds a day! I have done my best.

2. FPU Code Analysis

The IEEE 754 floating point number representation specifies both single precision (4 bytes) and double precision (8 bytes) encodings. For single precision encodings, please refer to table 1. The decimal values in the third row of the table is called **denormalized numbers**; value "*NaN"* in the fifth row means **not a number**.

The standard also defines five rounding types. The first one is **round half to even**, a.k.a. **unbiased rounding**, which rounds the fraction to the nearest even value; the second **round half away from zero**, which rounds the fraction to the nearest value above if it is positive, and to the nearest value below if it is negative; the third **round toward zero**, which truncates the fraction; the forth **round toward positive infinity**, which rounds the fraction up to its ceiling; the fifth **round toward negative infinity**, which rounds the fraction down to its floor [3]-[5].

Table 1. IEEE 754 single precision encodings

Sign (1 bits)	Exponent (8 bits)	Mantissa (23 bits)	Decimal Value
0/1	1 to 254	anything	$(-1)^S \times (1.M)_2 \times 2^{(E-127)}$
0/1	0	nonzero	$(-1)^S \times (0.M)_2 \times 2^{-126}$
0/1	0	0	$(-1)^S \times 0$
0/1	255	0	$(-1)^S \times \infty$
0/1	255	nonzero	NaN

2.1. Second-order headings

As in this heading, they should be Times 11-point boldface, initially capitalized, flush left, with one blank line before, and one after.

3. Porting to FPGA

I have not ported the FPU core to an FPGA yet due to insufficient time.

4. Experiments

Author names and affiliations are to be centered beneath the title and printed in Times 12-point, non-boldface type. Multiple authors may be shown in a two- or three-column format, with their affiliations italicized and centered below their respective names. Include e-mail addresses if possible. Author information should be followed by two 12-point blank lines.

5. Conclusions

Use footnotes sparingly (or not at all) and place them at the bottom of the column on the page on which they are referenced. Use Times 8-point type, single-spaced. To help your readers, avoid using footnotes altogether and include necessary peripheral observations in the text (within parentheses, if you prefer, as in this sentence).

6. References

[1] Rudolf Usselmann (2000, September 16). *Open Floating Point Unit*, The Free IP Cores Projects [Online]. Available: http://opencores.org/project,fpu

[2] fullmetalsoa. *Author's Instruction for the Preparation of Regular Papers* [Online]. Available: http://templates.openoffice.org/en/template/ieee-paper-format

[3] *IEEE Standard for Floating-Point Arithmetic*, IEEE Standard 754-2008, August 29, 2008.

[4] various contributors (2016, January 9). *Floating Point*, Wikipedia [Online]. Available: https://en.wikipedia.org/wiki/Floating_point

[5] Kevin J. Brewer and Quanfei Wen (1999, May 28). *IEEE-754 Floating-Point Conversion from Floating-Point to Hexadecimal*, City University of New York [Online]. Available: http://babbage.cs.qc.edu/courses/cs341/IEEE-754.html