ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

High Speed IEEE-754 Double Precision Floating Point Adder/Subtractor and Multiplier Using Verilog

K.Sridhar
M.Tech,(VLSI Design),
Dept ECE,
SVCET, Chittoor.

S.Nagaraj, M.Tech
Associate Professor,
Dept ECE,
SVCET, Chittoor.

Abstract:

Floating Point(FP) addition, subtraction and multiplication are more used in large set of scientific and signal processing computation. A high speed floating point double precision adder/subtractor and multiplier are implemented on a Virtex -6 FPGA. In addition, the proposed design are compliant with IEEE-754 format and handles overflow,under flow,rounding and various exception conditions.The adder/subractor and multiplier designs achieved the operating frequencies of 363.76 MHz and 414.714 MHz with an area of 660 and 648 slices respectively.

Keywords:

Double precision, floating point,adder/subtractor, multiplier, FPGA, IEEE 754, Virtex-6.

1. INTRODUCTION:

The real numbers represented in binary format are known as floating point numers.Based on IEEE -754 standard,floating point formats are classified into binary and decimal interchange formats.floating oint multipliers are very important in dsp applications.This paper focus on double precision normalized binary interchange formats. Figure 1 shows the IEEE-754 double precision binary format representation.Sign(S) is represented with one bit , exponent (E) and fraction(M or Mantissa) are represented with eleven and fifty two bits respectively.For a number is said to be a normalized number,it must consist of 'one' in the MSB of the significant and exponent is greater than zero and smaller than 1023.The real number is represented by equations (1) and (2).

Figure 1.IEEE-754 double precision floating point format.

$$Z = (-1S) * 2 (E - Bias) * (1.M) \quad (1)$$

$$Value = (-1Sign\ bit) * 2 (Exponent -1023) * (1.Mantissa) \quad (2)$$

Floating point implementation on FPGAs has been the interest of many researches.Oklobdzija implemented 32-bit and 64-bit leading zero detector (LZD) circuit using CMOS and ECL technology [1]. In [2], Pavle Belanovic and Miriam Lesser implemented reconfigurable floating point arithmetic unit using VHDL, which is mapped on to Xilinx XCV1000 FPGA.K>Scott Hemmert Keith and D.Underwood implemented open source library of highly optimized floating point units for Xilinx FPGAs. The units are fully IEEE compliant.The double precision add and multiply achieved the operating frequency of 230MHz using a 10 stage adder pipeline and a 12 stage multiplier pipelined [3].In [4],floating point adder was implemented using Leader One Predictor (LOP) algorithm instead of Leading One Detector (LOD) algorithm.

The main function of the LOP is to predict the leaading zeros in the addition result,working with the 2's complement adder.Dhiraj Sangwan and Mahesh K. Yadav implemented adder/subtrator and multiplication units for floating point arithmetic using VHDL. The floating point multiplication operation was implemented using sequential architecture based on booth's radix-4 recording algorithm.For floating point addition, the sequential addition could have been complex so the combinational architecture has been implemented [5]. In [6] , double precision floating point adder/subtractor was implemented using dynamic shifter, LOD,priority encoder.The design achieved the operating frequency of 353MHz for a latency of 12 clock cycles.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

In [7], an IEEE-754 single precision pipelined floating point multiplier is implemented on multiple FPGAs (4 Actel A 1280).Nabeel Shirazi,Walters, and Peter Athanas implemented custom 16/18 bit three stage pipelined floating point multiplier, that doesn't support rounding modes [8]. L.Locua, T.A.Cook, W.H. Johnson [9] implemented a single preicsion floating point multiplier by using a digit-serial multiplier and Altera FLEX 8000.The design achieved 2.3 Mflops and doesn't support rounding modes. In [10] , a parametedizable floating point multiplier is implemented using five stages pipeline,Handle-C software and Xilinx XCV1000 FPGA.The design achieved the operating frequency of 28Mflops.The floating point unit [11] is implemented using the primitives of Xilinx Virtex II FPGA. The design achieved the operating frequency of 100 MHz with a latency of 4 clock cycles.Mohamed A1-Ashrafy,Ashraf Salem,and wagdy Anis [12] implemented an efficient IEEE-754 single precision floating point multiplier and targeted for Xilinx Virtex-5 FPGA.The multiplier handles the overflow and underflow cases but rounding is not implemented .The design achieves 301 MFLOPs with latency of three clock cycles.The multiplier was verified against Xilinx floating point multiplier core.

The double precision floating point adder/subtractor and multiplier presented here is based on IEEE-754 binary floating standard.We have designed a high speed double precision floating point adder/subtractor and multiplier using Verilog language and ported on Xilinx Vertex-6 FPGA.Adder/subtractor and multiplier operates at very high frequencies of 363.78 and 414.714 Mflops and occupies 660 and 648 slicies respectively. It handles the overflow,underflow cases and rouding mode.

2.IMPLEMENTATION OF DOUBLE PRECISION FLOATING POINT ADDER/SUBTRACTOR:

The black boxand block view diagram of double precision floating point adder/subtractor is shown in figures 2 and 3 respectively .The input operands are seperated into their sign,mantissa and exponent components.This module has inputs opa and opb of 64-bit width and clk,enable,rst are of 1-bit width.One of the operands is applied at opa and other operand at opb. Larger operand goes into 'mantissa_large' and 'exponent _large',simillary the smalleroperand goes into 'mantissa_small' and 'exponent_small' .

To determine which operand is larger,compare only the exponents of the two operands,so in fact ,if the exponent are equal,the smaller operand might populate the mantissa_larger and exponent_larger registers. This is not an issue because the reason the operands are compared is to find the operand with the larger exponent,so that the mantissa of the operand with the smaller exponent can be right shited before performing the addition.If the exponent are equal,the mantissa are added without shifting. The inter-connection of sub-modules of double precision floating point adder/subtractor is shown in figure .Subtration is similar to addition in that you need to calculate the difference in the exponents between the two operands,and then shift the mantissa of the smaller exponent to the right before subtracting.The flow chart of double precision floating point adder/subtractor is shown in figure .

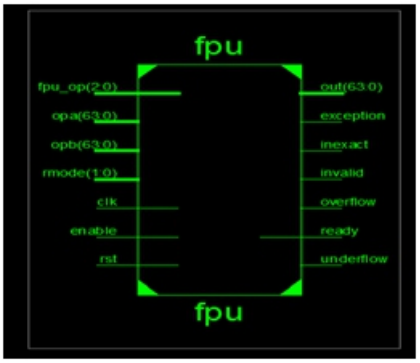

Figure 2.Black box view of double precision floating point adder/subtractor.

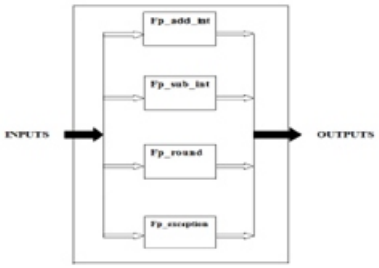

Figure 3. Block diagram of double precision floating point adder/subtractor.

3.IMPLEMENTATION OF DOUBLE PRECISION FLOATING POINT MULTIPLIER:

3.1 IMPLEMENTATION:

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

In this paper we implemented a double precision floating point multiplier withexceptions and rounding. Figure 4 shows the multiplier structure that includes exponent addition,siginificand multiplication,and sign calculation.Figure 5 shows the multiplier,exceptions and rounding that are independent and are done in parallel.

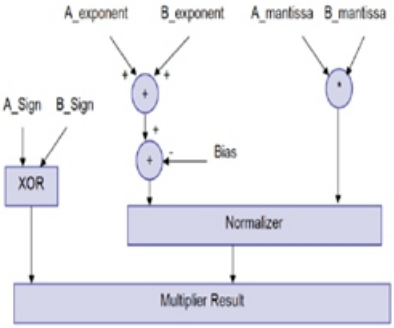

Figure 4. Multiplier structure

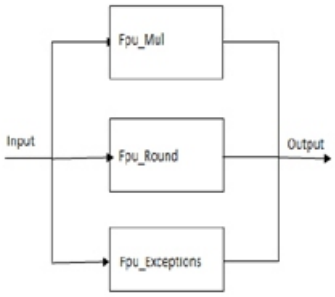

Figure 5.Multiplier structure with rounding and exceptions

3.1.1 MULTIPLIER:

The black box view of the double precision floating point multiplier is shown in figure 6.The multiplier recieves two 64-bit floating point numbers. First these numbers are unpacked by separating the numbers into sign,exponent,and mantissa bits.The sign logic is a simple XOR.

The exponents of two numbers are added and then subtracted with a bias number i.e.,1023.Mantissa multiplier block performs multiplication operation.After this the output of mantissa division is normalized, i.e., if the MSB of the result obtained is not 1,then it is left shifted to make the MSB 1.If changes are made by shifting then corresponding changes to be made in exponent also.

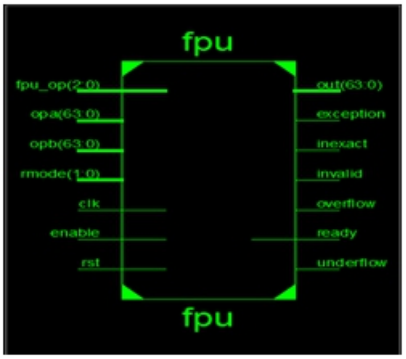

Figure 6. Black box view of floating point double precision multiplier.

4.ROUNDING AND EXCEPTIONS:

The IEEE standard specifies four rounding modes round to nearest,round to zero,round to positive infinity, and round to negative infinity. Table 1 shows the rounding modes selected for various bit combinations of rmode.Based on the rounding changes to the mantissa corresponding changes has to be made in the exponent part also.

TABLE 1. ROUNDING MODES SELECTED FOR VARIOUS BIT COMBINATIONS OF MODE:

Bit combination	Rounding Mode
00	round_nearest_even
01	round_to_zero
10	round_up
11	round_down

5.RESULTS:

The double precision floating point adder/subtractor and multiplier designs were simulated in mdelsim and synthesized using Xilinx ISE which are mapped on to Virtex-6 FPGA.the simulation results of 64-bit floating point double precision adder/subtractor and multiplier are shown in figure 7 and 8 respectively.the ‘opa’ and ‘opb’ are the inputs and ‘out’ is the output.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

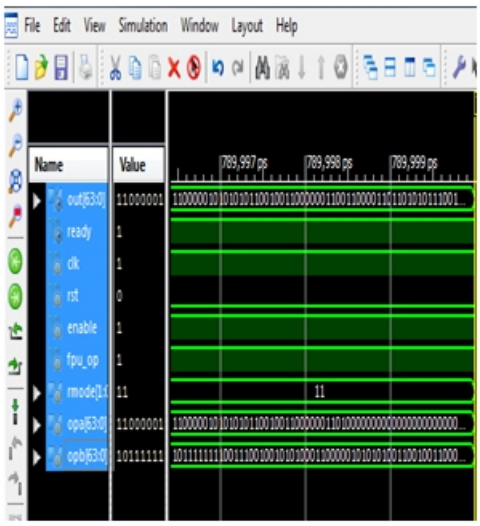

Figure 7.Simulation result of double precision floating point adder/subtractor.

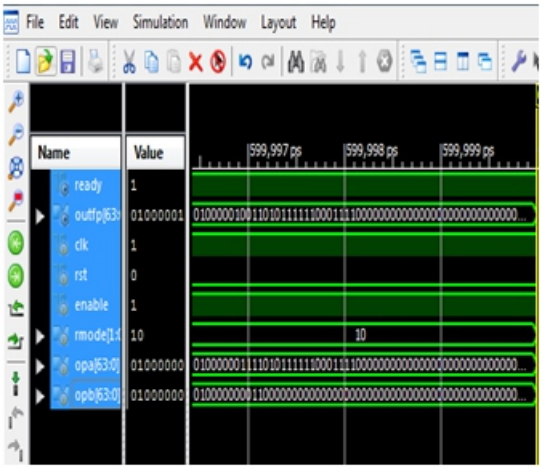

Figure 8. Simulation result of double precision floating point multiplier.

6. CONCLUSION:

The double precision floating point adder/subtractor and multiplier support the IEEE-754 binary interchange format,targeted on a Xilinx Virtex-6 FPGA. it provides more accuracy when compared to single precision floating point.These designs handles the overflow,underflow,rounding mode and various exception conditios.

REFERENCES:

[1] V. Oklobdzija, "An algorithmic and novel design of a leading zero detector circuit: comparison with logic synthesis", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 2, no. 1, (1994) March, pp. 124–128.

[2] P. Belanovic and M. Leeser, "A Library of Parameterized Floating-Point Modules and Their Use", in 12th International Conference on Field-Programmable Logic and Applications (FPL-02). London, UK: Springer-Verlag, (2002) September, pp. 657–666.

[3] K. Hemmert and K. Underwood, "Open Source High Performance Floating-Point Modules", in 14th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM-06), (2006) April, pp. 349–350.

[4] A. Malik and S. -B. Ko, "A Study on the Floating-Point Adder in FPGAs", in Canadian Conference on Electrical and Computer Engineering (CCECE-06), (2006) May, pp. 86–89.

[5] D. Sangwan and M. K. Yadav, "Design and Implementation of Adder/Subtractor and Multiplication Units for Floating-Point Arithmetic", in International Journal of Electronics Engineering, (2010), pp. 197-203.

[6] M. K. Jaiswal and R. C. C. Cheung, "High Performance FPGA Implementation of Double Precision Floating Point Adder/Subtractor", in International Journal of Hybrid Information Technology, vol. 4, no. 4, (2011) October.

[7] B. Fagin and C. Renard, "Field Programmable Gate Arrays and Floating Point Arithmetic", IEEE Transactions on VLSI, vol. 2, no. 3, (1994), pp. 365–367.

[8] N. Shirazi, A. Walters and P. Athanas, "Quantitative Analysis of Floating Point Arithmetic on FPGA Based Custom Computing Machines", Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines (FCCM95), (1995), pp. 155–162.

[9] L. Louca, T. A. Cook and W. H. Johnson, "Implementation of IEEE Single Precision Floating Point Addition and Multiplication on FPGAs", Proceedings of 83rd IEEE Symposium on FPGAs for Custom Computing Machines (FCCM96), (1996), pp. 107–116.

[10] A. Jaenicke and W. Luk, "Parameterized Floating-Point Arithmetic on FPGAs", Proc. of IEEE ICASSP, vol. 2, (2001), pp. 897-900.

[11] B. Lee and N. Burgess, "Parameterisable Floating-point Operations on FPGA", Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems, and Computers, (2002).

[12] M. Al-Ashrafy, A. Salem, W. Anis, "An Efficient Implementation of Floating Point Multiplier", Saudi International Electronics, Communications and Photonics Conference (SIECPC), (2011) April 24-26, pp. 1-5.