International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 07, July-2015

Double Precision Floating Point Arithmetic Unit Implementation- A Review

Prabhjot Kaur
Arni University
Indora, HP

Rajiv Ranjan
Arni University
Indora,HP

Raminder Preet Pal Singh
Arni University
Indora,HP

Onkar Singh
Arni University
Indora, HP

***Abstract* - Arithmetic circuits form an important class of circuits in digital systems. Since the invention of FPGAs, the increase in their size and performance has allowed designers to use FPGAs for more complex designs. Very large number and very small number is very hard to represent by fixed point unit so these values can be represented using the IEEE-754 standard based floating point representation. Floating point unit is a unit which is used to perform various mathematical operations such as addition, subtraction etc. This paper reviewed various floating point arithmetic unit implementations by using IEEE-754 standard.**

Keywords - IEEE-754, Floating Point Unit, Verilog, Arithmetic Unit.

1. INTRODUCTION

The digital arithmetic operations are very important in the design of digital processors and application specific systems. Arithmetic circuits play an important role in digital systems. Advances in process technology have led to dramatic increase in FPGA densities and speeds. FPGA's are new becoming more suitable for supporting design with dense computation and high operating frequencies. Floating Point Units are widely used in digital application such as digital signal processing, digital image processing and multimedia.

The term floating point is derived from the meaning that there is no fixed numbers of digits before and after the decimal point can float. There was also a representation in which the number of digits before and after the decimal are set, called fixed point representation.

The IEEE 754 floating point formats need three subfields: sign, exponent and fraction.The IEEE Standard for floating point defines the format of the numbers and also specifies various rounding modes that determine the accuracy of the result.

***IEEE Single Precision Format*:** The IEEE single precision format uses 32 bits for representing a floating point number.

Sign	Exponent	Fraction
1 bit	8 bits	23 bits

Table 1: IEEE single precision floating point format

***IEEE Double Precision Format*:** The IEEE double precision format uses 64 bits for representing a floating point number.

Sign	Exponent	Fraction
1 bit	11 bits	52 bits

Table 2: IEEE double precision floating-point format

2. FLOATING POINT ARITHMETIC UNIT

The unit supports basic four arithmetic operations: Add, Subtract, Multiply and Divide. All arithmetic operations have three stages:

Pre-normalize*:* The operands are transformed into formats that makes them easy and efficient to handle internally.

***Arithmetic core*:** The basic arithmetic operations are done here.

***Post-normalize*:** The result will be normalized if possible and then transformed into the format specified by the IEEE standard.

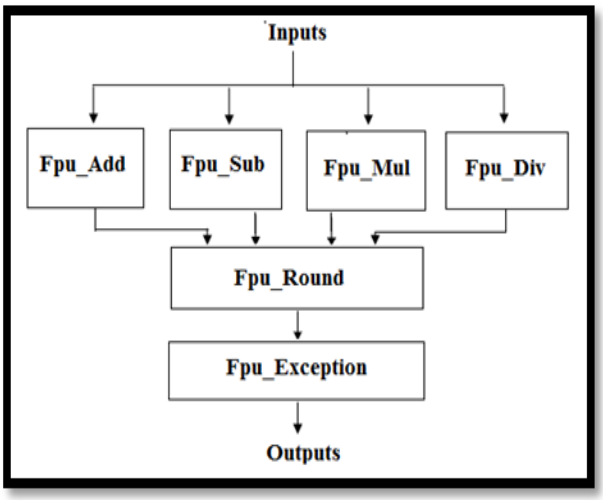

Figure 1:-Block diagram of FPU

3. DIFFERENT TECHNIQUES TO IMPLEMENT FLOATING POINT ARITHMETIC UNIT.

3.1 FPU Design Using Aligned Partition

In this technique present an automatic method to synthesize general FPU by aligned partition. In this method the grouping of floating point numbers is done by zones. By using aligned partition algorithm the

domain handles excessive floating point segment boundaries at high speed and low hardware cost.

3.2 FPU Design using different Algorithms

In this technique the floating point arithmetic unit is designed using different algorithms which provides a comparison in performance. For floating point add/subtract three algorithms were discussed: standard, leading one predictor (LOP), and 2-path. In standard algorithm the exponent comparator is implemented with a subtractor and a multiplexer which is used to perform addition and subtraction. In LOP technique the no. of preceding 1's or 0's in the result can be predicted directly from the input operands to within an error of 1-bit, in parallel with addition/subtraction step. By using this technique the area may be increased but overall latency will decreased. In 2-path algorithm there are two data paths. Due to this it will find the smallest latency for adders. The preshifter may be eliminated in this algorithm. For the multiplication operation a Radix-4 modified booth encoded Wallace multiplier is used. By using this algorithm the number of levels required in the Wallace tree becomes less so it reduces area. To perform division simple division algorithm is used which perform division operation

3.3 FPU Design by using Latches

In this technique a floating-point arithmetic unit is designed using latches. By using latches in place of flip flops the maximum frequency of the design may be increased but it requires a lot of efforts to make a latch base design. In this technique large combinational paths can be compensated by shorter path delays in the subsequent logic gates. All the operations are performed on 64 bit operands. Addition and subtraction operations are performed firstly by normalizing the exponent part and then perform the operation add/subtract. The complete design is captured in Verilog Hardware description language (HDL), tested in simulation using Questa Sim, placed and routed on a Vertex 5 FPGA from Xilinx. The overall performance is increased in this design.

3.4 FPU Design by Modified Units

In this technique a high speed ASIC implementation of a floating point arithmetic unit which can perform addition, subtraction, multiplication, division functions on 32-bit operands. To perform addition operation various NAND, XOR gates and carry look ahead adders are used to implemented but to modify the previous method 24 bit carry look ahead adder has been constructed and performed the addition operation. For subtraction 2's complement of second number is taken and then the number is added. To perform multiplication operation i.e. to multiply the mantissas Bit Pair Recoding (or Modified Booth Encoding) algorithm has been used, because of which the number of partial products get reduces by about a factor of two, but to further increase the efficiency of the algorithm and decrease the time complexity, Karatsuba algorithm can be paired with the bit pair recoding algorithm. Division of the fractional bits has been performed by using non restoring division algorithm which is modified to improve the delay. To further decrease the delay order of the computations is rearranged also one adder and one inverter are removed by using a new quotient digit converter. So, the delay from one adder and one inverter connected in series will be eliminated.

3.5 FPU Design by using Pipelining

In this technique reconfigurable pipelined Floating-Point hardware architecture is designed by exploring the similarities of individual floating-point operations; this architecture is capable of handling floating point addition, subtraction, multiplication and comparison in a pipelined manner resulting in an increase in performance in terms of area and latency. Floating-point arithmetic or logical operation is high as each operation is divided into three stages: pre-normalization, arithmetic-operation and post-normalization. The architectural explorations include: Kogge-Stone Adder, Carry-Skip adder, carry-ripple adder, carry-look-ahead adder, Wallace-Tree multiplier, Serial multiplier, Parallel Multiplier, Boot's multiplier, Array Multiplier, Barrel shifter, Zero-detector using XOR gates and Cascaded comparator. Unified Reconfigurable Floating-Point Architecture results in an improvement in latency by a factor of 3 and an improvement in area (number of LUTs) by a factor of 1.5 as compared to the summed up area of individual floating-point units.

4. ADVANTAGES OF FLOATING POINT UNIT

The advantage of floating-point representation over fixed-point representation is that it can support a much wider range of values.. The floating-point format needs slightly more storage (to encode the position of the radix point), floating-point numbers achieve their greater range at the expense of slightly less precision. Floating Point numbers has more flexibility than Fixed-point numbers which has limited or no flexibility. The internal representations of data in floating-point hardware are more exact than in fixed-point, ensuring greater accuracy in the results.

5. APPLICATIONS OF FLOATING POINT UNIT

The applications of using the floating-point format can be readily seen by contrasting the data set requirements of video and audio applications. Floating Point units are used in high speed objects recognition system and also in high performance computer systems as well as embedded systems and mobile applications. Floating Point units are used in high speed objects recognition system [2] and also in high performance computer systems as well as embedded systems and mobile applications[18]. In medical image recognition, greater accuracy supports the many levels of signal input from light, x-rays, ultrasound and other sources that must be defined and processed to create output images with useful diagnostic information. By contrast with these applications, the enormous communications market is better served by floating-point devices.

REFERENCES

[1] Liangwei Ge, Song Chen,Yuichi, nakamura,Takeshei Yoshimura"A Synthesis method of General Floating Point Arithmetic Unit by Partial Partition". A 23rd International technical Conference on circuit/system,computers and communication.

[2] Yedukondala Rao Veeranki, R. Nakkeeran," Spartan 3E Synthesizable FPGA Based Floating-Point Arithmetic Unit" International Journal of Computer Trends and Technology (IJCTT) - volume4Issue4 –April 2013 ISSN: 2231-2803.

[3] Onkar Singh, Kanika Sharma" Design and Implementation of High Speed Area Efficient Double Precision Floating Point Arithmetic Unit" IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 10, Issue 1, Ver. 1 (Jan - Feb. 2015), PP 49-54

[4] Ushasree G, R Dhanabal, Sarat Kumar Sahoo "Implementation of a High Speed Single Precision Floating Point Unit using Verilog" International Journal of Computer Applications National conference on VSLI and Embedded systems, pp.32-36, 2013

[5] Sateesh Reddy, Vinit T Kanojia "Unified Reconfigurable Floating-Point Pipelined Architecture" International Journal of Advanced Engineering Sciences and Technologies, Vol No. 7, Issue No. 2, pp. 271 – 275, 2011

[6] Karan Ghmber, Sharmelle Thangjam "Performance Analysis of Floating Point Adder using VHDL on Reconfigurable Hardware" International Journal of Computer Application, Vol.46-No 9, May 2012

[7] Paschalakis, S., Lee, P., "Double Precision Floating-Point Arithmetic on FPGAs", In Proc. 2003 2nd IEEE International Conference on Field Programmable Technology (FPT '03), Tokyo, Japan, pp. 352-358, 2003

[8] Shamna.K, S.R Ramesh "Design and Implementation of an Optimized Double Precision Floating Point Divider on FPGA", International Journal of Advanced Science and Technology, Vol. 18, pp.41-48, May 2010

[9] Ms. Anjana Sasidharan, Mr. M.K. Arun, "VHDL Implementation Of IEEE 754 Floating Point Unit" International Journal Of Advanced information and Communication Technology, Volume 1, Issue 2, June 2014