DOI: 10.35598/mcfpga.2021.004

FPGA Implementation of Floating-Point Significand Multiplier

Artem Didenko
ORCID 0000-0001-7556-2276
Bachelor student of Computer Engineering
National University Zaporizhzhia Polytechnic
Zaporizhzhia, Ukraine
didenko.art121@gmail.com

Irina Zeleneva
ORCID 0000-0002-4042-4540
Department of Computer Systems and Networks
National University Zaporizhzhia Polytechnic
Zaporizhzhia, Ukraine
irina.zeleneva@gmail.com

***Abstract*—Development of high-performance computational devices needs fast methods of operations with numbers which are presented in floating-point format. One of the most common approaches to build a significand multiplier is using a Wallace CSA-adder tree structure which shows good performance in adding partial sums. Significand multiplier was implemented in FPGA as a part of floating-point multiplier which works with single precision numbers in IEEE-754 standard.**

Keywords—multiplier, floating-point number, carry-save adder, FPGA

I. Introduction

FPGA finds its usage in many different fields such as sorting and data search, audio, video and image processing, cryptography, processing of data packets in networks, random numbers generation. Among new important markets one can distinguish finance, bioinformatics and hardware for radio communication [1]. High performance of FPGA comes from its flexibility, which gives a possibility to realize optimized devices for specific purposes [2]. It is worth mentioning that inner structure of FPGA aslo allows to parallelize data processing.

Lots of computational devices use floating point arithmetic in their operations with numbers. Floating point numbers have such advantage as wide range, which makes them suitable in digital signal processing and scientific modeling.

Multiplication is one of most common operations which computational devices perform. There are a lot of methods of implementing floating point multiplication in hardware. Significand or mantissa multiplication is the main part of floating point multiplication algorithm, therefore much attention should be paid to a structure of significand multiplier.

This paper describes FPGA implementation of significand multiplier which is based on CSA-adder tree. CSA-adder tree performs summation of partial sums that are generated during significands multiplication process.

II. Theoretical Background

A. IEEE-754 Standard

In modern computers and other computational devices floating point numbers are represented according to IEEE-754 standard. This standard defines single and double precision floating point numbers. Single precesion numbers are represented as one which consists of one sign bit, 8 exponent bits and 23 significand bits.

B. Wallace multiplier

Multiplication of significands of floating-point numbers uses the same methods as fixed-point numbers multiplication. According to [3] Wallace multiplier have better performance than array multiplier which is another common method of multiplication. Wallace multiplier consists of CSA-adder tree.

Wallace multiplier is a method of acceleration of multiplication which allows to add several partial sums simultaneously. Wallace CSA-adder tree reduces several k-bit operands to two k+2 bit operands. Unlike Dadda CSA-adder tree Wallace CSA-adder tree reduces the number of operands as early as possible what minimizes the overall delay by making the final carry-propagation adder as short as possible [4].

Another method that can speed up multiplication process is high-radix multiplication which decreases number of partial sums. In this work these two methods were combined together.

C. CSA-adder

One way to add more than two operands simultaneously is to use CSA-adder. CSA-adder is a full adder without carry propagation chain. On the output of CSA-adder there are sum and carry vectors, which then can be added using usual carry-propagation adder. A typical structure of CSA-adder is shown below.

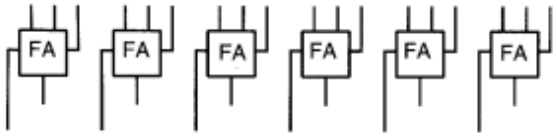

Fig. 1. Structure of CSA-adder [4]

As it can be seen from fig. 1 CSA-adder adds three operands simultaneously. Another structure that adds several operands simultaneously is 4:2 compressor, which is in fact two CSA-adders. The structure of 4:2 compressor is shown below.

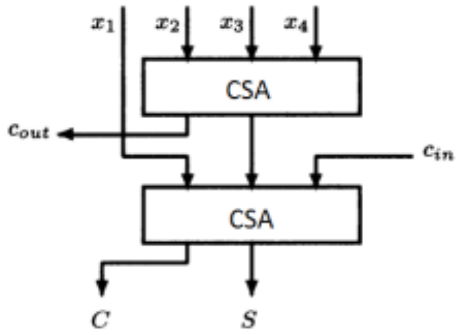

Fig. 2. Structure of 4:2 compressor [5]

The use of 4:2 compressors allows to simplify the structure of a multiplier.

III. FGPA IMPLEMENTATION

As mentioned previously significand of floating point number in IEEE-754 standard consists of 23 bits with one implicit bit will lead to 24 bits. Therefore there will be 24 partial sums. In this work Wallace radix-4 multiplier is used what decreases number of partial sums to 12. The developed structures of significand multiplier and CSA-adder tree are shown below.

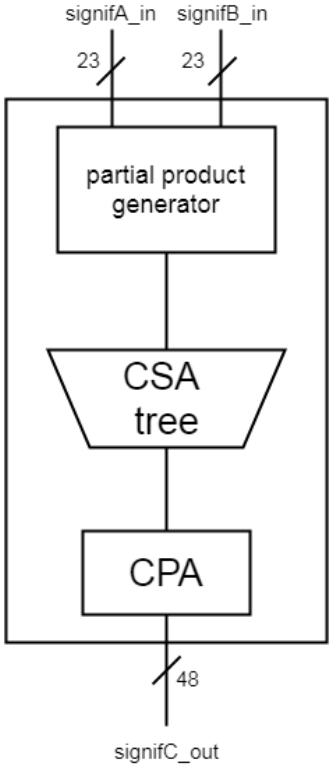

Fig. 3. Structure of significand multiplier

As it can be seen from fig. 3 significand multiplier consists of partial product generator module, CSA-adder tree which reduces number of partial sums to two operands and carry-propagation adder which adds these two operands.

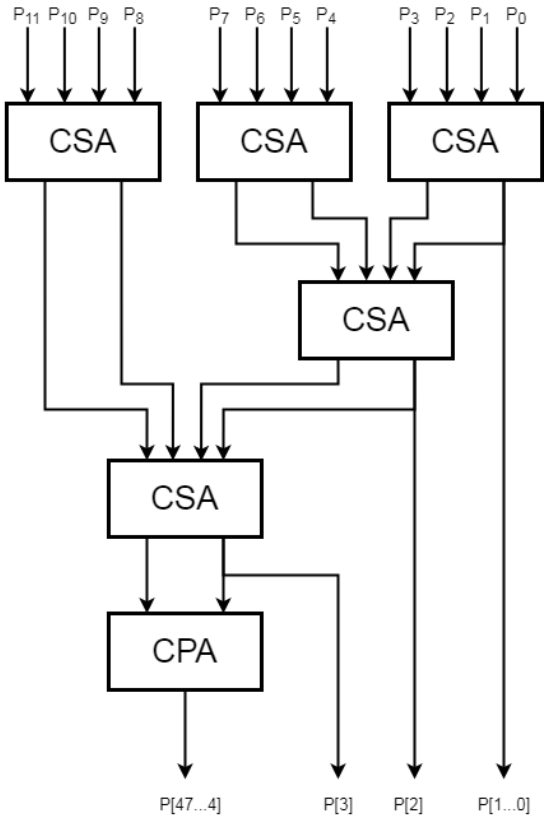

Fig. 4. Structure of Wallace CSA-adder tree

Significand multiplier was simulated in Modelsim and implemented on FPGA Altera Cyclone IV using VHDL. According to compilation report significand multiplier utilizes 1,196 logic elements.

IV. CONCLUSION

Significand multiplier which is based on CSA-adder tree is one of the most common ways to implement high-performance floating point multipliers. The use of significand multiplier which is based on Wallace tree in FPGA projects allows to speed-up multiplication and save embedded FPGA multiplier elements.

REFERENCES

[1] Leong, P. Recent Trends in FPGA Architectures and Applications / P.H.W. Leong // 4th IEEE International Symposium on Electronic Design, Test and Applications (delta 2008), Hong Kong, 23-25 January 2008: proceedings. – Los Alamitos: IEEE, 2008. – pp.137–141.

[2] Asano, S. Performance comparison of FPGA, GPU and CPU in image processing / S. Asano, T. Maruyama, Y. Yamaguchi // 2009 International Conference on Field Programmable Logic and Applications, Prague, 31 August - 2 September 2009 : proceedings. – Los Alamitos: IEEE, 2009. – P.126 –131.

[3] Bečvář M. Fixed-Point Arithmetic in FPGA / M. Bečvář, P. Štukjunger // Acta Polytechnica Vol. 45 No. 2/2005. – pp. 67–72.

[4] Parhami, B. Computer Arithmetic: Algorithms and Hardware Designs, 2nd edition / B. Parhami. – New York : Oxford University Press, 2010. – 672 p.

[5] Koren, I. Computer Arithmetic Algorithms 2nd Edition / I. Koren. – Natick : A K Peters/CRC Press, 2001. – 296p.