

A Butterfly Architecture Based on Binary Signed-Digit Representation of Floating-Point

I. PRADEEP KUMAR¹, P. RAJYALAXMI², K. VIJAYA PRASAD³

¹PG Scholar, Dept of VLSI & Embedded, DJR College of Engineering & Technology, AP, India,

E-mail: ila.pradeepkumar@gmail.com.

²Associate Professor, Dept of VLSI & Embedded, DJR College of Engineering & Technology, AP, India.

³Professor & HOD, Dept of VLSI & Embedded, DJR College of Engineering & Technology, AP, India,

E-mail: vijayaprasad835@gmail.com.

Abstract: Fast Fourier transform (FFT) coprocessor, having a significant impact on the performance of communication systems, has been a hot topic of research for many years. Applying floating-point (FP) arithmetic to FFT architectures, specifically butterfly units, has become more popular recently. It offloads compute-intensive tasks from general-purpose processors by dismissing FP concerns (e.g., scaling and overflow/underflow). However, the major downside of FP butterfly is its slowness in comparison with its fixed-point counterpart. A carry-limited BSD adder is proposed and used in the three-operand adder and the parallel BSD multiplier so as to improve the speed of the FDPA unit. Moreover, modified Booth encoding is used to accelerate the BSD multiplier. The synthesis results show that the proposed FP butterfly architecture is much faster than previous counterparts but at the cost of more area.

Keywords: Fast Fourier Transform (FFT), Floating-Point (FP), BSD Multiplier.

I. INTRODUCTION

Fast Fourier transform (FFT) circuitry consists of several consecutive multipliers and adders over complex numbers; hence an appropriate number representation must be chosen wisely. Most of the FFT architectures have been using fixed-point arithmetic, until recently that FFTs based on floating-point (FP) operations grow. The main advantage of FP over fixed-point arithmetic is the wide dynamic range it introduces; but at the expense of higher cost. Moreover, use of IEEE-754-2008 standard for FP arithmetic allows for an FFT coprocessor in collaboration with general purpose processors. This offloads compute-intensive tasks from the processors and leads to higher performance. The main drawback of the FP operations is their slowness in comparison with the fixed-point counterparts. A way to speed up the FP arithmetic is to merge several operations in a single FP unit, and hence save delay, area, and power consumption. Using redundant number systems is another well-known way of overcoming FP slowness, where there is no word-wide carry propagation within the intermediate operations. The conversion, from no redundant, to a redundant format is a carry-free operation; however, the reverse conversion requires carry propagation [4]. This makes redundant representation more useful where many consecutive arithmetic operations are performed prior to the final result. This brief proposes a butterfly architecture using redundant FP arithmetic, which is useful for FP FFT coprocessors and contributes to digital signal processing applications.

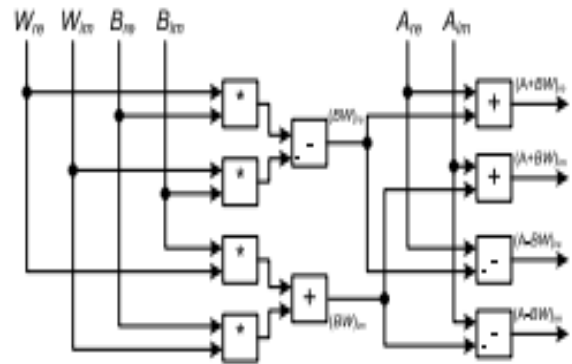


Fig 1. FFT butterfly architecture with expanded complex numbers.

Although there are other works on the use of redundant FP number systems, they are not optimized for butterfly architecture in which both redundant FP multiplier and adder are required. The novelties and techniques used in the proposed design include the following.

II. LITERATURE SURVEY

This section introduces fixed-point computer arithmetic and its limitations, the IEEE-754 floating-point standard, and current usage of combined (fused) arithmetic functions, a quick introduction to the Fast Fourier Transform (FFT), floating-point and FFT error analysis.

A. DSP processors Arithmetic Overview

DSP processors arithmetic is concerned with the hardware realization of mathematical formulas, algorithms and complex models from a theoretical world. Hardware functions calculate arithmetic's in both fixed-point and floating-point (Scientific Notations)

B. IEEE-754 Floating Point Standard

The IEEE Standard for Floating-Point Arithmetic (IEEE-754) is a technical standard established by the Institute of Electrical and Electronics Engineers. It is the most ubiquitous standard for floating-point computations representation in today's microprocessors, including Intel-based Processor's, Macintoshes and UNIX platforms. IEEE floating point numbers have three basic components: a sign, an exponent and a significant. The significant is composed of the fraction and an implicit leading digit

III. FLOATING POINT REPRESENTATION

Most of the DSP applications need floating point numbers multiplication. The possible ways to represent real numbers in binary format floating point numbers are; the IEEE 754 standard represents two floating point formats, Binary interchange format and Decimal interchange format. Single precision normalized binary interchange format is implemented in this design. Representation of single precision binary format is shown in Figure 2; starting from MSB it has a one bit sign (S), an eight bit exponent (E), and a twenty three bit fraction (M or Mantissa). Adding an extra bit to the fraction to form and is defined as significand1. If the exponent is greater than 0 and smaller than 255, and there is 1 in the MSB of the significand then the number is said to be a normalized number.

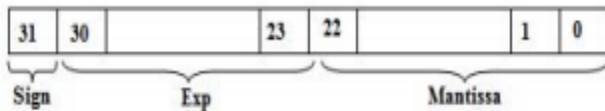


Fig 2. IEEE single precision floating point format.

A. Floating Point Unit

A floating point (FPU), also known as a math co-processor or numeric processor is a specialized co-processor that manipulates numbers more quickly than the basic microprocessor circuitry. The FPU does this by means of instructions that focus entirely on large mathematical operations. Floating point computational logic has long been a mandatory component of high performance computer systems as well as embedded systems and mobile applications. The performance of many modern applications which give a high frequency of floating point operations is often limited by the speed of the floating point hardware.

B. FLOATING POINT MULTIPLICATION

A = 0 10000001 01100 = 5.5, B = 1 10000100 00011 = -35

By following the algorithm the multiplication of A and B is

1. Significand Multiplication:

$$\begin{array}{r} 1.01100 \\ \times 1.00011 \\ \hline 101100 \\ 101100 \\ 000000 \\ 000000 \\ 000000 \\ \hline 101100 \\ 01100000100 \end{array}$$

2. Normalizing the result: 1.1000000100
3. Adding two exponents: 10000001
+10000100
100000101

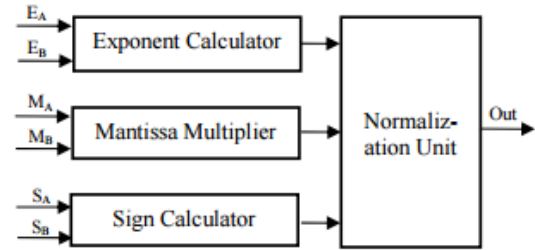


Fig 3. Floating point multiplier block diagram.

IV. PROPOSED METHOD

The FFT could be implemented in hardware based on an efficient algorithm [5] in which the N-input FFT computation is simplified to the computation of two (N/2)-input FFT. Continuing this decomposition leads to 2-input FFT block, also known as butterfly unit. The proposed butterfly unit is actually a complex fused-multiply-add with FP operands. Expanding the complex numbers, Fig. 1 shows the required modules. According to Fig. 1, the constituent operations for butterfly unit are a dot-product (e.g., BreWim + BimWre) followed by an addition/subtraction which leads to the proposed FDPA operation (e.g., BreWim + BimWre + Aim). Implementation details of FDPA, over FP operands, are discussed below.

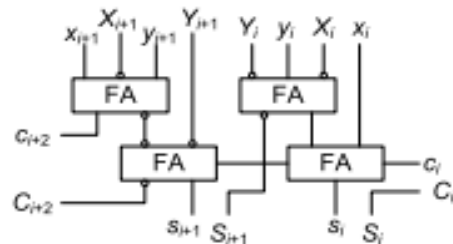


Fig 4. BSD adder (two-digit slice).

Table I. Generation of A Pp

$W_{i+1}^- W_{i+1}^+$	$W_i^- W_i^+$	$\ W_{i+1}^- W_{i+1}^+ W_i^- W_i^+\ $	PP_i
0 0	0 0	0	0
0 0	0 1	1	B
0 0	1 1	-1	-B
0 1	0 0	2	$2 \times B$
1 1	0 0	-2	$-2 \times B$

A Butterfly Architecture Based on Binary Signed-Digit Representation of Floating-Point

A. Proposed Redundant Floating-Point Multiplier

The proposed multiplier, likewise other parallel multipliers, consists of two major steps, namely, partial product generation (PPG) and PP reduction (PPR). However, contrary to the conventional multipliers, our multiplier keeps the product in redundant format and hence there is no need for the final carry-propagating adder. The exponents of the input operands are taken care of in the same way as is done in the conventional FP multipliers; however, normalization and rounding are left to be done in the next block of the butterfly architecture (i.e., three-operand adder). Partial Product Generation: The PPG step of the proposed multiplier is completely different from that of the conventional one because of the representation of the input operands (B , W , B , W). Moreover, given that W_{re} and W_{im} are constants [5], the multiplications in Fig. 1 (over significands) can be computed through a series of shifters and adders.

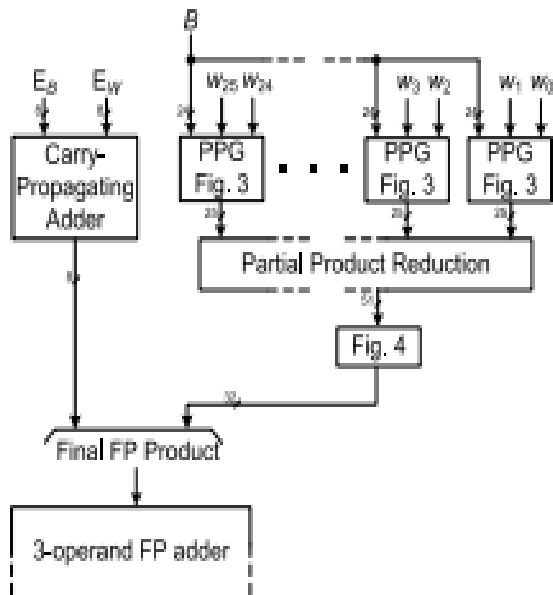


Fig5. Proposed redundant FP multiplier

V. RESULTS AND OUTPUT SCREENS

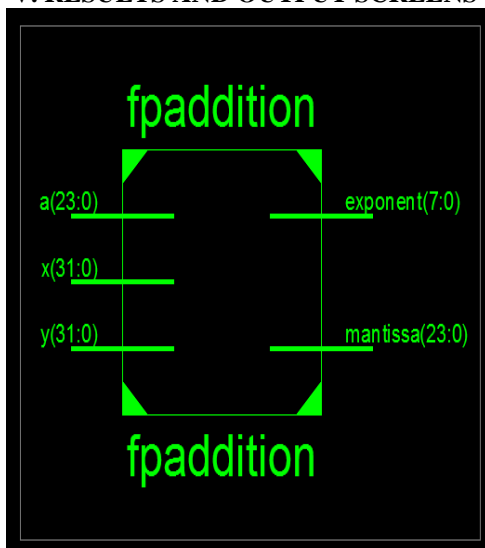


Fig6. Schematic 1.

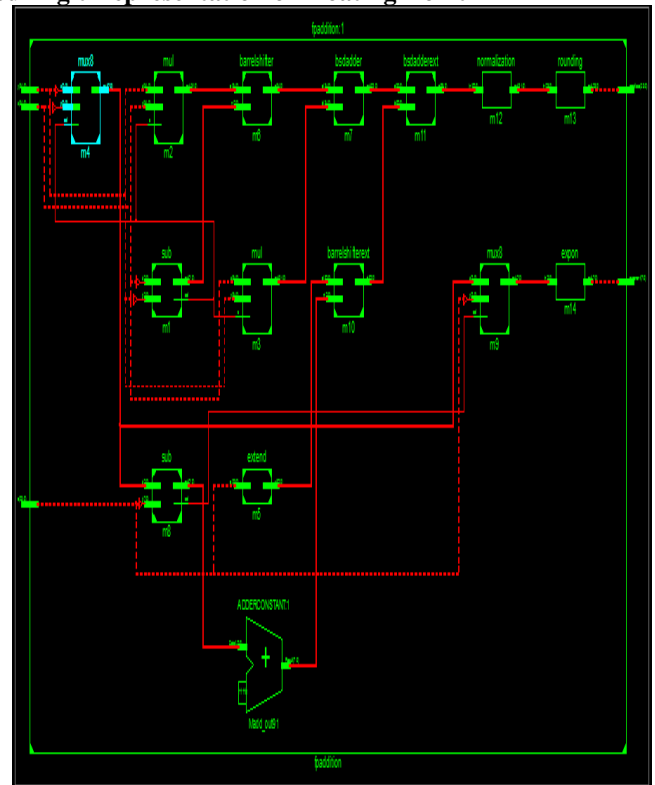


Fig7. RTL Schematic2.

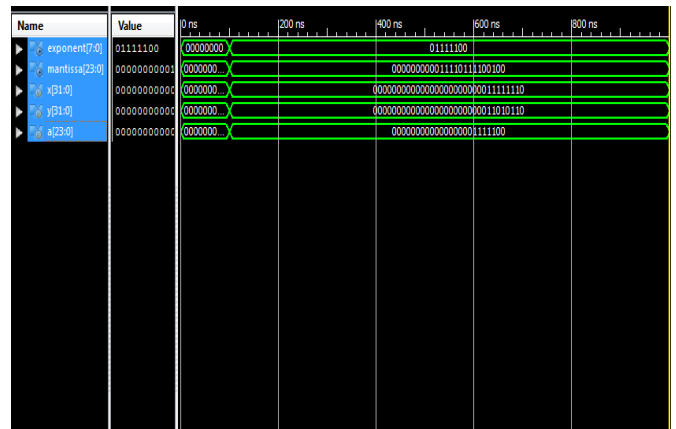


Fig8. Output Waveform.

VI. CONCLUSION

We proposed a high-speed FP butterfly architecture, which is faster than previous works but at the cost of higher area. The reason for this speed improvement is twofold: 1) BSD representation of the significands which eliminates carry-propagation and 2) the new FDPA unit proposed in this brief. This unit combines multiplications and additions required in FP butterfly; thus higher speed is achieved by eliminating extra LZD, normalization, and rounding units. Further research may be envisaged on applying dual-path FP architecture to the three-operand FP adder and using other redundant FP representations. Moreover, use of improved techniques in the termination phase of the design (i.e., redundant LZD, normalization, and rounding) would lead to faster architectures, though higher area costs are expected.

VII. REFERENCES

- [1] E. E. Swartzlander, Jr., and H. H. Saleh, "FFT implementation with fused floating-point operations," IEEE Trans. Comput., vol. 61, no. 2, pp. 284–288, Feb. 2012.
- [2] J. Sohn and E. E. Swartzlander, Jr., "Improved architectures for a floating-point fused dot product unit," in Proc. IEEE 21st Symp. Comput. Arithmetic, Apr. 2013, pp. 41–48.
- [3] IEEE Standard for Floating-Point Arithmetic, IEEE Standard 754-2008, Aug. 2008, pp. 1–58.
- [4] B. Parhami, Computer Arithmetic: Algorithms and Hardware Designs, 2nd ed. New York, NY, USA: Oxford Univ. Press, 2010.
- [5] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," Math. Comput., vol. 19, no. 90, pp. 297–301, Apr. 1965.
- [6] A. F. Tenca, "Multi-operand floating-point addition," in Proc. 19th IEEE Symp. Comput. Arithmetic, Jun. 2009, pp. 161–168.
- [7] Y. Tao, G. Deyuan, F. Xiaoya, and R. Xianglong, "Three-operand floating-point adder," in Proc. 12th IEEE Int. Conf. Comput. Inf. Technol., Oct. 2012, pp. 192–196.
- [8] A. M. Nielsen, D. W. Matula, C. N. Lyu, and G. Even, "An IEEE compliant floating-point adder that conforms with the pipeline packetforwarding paradigm," IEEE Trans. Comput., vol. 49, no. 1, pp. 33–47, Jan. 2000.
- [9] P. Kornerup, "Correcting the normalization shift of redundant binary representations," IEEE Trans. Comput., vol. 58, no. 10, pp. 1435–1439, Oct. 2009.
- [10] 90 nm CMOS090 Design Platform, STMicroelectronics, Geneva, Switzerland, 2007.
- [11] J. H. Min, S.-W. Kim, and E. E. Swartzlander, Jr., "A floating-point fused FFT butterfly arithmetic unit with merged multiple-constant multipliers," in Proc. 45th Asilomar Conf. Signals, Syst. Comput., Nov. 2011, pp. 520–524.

Author's Profile:



I Pradeep Kumar received his B.Tech degree in Electronics and communication engineering and pursuing M.Tech degree in VLSI&ES from DJR College of Engineering & Technology, AP, India.

P Rajyalaxmi M.Tech, received her M.Tech degree and B.Tech degree in Electronics & Communication engineering. She is currently working as an Assoc Professor in, DJR College of Engineering & Technology, AP, India.



Mr. Kommani Vijaya Prasad, Professor & HOD, Department of ECE, DJR College of Engineering & Technology, is a well experienced teacher of Engineering for the past 17 years. He holds his B.E. from Osmania University, M.Tech from JNTU University and professional Diploma in information technology. His student & Teacher experience have created a lot of interest in him for research.