Floating-Point Butterfly Architecture Representation using Hybrid Number Representation

[1]Divya Srivastava, [2]Krishna Raj
[1]M.Tech Student , [2]Professor
[12]Electronics Engineering Department
[12]Harcourt Butler Technical University, Kanpur, India.

Abstract : A Fast Fourier transform architectural processor, will shows the significant effect for the performance of various communication-based subsystems. The FFT function can be pinned down to be consisting of consecutive multiplication and arithmetic adder operations for the complex numbers system representation, which is cited as the butterfly architectural units. The paper deals with the implementation of the floating-point arithmetic number architecture to the FFT system processor architecture, the focus is done mainly on the butterfly architectural units. The implementation of the FFT in FP number system makes the general-purpose processor get rid of the FP overhead of scaling and overflow. As the FP butterfly architectural unit gets beaten by its successor fixed-point butterfly unit on the terms of the computational delay offered, an optimum architecture of the FP butterfly unit could turn into a faster unit using the different number system approach that will help the system to do the faster computation in a fixed duration of time. A carry limited hybridized signed digit adder is proposed in the floating-point adder unit. The simulated result shows that the proposed architecture is faster in computation and have a less delay approach. In this paper, the analysis the gate delay, area and power consumption are done using the using Xilinx 14.2.

***IndexTerms* - Floating-point(FP),FFT, Hybrid signed digits.**

I. Introduction

In real-time digital signaling applications uses data from other devices, these data may be corrupted due to various factor but mostly due to noise. Thus in these DSP applications silicon area, power dissipation or computational processing speed is important than accuracy, less-power, and less-hardware consuming. So, a need for optimum design is required to obtain the desired architecture. A number system has an effect on several levels of the design abstraction it also affects the power dissipation of the system, thus choosing the right number system is quite important. Particularly, an appropriate choice of the number system can reduce the number of algorithmic operations and eventually reduce the power dissipation of the system.

The Fast Fourier transform (FFT) architecture is constituted of multiple combinations of the adders and multiplier units to perform the multiplication of the complex number[10]. The single unit of the adder and multiplier perform the function of a single butterfly architecture. There is a need for suitable number system representation. Earlier the FFT architecture was implemented in the Fixed-point arithmetic number system. Nowadays floating-point (FP) based FFT operations have taken over as a better alternative to the fixed point. The main advantage that makes the FP better than it's successor is that it provides a greater dynamic range of numbers to the designer, this is achieved at the expense of more cost.

The IEEE-754-2008 standard allows the processor to the alliancing of the FP arithmetic number system with the FFT architecture. In this paper, a discussion is made about the power dissipation factor which is one of the three requirements for the current design technology to be fulfilled by the designer. The power dissipation factor is discussed in the first section of the paper. It shows mathematically about the importance of the capacitor for any system designing process. The average power is the power the designer should keep in mind during the designing process. The value of the capacitor should be selected appropriately to get the right average power of the system. The importance of the number system representation for the FFT butterfly is also discussed in the fourth section. The section deals with the various number representation for the system i.e. signed digit number system, redundant binary signed digit representation and eventually to the proposed hybrid signed digit representation.

The paper discussed how HSD representation shows the optimum result for designing. This optimum result is considered after reviewing the impact each number of representation gives in terms of the area, cost, performance and also power. In the later section, we discussed the fixed and floating-point number system their impact of the system accuracy and the dynamic range they provide to the FFT architecture. The later part of this paper describes the conclusive result about the paper and as discuss the future scope of the paper. The simulated results have the RTL level designing of the fixed point and floating-point number system along with the analysis table which is a comparative analysis between the traditional FP adder and the proposed HSD adder. The analysis table clearly shows how the improvement on the structural level is benefitted by using the HSD adder in the system. A Barrel shift which a shifter that can shift up to a desired number of point is also used in this design which is the essential feature of this proposed FP adder using HSD representation.

II. Literature Survey

The digital Signalling application earlier uses fixed-point number system because of the higher cost of the Floating-point number system had [8]. The floating-point arithmetic is uniquely useful by the designer in implementing the DSP application-oriented algorithm as it allows the designer to concentrate only on the architecture and algorithm involved without worrying about the numerical issues [1, 6] raised. The DSP application mainly uses floating-point architecture[13] to perform the real-time oriented operations, using this they can overcome the limitation of the fixed-point[8].

algorithm as it allows the designer to concentrate only on the architecture and algorithm involved without worrying about the numerical issues [1, 6] raised. The DSP application mainly uses floating-point architecture[13] to perform the real-time oriented operations, using this they can overcome the limitation of the fixed-point[8].

The FFT is implemented to perform the complex number multiplication thus consideration is given to transform as it includes the various operation in performing the single butterfly namely multiplication, addition and subtraction at the same time[7,11]. The FFT algorithm is very useful transform it is made to compute the discrete Fourier transform(DFT). A Fourier transform is a tool for computing a signal that is in the frequency domain which is converted from the time domain. The discrete Fourier transform breaks this frequency signal into different frequencies. But the processing time of DFT is quite slow thus FFT is very helping in those cases. An FFT is rapid processing transform it can reduce the DFT computation from O(N2) to O(N log N) where N is the data size. This difference of the computational speed is very useful for processing long length of data. The FFT algorithm is used for evaluating the theories related to complex numbered arithmetic's, group theory to number system algorithms. There are various types of FFT algorithm namely Cooley-Turkey algorithm, the Rader's algorithm, Winograd FFT algorithm and the Bruun's algorithm.

In this paper consideration is given for the Radix 2 FFT algorithm, the butterfly is evaluated on the same basis

I. POWER DISSIPATION EFFECT OF THE SYSTEM

Power dissipation is the main design issue, this is due to the ever-growing demand for portability in the electronic devices. A low-power oriented design will require optimization at each level of the abstraction it undergoes. The dynamic-power that is consumed by the devices is due to the charging and discharging electronic component(i.e. of capacitance). The energy which is consumed for N clock number of the cycles is

$$E_i = n(N)C\,V_{DD}^2 \qquad (1)$$

here n(N) will be the total number of binary transitions in N clock cycles, C is the capacitance value of the switching capacitor and VDD will be the voltage supply. Switching power is given as energy per transition and can be expressed as

$$\boldsymbol{P_{avg} = \lim_{N\to\infty} \frac{E_N}{N} . f} \qquad (2)$$

$$\boldsymbol{= \lim_{N\to\infty} \frac{E_N}{N} . f . C . V_{DD}^2} \qquad (3)$$

Here f is the clock frequency and $\lim_{N\to\infty} \frac{E_N}{N}$ is denoting the signal line. Replacing the $\lim_{N\to\infty} \frac{E_N}{N}$ to $\alpha_{0\to1}$ The switching power dissipation in a circuit is then given as:

$$\boldsymbol{P_{avg} = f . C . V_{DD}^2\, \alpha_{0\to1}} \qquad (4)$$

The principles of the power reductions: reduction in the signal switching activity and reduction in the area requirement.

II. VARIOUS NUMBER SYSTEM

A. Signed Digit Number System

The binary signed-digit number representation doesn't have proper redundancy for the carry-free algorithm implementation. So to overcome this a greater than ½r radix base is required[11]. The BSD number representation can represent the intermediate sum and carry in two's complement form. This tow's complement form representation will lead to high-speed multiplication which is the basic principle for booth multiplier algorithm. An SD numbers representation defined as:

Assuming the r as radix base, and SD can represent each of the digit by assuming r to $2\alpha + 1$ and having value from $\varepsilon_r = \{-\alpha, -\alpha + 1, \dots \dots \dots . -1, 0, 1, \dots \dots \dots . -\alpha - 1, \alpha\}$. Here α is in the range of $|\frac{(r-1)}{2} \le \alpha \le (r-1)|$ and r ≥ 2.

The algebraic value of the SD representation can be summed as U = u_{n-1}, u_{n-2}, u_{n-3},u_{n-4},…..u_0,u_1,……..u_{-m} with radix r is given as:

$$U_a = \sum_{i=m}^{n-1} u * r^i \qquad (5)$$

The originally the SD arithmetic number system uses a symmetric number set after setting the radix base r as 2.

B. Redundant Signed Digit Number System

The RSD is a superset of SD number representation. RSD number representation was first proposed by Avizienis in 1961 [1]. RBSD number represents a number from the digit set of (1, 0,-1), while the binary unsigned number system, that represented by the digit set of (0, 1)[12]. Decimal values of the RSD is evaluated by the following formula.

$$A \ = \sum_{k=0}^{n-1} \binom{n}{k} x_k \ \ 2^k \qquad (6)$$

The number (4)10 representation in binary is 0100 but, (4)10 representation in RSD by using the (5) can be more than one i.e. $1\bar{1}00$ or $11\bar{1}\bar{1}0$ 0. Hence more than one representation of the same number coined the term redundant representation. The RBSD addition for carry propagation free algorithm is performed in two steps [5] as follow:

1). The elimination of the carry for each digit position will produce the transfer digit t_i and the interim sum digit w_i is calculated according to Table 1. Assuming Ai and Bi are the two operands then the relationship between A_i, B_i, t_i and w_i is mathematically represented as

$$A_i + B_i = 2t_i + w_i \qquad (7)$$

2). The transfer digit is added with the interim sum and final sum digit is obtained using the relation $s_i = w_i + t_i$.

Table 1: Obtained for the evaluation of the transfter digit t_i and w_i[1].

A_i	B_i	A_i+B_i	t_i	w_i
-1	-1	-2	-1	0
-1	0	-1	-1	1
0	-1	-1	0	-1
-1	1	0	0	0
1	-1	0	0	0
0	0	0	0	0
0	1	1	0	1

C. Hybrid Number System

In place of having every digit to be signed digit, the HSD has alternative digits as signed and unsigned bits. An HSD representation will limit the maximum length of the carry propagation to the user-specified length (i.e. the maximum length of the carry propagation chain should be equal to (d + l), where d is the longest distance between neighboring signed digits)[1][9].

The signed digit should be the most significant digit in the HSD representation so that it can include enough negative numbers for representation and the remaining digits can be unsigned[9]. Taking the example, if the word length is of 32 digits, then the 32nd digit which is the most significant digit should be the signed digit keeping the remaining digits are at the designer's disposal to use. When proper regulation is not so important the designer can make 1st, 2nd, 4th, and 8th digit a signed digit and keeping the remaining as the unsigned. The longest possible length of the carry–propagation chain between the consecutive signed digits will determine the addition time of the system.

III. Proposed Architecture Of HSD FP Adder

A. Fixed Point Number System

Fixed-point number system which is the subset of the floating-point number system[13]. The Fixed point represents the fractional values in the base 2 or base 10 form[14]. The fixed-point data type has a value which is integer type and scaled by an implicit factor. This can be understood by considering, the value 1.23 is represented as 1230 in the fixed-point, having a scaling factor of 1/1000. Also, the value 1,230,000 be represented as 1230 having the scaling factor of 1000. In fixed-point arithmetic, the scaling factor will be the same for all the values of the same type and this doesn't change while the computation period.In a fixed point, the scaling factor is mainly of the power of 10 which is convenient for human and may also alternative scaling factor of power 2 which is convenient for the computation processing[13]. It can represent it's the largest value is the key integer type which should be multiplied by the assumed scaling factor and so forth the minimum value can also be calculated.

B. Floating Point Number System

Floating-Point arithmetic is the IEEE Standard for (IEEE-754), which is the most pervasive standard for the floating-point computations. The various microprocessors, (like Intel-based Processor's and UNIX platforms) uses this standard as the prime representation in their processor. The floating-point numbers system mainly have the three basic components: a sign, a significant and an exponent. The significant constitute of two basic digits the fractional digit and the leading implicit digit. The Floating-point have the two standard representation which is the Half Precision having 16-bit representation and the Single Precision having

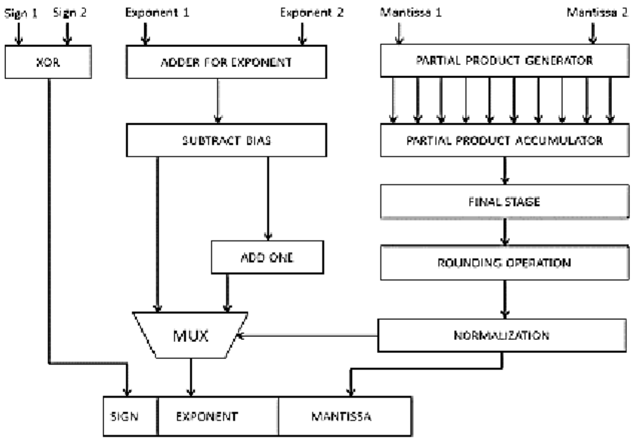

Figure 1. Floating point multiplier circuit for two number

32-bit representation. FP has the advantage of over fixed-point is that it provide a wide dynamic range to the number representation. But, this is enjoyed on the cost of higher cost involvement.

The floating-point multiplier architecture has two main constituents namely Partial product generator and partial product reduction. The working of the floating-point multiplier is explained as below:

1) Partial Product Generation: the multiplications in FFT for the significant digit is computed by the series of shifters and adders connected in a cascaded manner. The intention of this scenario is to reduce the number of adders used and to store the significand used in the Booth encoding.
2) Partial Product Reduction: The main constituents of this is to use the HSD adder to limit the carry which is lower compared to the BSD adder.

The output of the floating-point multiplier is sent to the floating-point adder unit for further processing.

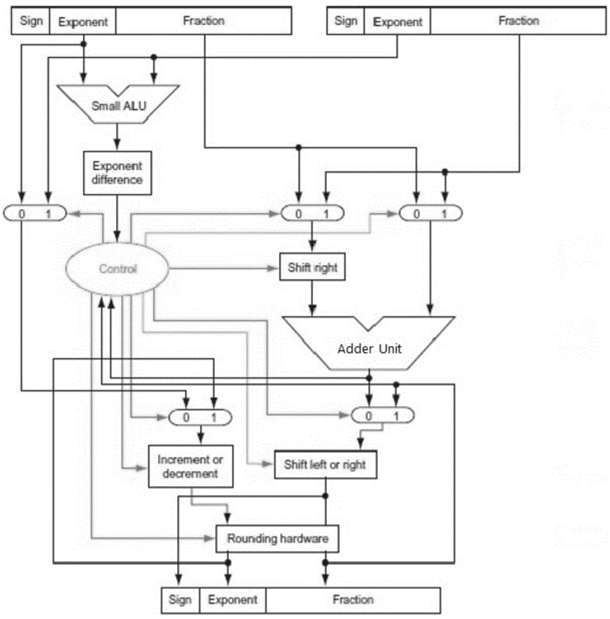

Figure 2. Traditional Floating point adder unit.

Here we see a sign computational unit, this unit will be exempt if we use the proposed floating-point adder unit as the HSD floating-point FFT unit will have both signed and unsigned digit alternatively placed thus there is no need to find the sign of the digits we are adding. Along with the use of the barrel shifter in place of the regular right shifter circuit, this is done because the regular shifter circuit will shift right on providing a single clock input will perform a single right shift. While the barrel shifter on providing a singe clock can provide the desired (more than one positioned) shift. The barrel shifter is costlier but provides a faster shifting which is very helpful the designer. The proposed floating point-adder unit will have the lower computational timing of the circuit and it shows the optimum result needed for the designer to achieve the technological goal required[14]. The proposed floating-point adder, when combined with the fused dot product unit and using the three operand input, will show a better result for designing of the FFT Butterfly unit. The rounding hardware used here is the same as that of the previous floating-point architecture.

The FP multplier which is the prior stage of the FP adder unit will be same as the previous used stages. The output of the FP multiplier will be feed to the proposed FP adder unit. Th obatined output of the FP adder unit will be in the HSD format. The HSD output can be changed into the desired binary ouput by first changing the output to the RBSD form and then converting back to the binary digit representation.It should be noted that the output can be converted back into the binary form but it will be much more regress compared to converting it from binary to HSD.

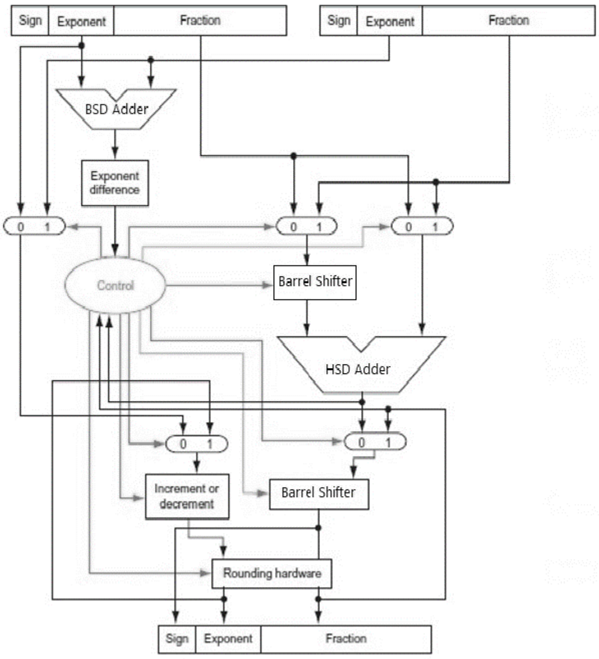

Figure 3. Proposed floating point adder unit.

VI. Results and Discussion

The analysis is done considering the computational time delivered by both the fixed point adder unit and the floating point adder unit. The figure 4 shows the RTL level representation of the the shift-and-add multiplier unit. The shift-and-add multiplier is almost similar to the primitive multiplication process. The basic idea behind this is that we need to add the first multiplicand to itself up to the second multiplicand time's(i.e. add A to itself up to B times). The shift add multiplier will takes the digit from left to right one at a time and generating the intermediate products after multiplying the digit one at a time.

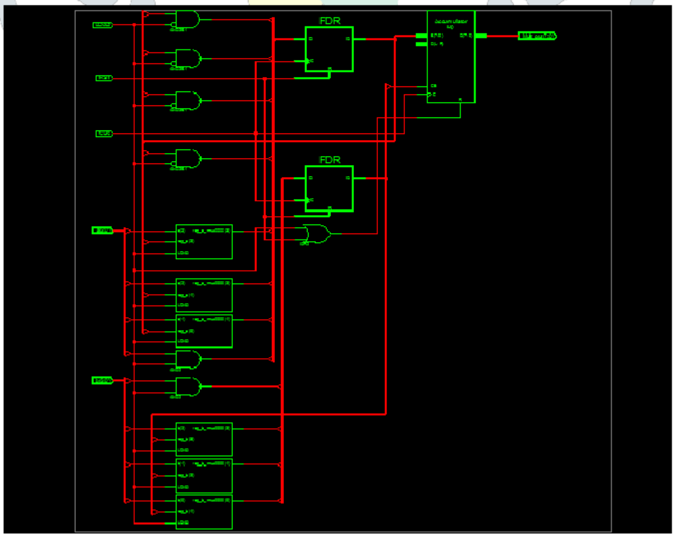

Figure 4. RTL representation of the Shift add-multiplier unit.

The figure 5 illustrate about the proposed floating point HSD represented adder unit. In this we can see the complexity it shows compared to the normal shift add multiplier. Th shift add multiplier can be considered as the primitive case of the fused dot product case. The figure shows more area requirement compared to the primitive one.

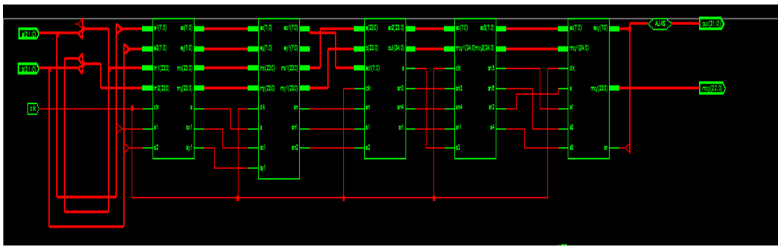

Figure 5. RTL representation of the Floating point adder unit.

Table 2. Comparative analysis on the basis of structural requirement and delay produce is done for the conventional FP adder and the proposed FP adder.

Criterion	Conventional	Proposed
Sign Addition	CSA+CPA	BSD+HSD
Critical Delay Path	Subtractor+MUX+Shifter +CSA+CPA	Subtractor+MUX+Barrel shifter+BSD+HSD
Sign Logic	Yes	No
Delay	2.99ns	2.92ns

From the table 2 it is observed that the structural requirement of proposed FP adder is 5.06 % less than that of the conventional FP adder unit this is achieved due to the fact that there is no CSA involved in the circuit which requires more area compared to the HSD adder used. Delay of the proposed FP adder, is less than that of the conventional FP adder unit because of the involvement of the Barrel shifter in the circuit which will provide the greater computational speed this the desired improvement required for the designer.

VII. CONCLUSION

We proposed a high-speed FP butterfly architecture, which is faster than the fixed point butterfly architecture but at the cost of the higher area. The reason for this speed improvement is in twofold: 1) HSD adder which replaces the BSD adder enjoys the benefits of signed and unsigned digit due to the signed digit the HSD adder make a carry-free propagation which results in faster addition 2) The removal of the barrel shifter in the circuits provides a faster shifting in a single clock pulse. The future scope of this work is using the three operands fused floating-point add subtract unit which provides a faster computation on the lower area.

In the paper, the emphasis is given to the correct representation of the number system to produce a faster arithmetic operation further research can also be done in combing a CSA(carry-save adder) with the HSD number representation which will provide the faster computational time along with a proper carry-save mechanism.

REFERNCE

[1]. 0. L. Mac Sorley, "High-speed arithmetic in binary computers," Proc. IRE, vol. 49, pp. 67-91, Jan. 1961

[2]. A. D. Booth, "A signed binary multiplication technique," Quarterly J. Mech. Appl. Math., vol. 4, part 2, pp. 236-240, 1951.

[3]. Avizienis, A., "Signed digit number representation for fast parallel arithmetic", IRE Trans. Electron Computer, vol EC-10, pp. 389-400, sept.1961

[4]. K. D. Tocher, "Technique for multiplication and division for automatic binary computers," Quarterly J. Mech. Appl. Math., vol. 11, part 3, pp. 364-384, 1958

IEEE J. Solid-State Circuits, vol. 31, pp. 773-783, 1996.

[5].Behrooz Pamami, "Generalized Signed-Digit Number Systems: A Unifying Framework for Redundant Number Representations", IEEE Transactions On Computers, Vol. 39, No. 1. January 1990.

[6]. IEEE Standard for Floating-Point Arithmetic, IEEE Standard 754-2008, Aug. 2008, pp. 1–58.

[7]. 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.

[8]. P. Kornerup, "Correcting the normalization shift of redundant binary representations," IEEE Trans. Computer, vol. 58, no. 10, pp. 1435–1439, Oct. 2009.

[9]. Vishal Awasthi and Krishna Raj, "Development of Optimum Addition Algorithm using Modified Parallel Hybrid Signed digit (MPHSD) Technique".2013 3rd IEEE International Advance Computing Conference.

[10]. V Awasthi, K Raj "Application of Hardware Efficient CIC Compensation Filter in Narrow Band Filtering", World Academy of Science, Engineering and Technology International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering Vol:8, No:9, 2014.

[11]. Krishna Raj & Suman Lata, “Fast Processing Using Signed Digit Number System”, India International Journal of Electronics Engineering, 2(1), 2010, pp. 173-175.

[12]. K Raj, B Kumar, P Mittal, “6. FPGA Implementation and Mask Level CMOS Layout Design of Redundant Binary Signed Digit Comparator”, IJCSNS International Journal of Computer Science 2009.

[13]. Rajkumar Tomar, Praveen Kumar Singh, Krishna Raj, “A Review of CORDIC Algorithms and Architectures with Applications for Efficient Designing”, International Journal of Scientific & Engineering Research, Volume 4, Issue 8, AUGUST 2013 ISSN 2229-5518.

[14]. K. Raj, P. K. Singh, and R. Tomar, "A review of low-cost multiplier using CORDIC subsystem," 2012 2nd International Conference on Power, Control and Embedded Systems, Allahabad, 2012, pp. 1-5.