International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 02 | Feb 2020 www.irjet.net p-ISSN: 2395-0072

DISTRIBUTED ARITHMETIC METHOD FOR COMPLEX MULTIPLICATION

C.V. Thejashwini[1], Dr. A. Sumathi[2]

[1]*P.G Student, Department of Electronics and communication Engineering, Adhiyamaan College of Engineering, Tamilnadu, India*
[2]*Professor, Department of Electronics and communication Engineering, Adhiyamaan College of Engineering, Tamilnadu, India*

Abstract –*In today's technology, delay, power and area are the crucial parameters to outline any kind of the algorithm on FPGA. FFT plays a vital role in acquiring the signal characteristics with the least use of carrying out parameters. Speed multipliers are fundamental parts of DSP systems. Multipliers are complex process and consumes more time. In order to lower the complexity multiplication, various multiplier less method are introduced. We planned an efficient DA based complex multiplier inplace of regular multiplier. Coding is done using Verilog language. Using Xilinx 14.5i tool with Spartan 6 kit, Simulation can be achieved.*

***Key Words*: FFT, DA algorithm, SDF, complex multiplier.**

1. INTRODUCTION

Currently, FFT has got the great prominence in the fields of biomedical applications and communication to analyze the signal features. The conversion of original domain in to a frequency domain signal called FFT. To make full use of hardware resources, algorithm of FFT is exposed. This FFT algorithm plays a important role in the communication field.

Arithmetic operations with complex numbers are required in many DSP algorithms, e.g. FFT. The complex multiplication operation is upscale and the most superior factor in computing the speed. If both the inputs are variables, better to use regular multiplier architecture. In case, one of the input is constant i.e. twiddle factors, so we use DA based multiplication. This can be proven that it is faster than conventional multiplier. Figure 1 shows the basic multiplication technique. Each black dot be regarded as a single digit.

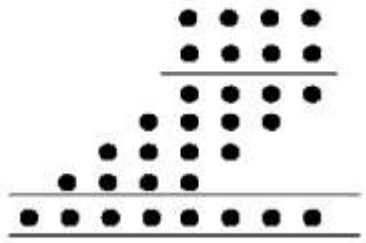

Fig -1: Basic multiplication

1.1 RADIX-2 FFT ALGORITHM

Cooley and Tukey presented the first FFT algorithm, which is Radix-2 algorithm. The DFT of a given resultant X[n] can be estimated using the formula.

$$X(k)=\sum_{n=0}^{N-1}x(n)W_N^{nk} \qquad (1)$$

Where W_n is twiddle factor.

$$X_1(r)=X(2_m) \qquad (2)$$

$$X_2(r)=X(2_m+1)$$

Then, the N-point DFT becomes

$$X(K)=\sum_{n(even)=0}^{N-1}x(n)W_N^{nk}+\sum_{n(odd)=0}^{N-1}x(n)W_N^{nk}$$

$$=\sum_{m=0}^{(N/2)-1}x1(2m)W_N^{nk}+W_N^{nk}\sum_{m=0}^{(N/2)-1}x2(2m+1)W_N^{nk}$$

Properties of twiddle factors are used. They are symmetry and periodicity.

$$X[k]=X_1(k)+W_N^kX_2(k)$$

$$X[k+N/2]=X_1(k).W_N^kX_2(k) \qquad (3)$$

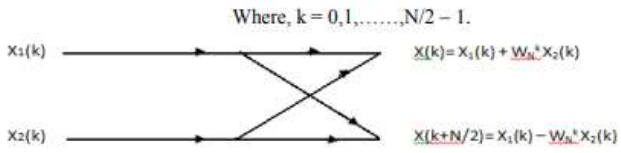

Fig -2: Basic butterfly unit.

The above figure 2 shows the basic butterfly structure.

1.2 RADIX-2 SDF

SDF FFT architecture has the most efficient utilization memory for pipelined FFT processor. The first half data input is saved in memory. So, the delayed input processed

with the remaining input in butterfly unit. The output data fed back to input of butterfly unit through buffer for further processing.

SDF has a great advantage that it requires less memory space. Especially for applications like low power design, this SDF offers several advantage. This is the reason SDF is adopted. SDF block reduces the complex adder by 50% and also produce the output in normal order. The utilization of multiplier remains 50%. The SDF block is used to share delay elements between butterfly inputs and outputs to improve efficiency of hardware. The figure 3 describes the SDF architecture.

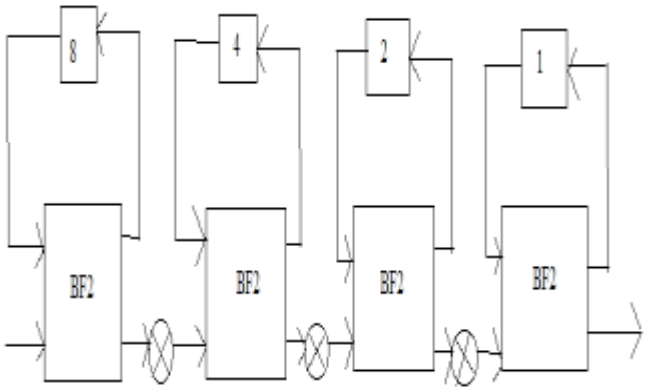

Fig -3: Radix-2 SDF architecture

2. RELATED WORK

Kavitha. M .V et.al [1] Radix-8 booth multiplier will be implemented as a complex multiplier for inplace FFT architecture. This multiplier operates in parallel and requires less adders. Using this multiplier, power is reduced and time requirement also very less. To diminish power dissipation and area, CSLA is used. Overall delay, area and power consumption will be minimized in the proposed method with the support of CSLA adder.

Laguri, Nisha et.al [2] has introduced an efficient split-radix FFT based on DA. To reduce the multipliers, DA is used. With the Radix-split FFT, few number of additions and multiplications were achieved but it introduces the overflow issue.

Anumol B .Chennattucherry et.al [3] in this paper stated that FIR Filter developed with DA is phenomenal of all regular multipliers. The multiply and accumulate operation is translated in to shift and add operation When this Distributed Arithmetic algorithm is directly applied to realize FIR Filter. Due to this, speed is optimized. Pipeline architecture can achieve a low latency and a high throughput which are suitable for real time applications. In pipeline FFT architecture, we made use of SDC. This SDF pipeline FFT architecture is exceptional because it requires less memory space. Especially for the application like DSP or low power design, SDF has a great advantage. SDF is selected due to this reasons.

P.Sritha et.al [4] this paper has presented several DA based filter techniques. But modified DA technique has requires only lower area. Various DA technique proven that these techniques were effective in reduction of area requirement.

3. COMPLEX MULTIPLIER

The complex multiplication is a very expensive operation, to minimize the multiplicative complexity of the twiddle factor inside of the butterfly unit by calculating the only real multiplications, additions, and subtractions.

The twiddle factor multiplication,

$$A+ j B = (Y+ j Z) (C+ j S) \quad (1)$$

The complex multiplications can be simplified as

$$A=(C-S) Y+X \quad (2)$$

$$B=(C+S) Z-X \quad (3)$$

Where X=C(Y-Z). S, C are preempted and stored in a memory. C+S, C-S and C are the three coefficients. Figure 4 shows the signal flow graph.

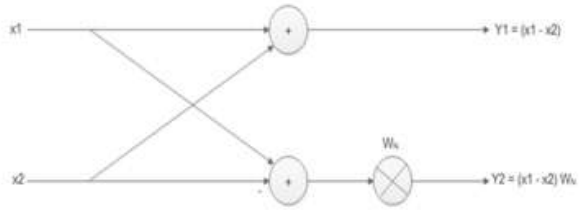

Fig -4: Signal flow graph

A complex multiplier consists of 2 regular adder and 4 regular multiplier. Instead of using the actual multiplier and adder blocks, we introduce a multiplier and adder less complex multiplier using the concept of Distributed Arithmetic (DA). Figure 5 represents the complex multiplier which is shown below.

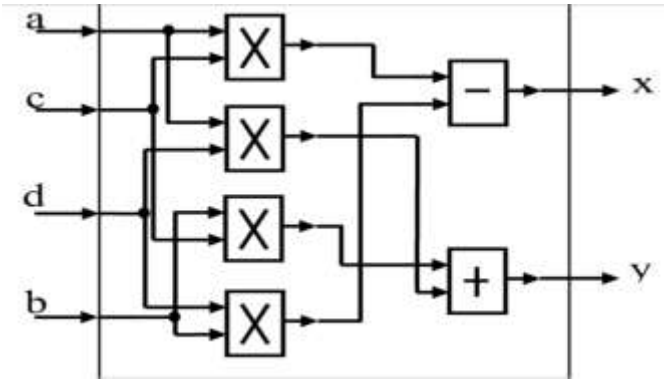

Fig -5: Complex Multiplier

4. DISTRIBUTES ARITHMETIC ALGORITHM

Multipliers are complex and tedious process. In order to reduce the complexity in multiplication process, several multiplier less method were evaluated. This has two types. One is methods based on conversion and other is methods based on memory. DA techniques are based on memory methods.

The method for DA based complex multiplication can be summarized as,

$Z_R+j\ Z_I= (B_R+j\ B_I)*(T_R+j\ T_I)$ (1)

Where $Z_R=B_RT_R-B_IT_I$

$Z_I=B_RT_I+B_IT_R$

It shows that 4 real multiplication and 2 two addition are required to compute ZR and ZI. But these equations can be considered as multiply and accumulate operation.

$$y=\sum_{k=1}^{k} C_k X_k$$

Let, C_k are fixed coefficients and X_K are the input words. If X_K is M-bit fractional number in 2‟s complement form then it can be expressed in following form

$$X_k=-b_{k0}\sum_{m=1}^{M-1} b_{k0}2^{-m} \quad (2)$$

It is a trending architecture in recent years due to its high performance.

5. PROPOSED ARCHITECTURE

In recent years, DA is a trending architecture. The multiplier is not necessary, instead it is implemented based on LUT. The distributed arithmetic based complex multiplier is presented.

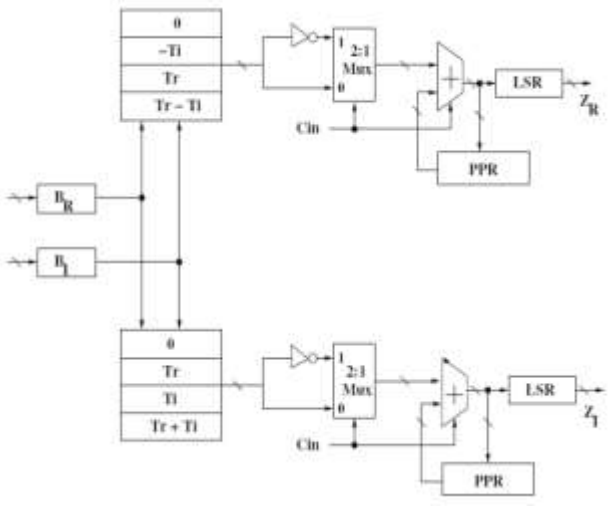

Fig -6: DA based complex multiplier

The real part and imaginary parts of incoming words BR and BI are stored in two 8 bits wide “parallel in serial out “register. Shifting is carried out starting from LSB to MSB. Each output bit of two registers is used as address lines of the ROM. The ROM stores pre calculated outcomes for both ZR and ZI. The size of each ROM is 4x8. The input to the 2:1 MUX is directly fed from the output of ROM and other input to MUX is inverted. Input and output bit width for MUX is also 8 bits. The selection line of MUX is, signal Cin =0, till the MSB arrives at output. It Cin=1, it selects the inverted output from ROM and it is added to the value saved in the PPR which also performs 1-bit right shift operation. Finally, the output is fetched from left shift register.

6. RESULTS AND DISCUSSIONS

Efficient DA based complex multiplier architecture is implemented in Verilog, simulated using Xilinx ISE 14.5i. The RTL defines the digital portions of the design. It actually a representation of translation of our logic to a digital circuit. RTL design for Distributed Arithmetic based complex multiplier is shown in figure 7, 8. Technology schematic shows the representation of our design in terms of LUTs, buffers, I/O’s and carry logic, which is given in figure 9. Figure 10, 11 and 12 shows the simulation output for DA based complex multiplier (signed, unsigned and hexadecimal)

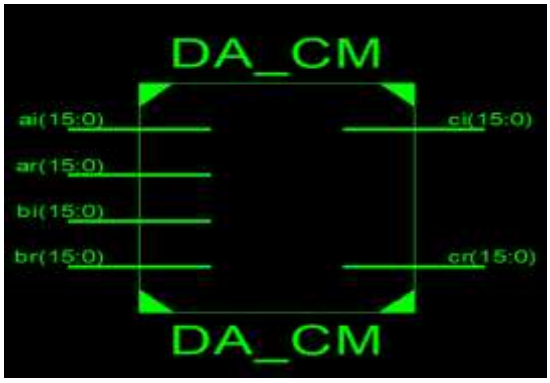

Fig -7: RTL schematic view for DA based complex multiplier

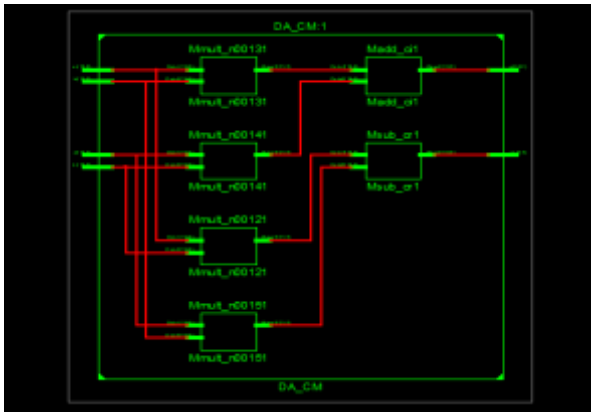

Fig -8 : Internal block of RTL design

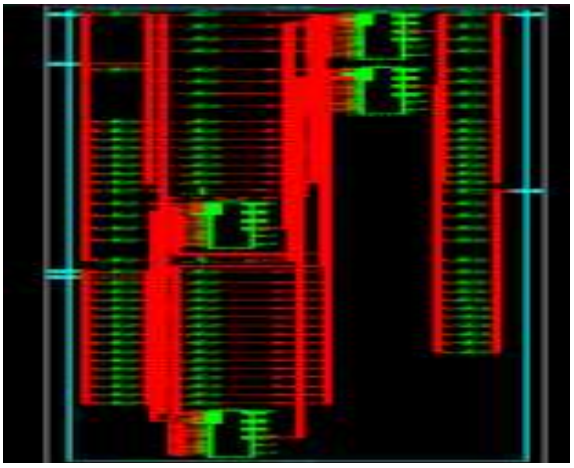

Fig -9: Technology schematic view for DA based complex multiplier

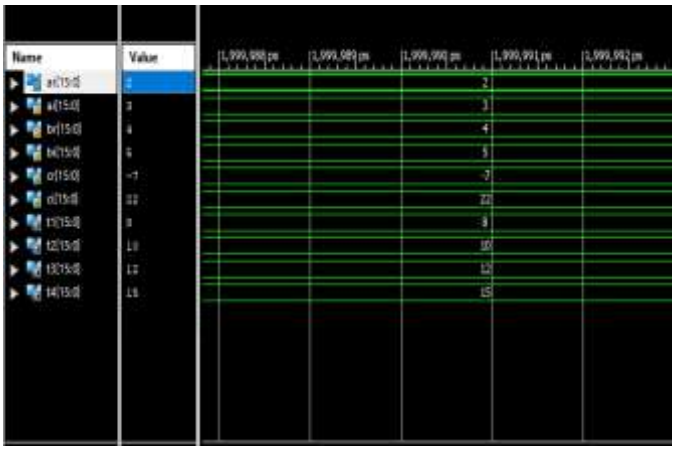

Fig -10: Simulation output for DA based complex multiplier(signed number)

Fig -11: Simulation output for DA based complex multiplier(unsigned number)

Fig -12: Simulation output for DA based complex multiplier(hexadecimal)

Table 1 shows the performance comparison of multipliers. . It is clear from the Table I that the LUTs is less than regular mutiplier. The total number of LUTs and delay is reduced.

Table -I: Comparison of Multipliers

MULTIPLIER TYPE	NO.OF.LUTs	DELAY(ns)
WALLACE MULTIPLIER	43	15.195
BOOTH MULTIPLIER	35	12.783
PROPOSED DISTRIBUTED ARITHMETIC BASED COMPLEX MULTIPLIER	26	11.249

7. CONCLUSION

In this paper, distributed arithmetic based complex multiplier has been proposed. Parameters like area and delay are considered for performance evaluation needed by the SDF FFT. DA based complex multiplier has been used as a complex multiplier for proposed model. The replacement of booth multiplication, Wallace multiplier with DA based complex multiplier offers great advantage. This shows that our proposed system reduces the overall area and delay than conventional multipliers. The proposed architecture can also be adapted to high point FFT applications with lower size off twiddle factor ROM's. DA based multipliers are used in FIR filters.

REFERENCES

[1] S,Ranjitha, S.G.Hiremath, Kavitha M.V, Suresh H.N,"A Novel RTL Architecture for FPGA Implementation of 32-point FFT for High Speed Applications" IOSR-JCE,ISSN:2278-0661, Volume 19, Issue 2, Ver.II(Mar 2017).

[2] Laguri, visha and K,Bahnusudha, "DA based VLSI radix-split FFT "ICGCCEE, International Conference, pp:1-5,IEEE,2014.

[3] Anumol B. Chennattucherry, Diego James" A Novel Approach to lessen power and area for FFT Implemention", IJAREEIE, ISSN: 2278-8875. Vol. 2, Special issue 1, Dec 2013.

[4] P. Srinivas, M.Suhasini "performance estimation and design of multiplier with DA algorithm", ISSN: 2231-0371, Vol.3, Issue- 4, 2012.