Sindh Univ. Res. Jour. (Sci. Ser.) Vol.46 (2): (2014)

SINDH UNIVERSITY RESEARCH JOURNAL (SCIENCE SERIES)

Comparative Analysis of Alternatives of Rotators in Single Path Delay Feedback FFT architecture

I. A. QURESHI++, F. QURESHI*

School of Information and Electronics, Beijing Institute of Technology, Beijing, P.R.China

Received 6th May 2014 and Revised 5th June 2014

Abstract: Rotators are an essential part of fast Fourier transform (FFT) butterfly architecture. It defines the overall complexity of the FFT computation between the stages in the signal flow graph of the single path delay feedback FFT architecture. Rotations are basically multiplication by a complex number having magnitude equal to one, therefore only the phase of the multiplicand is changed. In 16 point radix 2 Single path delay feedback (SDF) FFT the number of stages is four and every stage the rotations are involved. Increasing the radix to 4, reduces the number of stages and consequently the rotation between stages is also reduced but it increases the complexity of the architecture. Different realizations of rotations exist like general complex multiplier, Coordinate rotation digital computer (CORDIC) and the constant multiplication. These techniques use shift and add based methods to realize the rotations. This paper compares the different alternatives of rotations for different angles in terms of the number of adders as a function of bits.

Keywords: Single path delay feedback(SDF), Fast Fourier transform(FFT),Coordinate rotation digital computer (CORDIC)

1. INTRODUCTION

A hardware architecture of the FFT (Cooley *et al.*, 1965)consists of a set of butterflies, rotators and circuits for data management as well as certain control circuitry (Garrido *et al.*, 2011) (Qureshi *et al.*, 2010). Butterflies and rotators calculate the mathematical operations of the flow graph. Each butterfly of the architecture usually computes several butterflies of the algorithm and the same occurs for rotators. This reuse of components allows reduction in the area of the circuit. Circuits for hardware management usually consist of memories or buffers, which are also used for data storage, as well as multiplexers and other hardware components. (Garrido *et al.*, 2011) They ensure that mathematical operations are carried out in the right order and at the right time.Finally the data management is the main difficulty in designing a FFT architecture. It requires not only to decide which butterfly and rotator of the architecture performs each butterfly and rotation operation within the flow graph but also to interconnect them so that there are no conflicts(Garrido *et al.*, 2011) .

2. MATERIALS AND METHODS

SDF FFT ARCHITECTURE

The single-path delay feedback architecture has a feedback loop at every stage) (Qureshi *et al.*, 2010). The butterfly processing element stores the input samples in feedback memory, until they are required for butterfly computation. These butterflies have 50% utilization ratio. This architecture has one continuous data stream of one sample per clock cycle, which means that at every clock cycle one sample is fed and one sample is received as output. However, higher throughput can be achieved by increasing the clock frequency.

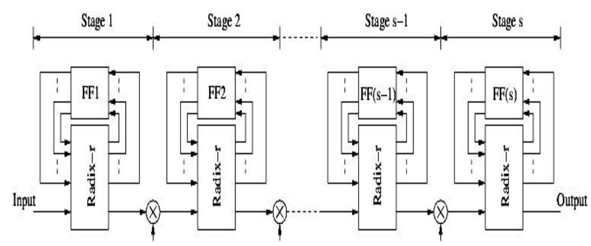

Figure 1 Radix-r SDF FFT architecture

Fig.1 shows the radix-*r* butterfly SDF pipelined FFT architecture. In this architecture, the number of stages is $s = \log_r(N)$ and each stage contains a radix-*r* butterfly, a rotator and memory or buffers. The radix-*r* butterfly comprises of an adder, and multiplexers. However, in case of radix higher than 4, rotators are also required for computation of non-trivial rotations. The memory is also used in this architecture for storing the incoming samples Furthermore, multiplexers are used to control the inputs of the memory whether fed directly without any computation or after the butterfly computation is performed(Garrido *et al.*, 2011).

These multiplexers are usually controlled by a clock signal. In case of radix-2, multiplexers of first stage switch their position after $N/2$ clock cycles and

++Corresponding Author:- i.ali1225@yahoo.com

*Department of Electrical Engineering, Linkoping University, Linkoping, Sweden

every *N*/4 clock cycles in the consequent stages. Likewise, the memory requirement is only $N - 1$ because it requires *N* cycles to provide first output. There are two types of memory or buffer requirements in the SDF FFT architecture.The first one is the memory for saving the input samples till they are required for computation.The other memory is the twiddle factor coefficient memory which are precomputed and stored in memory and fed to the multiplier when required.The coefficients are multiplied using rotations which can be implemented using different architectures(Garrido *et al.,* 2011). In general, a single-path delay feedback architecture for FFT is considered as an optimal choice in terms of the hardware cost and performance for many applications.

ROTATIONS

A rotation is a mathematical operation where multiplication by a complex number having magnitude equal to one is carried out .This means that this operation affects the phase of the data only (Qureshi *et al.*, 2010). This operation is an essential part in many signal processing algorithms, including digital filters and transforms like fast Fourier transform (FFT) (Qureshi *et al.,* 2010). The rotation is often the most expensive arithmetic operation and one of the dominating factors in determining the performances in terms of accuracy, speed and power consumption. Therefore, the number of rotators and their implementation is always an important issue in signal processing algorithms.

The rotation of a complex number $x + iy$ by an angle γ is given as

$$W = x.\cos\gamma - y.\sin\gamma$$
$$Z = y.\cos\gamma + x.\sin\gamma \quad (1)$$

Thus the rotation can be written as,

$$W + iZ = (x.\cos\gamma - y.\sin\gamma) + i(y.\cos\gamma + x.\sin\gamma) \quad (2)$$

The rotation in equation 1 is shown in figure 1.As rotators rotate complex data they usually comprise of four real multipliers and several adders. Other approaches only use three real multipliers, general complex multiplier or the CORDIC algorithm to compute the rotations.

GENERAL COMPLEX MULTIPLIER

The most common way of implementing the rotation is by using four multipliers and two adders. In complex multiplier, the sine and cosine of the rotation angle are stored in memory. The architecture of the multiplier has a multiplier symbol that represents the multiplication and addition of the inputs and the memory is used to store the sine and cosine component of the rotation angles(Qureshi et al .,2010). In pipelined FFT architecture, each stage performs *N* number of rotations, so to store *N* number of twiddle factor coefficients in the memory is required. In order to reduce the size of the coefficient memory, there are a number of approaches, which have been applied .The coefficients in memory are the input to the complex

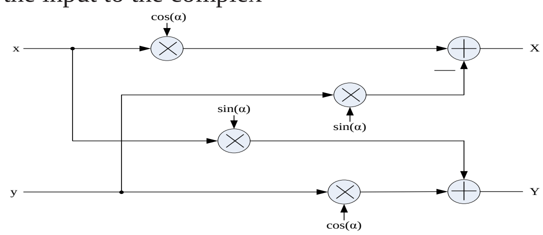

Fig. 2 Operational diagram of rotation

multiplier in order to calculate the rotation of the input sample. Furthermore, the number of 0 → 1 bit transitions between successive coefficients that are read from the coefficient memory is defined as switching activity (SA), which is related to the power consumption of the circuit (Qureshi *et al* ., 2011).

The complex multiplier consists of four real multiplications and two additions. Regardless of the application of rotation, the complex multiplier can be realized by different approaches which reduce the number of real multiplications from four to three at additional cost.

CORDIC

CORDIC stands for (Coordinate Rotation Digital Computer)(Jack *et al.,* 1959) (Lee *et al.,* 1992).It is well known algorithm for the implementing multiplier-less rotations (Garrido *et al.,* 2011). It reduces the amount of hardware by realizing rotation using a number of shifts and additions. It is also suitable where multipliers are not available. However, it may affect the accuracy since it is based on approximation. The CORDIC algorithm decomposes the angle θ, into a sum of angles α_i as

$$\theta = \sum_{i=0}^{M-1} \delta_i \alpha_i + \varepsilon \quad (3)$$

where θ is the angle that has to be rotated, ε is the error approximation, δ_i indicates the micro-rotation as shown in figure 4 .The predefined angles α_i is calculated as,

$$\alpha_i = \tan^{-1}(2^{-i}) \quad (4)$$

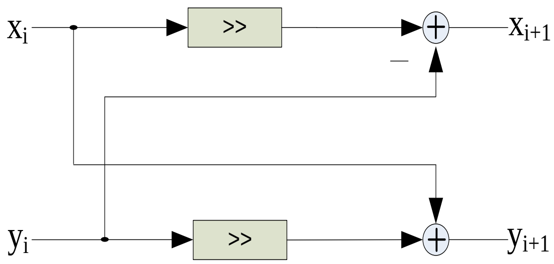

Fig 3. CORDIC Mircro rotation

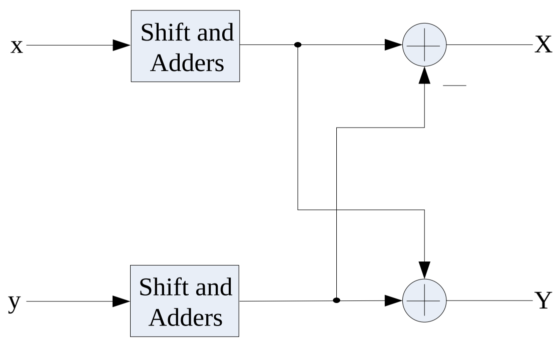

Fig..4 Rotation based on constant multiplication

CONSTANT MULTIPLICATION

Multiplication is an operation where general multiplier can be replaced by adders, subtractors and shifts (Qureshi *et al.*, 2010). As far as complexity is concerned, shift operations are trivial as they only decrease the number of adders in the implementation of the constant multiplications. Multiple constant multiplication (MCM) is constant multiplication algorithm (Dempstar *et al.,* 2005), where the input is multiplied with several constant coefficients. A general block diagram of the rotator is shown in Fig. 5.This consists of two constant multiplication blocks, where two additional adders are used to complete the rotation. Thus constant multiplication could be a good choice to implement the rotations because it can achieve any rotation angle. (Qureshi *et al.,* 2010)(Gustafsson *et al.,* 2006).

3. DISCUSSION

The varying method for computing the rotation part of the FFT computation plays a vital role in finalizing which rotation hardware is useful when used in the algorithm.The general complex multiplier analyses between the different alternatives of rotators for calculating a rotation is studied through approximation.

The best way to see which method is better is through the error calculation calculated in terms of the fractional bits based on the error calculation which is given by

$$N_{CFB} = -\log_2 \epsilon \qquad (5)$$

where ϵ is the error and N_{CFB} is the number of correct fractional bits. Error is introduced because of the quantization of the coefficients in scaled constant multiplication , whereas in CORDIC the error is because of the fact that any finite number of predefined rotations cannot perfectly approximate any angle.

For CORDIC the angles are calculated using the series of micro rotations that best approximates the angle whereas addition aware quantization(Gustafsson *et al.,* 2010). method is applied for getting the results for the remaining two methods. The addition operation introduces overflow which is accommodated through scaling, thus CORDIC is usually used when lesser number of adders are used .However as the number of adders increase the scaled constant multiplication method is usually employed.

The constant multiplication doesn't give good results as compared to results obtained from CORDIC and scaled constant multiplication.The reason is because the quantization of the coefficients introduces error so as to consider the finite word length effects.Thus no single method is superior than the others in all situations.

4. CONCLUSION

The impact of rotators on the complexity of a signal processing algorithm is very important.The rotations are considered as the most significant and complicated part of the fast fourier transform algorithm.It describes the overall complexity of the processing.It has been shown that different alternatives of rotators could be used like constant multiplication, CORDIC and scaled constant multiplication. The selection of the rotations thus could describe the intricacy of the algorithm where it is employed. Thus there is a trade off exists among the different methods used for finding the rotations and any one algorithm cannot outstand others in all circumstances, because of the varying methods for seeking the rotation and their hardware.

REFERENCES

Cooley J. and W. Tukey.(1965) An algorithm for the machine calculation of complex Fourier series, Math. Comput.,, vol. 19, pp. 297-301.

Dempster. A. G. and Macleod M. D. (2005) Multiplication by two integers using the minimum number of adders, in Proc. IEEE Int. Symp. Circuits Syst., Kobe, Japan, May 24-26, pp. 1814-1817.

Garrido. M., Gustafsson O. and Grajal J.(2011) Accurate Rotations Based on Coefficient Scaling, IEEE Transactions on Circuits and Systems II Express Briefs, vol, 4, pp 124-130.

Garrido. M. and Grajal. J.(2007) Efficient Memoryless CORDIC for FFT Computation,in Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, vol. 2, pp. 113-116.

Gustafsson. O. and Qureshi. F. (2010) Addition aware quantization for low complexity and high precision constant multiplication, IEEE Signal Processing Letters, vol. 17, no. 2, pp. 173-176.

Gustafsson O., Dempster A. G., Lohansson. K., Macleod. M.D. and Wanhammar. L.(2006) Simplified design of constant coefficient multipliers, Circuits, Systems and Signal Processing, vol. 25, no. 2, pp.n 225-251.

Lee. A. and Lang. T. (1992) Constant-factor redundant CORDIC for angle calculation and rotation, IEEE Transactions on Computers, vol. 41 no. 8, pp. 1016-1022.

Qureshi F., Garrido. M. and Gustafsson.O. (2010) Alternatives for low complexity complex rotators , in Proc ,IEEE Int Conf Electronic.circuits and systems ,Athens, Greece, Dec12-15,pp. 267-272.