International Journal of Knowledge Engineering
ISSN: 0976-5816 & E-ISSN: 0976-5824, Volume 3, Issue 1, 2012, pp.-107-110
Available online at http://www.bioinfo.in/contents.php?id=40

COMPARATIVE STUDY OF PIPELINED RECONFIGURABLE FFT PROCESSOR

MORE T.V.* AND PANAT A.R.

Department of Electronics & Communication Engineering, P.I.G.C.E, Nagpur, MS, India.
*Corresponding Author: Email- engi.mtv@gmail.com

Received: February 28, 2012; Accepted: March 06, 2012

Abstract- Fast Fourier Transform is one type of algorithm which is of immense importance in DSP system. It is one of the major blocks in communication and DSP system. FFT processor is also used in various applications such radar, image processing, audio and video broadcasting etc. This FFT is inefficient in terms of area, power and speed. Hence, there is need to build up low power, high speed and reduced area FFT processor. In this paper, we will focus on fundamental of FFT, concept of reconfigurability and pipelining. Then, we will go for various works going on FFT processor. We will discuss about complex multiplication which plays vital role in FFT and do comparison of various architectures for multiplier design.
Keywords- FFT, FFT Processor, DSP, Low Power, High Speed, Reconfigurable, Pipeline, Multiplier

Citation: More T.V. and Panat A.R. (2012) Comparative Study of Pipelined Reconfigurable FFT Processor. International Journal of Knowledge Engineering, ISSN: 0976-5816 & E-ISSN: 0976-5824, Volume 3, Issue 1, pp.-107-110.

Introduction

Nowadays, VLSI technology becomes very popular and growing rapidly. Due to the encroachment of VLSI, the demand for hand held devices such as Laptops, Mobile Phones etc. and DSP system has increased tremendously. Three major issues for VLSI are area, power and speed. So, there is a need to develop efficient system. The Fast Fourier Transform (FFT) is a major block in DSP and Communication system.

The FFT is a faster version of the Discrete Fourier Transform (DFT) and efficiently calculates Discrete Fourier Transform. The DFT is one of the specific forms of Fourier analysis. FFT is also used in various application such as image processing, optical communication, radar, audio and video broadcasting. Thus, due to the huge application of FFT, there is need of efficient FFT processor which should have low power, occupies less area and high speed. One can achieve this efficiency with the help of reconfigurability and pipelining. Reconfigurability provides flexibility to the design reduces time to market and also requires less area where as pipelining yields high throughput, lower clock cycles and low power consumption. The DFT is extremely important in the area of frequency (spectrum) analysis because it takes a discrete signal in the time domain and transforms that signal into its discrete frequency domain representation i.e. it maps a sequence x(n) in frequency domain. Without a discrete-time to discrete-frequency transform we would not be able to compute the Fourier transform with a microprocessor or DSP based system. DSP technology is omnipresent in every engineering discipline. As FFT is a major block in this system it tends to consume more power and area. The solution to this problem has discussed here. This paper has organized as follows in the following sections, detailed analysis of the radix-2 FFT algorithm and signals flow graph are illustrated in Section II. The description of the literature survey is explained in Section III. In Section IV, the various FFT designs are discussed. Section V, speaks about major block in FFT design. Finally, a conclusion is given in Section VI.

FFT Algorithm

The DFT is first proposed by the Cooley and Tukey followed by several enhancement or development by other researchers. The Fast Fourier Transform is an efficient implementation of Discrete Fourier Transform (DFT). The distinction between DFT and FFT is: “DFT” refers to a mathematical transformation, regardless how

it is computed, while "FFT" refers to any one of several efficient algorithms for DFT. DFT is most widely used in DSP. By the DFT, the sequence of N complex of numbers $x_0,\ldots,x_{N-1}$ is transformed into sequence of complex numbers $X_0,\ldots,X_{N-1}$ according to formula

$$X_k = \sum_{n=0}^{N-1} x_n e^{-j2\pi kn/N} \quad k=0\ldots N-1 \qquad (1)$$

W indicates the value of coefficients of the FFT and are often referred as *twiddle factors, x*(*n*) is a time sequence and X[k] is a frequency sequence. The 'n' is the time index and the 'k' is the frequency index. FFT reduces the complex number multiplication and addition from N^2 to $N/2\log_2 N$ and $N\log_2 N$ respectively. Due to this the FFT is a highly elegant and efficient algorithm, which is still one of the most used algorithms in speech processing, communications, frequency estimation, etc. one of the most highly developed area of DSP. In FFT, N indicates integer power of 2, i.e. $N=2^L$ where, L is the number of stages. The FFT derivation mentioned above relies on redundancy in the calculation of the basic DFT. It is one type of recursive algorithm that repeatedly rearranges the problem into two simpler

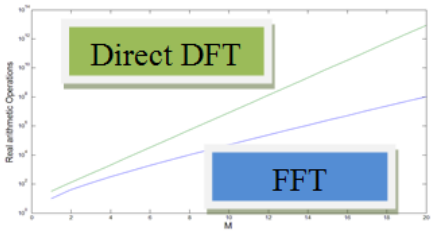

Fig. 1- DFT and FFT Comparison

problems of half the size. Hence the basic algorithm operates on signals of length a power of 2. If we consider radix 2-FFT, it has *M* = $\log_2$ *N* stages, each using *N* / 2 butterflies. Complex multiplication requires 4 real multiplications and 2 real additions. Complex addition/subtraction requires 2 real additions Thus, butterfly requires 10 real operations. Hence the radix-2 *N*-point FFT requires 10(N / 2) log2 *N* real operations compared to about $8N^2$ real operations for the DFT. This is a huge speed-up in typical applications, where N is 64 - 4096 as shown in figure 1.

Literature Survey

The development of the pipelined reconfigurable FFT processor will be surveyed below, because, on the one hand, its history abounds in interesting events, and on the other hand, the important steps correspond to parts of algorithms that will be detailed later. The FFT algorithm is divided into time-based (DIT) and based on the frequency (DIF) fast Fourier transform. Both the DIF and the DIT FFT reorder the data from normal to bit reversed order (or the converse). The basic idea of these algorithms are that the N point FFT is divided into smaller and smaller parts until only two points FFT(Radix-2). Long FFTs are quite often used for frequency analysis and communications applications. These long word lengths affect the memory architecture because long word lengths require more memory bandwidth for the matrix transpositions. So, to resolve this problem many authors suggested different design. The author Bevan Bass [1] proposed design of cached FFT algorithm which is suitable for FFT transforms of any length and radix. But, it is not efficient in terms of area, power, and performance. The author *Jen-Chih Kuo et al.*[2] proposed design of a programmable 64-2048-point FFT/IFFT processor by using CORDIC algorithm which replace the multiplier-based PE. The author *Zheng Wang et al.* [3] proposed design of comparison of finite word length effect of the Decimation In Time (DIT) and Decimation In Frequency (DIF) algorithm, and simplifies the complex multiplication operation in the design of the FFT structure for more general case in which the FFT point is only power of two rather than four. The author *Chao-Ming Chen et al.*[7-2010] proposed design of a 128- to 1024-point partial pipelined/cached-FFT processor for the OFDMA system which results in energy efficiency but it is bit complex and requires more area. The author WANG Xiu-fang et al. [8] proposed a variable point FFT processor using FPGA using 2-D Fourier transform to reduce the memory architecture. However, it does not focus upon the power consumption issues. The author *Ashish Raman et al.* [9-2010] proposed design of a reconfigurable FFT design using Vedic multiplier with high speed and small area. It requires less area, delay than the conventional FFT. But it is limited to 2- point and 4- point FFT. The author *N Kirubanandasarathy et al.[*10-2010] presented a pipelined Fast Fourier Transform (FFT) / Inverse Fast Fourier Transform (IFFT) processor for the applications in a MIMO OFDM based IEEE 802.11n WLAN baseband processor. Here, an attempt is made to achieve high throughput, memory reduction, low power and complex multiplier reduction. But still requires large area, power consumption as compared to design proposed in paper [11]. The author Anand D Darji et al. [11-2010] proposed design of Balanced Binary Tree Decomposition (BBTD) based FFT/IFFT processor for WI-MAX (IEEE 802.16e standard) using VLSI. This proposed design requires very few multipliers compared to conventional pipeline based design using Radix-2 Method. The author Chu Yu,Mao-Hsu Yen et al.[12-2011] designed a novel ROM-less and low-power pipeline 64-point FFT/IFFT processor for OFDM applications also showed that it is low power & low cost design. The author Anuj Kumar Varshney et al. [13] proposed design architecture of reconfigurable FFT using conventional multiplier and Vedic multiplier for achieving high performance of the Vedic reconfigurable FFT in wireless sensor network. It is shown that Vedic requires less power & delay as compared to conventional reconfigurable FFT. But it again limited to 2-point and 4-point. M. Hasan [14] proposed design of Low Power Pipelined Architecture for a MC-CDMA receiver. He has implemented 64-point FFT block based on low power pipelined radix-4 architecture in which coefficient ordering is applied to its second stage to further bring down its power consumption. But, it uses simple complex multipliers which consume most of the power. Qihui Zhang et al. [15] have implemented Low Area Pipelined FFT Processor for OFDM Based Systems. In that, he used memory based architecture to implement radix - 2 DIF FFT processor. Although it is area efficient but it takes more power as compared to proposed design by Hasan [14]. From above survey, it is observed that lot of work has been going on the FFT processor. Some authors working on memory architecture to store twiddle factor, some doing work on programmable architecture and some focusing on pipelining. Based on this, next section takes you towards various FFT architectures.

FFT Design

Due to the increasing demand of FFT processor in numerous applications, there has been tremendous growth in design of high performance FFT processor.

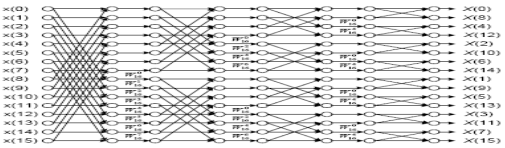

Fig. 2- Radix-2 16 Point FFT Processor

There are various structures for the implementation FFT processor. Some of them are listed here:

a. Simple FFT Processor
b. Reconfigurable FFT Processor
c. Pipeline FFT Processor

a. Simple FFT Processor

It is based on divide and conquer approach. Here, large size FFT is divided into smaller sizes and at each stage input values are added, subtracted and subtracted values are multiplied with twiddle factor as shown fig 2. Though this method is very simple, fast but it requires large memory to store twiddle factor thus degrades the performance and also power consumption.

b. Reconfigurable FFT Processor

Next advance class of FFT processor is reconfigurable FFT processor. There are huge applications of FFT processor. This application requires different sizes of FFT. So, if we design specific processor for particular application, it does not allow reuse of component and causes under optimization of hardware resources. The solution to this problem is to use reconfigurable FFT processor. It allows change of functionality as per users need or environmental condition without altering the entire hardware. Thus, this adds flexibility to the design. If we design single reconfigurable FFT processor, it can be used for various applications and this will reduce time to market. Fig.3 shows example reconfigurable FFT processor. Here we can go for either for 2-point or 4-point FFT with the help of switch.

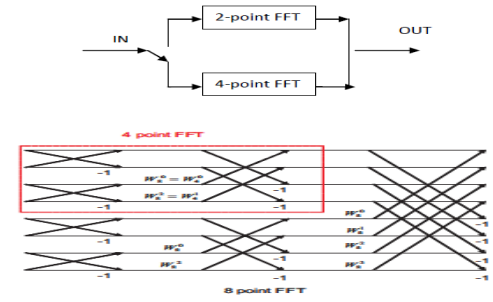

Fig.3- Concept of Reconfigurable Processor

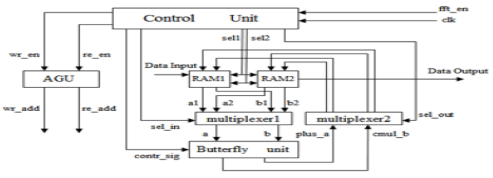

Fig. 4- Example of Pipeline FFT processor

c. Pipeline FFT Processor:

Communication is need of today's world. Entire world is driven by communication in form of internet, mobile phones, fax etc. This Communication system requires real time and non-stop processing of data. High data throughput is also an essential part of communication. Simple or reconfigurable FFT processor doesn't achieve this thing. This scenario can be overcome with the help of another advance of FFT processor i.e. pipeline FFT processor. The pipeline FFT processor processes the data in sequential manner so as to achieve high data throughput necessary for communication system. Pipeline FFT process one sample at each clock cycle. It also provides real and non stopping processing as the data sequence passing the processor. The fig 4. [16] shows one example of pipeline FFT processor. The control unit controls entire operation of the circuit it issues read & write signal to address generation unit (AGU) which generates 8 read and 8 write addresses, which determine the data access to outer memories.

The DIF Butterfly Unit

For FFT algorithm, the central component is the BU that calculates the sum and difference of two input data, and plays an extremely important role in computing the product of the difference and twiddle factors.

The RAM Unit

The RAM1 and RAM2 are made up of 8 32-bit registers respectively. Data is always written to the outside memories from RAM2, and it is always read to RAM1 from the outside memories.

FFT Bloc

In butterfly operation, the input signal is multiplied with twiddle factor. This operation results in complex multiplication. Thus, the complex multiplication is major block of FFT processor. This complex multiplication put the constraint on computational performance of communication and DSP system. As, complex multiplication dominate the execution time. Complex multiplication of two numbers requires 4 multiplier, 2 adders and 1 subtractor. This results in high power consumption, large area and poor performance. Thus, there is need to design good multiplier which should be fast, occupies less area and consumes less power. This is feasible only if we reduce the number operation. In literature, we could get various multiplier structures. Few of them are enlisted in table 1. If we compare this various structures it is found that Wallace tree multiplier is fast, consumes low power and high speed.

Table 1- Various Multiplier Structures

Structure Parameters	Array Multiplier	BOOTH Multiplier (Radix-4)	Wallace Tree Multiplier
No. of Slices	229	96	69
Delay (ns)	47	26.67	25.43
Power (mW)	104	79	87

Conclusion

The purpose of this paper has been to focus on the various works going in the field of FFT processor. Here, we have discussed about various FFT processor structures used for implementation of FFT. It is observed that the reconfigurable or pipelining FFTs

are designed. This design either requires less area or some of them are high speed but we are not able to achieve both. If we combine this two approach i.e. pipeline and reconfigurable we could achieve high speed and low power. But, there is also one problem. The main block in FFT processor is complex multiplication which consumes most of the power, area and speed. Though if we design reconfigurable pipeline FFT processor, multiplication operation always going to limit the computational performance of DSP system. So, the basic objective for any FFT processor is to design good multiplier consumes low power, occupies less are and high speed. Taking into consideration this aspect, various structures for multiplier implementation has listed. One has to take proper decision at proper level so as to achieve desired goal.

References

[1] Saha P.K., Banerjee A. and Dandapat A. (2009) *International Journal on Electronic and Electrical Engineering*, 07(II), 38-46.

[2] Blahut R.E. (1987) *Reading, MA: Addison-Wesley*.

[3] He S. and Torkelson M. (1995) *IEEE International Symposium on Circuits and Systems*., 2313-2316.

[4] Volder J.E. (1959) *IRE Trans. Electron. Comput*., EC-8, 330-334.

[5] Krishnan R., Jullien G.A. and Miller W.C. (1985) *IEEE Trans. Acoust., Speech, Signal Processing*, 34.

[6] Aoki T., Hoshi K. and Higuchi T. (1999) 29*th IEEE International symposium on Multiple-Valued-Logic*, 200-207.

[7] Huang Z. and Ercegovac M.D. (2005) *IEEE Transactions on Computers*, 54(3), 272-283.

[8] Kung S.Y., Whitehouse H.J. and Kailath T. (1985) *VLSI and Modern Signal Processing*.

[9] Prabir Saha, Arindam Banerj, Partha Bhattacharyya, Anup Dandapat (2011) *IEEE Students' Technology Symposium*, 238-241.

[10] Anvesh Kumar, Ashish Raman (2010) *IEEE* pp. 836-838.

[11] Laxman P. Thakre, Suresh Balpande, Umaeh Akare, Sudhair-Lande (2010) *International Conference on emerging trends in Engineering and Technology*, 614-619.

[12] Nidhi Mittal Abhijeet Kumar (2011) *International Journal of Computer Applications* 35(1), 17-20.

[13] Mehta P. and Gawali D. (2009) *IEEE International Conference on Advances in Computing, Control, and Telecommunication Technologies*, 640-642.

[14] Tiwari H.D., Gankhuyag G., Kim C.M. and Cho Y.B. (2008) *IEEE International SoC Design Conference*, 65-68.

[15] Ashish Raman, Anvesh Kumar and Sarin R.K. (2010) *journal of computer science and engineering*, 1(1), 59-63.

[16] Bingrui Wang, Qihui Zhang, Tianyong Ao, Mingju Huang, (2010) *Second International Conference on Computer Modeling and Simulation*, 432-435.