IJESRT
INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

REVIEW PAPER ON EFFICIENT VLSI AND FAST FOURIER TRANSFORM ARCHITECTURES

Kavitha MV, S.Ranjitha, Dr Suresh H N
*Research scholar, Dept. Of E&IE, BIT, Bengaluru-560004.
Assistant professor, Gopalan college of Engineering and management ,Bengaluru,
MTech, Bangalore
Professor &amp; Coordinator for PG Studies and Research, Dept. Of E&amp;IE BIT,

DOI: 10.5281/zenodo.345685

ABSTRACT

A fast Fourier transform (FFT) is any fast algorithm for computing the DFT. The development of FFT algorithms had a tremendous impact on computational aspects of signal processing and applied science. The decimation-in-time (DIT) fast Fourier transform (FFT) very often has advantage over the decimation-in-frequency (DIF) FFT for most real-valued applications, like speech/image/video processing, biomedical signal processing, and time-series analysis, etc., since it does not require any output reordering. Fast Fourier Transform (FFT) has the major role in obtaining the signal characteristics with minimum use of resources. Some of the algorithms have been proposed on FFT, such kind of algorithms were less effective in the performance parameters. In this paper, a variety of available FFT algorithms are presented and then different architectures are outlined by exploring the techniques and algorithms involved in each of the architectures. The widely adopted architectures and trends in architectural modification to reduce power consumption and area and to achieve high throughput are discussed.

KEYWORDS: FFT algorithm, FPGA, Mixed radix, Pipeline FFT

INTRODUCTION

There has been an increasing interest in the computation of fast Fourier transform (FFT) of real-valued signals, referred to as real fast Fourier transform (RFFT), and in-place fast Fourier transform (IFFT) of Hermitian-symmetric signals, referred to as real in-place fast Fourier transform (RIFFT). This is because most of the physical signals, such as biomedical signals, are real. The real-valued signals exhibit conjugate symmetry, giving rise to redundancies. This property can be exploited to reduce both arithmetic and memory complexities. Most FFT architectures can be divided into two categories: pipelined and in-place architectures. Pipelined architectures contain either feed forward or feedback data paths . The feedback architectures have been referred to as single-path delay feedback, and the feed forward architectures have been referred to as multipath delay commutator. Much research has been carried out on the design of pipelined architectures for computing the FFT of complex and real-valued signals for high-throughput applications. These pipelined architectures are suitable for high-throughput applications. For example, two-parallel architecture requires N/2 cycles for an N-point FFT. The focus of this brief is on in-place architectures where few processing elements (PEs) are used in a memory-based architecture. These architectures, also referred to as continuous-flow architectures, are well suited for low to moderate speed applications.

Several memory-based architectures have been proposed to achieve smaller area [5]–[10]. Higher radix butterfly units and/or parallel processing can be utilized to increase the throughput [9]. Very few in-place architectures have been proposed for real-valued signals [11], [12] based on the packing algorithm, which computes a complex FFT and requires additional operations for post processing.
In [14], a memory-based FFT architecture that computes the RFFT based on the modified radix-2 algorithm in [13] was proposed. The algorithm computes only half of the output samples and removes the redundant operations

from the flow graph. The modified flow graph contains only real data paths as opposed to complex data paths in a regular flow graph. Therefore, the word length in the memory units of the proposed FFT processor is W, where W is the word length chosen to represent either the real/imaginary component. A PE was proposed to efficiently implement the operations in the modified flow graph. The PE consists of two radix-2 butterflies and can process four samples in parallel. Novel addressing and bank assignment schemes were proposed for conflict-free memory accesses. Furthermore, it is shown that the proposed scheme can be extended to support multiple parallel PEs. The key contribution of their brief is the design of an in-place RFFT/RIFFT architecture based on an FFT flow graph optimized for real signals that achieves the least area–time (AT) product for RFFT/RIFFT among all known designs, where area corresponds to the data-path area and time corresponds to the number of cycles.

LITERATURE SURVEY

Scalable architecture for in-place fast Fourier transform (IFFT) computation for real-valued signals was presented in [14]. Their proposed computation was based on a modified radix-2 algorithm, which removed the redundant operations from the flow graph. The processing element (PE) was proposed using two radix-2 butterflies that can process four inputs in parallel. A conflict-free memory-addressing scheme was proposed to ensure the continuous operation of the FFT processor. Furthermore, the addressing scheme was extended to support multiple parallel PEs. Their proposed real-FFT processor simultaneously requires fewer computation cycles and lower hardware cost compared to prior work. The number of computation cycles was reduced proportionately with the increase in the number of PEs. It can be seen that their proposed design with a single PE required the same number of computational cycles as the one in [11]. This is achieved with a PE of two real butterflies and one complex multiplier, while the prior design requires a complex radix-4 butterfly consisting of 12 complex adders and three complex multipliers. Furthermore, their proposed design with two PEs required 4 complex adders and 2 complex multipliers. Their proposed design is scalable with respect to the number of PEs [14]. But, the multipliers are rather complex circuits and must typically operate at a high system clock rate. Reducing the delay of a multiplier is an essential part of satisfying the overall design. Multiplications are very expensive and slow the overall operation. The performance of many computational problems is often dominated by the speed at which a multiplication operation can be executed. Even though the work presented in [14], decreases the total computation time approximately linearly with the increase in the number of PEs, in this article i have decided to present a low power and a fast complex multiplier, which makes the system take reduced computation time. From the literature, it is clear that the Radix-4 booth multiplier consumes less power and is faster than the other multipliers. While comparing the radix 2 and the radix 4 booth multipliers we found that radix 4 consumes lesser power than that of radix 2. This is because it uses almost half number of iteration and adders when compared to radix 2.

FFT ARCHITECTURES

3.1] Single Memory Bevan M. Baas described about single, dual, array and cached memory architectures in his paper [13]. Single-memory architecture consists of a memory and a processor. A bidirectional data bus runs between the memory and the processor through which data exchange takes place between them. Data are read from memory and written back to the memory using bidirectional bus.

3.2] Dual Memory In dual-memory architecture [13] two memories are used. A processor is present in between the two memories. Bus connects the memory 1 and memory 2 to the processor. Data input from one memory will pass through the processor to the second memory during processing of data. This process is done until the transform has been calculated..

3.3] Array Memory Array architecture [13] consists of a number of independent processing elements with local buffers and they are connected through a network.

3.4] Cached Memory The cached-memory [12] architecture consists of a cache memory in addition to the single memory architecture. Cache memory is present in between the processor and main memory and it makes the process faster.

3.5] Pipelined Architectures In pipelined architectures continuous processing of data is done through a series of processing elements. Pipelined architectures have high throughput rate when compared to the other architectures. For real-time applications pipelined architectures are more suitable.

3.5.1 SINGLE PATH ARCHITECTURES :

3.5.1.1 R2SDF: In paper [3] described that in Radix-2 single path delay feedback (R2SDF) [14] architecture data is processed in a feedback manner. Every stage of this architecture has a data path that goes through the multiplier. The number of processing elements (PEs) and multipliers used for radix-2 multi-path delay commutator architecture and this architecture are same, but the delay elements required for this architecture is only N-1[3]. In a single-path delay feedback FFT processor several long and wide delay lines are used. Delay lines are designed using shift registers which is nothing but the cascading of D flip flops. For each clock edge, data movement is in a forward direction in a lock-step fashion. Due to this half of the registers change states resulting in the wastage of power. For achieving low power and low area a R2SDF architecture [7] has been proposed with SRAM replacing the shift registers. To perform read and write operations in one clock cycle a dual port memory is required. But it can be replaced by two single port SRAMs[11] which can save area up to 33%.The read operation is performed during the first half cycle and the write operation in the next half cycle. Current mode SRAMs [9] are used which helps in power reduction. Two registers, one before the PE and the other after, are inserted to prevent the data access of the SRAM becoming critical paths. To access data from SRAM a ring counter is used instead of address decoder and true-single-phase-clock flip flops are used in the ring counters to reduce power consumption.

3.5.1.2 R4SDC: In paper [3] described that in Radix-4 Single path Delay Commutator [5] a modified radix-4 algorithm is used .In this architecture the multipliers are utilized up to 75% using the programmable butterflies. The memory requirement can be reduced to 2N-2 using combined delay-commutator[3]. The most complicated parts in this architecture are butterfly and delay-commutator. For low power applications a new commutator is introduced [12].This commutator is termed as IDR architecture which is far superior than single port, Dual port and Triple port RAM architectures in terms of power and area for two stages. For the third stage single port RAM is the good choice. The RAM blocks in this IDR architecture are enabled only if the outputs are needed which helps in reducing switching activity which in turn reduces the power consumption.

3.5.2 MULTI PATH ARCHITECTURES :

3.5.2.1 R4MDC:

Multipath Delay Commutator processes data in parallel. In R2MDC [11] two parallel data paths are present and two data are processed parallel in butterfly units i.e., in this architecture both pipelining and parallel processing are used. The two data which are to be processed are fed correctly to the butterfly units with the help of delay elements and multiplexer. This architecture requires (log2N-2) multipliers, and 3N/2-2 delay elements[3].Radix-4 Multipath Delay Commutator [11] is same as R2MDC but here radix-4 algorithm is used and so four parallel data paths are present. In this architecture the components are utilized only up to 25% and this is because they are active once during the four cycle. This architecture requires 3log4N complex multipliers and 5N/2-4 registers [3].The main drawbacks are high hardware cost and low utilization. However, if the four inputs are fed at the same time, computational elements are utilized perfectly which in turn reduce memory requirements. Energy optimized high performance FFT processor [3] is designed to overcome the difficulties in conventional R4MDC architecture. In this architecture the input reordering requires only less hardware .ROM is used for storing the twiddle factors. Thus this modified architecture requires only 7N/4-4 memory cells. As the reordering of the input data requires only small buffers for reordering this architecture helps in reducing hardware cost as well as energy consumption to some extent. This modified architecture achieves improvement in memory utilization and reduction in memory leakage energy. The two pipeline architectures i.e., single path delay feedback [4] and multi-path delay commutator have both advantages and disadvantages. The advantage of multipath delay commutator is that the throughput rate is much higher when compared to the single path feedback scheme. But, the disadvantages of this architecture are the number of data path required is large, the size of the FFT is large. The requirement of the memory cells and complex multiplier in the Multipath Delay Commutator architecture is comparatively higher than that of the Single path Delay Feedback structure. The MDC scheme helps in achieving higher throughput rate ,on the other side the SDF scheme needs lesser memory and hardware cost. In UWB applications MDC architecture is preferred due to its high-throughput-rate. But in the conventional Radix-2 MDC architecture frequency has to be raised to maintain the required throughput rate [12]. The Radix-4 MDC architecture and the Split-Radix MDC architecture have their limitation on FFT size and higher hardware cost [3] respectively. In addition, the higher radix FFT algorithm is difficult to be implemented in the traditional MDC architecture. By increasing the number of data paths in the MDC architecture the higher throughput rate can be achieved. As this architecture helps multiple data to operate at the same time it requires more memory and complex multipliers which in turn increases the hardware cost. In order to handle the above difficulties the features of the SDF and MDC architectures are combined to produce a mixed-radix multipath delay feedback (MRMDF) architecture [15] which is a four-data-path pipelined FFT architecture. The proposed FFT/IFFT architecture is not only suitable for the proposed UWB physical layer but also provide the necessary throughput rate to meet the UWB specifications. This architecture has lower hardware cost compared with the conventional MDC architecture and also saves power dissipation by using the high-radix FFT algorithm.

3.5.2.2 MRMDF:

The MRMDF architecture [14] has the following features. It uses both radix-2 and radix-8 algorithm. The four parallel data paths provide higher throughput rate. There are two places where we need to reorder the data and they are to reorder the input data and to reorder the results of each module. As single feedback design is used the memory required for the above two reordering gets reduced. In this architecture scheduling scheme and specified constant multipliers are used and this reduces the number of complex multipliers. This architecture helps in achieving throughput of 1Gsamples per second at the clock rate of only 250MHz. Architectures R22MDF [27], R24MDF [8] are designed by combining the features of SDF and MDC architectures. In this architecture [8] canonic signed division multiplication is used which helps in achieving low area and less power consumption. This architecture is also a four parallel architecture with which the throughput rate can be increased. Further as this architecture uses radix-24 algorithm and the power dissipation is also reduced here.

3.5.2.3 SRMDC

Apart from these architectures there are also architectures based on split-radix FFT algorithm. They are Split radix Multipath Delay Commutator and Split radix single path Delay Feedback structures. But due to the disadvantages in the single path structures ,multipath split-radix commutator [16] is designed. This architecture is a four parallel path architecture in which the data arrive at the computing elements perfectly with the help of delay commutator. In this architecture a dual port memory is used. This processor will operate at a frequency of 350MHz. The main area to be concentrated in a FFT processor is twiddle factor generation and multiplication. There are different multipliers including complex multipliers and CO ordinate Rotation Digital Computer (CORDIC) [2],[3] based multipliers. The area required for storing the twiddle factors is large as it is stored in a ROM memory and used for further multiplications. In order to overcome this disadvantage a ROM-free twiddle factor generator is used in [10] in which simple accumulators, shifters, registers and adders are used and this architecture is also a reconfigurable architecture .Multiplication is another important area to be considered. A CORDIC based multiplier [9] is used to achieve hardware simplicity when compared to booth multipliers. Apart from these architectures the architectures are now derived for simplifying the architecture. These architectures are derived using decomposition of algorithm. There are also FFT architectures when the inputs are only real. For real valued FFT architectures some of the outputs are repetitive in nature. Using this concept real valued FFT architectures are designed with hardware simplicity by removing the redundant outputs.

The FFT computation depends on different radix algorithms and architectures. To meet the communication requirement, different designs have been developed to implement corresponding communication condition. Discrete Fourier Transform (DFT) is usually adapted to process digital signals, but it has high computational complexity and takes a long time to implement. Therefore, Cooley and Tukey proposed in 1965 [3] a method to re- duce the computational complexity of DFT. FFT computation rules follow Radix- *n*, where a higher *n* indicates more complex computational standards and circuits. A Radix- *n* algorithm can compute an FFT that has a point size of a power of any multiple of *n*.

Therefore, if the required computation is 512-point FFT, the function needs to be implemented with both Radix-2 and Radix-4. The Radix-2 algorithm reduces DFT computational complexity from $O (N 2)$ to $O (N \log N)$ [4,5]. Many FFT architectures have been designed, including memory- based design, cached memory architecture and pipelined architecture [7]. A primary N -point wave pipelined technique has several campus- national stages in N -point FFT computation, and each stage con- trains a register element, a butterfly (BF) circuit, and a twiddle-factor computational circuit. The entry part of each stage decreases at each computational stage. After BF computation, the output data are multiplied by the corresponding twiddle factors, then transmitted to the next level. The butterfly circuits and register elements are each changed by a different radix. The complex multiplier number of N -point wave pipelined technique is $(\log 2\ N) - 1$. The butter- fly circuits have a complexity of $\log 2\ N$, and require $(N - 1)$ register elements. However, the throughput and speed of FFT computation are insufficient to meet the requirement. Therefore, the MDC architecture has been proposed to obtain high efficiency. The wave pipelined technique architecture is less efficient than the MDC architecture. The MDC architecture is multi-path, and therefore can input multiple FFT data simultaneously. The complexity of N -point MDC multiplier is $(\log 2\ N) - 1$; that of the butterfly circuits is $\log 2\ N$, and the number of register elements is $(3\ N /2) - 1$. The comparison of property with wave pipelined technique and MDC. The implementation tools for FFT with FPGA device.

PROBLEM STATEMENT

1. A power efficient complex multiplier in the PE will be proposed. Multipliers are the key components of many high performance systems such as FIR filters, microprocessors, digital signal processors, etc. A system's performance is generally determined by the performance of the multiplier.

2. Complex number multiplication is important in digital single processing, especially in DIT-FFT twiddle factor multiplied with input is complex number. Four real number multiplication and two additions or subtractions are involve in complex number multiplication. Carry needs to be diffused from the least significant bit (LSB) to most significant bit (MSB) when binary partial products are added in real number multiplication. After binary multiplication, the overall speed is limit by the addition and subtraction.

3. The Array multiplication and Booths algorithm are the most common multiplication algorithms. The Booth algorithm was invented by Andrew Donald Booth in 1950 , that multiplies two binary numbers. These Booth algorithms are used for multi-bit and exponential operations that require large partial results and carry registers , and takes comparable computational time. For designing of complex number multiplier basic idea is adopted from designing of multiplier.

PROPOSED METHODOLOGY

1. Radix-8 booth multiplier will be implemented as a complex multiplier for In place FFT architecture. And since this multiplier operates in parallel and requires less number of adders.
2. Using Radix- 8 booth multiplier the computation time required will be less and also less power consuming technique.
3. The proposed In place FFT architecture for real valued signals will be designed using verilog language in Xilinx ISE tool and the design will be synthesized and implemented to generate the area and power report. For generating power report X power analysis tool is executed after implementing the design on the FPGA device.

CONCLUSION

Basic architectures for FFT implementations are explored and a comparison between the widely adopted architectures in terms of hardware complexity and scope of optimization is discussed .In this paper, we tend to expected a radix-*r* primarily based MDC MIMO FFT/IFFT processor for process *Ns* streams of parallel inputs, wherever *r* = *Ns* for achieving a 100 percent utilization rate.

REFERENCES

1. J. Lee, H. Lee, S. I. Cho, and S. S. Choi, "A high-speed two parallel radix-24 FFT/IFFT processor for MB-OFDM UWB systems," in Proc. IEEE Int. Symp. Circuits Syst., May 2006, pp. 4719–4722.
2. M. Ayinala, M. Brown, and K. K. Parhi, "Pipelined parallel FFT architectures via folding transformation," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 20, no. 6, pp. 1068–1081, Jun. 2012.
3. R. Radhouane, P. Liu, and C. Modin, "Minimizing the memory requirement for continuous flow FFT implementation: Continuous flow mixed mode FFT (CFMM-FFT)," in Proc. IEEE Int. Symp. Circuits Syst., May 2000, pp. 116–119.
4. B. G. Jo and M. H. Sunwoo, "New continuous-flow mixed-radix (CFMR) FFT processor using novel in-place strategy," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 52, no. 5, pp. 911–919, May 2005.
5. A. T. Jacobson, D. N. Truong, and B. M. Baas, "The design of a reconfigurable continuous-flow mixed-radix FFT processor," inProc. IEEE Int. Symp. Circuits Syst., May 2009, pp. 1133–1136.
6. C. F. Hsiao, Y. Chen, and C. Y. Lee, "A generalized mixed-radix algorithm for memory-based FFT processors," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 57, no. 1, pp. 26–30, Jan. 2010.
7. P.-Y. Tsai and C.-Y. Lin, "A generalized conflict-free memory addressing scheme for continuous-flow parallel-processing FFT processors with rescheduling," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 19, no. 12, pp. 2290–2302, Dec. 2011.
8. D. Reisis and N. Vlassopoulos, "Conflict-free parallel memory accessing techniques for FFT architectures," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 55, no. 11, pp. 3438–3447, Dec. 2008.
9. H. Chi and Z. Lai, "A cost-effective memory-based real-valued FFT and Hermitian symmetric IFFT processor for DMT-based wire-line transmission systems," in Proc. IEEE Int. Symp. Circuits Syst., May 2005, vol. 6,

10. A. Wang and A. P. Chandrakasan, “Energy-aware architectures for a real-valued FFT implementation,” in Proc. Int. Symp. Low Power Electron. Design, Aug. 2003, pp. 360–365.
11. M. Garrido, K. K. Parhi, and J. Grajal, “A pipelined FFT architecture for real-valued signals,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 56, no. 12, pp. 2634–2643, Dec. 2009.
12. Manohar Ayinala, Yingjie Lao, and Keshab K. Parhi, “An In-Place FFT Architecture for Real-Valued Signals,” IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS, VOL. 60, NO. 10, OCTOBER 2013.
13. Sandesh S. Sao kar, R. M. Banakar, and Saroja Siddamal, “High Speed Signed Multiplier for Digital Signal Processing Applications,” 2012 IEEE.
14. S.S. Kerur, Prakash Narchi Jayashree C .N., Harish M. Kittur “Imp lementation of Vedic mu ltip lier for digital signal processing,” international journal of computer applicatio n, 2011, vol. 16, pp 1-5.
15. Devika Jaina, Kabiraj Sethi, and Rutuparna Pamda, “Vedic Mathematics Based Multiply Accumulate Unit,” International conference on computational intelligence and commun ication system, 2011, pp 754-757.
16. Man Yan Kong, J.M. Pierre, and Dhamin Al-Khalili, “Efficient FPGA Implimentation of Co mplex Multipliers Using the Logarithmic Nu mber System,” 2008 IEEE. pp 3154-3157.