

# Low Power and High Performance Structures for Fast Fourier Transform Processor

Anwar Bhasha Pattan  
 Research Scholar,  
 ECE Department, JNTUH, Hyderabad

M. Madhavi Latha  
 Professor, ECE Department  
 JNTUH, Hyderabad

## ABSTRACT

Fast Fourier Transform (FFT) being the most important block in many signal processing and communication systems, consumes more power due to its huge computational complexity. Hence, low power design for FFT hardware gains the focus of researchers now-a-days. There are many algorithms and architectures proposed in the literature to achieve lower computational complexity and power dissipation. In this work, some of the best suitable algorithms and architectures for hardware implementation are analyzed in terms of complexity, speed and power consumption by using Xilinx ISE tools and proposed a low power and high performance architecture and algorithm combination for FFT computations.

## General Terms

FFT Algorithms, FFT Architectures, Butterfly Structures.

## Keywords

Architecture, Computational complexity, FFT, Low power.

## 1. INTRODUCTION

Fast Fourier Transform (FFT) is a method for fast computation of Discrete Fourier Transform (DFT). It is widely used in many signal processing and communication applications. There are many FFT algorithms proposed for computing DFT [1], but a few are suitable for VLSI implementation due to their regular structure [2]. Cooley-Tukey FFT (CTFFT) algorithms are widely adopted due to their simple structure and suitable for any composite length.

CTFFT algorithms used the divide and conquer approach to reduce the computational complexity. In a radix-2 algorithm, the N point FFT is recursively decomposed into an (N/2) point FFT, then (N/4) and so on to reach the 2 point FFT to achieve reduced computations of the order of  $N \log_2 N$  from  $N^2$  in direct computations [3]. In Decimation in Time (DIT) algorithms, the time samples are in decimated order and frequency samples are in regular order whereas in Decimation in Frequency (DIF), the frequency samples are in decimated order and time samples are in regular order. Higher radix algorithms can be used to reduce the computational complexity, but the structure becomes more complex compared to the lower radix algorithms. Hence, higher radices over 8 are not preferred for hardware implementation. Radix-8 algorithms yield lesser number of multiplications [4] and the complexity of radix-8 structure can be simplified by using radix- $2^3$  structure instead of direct implementation. The signal flow graphs/butterfly structures for radix-2, radix-8 and radix- $2^3$  algorithms for an 8 point FFT are shown in figures 1, 2 and 3 respectively.

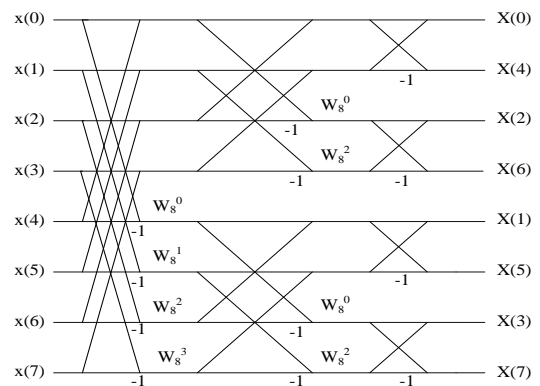


Fig 1: radix-2 butterfly structure for 8 point FFT

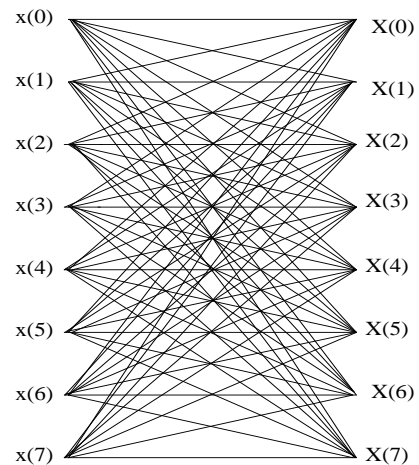


Fig 2: radix-8 butterfly structure for 8 point FFT

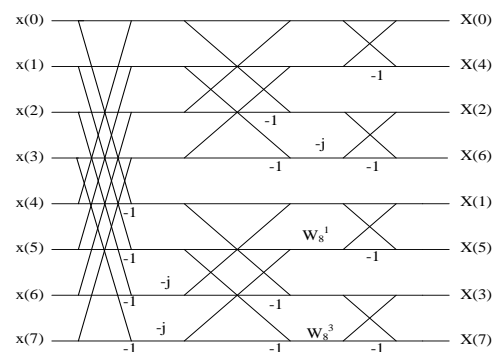


Fig 3: radix- $2^3$  butterfly structure for 8 point FFT

From Fig 3, it is evident that the radix- $2^3$  algorithm for 8 point FFT needs only two non trivial complex multiplications. Multiplication with  $-j$  is simply swapping the real and imaginary parts of the complex number with a sign change in imaginary part. This leads to lesser number of multiplications. The two non trivial multiplications, i.e., multiplications with

$W_8^1$  and  $W_8^3$  can be implemented with a structure shown in Fig 4. This structure leads to a multiplier less design for the 8 point FFT and consumes low power.

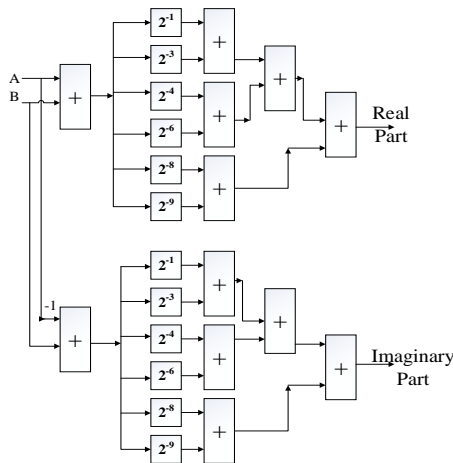


Fig 4: Structure for multiplication with  $\{(\sqrt{2}/2)(1\pm j)\}$

## 2. FFT ARCHITECTURES

There are various types of FFT architectures present in the literature. Among them, memory based architectures and pipelined architectures are widely adopted. Memory based architectures are simple structures with a processing element and memory banks, but slower because of the large memory access time between processing element and memory banks. To increase the speed, cache memory based architecture was proposed in [5] where, a data cache is employed between the processing element and memory to enable the pre-fetching of data but, it leads to extra hardware and complexity.

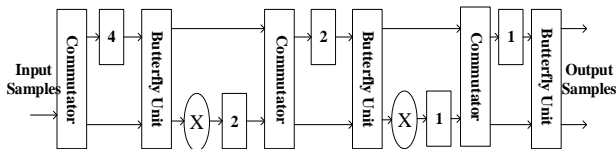


Fig 5: Radix-2 pipelined MDC architecture for 8 point FFT

Pipelined architectures achieve high throughput because of their pipelined structure to enable parallel processing. These architectures are the best suitable for real time serial data [6]. They are classified as Multipath Delay Commutator (MDC) and Single-path Delay Feedback (SDF) architectures. Fig 5 shows the radix-2 MDC for 8 point FFT where the serial samples are split into two paths using commutator, butterfly operations are performed between the right samples according to the radix-2 structure with proper delay and then twiddle factor multiplications. In radix-2 SDF, the delay elements are effectively utilized by sharing the same storage between the input and output of the processing element as shown in Fig 6. Higher radix pipelined architectures require careful design of processing elements to reduce the complexity.

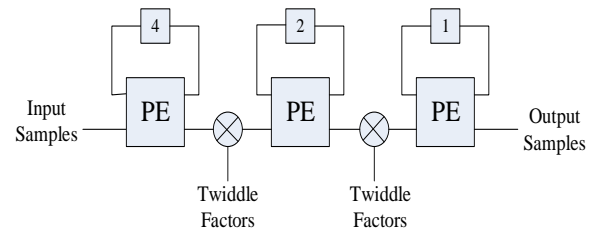


Fig 6: Radix-2 pipelined SDF architecture for 8 point FFT

## 3. PROCESSOR DESIGN FOR 8 POINT FFT

The algorithms and architectures discussed in previous sections are designed using VHDL which is a powerful hardware description language with rich constructs for synthesis [7]. Block diagram of the 8 point FFT processor is shown in Fig 7. In this block diagram, the 8 point butterfly structure is replaced by one of the four structures, i.e., radix-2, radix-8 and radix-23 structures with and without multipliers to form four different processor designs. The serial to parallel and parallel to serial conversion blocks are used to operate on serial input samples and generate serial output samples respectively.

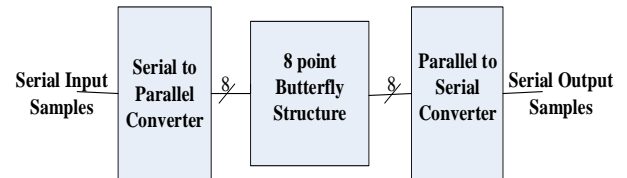


Fig 7: Block diagram of an 8 point FFT processor

The multiplier unit shown in Fig 8 is same for both SDF and MDC architectures for twiddle factor multiplications at each of the three stages. Processing element in SDF consists of multiplexers, de-multiplexers and butterfly unit to selectively perform the butterfly operations whereas in MDC, commutator is used for this purpose. Fig 9 shows the butterfly unit which is same for both the architectures to perform additions and subtractions between the complex inputs. Control units generate the select signals for commutator, processing element and multiplier at each stage.

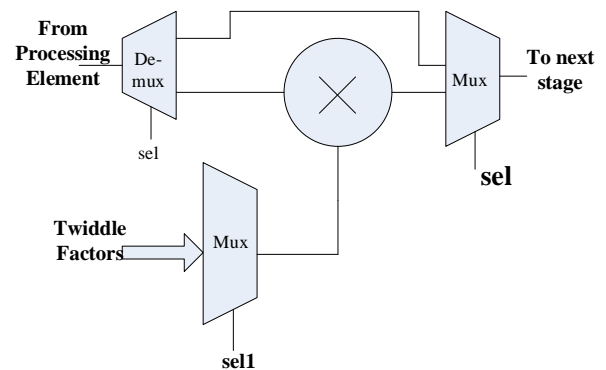


Fig 8: Multiplier unit for both MDC and SDF architectures

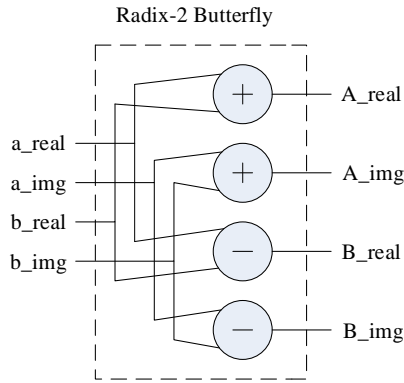


Fig 9: Butterfly unit for both MDC and SDF architectures

#### 4. RESULTS AND DISCUSSION

For functional verification, ISIM simulator of Xilinx is used. The results for all the six designs are same and are shown in Fig 10.

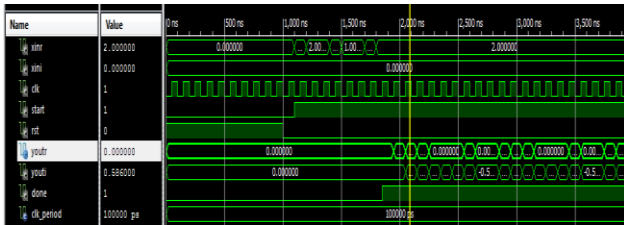


Fig 10: Simulation results of 8 point FFT processor

Synthesis is done on Vertex 4 FPGA, Xpower analyzer is used for the power analysis and the results are depicted in Table.1. Though the radix-8 algorithm takes less number of multipliers than radix-2 algorithm, it occupies huge area due to more number of adders. It is seldom used because of its low performance and high power consumption. The radix-23 multiplier less structure is proved to be the best algorithm in terms of speed and power consumption compared to other algorithms.

Table 1. Comparison of results between different designs

Algorithm/ Architecture	radix-2	radix-8	radix-2 <sup>3</sup> design-I	radix-2 <sup>3</sup> design-II	radix-2 MDC	radix-2 SDF
No. of Multipliers	48	30	06	-	08	08
No. of 16 bit adders/ subtractors	48	14	51	71	12	12
No. of 32 bit adders/ subtractors	24	116	04	-	04	04
No. of Slices	762	1703	658	722	434	412
No. of flip-	293	293	293	293	619	557

flops						
No. of LUTs	1229	3328	1027	1082	558	590
Frequency (MHz)	67.088	58.648	86.338	109.719	143.801	187.573
Power(mW)	292	341	281	278	249	232

The pipelined radix-2 architectures achieve even lesser power consumption than the direct implementation of radix-23 algorithms. From Table 1, it is shown that both the MDC and SDF architectures have same computational complexity but SDF is the efficient architecture in terms of power consumption and performance. For a higher point FFT, specifically, if FFT length is a power of 8, then radix-23 algorithm can be used in SDF architecture to achieve lowest power and high performance compared to any other structures.

#### 5. CONCLUSION

An 8 point FFT processor is designed using radix-2, radix-8 and radix-23 algorithms. Results show that radix-23 algorithms achieve high performance and low power. But, direct implementation of algorithm is not sufficient to fully optimize the processor. So, the pipelined radix-2 MDC and SDF architectures are investigated. For an 8 point FFT, radix-2 SDF achieves highest performance and lowest power.

In future, the radix-23 algorithms are tested in pipelined architectures for higher point FFT. For FFT of length which is a power of 8, the radix-23 algorithm and SDF architecture combination is expected to be the best optimized structure.

#### 6. REFERENCES

- [1] P. Duhamel and M. Vetterli "Fast Fourier transforms: a tutorial review and a state of the art", Signal Process.vol. 19, no. 4, pp. 259-299, Elsevier, 1990.
- [2] Tzi-Dar Chiueh, Pei-Yun Tsai, "OFDM baseband receiver design for wireless communications", John Wiley & Sons (Asia) Pte Ltd, 2007.
- [3] J. W. Cooley and J. Tukey, "An algorithm for machine calculation of complex Fourier series", Math. Comput., vol. 19, pp. 297–301, Apr.1965.
- [4] G.D. Bergland, "A Fast Fourier Transform algorithm using base 8 iterations", Math. Comp., Vol. 22, No. 2, April 1968, pp. 275-279
- [5] B. M. Baas, 'A low-power, high-performance, 1024-point FFT processor', IEEE J. Solid-State Circuits, Vol. 34, No. 3, Mar. 1999, pp. 380–387.
- [6] S. He and M. Torkelson, "Designing pipeline FFT processor for OFDM (de)modulation", in Proc. of IEEE URSI International Symposium on Signals, Systems, and Electronics, Sep. 1998, pp. 257–262.
- [7] J. Bhaskar, "A VHDL Primer", Third Edition, PHI, 2013.