Implementation of High Throughput FFT for Communication

Leenu Mathew Thomas[1], R.Ramya[2]

[1]*Student, Department of ECE, Sree Buddha College of Engineering, Kerala, India.*

[2]*Associate Professor,* Department *of ECE, Sree Buddha College of Engineering, Kerala, India.*

***Abstract*— Most of the wireless standards utilize OFDM technique to enhance the transmission rate which is the vital requirement that is to be met today. The increases in the throughput of FFT block in OFDM system which in turn improves the transmission rate of the system. This paper presents high throughput FFT processor based on Multi-path delay feedback architecture for OFDM systems. Mixed Radix-2/2^2/2^3/2^4 DIF FFT algorithm is used. A comparison is done for different pipeline architecture such as MDF, MDC and SDF using the same algorithm with respect to throughput. The design is fully coded in VHDL, simulated in Xilinx ISE 13.2 and is implemented on Virtex-5 FPGA.**

***Keywords*— OFDM, FFT, FPGA**

I. INTRODUCTION

In recent years the advancement in wireless communication is high. This necessitated the invention of communication system with higher data transmission rate, which is the key factor influencing the performance of communication systems. Transmission rate is directly associated with the system throughput. The demand for very high data rates has motivated the use of OFDM in wireless standards like IEEE 802.11, IEEE 802.16. Orthogonal Frequency Division Multiplexing is a multi-carrier digital transmission scheme that combines modulation and multiplexing. In an OFDM system, the input data sequence is first baseband modulated using modulation schemes like BPSK. The coded data symbols are then parallelised into N sub-streams. Each sub-stream of coded data will then modulate a separate carrier through the Inverse Fast Fourier Transform (IFFT) modulation block. At the receiver the data symbols are obtained from the stream by using a FFT demodulator block. The FFT block is the main block of an OFDM system. The size of FFT depends on the number of subcarriers used in the wireless standard. Hence efficient implementation of FFT processor which have higher throughput is essential which in turn will improve the transmission rate of the system.

Among the various FFT architectures, pipeline architecture is the best choice for wireless system since it can provide high throughput rate with acceptable hardware cost. It is characterized with real-time, continuous processing of the data sequence. In addition to this, pipeline structure is highly regular, which can be easily scaled and parameterized when Hardware Description Language (HDL) is used. Pipeline FFT architecture can be divided into Single-path Delay Feedback [1]-[4] and Multi-path Delay Commutator Architectures [5], [6]. The SDF architecture has reduced hardware complexity and requires less memory, but hardware complexity is more. Contrary to single-path, the multi-path architecture has more number of input or output paths. Multi-path architecture achieves higher parallelism than regular pipeline architecture. MDC architecture can achieve higher throughput. This paper presents implementation of Multi-path Delay Feedback (MDF) architecture which combines SDF and MDC style. It is compared with other architectures based on throughput.

II. MIXED RADIX FFT ALGORITHM

Mixed Radix-2/2^2/2^3/2^4 DIF FFT algorithm is used. The DIF algorithm takes complex number inputs in natural order and outputs in bit-reversed order. Radix-2^4, a high radix algorithm, is the dominant kernel used which can reduce hardware complexity. 32-point FFT can't be realized using radix-2^4 only and so radix 2 algorithm is also used. Likewise, 64-point FFT by radix-2^2/2^4, 128-point FFT by radix-2^3/2^4 and 256-point FFT by radix-2^4/2^4. The mixed radix algorithm provides more flexibility in the selection of data points, regular butterfly structure and can be easily implemented.

III. MDF FFT ARCHITECTURE

Fig.1 shows the block diagram of 256-point MDF architecture. The MDF architecture based FFT processor can deal with 1-2 simultaneous data sequence. The architecture includes 2 types of butterfly units BF1 and BF2, ROM-based multiplier, CSD1 multiplier, CSD2 multiplier, Shift registers and Switch unit. It consists of two paths. The upper path has 8 butterfly stages and lower path has 8 butterfly stages.

A. Butterfly Unit

The BF1 and BF2 structure is shown in Fig.2 and Fig 3 respectively. The butterfly units does the calculation of the sum and difference of the 2 values given as input. In addition, BF2 does pre-multiplication of the data with −j when needed. It uses a simple swapping and negating logic of the real and imaginary parts of the input. Both units are controlled by signals for either storing the input into registers or performing the computation. Both butterflies share a control signal s, which controls four multiplexers. The signal s is driven by a bit in the counter output. This signal switches the butterfly between two different modes. When s is zero, the butterfly is in passing mode. When in passing mode, x1 and x2 are passed to z2 and z1 respectively. When s is one, the butterfly is in computation mode. In this mode, the butterfly operation is performed and the results are sent to the output signals.

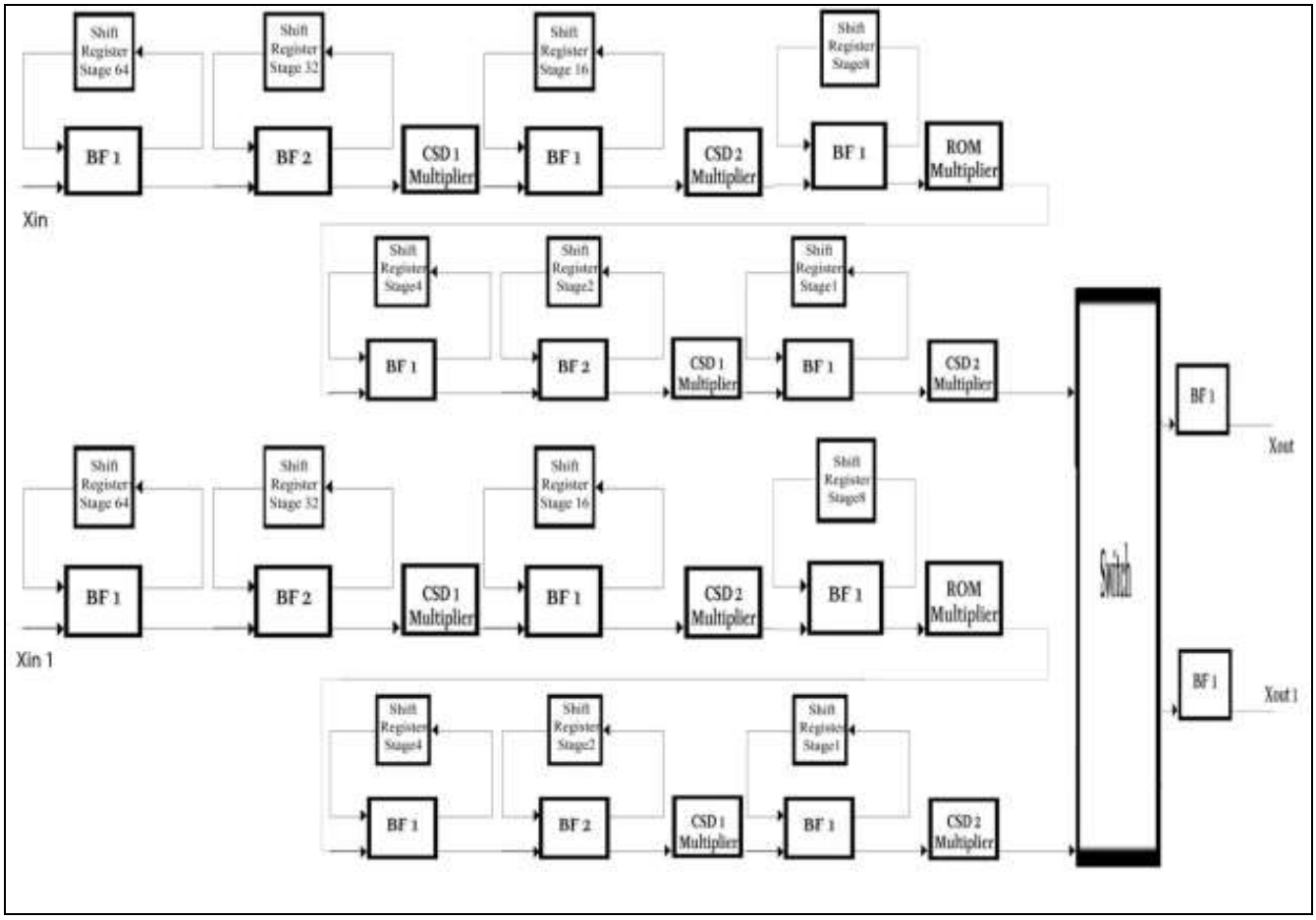

Fig.1 256-point MDF Architecture.

Compared with BF1, BF2 has another control signal. The control signal t is used in BF2 to add some extra functionality to the butterfly when multiplication by –j is required.

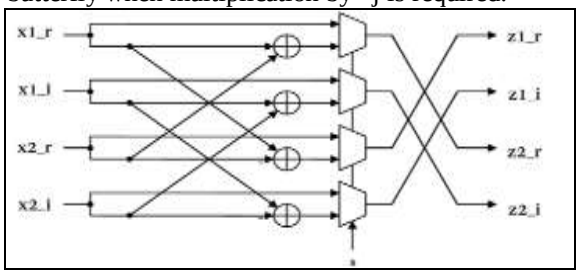

Fig.2 BF1 structure

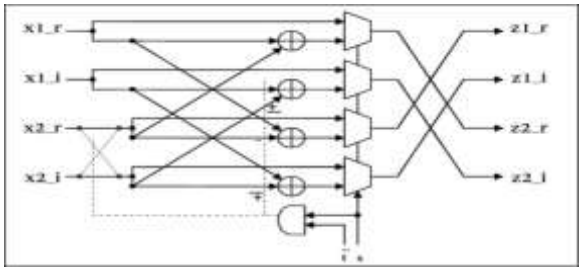

Fig.3 BF2 structure

B. CSD Multiplier

The FFT architecture needs fixed-coefficient multiplications in the case of radix-2^4 algorithm. If fixed-coefficient multiplications is realised with general multipliers, then extra costs is to be paid for chip area and power consumption. This means that implementing a constant multiplier will be enough to eliminate the need for the whole complex multipliers and the ROM to store the twiddle factor coefficients. The multiplication with a constant can be carried out by shifting and adding operation. To reduce the area and power consumption, the constant coefficient can be encoded such that it contains the fewest number of nonzero bits, which can be accomplished using CSD representation. Canonic signed digit is based on a ternary number system (1, 0, -1). It has 2 properties. One is the number of non zero digit is minimal and the other is no two consecutive digits are non-zero.

IV. Results and Discussion

The simulation result for 32, 64,128 and 256 point FFT are shown in Fig.4, Fig.5, Fig.6, Fig.7 respectively.

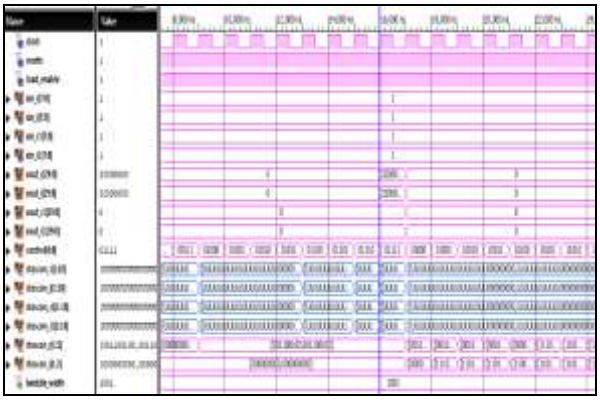

Fig.4 Simulation Result for 32-point MDF architecture

Fig.5 Simulation Result for 64-point MDF architecture

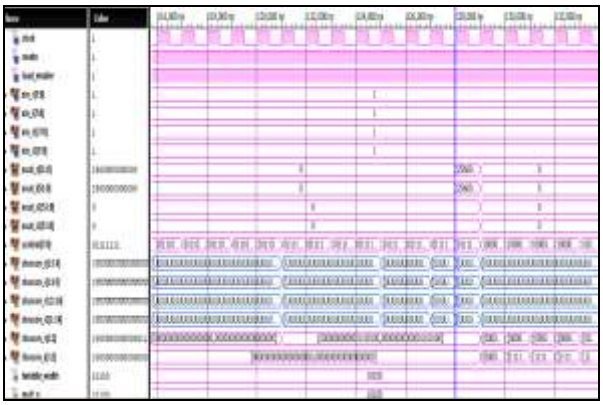

Fig.6 Simulation Result for 128-point MDF architecture

Fig.7 Simulation Result for 256-point MDF architecture

For a 256-point FFT, inputs xin(0) ,xin(2).....xin(254) are given to the upper path and xin(1), xin(3), ..., xin(255) are given to lower path. An 8 bit counter controls the entire structure. There is 8 stages of butterfly unit in each of the parallel path. The first set of input sample x(0) is fed to upper path and x(1) is fed to the lower path in the first clock cycle. In the second clock cycle x(2) and x(3) is fed . Then in consecutive clock cycle rest of the samples are given and the last set of samples x(254) and x(255) is given at 127th clock cycle. The starting clock cycle at which output is obtained from each butterfly unit is as follows: For first butterfly unit at 65th clock cycle, For second butterfly unit at 97th clock cycle, For third butterfly unit at 113th clock cycle, For fourth butterfly unit at 121th clock cycle, For fifth butterfly unit at 125th clock cycle, For sixth butterfly unit at 127th clock cycle, For seventh butterfly unit at 128th clock cycle, For eighth butterfly unit at 128th clock cycle. The first output sample is obtained at 128th clock cycle and rest of the output samples are obtained at consecutive clock cycle and last output sample is obtained at 255th clock cycle.

Fig.8 shows the implementation setup of 256-point FFT architecture on Virtex-5 board

Fig.8 Implementation of 256-point MDF architecture

Table I shows the comparison of different architectures- SDF, MDC, MDF.

TABLE I
COMPARISON OF DIFFERENT ARCHITECTURES

Architecture	FFT size	Min. Period (ns)	Frequency (MHz)	Throughput (MS/s)
SDF	32	27.18	36.79	36.79
	64	28.58	34.99	34.99
	128	36.07	27.72	27.72
	256	43.32	23.08	23.08
MDC	32	22.21	45.03	90.05
	64	25.04	39.94	79.88
	128	28.83	34.68	69.36
	256	35.58	28.11	56.22
MDF	32	18.9	52.9	105.66
	64	20.43	48.94	97.88
	128	27.08	36.92	73.84
	256	34.65	28.86	57.72

Fig.9 shows the comparison of different architecture based on throughput.

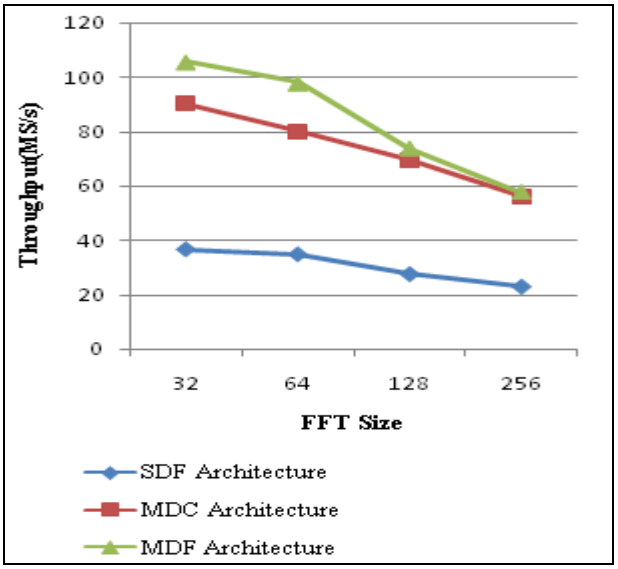

Fig.9 Comparison of different architectures

V. Conclusion

The Fast Fourier Transform is an inevitable module in OFDM systems, so it is to be designed in efficient way. The MDF architecture have high throughput than SDF and MDC architecture and it is found to meet the requirement of wireless standards. The architectures are fully coded in VHDL language. The simulation in done in ISim and a netlist is generated for FPGA configuration and is implemented in Virtex 5 FPGA board.

Acknowledgment

Author wish to remark the great task carried out by the Xilinx and Virtex-5 user guide; and the author wish to thank Prof. Somi Sebastian and R. Ramya for their contribution in the design process.

References

[1] Shousheng He and Mats Torkelson, "Designing Pipeline FFT Processor for OFDM (de)Modulation", *IEEE International Symposium on Signals, Systems, and Electronics*, pp.257– 262, 1998.

[2] C. Fan, M. Lee, G. Su, "Efficient low multiplier cost 256-point FFT design with radix 2^4 SDF architecture", *IEEE Asia Pacific Conference on Circuits and Systems*, pp.1935–1938, 2006.

[3] M.J.S Rangachar, "VLSI Implementation and Performance Analysis of Efficient Mixed-radix 8-2 FFT Algorithm with Bit Reversal for the Output Sequences", *Journal of Theoretical and Applied Information Technology* 2010.

[4] C.Vennila, G.Lakshminarayanan, Seok-Bum Ko, "Dynamic Partial Reconfigurable FFT for OFDM based Communication Systems", *Springer Circuit System Signal processing*, p 1-18, October 2011.

[5] Y.-N. Chang, "An efficient VLSI architecture for normal I/O order pipeline FFT design", *IEEE Trans. Circuits Syst. II, Exp. Briefs*, Vol. 55, No. 12, pp. 1234-1238, Dec. 2008.

[6] Yang K.J, and Chuang G.C.H, "A MDC FFT Processor with Variable Length for MIMO-OFDM", IEEE *Transactions on VLSI systems*, Vol. 21, pp.720-731, 2013.