Journal of Engineering and Applied Sciences 12 (Special Issue 12): 9430-9435, 2017
ISSN: 1816-949X

Design and Implementation of Memory-Based Pipelined FFT Architecture for Complex-Valued Signals

[1]Amit Kumar, [2]Adesh Kumar, [3]Manish Pandey and [4]Aakanksha Devrari
[1]Department of Electronic and Communication Engineering,
Uttarakhand Technical University (UTU), Dehradun, India
[2]Department of Electronics Instrumentation and Control Engineering,
University of Petroleum and Energy Studies (UPES), Dehradun, India
[3]Doon Institute of Engineering and Technology, Rishikesh, India
[4]Department of Eelectronic and Communication, Women Institute of Technology (WIT),
Uttarakhand Technical University (UTU), Dehradun, India

Abstract: This research study proposes a novel architecture for a memory based pipelined fast fourier transform for a complex-valued signals which is based on radix-2 Decimation-In-Frequency (DIF) algorithm. A method of stage partition for a Complex Fast Fourier Transform (CFFT) to minimize the computation clock cycles and maximize the utilization of the Processing Element (PE) is proposed. In addition to this the CFFT architecture can also support more PEs in two dimensions as well. As compared to the previous research, the suggested CFFT processors have fewer computation cycles and lower hardware usage. ISE Xilinx 12.2 tool in verilog language has been utilized for designing and simulation.

Key words: Orthogonal Frequency Division Multiplexing (OFDM), Fast Fourier Transform (FFT), Decimation in Frequency (DIF), Decimation in Time (DIT), Complex FFT (CFFT), Processing Element (PE)

INTRODUCTION

In order to calculate the discrete fourier transform of an input fast fourier transform algorithm can be used. In this process, the input signal is transformed from its domain into frequency domain and vice versa. This can be calculated by factorizing the matrix into a product of sparse factors. Consequently, it reduces the complexity of calculating the DFT from $O(n^2)$ to (nlogn) where n is the data size which ascends if one just applies the definition of DFT (Sorensen *et al*., 1987). FFT is widely used in digital signal processing applications such as in speech processing, image processing, biomedical signal processing, etc.

In Orthogonal Frequency Division Multiplexing (OFDM) technique, the Fast Fourier Transform (FFT) algorithm plays a major role. During the advent of technology, various FFT architectures have been proposed which are commonly divided into two types, namely pipelined architectures and Memory-Based (MB) architectures (Karthik and Kumaran, 2014; Nussbaumer, 1981). The main standard for architecture choice is the interchange between speed and hardware overhead. For improving the performance of memory-based architectures, the resulting problems are often considered.

Long computation cycles: $N/r \times \log_r N$ cycles are required to calculate an N-point FFT with a single radix-r Butterfly Unit (BU).

Memory access bottleneck: $2N \times \log_r N$ memory read/write entrees are desired for FFT computation pretentious a single BU implementation.

FFT has been developed in the field of real-time digital signal processing (Duhamel and Vetterli, 1987). Many algorithms were projected, following the essential idea of decimation, either in terms of frequency or in time-domain. The real valued sequences which are important in real-time signal processing, exhibits the conjugate symmetry which gives rise to redundancies. This could be used to reduce the arithmetic complexity as well as the memory requirements.

In this brief, a novel memory-based pipelined architecture which computes the complex FFT which is based on the radix-2 Decimation-In-Frequency (DIF) has been proposed (Sekhar and Prabhu, 1999). The improved algorithm requires the small number of operations for radix-2 by calculating only half of the samples (output). The modified algorithm separates the data into two parts one is real and the other is imaginary components with real data paths in a regular flow graph.

Corresponding Author: Amit Kumar, Department of Electronic and Communication Engineering,
Uttarakhand Technical University (UTU), Dehradun, India

MATERIALS AND METHODS

Strategy of RFFT stage partition: The N-point DFT of a sequence is defined as:

$$x(k)=\sum_{n=0}^{N-1} x(n) W_N^{kn}$$

Where:

$$W_N^{kn} = e^{-j\left(\frac{2\pi}{N}\right)nk}, k = 0, 1, 2, ..., N-1$$

The computation cycles for an N-point RFFT can be designed by $N/4*\log_2 N$, for the processor with only one PE. A new strategy of the stage partition is presented (Gnanishivaram and Neeraja, 2014; Garrido *et al*., 2009) to additionally reduce the RFFT computation cycles. The multiplication and addition together appear in all the stages of the RFFT, excluding the last two stages. The multiplication can be unnoticed when the twiddle factor is W_0 for instance W_0 is a real value of one. At that moment, the No. of the complex multiplication and complex addition essential in all the stages, except the last two stages can be measured as N/4 and N/2, respectively.

Thus, the PE working in the given architecture comprises one CM and two CAs that can practice four inputs in parallel (Ayinala *et al*., 2013). Related with the traditional strategy, there occurs only one real addition in the last stage and extra multiplications and additions are administered in the first two stages. The requisite computational cycle for N-point RFFT is only $(N/4* (\log_2 N-1)+1)$ in the proposed processor with one PE. It is understandable that the proposed RFFT architecture centered on the fresh strategy of the RFFT stage partition attains smaller amount of computation cycles as compared with the traditional one. Figure 1 and 2 display the high-level architecture of the offered RFFT processor with one PE (Chi and Lai, 2005; Ma *et al*., 2011). Four memory banks are used to store the samples and intermediate data in the processor and each memory bank can store N/4 words of data length W. As a result, the total ability of the required memory is N*W bit.

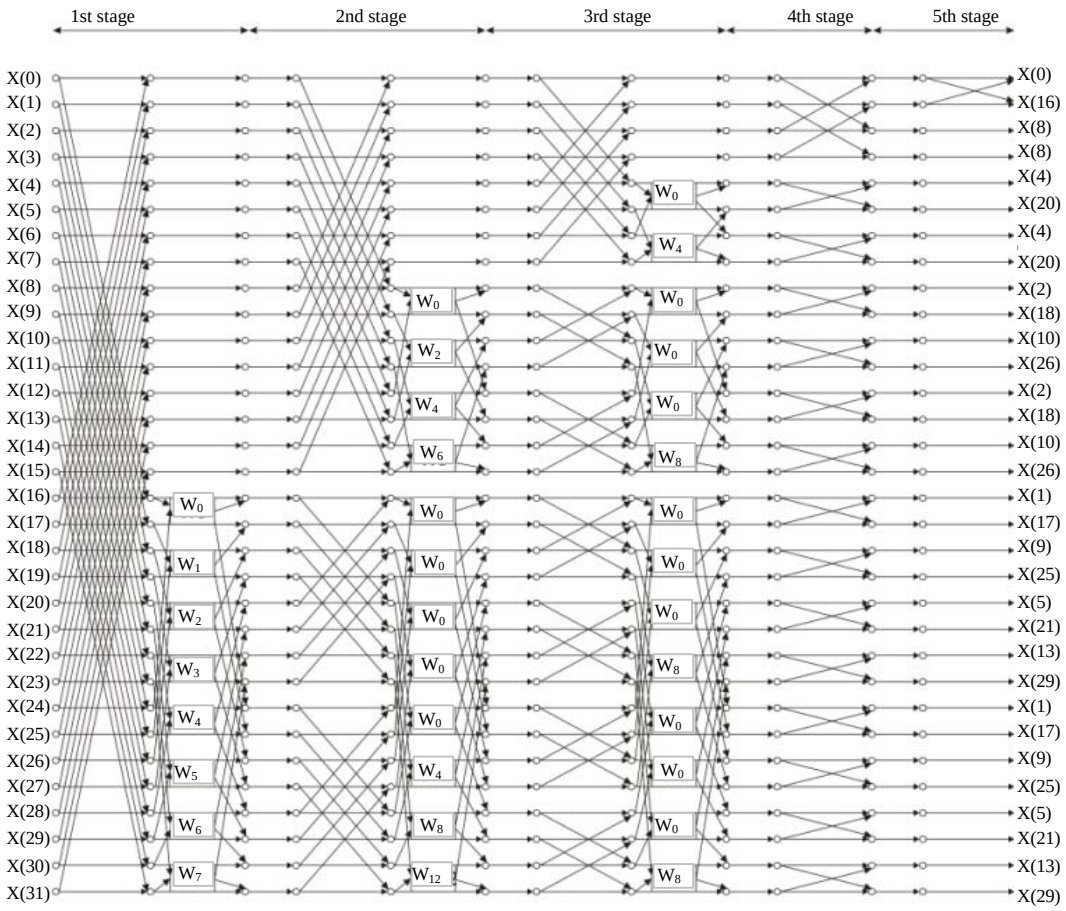

Fig. 1: Stage partition for 32 point RFFT

Here, the memory provisions dual-port access where the data can be read and written at the identical clock cycle. The four multiplexers known as "m_0" will choose the input data path, when the input samples are addressed to the memory banks. Firstly all the samples are put away, the four multiplexers named "m_0" will select additional input data path for the transitional results. The four multiplexers represented as "m_{11}-m_{14}" choose the input sequence of memory bank whereas the four multiplexers represented as "m_{21}-m_{24}" determines which memory banks the intermediate results are written to. The number in the multiplexers stipulates the number of the memory bank. The hardware usage and logic design is recognized such that the output shares the similar data path with the read data path (Jo and Sunwoo, 2005; Duhamel and Vetterli, 1987). During the reading of the result, the reordering work could be completed at the equal time and the next frame of N-point samples could be entering all together. In the offered RFFT architecture, a new PE that can route four samples in parallel is offered by using two butterflies and two multiplexers as shown in Fig. 3. This selection of PE leads to less complexity and high output. The multiplexers termed as "m_{31} and m_{32}" are used together to select whether the multiplier is bypassed. PE consists of four input ports regardedas "in 0-3" and four output ports regarded as "out0-out3". Two real-valued data from some of two memory banks and the twiddle factor from the ROM are multiplied in the CM. The traditional PE that contains two butterflies and six multiplexers is displayed in Fig. 1 where the switch can be deliberated as two multiplexers (Tsai and Lin, 2011). As compared with the customary PE, the offered PE attains lower hardware usage and complexity.

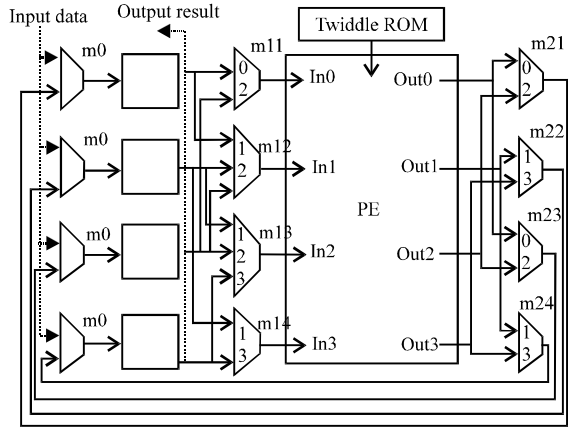

Fig. 2: High-level RFFT architecture with one PE

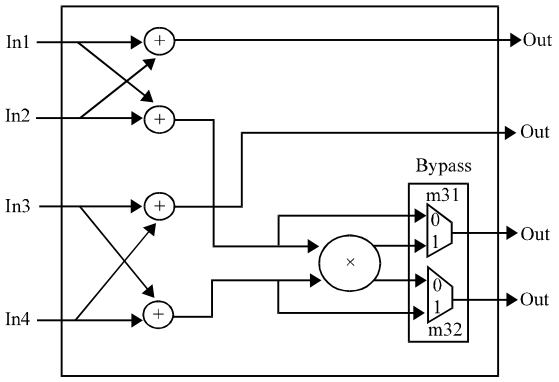

Fig. 3: New PE for RFFT

RESULTS AND DISCUSSION

With the use of the memory based FFT architectures described in the previous section. As compare to the prior works, the modified work is done with the help of three different specifications. Firstly, complex valued signals are used as the input sequences. This sequences are applied to the radix-2 DIF FFT.

Secondly, the pipelined architecture is used. A pipeline is an established of data processing elements which is connected in series, so that, the output of any one section acts as an input of the following one. Mostly in such cases a pipeline is generated by allotting a complex operation into modest operations. In the proposed research, we are implementing the pipelined structure in adders and multipliers. So, computational speed of adder and multiplier is increased and optimized.

Lastly, the floating point IEEE 754 modules are used. The floating point units and two algorithms for the adder and multiplier are utilized. The results of modified work means the pipelined FFT architecture which is memory based for a complex valued signals on radix-2 DIF algorithm is given in Fig. 4 and 5 where Fig. 4a, b represents the RTL structure and Fig. 5 represents the simulation result. The detailed design statistics of the novel memory based pipelined FFT architecture for the complex valued signals which is based on the radix-2 DIF algorithm is given in Table 1 while Table 2 describes the structure of hardware (Table 3-5).

Table 1: Comparison of computational complexity for direct computation of DFT versus FFT algorithm

No. of N	Complex multiplication in direct computation (N^2)	Complex multiplication in FFT algorithm (N/2log2N)	Speed improvement factor
4	16 points	4	4.0
8	64	12	5.3
16	256	32	8.0
32	1024	80	12.8
64	4096	192	21.3
128	16384	448	36.6
256	65536	1024	64.0
512	262144	2304	113.8
1024	1048576	5120	204.8

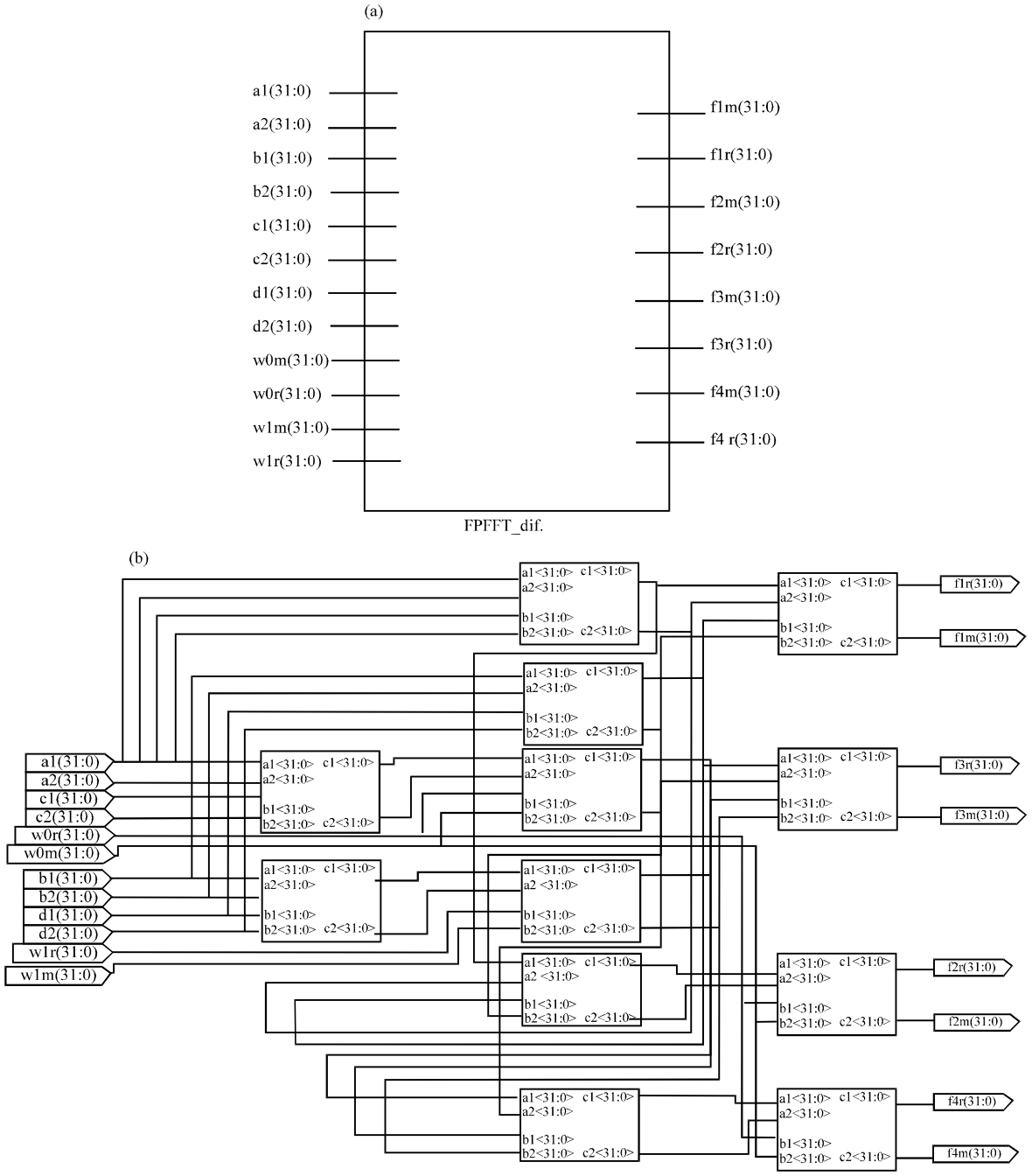

Fig. 4: a, b) RTL schematic of modified work

Table 2: Detail of pin description of RTL schematic

Pin	Descriptions
a1(31:0), a2(31:0), ... , d1(31:0), d2(31:0)	Real and imaginary input value
W0 m(31:0), W1 m(31:0)	Input twiddle factor imaginary value
W0 r(31:0), W1r(31:0)	Input twiddle factor real value
F1 m(31:0), f1r(31:0), ... , F4 m(31:0), f4r()31:0	Real and imaginary outputs

Table 3: Details of input and output values in decimal form

Inputs values	Value of twiddle factor	Output values
a1 = -2.0	W0r = 1.0	f1r = 2.0
a2 = 0.0	W0m = 0.0	f1m = -3.14
b1 = 1.0	W1r = 0.0	f2r = 2.0
b2 = -0.707	W1m = -1.0	f2m = -0.585
c1 = 0.0	-	f3r = 0.2
c2 = -2.0	-	f3m = 0.585
d1 = -0.707	-	f4r = 2.0
d2 = -0.707	-	f4m = 3.414

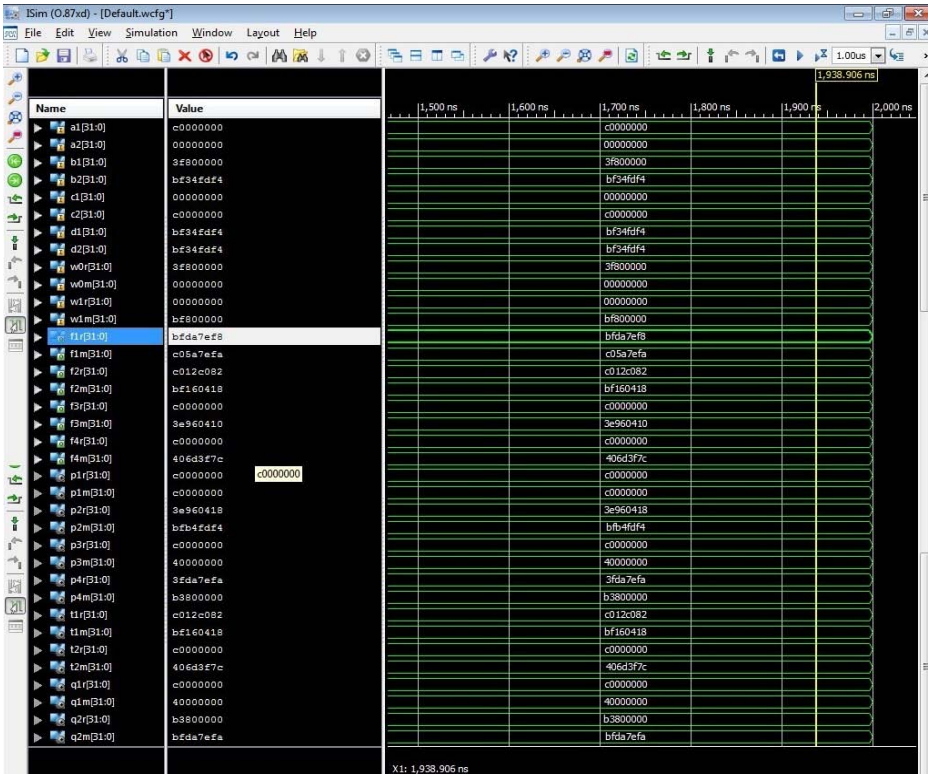

Fig. 5: Simulation results

Table 4: Details of design statistics

Logic utilization	Used	Available	Utilization (%)
No. of slices	25074	204000	12
No. of LUTs	3574	25074	14.25
No. of bonded IOBs	64	600	10.6
No. of DSP48E1s	56	1120	5

Table 5: Hardware description

Variables	Values
Multipliers (total 28)	
24×24 bit multiplier	28
Adders/Subtractors (total 804)	
23 bit adder	28
25 bit adder	24
25 bit addsub	24
8 bit adder	48
8 bit subtractor	624
9 bit adder carry in	28
Comparators (total 48)	
8 bit comparator equal	24
8 bit comparator greater	24
Multiplexers (total 1524)	
1 bit 2-1 multiplexer	52
23 bit 2-1 multiplexer	56
24 bit 2-1 multiplexer	720
25 bit 2-1 multiplexer	24
32 bit 2-1 multiplexer	84
8 bit 2-1 multiplexer	648
Logic shifters (total 48)	
24 bit shifter logical right	48
XORs (total 100)	
1 bit XOR2	100

CONCLUSION

In this memory based architecture for the real valued signals concentrated on radix-2 decimation in frequency algorithm is divided into different steps, i.e., firstly, the inputs are distributed among the real values and imaginary values then apply these real values into the architecture and then check the computation cycles and the hardware complexity.

Here, a novel memory based pipelined FFT architecture for the complex valued signals based on radix-2 Decimation in Frequency (DIF) algorithm, the complex values are applied to the architecture developed for RFFT, although, pipelining is used. Also, for the complex inputs the floating point unit is used (IEEE Standard 754). Then note down the computation cycles and the hardware complexity. This study give a perfect picture of detailed description of computation cycles and hardware description of the modified research.

IMPLEMENTATION

The results obtained after the implementation of the proposed processors with ISE Xilinx12.2 tool. The floating-point samples and twiddle factors are stored in the block RAMs.

RECOMMENDATIONS

In future this research may be done with the help of multiple PE's and for upper radix number FFT's. Also, we can implement pipelining in the algorithms of FFT computation, so that, the speed of computation can be increased. Also, pipelining is used in the memory units which reduces the memory consumption.

REFERENCES

Ayinala, M., Y. Lao and K.K. Parhi, 2013. An in-place FFT architecture for real-valued signals. IEEE. Trans. Circuits Syst. II Express Briefs, 60: 652-656.

Chi, H.F. and Z.H. Lai, 2005. A cost-effective memory-based real-valued FFT and Hermitian symmetric IFFT processor for DMT-based wire-line transmission systems. Proceedings of the 2005 IEEE International Symposium on Circuits and Systems (ISCAS'05), May 23-26, 2005, IEEE, Kobe, Japan isBN: 0-7803-8834-8, pp: 6006-6009.

Duhamel, P. and M. Vetterli, 1987. Improved Fourier and Hartley transform algorithms: Application to cyclic convolution of real data. IEEE. Trans. Acoust. Speech Signal Process., 35: 818-824.

Garrido, M., K.K. Parhi and J. Grajal, 2009. A pipelined FFT architecture for real-valued signals. IEEE. Trans. Circuits Syst. I Regul. Pap., 56: 2634-2643.

Gnanishivaram, K. and S. Neeraja, 2014. FFT-IFFT processor design for 5G MIMO OFDM systems. Intl. J., 3: 54-60.

Jo, B.G. and M.H. Sunwoo, 2005. New Continuous-Flow Mixed-Radix (CFMR) FFT processor using novel in-place strategy. IEEE. Trans. Circuits Syst. I. Regul. Pap., 52: 911-919.

Karthik, P. and G. Kumaran, 2014. Techniques performance analysis and linking to OFDM and MIMO. Asian J. Inf. Technol., 13: 395-400.

Ma, Z.G., F. Yu, R.F. Ge and Z.K. Wang, 2011. An efficient radix-2 fast Fourier transform processor with ganged butterfly engines on field programmable gate arrays. J. Zhejiang Univ. Sci. C, 12: 323-329.

Nussbaumer, H.J., 1981. Fast Fourier Transform and Convolution Algorithms. 2nd Edn., Springer-Verlag, Berlin, Germany isBN-13: 9780387101590, Pages: 248.

Sekhar, B.R. and K.M.M. Prabhu, 1999. Radix-2 decimation-in-frequency algorithm for the computation of the real-valued FFT. IEEE. Trans. Signal Process., 47: 1181-1184.

Sorensen, H.V., D. Jones, M. Heideman and C. Burrus, 1987. Real-valued fast Fourier transform algorithms. IEEE. Trans. Acoust. Speech Signal Process., 35: 849-863.

Tsai, P.Y. and C.Y. Lin, 2011. A generalized conflict-free memory addressing scheme for continuous-flow parallel-processing FFT processors with rescheduling. IEEE. Trans. Very Large Scale Integr. Syst., 19: 2290-2302.