Reconfigurable MDC Architecture Based FFT Processor

Srivatsava PSV [1], Sarada V[2]
Department of Electronics and Communication Engineering
SRM University,Chennai,India.

***Abstract* - Fast Fourier Transform (FFT) is the crucial block in MIMO-OFDM systems. Multipath Delay Commutator based pipelined architecture is used to for variable length MIMO-OFDM systems by putting Ns data streams at the input. By implementing 4 stream data fixing radix-4 at each pipelined stage with variable lengths of 16, 32, 64 and 128 respectively. To emphasize those, needs only one butterfly at each pipeline stage. So the input data stream needs to be scheduled properly before sending it to the processor. They can be used for major communication systems like Wi-Max, LTE etc..By implementing this MDC processor practically with multiple input multiple outputs data throughput can be increased and canbe achieved some major constraints like area and power expected to be decreased.**

***IndexTerms*—Fast Fourier Transform (FFT), Multiple input multiple output (MIMO), Orthogonal frequency division multiplexing (OFDM), pipeline multipath delaycommutator (MDC).**

1. INTRODUCTION

Discrete Fourier Transform (DFT) is used in many communication and signal processing systems[1]. But DFT is difficult to implement due to its computational complexity. So Fast Fourier Transform (FFT) is used to reduce the hard ware complexity. In the field of communications FFT has got wide range of applications in Orthogonal Frequency Division Multiplexing(OFDM) systems like Wi-Max or 3GPP long term evolution(LTE) also wireless modems etc.[2][3]. In OFDM systems Inverse Fast Fourier Transform(IFFT) converts modulated signals from frequency domain to time domain where as Fast Fourier Transform(FFT) is an vice versa. Use of multiple inputs multiple outputs for the OFDM devices data throughput can be increased drastically. In order to manage this we need to design a specific processor which can handle the complexity as the number of input paths increases.

In major of the FFT/IFFT processors pipelined type architecture is used in order to achieve good results. One among the pipelined architecture is continuous flow mixed radix algorithm (CFMR). Continuous Flow Mixed Radix (CFMR) FFT processor that uses the mixed radix algorithm. This algorithm supports is an continuous flow of FFT inputs irrespective of the length. CFMR FFT uses two N-sample memories to create a continuous output data path[4]. One of the memories is used to calculate current FFT/IFFT values and the other stores the previously computed results and controls the output flow.

Advantages:

CFMR is used in MIMO systems; the required memory is increased in a trend proportional to 2xN$_{s.}$ Hardware reduction method such as complex multiplier rescheduling.Flexibility in terms of arithmetic complexity, memory sizes and computation cycles when compared with conventional methods.

Disadvantages:

Complexity of the CFMR becomes more as the input length i.e., N$_s$becomesmore. Because of this memory area also becomes more.

Apart from in place memory scheduling the other two pipelined architectures that are mostly used are Single path Delay Feedback (SDF) architecture and Multipath Delay Commutator(MDC) architecture. SDF architecture uses feedback path to generate intermediate values in order to get output without delay [1]. The difference between the single-path delay feedback (SDF) and the multiple-path delay commutator (MDC) architecture is the approach for data buffering. The multiple-path delay commutator (MDC) pipeline architecture adopts the DC approach [5] for data buffering. By using this approach, intermediate data are directly output to the next stage or coefficient multiplier instead of being written back in the MDC architecture. The input sequence has been break into r (depend on the radix-r to be used) parallel data streams flowing forward with correct ordering for the data elements entering the butterfly unit by proper delays. The normal MDC structure radix-r butterfly will be in idle state until the rthinput is received to the butterfly. So because of this 1/rth part of memory can be used. To overcome this, fix radix value should be the same as data input stream value. Considering a 4 path data stream so fix the value of butterfly unit as radix-4 [6]. At each stage of FFT processor we use only one butterfly unit so that the memory utilization will be 100%. So by this throughput and memory utilization can be increased dramatically.

2. FFT ARCHITECTURE

The N-point FFT equation can be expressed as

$$X[k] = FFT\{x[n]\} = \sum_{n=0}^{N-1} x[n] W_N^{nk} \quad (1)$$

Where $W_N^{nk} = \cos(2\prod nk/N) - j\sin(2\prod nk/N)$
(2)
is the twiddle factor.

This paper mainly is an application of LTE and Wi-Max which requires four different lengths 16, 32, 64 and 128. For larger communications systems the length of the input will be much higher of 2048, 1024,512 and 128 respectively in which the last three stages shares the same hardware.

The decomposition of the above lengths can be given as shown in Table1.

128 point decomposition	4x4x8
512 point decomposition	4x4x4x8
1024 point decomposition	4x4x4x4x4
512 point decomposition	4x4x4x4x8

Table.1.Decomposition of four different FFT/IFFT Lengths

Based on the decomposition shown in Table.1 the equation can be expressed as [1].

$$X[K]=\sum_{n5=0}^{7}\{\sum_{n4=0}^{3}\{\sum_{n3=0}^{3}\{\sum_{n2=0}^{3}\{\sum_{n1=0}^{3}X[N]W_4^{n1k1}W_{2048}^{N1k1}\}$$

$$*W_4^{n2k2}W_{512}^{N2k2}\}W_4^{n3k3}W_4^{N3k3}\}W_4^{n4k4}W_4^{n5k4}\}W_8^{n5k5}$$

(3)

Where K = k1 + 4k2 + 16k3 + 64k4 + 256k5, k1 = 0 ~3, k2 = 0 ~3, k3 = 0 ~3, k4 = 0 ~3, k5 = 0 ~7, and N = 512n1 + 128n2 + 32n3 + 8n4 + n5, N1 = 128n2 + 32n3 +8n4 +n5, N2 = 32n3 +8n4 +n5, and N3 = 8n4 +n5.

Each brace includes computations of a radix-4 butterfly and a twiddle factor multiplication. For non-power-of-4 FFT/IFFT, a radix-8 butterfly is placed at the last stage.

Hence, the last stage is configurable for both radix-4 and radix-8 computation. The proposed radix-4/radix-8 butterfly for the last stage is shown in Fig. 1.where a radix-8 butterfly has the data path indicated by both solid and dashed lines, whereas a radix-4 butterfly has the data path indicated by solid lines only. The regularity of the decomposition makes the processor scalable[1].

This means parameterized register-transfer-level source code is highly reusable to extend the number of stages for a large N. By choosing this type of architecture the hardware can be utilized efficiently by sharing of same structure at each stage of the processor.

3. DELAY COMMUTATORARCHITECTURE

Storage elements are the crucial parts in MDC architecture.For MIMO-OFDM systems is inputs are scheduled properly such that memory utilization and also sharing of hardware can be increased from 25% to 100%. This makes processor is much suitable for MIMO-OFDM systems [7]. So, go for memory scheduling principle in order to decrease the memory utilization on the architecture before the inputs has been sent to the processor. The radix-8 can be designed at the last stage for N is a power of 2 with a flexibility of radix-4 and radix-2 combination. The block diagram of such an reconfigurable radix4/8 structure is shown in fig.2.

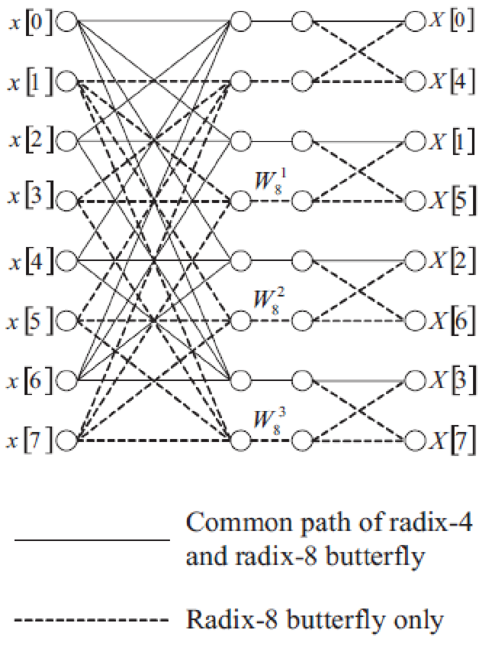


Fig. 1. Signal Flow Graph for Proposed System

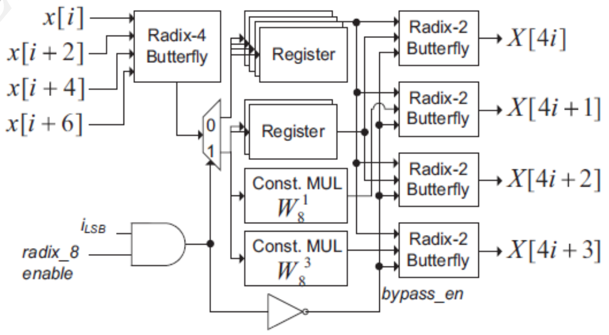


Fig.2.Reconfigurable Radix-4/8 structure.

Depending on the radix_8 enable signal the operation can be processed either it should work as radix-4 or radix-8. The value of i_{LSB} is kept constant as high for entire operation.

3.1. Basic Radix-4 MDC operation

The input sequence has been split into r (depend on the radix-r to be used) parallel data streams flowing forward with correct ordering for the data elements entering the butterfly unit by proper delays. As we are using fixed radix-4 the inputs are divided into 4 parallel streams [7]. The block diagram of N-point radix-4 DIF MDC pipeline FFT processor is shown in Fig 4. From the Fig.4, the elements between PEs consist of shift registers and a commutator switch. The goal of these elements is to form a proper set of data for the next PE. The R4MDC architecture contains 3log$_4$N-3multipliers,

$8\log_4 N$ adders and 2.5N − 4 registers. The block diagram of MDC based architecture for MIMO-OFDM systems is showing in fig.3.

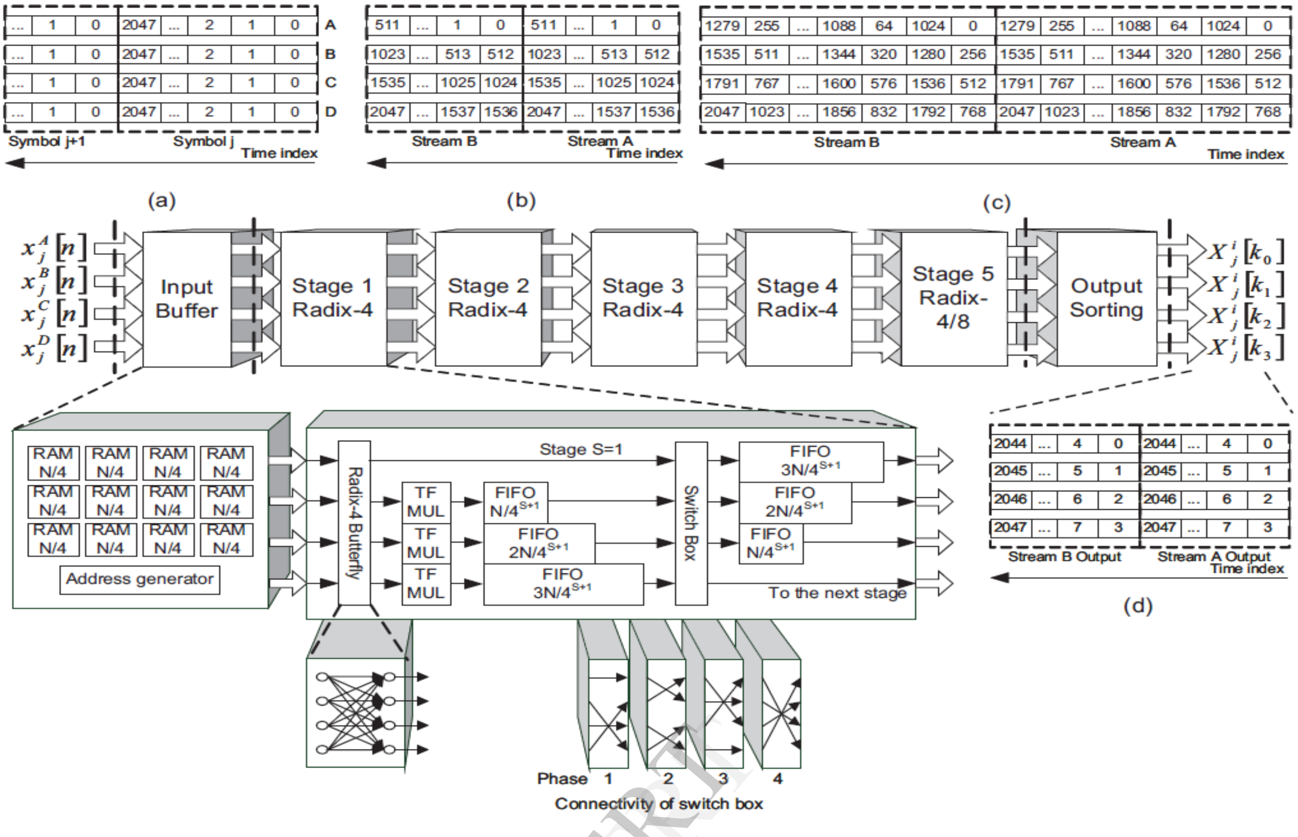


Fig.3 Block diagram of MDC based FFT processor for MIMO-OFDM systems
a) Input stream b) Sorted inputs by memory scheduling c) unsorted outputs from processor d) sorted outputs

3.2. *Input Memory Scheduling*

As shown in fig.3 (a) initially the inputs streams are sent parallel which are four streams. All the four streams are of same length. Here the task is to convert the input streams as fig.3 (a) to fig.3 (b) i.e., the data in parallel is to be converted into serial inputs. In order to make this conversion, make use of memory banks or registers to store the sorted inputs and for the processing of the inputs to be scheduled properly.

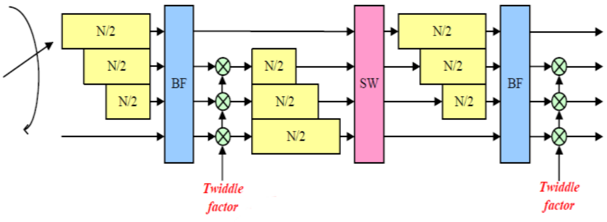


Fig.4.The two stages of N-point R4MDC Architecture

There are totally 12 memory cells to store the 4 input streams similarly remaining streams first ¾ parts each of 1/4th part stored in the 9 memory banks for the B, C and D streams respectively [8],[9]. Each stream divided into four sub parts a,b,c and d. Where a,b, and c are stored in 3 memory banksand fourth sub stream is directly fed to the radix-4 butterfly unit which lies in the first stage of the architecture.

4. COMPLETE ARCHITECTURE OPERATION OF THE MDC STRUCTURE

Thescheduled inputs are stored in the memory cells according to the stream wise. The first stream with scheduled inputs is sent to the stage-1 radix-4 butterfly unit. The corresponding generated outputs from the butterfly unit are sent to the FIFO registers in different clocks.

The FIFO units are fed to the MDC radix-4 switch box structure as shown in fig.3. Later the crisscrossed outputs from the switch box circuit of the stage-1 are passed to the stage-2 butterfly unit structure via second set of FIFO units [9].The last stage of the MDC structure has reconfigurable radix-4/8 structure such that depending on the input value length if the value is powers of 2 then it can be used as 8 if the value is powers of 4 then it can be used as radix-4 butterfly structure.

5. SIMULATION RESULTS AND MODIFICATIONS

The current MDC structure consists of switch box circuits and FIFO units. The architecture can also be implemented using shift registers and multiplexers instead of FIFO and switch box units [11],[12]. The block diagram of such implementation is shown in fig.5. The MDC architecture is implementedusing ModelsimVerilog. The fig.6. shows the simulation results of 16point MDC structure with the input sequence x(n)={0,1,2,3,0,1,2,3,0,1,2,3,0,1,2,3} and the corresponding output.Similarly fig.7.shows the last stage of MDC structure when radix_8 enable signal is high i.e.,1. This means that the circuit operates in the mode of powers of 2 instead of powers of 4. Example, for an 32 point the above selection has to be done to obtain the results.

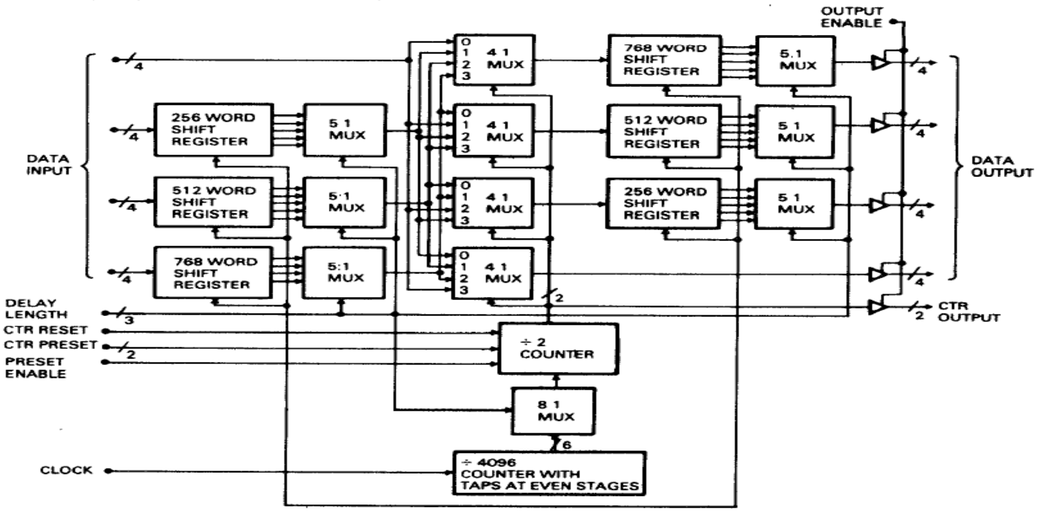
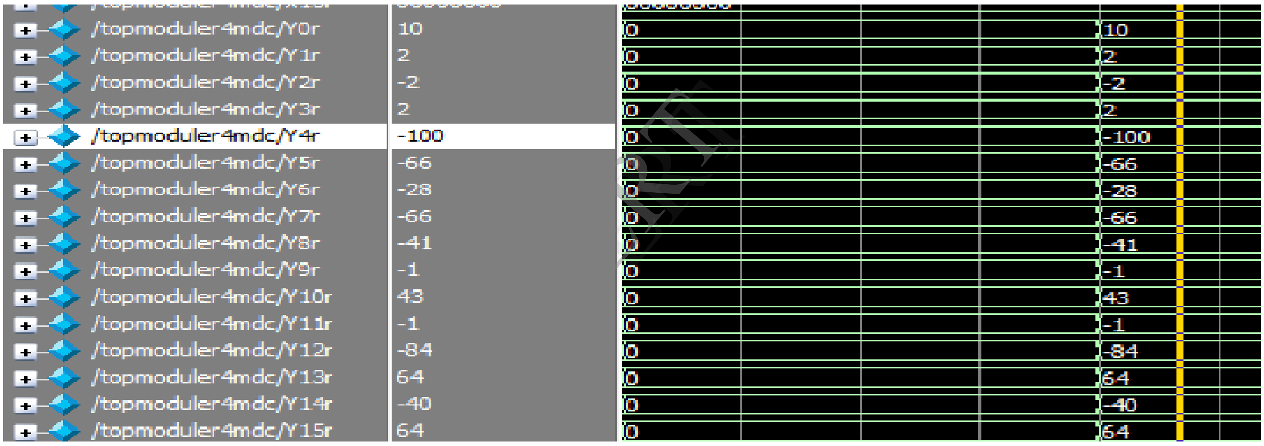
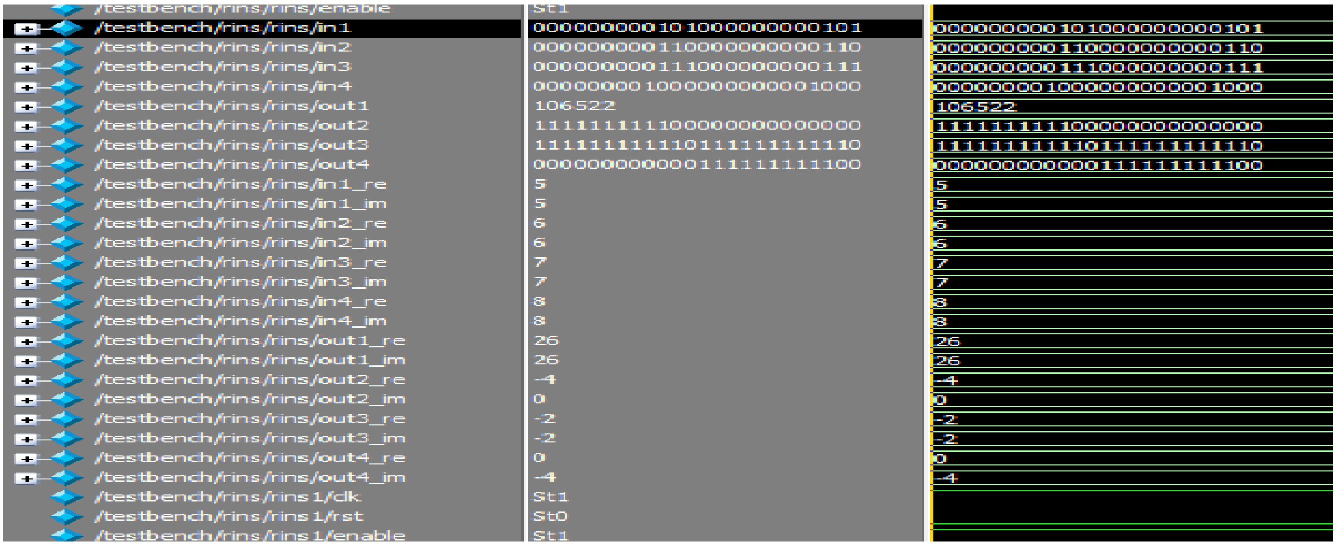


Fig.5.Basicdelaycommuataor block diagram



```
/topmoduler4mdc/Y0r    10     0  10
/topmoduler4mdc/Y1r    2      0  2
/topmoduler4mdc/Y2r    -2     0  -2
/topmoduler4mdc/Y3r    2      0  2
/topmoduler4mdc/Y4r    -100   0  -100
/topmoduler4mdc/Y5r    -66    0  -66
/topmoduler4mdc/Y6r    -28    0  -28
/topmoduler4mdc/Y7r    -66    0  -66
/topmoduler4mdc/Y8r    -41    0  -41
/topmoduler4mdc/Y9r    -1     0  -1
/topmoduler4mdc/Y10r   43     0  43
/topmoduler4mdc/Y11r   -1     0  -1
/topmoduler4mdc/Y12r   -84    0  -84
/topmoduler4mdc/Y13r   64     0  64
/topmoduler4mdc/Y14r   -40    0  -40
/topmoduler4mdc/Y15r   64     0  64
```

Fig.6. Simulated results of 16point MDC architecture output



```
/testbench/rins/rins/enable    St1
/testbench/rins/rins/in1       0000000000101000000000000101   0000000000101000000000000101
/testbench/rins/rins/in2       0000000000110000000000000110   0000000000110000000000000110
/testbench/rins/rins/in3       0000000000111000000000000111   0000000000111000000000000111
/testbench/rins/rins/in4       000000001000000000000001000    000000001000000000000001000
/testbench/rins/rins/out1      106522                         106522
/testbench/rins/rins/out2      1111111111000000000000000000   1111111111000000000000000000
/testbench/rins/rins/out3      1111111111101111111111110      1111111111101111111111110
/testbench/rins/rins/out4      0000000000001111111111100      0000000000001111111111100
/testbench/rins/rins/in1_re    5     5
/testbench/rins/rins/in1_im    5     5
/testbench/rins/rins/in2_re    6     6
/testbench/rins/rins/in2_im    6     6
/testbench/rins/rins/in3_re    7     7
/testbench/rins/rins/in3_im    7     7
/testbench/rins/rins/in4_re    8     8
/testbench/rins/rins/in4_im    8     8
/testbench/rins/rins/out1_re   26    26
/testbench/rins/rins/out1_im   26    26
/testbench/rins/rins/out2_re   -4    -4
/testbench/rins/rins/out2_im   0     0
/testbench/rins/rins/out3_re   -2    -2
/testbench/rins/rins/out3_im   -2    -2
/testbench/rins/rins/out4_re   0     0
/testbench/rins/rins/out4_im   -4    -4
/testbench/rins/rins1/clk      St1
/testbench/rins/rins1/rst      St0
/testbench/rins/rins1/enable   St1
```

Fig.7. Simulated results of butterfly when radix_8 enable is “1”

6. CONCLUSION

For better performance of an MIMO-OFDM the above MDC structured based FFT processor design makes the utilization of the memory cells more efficiently and also the utilization of the hardware can be increased. This type of FFT processor can be used in major of the communication systems like LTE, wireless modems. The power and area of this kind of architecture is less when compared with previous work reports that are available from standard architectures.

7. REFERENCES

1. Kai-JiunYang,Shang-Ho Tsai, Senior Member, IEEE, and Gene C.H.Chuang, Member,IEEE"MDC FFT/IFFT Processor with variable length for MIMO-OFDM systems", IEEE transactions on very large scale integration(VLSI)systems, Vol.21.N0.4,April 2013.
2. Very-High-Bit-Rate Digital Subscriber Line Transceiver 2(VDSL2), ITUTStandard G.993.2, Feb. 2006.
3. The Wireless LAN Media Access Control (MAC) and Physical Layer(PHY) Specifications, IEEE Standard 802.11, 1999
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. S. He and M. Torkelson, "A new approach to pipeline FFT processor," in Proc. IEEE Int.Parallel Process.Symp., Apr. 1996, pp. 766–770.
6. John G. Proakis and DimitrisG.Manolakis "Digital Signal Processing Principles, Algorithms and Applications" Third Edition.
7. Chien Sung-Li "Design and Implementation of Fast Fourier Transforms Processor"
8. Yu-Wei Lin and Chen-Yi Lee, Member, IEEE "Design of an FFT/IFFT Processor for MIMO OFDM Systems", IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS-I: REGULAR PAPERS, VOL. 54, NO. 4, APRIL 2007.
9. P. Y. Tsai and C. Y. Lin, "A generalized conflict-free memory addressingscheme 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.
10. Y.-W. Lin, H.-Y.Liu, and C.-Y. Lee, "A 1-GS/s FFT/IFFT processor for UWB applications," IEEE J. Solid-State Circuits, vol. 40, no. 8, pp. 1726–1735, Aug. 2005.
11. E. E. Swartzlander, W. K. W. Young, and S. J. Joseph, "A radix 4 delay commutator for fast Fourier transform processor implementation," IEEE J. Solid-State Circuits, vol. 19, no. 5, pp. 702–709, Oct. 1984.
12. Chein-Sung Li and Prof. Shuenn-Shyang Wang, "Design and Implementation of VeriableLegth FFT Processors,Tatung University – July 2006".