A Pipelined FFT Architecture to ProcessTwo Independent Data Streams

Mr.MD.HAMEEDPASHA[1] , MRS.P.RADHIKA[2]
1. Associate Professor, Department of ECE ,Jayamukhi Institute of Technogical Sciences , Warangal, India
2. Department of ECE, Jayamukhi Institute of Technological Sciences, Warangal, India.

Abstract—This project proposes the multiple independent FFT computation of two independent data stream is introduced. Multipath delay communicator FFT architecture is the basis of proposed architecture. In time FFT and in frequency FFT it has N/2-point decimation to process the odd and even samples of two data streams separately. The bit reversal operation is performed by the architecture itself is the main feature of the architecture. So without any dedicated bit reversal circuit the outputs are generated in normal order. By interleaving the data the bit reversal operation is performed by the shift registers in the FFT architecture. So high throughput and lower number of register is necessary for the proposed architecture. System throughput is a key factor influencing performance in wireless communication.

Nowadays, many applications require simultaneous computation of multiple independent fast Fourier transform (FFT) operations with their outputs in natural order. Therefore, this brief presents a novel pipelined FFT processor for the FFT computation of two independent data streams. The proposed architecture is based on the multipath delay commutator FFT architecture. It has an *N*/2-point decimation in time FFT and an *N*/2-point decimation in frequency FFT to process the odd and even samples of two data streams separately. The main feature of the architecture is that the bit reversal operation is performed by the architecture itself, so the outputs are generated in normal order without any dedicated bit reversal circuit. The bit reversal operation is performed by the shift registers in the FFT architecture by interleaving the data. Therefore, the proposed architecture requires a lower number of registers and has high throughput.

Keywords: Fast Fourier Transform (FFT), Multiple Input Multiple Output (MIMO), Pipelined FFT.

I. INTRODUCTION

FFT is most commonly used in wireless communication application. Single path delay feedback and multiple path delay commutator are very popular in a family of pipelined FFT architecture .More than one data stream need to be processed in applications such as image processing, multiple input-multiple output OFDM, and array signal processing and so on. In order to generate the outputs in natural order a dedicated bit reversal circuit and simultaneous multiple FFT operations are required. Multiple independent data streams can be handled by FFT architectures. A single FFT processor has the ability to process all the data streams. Four dependent data streams are included, they are processed one by one. At two domains eight data streams are processed. When multiple data streams are processed the output is not achieved as parallel, so more than one FFT processors are used. One to four data streams are processed in wireless area network application by using multiple data path. The proposed architecture is designed to process parallel input data streams continuously with less amount of hardware. The odd and even inputs are in the natural order. The odd samples are processed by N/2 point DIT FFT. The even samples are processed by N/2 point DIF FFT. To generate the outputs of N-point FFT in natural order, two parallel butterflies processed the outputs of the two N/2 point FFTs.

Main Objectives of the project:

FFT has a significant role in the digital signal processing. The FFT computations have high throughput and low latency. High performance FFT circuit must be design to achieve high throughput and latency. System throughput is a key factor influence performance in wireless communication. To increase the transmission rate of the system key research is done. The proposed architecture is designed to processparallel input data streams continuously with less amount ofhardware. The results of different architecture are comparedwith their area, power and delay.

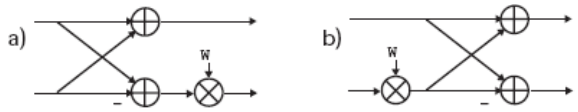
Fast fourier transform:

Direct computation of DFT for a given input signal requires large amounts of computational speed and time. It is also quite complex if the number of inputs is more. To have speedy computation and to perform complex multiplication effectively, the twiddle factors of DFT sequence are decomposed as a matrix and it is used. This technique of transformation is referred to as FFT. Major properties employed in FFT are symmetrical and periodicity property. Various FFT applications are image processing, speech processing, spectrum analysis and OFDM applications

Computation of FFT in conventional method:

The FFT and IFFT has the property that, if FFT(Re(xi)+ jIm(xi)) = Re(Xi)+ jIm(Xi) and IFFT(Re(Xi)+ jIm(Xi)) = Re(xi)+ jIm(xi) where xi and Xi are N words long sequences of complex valued, samples and sub-carriers respectively, then 1/N * FFT(Im(Xi)+ jRe(Xi)) = Im(xi)+ jRe(xi). Therefore, it is necessary not only to discuss the implementation of the FFT equalizer.

To calculate the inverse transform, the real and imaginary part of the input and output are changed. N 1 / n scaling a power of two, so that the right binary word log2 (N) bits in the same switch. Even simpler, binary point left log2 (N) bits is shifted to remember. If ever, did not show up



tochange a bit, depending on how it is used, the output from IFFT, is required.

Figure 1 :A radix-2 DIF butterfly (a) and a radix-2 DIT butterfly (b), where W is the twiddle factor.

FFT algorithm is the basic building block can be realized with a butterfly operation. Frequency (DIF) time, (d) and the decimation of the butterfly in the death of two types of operations, the two are shown in Figure 1. DIF is the difference between the before and after the addition or subtraction and multiplication of featured twiddle factor is in place. FFT based on the division to conquer and due to the input range is N R source, known as the point length N = RP, so, and p positive integer is the most effective.

II. Proposed system

A. A Normal I/O Order Radix-2 Fft Architecture to Process Twin Data Streams for MIMO

It is designed to process parallel input data streams continuously. The odd and even inputs are in the natural order. The odd samples are processed by N/2 point DIT FFT. The even samples are processed by N/2 point DIF FFT. To generate the outputs of N-point FFT in natural order, two parallel butterflies processed the outputs of the two N/2 point FFTs.

The proposed architecture is designed to process two independent data streams simultaneously with less amount of hardware. The odd inputs, which are in natural order, are first bit reversed and then they are processed by *N*/2-point decimation in time (DIT) FFT. The even samples are directly processed by *N*/2-point DIF FFT, so its outputs are in bit reversed order. Therefore, the outputs of *N*/2-point DIF FFT are bit reversed. The outputs of the two *N*/2-point FFTs are further processed by the two-parallel butterflies to generate the outputs of *N*-point FFT in natural order. The bit

reversing is carried out by the scheduling registers, which are actually used to delay the samples for performing the butterfly operations. Thus, the FFT architecture does not use any dedicated circuit to bit reverse the data. As a result, the proposed architecture requires less number of registers than the prior FFT designs.

Proposed pipelined FFT architecture:

The idea of computing an *N*-point FFT using two *N*/2-point FFT operations with additional one stage of butterfly operations is shown in Fig. 2, which is not the exact architecture but provides the methodology. The reordering blocks in Fig. 2 are merely present to state that the *N*/2 odd samples ($x(2n + 1)$) are reordered before the *N*/2-point DIT FFT operation and *N*/2 even samples ($x(2n)$) are reordered after the *N*/2-point DIF FFT operation. In order to compute the *N*-point DIT FFT from the outputs of two *N*/2-point FFTs, additional one stage of butterfly operations are performed on the results of the two FFTs. Thus, the outputs generated by additional butterfly stage are in natural order.

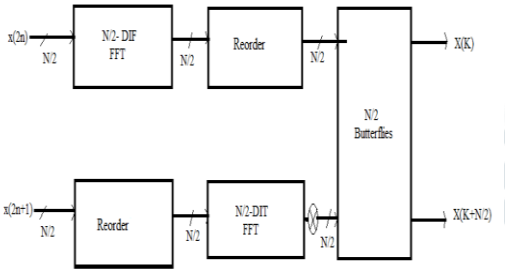


Fig. 2. Idea of the proposed method.

It is designed to process parallel input data streams continuously. The odd and even inputs are in the natural order. The odd samples are processed by N/2 point DIT FFT. The even samples are processed by N/2 point DIF FFT. To generate the outputs of N-point FFT in natural order, two parallel butterflies processed the outputs of the two N/2 point FFTs.The two N/2-point FFT operations with additional one stage of butterfly operations for computing an N-point FFT is shown in Fig 2, it provides the methodology but it is not the exact architecture. To process two data steams using two eight-point MDC FFT. The inputs are received by the RSR unit present at the left side of SW1. The last stage RSR is used store eight-point DIF FFT output and bits reverse them. Butterfly operations done by the bit reversed data in the RSR in the last stage and outputs from the eight-point DIT FFT. The first and second data streams generated FFT outputs of in natural order using upper and lower BF2 in the last stage. In last stage the word length is twice due to the two data paths in the last stage is combined.

Operation of the proposedArchitecture

L1, L2, L3, M1, M2 and M3 are the six levels of FFT architecture in Fig.3. In the levels M2 and M1 respectively the eight-point DIF and DIT FFT operations are performed. With the help of SW2 the data from L2 and M2 can be forwarded to L3 and M3 respectively or vice versa.

Fig.2: Proposed 16-Point Radix-2 Fft Architecture with Outputs In Natural Order

The partially processed even data is reordered by the RSR registers in the levels L3 and M3 and the odd input data is reordered by the RSR registers in the levels L1 and M1.In the levels L2 and M2 respectively the eight-point DIF and DIT FFT operations are performed. With the help of SW2 the data from L2 and M2 can be forwarded to L3 and M3 respectively or vice versa, x(2n) and x(2n+1) are the representation of the two input streams to the FFT processor. In order to swap the data

path and propagate the data to different levels SW1 and SW2 have to switches.

The switches SW1 or SW2 pass the data at u1,u2,u3 and u4 to v3,v4,v1 and v2 respectively during the swap mode and switches SW1 or SW2 pass the data at u1,u2,u3 and u4 to v1,v2,v3 and v4 respectively during the normal mode. During N/2+1to NSW1 is in the normal mode and during the first N/2 clock cycles,SW1 is in the swap mode otherwise during N/2+1 to N SW2 is in the swap mode and during the first N/2 clock cycle SW2 is in the normal mode. Thus SW1 and SW2 change their modes for every N/2 clock cycles and are indifferent modes at any time. The switches (SW1 or SW2) are in the normal mode, if there is transition of data between Ly and Ly+1 or My and My+1(y can be 1 or 2).The switches (SW1 or SW2) are in the swap mode, if there is transition of data between Ly and My+1 or My and Ly+1.The control signals of SW1 and SW2 are given as external. This switch helps the signals to swap in the N/2 clock cycle.

TABLE I
DATA FLOW THROUGH DIFFERENT LEVELS

Level	Time —>						
L_1	$E_1(1,1)$	$O_1(1,1)$	$E_1(1,2)$	$O_1(1,2)$			
L_2		$E_1(2,1)$	$E_2(2,1)$	$E_1(2,2)$	$E_2(2,2)$		
L_3			$E_1(3,1)$	$O_1(3,1)$	$E_1(3,2)$	$O_1(3,2)$	
M_1		$E_2(1,1)$	$O_2(1,1)$	$E_2(1,2)$	$O_2(1,2)$		
M_2			$O_1(2,1)$	$O_2(2,1)$	$O_1(2,2)$	$O_2(2,2)$	
M_3				$E_2(3,1)$	$O_2(3,1)$	$E_2(3,2)$	$O_2(3,2)$

The two input streams to the FFT processor are represented as *X*1 and *X*2. The odd and even samples of two input streams are disassociated by the delay commutator units in *L*1 and *M*1 (*X*1 is disassociated into {*E*1(*i, j*), *O*1(*i, j*)}, respectively, and *X*2 is disassociated into {*E*2(*i, j*), *O*2(*i, j*)}). In these representations, *i*defines the nature of the data and *j* defines the number of the data set whose FFT has to be computed. The even set of input data [*x*(0), *x*(2), *x*(4) . . .] is defined as *E*(1, *j*) and the odd set of input data [*x*(1), *x*(3), *x*(5) . . .] is defined as *O*(1, *j*). *E*(2, *j*)/*O*(2, *j*) is the set of scheduled or ordered even/odd data, which are ready to be fed to eight-point DIF/DIT FFT. The outputs of eight-point DIF/DIT FFT are defined as *E*(3, *j*)/*O*(3, *j*), which are fed to the third level for 16-point FFT computation. Table I explains the operation of FFT and the data propagation through different levels.

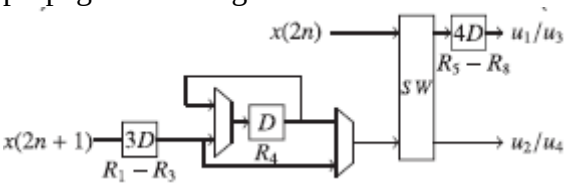


Fig 3. Structure of RSR in delay commutator unit in L1 and M1

1) The first eight samples of *X*1 are loaded into the registers (4*D* in the upper and lower arms of delay commutator unit) in *L*1. After eight clock cycles, the switch (SW1) is set in the normal mode and the first eight samples of *X*2 are loaded into the registers (4*D*) in *M*1. Simultaneously, *E*1*(*1, 1*)* (even samples of *X*1) is forwarded from *L*1 to *L*2 as *E*1*(*2, 1*)* to perform the eight-point FFT operation. The odd samples of *X*1 and *X*2 are bit reversed by the RSR in *L*1 and *L*2, respectively.

2) After eight clock cycles, the positions of the switches SW1 and SW2 are set in the swap mode and the normal mode, respectively. The odd samples (*O*1*(*1, 1*)*) of *X*1 are forwarded from *L*1 to *M*2 as *O*1*(*2, 1*)* and the even samples (*E*2*(*1, 1*)*) of *X*2 is forwarded from *M*1 to *L*2 as *E*2*(*2, 1*)*. Simultaneously, *E*1*(*2, 1*)* is forwarded from *L*2 to *L*3 as *E*1*(*3, 1*)* and reordering is performed.

3) After eight clock cycles, SW1 and SW2 are set in the normal mode and the swap mode, respectively. The odd samples of *X*2(*O*2*(*1, 1*)*) are forwarded from *M*1 to *M*2 as *O*2*(*2, 1*)* and
*O*1*(*2, 1*)* is forwarded from *M*2 as *O*1*(*3, 1*)* to *L*3 where the butterfly operations with *E*1*(*3, 1*)* corresponding to the last stage (of the data stream *X*1) are performed. In the Meantime, *E*2*(*2, 1*)* from *L*2 is forwarded to *M*3 as *E*2*(*3, 1*)* and reordering is performed in the RSR.

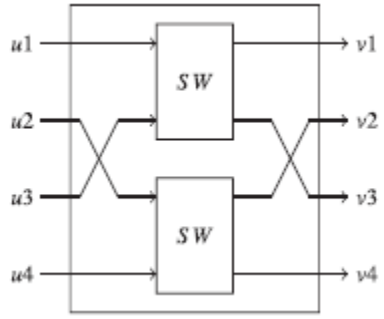


Fig.4. Structure of sw1 and sw2

4) After eight clock cycles, the switch (SW2) is set to normal position to allow the partially processed odd samples ($O2(3, 1)$) from $M2$ to $M3$ and perform the butterfly operations of the last stage (of the data stream $X2$).

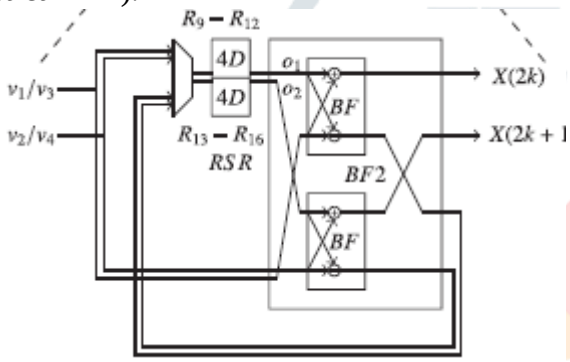


Fig.5. Detailed structure of M3 and L3

III. Simulation results

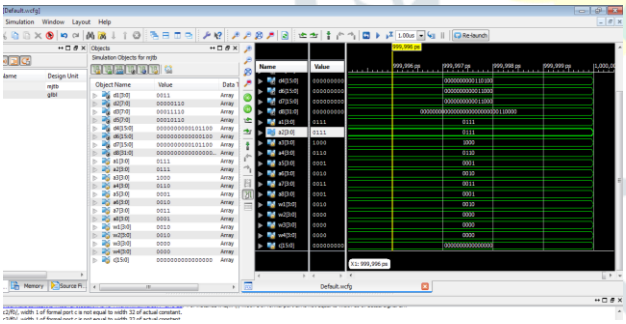


Fig.6. Simulated output

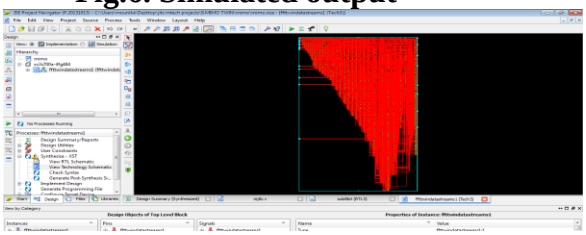


Fig.7. Technology schematic of proposed method

IV CONCLUSION

In this paper, butterfly inputs are used to generate outputs in natural order. The processor can handle two independent data streams simultaneously. The bit reversal circuit present in prior designs is eliminated by using two pipeline architecture. The proposed architecture provides throughput higher than the prior architectures and also reduces the complexity of the circuit. Simultaneously two data streams can be processed by the proposed architecture which has two N-point FFT architectures .So the complexity can be normalized. As a result, the need of additional registers to bit reverse the outputs is avoided.

REFERENCES

[1] S. He and M. Torkelson, "A new approach to pipeline FFT processor," in *Proc. 10th Int. Parallel Process. Symp.*, 1996, pp. 766–770.

[2] Y. Chen, Y.-W. Lin, Y.-C. Tsao, and C.-Y. Lee, "A 2.4-Gsample/s DVFS FFT processor for MIMO OFDM communication systems," *IEEE J. Solid-State Circuits*, vol. 43, no. 5, pp. 1260–1273, May 2008.

[3] S.-N. Tang, C.-H. Liao, and T.-Y. Chang, "An area- and energy efficient multimode FFT processor for WPAN/WLAN/WMAN systems," *IEEE J. Solid-State Circuits*, vol. 47, no. 6, pp. 1419–1435, Jul. 2012.

[4] P. P. Boopal, M. Garrido, and O. Gustafsson, "A reconfigurable FFT architecture for variable-length and multi-streaming OFDM standards," in *Proc. IEEE ISCAS*, May 2013, pp. 2066–2070.

[5] K.-J. Yang, S.-H. Tsai, and G. C. H. Chuang, "MDC FFT/IFFT processor with variable length for MIMO-OFDM systems," *IEEE Trans.Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 4, pp. 720–731, Apr. 2013.

[6] 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.

[7] M. Ayinala, M. Brown, and K. K. Parhi, "Pipelined parallel FFT architectures via folding transformation," *IEEE Trans. VeryLarge Scale Integr. (VLSI) Syst.*, vol. 20, no. 6, pp. 1068–1081,Jun. 2012.

[8] M. Garrido, J. Grajal, and O. Gustafsson, "Optimum circuits for bit reversal," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 10, pp. 657–661, Oct. 2011.
[9] S.-G. Chen, S.-J. Huang, M. Garrido, and S.-J. Jou, "Continuous-flow parallel bit-reversal circuit for MDF and MDC FFT architectures," *IEEETrans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 10, pp. 2869–2877, Oct. 2014.
[10] M. Garrido, J. Grajal, M. A. Sanchez, and O. Gustafsson, "Pipelined radix-2*k* feedforward FFT architectures," *IEEE Trans. Very Large ScaleIntegr. (VLSI) Syst.*, vol. 21, no. 1, pp. 23–32, Jan. 2013.
[11] Z. Wang, X. Liu, B. He, and F. Yu, "A combined SDC-SDF architecture for normal I/O pipelined radix-2 FFT," *IEEE Trans. Very Large ScaleIntegr. (VLSI) Syst.*, vol. 23, no. 5, pp. 973–977, May 2015.
[12] A. X. Glittas and G. Lakshminarayanan, "Pipelined FFT architectures for real-time signal processing and wireless communication applications," in *Proc. 18th Int. Symp. VLSI Design Test*, Jul. 2014, pp. 1–2.