Indian Journal of Science and Technology, *Vol 9(5), DOI: 10.17485/ijst/2016/v9i5/87165, February 2016*

ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

ASIC Implementation of High throughput FFT Processor for Scientific Applications

Sivanantham Sathasivam* **and G. Venu Reddy**

VLSI Division, School of Electronics Engineering, VIT University, Vellore - 632014, Tamil Nadu, India; ssivanantham@vit.ac.in, venureddyg2028@gmail.com

Abstract

Objectives: This paper aims at designing high speed and high throughput Fast Fourier Transform (FFT) processor which is a critical block and is widely used in many Digital Signal Processing applications. **Methods:** The proposed model works with both real and complex type of input data. Pipelining in the proposed architecture is achieved with single delay feedback methodology. **Findings:** The CMOS 0.18 µm is used to design Application Specific Integrated Circuit (ASIC) for the proposed FFT processor and it works with an input size of 36 bits at the operating frequency of 100 MHz, occupies an area of 1.27 mm and consumes 39 mW, at an operating voltage of 1.8V.Obtained results are compared with existing methods in terms of input word length, throughput, power dissipation and it shows that the proposed architecture gives high throughput, uses 3x more word length and 2x less power dissipation. **Applications:** The designed chip can be used in scientific computations since it require less power and operates in high speed.

Keywords: ASIC, Fast Fourier Transform, Pipelining, Single Delay Feedback, Throughput

1. Introduction

FFT is an alternative computation technique for discrete Fourier transform as it computes quickly. FFTs are used in a large number of applications ranging from digital signal processing and also in the computations such as, solving the partial differential equations, multiplication of large value integers etc. FFT processor operating frequency determines the range for which the processor can be used. The processor is designed to compute a large number of complex multiplications both at high speed and with consistent throughput. The other design challenges lie in the fact that by changing the number of points in FFT, the processor can be suited for various applications ranging from both engineering and scientific applications. Though this seems to be very convenient, varying the number of points leads to a number of changes both at register level through change in the number of bits and at architecture level by change in the length of shift registers, as well as modification of ROM depth. Also there is an additional overload of computational complexity as well. This complexity is not limited just to the butterfly multiplication unit or pipelined architecture but also to the modification of ROM depth through the addition of twiddle factors[1]. Pipelining is the key concept in FFT processor and these pipelining architectures are of various types, out of which single delay path feedback has been selected. The change in number of bits in this feedback path is the key point in designing the required N- point FFT. Further addition of the single feedback modules is done with respect to the primary block where the number of bits in the register is half the number from the predecessor block. This process of designing is done for consequent blocks until a one register in feedback path is formed which forms the final block for the pipelined design[2,3]. Additional multipliers are added alternatively between the single delay feedback models. The single delay feedback mechanism for a two radix system is as shown in Figure 1.

The butterfly units can be either of two points or four points. The delay element in the feedback path is essential, it indicates the points that are stored in the register and it is customized with respect to the input number of points. The radix-2[4] single delay feedback path architecture include ROM unit to contain the necessary twiddle

**Author for correspondence*

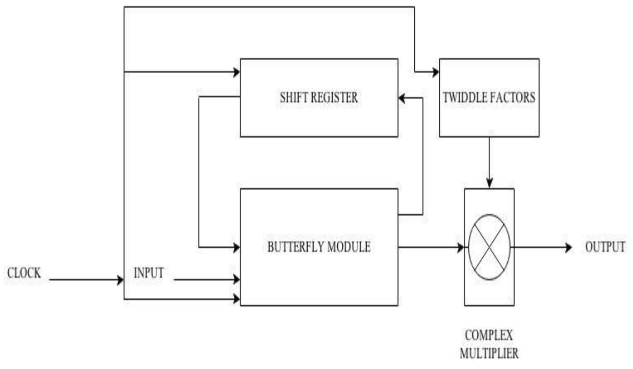

Figure 1. Basic Single Delay Feedback Path Unit Model.

factors. The complex multiplier is another key block and is added alternatively between single delay feedback models. Various floating-point arithmetic architectures are presented in[4-8] for signal processing applications.

2. FFT Algorithm

The discrete cosine transform of a particular N point sequence is[9]

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot W_N^{nk}, k = 0 \ldots N-1 \qquad (1)$$

Where n refers to time index, k stands for frequency index, W represents twiddle factor and both input and output are complex data. As higher number of multiplications and additions are required for these computations, modifications in the algorithm are made further which results in Fast Fourier Transform (FFT). These are of either radix two or four or split radix methods, even radix two raised to the power of k are used in many transform models [9-11]. The models for FFT used are either decimation in time or decimation in frequency. The design complexity issues for both decimation in time and decimation in frequency models are almost the same and hence can be ignored.

3. Architecture Design

3.1 Algorithm Implementation

The basic implementation follows the core idea of executing the signal flow graph of the N point FFT[12] as shown in Figure 2.The flow graph for the 16 point FFT is used. Further design of the signal flow graphs for any point FFTs can be designed based on the simple fact that at each stage half of the number of twiddle factors is to be used. In the first stage half the number of twiddle factors are required and it has to be noted that the twiddle factors required are N/2, where N is the required number of points. In the next stage lesser number of twiddle factors are required starting with the first twiddle factor but skipping alternate twiddle factors, therefore depending upon the stage of decomposition, the twiddle factors to be skipped is the 'stage of decomposition' minus one. This proceeds until we are left with a single twiddle factor[13].

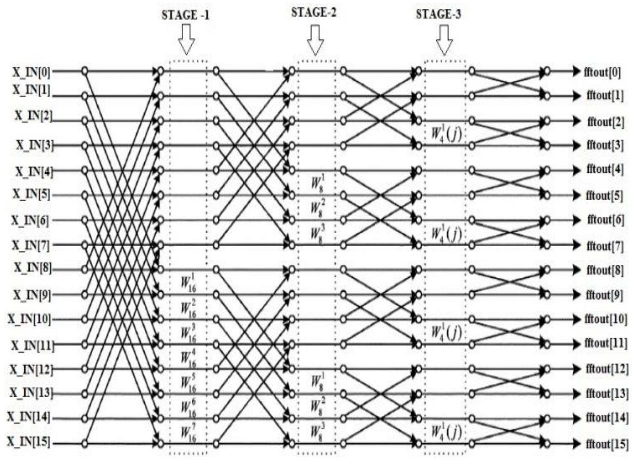

Figure 2. Signal Flow Graph for 16 point FFT.

3.2 Pipeline Design

The radix-two single delay feedback path is the major building block for the pipelined design. Though the structure is same, there has been a change in the FIFO register used at every stage leading to a highly customized design[14]. FIFO length is varied according to number of points in FFT and stage of computation[15]. In the first stage half the number of points is the length of the FIFO used in the feedback path. Successive stages FIFO are designed by halving the previous stage FIFO length until the length of one is reached marking the end of the operation. This type of architecture improves the throughput of the FFT processor. The butterfly structure shown in Figure 1, calculates the sum and difference of current input and the data in the FIFO register. This is followed by a multiplication unit with the butterfly output as a multiplicand and the twiddle factor as the multiplier. The twiddle factors are generated separately and are incorporated into the ROM[16]. These twiddle factors are accessed one after the other and omit the use of additional address generation unit as in many of the previous design techniques[17].

3.3 Proposed RTL Architecture

The input word length of proposed architecture is 36 bit wide in which the first eighteen bits are assigned for the real part of the data and the next eighteen bits are assigned to the imaginary part. Here, each eighteen bits, the first eleven are designated for abscissa and the last seven associate with the mantissa. The first bit represents the sign bit and the second bit is an optional overflow bit to indicate further computations. Such designing of the input data width helps in representing both real and complex types of data and varying this data width helps in designing higher point FFT[18]. The input bits are fed into the first register as shown in Figure 3, and a selector is used for selecting the registers of the two point butterfly structure. Here it has to be noted that the first N/2 points are stored in the shift register until the (N/2 +1)th bitpointarrives[19]. After the arrival of the (N/2 +1)th point, the points are directly fed to the butterfly module along with the register data which stores the first N/2 points. The output of the butterfly module is the sum and difference of the input and data in FIFO register. These are then given to the complex multiplier with butterfly module output as multiplicand and twiddle facto as multiplier. This process of computing continues until half the numbers of points from the previous stage are obtained. The twiddle factor that is the multiplier in the complex multiplication is preemptively stored in the ROM unit. These twiddle factors are half the number of points of FFT and based on the stage used, the number of twiddle factors necessary also varies. The twiddle factor suited for the computation is sent to the multiplier by a counter which counts the respective computation and sends out the twiddle factor. Twiddle factor is usually complex in nature but the data that is to be given to the complex multiplier must be binary in nature, hence the twiddle factor must be converted from complex to binary format[20].

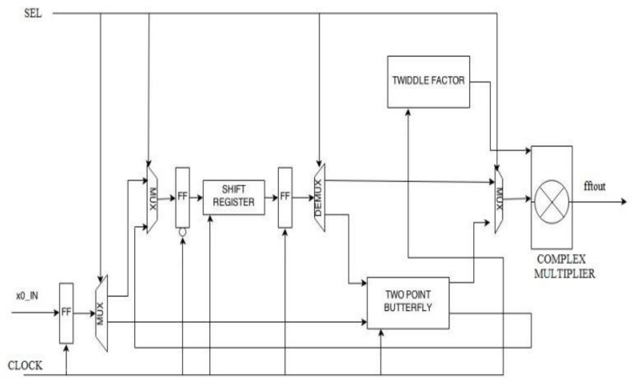

Figure 3. Proposed Architecture.

3.4 Customization

The straightforward advantage of using a single delay feedback path is that, the basic blueprint of the unit radix two single delay feedbacks remains the same. But modification of the FIFO length in the feedback path is such that it is equal to the half of the length of the predecessor FIFO length and is continued till a length of one is reached which marks the computation of the butterfly structure for each of the two inputs. The other repeatable unit in the single delay feedback model is the complex multiplier for which the twiddle factors from the ROM unit and output of butterfly structure are given[21]. Asynchronous clock is used for butterfly architecture block and twiddle factor block. The use of typical ROM unit for storage of the twiddle factors serves the part of a twiddle factor lookup table.

4. Results and Comparison

The ASIC for FFT processor shown in Figure 4, has been implemented using CMOS 0.18 μm technology and its power, area, timing reports have been studied. The implemented FFT processor works with an input size of 36 bits at the operating frequency of 100MHz, occupies an area of 1.27mm^2 and consumes 39 mW, at an operating voltage of 1.8V.The proposed design is compared with previous designs[22-24], results are tabulated in Table 1, Table 2 and a better performance is observed than existing designs.

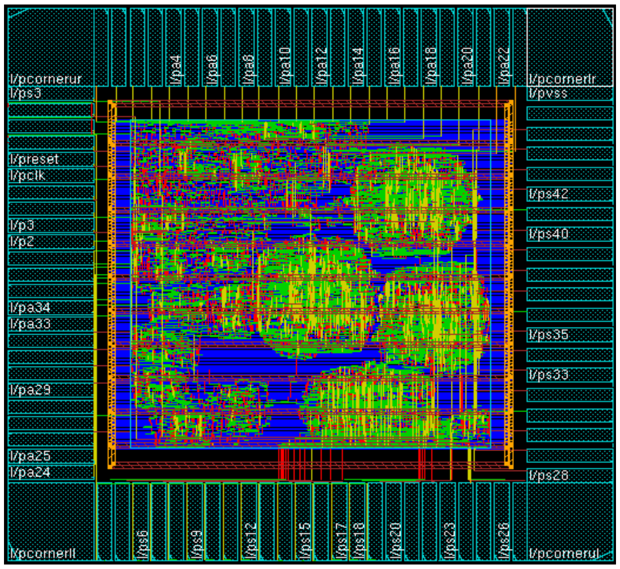

Figure 4. Chip Layout.

Table 1. Implementation results

Technology	0.18µm
Voltage	1.8 V
Word Length	36 bits
Gate Count	18030*
Core Size	1.27 mm^2
Frequency	100 Mhz
Power	39 mW*

*with low power constraints

Table 2. Comparison with existing architectures

	Proposed	*Jen-Chih*[20]	*Chin-Long Wey*[19]	*Bo Fu*[15]
Technology	0.18µm	0.35µm	0.18µm	0.18µm
Voltage	1.8 V	3.3 V	1.8 V	1.8 V
Word Length	36 bits	16 bits	24 bits	12 bits
Gate Count	18030*	14732	-	-
Core Size	1.27 mm^2	6.76 mm^2	1.8 mm^2	2.1 mm^2
Frequency	100 Mhz	80 Mhz	173Mhz	120 Mhz
Power	39 mW*	574 mW	40.80 mW	87 mW

5. Future Work

Through modification of input data length and also by designing the appropriate FIFO registers in the single delay feedback model we can extend the number of points of FFT and also by two's complementing the existing twiddle factors we can generate for Inverse Fast Fourier Transform (IFFT) that can operate at the same clock frequency but an extra selector is required for selection between them. Apart from this at the end in case of IFFT, output has to be divided by N where N is the number of points, thus an extra register or multiplier circuit will be required.

6. Conclusion

In this paper we have proposed a FFT processor which has a better input data register designations and works for complex, real and integer data types. We have also emphasized on maintaining the operating frequency suitable for scientific applications. The architecture that has been designed is highly flexible in terms of extending the number of operating points and also for computation of inverse fast Fourier transform.

7. References

1. Cho T, Lee H. A High-speed low-complexity modified FFT processor for high rate WPAN applications. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2013; 21(1):187–91.
2. Tsai P-Y, Lin C-Y. A generalized conflict-free memory addressing scheme for continuous-flow parallel-processing fft processors with rescheduling. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2011; 19(12):2290–302.
3. Patyk T, Guevokiran D, Pitkanen T, Jaaskelainen P. Low-power application-specific FFT processor for LTE applications. 2013 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS XIII), IEEE, Agios Konstantinos. 2013. p. 28–32.
4. Li W, Wanhammar L. A complex multiplier using overturned-stairs adder tree. 1999. Proceedings of ICECS'99. The 6th IEEE International Conference on Electronics, Circuits and Systems, Pafos. 1999; 1. p. 21–4.
5. Sivanantham S, Jagannadha Naidu K, Balamurugan S, Bhuvana Phaneendra D. Low power floating point computation sharing multiplier for signal processing applications. International Journal of Engineering and Technology. 2013; 5(2):979–85.
6. Niharika S, Suhas Sali A, Nithin V, Sivanantham S, Sivasankaran K. Implementation of Radix-4 Butterfly Structure to Prevent Arithmetic Overflow. 2015 Online International Conference on Green Engineering and Technologies. 2015. P. 16–20.
7. Rakesh Babu A, Saikiran R, Sivanantham S. Design of floating point multiplier for signal processing applications. International Journal of Applied Engineering Research. 2013; 8(6):715–22.
8. Sivanantham S. Design of low power floating point multiplier with reduced switching activity in deep submicron technology. International Journal of Applied Engineering Research. 2013; 8(7):851–59.
9. Jung Y, Tak Y, Kim J, Park J. Efficient FFT algorithm for OFDM modulation. 2001 Proceedings of IEEE Region 10 International Conference on Electrical and Electronic Technology. TENCON'01. 2001; 2:676–78.
10. Vetterli M, Duhamel P. Split-radix algorithms for length-p/sup m/DFT's. IEEE Transactions on Acoustics, Speech and Signal Processing. 1989; 37(1):57–64.
11. Chen J, Hu J, Lee S, Sobelman GE. Hardware Efficient Mixed Radix-25/16/9 FFT for LTE Systems. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2015; 23(2):221–29.
12. Xiao H, Pan A, Yun C, Zeng X. Low-cost reconfigurable VLSI architecture for fast fourier transform. IEEE Transactions on Consumer Electronics. 2008; 54(4):1617–22.

13. Guan X, Fei Y, Lin H. Hierarchical design of an application-specific instruction set processor for high-throughput and scalable FFT processing. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2012; 20(3):551–63.
14. Rudagi JM, Lobo R, Patil R, Biraj N. An Efficient 64-point Pipelined FFT Engine. 2010 International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom), IEEE, Kottayam. 2010. p. 204–8.
15. Kang H-J, Lee J-Y, Kim J-H. Low-complexity twiddle factor generation for FFT processor. Electronics Letters. 2013; 49(23):1443–45.
16. Varma B, Paul K, Balakrishnan M. Accelerating 3D-FFT using hard embedded blocks in FPGAs. 2013 26th International Conference on VLSI Design and 2013 12th International Conference on Embedded Systems (VLSID), IEEE, Pune. 2013. p. 92–7.
17. Lin Y-W, Lee CY. Design of an FFT/IFFT processor for MIMO OFDM systems. IEEE Transactions on Circuits and Systems I: Regular Papers. 2007; 54(4):807–15.
18. More TV, Panat AR. FPGA implementation of FFT processor using vedic algorithm. 2013 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC). IEEE, Enathi. 2013. p. 1–5.
19. Wey C-L, Lin S-Y, Tang W-C. Efficient memory-based FFT processors for OFDM applications. 2007 IEEE International Conference on Electro/Information Technology, IEEE, Chicago, IL. 2007. p. 345–50.
20. Kim EJ, Sunwoo MH. High speed eight-parallel mixed-radix FFT processor for OFDM systems. 2011 IEEE International Symposium on Circuits and Systems (ISCAS), IEEE, RiodeJaneiro. 2011. p. 1684–87.
21. Xi J, Chicharo JF. Computing running discrete Hartley transform and running discrete W transforms based on the adaptive LMS algorithm. IEEE transactions on circuits and systems. 2, Analog and digital signal processing. 1997; 44(3):257–60.
22. Cho T, Lee H. A High-speed low-complexity modified FFT processor for high rate WPAN applications. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2013; 21(1):187–91.
23. Yu C, Yen M-H, Hsiung P-A, Chen S-J. A low-power 64-point pipeline FFT/IFFT processor for OFDM applications. IEEE Transactions on Consumer Electronics. 2011; 57(1):40.
24. Bashir AF, Suruliandi A. Comparative evaluation of various FFT processor architectures. 2014 International Conference on Circuit, Power and Computing Technologies (ICCPCT), IEEE. 2014. p. 1105–13.