Available online at www.elixirpublishers.com (Elixir International Journal)

Electrical Engineering

Elixir Elec. Engg. 116 (2018) 50266-50268

ISSN: 2229-712X

VLSI Architecture for FFT Using Radix-4 of Complex Valued Data

Ch.Kamala Kumarai , M.Praveena Sindhu , M.B.Rashmi Teja ,P.Ramesh, K.Sudhakar, Nasreen and B.Thriveni

ARTICLE INFO

Article history:
Received: 20 February 2018;
Received in revised form:
13 March 2018;
Accepted: 24 March 2018;

Keywords
Discrete Fourier Transform,
Fast Fourier Transform.

ABSTRACT

The high growth of the semiconductor industry over the past two decades has put Very Large Scale Integration in demand all over the world. Digital Signal Processing has played a great role in expanding VLSI device area. The recent rapid advancements in multimedia computing and high speed wired and wireless communications made DSP to grab increased attention. For an N-point transformation the direct computation of the Discrete Fourier Transform (DFT) requires N^2 operations. Cooley and Tukey explained the concept of Fast Fourier Transform (FFT) which reduces the order of computation to Nlog¬$_2$N. The FFT is not an approximation of the DFT, it's exactly equal to the DFT. FFT decomposes the set of data to be transformed into a series of smaller data sets to be transformed. The size of FFT decomposition is called "radix". Then, it decomposes those smaller sets into even smaller sets. At each stage of processing, the results of the previous stage are combined with twiddle factor multiplication. Finally, FFT is calculated for each small data set. Generally, FFT's can be decomposed using DFT's of even and odd points, which is called a Decimation-In-Time (DIT) FFT, or they can be decomposed using a first-half/second-half approach, which is called a "Decimation-In-Frequency" (DIF) FFT.A large number of FFT algorithms have been developed, but among all radix-4 are most widely used for practical applications due to their simple architecture, with constant butterfly geometry and the possibility of performing them ‘in place’. The algorithm for 16-point radix-4 FFT can be implemented with decimation either in time or frequency. In this work, the decimation in time (DIT) technique will be adopted in order to implement the 16-point radix-4 FFT.

1. INTRODUCTION

The Fourier Transform is an inevitable approach in signal processing, particularly for applications in Orthogonal Frequency Division Multiplexing (OFDM) systems.The Fast Fourier transform(FFT) is an appropriate technique to do manipulation of DFT. The algorithm of FFT was devised by Cooley and Tukey in order to decrease the amount of complexity with respect to time and computations. A large number of FFT algorithms have been developed, but among all radix-2, radix-4 and split radix are most widely used for practical applications due to their simple architecture, with a constant butterfly geometry and the possibility of performing them ‘in place’.

The hardware of FFT can be implemented by two types of classifications– memory architecture and pipeline architecture.

The memory architecture comprises a single processing element and various units of memory. The merits of memory architecture include low power and low cost when compared to that of other styles. The specific demerits are greater latency and lower throughput. The above demerits of the memory architecture are totally eliminated by pipeline architecture at the expence of extra hardware in an acceptable way.

2. BUTTERFLY ARCHITECTURE

The most important element in FFT processor is a butterfly structure. It takes two signed fixed-point data from memory register and computes the FFT algorithm. The output results are written back in same memory location as the previous input stored. This method is called in-placement memory storage whereby it can reduce the hardware utilization. The butterfly architecture is shown in Fig. 1. The adder sums the input before being multiplied by the twiddle factor. The multiplier forms the partial product of the complex multiplication and produce two times bigger than input bit. Shift register would shift the bits to avoid overflow issue. Output of this butterfly would be kept in the register for the subsequent stage.

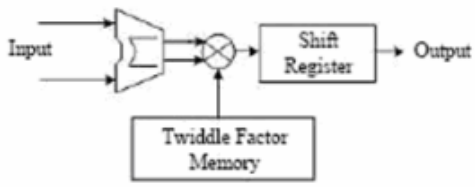


Fig 1. Butterfly architecture.

3. RADIX-4 FFT ALGORITHM

The algorithm for 16-point radix-4 FFT is studied for implementation. The FFT can be implemented with decimation either in time or frequency. Here the decimation in time (DIT) block diagram is studied. For the required specification two stages of 4-point butterflies are required. In each stage first part of the value corresponds to cosine function and the second to sine function. The other popular algorithm is the radix-4 FFT, which is even more efficient than the radix-2 FFT. The radix-4 FFT equation is listed below:

Tele:
E-mail address: Kamalaece.au@gmail.com

$$X(k) = \sum_{n=0}^{\frac{N}{4}-1} \left[x(n) + (-j)^k x\left(n+\frac{N}{4}\right) + (-1)^k x\left(n+\frac{N}{2}\right) + (j)^k x\left(n+\frac{3N}{4}\right) \right] W_N^{nk}$$

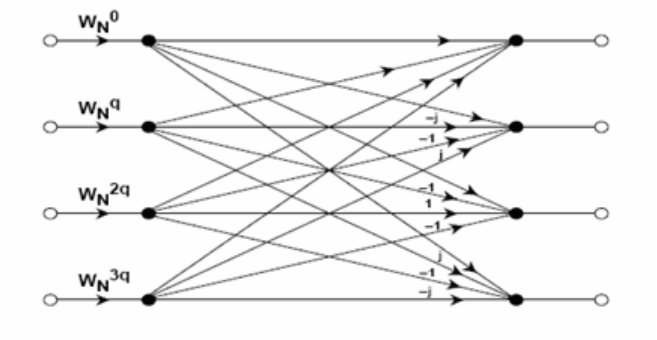


Fig 2. Radix 4 Butterfly.

4. GENERAL BLOCK DIAGRAM OF 16-POINT FFT

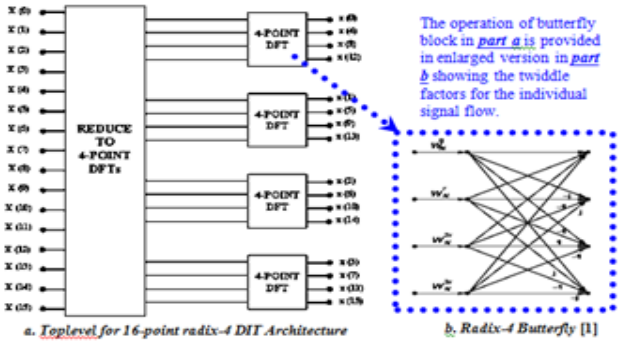


Fig 3. Block Diagram for 16-point radix-4 FFT.

The 16-point FFT implementation in radix-4 has two stages. Each stage has four 4-point butterflies. The implementation being done in DIT, before the butterfly calculation takes place the multiplication with the twiddle factor is to be done. Finally the order of the output sequence is found to be digit inversed.

Each butterfly has complex additions and subtraction being performed. The twiddle factor multiplier also has the complex inputs from the twiddle factor. But the twiddle factor value for radix-4, point 16 are limited to be 1, -1, j and –j.

Twiddle Factor, $W_N^k = e^{-j2\Pi k/N}$

The multiplication with these values can be easily be obtained by proper swap of the real and imaginary parts and using adder/subtractor to get the required sign being implemented in the result.

5. RESULTS AND DISCUSSION

The Radix 4 FFT algorithm and its functionality were discussed before. Now we deal with the simulation and synthesis results of 16 point Radix 4 FFT design. Here Modelsim tool is used in order to simulate the design and to check the functionality of the design. Once the functional verification is done, the design will be taken to the Xilinx tool for Synthesis process and the netlist generation.

The Appropriate test cases have been identified in order to test this modelled Radix 4 FFT design. Based on the identified values, the simulation results describes the operation of the Radix 4 FFT has been achieved. This proves that the modelled design works properly as per the process.

The test bench is developed in order to test the modeled design. This developed test bench will automatically force the inputs and will make the operations of 16 point radix 4 algorithms to perform.

Above description explains the implementation process and the different modes of operation. Now the implemented Verilog code is ready for the verification process that can be performed by passing the suitable test cases sequentially. So test passing of test case can be done through the test bench.

These test cases will clearly explain the operation of Radix 4 FFT and can be verified.

The identified test cases are simulated through the simulation tool and the main module is synthesized using the synthesis tool. The simulation results are obtained by using the identified test cases, which shows the different modes of operation.

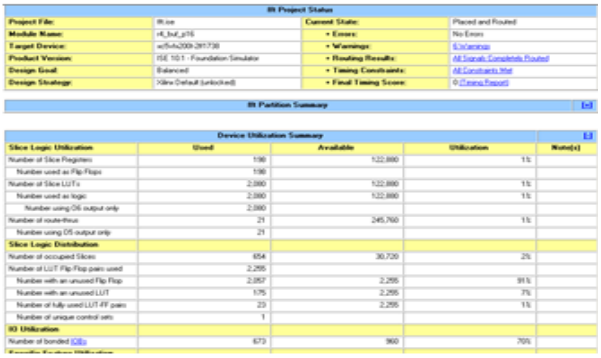
6. SUMMARY

This device utilization includes the following.

- Logic Utilization
- Logic Distribution
- Total Gate count for the Design

The device utilization summary is shown below in which it gives the details of number of devices used from the available devices and also represented in %. Hence as the result of the synthesis process, the device utilization in the used device and package is shown below.

Table 1. Device utilization summary.



6.1. Memory Usage:

Peak Memory Usage: 482 MB

6.2. Device Utilization Summary:

Number of BUFGs	1 out of 32	3%
Number of DSP48Es	36 out of 384	9%
Number of External IOBs	673 out of 960	70%
Number of LOCed IOBs	0 out of 673	0%
Number of Slice Registers	198 out of 122880	1%
Number used as Flip Flop	198	
Number used as Latches	0	
Number used as LatchThrus	0	
Number of Slice LUTS	2080 out of 122880	1%
Number of Slice LUT-Flip Flop pairs	2255 out of 122880	1%

7. CONCLUSION

Fast Fourier Transform (FFT) techniques have revolutionized the Digital Signal Processing techniques in the past 30 years. It does not only provide a fast response but also provide many logic thought to be un-realizable come easily in the range to be realized. The parallel processing of FFT hence has been proposed to be used in multiprocessor algorithms. When an FFT is implemented in special-purpose hardware, errors and constraints due to finite word lengths are unavoidable. While deciding the wordlength needed for an FFT, these quantization effects must be considered.

The algorithm for 16-point radix-4 FFT is studied for implementation. The HDL coding in DIT scheme is done and tested with some test vectors generated using Matlab. The design is synthesized on FPGA Virtex 5 configuration. A different implementation methodology could be used to reduce the quantization errors.

Also, the quantization factor could be made floating, i.e. dependent on the result rather only the inputs. This could provide different scaling levels from the same logic and have a large scope of application of the design.

8. REFERENCES

[1]James W.Cooley,and John W.Tukey,"An algorithm for the machine calculation of complex Fourier series," Mathematics of Computation,vol. 19,no. 90,pp.297301 ,Apr. 1965.

[2]R. Yavne , "An economical method for calculating the discrete Fourier transform, "in Proc AFIPS Fall Joint Computer Conf., vol.33, pp.115125,1968.

[3]P. Duhamel and H.Hollmann ,"Split-radix FFT algorithm, "Electronics Letters , vol. 20, no.1,pp. 1416,1984.

[4]S. S. Kerur,Prakash Narchi, C N Jayashree, Harish M Kittur ,V A Girish,"implementation of vedic multiplier for digital signal processing ".international conference on VLSI communication & instrumentation (ICVCI) 2011 proceedings published by international journal of computer applications (IJCA) , pp. 1-6.

[5]Charles. Roth,"Digital Systems Design using VHDL", Thomson Brooks/Cole 7th reprint,2005

[6]Himanshu Thapliyal, M.B Srinivas, "VLSI implementation of RSA Encryption system using Ancient Vedic Mathematics "Center for VLSI and Embedded System Technologies International Institute of information Technology Hyderabad India.