Volume : 1 | Issue : 6 | Nov 2012 • ISSN No 2277 - 8179

Research Paper

Engineering

KEYWORDS : FFT, VHDL, Pipeline.

Implementation of Virtex-5 Based 1024-Point Fast Fourier Transform (Fft) Computational Module for Wireless Communications

C.Shashikanth — M. Tech. Student, VLSI-SD, GNIT, Hyderabad

B.Kedarnath — Department of ECE, GNIT, Hyderabad

ABSTRACT

It deals with the design and implementation of Fast Fourier Transform (FFT) algorithm which can be utilized for wireless-communication. The computational scheme is two dimensional and implementation of this scheme on VIRTEX 5 LX330 FF1145 based XILINX's FPGA. This algorithm written in VHDL Language, which is verified and simulated in the MODEL SIM SE tool This provides new methodology to design and develop the processor for the Digital Signal processing aspects, and is implemented on the basis of pipelined architecture and Fast Fourier Transform approach. The proposed design has successfully worked on VIRTEX-5 FPGA, and it is applicable wireless communications. This will provide new era of the DSP processing technique and has the capability to extend the application to the REAL TIME ENVIRONMENT, which will provide a unique solution.

Introduction:

FFTs have use in innumerable signal processing applications and are often an important building block in such systems. Many of these applications require real- time operation in order to be useful. While Digital Signal Processors (DSPs) are available that can perform an FFT fast enough to keep up with many real-time applications, some systems require additional computation or have speed requirements that exceed the capabilities of a DSP alone. It is in these situations that dedicated logic for computing an FFT can be useful. Described in this paper is design, implementation, and testing of a Variable points (64, 128, 512, 1024) FFT implementation that takes advantage of pipelining, memory inference switching, and smaller FFTs to create a design capable of continuous real-time operation at high speeds.

Design of FFT Processor

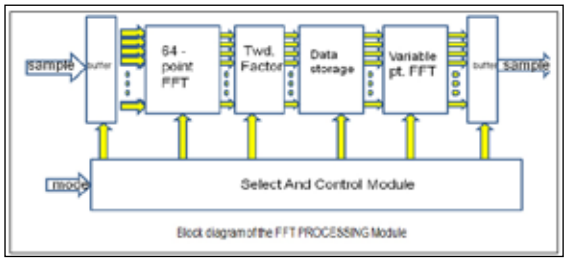
FFT scheme is employed at the DOWNLINK part of the computation where the retrieval of the original data from various sources is obtained and computed. Fast Fourier Transform (FFT) computation of the given samples is computed mainly on the basis of the Discrete Fourier Transform (DFT), which is given by

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

$$W_N = e^{-j2\pi/N}$$

Where x(n) is number of input samples, WN is a Twiddle factor, (k) is number of FFT samples.

The Fast Fourier Transform (FFT) which presented by Cooley and Tukey in their paper: "An algorithm for the machine calculation of complex Fourier Series" in 1965 is an optimized fast algorithm and increase the computational efficiency of the DFT. The algorithm is divided into time-based (DIT) and based on the frequency (DIF) fast Fourier transform. Both the DIF and the DIT FFT reorder the



Block diagram of the FFT PROCESSING Module

data from normal to bit reversed order (or the converse).

The basic idea of these algorithms are that the N point FFT is divided into smaller and smaller parts until only two points FFT(Radix-2).

Long FFTs are quite oft en used for frequency analysis and communications applications [61] • These long word lengths affect the memory architecture because long word lengths require more memory bandwidth for the matrix transpositions. The architecture is implemented using two shorter length FFTs (lengths NJ and N2) to calculate an FFT of length N = NIxN2•

. Mathematical representation of the 2D computations of the data is given by

$$X[k_1 N_2 + k_2] = \sum_{n_2=0}^{N_2-1} \left[e^{-j\frac{2\pi n_2 k_2}{N}} \left(\sum_{n_1=0}^{N_1-1} x[n_2 N_1 + n_1] e^{-j\frac{2\pi n_1 k_1}{N_1}} \right) \right] e^{-j}$$

where $0 \le k_1 \le N_1 - 1 \;:\; 0 \le k_2 \le N_2 - 1$.

In order to compute the data samples, the 2 Dimensional computation approach is utilized, where these data samples are arranged in set of sequences and computed

Therefore, this 2 Dimensional way of computing the data samples will

Provide considerable speed in the computation.

The minimum number of samples that are supported by this design is 64-samples and it requires only 1-dimensional computation.

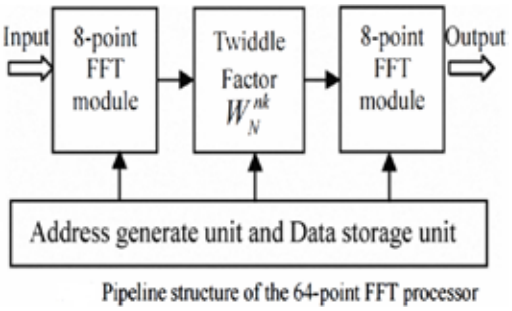
In FFT module, there are four important sub-modules which are used to compute FFT process operation. They are described as follow:

a. 64 point FFT:

The pipeline streaming of the data samples are carried initially in 64-point FFT module as shown above. 64 point FFT module is basic module for all the computations carried out the design.

This module consist three sub-modules which are used to compute the 64 point FFT Transform as shown above. The three sub-modules are cascaded one after the other for radix-8 computation of 64-point FFT Transforms

The 8-point FFT module performs computation of given 8 samples in one clock cycle. The Twiddle factor module is used to store the pre-calculated values of Twiddle factors for computation of 64-point FFT.



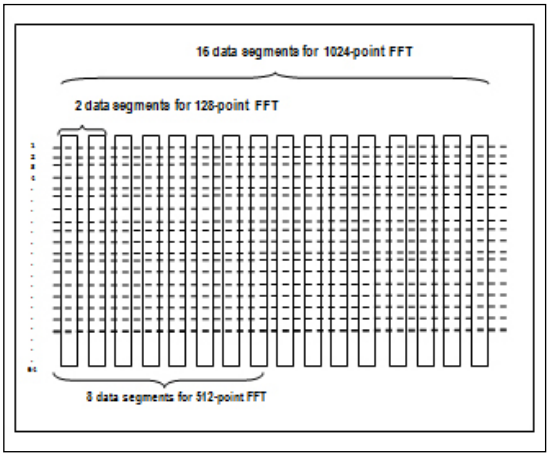
Pipeline structure of the 64-point FFT processor

Volume : 1 | Issue : 6 | Nov 2012 • ISSN No 2277 - 8179

The flow of the data samples across the three sub-modules is controlled by Address and Data storage module which is implemented on the basis of Finite State Machine FSM algorithm. The results of 64-point FFT samples are produced approximately after twenty clock cycles.

b. Twiddle Factor:
In order to compute the total number of samples greater than 64, twiddle factor module in the main architecture is used. The values of Twiddle factors that are stored in this module are used for computation of 128, 512, 1024-FFT respectively. While implementing in hardware these values are stored in the form of Look-up-table.

c. Data Module:
This module is employed for the compute the samples above 64. The data storage module in the main architecture is used for 2 Dimensional computations carried out for 128, 512, 1024 samples respectively i.e. for 128 samples, two 64 samples of computed data are stored two rows, similarly for 512 samples, its stores eight 64 samples of computed data. These samples are fed to the variable point FFT module for further computations.



d. Variable point FFT:
It consists of 2 point, 8 point and 16 point FFT computational logic. It is used to compute the data above the 64 data samples. i.e. to compute the 128 points FFT variable, this module use the 2 point FFT operation after computation of two sequence

This operation is initiated by first comparing and selecting the number of samples by applying mode signal i.e. if the number of samples are 64,128,512,1024 to compute FFT then accordingly assigning the mode signal to "00","01","10" and "11" respectively will select no-computation, 2-point FFT, 8-point FFT, and 16 point-FFT computations in this module.

e. Select and control module:
This module mainly used to select the individual sub modules in the FFT module for computation and precise flow of data. All the operations which are required to compute are carried with control flow of the data across the sub-modules. The algorithm implemented for this module FINITE STATE MACHINE (FSM).

Operational Description
When the FFT operations are initiated, the process of computation starts by comparing the number of samples at the inputs of the device/module by assigning mode signal.

If the number of samples at the input exceeds the 64 samples then the process 2D computation is initiated i.e., if N=128 samples, it can be computed as 128 = 2 * 64 -point FFT (firstly the data is arranged in 64 lines and 2 rows, secondly the input data will transform the 64 points FFT, then the result is multiplied by twiddle factor and thirdly, the result is computed with 2 points FFT).

RESULTS Description
Description:
The computation of 64 point FFT is initiated by selecting the mode signal. The input samples are 16 bit complex valued numbers and initated init value is fed to '1'.

After initiating the sample, the internal counter will start counting the samples and fed them to the internal buffers, when count is at multiples of 64, the buffer initiating signal will glow high and data is fed to computational module and continues till count is 1024 respectively.

The buffered samples are fed to 64-point FFT module where the computed sample are multiplied with the twiddle factors and stored in the DATA STORAGE MODULE respectively. After all the 1024 samples buffered and computed with initial counter value to 1024, then the accessing of the data samples data storage module is done with two-dimensional\approach and fed to Variable point FFT.

This Module is initiated with 16 point FFT computational algorithm for 1024 point FFT.

FFT computation initiated at Variable point FFT module. The result generated at the output of FFT computational module is buffered out at output buffer module and fed out in the serial manner with the frequency equal tothe clk signal.The generation of the output FFT computed samples taken 29 us approximately.

Similarly, 512-point FFT firstly computes for 64 points FFT, and then transforms them into 8 points FFT. The same way, for 1024 point's FFT can be computed accordingly, With 16 point

Conclusion
The proposed design has successfully worked on VIRTEX-5 FPGA, and it is applicable wireless communications. This will provide new era of the DSP processing technique and has the capability to extend the application to the REAL TIME ENVIRONMENT, which will provide a unique solution.

REFERENCE

1. "Design and Implement of FFT Processor for OFDMA System Using FPGA", by WANG Xiu-fang and HOU Zhen-long. | 2. "LTE for UMTS OFDMA and SC-FDMA Radio Based Access", by Harri Holma and Antti Toskal. | 3. "Pipelined 64-point fast Fourier transform for Programmable Logic Devices "by Joel J. Fúster and Karl S. Gugel | 4. " Design and Implementation of A Scalable Fast Fourier Transform Core", by Cheng-han Sung, Kun-bin lee, | 5. L. J. Cimini. Analysis and simulation of a digital mobile channel using orthogonal frequency division multiplexing. IEEE Trans. Commun., COM-33(7):665-675, Jul. 1985. | 6. M. Alard and R. Lassalle. Principles of modulation and channel coding for digital broadcasting for mobile receivers. EBU Review, (224):47-69, Aug. 1987. | 7. A.P. Chandrakasan and R.W. Brodersen. Low Power Digital CMOS Design. Kluwer Academic Publishers, 1995. | 8. L.R. Rabiner and B. Gold. Theory and Application of Digital Signal Processing. Prentice-Hall, Inc., 1975. | 9. E.H. Wold and A.M. Despain. Pipeline and parallel-pipeline FFI7 processors for VLSI implementation. IEEE Trans. Comput. C-33(5):414-426, May 1984. |