Design and Implementation of 3D FFT

VRINDA KULKARNI[1], SHRUTI OZA[2]
[1]Department of Electronics, Bharati Vidyapeeth Deemed University College of Engineering, Pune, India
[2]Professor, Department of E&TC, Bharati Vidyapeeth Deemed University College of Engineering, Pune, India

Abstract— Molecular Dynamics simulation is a fundamental method to know the chemical and biological systems. Molecular Dynamics simulation has long range and short range force computations. Electrostatic computations are used for computing long range force in Molecular Dynamics simulation. The 3D FFT is the base of this computation. This paper focuses on design and implementation of 3D FFT for generic number of Fast Fourier Transform points on hardware such as field programmable gate arrays. The design is partially generic. It uses standard cores for memory and one dimensional Fast Fourier Transform calculation. Rest of the modules is coded using hardware description language. As 3D FFT computation is critical and time consuming this design eases out the computation.

Index Terms— Computation, Fast Fourier Transform, Field Programmable Gate Array, Generic, Molecular Dynamics Simulation, Multidimensional Signal Processing, Three dimension

1 Introduction

Molecular Dynamics simulation (MD) is a computer simulation method for studying the physical movements of atoms and molecules.MD consists of numerically solving the classical equations of motion for a collection of atoms. MD process consists of two phases 1. Force calculation 2.Motion update. Evaluation of the forces is more compute intensive. The forces are of two types, bonded and non-bonded forces. The non-bonded forces are again classified into short range and long range forces.

When the long range electrostatic interactions are included in the Molecular Dynamics simulation, it increases the computational cost. Computation time for long range interactions is also very high.

Fast Fourier Transform (FFT) is an analytical and numerical tool widely used in sciences and engineering. Application area of FFT is signal processing, applied mathematics, image processing, spectral analysis to solve differential equations etc. [1]

3D FFT is the heart of electrostatic computations in the calculation of long range force in Molecular Dynamics simulation. [2]. 3D FFT reduces the complexity of such computations.

3D FFT basic concepts:

Three dimensional FFT can be visualized as a cube of N points. These N points are considered for FFT calculation. [3] The data is assumed to be in three dimensions X, Y and Z. The data point 3D FFT can be represented as shown by equation

$$F(k_x,k_y,k_z)=\sum_{z=0}^{N-1}\sum_{y=0}^{N-1}\sum_{x=0}^{N-1}f(x,y,z)W_N^{xk_x}W_N^{yk_y}W_N^{zk_z} \quad (1)$$

where $W_N = e^{-i\frac{2\pi}{N}}$.

Thus FFT computation is done in three dimensions to achieve 3D FFT computation. The method for 3D FFT implementation on FPGA is available. This design has referred the method of computation from paper "3D FFTs on a single FPGA" by Benjamin Humphries, Hansen Zhang, Jiayi Sheng, Raphael Landaverde and Martin C. Herbordt. This paper details on design and implementation of 3D FFT for generic number of FFT points. The main purpose is to have ease in computation of 3D FFT for any number of FFT points.

The rest of the paper is organized as follows. Section II describes the method of 3D FFT computation. Section III describes the implementation details for generic number of 3D FFT computation. The next section discusses results with post synthesis simulation waveforms. The section V concludes the paper.

2 3D FFT Computation Method

As shown in figure 1, 3D FFT can be visualized as a cube. Each small cube represents a point of data in N point FFT calculation. [3]

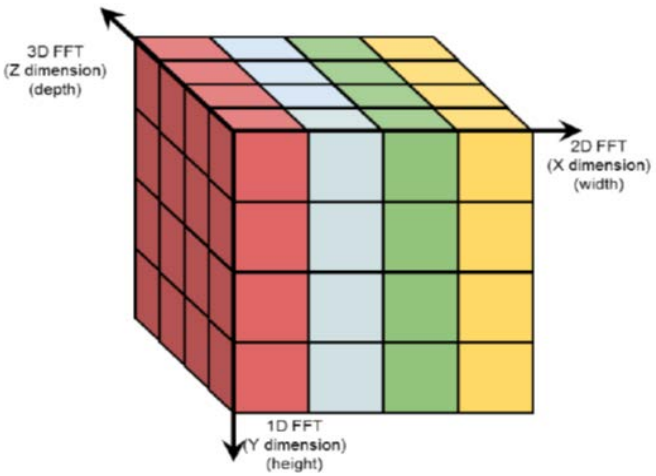

Figure 1 FFT computation in three dimensions

For 3D FFT, it is needed to calculate FFT in all three dimensions.1D FFT of each column data in Y dimension is calculated. Now 2D FFT of each row data in X dimension is calcu-

lated. Finally 3D FFT of each deep row is calculated in Z dimension. Each dimension requires N^2 single dimension FFTs. [4] One important point to note is that each 1D FFT computation is performed after completion of prior 1D FFT computation.

Design overview. Figure 2 shows the block diagram for implementation of 3D FFT.The input data is stored in RAMs. It is assumed that the cube is sliced in 2D slices and each such slice is mapped onto each RAM.

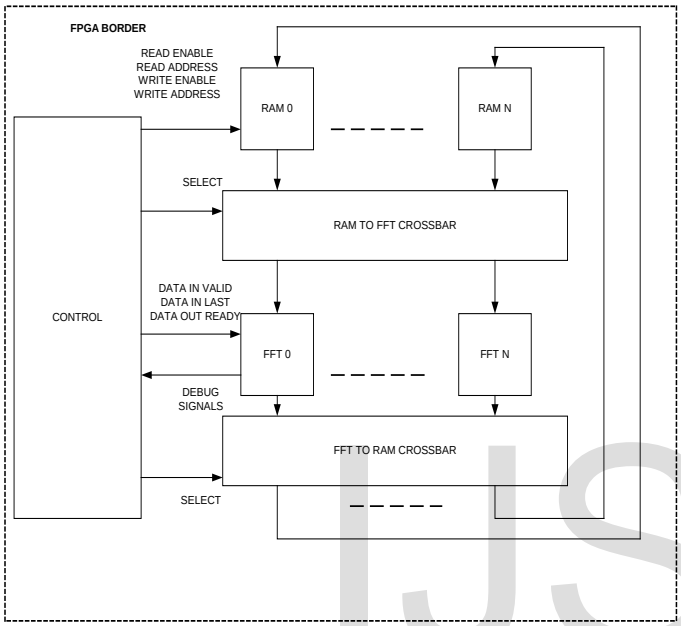

Figure 2 Block Diagram for 3D FFT Design

The basic blocks of design are state machine controller, RAMs, FFT IP cores and crossbar logic. RAMs are for storing the input and intermediate computation data. FFT cores are for calculation of one dimension Fast Fourier Transform. Crossbars plays important role in reading 3D input data from memory and writing 3D output data to memory.

In 1D and 2D FFT computation, for reading input data from memory and writing output data to memory crossbar connections are one to one. State machine mainly controls memory read and write operation, memory addressing and generates control signals for FFT core.

3 DESIGN AND IMPLEMENTATION

3D FFT for various number of FFT points is available. In this design generic number of FFT points are considered for computation.

Xilinx LogiCORE IP Fast Fourier Transform V7.1 is used for 1D FFT computation.[5] LogiCORE IP Block Generator V7.3 is used for BRAM cores generation.[6] Target device considered is Xilinx Virtex 5 XC5VLX155 –FF1153 -1.[7]

The other logic modules such as crossbars, state machine controller, memory address generation modules etc. are implemented using VHDL.

For 1D computation memory address is incremented in continuous fashion. In 2D memory address is incremented with a stride equal to size of FFT point. For 3D computation memory addressing is different from 1D and 2D. Here for addressing, cross bar selection for each FFT point will go across all of the RAMs. Thus for an N point FFT the data is collected starting from RAM 0 to RAM N. The N point input data for FFT core will be from each RAM. This is repeated for all cores.
This design has limitation. To make the design fully generic some VHDL constructs are needed which are supported only in VHDL 2008. Questasim and Vivado support VHDL 2008 but ISE does not support it. The Vivado tool is not backward compatible.
This design is partially generic in the sense that the BRAM and FFT cores are explicitly generated with required specification of parameters for number of FFT points. It can be made fully generic with the tools mentioned earlier in next phase.

For this design Questasim 10.4 is used for functional and post synthesis simulation and ISE14.7 is used for synthesis and post synthesis simulation model generation. VHDL 2002 is used for coding.

4 RESULT

Figures 3 and 4 shows post synthesis simulation result for 16 point FFT. Figure 3 shows input waveforms. For 16 point 3D FFT 16 BRAMs, 16 1D FFT cores are generated. The other modules are coded in VHDL. Input data to the design is given through test bench. Each of 16 BRAMs will have slice data of cube.

Start signal of FFT core is asserted through state machine controller. The first data sample is applied as soon as "RFD" (ready for data) goes high. Thus the real and imaginary inputs of core are driven. The input data corresponds to "XN_INDEX".

As shown in figure 4 after completion of 1D and 2D computation, 3D computation is done.2D output data will be input for 3D computation. It can be seen from XN_INDEX in waveforms that the 3D input to any core spans through all the memories. When data valid signal "dv" is asserted by the core the output on data bus Xk_re and Xk_im is valid. The output also corresponds to XK_INDEX.

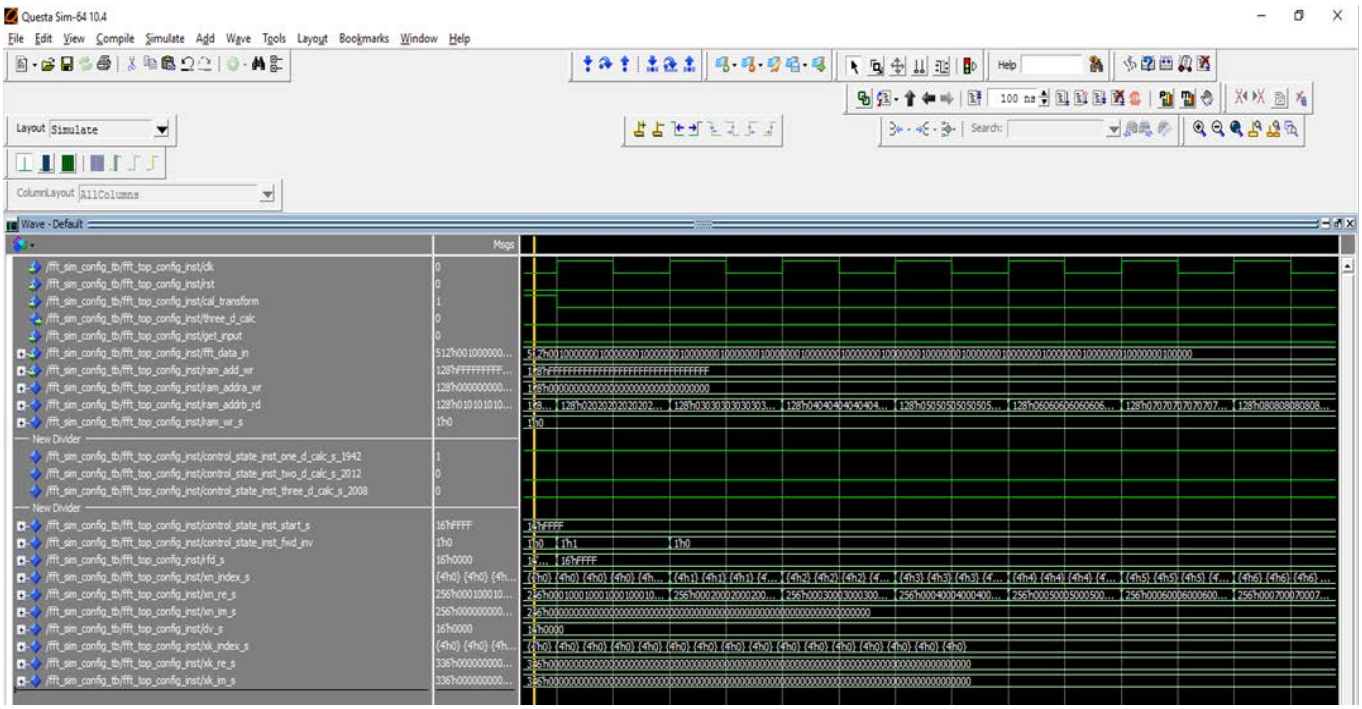

Figure 3 Input Waveforms

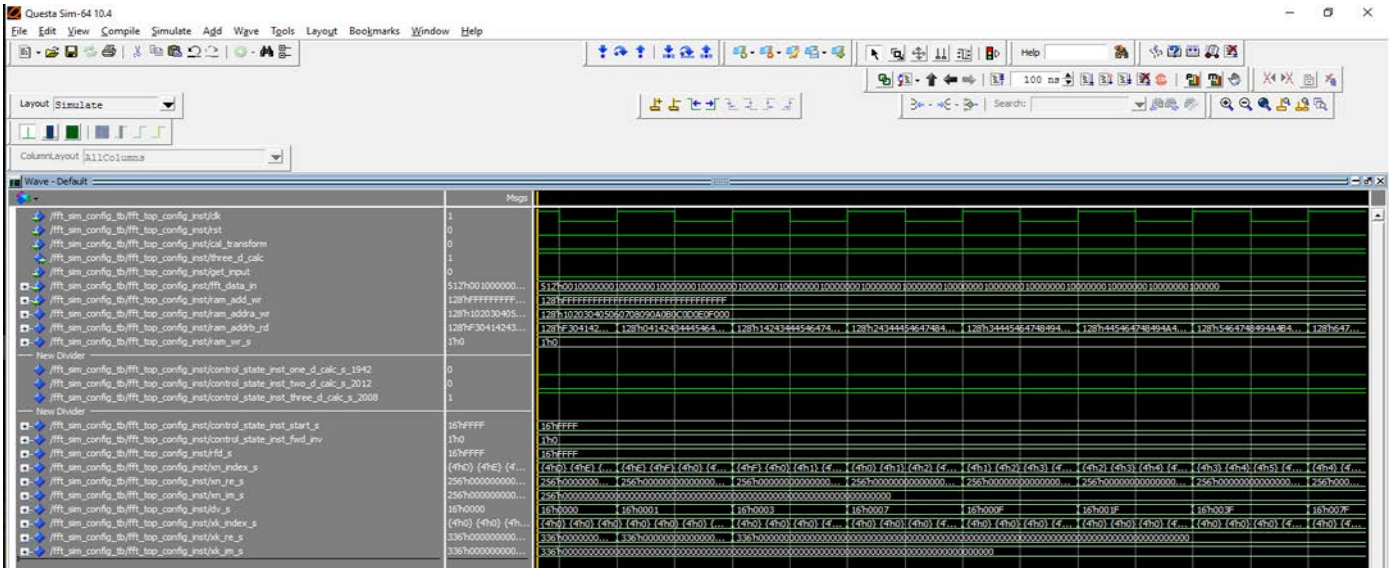

Figure 4 Output waveforms

5 CONCLUSION

This paper highlights the 3D FFT computation for generic number of FFT points on hardware such as FPGA. The design is made partially generic. The cores used in design are generated with explicit values due to tool limitations. Other modules such as state machine controller, memory addressing and crossbars are made generic. The core is based on Cooley Tukey algorithm for 1D FFT calculation. The width of output data for FFT is greater than input data width. The amount by which the output width increases can be calculated by a simple formula available in Xilinx core generator document. This parameter is also included in generic parameters list, which helped in port width, data path and control path width calculation. Post synthesis results are compared with those obtained from software program.

The design can be made fully generic in future work by use of VHDL 2008 language for coding and Vivado tool. Latest versions of cores for 1D FFT and Bram can be used with big FPGA device.

REFERENCES

[1] Orlando Ayala, Lian-Ping Wang.(2013,January).Parallel implementation and scalability analysis of 3D Fast Fourier Transform using 2D domain decomposition. Parallel Computing. [Online].39(1), pp. 58–77. Available: www.elsevier.com/locate/parco

[2] B. Humphries, H. Zhang, J. Sheng, R. Landaverde, and M. Herbordt, "3D FFTs On A Single FPGA," in Proc. IEEE Symp. On Field Programmable Custom Computing Machines (FCCM), 2014, pp. 68–71

[3] J. Sheng, B. Humphries, H. Zhang, and M. Herbordt, "Design of 3D FFTs with FPGA clusters," in Proc. High Performance Extreme Computing Conference (HPEC), 2014, p. TBD

[4] Benjamin Humphries," Using Offline Routing To Implement A Low Latency 3D FFT In A Multinode FPGA System", M.S. thesis, B.S., Georgia Institute of Technology, Boston Univ., 2013.

[5] Xilinx.(2011) LogiCORE IP Fast Fourier Transform v7.1: Product Specification.[Online].Available:https://www.xilinx.com/support/documentation/ip_documentation/xfft_ds260.pdf

[6] Xilinx.(2012) LogiCORE IP Block Memory Generator v7.3: Product Guide[Online].Available:https://www.xilinx.com/support/documentation/ip_documentation/blk_mem_gen/v7_3/pg058-blk-mem-gen.pdf

[7] Xilinx.(2015) Virtex-5 Family Overview: Product Specification [Online].Available:https://www.xilinx.com/support/documentation/data_sheets/ds100.pdf