DESAIN DAN SINTESIS ARSITEKTUR HARDWARE IFFT (INVERSE FAST FOURIER TRANSFORM) 64 TITIK BERBASIS BAHASA PEMROGRAMAN VHDL

Desia Ilmina Suprapto[1], Koredianto Usman[2], Iswahyudi Hidayat[3]

[1]Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

Abstrak

Metode IFFT (Invers Fast Fourier Transform) adalah inverse atau kebalikan dari FFT (Fast Fourier Transform), dimana FFT merupakan metode untuk pemecahan sinyal diskrit. IFFT merupakan algoritma komputasional yang cepat untuk menghitung IDFT (Inverse Discrete Fourier Transform). Subcarrier pada IFFT memiliki frekuensi harmonisasi kelipatan bulat dari frekuensi dasarnya seperti halnya komponen deret Fourier pada sinyal komposit. Salah satu penggunaan IFFT adalah pada system OFDM (Orthogonal Frekuensi Division Multiplexing), disini IFFT berperan sebagai modulator pada transmitter.

Pada penelitian ini akan dibuat desain struktur hardaware dari IFFT dengan mengkodekan setiap blok-blok dalam IFFT menggunakan bahasa VHDL . Desain sistem dengan VHDL ini akan memodelkan sistem sesuai dengan kebutuhan dari sistem IFFT dan mensimulasikan sebelum perangkat sintesis mentranslate desain dalam hardware secara nyata dengan ModelSim sebagai software pendukung . Dari hasil permodelan dan simulasi maka akan dilakukan sintesis pada tingkat hardware FPGA dengan Xilinx.

Sebagai input dari IFFT berupa sinyal diskrit hasil keluaran kuantizer dan keluarannya akan didapatkan sinyal diskrit dalam domain waktu. Disini diharapkan akan dapat dilihat hasil output dari IFFT. Hasil implementasi di FPGA akan dianalisis performansinya yaitu meliputi : parameter kinerja hasil desain, jumlah slice yang diperlukan, delay proses dan keberhasilan algoritma untuk perhitungan IFFT.

Kata Kunci : IFFT,VHDL,ModelSim,Xilinx

Abstract

The method of IFFT is the inverse or the opposite of FFT, in which FFT is a method to solve the discrete signal. IFFT is a quick computational algorithm to count IDFT. The sub carrier in IFFT has a rounded common multiple harmonize frequency of it basic frequency as well as Fourier row component in composite signal. One of the IFFT usages is the OFDM system; in here IFFT has the role as the modulator in transmitter.

In this research, it would be made a hardware structure design of IFFT by coding every block in IFFT by using VHDL language. The system design with VHDL would give a model of system which matched with the need of IFFT system and simulated before the synthesize tools translated the design in hardware as a real with ModelSim as support software. From the result of modeling and simulation there would be synthesize at the hardware FPGA level with Xilinx.

As the input of IFFT was the discrete signal as the output of quantizer, and from the output there would be diskrit signal in time domain. In here, it was hoped that the output of IFFT could be seen. The result of the implementation in FPGA would be analyzed its performance which included: the design result of parameter performance, number of slice needed, process of delay and the success of algorithm to count IFFT.

Keywords : IFFT,VHDL,ModelSim,Xilinx

BAB I
PENDAHULUAN

1.1. Latar Belakang

Pada perkembangan DSP (Digital Signal Processing) yang dapat mensintesis penjumlahan sinyal termodulasi dengan akurat, maka Modulasi Multicarrier selalu sukses dilaksanakan pada dasawarsa terakhir ini. Salah satu metode yang digunakan untuk mengatasi kendala dalam modulasi multicarrier adalah metode IFFT. Berbagai macam teknologi yang menggunakan metode IFFT antara lain pada system OFDM (Orthogonal Frequency Division Multiple Access) dan pada system DMT (Discret Multitone) untuk teknologi ADSL (*Asymmetric Digital Subscriber Lines*) dan VDSL (*Very-High-Speed Digital Subscriber Lines*). Kedua system tersebut menggunakan metode IFFT sebagai modulator di transmiter. Sudah ada beberapa riset mengenai IFFT pada FPGA dengan 8 titik, namun secara kebutuhan dalam telekomunikasi telah digunakan IFFT dengan 256 titik.

IFFT pada dasarnya merupakan invers/kebalikan dari metode FFT. FFT adalah algoritma komputasional yang efisian untuk menghitung DFT (Discret Fourier Transform) bila ukuran N adalah pangkat 2 dan bila pangkat 4. Dimana dalam pemrosesan sinyal DFT mempunyai peran penting, termasuk analisis korelasi dan analisis spectrum. Dapat dikatakan bahwa IFFT merupakan algoritma cepat untuk menghitung IDFT (Invers DFT). Algoritma cepat dalam IFFT adalah kemampuan menghitung lebih cepat jumlah perkalian kompleks pada perhitungan langsung yaitu N^2 menjadi $(N/2)\ \log_2 N$ perkalian. Metode IFFT ini akan memberikan perhitungan yang lebih efisien sehingga mempercepat proses sinyal.

Pada penelitian ini akan diaplikasikan algoritma IFFT 64 titik pada FPGA. Dengan melakukan permodelan sistem/desain hardware pada IFFT, melukan simulasi dan sintesis hardware pada FPGA dengan bahasa VHDL. Dari implementasi ini akan dapat dilihat hasil keluaran dari IFFT 64 titik.

1.2. Perumusan Masalah

Masalah yang akan diteliti dalam Tugas Akhir ini adalah :

1. Bagaimana struktur/desain hardware yang dibutuhkan dalam sistem IFFT.
2. Mendesain blok – blok arsitektur IFFT pada VHDL.
3. Melakukan simulasi pada desain blok – blok arsitektur dengan memberikan tes vector dalam bentuk test bench.
4. Melakukan sintesis terhadap arsitekture IFFT yang telah dibuat.

1.3. Pembatasan Masalah

Batasan-batasan masalah yang digunakan dalam tugas akhir ini adalah :

1. Menggunakan algoritma IFFT Radiks-2.
2. Menggunakan algoritma peruraian dalam frekuensi (*Decimation in Frequency*)
3. Jumlah titik inputan dari IFFT (N) yang digunakan adalah 64 titik.
4. Penelitian sampai dengan sintesis hardware.
5. Bahasa pemrograman yang digunakan adalah VHDL (*VHSIC Hardware Description Language*), menggunakan software *ModelSim SE 6.0*.
6. Sintesa hardware yang menggunakan software *Xilinx ISE 8.1i.*
7. Sintesa pada Xilinx ISE 8.1 menggunakan hardware FPGA seri Virtex xc4vlx25-10sf363

1.4. Tujuan Penelitian

Tujuan dari penelitian adalah sebagai berikut :

1. Mendesain arsitektur hardware algoritma IFFT dengan bahasa pemrograman VHDL.
2. Sintesis hasil desain VHDL algoritma IFFT dengan *Xilink Synthesis Tool seri ISE 8.1*.
3. Mendapatkan hasil sintesis pada IFFT, meliputi kebutuhan slice, IOB, LUT, Flip – Flop, GCLK, FIFO/RAM, dan DSP untuk N titik.
4. Memprediksi hasil sitesis untuk kebutuhan slice, IOB, LUT, FliP – Flop, GCLK, FIFO/RAM, dan DSP pada jumlah N yang lebih banyak.
5. Menghitung delay yang terjadi pada saat simulasi algoritma IFFT.

6. Analisa hasil outputan simulasi IFFT pada ModelSim dengan hasil keluaran pada Matlab.

1.5. Metodologi Penelitian

Penelitian ini dilakukan dengan metodologi sebagai berikut :

1. Tahap studi literature

   Mempelajari dan memahami konsep dari IFFT serta study awal mengenai bahasa VHDL.
2. Tahap perancangan

   Perancangan blok diagram system fungsional IFFT berdasarkan hasil studi literatur dengan menggunakan bahasa VHDL.
3. Tahap Simulasi

   Melakukan simulasi pada program yang dibuat dengan testbench sinyal pada arsitektur desain yang telah dibuat.
4. Tahap Sintesis

   Mensitesis hasil aristekture VHDL setelah dilakukan simulasi dan didapatkan parameter – parameter yang dinutuhkan.
5. Tahap analisa dan kesimpulan

1.6. Sistematika Pembahasan

Laporan tugas akhir akan dirancang dengan sistematika sebagai berikut :

BAB 1 : PENDAHULUAN

Pada bab 1 ini, dijelaskan mengenai latar belakang, tujuan, batasan masalah, dan metoda pelaksanaan penelitian serta sistematika pembahasan laporan.

BAB 2 : DASAR TEORI

Bab ini merupakan tinjauan pustaka dari IFFT (Invers Fast Fourier Transform)

BAB 3 : PERANCANGAN DAN IMPLEMENTASI

Bab ini akan membahas perancangan blok-blok system hardware yang mendukung dan simulasi hasil rancangan serta implementasi pada hardware secara nyata.

BAB 4 : PENGUJUIAN DAN ANALISA

Bab ini berisi analisa hasil simulasi dan hasil outputan yang dihasilkan sesuai dengan perumusan masalah.

BAB 5 : KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari tugas akhir dan saran untuk pengembangan selanjutnya.

Telkom University

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan dalam tugas akhir ini dapat disimpulkan beberapa hal sebagai berikut:

1. Sudah dapat dilakukan perancangan arsitektur IFFT 64 titik sampai dengan tahap simulasi fungsional sistem dan sintesis rangkaian hasil desain.
2. *Output* pada IFFT yang dihasilkan setelah dilakukan penelitian pada perancangan FFT hasil yang didapatkan sesuai dengan input pada IFFT.
3. *Output* IFFT hasil simulasi sama dengan hasil *output* dari Matlab.
4. Berdasarkan hasil sintesis IFFT 64 titik didapatkan jumlah resource yang dibutuhkan adalah jumlah slice 9%, jumlah flip – flop 6%, jumlah LUT 7%, jumlah IOB 31%, jumlah FIFO16/RAMB16 23%, jumlah GCLK 3%, jumlah DSP48 66%.
5. Didapatkan hasil periode minimum hasil sintesis 12,689 ns, maka frekuensi maksimumnya 78,807 MHz.

5.2 Saran

Dari serangkaian penelitian yang telah dilaksanakan, beberapa saran pengembangan yang dapat dilakukan adalah:

1. Dilakukan pengujian untuk jumlah titik input yang lebih besar untuk kemudian ditentukan optimasi hasil design yang dapat dicapai.
2. Dilakukan proses implementasi pada board FPGA.
3. Dilakukan penelitian dengan menggunakan algoritma radiks 4 yang kemudian dibandingkan dengan sistem IFFT yang menggunakan algoritma radiks 2.

DAFTAR PUSTAKA

Chang, K.C., *Digital Systems Design with VHDL and Synthesis,* Matt Loeb, USA.1999

Ludeman, Lonnie C., *Fundamental of Digital Signal Processing,* John Wiley & Sons, Inc, Canada. 1987.

Miller, Adam Robert, *Development and Verificationof ParameterizedDigital Signal Processing Macros for Microelectronics Systems,* The University of Tennesee, Knoxville. 2003.

Pedroni, Volnei A., *Circuit Design With VHDL*, Massachusetts Institute of Technology, USA. 2004.

Proakis, John G. dan Manolakis, Dimitris G., *Pemrosesan Sinyal Digital,* Edisi Bahasa Indonesia Jilid 1, Prenhallindo, Jakarta. 1997.

Wardana, Ali, *Desain dan Implementasi IDFT (Inverse Descrete Fourier Transform) untuk OFDM dengan FPGA*. Tugas Akhir STT Telkom. Bandung. 2007.

Wada, Tom, *64 point Fast Fourier Transform Circuit (Version 1.0)*. www.ie-u-ryuku.ac.id. 2006

www.opencores.org

www.wikipedia.com