FPGA Implementation of 8K Points FFT/IFFT for a 5G Channel Emulator

By

Habib-ur-Rehman 01-133152-039

Hamna Tariq 01-133152-043

Supervised by

Dr. Atif Raza Jafri

2015-2019

A Report is submitted to the Department of Electrical Engineering,
Bahria University, Islamabad.
In partial fulfillment of requirement for the degree of BS(EE).

Dedication

This work of ours is dedicated to our families and our supervisor. With their continual support and backing we accomplished the task with maximum possible competence.

Acknowledgements

We are deeply grateful to the Almighty ALLAH for His countless blessings upon us, and for His bestowed strength and abilities, and for the opportunity to commence this project. We are forever grateful to our supervisor Dr. Atif Jafri, without his aspiring guidance and polite supervision this project would not be an achievement. In addition, we are thankful to our instructor Engineer Muhammad Noman for his assistance and dedicated involvement. Thanking Bahira University Islamabad for providing us the Equipment and labs whenever needed to complete this project.

Abstract

Channel emulator is an instrument for emulating the channel environment faced by transmitted waveforms. In a test environment, channel emulators replicate the channel between a transmitter and a receiver and provide a transmitted waveform affected through channel effects to the receiver input. FFT and IFFT blocks are major components of a channel emulator.

Fast Fourier Transform (FFT) is a proficient calculation used to process Discrete Fourier Transform (DFT). When deciding among alternate implementation methods, the algorithm chosen should be considering the execution speed, utilization rate of the placed hardware and hardware complexity of the system. For real time systems, execution speed is the major concern.

Objective of this project is to design a hardware for computation of 8K (8192) points FFT/IFFT with a throughput of 34 Mega samples per second. Since Cooley Tuckey algorithm breaks the DFTs, so it can be combined with any other DFT algorithm hence implying stages of multiple techniques in a row to achieve the results. We have designed Mix-Radix Butterfly architecture which uses stages of Radix 2 and Radix 4 simultaneously to achieve high speed.

Table of Contents

List of Figures

List of Tables