EE321: Digital Signal Processing

Course Project

64-point FFT Processor

Dhruv Kumar(20021062)
Soumya Gupta (210102087)
Hrishikesh (2101020) November 15, 2023

Abstract

A Fast Fourier transform is an efficient algorithm to compute the discrete Fourier Transform (DFT). The operation has a high computational requirement of large number of operations (N^2 complex multiplications and $N(N-1)$ additions). This makes computational and implementation very difficult. Short length structures are can be obtain higher length FFT. In this project the proposed architecture is highrer order FFT and it is split into three stages and each stage is radix-2 based 4-point FFT to reduce the number of operations. Proposed architecture is implemented using verilog HDL XILINX Vivado 2019.1.

1 Introduction

The Discrete Fourier Transform (DFT) is one of the most important tools used in digital signal processing applications. It has been widely implemented digital communications such as Radars, Ultra wide band receivers (UWB) and many other applications. Computing this operation has high computational requirement and large number of operations (N^2 complex multiplications and $N(N-1)$ additions. This makes computing and implementation very difficult to realize. To reduce the number of operations a fast algorithm has been introduced by Cooley-Tukey [2] called Fast Fourier Transform (FFT). Later FFT reduces the computational complexity from O (N2) to O (NlogN). To reduce the complexity of FFT algorithm other researchers propose numerous techniques like radix-4 [2], split radix [3]. By using these two techniques we can able to avoid the radix-2 structure. These architectures are based on either Decimation in Time Domain (DIT) or Decimation in Frequency (DIF). Much other architecture was proposed on the basis of these architectures. In another way there is growing interest in the Field of Field Programmable Gate Arrays (FPGA). FPGA's have potentially substantially accelerated computational algorithms like FFT's. The Higher Order FFT's are implemented by using High-Cost FPGA's.

In this project we propore a combination logic for a Radix-4 64-point FFT processor. Memory is used just to store the input to be processed and output calculated. It has 3 stages, with each stage containing 16 4-point FFT blocks and 64 complex adders. 2 of these stages have 64 complex multipliers. Twiddle factors to be multiplied are stored as a lookup table and directly connected to one of the complex inputs of the multipliers.

2 Theory

This section includes mathematical basis of the FFT and the proposed architecture.

2.1 Mathematical Background

For a given sequence x of n samples, the Discrete Fourier transform (DFT) frequency components X (k) may be defined.

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{n.k}$$

where, $W_N = e^{\frac{-2j\pi}{N}}$ the twiddle factors, n and k are respectively the time and frequency domain indices. $0 \leq k \leq N-1, 0 \leq n \leq N-1$, N is the DFT length.

In the project, N equals 64. The 64-point FFT is constructed by 4-point FFTs. The FFT is splitinto 3 stages of 4-point FFTs. It can be represented mathematically by:

$$X[s+4q+16p] = \sum_{l=0}^{3}\sum_{m=0}^{3}\sum_{k=0}^{3} x[16l+4m+k]W_{64}^{(16l+4m+k)(s+4q+16p)}$$

More clear way to represent the 3 stages mathematically with coefficients to be multiplied can be given as follows:

$$X[k] = \sum_{n=0}^{N-1} x[n]W_N^{n.k}$$

Splitting the sum into 4 16-pint DFTs

$$X[k] = \left[\sum_{i=0}^{15} x[4i]W_{64}^{4jk}\right] + \left[\sum_{i=0}^{15} x[4i+1]W_{64}^{4jk}\right] W_{64}^{k} + \left[\sum_{i=0}^{15} x[4i+2]W_{64}^{4jk}\right] W_{64}^{2k} + \left[\sum_{i=0}^{15} x[4i+3]W_{64}^{4jk}\right] W_{64}^{3k}$$

$$X[k] = G[x] + H[k]W_{64}^{k} + I[k]W_{64}^{2k} + J[k]W_{64}^{3k}$$

where G[k], H[k], I[k] and J[k] are periodic with period equals 16. Splitting further each 16-point DFT into 4 4-point DFTs.

G[k]= $\left[\sum_{i=0}^{3} x[16i]W_{64}^{16jk}\right] + \left[\sum_{i=0}^{3} x[16i+4]W_{64}^{16jk}\right] W_{64}^{4k} + \left[\sum_{i=0}^{3} x[16i+8]W_{64}^{16jk}\right] W_{64}^{8k} + \left[\sum_{i=0}^{3} x[16j+12]W_{64}^{16jk}\right] W_{64}^{12k}$

H[k]= $\left[\sum_{i=0}^{3} x[16i+1]W_{64}^{16jk}\right] + \left[\sum_{i=0}^{3} x[16i+5]W_{64}^{16jk}\right] W_{64}^{4k} + \left[\sum_{i=0}^{3} x[16i+9]W_{64}^{16jk}\right] W_{64}^{8k} + \left[\sum_{i=0}^{3} x[16j+13]W_{64}^{16jk}\right] W_{64}^{12k}$

I[k]= $\left[\sum_{i=0}^{3} x[16i+2]W_{64}^{16jk}\right] + \left[\sum_{i=0}^{3} x[16i+6]W_{64}^{16jk}\right] W_{64}^{4k} + \left[\sum_{i=0}^{3} x[16i+10]W_{64}^{16jk}\right] W_{64}^{8k} + \left[\sum_{i=0}^{3} x[16j+14]W_{64}^{16jk}\right] W_{64}^{12k}$

J[k]= $\left[\sum_{i=0}^{3} x[16i+3]W_{64}^{16jk}\right] + \left[\sum_{i=0}^{3} x[16i+7]W_{64}^{16jk}\right] W_{64}^{4k} + \left[\sum_{i=0}^{3} x[16i+11]W_{64}^{16jk}\right] W_{64}^{8k} + \left[\sum_{i=0}^{3} x[16j+15]W_{64}^{16jk}\right] W_{64}^{12k}$

16 4-point DFTs in the brackets are periodic with period equals 4. The butterfly diagram for the given equations is given in the figure.

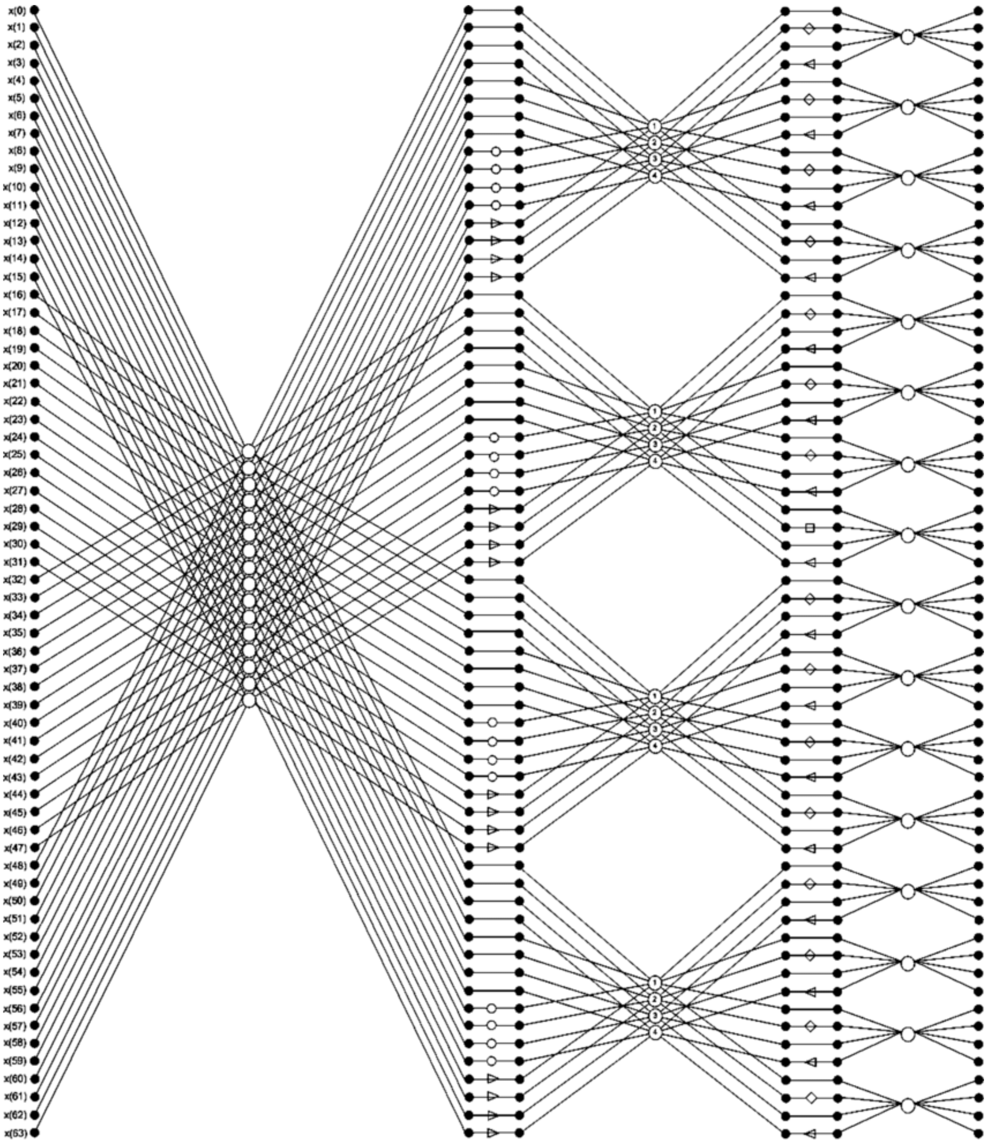

Figure 1

2.2 Proposed Architecture

2.2.1 Stage 1

1. The inputs are stored in the 64-inputs complex memory.

2. These 64 inputs go into 16 4-point FFT block in the pair of 4 as shown in the butterfly diagram.

3. Outputs if FFT blocks are multiplied with correct coefficients according to equations given in previous sections using 4 LUTs with 16 constants each (lookup tables).

2.2.2 Stage 2

1. The outputs of first stage fed into 16 4-point FFT blocks in the pair of 4 according to next stage of the butterfly diagram.

2. Outputs of these block are again multiplied with stage 2 coefficients stored in a big LUT of 64 constants.

2.2.3 Stage 3

1. Outputs are stored into the memory.

2. The values we get are the final FFT values.

2.3 Block Diagram

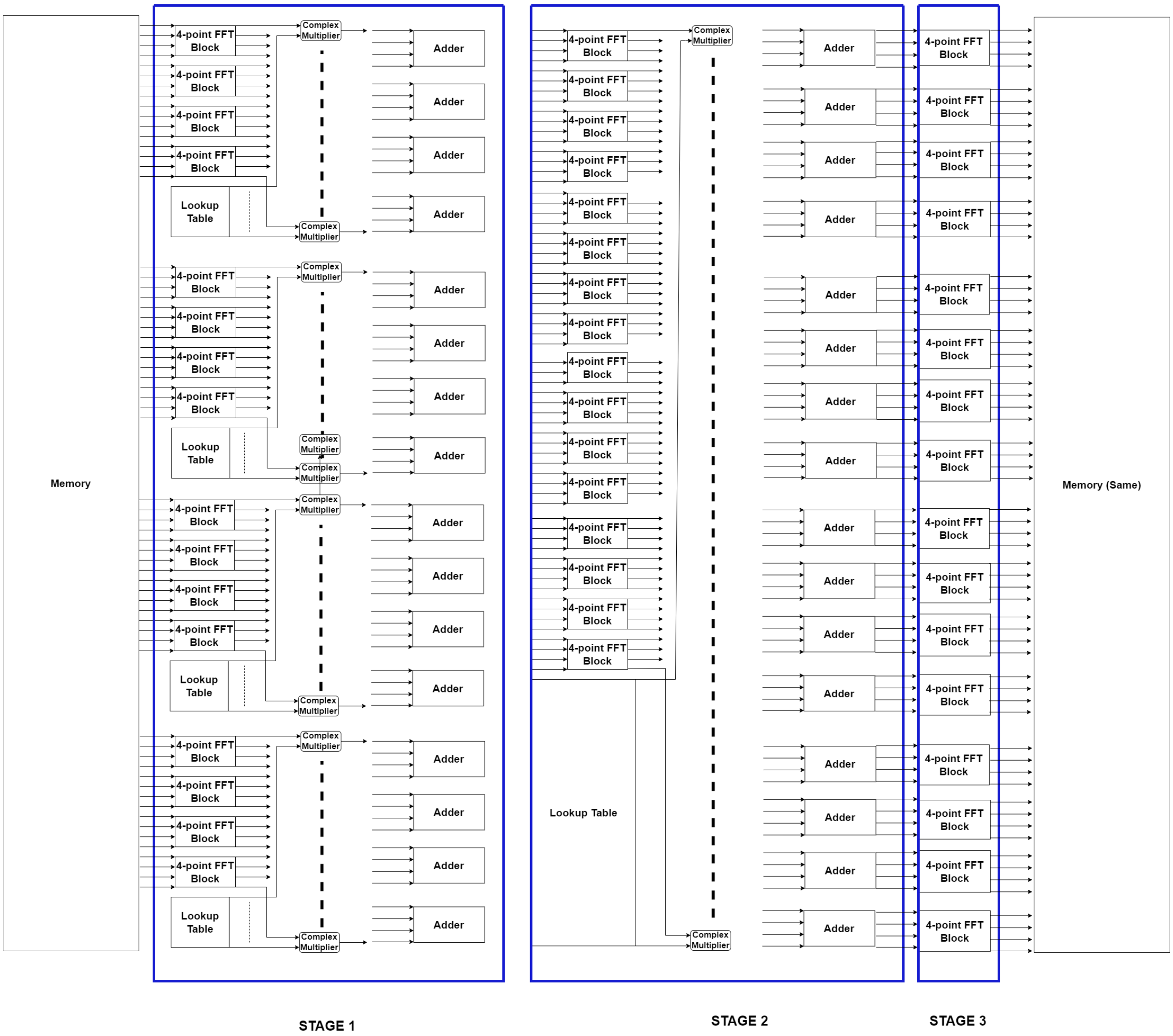

Figure 2