Design of Radix-4 and Radix-8 butterfly units using VHDL

1Jaishri Katekhaye, 2Amit Lamba, 3Vipin Bhure
1PG Scholar, 2Assistant Professor, 3Assistant Professor
1Electronics and Communication Engineering.

Abstract— Orthogonal Frequency Division Multiplexing (OFDM) is the promising solution for enhancing the data rates of wireless communication. Fast Fourier Transform (FFT) processor is one of the computationally complex modules of OFDM system. There are various algorithms available for designing the FFT processor viz., radix-2, radix-4, radix-8, split radix, mixed radix. Out of these, we have designed here butterfly units of radix-4 and radix-8. It is found that hardware required for single radix-8 butterfly is more as compare to the radix-4 butterfly. For simulation we used XILINX 14.2 ISE software and for coding we employed Very High Speed Integrated Circuit Hardware Description Language (VHDL).

IndexTerms— Radix-4, Radix-8, VHDL.

I. INTRODUCTION

Fast Fourier transform (FFT) are widely used in different areas of applications such as communications, radars, imaging, etc. One of the major concerns for researchers is to meet real-time processing requirements and to reduce hardware complexity mainly with respect to area and power and to improve processing speed.

Discrete fourier transform (DFT) is defined as

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

Where $X[k]$ and $x[n]$ are frequency domain and time domain sequences. To compute all N values DFT requires $N^2$ complex multiplications and N(N-1) complex additions. Since the amount of computation and thus the computation time, is approximately proportional to $N^2$, it will cost a long computation time for large values of N. For this reason, it is very important to reduce the number of multiplications and additions. This algorithm is an efficient algorithm to compute the DFT, which is called Fast Fourier Transform (FFT) algorithm.

A. Radix-4

The butterfly of a radix-4 algorithm consists of four inputs and four outputs Fig 1. The FFT length is 4M, where M is the number of stages. A stage is half of radix-2. The radix-4 DIF FFT divides an N-point discrete Fourier transform (DFT) into four N/4-point DFTs, then into 16 N/16-point DFTs, and so on. In the radix-2 DIF FFT, the DFT equation is expressed as the sum of two calculations. One calculation sum for the first half and one calculation sum for the second half of the input sequence. Similarly, the radix-4 DIF fast Fourier transform (FFT) expresses the DFT equation as four summations, then divides it into four equations, each of which computes every fourth output sample. The following equations illustrate radix-4 decimation in frequency. There are two types of algorithms Decimation-in-time (DIT) and Decimation-in-frequency (DIF).

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

Above formula is split into four summations:

$$X[k] = \sum_{n=0}^{N-1} x(n) W_N^{nk} + \sum_{n=0}^{2N} x(n) W_N^{nk} + \sum_{n=0}^{3N} x(n) W_N^{nk} + \sum_{n=0}^{4N} x(n) W_N^{nk}$$

$$X[k] = \sum_{n=0}^{N-1} x(n) W_N^{nk} + \sum_{n=0}^{N-1} x(n) W_N^{nk} + \sum_{n=0}^{N-1} x(n) W_N^{nk} + \sum_{n=0}^{N-1} x(n) W_N^{nk}$$

$$= \sum_{n=0}^{N-1} x(n) \left[ W_N^{(n+N/4)k} + x \left( n + \frac{N}{4} \right) W_N^{(n+N/2)k} + x \left( n + \frac{3N}{4} \right) W_N^{(n+N/2)k} \right]. W_N^{nk}$$

The three twiddle factor coefficients can be expressed as follows:
Equation (2) can thus be expressed as:

$$X(k) = \sum_{n=0}^{N-1} \left[ x(n) + (-1)^n x(n+N/4) + (-1)^n x(n+N/2) + (-1)^n x(n+3N/4) \right] W_N^k$$

(3)

To arrive at four point DFT decomposition, let $W_N = W_{N/4}$. Then equation (3) can be written as four N/4 point DFTs

$$X(4k) = \sum_{n=0}^{N/4-1} \left[ x(n) + x(n+N/4) + x(n+N/2) + x(n+3N/4) \right] W_{N/4}^k$$

$$X(4k+1) = \sum_{n=0}^{N/4-1} \left[ x(n) - x(n+N/4) - x(n+N/2) + x(n+3N/4) \right] W_{N/4}^k W_{N/4}^n$$

$$X(4k+2) = \sum_{n=0}^{N/4-1} \left[ x(n) - x(n+N/4) - x(n+N/2) + x(n+3N/4) \right] W_{N/4}^k W_{N/4}^n$$

$$X(4k+3) = \sum_{n=0}^{N/4-1} \left[ x(n) + x(n+N/4) + x(n+N/2) - x(n+3N/4) \right] W_{N/4}^k W_{N/4}^n$$

(4)

for k=0 to N/4

X(4k), X(4k+1), X(4k+2) and X(4k+3) are N/4 -point DFTs. Each of their N/4 points is a sum of four input samples $x(n)$, $x(n+N/4)$, $x(n+N/2)$ and $x(n+3N/4)$, each multiplied by either +1, -1, j, or -j. The sum is multiplied by a twiddle factor ($W_{N0}$, $W_{Nn}$, $W_{N2n}$ and $W_{N3n}$). The four N/4 -point DFTs together make up an N-point DFT. Each of these N/4 -point DFTs is divided into four N/16 -point DFTs. Each N/16 DFT is further divided into four N/64 -point DFTs, and so on, until the final decimation produces four-point DFTs. The four-point DFT equation makes up the butterfly calculation of the radix-4 FFT. A radix-4 butterfly is shown graphically in fig 1.

The SFG for radix-4 is derived from Eq.4

### B. Radix-8

![Figure 1. Radix-4 DIF FFT butterfly](image)

![Figure 2. SFG of the radix-4 butterfly](image)
Radix-8 algorithm has eight inputs and eight outputs. It operates on the DFT equation and divides it into eight N/8 point DFTs. By using the FFT algorithm the computational complexity reduces to $N \log_r N$, where $r$ represents the Radix-$r$ FFT. The Radix-$r$ FFT can easily derived from DFT by decomposing the N point DFT into a set of recursively related $r$-point transform and $x(n)$ is powers of $r$. In Radix-$r$ algorithm the $r$ is 8. The Radix-8 Decimation-In-Time and Decimation-In-Frequency Fast Fourier Transform (FFTs) gain their speed by reusing the results of smaller, intermediate computations to compute multiple DFT frequency outputs. The following equation illustrate radix-8 DIF.

$$X[k] = \sum_{n=0}^{N-1} x(n) W_N^{kn} + \sum_{n=0}^{N-1} x\left(n + \frac{N}{8}\right) W_N^{k(n+\frac{N}{8})} + \sum_{n=0}^{N-1} x\left(n + \frac{2N}{8}\right) W_N^{k(n+\frac{2N}{8})} + \sum_{n=0}^{N-1} x\left(n + \frac{3N}{8}\right) W_N^{k(n+\frac{3N}{8})} + \sum_{n=0}^{N-1} x\left(n + \frac{4N}{8}\right) W_N^{k(n+\frac{4N}{8})} + \sum_{n=0}^{N-1} x\left(n + \frac{5N}{8}\right) W_N^{k(n+\frac{5N}{8})} + \sum_{n=0}^{N-1} x\left(n + \frac{6N}{8}\right) W_N^{k(n+\frac{6N}{8})} + \sum_{n=0}^{N-1} x\left(n + \frac{7N}{8}\right) W_N^{k(n+\frac{7N}{8})}$$

(5)

The basic butterfly of radix-8 is shown in following figure:

![Figure 3. Radix-8 butterfly unit](image)

From Eq.1, let $x = [x(0), x(1), ... , x(N-1)]^T$ be the input signal sequence and $X = [X(0), X(1), ... , X(N-1)]^T$ be the output signal sequence. Then

$$X = FN \times x$$

where $FN$ is the $N \times N$ DFT matrix whose elements are defined by $[FN]_{ij} = W_N^{ij}$, $i, j \in [0, N-1]$. For radix-8 FFT $FN=F8$.

The signal flow graph (SFG) of a butterfly implementing $F8$ is shown in Fig. 4. Note that all the signals in the SFG are complex valued. From the above matrix SFG is derived for radix-8. We have divided SFG into three stages and the output of third stage is multiplied with twiddle factors to get the final output.

Figure below shows the radix-8 SFG.
Following figures show the simulation results of radix-4 and radix-8 butterfly units.

Figure 5: RTL view of radix-4 butterfly

Figure 6: Output of radix-4 butterfly

Table 1: Device utilization summary for radix-4

<table>
<thead>
<tr>
<th>Device Utilization Summary (estimated values)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic Utilization</td>
<td></td>
</tr>
<tr>
<td>Used</td>
<td>Available</td>
</tr>
<tr>
<td>6535</td>
<td>303601</td>
</tr>
<tr>
<td>0</td>
<td>4935</td>
</tr>
<tr>
<td>556</td>
<td>701</td>
</tr>
<tr>
<td>1</td>
<td>32</td>
</tr>
</tbody>
</table>

Figure 7: RTL view of radix-8 butterfly
Figure 8: Output of radix-8 butterfly

Table 2: Device utilization summary of radix-8

<table>
<thead>
<tr>
<th></th>
<th>Used</th>
<th>Available</th>
<th>Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Slice LUTs</td>
<td>1263</td>
<td>90870</td>
<td>0%</td>
</tr>
<tr>
<td>Number of fully used LUT-FF pairs</td>
<td>0</td>
<td>1263</td>
<td>0%</td>
</tr>
<tr>
<td>Number of bonded EDSs</td>
<td>544</td>
<td>702</td>
<td>77%</td>
</tr>
<tr>
<td>Number of I/O Buffers</td>
<td>1</td>
<td>32</td>
<td>3%</td>
</tr>
</tbody>
</table>

Table 3: Comparison of radix-4 and radix-8 butterfly

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Radix-4</th>
<th>Radix-8</th>
</tr>
</thead>
<tbody>
<tr>
<td>32-bit adder</td>
<td>84</td>
<td>195</td>
</tr>
<tr>
<td>32-bit subtractor</td>
<td>11</td>
<td>33</td>
</tr>
<tr>
<td>1-bit register</td>
<td>256</td>
<td>512</td>
</tr>
<tr>
<td>32-bit 2-to-1 multiplexer</td>
<td>84</td>
<td>196</td>
</tr>
</tbody>
</table>

III. CONCLUSION
In this paper, we have proposed basic butterfly unit of Radix-4 and Radix-8. Single unit of radix-4 butterfly requires 84 adders and 11 subtractors whereas radix-8 butterfly requires 195 adders and 33 subtractors. From table 3, it is clear that radix-8 butterfly requires more hardware than radix-4 butterfly. The simulation is done by Xilinx synthesis tool. The waveforms are displayed by XILINX ISE design suit 14.2.

REFERENCES