DESIGN AND IMPLEMENTATION OF FFT ARCHITECTURE FOR REAL-VALUED SIGNALS BASED ON RADIX-2³ ALGORITHM

Pradnya Zode, A.Y. Deshmukh and Abhilesh S. Thor
1,3Assistant Professor, Yeshwantrao Chavan College of Engineering
2Professor, Department of Electronics Engg., G.H. Raisoni College of Engineering, Nagpur
(E-mail: 1 pradnya.u@rediff.com, 2aydeshmukh@gmail.com, 3abhileshthor@yahoo.com).

Abstract- A new FFT architecture for real-valued signal is proposed using Radix-2³ algorithm. It is based on modifying flow graph of the FFT algorithm such that it has both real and complex datapaths. A redundant operation in flow graph is replaced by imaginary part. Using folding technique RFFT architecture with any level of parallelism can be achieved. This RFFT architecture will lead to low hardware complexity as compare to radix-2 and radix 2² algorithm in terms of adder, multiplier and delay. N-point 2 parallel radix-2³ architecture requires (log₂N-1) complex multiplier, 2log₂N adders, 3N/2-2 delays. RFFT which is used for real time applications and in portable devices for which low power consumption is main requirement, so accordingly carry propagate adder which has least power consumption and CSD multiplier is selected for our proposed architecture.

Keywords— FFT, Parallel Processing, Pipelining, Real Signals, radix-2³, Folding

1. Introduction

Fast Fourier transform (FFT) is one of the widely used algorithms in digital signal processing [1]. Now a day’s interest in the computation of FFT for real valued signals (RFFT) is increased since most of the physical signals are real. RFFT is very important algorithm used in various real time applications. In the area of digital signal processing (DSP) [2]. FFT is very important algorithm. Hardware complexities can be reduce in asymmetric digital subscriber line (ADSL) [3] by using RFFT. In applications like spectral and filtering analysis [4] FFT also plays an important role which helps to analyze spectral components. RFFT is very vital algorithm for analyzing signals like electroencephalography (EEG) and electrocardiography (ECG). RFFT is also used in various portable devices which leads to low power consumption. FFT is also used in power spectral density which can detect whether signal is perfect or there is any problem. This paper tells about designing RFFT architecture. In this flow graph is modified and redundant part is replaced by imaginary part in order to reduce complexity. Using folding technique RFFT architecture with any level of parallelism can be achieved. As imaginary part is injected in butterfly structure it will have both real and complex datapath. RFFT is used for real time applications and in portable devices for which low power consumption is main requirement, so accordingly carry propagate adder which has least power consumption and CSD multiplier is selected for our architecture.

The paper is organized as follows. Section II describes previous work related to RFFT. Section III describes proposed architecture for 16 point RFFT radix-2³ DIF. Section VI describes FPGA Implementation of adder and CSD multiplier. The Experimental results are discussed in section V and finally, concluding remarks are in section VI.

2. Previous Work

Previously, in the past various algorithms for computation of RFFT is presented but they did not have proper regular geometry. This is very important for deriving pipelined architecture. Firstly pipelined architecture for real valued signal was designed in [5]. But it is restricted to only radix-2 and 4 parallel RFFT architecture. Also, it has only real datapaths. To derive FFT architecture for real and complex inputs a design is presented in [6].But in this architecture even after removing redundant operations, it still calculate this samples. Again there was no full hardware utilization of architecture derived in this paper.

In [7] pipelined RFFT architecture is derived but it has more hardware complexity than our proposed architecture as it has more number of adders, multipliers and delays than radix-2³ algorithm in [8].
3. Proposed Work

The N-point discrete Fourier transform (DFT) of a sequence \( x[n] \) is defined as:

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

Where \( W_N = e^{-j(2\pi/N)} \)

In RFFT inputs are real. If \( x[n] \) is real, then output \( X[k] \) have conjugate symmetric

\[
X[N-k] = X^*[k] \\
\text{(1)}
\]

Due to this, \( (N/2)-1 \) output calculations can be removed as they are redundant. Proposed work involves following 2 steps.

A. Modified butterfly structure

Redundant samples can be find by approach in [5]. After finding redundant samples these are removed. But it leads to irregular geometry, so efficient pipelining cannot be done. In order to get regular geometry redundant operations are replaced by imaginary part. Now efficient pipelining can be done. Modified flow graph of 16 point RFFT DIF radix-2^3 shown in Fig.1

II. UNITS

Since flow graph has both complex and real parts, it has both complex and real datapath. So to handle this we have two butterfly structures. First is very straight it involves two real inputs and consists of real adder and subtraction. This butterfly structure is shown in Fig.2. Butterfly structure is shown in Fig.2

Fig. 1 Modified flow graph 16 point RFFT radix-2^3 DIF

B. Folding.

Using folding technique in [9] pipelined architecture can be derived from DFG. Also it leads to optimized datapath [10]. Nodes which are in DFG can be implemented by butterfly structure I and II. Proposed work 2 parallel architecture shown below in Fig. 4

Fig. 2 Butterfly structure I for proposed architecture [8]

Fig. 3 Butterfly structure II for proposed architecture [8]

Fig. 4 Proposed work

4. FPGA IMPLEMENTATION

In this various adders are simulated on Spartan 6. Xc6sl16 device, CSG324 package and adder with least power power consumption is selected and also CSD multiplier is simulated on Spartan-6.

Adder

RFFT is generally used for for real time applications and portable devices like ECG and EEG etc. Low power and low area is main requirement for such devices so, we have select adder accordingly. So various adders like carry propagate as in [11], carry skip as in [12] and carry look ahead adder as in [13] is simulated on Spartan 6. Xc6sl16 device, CSG324 package. Table I shows their performance based on area, power and delay. Amongst these carry propagate adder suits best as it has least power consumption. Also its RTL view shown in Fig.5 and waveform in Fig.6.
multiplier has multiplexer which is controlled by these pair of bits. Depending on these input pair, multiplexer in multiplier output will be input data, inverse of input data or all zeros. Depending on these pair of bits shifts are applied. It also has circuitry which receives and combines these outputs and further shifts of bits are applied to generate final multiplier output. Normal multiplier and CSD[16] multiplier(8 bit x 8 bit) is simulated on Spartan 6, Xc6sl16 device, CSG324 package and their performance based on area, power and delay is shown in Table II.

Clearly it is seen that CSD multiplier has less delay and power consumption, so it is selected for our proposed architecture. Its waveform shown in Fig.7 and RTL view in Fig.8.

### TABLE I

<table>
<thead>
<tr>
<th>Adder(16bit)</th>
<th>Power (mw)</th>
<th>Delay (ns)</th>
<th>Area (slices)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carry propagate</td>
<td>30</td>
<td>15.243</td>
<td>25 / 9112</td>
</tr>
<tr>
<td>Carry look ahead</td>
<td>42</td>
<td>13.567</td>
<td>59</td>
</tr>
<tr>
<td>Carry skip</td>
<td>37</td>
<td>14.767</td>
<td>44</td>
</tr>
</tbody>
</table>

### TABLE II

<table>
<thead>
<tr>
<th></th>
<th>Power (mw)</th>
<th>Delay (ns)</th>
<th>Area (slice)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal multiplier</td>
<td>35</td>
<td>11.7</td>
<td>116 / 9112</td>
</tr>
<tr>
<td>CSD multiplier</td>
<td>32</td>
<td>10.5</td>
<td>88</td>
</tr>
</tbody>
</table>

**B. Multiplier**

In normal multiplier, multiplication is done by shifts producing partial products and then adding all the partial products. So multiplication of two ‘N’ bits will generate N x N partial products and subsequently (N-1) adders will require if two inputs ‘N’ bit adders are used. So number of hardware component increases and also time required for multiplication increases. So there is need of multiplier which generates less partial products which in turn reduces time as well as power consumption for multiplication. CSD multiplier suits best to our requirement.

Proposed CSD multiplier is very efficient way of multiplication; it leads to reduction in number of partial products by using redundancy of sign code CSD multiplier in [14] and provides an efficient way of multiplication as in [15]. Number of partial products also gets reduced. As number of partial products are reduced, it is more hardware efficient, so it require less time for multiplication and low power consumption than normal multiplier. Constant value with which multiplicand is to be multiplied has pair of bits,
C. Simulation result of FFT architecture

Fig. 9 shows inputs to FFT and Fig. 10 shows outputs of FFT.

![Fig. 9 Inputs to FFT](image)

![Fig. 10 Outputs of FFT](image)

will have low hardware as compared to other previous designs.

### TABLE III
CONSIDER N=64

<table>
<thead>
<tr>
<th>Complex multiplier</th>
<th>Adders</th>
<th>Delays</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radix-2 [8]</td>
<td>2(log2 N - 1) + 4 C.M</td>
<td>4log2 N + 24 adders</td>
</tr>
<tr>
<td>Radix-2^2 [8]</td>
<td>(log2 N - 1) + 2 C.M</td>
<td>4log2 N - 2 adders</td>
</tr>
<tr>
<td>Proposed Radix 2^3  (2 parallel)</td>
<td>(log8 N - 1) + 1 C.M</td>
<td>2log2 N + 12 adders</td>
</tr>
</tbody>
</table>

6. Conclusion

Efficient architecture for computation of RFFT has been proposed in this paper. Datapaths will be optimized by folding. It will also have less number of adders, multiplier and delay with respect to previous architecture. This architecture will have low power with respect to adder as carry propagate adder will be used and relatively fast as CSD multiplier will be used.

References


