DESIGN OF A HIGH SPEED MULTIPLIER LESS RADIX 2^K FFT ARCHITECTURE FOR HIGH RATE OFDM APPLICATIONS

TALLURI NIRMALA, M.SRIKANTH, SHANMUKACHAITANYA

152W1D6812 GVR&S College of Engineering and Technology
2 Asst.Prof. Dept.Of ECE GVR&S College of Engineering and Technology
162W1D6806 GVR&S College of Engineering and Technology

E-mail: 1nirmala1914@gmail.com., 2srikanthmuvva695@gmail.com, 3shanmukachaitanyal6@gmail.com

Abstract- In this paper, we propose the design of a high speed reconfigurable CORDIC based FFT architecture which is a key component in the Orthogonal Frequency Division Multiplexing modulation scheme. FFT has been used in the sub carrier mapping layer which is main driving force in high rate WPANs. This proposed architecture consists of radix-2^k where twiddle factor multiplications is optimized using pre processing steps. In particular, this generalized radix-2^k FFT architecture uses only three multiplication unit for complex multiplications and to replace the multiplication units by shift and add units using CORDIC. Thus, the proposed architecture is built with reduced hardware requirements and less number of complex multiplications in computing FFT. The twiddle factor memory is also reduced thereby increasing the speed of the computation.

Index Terms- FFT (Fast Fourier transform), radix-2^k algorithm, CORDIC, CSD multiplication.

I. INTRODUCTION

The present day real time applications demand, the FFT to be calculated not only with high throughput rates, sometimes it is desirable to achieve low power FFT computations. These high performance requirements appear in applications such as wireless local area network, OFDM and real time video streaming services in short range indoor environments. The FFT/IFFT processor has a high hardware complexity since large number multiplications are involved. By applying parallel computation into the butterfly operations, this brief improves the throughput rate of the previously proposed FFT architectures by a factor of 2. High processing speed leads to reduced number of required delay elements since fewer intermediate results need to be stored. But it will lead latency reduced by a factor of 2. The radix of the algorithm greatly influences the architecture of the FFT processor and the complexity of the implementation. A small radix is desirable because it results in a simple butterfly. Nevertheless, a high radix reduces the number of twiddle factor multiplications. The radix 2^k algorithms simultaneously achieve a simple butterfly and a reduced number of twiddle factor multiplications. The radix-2 algorithm is a well known simple algorithm for FFT processors, but it requires many complex multipliers. Recently various radix 2^k FFT algorithms and architectures have been studied in order to reduce the number of complex multipliers.

In this brief, a reconfigurable FFT architecture to compute FFT using radix-2^k algorithm with the CORDIC based rotational unit is proposed. The key ideas for achieving high date throughput and with reduced hardware complexity are described. The power consumption and hardware cost can be saved in our processor by using the higher radix FFT algorithm and less memory and multiplier less complex multiplications.

The organization of this brief is as follows. Section II describes the radix 2^k FFT algorithm, and Section III describes the proposed CORDIC radix 2^k FFT architecture. In Section IV, the implementation and comparison are presented. Finally, conclusions are provided in Section V.

II. RADIX -2^K ALGORITHM

The N-point DFT is formulated as

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk}, k = 0, 1, ..., N-1 \quad (1)$$

Where the twiddle factors is defined as

$$W_N^{nk} = e^{-j \frac{2\pi nk}{N}}$$

Then denotes the time index and the k denotes the frequency index. The radix 2^k algorithm can be derived by integrating twiddle factor decomposition through a divide and conquer approach.

A. Radix -2^2 Algorithm

Consider the first two steps of decomposition in radix-2 DIF FFT together. Applying a 3-dimensional linear index map as follows

$$n = \frac{N}{2}n_1 + \frac{N}{4}n_2 + n_3 \{n_1, n_2 = 0,0, n_1 = 0,0, n_2 = 0,0, n_3 = 0,0, n_3 = 0, N - 1\} \quad (2)$$

$$k = k_1 + 2k_2 + 4k_3 \{k_1, k_2 = 0,0, k_1 = 0,0, k_2 = 0, N - 1\}$$

Design of a High Speed Multiplier Less Radix 2^K Fft Architecture For High Rate OFDM Applications
The DFT has the form
\[ X(k_1 + 2k_2 + 4k_3) = \sum_{n_0, n_1, n_2, n_3=0}^{N-1} x(n_0, n_1, n_2, n_3) W_N^{n_0k_1 + n_1k_2 + n_2k_3 + n_3k_4} \]

\[ = \sum_{n_0=0}^{N-1} \sum_{n_1=0}^{N-1} \left\{ B_N^{n_0}(n_1, n_2, n_3) \right\} W_N^{n_0k_1 + n_1k_2 + n_2k_3 + n_3k_4} \]  

(3)

where the first butterfly structure has the form
\[ B_N^{n_0}(n_1, n_2, n_3) = R(n_0, n_1) \]  

(4)

Decomposing the composite twiddle factor, it can be expressed in Eq.(5).
\[ W_N^{n_0k_1 + n_1k_2 + n_2k_3 + n_3k_4} = \left( -j \right)^{n_0k_1 + n_1k_2 + n_2k_3 + n_3k_4} W_N^{n_0k_1} W_N^{n_1k_2} W_N^{n_2k_3} W_N^{n_3k_4} \]  

(5)

Substituting the Eq.(5) into Eq.(3) and expanding the summation with regard to index \( n_2 \), we have a set of 4 DFTs of length \( N/4 \).
\[ X(k_1 + 2k_2 + 4k_3) = \sum_{n_0=0}^{N/4-1} \left\{ H_N^{k_1k_2k_3}(n_0) \right\} W_N^{n_0k_1} W_N^{n_1k_2} W_N^{n_2k_3} W_N^{n_3k_4} \]  

(6)

where a secondary butterfly structure \( H_N^{k_1k_2k_3}(n_0) \) is expressed as
\[ H_N^{k_1k_2k_3}(n_0) = B_N^{k_0}(n_0) \left( -1 \right)^{n_0} B_N^{k_0}(n_3) \]  

(7)

After these two columns, full multiplications are used to apply the decomposed twiddle factor \( W_N^{n_0k_1 + n_1k_2k_3} \) in Eq.(6). Applying this cascade decomposition recursively to the remaining DFTs of length \( N/4 \) in Eq.(6), the complete radix-2 FFT algorithm is obtained. Equation (7) represents the first two columns of butterflies with only trivial multiplication of \(-j\) which can be implemented using only real-imaginary swapping and sign inversion.

The radix-2\(^2\) algorithm is characterized according to the merit that it has the same multiplicative complexity as the radix-4 algorithm but still retains simple structures of the radix-2 butterfly.

### III. BINARY CONVERSION

Many techniques have been used to efficiently convert this floating point (sine and cosine values as a twiddle factors) values into binary representation for digital implementation. Then only we can implement FFT in VLSI.

The two ways of floating point to binary conversion are

1. Both integer and fractional part is converted separately by repeatedly multiply 2, and considers each one bit as it appears left of the decimal.
2. Representing the floating number using IEEE 754 format (single or double precision).

![Fig 1. CORDIC rotational unit](image)

**A. FFT twiddle factors coefficients:**

The FFT equations given in equation (1) can be split into two matrices, as shown in Fig 1 and the complex sequences multiply the rotation factors in radix-2 calculation. In order to avoid the use of multipliers, the design uses the CORDIC algorithm with pipelined structure which only requires simple adders and shifters.

From the equations (3) and (4), it can be stated that the FFT operation involves multiplication of various sine and cosine coefficients with a fixed input sequence. Hence sub structure sharing technique is used to reduce the number of operators [6]. The cosine basis is quantized to 8-bits for energy efficiency. The cosine coefficients are represented as CSD number which has the advantage of reduced number of one’s compared to the binary representation. The cosine basis is chosen up to four decimal places and each one is represented as 7 bit binary number. The number of bits has an impact on...
the quality of the system. The values of the cosine basis are shown in the Table below.

<table>
<thead>
<tr>
<th>Basis</th>
<th>Real Value</th>
<th>Binary/CSD*</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0.4904</td>
<td>0100 0000,1*</td>
</tr>
<tr>
<td>b</td>
<td>0.4619</td>
<td>0100 1011,1*</td>
</tr>
<tr>
<td>c</td>
<td>0.4157</td>
<td>0111 0101</td>
</tr>
<tr>
<td>d</td>
<td>0.3536</td>
<td>0101 1101</td>
</tr>
<tr>
<td>e</td>
<td>0.2778</td>
<td>0101 0100</td>
</tr>
<tr>
<td>f</td>
<td>0.1913</td>
<td>0001 1000</td>
</tr>
<tr>
<td>g</td>
<td>0.0975</td>
<td>0000 1100</td>
</tr>
</tbody>
</table>

Table I sine and cosine basis Set

The stronger operator, multiplication is transformed to simple shift and adds operations by applying Horner’s rule. This reduces the power consumption. For example, consider the cosine coefficients c and g,

\[
c \ast X = \frac{2^5 + 2^4 + 2^2 + 1}{2^7} \ast (X) = (2^5 \cdot 3 + 5) \ast (X)\]

\[
g \ast X = \frac{2^3 + 2^2}{2^5} \ast (X) = 2^2 \ast (X)\]

and the common terms they share is 3X. The common terms among the cosine basis are 1X, 3X, 5X, and -1X and are shared to compute the partial outputs.

Fig. 2 MATLAB results comparison report

IV. PERFORMANCE RESULTS

The design is scripted as a verilog HDL file and synthesized using QUARTUS II 9.0 v. The design is synthesized into Cyclone device. The FFT sample outputs are compared with MATLAB results as a values are stored in a text file. The text files for angle rotations is accessed by the Modelsim ALTERA and the corresponding FFT coefficients are calculated. These values are then fed to the FFT module which returns the DIF data sequence. The simulated results are shown below.

Fig. 3 Model sim output

The proposed FFT processor uses constant multipliers based on the canonical signed digit (CSD) representation for the complex multiplication arithmetic in appropriate stages. Mostly the existing research using complex multipliers for the twiddle factor multiplication. However, in this design, the cordinalional CORDIC has been used for the twiddle factor multiplication.

The constant multiplier using the CSD technique is implemented using the common calculation patterns X1, X2, and X3. The proposed FFT processor applied CORDIC instead of complex multiplier at several stages. Thus, the hardware complexity of complex multiplier is decreased by at least 50% in comparison with using complex Booth multiplier. In addition, the twiddle factor LUT size is reduced compared to the designs using the complex multipliers.

Fig. 4 Hardware Implementation of Shift/Add Module

Table II Hardware complexity report comparison using cyclone III family devices
CONCLUSION

In this paper, a radix $2^k$ algorithm and CORDIC based reconfigurable radix $2^k$ FFT architecture have been proposed for OFDM-based WPAN applications. The number of multipliers required for complex multiplications is greatly reduced using preprocessing unit twiddle factors are computed using LUT less come multiplier less CORDIC algorithm. The proposed radix $2^k$ FFT processor is the most area-efficient architecture for the DIF SDF FFT processors. The proposed architecture has potential applications in high-rate OFDM-based WPAN systems.

REFERENCES