An Efficient and Area Optimized Fused FFT Processor for High end Transceivers

K. DEEPTHI VATSALYA\textsuperscript{1}, P. SUDHAKAR RAO\textsuperscript{2}

\textsuperscript{1}PG Scholar, Dept of VLSI\&ES, Nova Engineering College, Vijayawada, AP, India, E-mail: deepthielitesu@gmail.com.
\textsuperscript{2}Assoc Prof \& HOD, Dept of VLSI, Nova Engineering College, Vijayawada, AP, India, E-mail: sudhavlsi718@gmail.com.

Abstract: This project presents a FUSING FFT system using OFDM application. It is demonstrated by a software reconfigurable OFDM system using a programmable floating point DSP. A new VLSI architecture for real-time pipeline FFT processor is proposed in this project. In this project, high radix floating point butterflies are implemented more efficiently with the two fused floating-point operations. The fused operations are a two-term dot product and add-subtract unit. Both discrete and fused radix processors are implemented; compared in regarded with area wise. OFDM systems and the associated clock cycles required to demodulate data using loop and straight-line FFT programming methods are provided. Higher execution speed is achieved by using straight-line code instead of looped code. Along with that, area optimization also possible in this project. By introducing A.P.C \(\frac{1}{2}\) rows in LUT can be decreased. Further by introducing O.M.S. more half area can be reduced. Implies 75\% area can be reduced by using these two techniques. The tradeoff of this optimization is a larger program memory requirement of the straight-line assembly code.

Keywords: Fusing, OFDM, FFT, Radix, Dot Product, Folding Transformation, Optimization, Complex Valued Fourier Transform, Add-Subtract Unit, APC, OMS, LUT.

I. INTRODUCTION

OFDM is a multimode modulation and multiple access technique used in a number of commercial wired and wireless applications. In the wired side, it is used for a variant of digital subscriber line (DSL). For wireless, OFDM is the basis for several television and radio broadcast applications, including the European digital broadcast television standard, as well as digital radio in North America. FUSING platform provides software control of variety of modulation schemes, wideband or narrow band operation, communications security functions such as frequency hopping and waveform requirements of current and evolving standards over a broad frequency range. It is viewed as a single radio platform, providing services to multiple cellular standards. Fast Fourier Transform (FFT) is widely used in the field of digital signal processing (DSP) such as filtering, spectral analysis, etc., to compute the discrete Fourier transform (DFT). FFT plays a critical role in modern digital communications such as digital video broadcasting and orthogonal frequency division multiplexing (OFDM) systems. Much research has been carried out on designing pipelined architectures for computation of FFT of complex valued signals (CFFT).

Various algorithms have been developed to reduce the computational complexity, of which Cooley-Tukey radix-2 FFT [1] is very popular. Note that this is not the only way to represent floating point numbers, it is just the IEEE standard way of doing it. Here is what we do: the representation has three fields:

\[ |S| \quad |E| \quad |F| \]

(1)

S is one bit representing the sign of the number, E is an 8-bit biased integer representing the exponent, F is an unsigned integer.

The decimal value represented is:

\[ S \times 2^{(E - bias)} \times f \times 2 \]

(2)

Where \( e = E - bias \)

\[ f = \frac{F}{2^{(n \times bias)}} + 1 \]

(3)

for single precision representation (the emphasis in this class) \( n = 23 \) bias = 127 for double precision representation (a 64-bit representation) \( n = 52 \) (there are 52 bits for the mantissa field) bias = 1023 (there are 11 bits for the exponent field).

II. FUSED FLOATING-POIN T ADD-SUBTRACT UNIT

The floating-point fused add-subtract unit (Fused AS) performs an addition and a subtraction in parallel on the same pair of data. The fused add-subtract unit is based on a conventional floating point adder [8]. Although higher speed adder designs are available (see [9] for example), the basic design shown here serves to demonstrate the concept. A block diagram of the fused add and subtract unit is shown in Fig.1 (after the initial design from [10]). Some details,
such as the LZA and normalization logic are omitted here to simplify the figure. The exponent difference calculation, signific and swapping, and the signific and shifting for both the add and the subtract operations are performed with a single set of hardware and the results are shared by both the operations. This significantly reduces the required circuit area. The signific and swapping and shifting is done based solely on the values of the exponents (i.e., without comparing the significands).

\[ A = \{A_0, A_1, A_2, A_3, A_4, A_5, A_6, A_7\} \]  
\[ B = \{B_4, B_5, B_6, B_7, B_8, B_9, B_{10}, B_{11}\} \]  
\[ C = \{C_2, C_3, C_4, C_5, \phi, \phi, C_6, C_7\} \]  
\[ D = \{D_1, D_2, \phi, D_4, D_5, \phi, D_0\} \]

The nodes from \(A_0\) to \(A_7\) represent the eight butterflies in the first stage of the FFT and \(B_0\) to \(B_7\). Represent the butterflies in the second stage. Assume the butterflies have only one multiplier at the bottom output instead of both outputs. First, the proposed design provides normal-order FFT output. The feature of normal-order FFT output eliminates the extra re-ordering buffer needed for MDF pipelined architectures. To our best knowledge, currently there are no MDF pipelined FFT processors with normal-order output capability in the literatures. Even in continuous-flow mode, the proposed design only requires a size-memory space, while MDF pipelined architectures require a memory space of size close to, including a memory space of size inside PE, and for alternating re-ordering buffer and output.

**III. HIGH THROUGHPUT FFT ARCHITECTURE**

The proposed architecture consists of the following main parts, together with their specific novelties and advantages.

1. A memory unit composed of 16 dual-port memory banks, which facilitates 16-way parallel data access.
2. A memory bank index and address generation unit (BAGU), which generates conflict-free and in-place memory bank indexes and address for the radix-16 FFT operation.
3. Four commutator blocks located in front of the input side and after the output side of the memory, provide efficient data routing mechanism which is governed by the BAGU signals.
4. A scaling unit (SU) coordinates controlled scaling operations for block floating point (BFP) operations, which generates higher signal-to-quantization noise ratio (SQNR) than the existing designs.
5. The kernel processing engine, which is a high performance computing engine for radix-16 butterfly operations. Four radix-16 PEs (i.e., PE_R16 0 through PE_R16 3), two sets of radix 2 PEs (each set contains four radix-2 PEs), and four sets of complex multipliers (each contains four complex multipliers) for twiddle factor multiplications. Those multipliers are optimized with the help of common-sub expression sharing technique and a new twiddle-factor multiplication scheme. All the function units inside the kernel processing engine are detailed. To avoid possible conflicts in simultaneously reading (or writing) 16 data from (or to) the memory banks during FFT operations, a proper memory addressing scheme is necessary.

The well-known non-conflict memory addressing schemes [5], [7] are only applicable to radix-2 FFT algorithm. Although the addressing scheme in [6] is for general radix-
An Efficient and Area Optimized Fused FFT Processor for High end Transceivers

FFT operations, its FFT size should be a power-of-number. Besides, those schemes are only limited to single-PE architecture. On the other hand, the radix-2 addressing scheme for multiple PEs [16] is relatively inefficient compared with higher-radix schemes. The proposed scheme has three special features. First, it ensures conflict-free FFT butterfly executions during the entire FFT operation. Second, it supports parallel data outputs with normal ordering. This feature is always desirable for providing immediate and normal-order FFT outputs to the succeeding functional blocks, such as channel estimator for timely operations. Thirdly, like many other designs, the in-place FFT computation strategy is also adopted for low memory overhead consideration.

IV. REORDERING OF THE OUTPUT SAMPLES

Reordering of the output samples is an inherent problem in FFT computation. The outputs are obtained in the bit-reversal order [5] in the serial architectures. In general the problem is solved using a memory of size. Samples are stored in the memory in natural order using a counter for the addresses and then they are read in bit-reversal order by reversing the bits of the counter. In embedded DSP systems, special memory addressing schemes are developed to solve this problem. But in case of real-time systems, this will lead to an increase in latency and area. The order of the output samples in the proposed architectures is not in the bit-reversed order. The output order change for different architectures because of different folding sets/scheduling schemes.

A. Area Reduction In LUT

LUT means “Look Up Table.” It’s helpful to think of it like a math problem: R= S+L “R” being your result or what you want to attain. “S” being your source or what you start with. “L” being your LUT or the difference needed to make up between your source and your desired outcome. In all cases of LUT use, the LUT is the means to make up the difference between source and result.

V. PRESENT TECHNIQUE

LUT optimization is the main key factor in our project, in order to reduce power and area. The following techniques have to be implemented in LUT to get required qualities.

- Anti symmetric Product coding (A.P.C)
- Modified Odd multiple storage (O.M.S)

In this project, for the reduction of look-up-table (LUT) size of memory-based multipliers to be used in digital signal processing applications. It is shown that by simple sign-bit exclusion, the LUT size is reduced by half at the cost of a marginal area overhead. Moreover, a novel anti-symmetric product coding (APC) scheme is proposed to reduce the LUT size by further half, where the LUT output is added with or subtracted from a fixed value. The proposed APC–OMS combined design of the LUT for L = 5 and for any coefficient width W is shown in Fig.3. It consists of an LUT of nine optimized LUTs for small input width could be used for efficient implementation of high-precision LUT-multipiers, where the total contribution of all such fixed offsets could be added to the final result or could be initialized for successive accumulations.

<table>
<thead>
<tr>
<th>TABLE I: LUT Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input, X</td>
</tr>
<tr>
<td>0000 01</td>
</tr>
<tr>
<td>0001 10</td>
</tr>
<tr>
<td>0010 11</td>
</tr>
<tr>
<td>0101 00</td>
</tr>
<tr>
<td>1011 11</td>
</tr>
<tr>
<td>0000 10</td>
</tr>
<tr>
<td>0010 01</td>
</tr>
<tr>
<td>0100 10</td>
</tr>
<tr>
<td>1000 01</td>
</tr>
<tr>
<td>0000 00</td>
</tr>
</tbody>
</table>

VI. RESULT

Simulation results of this paper is as shown in bellow Fig 2.

VII. CONCLUSION

Finally, This paper describes the design of two new fused floating-point arithmetic units and their application to the implementation of FFT butterfly operations. Although the fused add-subtract unit is specific to FFT applications, the fused dot product is applicable to a wide variety of signal processing applications. Both the fused dot product unit and the fused add-subtract unit are smaller than parallel implementations constructed with discrete floating-point adders and multipliers. The fused dot product is faster than the conventional implementation, since rounding and normalization is not required as a part of each multiplication. Due to longer interconnections, the fused add-subtract unit is slightly slower than the discrete implementation. An efficient and more flexible architecture of FFT is designed and experimental results are obtained with XILINX.
VIII. REFERENCES


Author's Profile:

Miss K. Deepthi Vatsalya completed her M.B.A (HR) and pursuing M.tech from Nova Engineering College, IBM, and Vijayawada. Her best part is that she galvanizes into the subject.

Mr. P. Sudhakar Rao, currently working as ECE HOD at Nova Engineering College, IBM, Vijayawada. His research and his enthusing interests include high-performance development at his college applications.