A Low-Energy Multi-Length FFT Processor for DVB-T/H Systems

Kuan-Hung Chen¹, Yueh-Shu Li¹, and Kuei-Chung Chang²

¹Department of Electronic Engineering, Feng-Chia University, Taichung, Taiwan
²Department of Information Engineering and Computer Science, Feng-Chia University, Taichung, Taiwan

ABSTRACT

This paper presents a low-energy 2K/4K/8K-point Fast Fourier Transformation (FFT) for Digital Video Broadcasting-Terrestrial/Handheld (DVB-T/H) systems. The proposed FFT processor possesses the features of simple structure, high flexibility, and light memory access. Besides, this study adopts a conflict-free address scheduling and a spurious signal suppression technique respectively to further reduce the energy consumption of the main memory block and the complex multipliers. After being implemented by using the 1P6M TSMC 0.18-μm CMOS technology, this work costs a core area of 4.49 mm², and meets the FFT performance requirements of DVB-T/H when operated at 20 MHz frequency with only 11.8 μJ energy dissipation for computing an 8K-point FFT. The measuring results show that this work increases more than 53% higher energy efficiency for long-length FFT implementation.

Index Terms—DVB, FFT, Low-energy, OFDM, VLSI

1. INTRODUCTION

Orthogonal Frequency Division Multiplexing (OFDM) attracts much attention in wireless communication because it is feasible to many diverse requirements. The long-length FFT is commonly adopted in OFDM systems to increase transmission bandwidth and transmission efficiency in many wireless applications such as Digital Video Broadcasting (DVB), Digital Audio Broadcasting (DAB), Worldwide Interoperability for Microwave Access (WiMAX) and etc. There has been much research discussing about short-length FFT design. However, it is still a challenge to implement a long-length FFT design owning real-time performance under the constraints of low energy consumption and low hardware cost.

In DVB-T systems, 2K- and 8K-point FFTs are needed for the OFDM computation [1]. In addition, 4K-point FFT is added to DVB-H systems to enhance the OFDM performance [2]. This denotes that an electronic device which wants to receive compatible signals of a DVB-T/H system requires a multi-length FFT processor for 2K-, 4K-, and 8K-point operations. Although this requirement can be realized by directly combining the three parts of 2K, 4K, and 8K FFTs together, the combined circuits may be very large and power consuming. To solve the difficulties resulting from the multi long-length computation, this study provides algorithmic analyses and mathematical models of 2K/4K/8K-point FFTs. Accordingly, this work presents an efficient and flexible FFT architecture as well.

The most common VLSI architectures of FFT can be approximately classified into three categories: memory-based architectures [3]-[5], cached-memory architectures [6]-[8], and pipeline architectures [9]-[14]. The memory-based architectures generally consist of a processing unit and a memory block. To finish the FFT computation, the intermediate data are transmitted between the memory block and the processing unit several times depending on the FFT length as well as the adopted radix number. Therefore, the memory-based architecture includes advantages such as low area cost, and high flexibility. However, it often suffers from low memory access speed, large memory power consumption, and high controller complexity. The cached-memory architecture adopts a data cache to reduce the memory access of the memory-based architecture so as to obtain higher memory access speed and lower memory power consumption. Nevertheless, the cached-memory architecture also induces additional cache area, higher controller complexity, as well as lower data throughput rate. Furthermore, the size of the data cache increases exponentially and may become very unrealistic when high-radix algorithms are adopted. On the contrary, the pipeline architecture owns the benefits of high data throughput rate, and low controller complexity. Still, the pipeline architecture bears the price of high area cost. This price grows with the increase of the FFT length, and may become even unacceptable in long-length FFTs such as 4K- or 8K-point FFTs.

This study proposes a low-energy multi-length VLSI architecture for the 2K-, the 4K-, and the 8K-point FFTs. The conceptual presentation can be found in our previous work [15] which includes only preliminary implementation results. Compared with [15], this work presents significant new parts including a detailed theoretical derivation, a thorough data throughput rate analysis, two low-energy strategies adopted to lower the energy consumption, chip fabrication details, and measuring results for the proposed FFT design. This work possesses the advantages of simple structure, high computation efficiency, and high flexibility. Besides, the proposed architecture reduces the memory access of the FFT computation to lower the energy consumption of the memory. Moreover, we exploit the Spurious Signal Suppression Technique (S³T) to decrease the energy dissipation of the multipliers equipped in the FFT processor [16]. After being implemented by using the TSMC 0.18-μm technology, the proposed low-energy FFT processor consumes only 11.8 μJ to compute an 8K-point FFT.

The rest of this paper is organized as follows: Section II presents the algorithmic analysis and the mathematical model of this FFT processor. Then, the design and implementation of the proposed FFT processor is given in Section III. Section IV describes the performance evaluation and comparison of this work. At last, Section V presents conclusions.

2. ALGORITHMIC ANALYSIS AND MATHEMATICAL MODELS OF 2K/4K/8K-POINT FFT

Because the selection of algorithm may result in very large impact on energy consumption, we examine the computation complexity of several radix numbers in the beginning of the design process. We first analyze the computation complexity of the 2K-, the 4K-, and the 8K-point FFTs based on several commonly used radices to choose the most suitable algorithm for this work. Then,
we disclose the mathematical model of the 8K-point FFTs based on the chosen algorithm as an example for illustrating the derivation of multi-length FFT computation. Such arrangement is to keep the paragraph succinct. This derivation also inspires us the innovation of multi-length FFT computation. Such arrangement is to keep the chosen algorithm as an example for illustrating the derivation

The N-point Discrete Fourier Transform (DFT) of a sequence \( x(n) \) is defined as Eq. (1) in the follows:

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

(1)

where \( x(n) \) and \( X(k) \) are respectively the input and the output sequence composed of complex numbers. The twiddle factor is shown in Eq. (2).

\[
W_N^{nk} = e^{-\frac{2\pi j nk}{N}} = \cos(\frac{2\pi nk}{N}) - j \sin(\frac{2\pi nk}{N})
\]

(2)

The computational complexity of Eq. (1) is \( O(N^2) \) when the required computations are directly executed. The computational complexity can be reduced to \( O(N \log N) \) by using Cooley-Tukey decomposition [17], where \( r \) denotes the radix number of the adopted FFT algorithm. Seemingly, the computational complexity decreases as the radix increases for a fixed-length FFT computation. This indicates that larger radices are more suitable for long-length FFT computations. However, the FFT algorithm with excessive large radix may be difficult to implement.

A. Algorithmic Analysis

According to [11], higher radix FFT algorithms will induce higher circuitry complexity. This is the main reason that algorithms with radix number higher than eight are seldom used. Consequently, this work explores higher radices, e.g. radix-16 and radix-32 algorithms, which may still be realizable. Table 1 analyzes the computational complexity of FFT algorithms in terms of the numbers of multiplications and additions based on the radix numbers from radix-2 to radix-32. This work concerns the lengths of 2K-, 4K-, and 8K-point of FFT, but one cannot finish these three kinds of FFT by using some certain radices. Hence, not all of them are listed in the second column of Table 1. Furthermore, to have a common length for easy comparison, we use smaller radices to compensate the remainder part of the 8K-point FFT computation in some algorithms and calculate the complete complexity of the 8K-point FFT. In the FFT decompositions with radix number larger than four, the complex multiplications can be further classified into non-trivial and trivial ones such as these with multipliers of \( \pm 1 \) or \( \pm j \). Again, for easy comparison, we convert the complex multiplications and complex additions to equivalent real multiplications and real additions by using the following facts:

1) one nontrivial complex multiplication involves four real multiplications and two real additions;  
2) one trivial complex multiplication includes two real multiplications and two real additions;  
3) one complex addition consists of two real additions.

The equivalent data are listed respectively in the last two columns of Table 1. These data are further normalized as shown in Table 2 where we find that using radix-16 algorithm can result in lower computation complexity than using algorithms with smaller radices. Although using radix-32 algorithm is also more efficient than using radix-16 algorithm, the circuitry complexity may be too high to implement. Besides, using radix-16 algorithm can reduce about 7000 real multiplications than using radix-8 algorithm; however, using radix-32 algorithm reduces only about 3000 real multiplications than using radix-16 algorithm. Therefore, we decide to adopt radix-16 algorithm in this work.

B. Mathematical Model for 8K-point FFT

Here we take the longest FFT, i.e. 8K-point FFT, as an example for demonstrating the mathematical derivation. Because 8K-point FFT cannot be exactly calculated using only radix-16 algorithm, radix-2 algorithm is also required. We first apply radix-2 algorithm on 8K-point FFT computation and get the following intermediate results:

Let

\[
n = n + \frac{8192}{2} \quad n_2 = n_1 + 4096n_2 \quad n_1 = 0 \ldots 8192 \quad n_2 = 0 \ldots 4095
\]

\[
n_2 = 0, 1
\]

\[
k = 2k_1 + k_2 \quad k_1 = 0 \ldots \frac{8192}{2} \quad k_2 = 0 \ldots 4095
\]

\[
k_2 = 0, 1
\]

Filling \( n \) and \( k \) into Eq. (1), it turns into Eq. (3)

\[
X(2k_1 + k_2) = \sum_{n_2=0}^{4095} \sum_{n_1=0}^{8192} x(n_1 + 4096n_2)W_N^{(n_1 + 4096n_2)2k_1 + n_2}W_N^{nk_2}
\]

(3)
using three iterations of radix-16 algorithm as follows:

\[ X(2(16k_1 + k_2) + k_3) = \sum_{n=0}^{4096} 8K_{-2}(n_i + 16n_j)W_{4096}^{n_i}W_{4096}^{n_j} \]

which is precisely a 4K-point FFT and can be computed exactly using three iterations of radix-16 algorithm as follows:

Let \( n_i = n_i + \frac{N}{32} \), \( n_j = n_j + 16n_i \), \( n_i = 0 \ldots \frac{N}{32} - 1 = 0 \ldots 15 \)

\[ k_1 = 16k_1 + k_4, \quad k_4 = 0 \ldots \frac{N}{32} - 1 = 0 \ldots 15 \]

Similarly, we can let

\[ 8K_{-3}(n_i) = \sum_{n=0}^{8K_{-1}} (n_i + 16n_j)W_{8192}^{n_i}W_{8192}^{n_j} \]

which denotes the computation result of the third pipeline stage. The remaining 8K-point FFT equation becomes

\[ X(2(16k_1 + k_2) + k_3) + k_4 \]

which is a 16-point DFT and can be calculated using the fourth pipeline stage of radix-16 FFT computation. The twiddle factors needed in the first three stages are shown in Eq. (3), Eq. (5) and Eq. (8). From these equations, we know that there are respectively 8192, 4096 and 256 twiddle factors in the first three stages. There is no complex multiplication of twiddle factor in the last stage of the computation as shown in Eq. (10). Besides, the twiddle factors shown in Eq. (3) form a superset of the twiddle factors of 2K- and 4K-point FFTs. Because of the symmetry of the sinusoidal waveforms, only 1/8 period of cosine and sine waveforms are stored in ROM and the remaining 7/8 period parts can be reconstructed by these stored coefficients.

From the architecture design view point, each stage of computations described in the above equations can be processed in a pipeline way, as illustrated in Fig. 1. In Fig. 1(a), the pipeline structure is composed of several coarse grain pipeline stages. Each coarse grain pipeline stage calculates a complete radix of FFT. Inside the coarse grain pipeline stage, there may be several fine grain pipeline stages inside. The details will be explained later. Substituting \( 8K_{-1}(n) \) into Eq. (3) we can obtain Eq. (4) as follows:

\[ X(2k_1 + k_2) = \sum_{n=0}^{4096} 8K_{-1}(n_i)W_{4096}^{n_i}k_2 \]

which is precisely a 4K-point FFT and can be computed exactly using three iterations of radix-16 algorithm as follows:

Let \( n_i = n_i + \frac{4096}{16} \), \( n_j = n_j + 256n_i \), \( n_i = 0 \ldots \frac{4096}{16} - 1 = 0 \ldots 255 \)

\[ k_1 = 16k_1 + k_4, \quad k_4 = 0 \ldots \frac{4096}{16} - 1 = 0 \ldots 255 \]

\[ k_1 = 0 \ldots 15 \]

Replacing \( n \) and \( k \), Eq. (4) becomes:

\[ X(2(16k_1 + k_2) + k_3) = \sum_{n=0}^{4096} 8K_{-2}(n_i)W_{4096}^{n_i}k_2 \]

\[ = \sum_{n=0}^{4096} 8K_{-2}(n_i + 16n_j)W_{4096}^{n_i}W_{4096}^{n_j} \]

\[ = \sum_{n=0}^{4096} 8K_{-2}(n_i + 16n_j)W_{4096}^{n_i}W_{4096}^{n_j} \]

which is similar to Eq. (1) and is exactly a 256-point DFT. Using similar steps to Eq. (7), we can get the results of the third pipeline stage as follows:

Let \( n_i = n_i + \frac{N}{32} \), \( n_j = n_j + 16n_i \), \( n_i = 0 \ldots \frac{N}{32} - 1 = 0 \ldots 15 \)

\[ k_1 = 16k_1 + k_4, \quad k_4 = 0 \ldots \frac{N}{32} - 1 = 0 \ldots 15 \]

\[ k_1 = 0 \ldots 15 \]

Similarly, we can let

\[ 8K_{-3}(n_i) = \sum_{n=0}^{4096} 8K_{-2}(n_i + 16n_j)W_{8192}^{n_i}W_{8192}^{n_j} \]

which denotes the computation result of the third pipeline stage. The remaining 8K-point FFT equation becomes

\[ X(2(16k_1 + k_2) + k_3) + k_4 \]

which is a 16-point DFT and can be calculated using the fourth pipeline stage of radix-16 FFT computation. The twiddle factors needed in the first three stages are shown in Eq. (3), Eq. (5) and Eq. (8). From these equations, we know that there are respectively 8192, 4096 and 256 twiddle factors in the first three stages. There is no complex multiplication of twiddle factor in the last stage of the computation as shown in Eq. (10). Besides, the twiddle factors shown in Eq. (3) form a superset of the twiddle factors of 2K- and 4K-point FFTs. Because of the symmetry of the sinusoidal waveforms, only 1/8 period of cosine and sine waveforms are stored in ROM and the remaining 7/8 period parts can be reconstructed by these stored coefficients.

From the architecture design view point, each stage of computations described in the above equations can be processed in a pipeline way, as illustrated in Fig. 1. In Fig. 1(a), the pipeline structure is composed of several coarse grain pipeline stages. Each coarse grain pipeline stage calculates a complete radix of FFT. Inside the coarse grain pipeline stage, there may be several fine grain pipeline stages where each fine grain pipeline stage computes a butterfly operation. Direct implementation of such pipeline architecture results in the drawbacks of large area cost and high power dissipation. In this study, we introduce the folding concept from digital signal processing for VLSI to solve the above drawbacks [18]. Fig. 1(b) shows the illustration of the folding structure of the original pipeline structure in Fig. 1(a). Comparing Fig. 1(a) and Fig. 1(b), we can realize that the hardware cost is largely decreased.

3. DESIGN AND IMPLEMENTATION OF MULTI-LENGTH FFT PROCESSOR

From the view point of system design, the specific and regular computation burdens are favorable to be implemented in hardware manner. With the aid of these hardware blocks, the CPU can
concentrate on the management tasks such as work allocation, data flow control, interrupt operation, etc. The hardware blocks own the advantages like high performance, low power consumption, and low area cost; however, they are usually not flexible to accelerate multi-length computations. To provide enough flexibility to a hardware block without weakening its good features, particular design efforts are required.

A. Proposed Multi-Length FFT Processor

Because the cache size of the cached-memory architecture grows exponentially as the radix number increases and we adopt a rather large radix size, i.e. radix-16, the cached-memory architecture is not suitable for this work. The pipeline architecture may be more proper; nevertheless, the hardware cost is too high, especially for long-length, e.g. the 8K-point, FFT computations. If we re-classify the FFT architectures according to the degree of time-multiplexing, the memory-based architecture (including the cached-memory architecture) and the pipeline architecture are located on the extremely two different ends [19]. Consequently, this work proposes a new architecture which compromises the opposite features between the memory-based architecture and the pipeline architecture.

In this study, both the 2K- and the 4K-point FFTs can be calculated in three stages of computation, and the 8K-point FFT can be finished in four stages of computations. From the processor design point of view, each stage of the FFT computation can be implemented in a coarse-grain pipeline stage. These coarse-grain pipeline stages may be folded due to their high similarity as Fig. 1(a) and the mathematical model presented in Section II show. Take the 8K-point FFT for example, Eq. (3), Eq. (5), Eq. (8), and Eq. (10) respectively represent four computation stages of the FFT. Except the first stage which is a radix-2 FFT, each stage consists of a radix-16 FFT computation. Besides, the first three stages require non-trivial multiplications for twiddle factors while the last stage does not. The situations are very similar in the 2K- and the 4K-point FFT.

Fig. 1 inspires us to develop a novel low-energy VLSI architecture which provides a bright trade-off between the pipeline architecture and the memory-based architecture. Fig. 2 illustrates the proposed FFT processor that possesses a memory block, a multi-radix butterfly unit, a complex multiplication block, commutators, and a controller. The memory block exists for temporal input data storage, intermediate data storage, and data regulation. The multi-radix butterfly unit, a central computation unit of the proposed architecture, is designed as a reconfigurable pipeline data-path to implement the multi-radix computations, i.e. the radix-16, radix-8, and radix-2 algorithms, in a cost-efficient way. The commutators regulate the data-flow of FFT computation. The controller sends proper signals to all components to finish the FFT computation seamlessly.

This work possesses the features such as low area cost and high flexibility. The high flexibility is achieved by using the controller unit and the reconfigurable butterfly unit shown in Fig. 2 and Fig. 3. Owning high flexibility enables this work to compute multi-length FFTs. The proposed multi-radix butterfly unit shown in Fig. 3, composed of four radix-2 butterfly units, is designed according to the signal-flow plots illustrated in Fig. 4. Two simple hard-wired multipliers are respectively located at the ends of the first and the second butterfly stage for trivial multiplications. Furthermore, there is a simple interchanging and complementing circuitry for multiplications of $-j$s at the end of the third butterfly stage. The two complex multipliers are arranged outside the butterfly unit as shown in Fig. 2. Hence, the proposed multi-radix butterfly unit can process two intermediate data per cycle. By
inserting some multiplexers and de-multiplexers properly as Fig. 3 shows, this unit can be reconfigured to compute multiple radices, i.e. radix-2, radix-8, and radix-16, required in the multi-length FFT processing.

Moreover, the word-length analysis of the proposed FFT processor is shown in Fig. 5. This figure demonstrates that the proposed FFT processor achieves near 60dB Signal to Quantization Noise Ratio (SQNR) value for the 8K-point FFT when adopting 12-bit data word-length. The Bit Error Rate (BER) of this FFT processor using 12-bit data wordlength is also lower than $2 \times 10^{-4}$, which satisfies the requirement of the DVB-T/H standards [1]-[2].

B. Low Energy Design Strategies

Besides reducing the memory access, we take advantage of the following two strategies to decrease the energy dissipation of this FFT processor:

1) Using dedicated single-port SRAM blocks

2) Adopting low-energy multipliers

Because the proposed design adopts a two-way multi-radix butterfly unit which may read and write the main SRAM block simultaneously, it is straightforward to equip a two-bank dual-port SRAM block for this design. However, the power dissipation of the dual-port SRAM block is too large to be accepted. After replacing the dual-port SRAM block with the functional-equivalent main SRAM formed by single-port SRAM blocks, we find that 16% power consumption of the FFT processor is saved, which is consistent with the report [20]. The dedicated main memory block for the proposed work is composed of 16 single-port SRAM blocks. We adopt the conflict-free memory addressing algorithm [21] to avoid the data confliction. Cooperating with the conflict-free addressing algorithm, the multiplexer and the de-multiplexer inside the main memory block direct the data flows in an interleaved way which enables the dedicated main memory block to behave as a functional-equivalent dual-port SRAM.

Furthermore, we adopt the low-energy modified Booth multiplier equipped with the spurious signal suppression technique (S3T) to decrease the energy consumed by the complex multipliers of this work [16]. To spare the augmented cost, only significant adders are replaced with the S3T adders. Besides, controlling gates which can keep the data from flowing into the non-significant part of the multiplier are equipped in the Booth encoder to eliminate the spurious signals in the circuits. Hence, the energy consumption resulted from the signal switching can be decreased. This optimization not only can reduce the energy consumed by Booth encoder but also can save the energy dissipated by the adder tree of the multiplier. In addition, the conformation of the detection logic unit is very simple so that the induced cost is slight. After equipping the S3T multipliers, more than 10% energy consumption of the FFT processor is reduced.

4. PERFORMANCE EVALUATION AND COMPARISON

After being fabricated by using the TSMC 1P6M 0.18-μm CMOS technology, the chip microphotograph of the proposed FFT processor with a core size of 2.12×2.12 mm² is illustrated in Fig. 6. The chip is packaged in a 160-pin CQFP package, where 34 pins are power pins and other 126 pins are signal pins. Among the signal pins, 25 pins are used for chip testing. The summary of the chip implementation as well as some measurement results are also listed in Fig. 6. When targeting the FFT computation of DVB-T/H, the proposed design consumes only 17.8 mW at 20 MHz operating frequency and 1.8V supply voltage. Under this operating condition, the execution time for calculating the 8K-point FFT is 660.65 μs. Furthermore, Table 3 illustrates the performance comparison of this work with three recent state-of-the-art designs [8], [14], and [22]. The designs [8] and [14] are respectively the representatives of the cached-memory and the pipeline architecture. The cached-memory
architecture offers a major advantage of higher flexibility. Nevertheless, it also induces additional cache overhead, higher controller complexity, and lower data throughput rate. To increase the data throughput rate, [8] uses four pairs of complex multipliers and butterfly units to speed up the processing capability. Besides, the size of the cache memory inside [8] may increase exponentially as the radix number of the adopted algorithm augments. This keeps the cached-memory architecture from adopting higher radix algorithms which are beneficial to long-length FFT computation. The design [14] exploits pipeline architecture which possesses the advantage of high data throughput rate. Nevertheless, the pipeline architectures spend higher area and power overheads. The design [22] presents a low-power and area-efficient pipelined FFT processor for DVB-T/H systems.

5. CONCLUSION

Based on the proposed multi-length architecture, this paper presents a low-energy 2K/4K/8K-point Fast Fourier Transformation (FFT) processor for Digital Video Broadcasting-Terrestrial/Handheld (DVB-T/H) systems. The proposed architecture provides a bright trade-off between hardware overhead and memory access. This work possesses the advantages of high flexibility, high energy efficiency, and low controller complexity. Besides architectural improvement, further energy reductions are obtained by optimizing the main memory block and the complex multipliers. After being implemented by using the TSMC 1P6M 0.18-μm CMOS technology, the proposed FFT processor can complete the 2K/4K/8K-point FFT in the performance budget specified in DVB-T/H with the operating frequency of 20 MHz, and dissipates less than 11.8 μJ of energy. This measuring result shows that this work enhances more than 53% higher energy efficiency for multi-long-length FFT implementation.

6. REFERENCES


