Area-efficient mixed-radix variable-length FFT processor

Chen Yang¹, Chunpeng Wei¹, Yizhuang Xie¹, He Chen¹a, and Cuimei Ma²

¹ Beijing Key Laboratory of Embedded Real-time Information Processing Technology, Beijing Institute of Technology, Beijing 100081, China
² Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China

a) chenhe@bit.edu.cn

Abstract: This paper presents a mixed-radix multipath delay feedback (MDF) FFT processor with variable-length. In order to minimize the number of occupied multipliers while supporting more flexible FFT length, a 4-parallel radix-2⁴/3/4 architecture is adopted. In order to further optimize the area and power consumption, we make efforts in constant multiplier design, twiddle factor generation and butterfly units multiplexing. CSD multiplier is adopted to realize the constant factor multiplication in radix-2⁴ and radix-3 butterfly. Only one CORDIC, several adders and multipliers are occupied to achieve the 4-parallel twiddle factor generation. A radix-2/3/4 multiplexing butterfly unit with simple control logic is also designed. The design is synthesized with 65 nm CMOS technology. Compared with previous works, the proposed design shows advantages in terms of area, power consumption, and processing latency.

Keywords: area-efficient, FFT, multipath delay feedback (MDF), mixed-radix, variable-length

Classification: Integrated circuits

References

[1] Y. W. Lin, et al.: “A 1-GS/s FFT/IFFT processor for UWB applications,” IEEE J. Solid-State Circuits 40 (2005) 1726 (DOI: 10.1109/JSSC.2005.852007).
[2] J. Chen, et al.: “Hardware efficient mixed radix-25/16/9 FFT for LTE systems,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 23 (2015) 221 (DOI: 10.1109/TVLSI.2014.2304834).
[3] S. He and M. Torkelson: “Designing pipeline FFT processor for OFDM (de)modulation,” ISSSE 98 (1998) 257 (DOI: 10.1109/ISSSE.1998.738077).
[4] J. Wang, et al.: “A mixed-decimation MDF architecture for radix-2⁴, parallel FFT,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 24 (2016) 67 (DOI: 10.1109/TVLSI.2015.2402207).
[5] C. H. Yang, et al.: “Power and area minimization of reconfigurable FFT processors: A 3GPP-LTE example,” IEEE J. Solid-State Circuits 47 (2012) 757 (DOI: 10.1109/JSSC.2011.2176163).
[6] M. Garrido, et al.: “Pipelined radix-2⁴ feedforward FFT architectures,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 21 (2013) 23 (DOI: 10.1109/
1 Introduction

Fast Fourier Transform (FFT) is one of the most fundamental algorithms used in digital signal processing field. FFT processors have been widely used in various applications such as communications, image processing, digital video broadcasting (DVB) and radar signal processing. The extension of application field brings more challenges to the design of FFT processor. For example, to meet the strict specifications ultra-wideband (UWB) communication system, high throughput data should be processed in low latency [1]. In long-term evolution (LTE) application, transform sizes ranging from 12 points up to 1296 points are required. The traditional methods for $2^n$-point FFTs are difficult to apply to a general mixed radix FFT design [2]. Pipeline architectures have an inherent advantage over other efficient hardware structures [3]. The multipath delay feedback (MDF) as well as multipath delay commutator (MDC) architecture is adopted for applications with higher throughput rate and better real-time performance requirements [4]. Mixed-radix algorithms make the processor more flexible for a variety of transform sizes [5]. Radix-$2^k$ algorithm is suitable for pipeline architecture. As the radix becomes higher, the number of occupied complex multipliers decreases [6]. In this brief, we combine the advantages of MDF architecture, mixed-radix and radix-$2^k$ FFT algorithm. A mixed-radix multipath delay feedback (MDF) FFT processor with variable-length is proposed.

A 128- to 2048-point FFT processor for 3GPP-LTE standard is implemented in [5]. A design methodology for power and area minimization of flexible FFT processors is proposed by analyzing the degree of parallelism and radix factorization. In [7], an MDC-based FFT processor for multiple input multiple output (MIMO) orthogonal frequency division multiplexing (OFDM) systems with variable length is presented. The transform size include 2048, 1024, 512 and 128. In [8], a high-speed low-complexity modified radix-$2^5$ 512-point FFT processor is designed for wireless personal area network applications using eight pipelined datapath. Another 512-point MDF FFT processor is designed in [9]. The radix-$2^4$-$2^2$-$2^3$ algorithm is devised to reduce the complexity of twiddle factor multiplication.
Parallel-pipelined (MDC or MDF) FFT achieves higher data throughput and lower processing latency at the cost of more arithmetic units (AUs). In order to reduce the complexity of AUs, we analyze the relationship between high-radix FFT algorithm and the degree of hardware parallelism. 4-parallel radix-2 MDF architecture is adopted as a much more balanced scheme. In order to support some special FFT lengths in specific applications, we also introduce radix-3 algorithm to our design. In the case of different FFT length, the output of each parallel data path requires corresponding twiddle factors. We design a module including one CORDIC, several adders and multipliers to achieve the 4-parallel twiddle factor generation. A radix-2/3/4 multiplexing butterfly unit with simple control logic is designed to further reduce the area and power consumption of our proposed FFT processor.

The rest of the paper is organized as follows: Section 2 introduces the top-level design of the FFT processor by analyzing the relationship between FFT algorithm and hardware parallelism. In Section 3, we discuss the proposed MDF architecture including the design and optimization of sub-modules. In Section 4, we show the pre-layout area and power results for comparison with previous works. Section 5 concludes the paper.

2 Top-level design of parallel-pipelined FFT

The N-point discrete Fourier transform (DFT) of an input sequence is defined as

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}.$$  \hspace{3cm} (1)

The Cooley-Tukey algorithm [10] leverages the divide-and-conquer technique to recursively partition a DFT of size $N = M \times L$ into many smaller DFTs of sizes $M$ and $L$. For $N = M \times L$ and $M = 2^k$, let

$$\begin{align*}
  n &= L \cdot m + l \\
  m &= 0, 1, \ldots, M - 1; l = 0, 1, \ldots, L - 1 \\
  k &= i + M \cdot j \\
  i &= 0, 1, \ldots, M - 1; j = 0, 1, \ldots, L - 1
\end{align*}$$

the $N$-point FFT can be represented in a two-dimensional form as:

$$X(i + M \cdot j) = \sum_{l=0}^{L-1} [X_l(i) \cdot W_{N}^{jl}] \cdot W_{L}^{jl},$$

where $W_{N}^{jl}$ is the twiddle factor and $X_l(i)$ is the $M$-point FFT defined as

$$X_l(i) = \sum_{m=0}^{M-1} x(L \cdot m + l) W_{M}^{im}.$$  \hspace{3cm} (4)

Hence, the calculation of the $M \times L$-point FFT can be decomposed into three steps: (1) calculate $M$-point FFT $X_l(i)$, (2) generate the twiddle factor $W_{N}^{jl}$ for corresponding $X_l(i)$, and (3) calculate $L$-point FFT. In order to map the algorithm to MDF architecture, $L$ is treated as the degree of hardware parallelism and each internal SDF path is responsible for managing one of the parallel input $M$-point data. A parallel twiddle factor generation block should be designed as well.

The degree of hardware parallelism $L$, to a great extent, determines the data throughput rate and processing latency of the FFT processor. However, excessively
high hardware parallelism results in much more arithmetic units and registers occupation which leads to larger area and power consumption. Therefore, a tradeoff should be made between the hardware parallelism and power-area consumption.

In order to reduce the number of occupied complex multipliers, the \( M \)-point FFT can be realized with radix-\( 2^k \) algorithm. Higher radix yields more complex multiplier reduction benefits while losing the flexibility of variable transform size. This shortcoming can be solved by designing a variety of \( L \)-point butterfly units. For example, if the \( M \)-point FFT is realized with radix-\( 2^4 \) algorithm, then the max value of \( L \) should be 8 and the \( L \)-point butterfly units need to switch between radix-2, radix-4 and radix-8 units in order to ensure the transform size is variable in accordance with the power of 2. Therefore, the degree of hardware parallelism \( L \) affects not only the circuit complexity, but also the choice of FFT algorithm and the design of variety butterfly units. The number of complex multipliers occupied by an \( L \)-channel parallel radix-\( 2^k \) MDF FFT is calculated by

\[
A_{\text{mul}} = L \cdot \left[ \log_2 \left( \frac{N_{\text{FFT}}}{L} \right) - 1 \right] + 2^k - 1
\]

(5)

Fig. 1 shows the amount of complex multipliers under different design schemes. Obviously, the radix-\( 2^4 \) 8 channels and radix-\( 2^3 \) 16 channels schemes consume large amount of multiplier resources. The radix-\( 2^2 \) 2 channels and radix-\( 2^3 \) 4 channels schemes both show some superiority compared with radix-2 single channel scheme. As indicated by the arrow, the number of multipliers occupied by radix-\( 2^3 \) 4 channels scheme is always less than radix-2 single channel scheme until the FFT length is greater than 2048. Although radix-\( 2^2 \) 2 channels seems more multiplier-efficient, taking into account the data throughput, processing latency, etc., we choose radix-\( 2^3 \) 4 channels scheme as the final solution.

Actually, the compatibility with the radix-3 butterfly is another advantage of radix-\( 2^3 \) 4 channels scheme. Let \( L = 3 \), (3) can be rewritten as:

\[
X(i) = X_0(i) + W_N^0 X_1(i) + W_N^2 X_2(i)
\]

\[
X(i + N/3) = X_0(i) + W_N^2 \cdot X_1(i) + W_N^4 \cdot X_2(i)
\]

(6)

\[
X(i + 2N/3) = X_0(i) + W_N^4 \cdot X_1(i) + W_N^8 \cdot X_2(i)
\]

Fig. 1. Amount of complex multipliers under different design schemes
According to the value of $W_1^3$, $W_2^3$ and $W_4^3$, (6) is organized as:

\[
X(i) = X_0(i) + W_1^3X_1(i) + W_2^3X_2(i)
\]

\[
X(i + N/3) = X_0(i) - \frac{1}{2}(W_1^3X_1(i) + W_2^3X_2(i)) - \left(\frac{j\sqrt{3}}{2}\right)(W_N^3X_1(i) - W_2^3X_2(i))
\]

\[
X(i + 2N/3) = X_0(i) - \frac{1}{2}(W_1^3X_1(i) + W_2^3X_2(i)) + \left(\frac{j\sqrt{3}}{2}\right)(W_N^3X_1(i) - W_2^3X_2(i))
\]

Using three of the four channels, combined with radix-3 butterfly unit, $M \times 3$-point FFT can be performed.

Based on the above analysis, we adopt radix-2 algorithm to realize the $M$-point FFT of 4 channel parallels. Each $M$-point FFT is variable in accordance with the power of $2^3$. Combined with a variable radix-$L$ ($L = 2, 3, 4$) butterfly unit, a $2^k$ or $2^{3k} \times 3$-point variable MDF FFT processor is achieved. The block diagram is shown in Fig. 2.

**3 Design of proposed MDF architecture**

In this section, we discuss the sub-module optimization and architecture design of the proposed FFT processor.

**3.1 Constant multiplier**

In radix-$2^3$ algorithm, $W_8^1 = \sqrt{2}/2 \cdot (1 - j)$ is an inherent constant factor. In the radix-3 algorithm presented in (7), $\sqrt{3}/2$ is another constant factor. In order to reduce hardware cost, we transform $\sqrt{2}/2$ and $\sqrt{3}/2$ to canonic signed digit (CSD) presentation. Table I shows the CSD presentation. The code $10\bar{1}0\bar{1}01\bar{0}1\bar{0}0000$ means $(1 - 2^{-2} - 2^{-4} + 2^{-6} + 2^{-8}) = 0.7070$. Fig. 3 shows the CSD constant multiplier design. Only four adders/subtractors and four shift registers are occupied for both constant factors.

**Table I.** CSD presentation of constant factors

| Coefficient | Decimal  | 2’s Complement | CSD         |
|-------------|----------|----------------|-------------|
| $\sqrt{2}/2$ | 0.7071   | 0.101101010000 | 10\bar{1}0\bar{1}01\bar{0}1\bar{0}0000 |
| $\sqrt{3}/2$ | 0.8660   | 0.110111011011 | 100\bar{1}000\bar{1}0\bar{1}0\bar{1}$ |
3.2 Twiddle factor generation

As shown in the block diagram in Fig. 2, a twiddle factor generation block should be designed for the multi-channel radix-2\(^3\) FFT corresponding to different radix-L butterfly units. When L varies from 2 to 4 and the FFT length \(N\) is \(2^k\) or \(2^{3k} \times 3\)-point variable, the twiddle factors \(W^i_N\) will change correspondingly. For example, let \(L = 4\), all four channels are enabled to perform radix-2\(^3\) FFT in this case. According to (3) the twiddle factors corresponding to each channel are \(W^0_N\), \(W^1_N\), \(W^2_N\) and \(W^3_N\). Thus, it occupies too much memory resource to design a twiddle factor generation module based on the ROM-based look up table method. We adopt Coordinated Rotation Digital Computer (CORDIC) to real-time generate twiddle factors corresponding to different twiddle factor address. However, in order to ensure parallel processing, at least three CORDIC units are required for the four parallel radix-2\(^3\) FFT channels. This also leads to large logic resource consumption. According to the trigonometric function relationship as follows:

\[
\begin{align*}
\cos 2x &= \cos^2 x - \sin^2 x \\
\sin 2x &= 2 \cdot \sin x \cdot \cos x
\end{align*}
\]
\[
\begin{align*}
\cos 3x &= \cos 2x \cdot \cos x - \sin 2x \cdot \sin x \\
\sin 3x &= \sin 2x \cdot \cos x + \cos 2x \cdot \sin x
\end{align*}
\]

(8)

It is easy to find that \(\cos 2x\), \(\sin 2x\), \(\cos 3x\) and \(\sin 3x\) can be deduced by \(\cos x\) and \(\sin x\) with several additions and multiplications. Let \(x = 2\pi i/N\), the twiddle factors can be expressed as follows:

\[
\begin{align*}
W^i_N &= \cos x - j \cdot \sin x \\
W^2_N &= \cos 2x - j \cdot \sin 2x \\
W^3_N &= \cos 3x - j \cdot \sin 3x
\end{align*}
\]

(9)

Consequently, we adopt one CORDIC unit to generate \(W^i_N\). We use additions, multiplications to generate \(W^2_N\) or \(W^3_N\) in the case of \(L = 3\) or 4. Table II shows the logic resource comparison result between two schemes based on FPGA logic units. Our scheme saves 31% logic resource. Fig. 4 shows the block diagram of the proposed twiddle factor generation module. The address generator is essentially a modulo-\(M\) counter. It generates the phase \(x = 2\pi i/N\) \((i = 0, 1, \ldots, M-1)\) for the CORDIC unit. Thus, \(W^0_N, W^1_N, W^2_N\) and \(W^3_N\) are generated for each channel.

3.3 Radix-2/3/4 multiplexing butterfly unit

As discussed above, to achieve a \(2^k\) or \(2^{3k} \times 3\)-point variable FFT processor, the radix-L butterfly unit should support radix-2, radix-3 and radix-4 three modes. If three butterfly units are designed separately, 32 real adders and a 3-to-1 multiplexer are required. We propose a radix-2/3/4 multiplexing butterfly unit as shown in Fig. 5. Only 20 real adders and 7 2-to-1 multiplexers are occupied.
The butterfly unit can be configured to three modes with the 2-bit control signal (S-2, S-3) as shown in Table III. When the butterfly unit is configured to radix-4 mode, the four parallel radix-2^3 FFT channels all work simultaneously. When the butterfly unit is configured to radix-3 mode, the input and output are asymmetric. Data is input from chn0, chn1 and chn3 while the result is output from O0, O2 and O3.

Table III. Number of twiddle factors in radix-2^k algorithm

| Scheme | Proposed CORDIC+Multiplier+Adder | CORDIC |
|--------|----------------------------------|--------|
| Unit   | 12-bit CORDIC                   | 12-bit real multiplier | 12-bit real adder | 12-bit CORDIC |
| LUTs per unit | 1090                           | 162                    | 12                 | 1090           |
| Amount  | 1                              | 7                      | 3                  | 3              |
| Total LUTs | 2260                           |                        |                    | 3270           |

![Fig. 4. Block diagram of the twiddle factor generation module](image)

![Fig. 5. Design of proposed radix-2/3/4 multiplexing butterfly unit](image)

The butterfly unit can be configured to three modes with the 2-bit control signal (S-2, S-3) as shown in Table III. When the butterfly unit is configured to radix-4 mode, the four parallel radix-2^3 FFT channels all work simultaneously. When the butterfly unit is configured to radix-3 mode, the input and output are asymmetric. Data is input from chn0, chn1 and chn3 while the result is output from O0, O2 and O3.

Table III. Input, output and mode control of the butterfly unit

| Mode   | S-2, S-3 | Input                  | Output   |
|--------|----------|------------------------|----------|
| Radix-2| 1, 0     | Chn0, Chn2             | O0, O2   |
| Radix-3| 0, 1     | Chn0, Chn1, Chn3      | O0, O2, O3 |
| Radix-4| 0, 0     | Chn0, Chn1, Chn2, Chn3| O0, O1, O2, O3 |
3.4 Proposed MDF architecture

Fig. 6 shows a 2048-point reconfigurable FFT processor of the proposed architecture. Each parallel channel performs an 8/64/512-point variable radix-2^3 single delay feedback (SDF) FFT by reconfiguring the data-path inter-connection between the radix-2^3 stages. Since the operation timing of the four channels is identical, they share the same twiddle factor ROM. Leveraging the symmetry of twiddle factors, the proposed design requires only one quarter as much ROM space for both real and imaginary parts. For example, a (512/4 = 128)-depth ROM is occupied by the first stage of the radix-2^3 FFT. The word lengths of internal data and twiddle factor are both 12 bit. The internal data buffer RAM size is (256 + 128 + 64 + 32 + 16 + 8) × 24 × 4 bit = 48 Kbit. The twiddle factor ROM size is (128 + 16) × 24 bit = 3.5 Kbit. The following TF.Gen module is the proposed twiddle factor generation block discussed in part 3.2. The mixed-radix BF unit is configured to radix-2, radix-3 or radix-4 mode to perform 2^{3k} × 2, 2^{3k} × 3 or 2^{3k} × 4-point FFT, or rather 2^k or 2^{3k} × 3-point FFT.

4 Evaluation and comparison

The main purposes of the proposed MDF FFT processor are short processing latency and variable FFT length. We make a summary of the processing latency and FFT length in Table IV. For the FFT length in accordance with the power of 2^{3k}, one channel is adopted to perform the FFT. Therefore, the processor can simultaneously perform 1~4 paths radix-2^3 FFT. The processing latency is close to that of the SDF-structure FFT processor. For the 2^{3k} × L-point FFT, L channels are adopted. The processing latency is approximately 1/L of that of the SDF-structure FFT processor.

| FFT length | 64 | 128 | 192 | 256 | 512 | 1024 | 1536 | 2048 |
|------------|----|-----|-----|-----|-----|------|------|------|
| Parallelism| 1  | 2   | 3   | 4   | 1   | 2    | 3    | 4    |
| Latency (clock periods) | 90 | 124 | 124 | 124 | 551 | 585  | 585  | 585  |
The architecture of the proposed FFT processor is designed in VHDL and simulated to verify its functionality. For comparison with previous works, we synthesize the design with Synopsys DC (design compiler) using SMIC (Semiconductor Manufacturing International Corporation) 65 nm technology. We perform the power analysis with Synopsys PrimeTime PX under 40 MHz clock constraint. Table V shows the performance comparisons between the proposed FFT processor and several existing FFT processors.

The selected comparison works are either parallel structure or variable-length FFT processors. The proposed design obtains shorter execute time. Compare with [5], the proposed design obtains a slightly smaller core area and a larger power consumption. However, taking into account that the working frequency and supply voltage of [5] are only half of ours, our processor still has advantage in power consumption. As the processing technology is different, the area and power of [7] are normalized as follows:

\[
\begin{align*}
A_{\text{nor}} &= \frac{\text{Area}}{(\text{Technology}/65)^2} \\
P_{\text{nor}} &= \frac{\text{Power}}{(\text{Technology}/65)}
\end{align*}
\]

The normalized area and power are 1.62 mm² and 46 mW, respectively. The superiority of the proposed design is obvious. [8] and [9] only focus on the design of 512-point FFT processor. The area is normalized as follows:

\[
A_{\text{nor}} = \frac{\text{Area}}{(\text{FFT length}/2048) \times (\text{Technology}/65)^2}
\]

The normalized area of [8] and [9] are 1.63 mm² and 1.69 mm², respectively. The core area of proposed design is also smaller.

### Table V. Comparison of several parallel FFT processors

| Works | Proposed | [5] | [7] | [8] | [9] |
|-------|----------|----|----|----|----|
| Architecture | MDF | MDF | MDC | MDF | MDF |
| FFT length | 64–2048/1536 | 128–2048/1536 | 2048/1024/512/128 | 512 | 512 |
| Word length (bit) | 12 | 12 | 10 | 12 | 14 |
| Clock rate (MHz) | 40 | 20 | 40 | 310 | 220 |
| Parallelism | 4 | 8 | 4 | 8 | 8 |
| Technology (nm) | 65 | 65 | 90 | 90 | 130 |
| Supply voltage (V) | 1 | 0.45 | 1 | 1.2 | 1.2 |
| Area (mm²) | 1.22 | 1.25 x 1.1 | 3.1 | 0.78 | 1.69 |
| Power (mW) | 11.5 | 8.6 | 63.7 | 92.8 | - |
| Execute time (µs) | 14.63 | 102.40 | 51.20 | - | - |

In this paper, we discuss the design of a MDF mixed-radix variable-length FFT processor. In order to minimize the number of occupied complex multipliers, we adopt radix-2³ FFT algorithm and 4 parallel processing channels. Combining with a
reconfigurable radix-$L$ butterfly unit, the processor can perform $2^k$ or $2^{3k} \times 3$-point FFT. With the design and optimization of CSD constant multipliers, twiddle factor generation block and radix-2/3/4 multiplexing butterfly unit, the proposed design obtains shorter processing latency, smaller core area and lower power consumption compared with several related works.

**Acknowledgements**

This work was supported by the Chang Jiang Scholars Programme under Grant T2012122, the Hundred Leading Talent Project of Beijing Science and Technology under Grant Z141101001514005.