IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 5 June 2023, accepted 12 July 2023, date of publication 19 July 2023, date of current version 25 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3296873

RESEARCH ARTICLE

Reconfigurable Hyper-Parallel Fast Fourier Transform Processor Based on Bit-Serial Computing

TINGYONG WU[1], (Member, IEEE), YUXIN WANG[1], (Student Member, IEEE), AND FUQIANG LI[2]

[1]National Key Laboratory of Wireless Communications, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China
[2]Key Laboratory of Technology on Datalink, China Electronics Technology Group Corporation (CETC), 20th Institute, Xi'an, Shanxi 710068, China

Corresponding authors: Tingyong Wu (wuty75@uestc.edu.cn) and Fuqiang Li (lfqnet@qq.com)

This work was supported in part by the China Electronics Technology Group Corporation (CETC) Key Laboratory of Data Link Technology under Grant CLDL-20202415, and in part by the Sichuan Science and Technology Program under Grant 2021YFG0127.

ABSTRACT The upcoming 5G communication is committed to providing ultra-high throughput and ultra-low delay service. However, digital signal processing technologies will be a critical challenge with the increasing bandwidth and transmitting streams. Especially, traditional fast Fourier transform (FFT) architectures are hard to meet the high-speed, high-performance, and low-overhead requirements. To overcome this issue, this article proposes a reconfigurable hyper-parallel bit-serial (HPBS) FFT processor. A bit-serial paradigm implements the datapath to reduce the hardware cost due to high parallelism as high as 64 to support FFT size from 64 to 2048. The HPBS design brings several significant advantages. Firstly, the hyper-parallel architecture supports large FTT radix with lower computation complexity, e.g., fewer processing stages and constant multipliers than conventional multipliers. Secondly, the bit-serial design enables the hyper-parallel FFT implemented by an acceptable hardware cost. Thirdly, the HPBS design supports the flexible and dynamic adjusting between latency and precision, which provides an additional optimization dimension. Based on the 55-nm process, the proposed 2048-point FFT can provide 32.128 Gbps throughput at 502MHz frequency with an average 12-bit word length. Besides, the normalized hardware efficiency can be up to 46.630 Gbps/mm^2, which is at least 2X that of the traditional design.

INDEX TERMS Fast Fourier transform (FFT), bit-serial, MIMO-OFDM, mixed-radix factorization, digital signal processing (DSP).

I. INTRODUCTION

For meeting the increasing requirements for high-speed and quality-of-service, the upcoming fifth-generation (5G) and beyond-5G (B5G) wireless communication systems are committed to providing multi-gigabit transmission, and ultra-low millisecond-level delay service [1], [2]. Consequently, 5G and B5G systems can meet the higher transmission rate, stability, and latency requirements in various applications, such as the internet of vehicles and remote telesurgery [3], [4], etc. The above various applications promote the development of digital signal processing (DSP) technology towards ultra-high throughput and low-power consumption.

As one of the most important DSP algorithms in 5G, the fast Fourier transform (FFT) is widely applied in communication systems [5], especially the multiple-input multiple-output (MIMO) orthogonal frequency division multiplexing (OFDM) system. And the MIMO-OFDM system is considered to be one of the most significant technologies for wireless communication in the 5G new radio (NR) standards [6]. In a demodulator of MIMO-OFDM system, multiple FFT processors are required to realize the demodulation of the multiple subcarriers. As the number of antennas in massive MIMO technology increases, hundreds and thousands of FFT processors are required in a MIMO-OFDM system.

The associate editor coordinating the review of this manuscript and approving it for publication was Tianhua Xu.

The increasing processing bandwidth of the MIMO-OFDM system in 5G puts forward higher processing speed requirements. Besides, the low-latency characteristics of the 5G system also put forward higher throughput requirements for FFT processors. Since the FFT algorithm consists of intensive complex multiplication and addition/subtractions, the various FFT architectures lead to huge differences in system complexity, processing delay, and hardware overhead. Therefore, a high-throughput and low-complexity FFT architecture has always been a hot research issue for 5G applications [7]. To solve the above challenges, various FFT architectures have been investigated over the past decades [8]. Compared with the iterative (known as memory/cache-based) FFT architectures [9], [10], the pipelined FFT architecture is more widely adopted for real-time applications due to the high-throughput and low latencies advantage [11], [12], [13]. The commonly pipelined FFT architectures can be classified into single-path delay commutator (SDC) [14], [15], multi-path delay commutator (MDC) [12], [16], [17], [18], single-path delay feedback (SDF) [19], [20], [21], [22], and multi-path delay feedback (MDF) [23], [24], [25], etc. In [14], an SDC-based FFT is characterized by the bit-dimension permutation of serial data. Furthermore, a real-valued SDC-based FFT architecture can achieve full hardware utilization by mapping each stage to a half-butterfly operation on real input signals [15]. In [12], a radix-2^k is proposed for any number of parallel samples, which is a power of two. And a rotator allocation approach is proposed to distribute the rotations and reduce the complexity of FFT in [16]. Moreover, a modified MDC-based FFT is proposed to achieve the FFT computation of the two independent data streams in [17]. Based on the power-area tradeoff, an efficient mapping of the pipeline SDF-based FFT architecture is proposed in [19]. And a flexible SDF-based pipeline FFT architecture with variable points is proposed in [20] and [21]. In order to improve the throughput and the process efficiency, more delay feedback paths can be used as in MDF architecture. In [23], a mixed-decimation MDF approach is proposed for radix-2^k FFT implementation. In addition, a multimode MDF-based FFT processor is proposed to support the flexible-radix-configuration and variable-length and multiple streams in [24]. However, as the digital chips are gradually developing into the Post-Moore era, a new paradigm is required to design the FFT for the future high complexity and power-area limited communication system.

By reviewing the development of the FFT processor, there are two possible directions for further improving FFT hardware performance, higher FFT radix with high parallel (par) architecture and low bit-width of the datapath. A large radix module leads to fewer radix stages, which means more low-cost constant multipliers can be applied to reduce the hardware costs instead of a general multiplier. For example, if a parallel radix-64 module is employed for 4096-point FFT, only two radix-64 stages are required, with constant coefficients the multiplications inside radix-64. Unfortunately, the high parallel and large radix FFT brings two critical challenges: the high hardware cost and ultra-complex parallel data addressing. On the other hand, reducing the bit-width of datapath can effectively reduce the hardware cost but with the penalty of precision loss.

To overcome the above two challenges, we propose a hyper-parallel bit-serial (HPBS) FFT processor based on the mixed-radix decomposition. In the proposed HPBS processor, a hybrid architecture of a parallel radix and multiple SDF is adopted. Besides, the hyper-parallel FFT datapath is implemented by a bit-serial paradigm. Consequently, the hyper-parallel large radix FFT can provide high throughput with acceptable hardware costs. Furthermore, the hybrid architecture can avoid intermediate-level data management circuits and storage resources. In our design, the first stage is a fully-parallel high radix module, i.e., radix-64. The second stage is multiple SDF FFT, i.e., 64-parallel 32-point SDF FFT. Thus, the HPBS FFT can perform full-pipelined data processing without additional memory. Moreover, the variable-length FFT is supported from 64, 128, 256, 512, 1024, to 2048. Although a low-power FFT proposed in [26] adopted bit-serial architecture to reduce the power consumption. However, the parallelism of this FFT is low since the design only considers the low-speed and low-throughput applications of the Internet of things (IoT) and ignores the problems in the high-parallel design under the requirements of 5G applications. Besides, the work also does not give the theoretical analysis of the impact of FFT quantization on the overall system.

For the traditional application, the bit-width of FFT is chosen to satisfy the worst-case scenarios. For example, 16-bit width is usually used in communication FFT processors to cover all communication scenarios in terms of different modulation coding scheme (MCS) and signal-to-noise ratio (SNR). However, if we review the purpose of communication systems, we find it is unnecessary to employ the large bit-width FFT processor for all scenarios. For instance, if the input signal SNR is high while MCS is low, we can employ a small bit-width FFT processor to demodulate the signal. That is to say, the variable-precision datapath in a communication system is a feasible method to accept the precision loss. The reason is that for the communication system, the purpose is to guarantee the communication quality, i.e., bit error rate (BER) performance instead of single module processing precision. Therefore, the bit-serial datapath is a good choice to implement the variable-precision datapath. This paper analyzes the quantization error in a MIMO-OFDM system and considers it as an additional dimension for the FFT design, which enables the dynamic adjusting between latency and communication performance. To this end, we can evaluate the performance of the proposed FFT by the equivalent average bit-width, i.e., 12-bit width, in our design, instead of the worst case. Due to the bit-serial scheme, we can dynamically adjust the bit width to compensate for the performance loss.

We summarize the advantages of the proposed HPBS FFT as follows.

- **Hyper-parallel hybrid architecture.** The proposed bit-serial FFT adopts a hybrid parallel and SDF architecture based on the radix factorization. The front stage adopts a high parallel and large-radix design, and the latter stage adopts the SDF structure to complete the full-pipeline operation without additional memory. Moreover, the variable-length FFT is supported, i.e., 64, 128, 256,512, 1024 and 2048-point FFT.
- **Bit-serial pipelined FFT processor.** In order to implement the hyper-parallel FFT with acceptable hardware cost, the bit-serial paradigms are adopted in HPBS datapath. More importantly, the HPBS can dynamically adjust the bit width of the input signal according to the SNR without changing the hardware design, which avoids the worst-case scenario.
- **Support variable bit width for different communication scenarios.** In order to satisfy the system requirements, the calculation precision is variable dynamically to adapt to the signals with different SNR. The dynamic balance between channel noise and processing noise is a flexible exchange of resources and performance in essence.

In summary, the proposed bit-serial FFT can dynamically adjust the calculation precision to match the requirements of various communication scenarios, avoiding the waste of computation resources by worst-case design.

This paper is organized as follows. Section II presents a review of FFT algorithms and MIMO-OFDM systems. Then the error analysis of the FFT with finite precision is provided in section III. Next, the proposed bit-serial pipelined FFT based on parallel-SDF architecture is shown in section IV. Moreover, section V presents the performance evaluation for finite-precision FFT. Finally, section VI summarizes the article.

II. THE FFT OVERVIEW

A. THE FFT ALGORITHM

The N-point discrete Fourier transform (DFT) of an input sequence $\boldsymbol{x}[n]$ is defined as

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

where $\boldsymbol{X}[k]$ is the spectral output at frequency k, and $W_N^{nk} = e^{-j\frac{2\pi}{N}nk}$ are a set of the twiddle factors. The above equation shows that the computation amount of N-point DFT is approximately proportional to N^2.

Based on the mixed-radix decomposition, the Cooley-Turkey algorithm (CTA) can map a one-dimensional (1D) DFT with size $N = N_1N_2$ into an N_1 by N_2 two-dimensional (2D) DFT, plus the complex multiplication with the twiddle factors [27]. Especially, the complex multiplications is $1, -j, -1$, and j respectively when $n \in [0, N/4, N/2, 3N/4]$ since the corresponding data must be rotated by $0^o, 270^o, 180^o, 90^o$. The above rotations are considered trivial since they can be performed by interchanging the real and imaginary parts and/or changing the sign of the data, which can greatly reduce the amount of computation [13]. As a result, the recursive decomposition of the mixed-radix CTA (MR-CTA) can effectively reduce the complexity to $o(N\log_2 N)$. Then based on (1), the MR-CTA FFT can be expressed as

$$\boldsymbol{X}[k] = \sum_{n_2=0}^{N_2-1} W_{N_2}^{n_2k_2} W_N^{n_2k_1} \left(\sum_{n_1=0}^{N_1-1} \boldsymbol{x}[n] W_{N_1}^{n_1k_1} \right), \tag{2}$$

where

$$\begin{aligned} n_1, k_1 &\in \{0, 1, \cdots, N_1 - 1\} \\ n_2, k_2 &\in \{0, 1, \cdots, N_2 - 1\}. \end{aligned} \tag{3}$$

Then the general factorization of the MR-CTA rewrites the indices n and k as

$$\begin{aligned} n &= N_2n_1 + n_2 \\ k &= k_1 + N_1k_2. \end{aligned} \tag{4}$$

From the above equation, it re-indexes the input n and output k as N_1 by N_2 two-dimensional arrays in column-major and row-major order, respectively. Specifically, the MR-CTA first completes N_2 times of N_1-point DFT in column-major order, then performs a rotation adjustment based on the twiddle factors, and finally completes N_2 times of N_1-point DFT in row-major order.

Based on the MR-CTA form in (2), each inner sum is a DFT of size N_2, each outer sum is a DFT of size N_1, and the item $W_N^{n_2k_1}$ are the twiddle factors. Thus, the MR-CTA FFT can be matrixed as

$$mat\{X\} = \left[\left(D_{N_1} mat(x) \right) \odot \Omega \right] D_{N_2}, \tag{5}$$

where $mat\{\cdot\}$ represents the matricization that can reshape a N-point vector into a $N_1 \times N_2$ matrix. Here, $\odot$ represents the hardmard product, $\Omega = \left[W_N^{n_2k_1} \right]_{N_1 \times N_2}$ represents the matrix of twiddle factors. Besides, $D_{N_1} \in \mathbb{C}^{N_1 \times N_1}$ and $D_{N_2} \in \mathbb{C}^{N_2 \times N_2}$ represent the DFT matrix. The above equation shows that the DFT of size N is decomposed into two DFT operations with smaller size N_1 and N_2 respectively and the complex multiplication with the twiddle factors.

B. THE FFT IN MIMO-OFDM SYSTEM

The MIMO-OFDM technology can achieve high-speed and high-efficiency transmission as a key technology in the 5G NR standards. In particular, the OFDM technology is a multi-carrier communication scheme that divides the channel spectrum into multiple orthogonal subbands. And the modulation and demodulation of multiple orthogonal sub-carriers are based on the inverse FFT (IFFT) and FFT operation, respectively. Furthermore, multiple data streams are distributed on each subband to achieve independent parallel transmission.

Fig. 1 shows the diagram of the MIMO-OFDM system. Here, N represents the number of subcarriers. And N serial

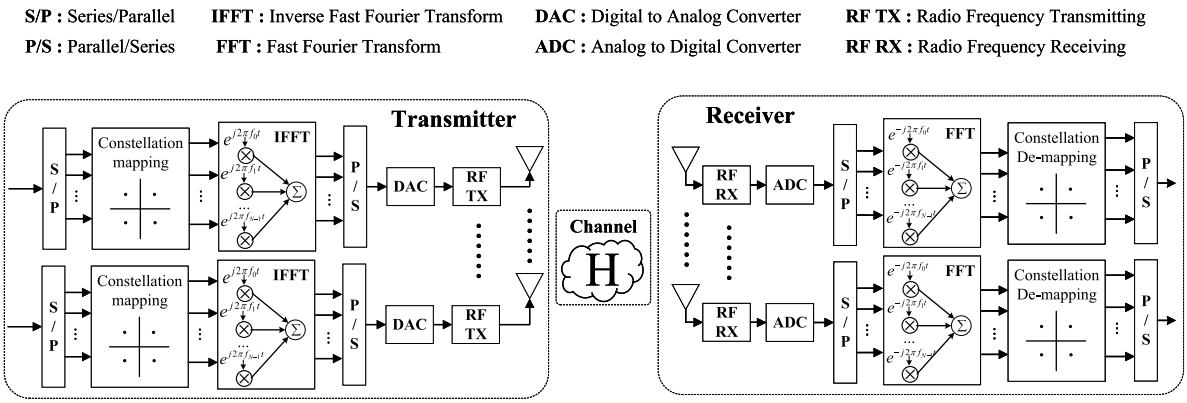

FIGURE 1. The diagram of the MIMO-OFDM system.

data can be turned into a group of N parallel data. Besides, $d_0, d_1, \cdots, d_{N-1}$ is a group of parallel data after constellation mapping, and is modulated on N sub-carriers with carrier frequency $f_i, i \in \{0, 1, \cdots, N-1\}$ respectively. The frequencies of different sub-carriers satisfy the constraint $f_i = f_c + i \cdot \Delta f$, where $\Delta f = 1/T$ represents the channel bandwidth of the sub-carrier, and T represents the duration time of the OFDM symbol on each subcarrier. Then the OFDM signal $s(t)$ is formed by superimposing the signals of each sub-carrier in the time-domain, which can be expressed as

$$s(t) = \begin{cases} \sum_{i=0}^{N-1} d_i \cdot e^{j2\pi f_i (t-t_s)}, & t \in [t_s, t_s + T] \\ 0, & t \notin [t_s, t_s + T], \end{cases} \tag{6}$$

where t_s is the start time of a MIMO-OFDM symbol, and d_i is the data symbol of each sub-channel. Through FFT operation, the symbol d_i can be modulated to a sub-carrier and mapped to the i_{th} transmitting antenna. All modulated signals will be transmitted by N_t antennas at the same time.

For the convenience of discussion, only the baseband representation of the communication signal is concerned, so the carrier is not considered here, i.e., the carrier frequency f_c can be assumed to be 0. Without loss of generality, the start time t_s of a OFDM symbol can be assumed to be 0, hence $t_s = 0$. And the received signal is sampled at the intervals of $T_s = T/N$, hence $t = kT_s$, where $k \in \{0, 1, \cdots, N-1\}$. Then the received signal $s[k]$ can be re-expressed as

$$s_k = s(kT_s) = \sum_{i=0}^{N-1} d_i \exp\left\{j\frac{2\pi jk}{N}\right\}. \tag{7}$$

From the above equation, the discrete OFDM signal s_k is equivalent to IFFT operation on the data d_i. Likewise, the original data symbol d_i can be obtained by IDFT of the signal s_k, which can be represented as

$$d_i = \sum_{k=0}^{N-1} s_k \exp\left\{-j\frac{2\pi ik}{N}\right\}, \quad i \in \{0, 1, \cdots, N-1\}. \tag{8}$$

Therefore, the FFT/IFFT algorithms, as the core module for data processing in the OFDM system, can complete the correct modulation and demodulation of the OFDM system. As the number of antennas in the MIMO-OFDM system increases, FFT size increases. Thus the complexity of the FFT module is also rapidly growing. Therefore, we propose a bit-serial pipelined FFT processor based on a serial and parallel hybrid architecture to improve hardware efficiency.

III. QUANTIFICATION AND COMMUNICATION PERFORMANCE

Generally, hardware platforms can only perform operations with limited precision, so the hardware processing inevitably introduces truncation errors to the operation results. This section will mainly focus on the quantization SNR of FFT and its effect on the MIMO-OFDM system.

A. THE QUANTIZATION ANALYSIS OF FFT

The digital systems need to quantify signals and parameters, which inevitably generate the quantization noise. Relative to the module output, the quantization noise is a kind of random disturbance, which can be analyzed using an appropriate noise analysis model and linear theory. Although the quantization is not a linear operation, the quantization noise can be analyzed since it is independent of other signals.

Based on the quantization error analysis of linear systems in [28], the received signal $y \in \mathbb{C}^{N_r \times 1}$ is quantized as

$$y_q = Q(y) = y + \bar{e}, \tag{9}$$

where $Q(\cdot)$ represents the quantization operations, $\bar{e} \in \mathbb{C}^{N_r \times 1}$ represents the quantization error introduced by analog-to-digital converters (ADC), $y_q \in \mathbb{C}^{N_r \times 1}$ represents the quantized signal. Here, N_r and N_t represent the number of receive antennas and transmit antennas, respectively.

According to [28], the each element $\bar{e}_i, i \in \{1, 2, \cdots, N_r\}$ of quantization error $\bar{e}$ is independent of each other and is assumed to satisfy a uniform distribution. Besides, the real and imaginary part are also independent of each other and satisfies the same uniform distribution that depends

on the quantization width w, which can be expressed as $Re[\bar{e}_i], Im[\bar{e}_i] \overset{iid}{\sim} U(-2^{-w}/2, 2^{-w}/2)$. Here, $Re[\cdot]$ and $Im[\cdot]$ represent the real and imaginary parts of a complex data, respectively. Then the probability density function of the real and imaginary parts of $\bar{e}_i$ can be expressed as $p_{\bar{e}_i} = 2^w$. Consequently, the mathematical expectation $\mathbb{E}[\bar{e}_i]$ of the error element $\bar{e}_i$ can be expressed as

$$\mathbb{E}[\bar{e}_i] = \mathbb{E}\{Re[\bar{e}_i]\} + j * \mathbb{E}\{Im(\bar{e}_i)\} = 0. \tag{10}$$

In addition, the variance $\mathbb{D}[\bar{e}_i]$ of the random variable $\bar{e}_i$ can be expressed as

$$\begin{aligned}\mathbb{D}[\bar{e}_i] &= \mathbb{E}\{\bar{e}_i \cdot \bar{e}_i^*\} = \mathbb{E}\{Re^2[\bar{e}_i] + Im^2[\bar{e}_i]\} \\ &= 2\int_{-2^{-w}/2}^{2^{-w}/2} u^2 \cdot p_{\bar{e}_i} du = 4^{-w}/6,\end{aligned} \tag{11}$$

where $(\cdot)^*$ represents the conjugation. Since the above variance is proportional to 4^{-w}, if the quantization bit width is increased by 1 bit, the quantization error can be reduced by 4 times. In summary, due to the mutual independence between the elements, the mathematical expectations $\mathbb{E}[\bar{e}]$ can be expresses as $\mathbb{E}[\bar{e}] = 0$. Meanwhile, the covariance matrix $\mathbb{D}[\bar{e}]$ only has non-zero values $\mathbb{D}[\bar{e}_i] = 4^{-w}/6$ on the diagonal, and the other elements are all 0. Hence, $\mathbb{D}[\bar{e}] = 4^{-w}/6 \cdot I_{N_r}$, where I_{N_r} represents the identity matrix with dimension N_r by N_r.

B. QUANTIZATION SNR DERIVATION OF FFT

The quantized signal is passed into the FFT module for operations. Due to physical constraints, both the input signals and twiddle factors need to be quantized. Since the bit width will be doubled during the multiplication operation, the product result needs to be rounded to keep the total bit width unchanged. In the above process, a round-off error is introduced into the FFT operations. In order to facilitate the analysis of the round-off errors, the hardware FFT processor can be abstracted as a module with external input error and full precision in [29]. Therefore, the output $Y_w \in \mathbb{C}^{N_r \times 1}$ of FFT can be represented as

$$Y_w = fft(y, w) = D(y + \bar{e}) + \tilde{e}, \tag{12}$$

where $D \in \mathbb{C}^{N_r \times N_r}$ represents the DFT matrix, $\tilde{e} \in \mathbb{C}^{N_r \times 1}$ represents the roundoff error generated by the multiplication and addition of FFT. According to the characteristics of FFT operation, the each element $\tilde{e}_i, i \in \{1, 2, \cdots, N_r\}$ of roundoff error $\tilde{e}$ is independent of each other. Besides, the real and imaginary parts are also independent of each other and satisfies the same uniform distribution that depends on the quantization width w, hence $Re[\tilde{e}_i], Im[\tilde{e}_i] \overset{iid}{\sim} U(-2^{-w}/2, 2^{-w}/2)$. Likewise, the mathematical expectation $\mathbb{E}[\tilde{e}_i]$ of the round-off error element $\tilde{e}_i$ can be expressed as

$$\mathbb{E}[\tilde{e}_i] = \mathbb{E}\{Re[\tilde{e}_i]\} + j * \mathbb{E}\{Im[\tilde{e}_i]\} = 0. \tag{13}$$

In addition, its variance $\mathbb{D}[\bar{e}_i]$ can be expressed as

$$\mathbb{D}[\tilde{e}_i] = \mathbb{E}\{\tilde{e}_i \cdot \tilde{e}_i^*\} = \mathbb{E}\{Re^2[\tilde{e}_i] + Im^2[\tilde{e}_i]\} = 4^{-w}/6. \tag{14}$$

In summary, due to the mutual independence between the elements, the mathematical expectations $\mathbb{E}[\tilde{e}]$ of the quantization error $\tilde{e}$ can be expresses as $\mathbb{E}[\tilde{e}] = 0$. Meanwhile, the covariance matrix $\mathbb{E}[\tilde{e}]$ only has non-zero values $\mathbb{D}[\tilde{e}_i] = 4^{-w}/6$ on the diagonal, and the other elements are all 0, hence $\mathbb{D}[\tilde{e}] = 4^{-w}/6 \cdot I_{N_r}$.

Based on (10), (12) and (13), the mathematical expectation $\mathbb{E}(Y_w)$ of the FFT output Y_w can be represented as

$$\mathbb{E}(Y_w) = \mathbb{E}\{Dy\} + D \cdot \mathbb{E}\{\bar{e}\} + \mathbb{E}\{\tilde{e}\} = \mathbb{E}\{Y\}, \tag{15}$$

where $Y = Dy$ represents the full-precision DFT results towards signal y. As shown in the above equation, the quantization error $\bar{e}$ of the input and the roundoff error $\tilde{e}$ during the operation process are unbiased and has no effect on the expectation of the FFT output, which indicates that the error introduced by the limited-precision digital system due to the bit width limitation will not affect the signal detection and estimation algorithms.

Since the MR-CTA FFT reduces the non-trivial twiddle factors and operation complexity [30], thus it is applied to analyze the quantization SNR of the FFT algorithm. According to (12), the hardware output Y_w of a MR-CTA FFT can be modeled as

$$mat\{Y_w\} = [\left(D_{N_1} mat\left(y_q\right) + \tilde{e}_{D_{N_1}}\right) \odot \Omega + \tilde{e}_\Omega] D_{N_2} + \tilde{e}_{D_{N_2}}. \tag{16}$$

Here, $\tilde{e}_{D_{N_1}} \in \mathbb{C}^{N_1 \times N_2}$ represents the roundoff error during the N_1-point FFT, $\tilde{e}_\Omega \in \mathbb{C}^{N_1 \times N_2}$ represents the roundoff error in matrix multiplication of twiddle factors in the MR-CTA algorithm, $\tilde{e}_{D_{N_2}} \in \mathbb{C}^{N_1 \times N_2}$ represents the roundoff error during the N_2-point FFT. Thus the error vector $n_w = Y - Y_w$ of the MR-CTA FFT can be expressed as

$$\begin{aligned}mat\{n_w\} = &\left[\left(D_{N_1} mat(\bar{e}) + e_{D_{N_1}}\right) \odot \Omega\right] D_{N_2} \\ &+ e_\Omega D_{N_2} + e_{D_{N_2}}.\end{aligned} \tag{17}$$

Since the quantization error $\bar{e}$ and roundoff errors $e_{D_{N_1}}$, e_Ω, $e_{D_{N_2}}$ are caused by truncation rounding, their elements are independent of each other and satisfy the same distribution. And the random variable ξ is introduced to represent the above error matrix, where $Re[\xi], Im[\xi] \overset{iid}{\sim} U(-2^{-w}/2, 2^{-w}/2)$. Then the error vector n_w in (17) can be reexpressed as

$$n_w = \xi \cdot vec\{G\}, \tag{18}$$

where $vec\{\cdot\}$ represents the vectorization. And the symbol $G \in \mathbb{C}^{N_1 \times N_2}$ is defined to simplify the representation, which is given as

$$G = \left[\left(D_{N_1} I + I\right) \odot \Omega\right] D_{N_2} + I \cdot D_{N_2} + I, \tag{19}$$

where the elements of the matrix $I \in \mathbb{R}^{N_1 \times N_2}$ are all 1.

The signal-to-quantization-noise ratio (SQNR) γ can be expressed as

$$\gamma(dB) = 10 \cdot \log_{10}\left(\frac{P_x}{N_w}\right) = 10 \cdot \log_{10}\left(\frac{\mathbb{E}\{x^H x\}}{\mathbb{E}\{n_w^H n_w\}}\right). \tag{20}$$

Here, $P_x = \mathbb{E}\{x^H x\}$ represents the signal power and $N_w = \mathbb{E}\{n_w^H n_w\}$ represents the noise power. Based on the (19), the noise power N_w can be simplified as

$$N_w = \mathbb{E}\left\{n_w^H n_w\right\} = \|G\|_F \cdot \mathbb{E}\left\{\xi^2\right\} = 2^{-2w}\|G\|_F/6. \quad (21)$$

And the quantization SNR γ in (20) can be re-expressed as

$$\gamma = 6P_x \cdot 4^w/\|G\|_F. \quad (22)$$

The above equation indicates that if the bit width increased by one bit, the quantization SNR could improve four times.

C. EXCHANGE OF PRECISION AND PERFORMANCE

Considering the received signal $y \in \mathbb{C}^{N_r \times 1}$ of the MIMO-OFDM system can be expressed as

$$y = Hx + n. \quad (23)$$

Here, $H \in \mathbb{C}^{N_r \times N_t}$ represents the channel with additive white Gaussian noise (AWGN). The noise $n \in \mathbb{C}^{N_r \times 1}$ satisfies a complex Gaussian distribution with mean 0 and variance matrix ${\sigma_n}^2 \boldsymbol{I}_{N_r}$, hence $n \sim \mathcal{CN}(0, \sigma_n^2 \boldsymbol{I}_{N_r})$. For ease of analysis, the input signal $x \in \mathbb{C}^{N_t \times 1}$ is assumed to satisfy a complex Gaussian distribution with mean 0 and variance matrix ${\sigma_x}^2 \boldsymbol{I}_{N_r}$, hence $x \sim \mathcal{CN}(0, \sigma_x^2 \boldsymbol{I}_{N_r})$.

In a MIMO-OFDM system, multiple FFT processors are required to perform demodulation of the received multi-carrier signal y. And the demodulated signal $\hat{y}_w$ can be given as

$$\hat{y}_w = fft\,(y, w) = Hx + n + n_w, \quad (24)$$

where n_w represents the error brought by the FFT module with finite precision. Besides, an equalizer is required to eliminate or reduce the inter-symbol interference (ISI) problem caused by the multipath delay in MIMO-OFDM system. Since minimum mean-square error (MMSE) equalizer can maximize the signal to interference plus noise ratio (SINR) and has better linear detection performance, thus it is widely used in communication systems [31]. The MMSE equalizer $W \in \mathbb{C}^{N_t \times N_r}$ can be expressed as

$$W = \left(H^H H + \sigma_n^2 \cdot I\right)^{-1} H^H, \quad (25)$$

where the noise power σ_n^2 can be obtained by statistics, and the channel H can be obtained by channel estimation. In order to avoid the influence of channel estimation, the channel H can be assumed to be obtained by perfect estimation. Then the signal $\hat{x}$ equalized by MMSE equalizer can be expressed as

$$\hat{x} = WHx + Wn + Wn_w. \quad (26)$$

Since original signal x, the channel noise n, and the processing error n_w of the FFT module are independent of each other, and their means are all 0. Then the mean $\mathbb{E}\{\hat{x}\}$ of equalized signal $\hat{x}$ can be given as

$$\mathbb{E}\{\hat{x}\} = WH \cdot \mathbb{E}\{x\} + W \cdot \mathbb{E}\{n\} + W \cdot \mathbb{E}\{n_w\} = 0 \quad (27)$$

Besides, the covariance matrix $\mathbb{D}(\hat{x})$ of equalized signal $\hat{x}$ can be given as

$$\mathbb{D}\{\hat{x}\} = WHH^H W^H \cdot \sigma_x^2 + WW^H \cdot \left(\sigma_n^2 + 4^{-w}\|G\|_F/6\right). \quad (28)$$

Here, the first item represents the power of the received signal; the second item represents the effect of channel noise that reflects the quality of the channel; the third item represents the impact of FFT with limited processing precision.

In mobile communications, since the wireless channel changes with time and frequency, the receiver of the MIMO-OFDM system statistics the noise power. When the detected channel noise power increases, the channel deteriorates at this time, and the communication quality of the MIMO-OFDM system will decrease. At this time, the processing noise of the FFT processor needs to be reduced by increasing the quantization bit width of the FFT module to ensure the communication performance of the MIMO-OFDM system. On the contrary, when the detected channel noise power decreases, the channel is improved. Thus the communication quality of the MIMO-OFDM system improves at this time. While meeting the communication requirements of the whole MIMO-OFDM system, the processing noise can be appropriately decreased by decreasing the quantization bit width of the FFT module.

In summary, the bit-serial design can support the variable precision and the flexible exchange of resources and performance. Therefore, the receiver of the MIMO-OFDM system can dynamically adjust the processing precision of the bit-serial FFT according to the statistical noise information.

IV. PRECISION ADJUSTMENT AND BIT-SERIAL DESIGN

A. THE CHALLENGE OF FFT PRECISION ADJUSTMENT

Due to the high throughput, the traditional hardware implementation is mainly based on the bit-parallel architecture. However, the signals in traditional bit-parallel hardware are represented by finite parallel circuits due to hardware resource constraints. Thus the bit-parallel hardware has a fixed processing width and resource overhead, making it naturally unable to support variable operation precision. If the processing width of bit-parallel hardware is expected to be variable, only the hardware structures with different widths can be reconstructed. Moreover, to ensure sufficient processing ability in various scenarios, the processing ability must be designed for the worst-case scenario and exceeds the requirements of most scenarios, which inevitably leads to a waste of resources and energy.

Unlike the bit-parallel design, the input and output of bit-serial units are a bit-level pipeline, i.e., the bit-serial unit only processes one bit per clock. Suppose the operation precision of the bit-serial unit is expected to be variable, the quantization bit width of the input can be adjusted according to the requirements, and the bits are input into the bit-serial operation unit. Furthermore, the least significant bit (LSB) is input and output first, while the most significant bit (MSB)

TABLE 1. The resource comparison of multiplier and adder with bit-parallel and bit-serial architecture.

Width			4bit			8 bit			12bit			16bit		
Resource Type			FA	D	MUX	FA	D	MUX	FA	D	MUX	FA	D	MUX
Adder	Bit-Parallel	Quantity	4	0	0	8	0	0	12	0	0	16	0	0
		Latency (CC)	1			1			1			1		
	Bit-Serial	Quantity	1	1	1	1	1	1	1	1	1	1	1	1
		Latency (CC)	4			8			12			16		
Multiplier	Bit-Parallel	Quantity	48	0	0	112	0	0	176	0	0	240	0	0
		Latency (CC)	1			1			1			1		
	Bit-Serial	Quantity	15	47	32	15	47	32	15	47	32	15	47	32
		Latency (CC)	16			16			16			16		

* FA: full adder. D: one register. MUX: multiplexer. CC: clock cycle.

is input and output later. Since the bit-serial design supports variable processing precision on hardware, it can provide the desired tradeoff among performance and resource overhead.

In the bit-serial units, the multiplexers (MUX) are required to isolate the carry between adjacent operations and avoid the adjacent data's influence. In addition, the control signal of each MUX changes periodically according to the bit width. Compared with the bit-parallel designs, the bit-serial multiplier and adder have a simple structure with low resources overheads, as shown in Table 1. Based on the bit-serial adders and multipliers, the hardware overheads of the FFT processor can be greatly reduced. Thus the bit-serial design has high hardware efficiency. In addition, since the bit-serial design shortens the critical path in the hardware units, it can work at a higher frequency. Therefore, the bit-serial design trades time resources for space resources, achieving extremely low hardware overhead and variable operation precision. In addition, the bit-serial design can improve the working frequency to increase the throughput by utilizing pipeline architecture.

Although bit-serial arithmetic trades performance with complexity and scalability, the traditional bit-serial logic has longer latency and low throughput. Therefore, the conventional bit-serial architecture is not suitable for the high throughput and low latency requirements of 5G-based applications, but it provides a way to support the variable precision. Unlike the bit-parallel design, the processing width in the bit-serial design depends on the length of the serial input, which realizes the flexible exchange of performance and hardware resources. Consequently, the proposed FFT utilizes the bit-serial design to achieve variable precision by adjusting the number of processing cycles. Besides, it also utilizes a parallel-serial hybrid architecture to improve the throughput. Moreover, the specific details are shown in section V.

B. BIT-SERIAL ADDER/SUBTRACTOR

Since the FFT are essential consists of a series of multiplication and addition, thus the basic multipliers and adders are significant to the resources consumption of the whole FFT processor. The basic multiplier and adder in the proposed FFT processor adopt a bit-serial design to reduce the resources overhead and support variable operation precision.

Fig. 2 shows the diagrams of the bit-serial adder and subtractor. Since both are composed of a full adder (FA) and a one-bit register, they all have a simple and similar structure. For the input with word-length l, the l clock cycles are required to complete the operation by the bit-serial adder/subtractor, while the MSB of the operation result is output on the l_{th} clock. The subsequent analysis is only based on the bit-serial adder without loss of generality. And the output bit s_i and carry bit c_i at the i_{th}, $i \in \{1, 2, \cdots, l\}$ clock of a bit-serial adder are given as

$$\begin{aligned} s_i &= \overline{a_i}\overline{b_i}c_{i-1} + \overline{a_i}b_i\overline{c_{i-1}} + a_i\overline{b_i}\overline{c_{i-1}} + a_ib_ic_{i-1} \\ c_i &= a_ib_i + a_ic_{i-1} + b_ic_{i-1}, \end{aligned} \tag{29}$$

where a_i and b_i respectively represent the input of the adder. In addition, c_{i-1} represents the carry bit from the last clock and the initial carry bit c_0 of each addition operation is 0, hence $c_0 = 0$. At the beginning of the operation, the LSB of the two inputs a and b take part in the operation first. The two data are input into the adder bit by bit during the operation process. The sum s_i is output directly, and the carry c_i is delayed by one clock and fed back to the bit-serial adder to achieve the operation at the next clock. Then iterative operations are performed bit by bit until the MSB is calculated, the carry c_l needs to be reset, which avoids the influence of the current operation on subsequent operations. In addition, the Pcontrol signal changes periodically according to the bit width l.

Different from the bit-serial adder, an inverter is set at the subtrahend input of the bit-serial subtractor, as shown in Fig. 2. Since the operation is based on the two's complement, the subtraction operation is equivalent to the bit-wise inversion of the input, then the bit "1" is added to the LSB. In addition, the initial carry bit c_0 should be 1 to ensure the functionality of the subtraction operation.

C. BIT-SERIAL MULTIPLIER

As shown in Fig. 3, a four-bit bit-serial multiplier paradigm is illustrated the principle of a general bit-serial multiplier.

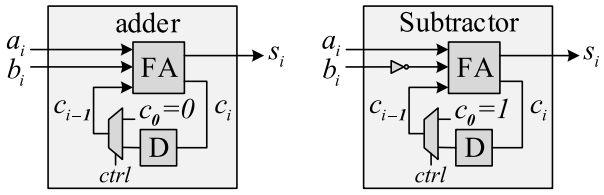

FIGURE 2. The diagram of the bit-serial adder and subtractor.

Here, the two input operators $\boldsymbol{a}$ and $\boldsymbol{b}$ are four-bit. From the paradigm, the structure of the bit-serial multiplier is determined by the bit width of the multiplicator $\boldsymbol{b}$. And both input $\boldsymbol{a}$ and output $\boldsymbol{p}$ are updated in a bit-level pipeline. When the bit width of the multiplicator $\boldsymbol{b}$ is fixed, the operation precision is determined by the bit width of the input $\boldsymbol{a}$. As the bit width of the input $\boldsymbol{a}$ increases, the operation precision of the bit-serial multiplier improves.

The bit-serial multiplier consists of multiple cascaded identical units shown in the dashed box in Fig. 3. The bit-serial unit consists of a FA, and multiple one-bit registers, and multiple MUXs. The cascade depth is determined by the bit width of the quantized multiplicator $\boldsymbol{b}$. Furthermore, the unit has the following key functions. The first is the bitwise *AND* (&) operation and accumulation; the second is the one-bit sign extension to prevent data overflow during operation process; the third is to truncate the LSB of the output to maintain the fixed bit width.

In particular, the input and output of a bit-serial multiplier are bit-level and full-pipelined. Thus the sign bit of the previous input is adjacent to the LSB of the current input. During the accumulation, to prevent the last operation's carry from affecting the present operation process, the enable signal of a MUX is switched to voltage level 0 only when the MSB of the previous operation ends and the LSB of the current data arrives. Otherwise, a MUX is enabled to connect to the one-bit register.

A one-bit sign extension of the current data and the low bit truncation of the previous data are performed. Similar to the bit-serial adder/subtractor, the control signal of the multiplexer is based on the periodic change of the bit width. Based on the four-bit multiplier paradigm, the enabling moments of all multiplexers are marked with numbers in Fig. 3. The overflow problem can not be solved entirely only by the internal sign extension. Consequently, there is also a one-bit sign extension for the input and a one-bit left shift for output, which completely solves the overflow problem of the traditional bit-serial multiplier in [29].

D. CONSTANT-COEFFICIENT BIT-SERIAL MULTIPLIER

According to [29], the structure of the constant-coefficient bit-serial multiplier is determined by the zero-bit and the valid bits (1 or -1) in the canonical-signed-digit (CSD) coding of the constant coefficient. When a certain bit of the constant coefficient is 0, the related operations are meaningless. Consequently, the valid bits determine the FAs and the MUXs in the constant-coefficient bit-serial multiplier. Besides, the zero-bits between adjacent valid bits determine the number of multiplexers. Compared with the general bit-serial multiplier, all zero-bit related operations are omitted in the constant-coefficient bit-serial multiplier. Thus it is obvious that a constant coefficient bit-serial multiplier has lower hardware overhead than a general bit-serial multiplier. Especially when constant-coefficient multipliers are used extensively, the resource consumption caused by many meaningless zero-bit operations can be saved.

Fig. 4 shows a paradigm of a bit-serial multiplier with the six-bit constant-coefficient encoded as "0 1 0 0 -1 0 1 0" by the CSD encoding. Unlike the one-bit sign extension and the truncation, the multi-bit sign extension and the truncation are applied in the constant-coefficient bit-serial multipliers by a longer delay line and more multiplexers. Moreover, the corresponding enable signal is marked in Fig. 4.

In the FFT processor, the twiddle factors of certain positions are fixed. Thus, the constant coefficient bit-serial multipliers are applied to reduce the multiplication resource consumption. In particular, a large number of constant-coefficient multipliers are required in a parallel FFT architecture.

In the bit-selrial Lyon multiplier, its resource consumption has a clear relationship with the data bit width, but the relationship between the resource consumption and the data bit width of the CSD multiplier is not fixed, which is mainly determined by the remaining significant bits (1 and -1) after CSD encoding. In order to facilitate comparison, the CSD multiplier resources are agreed as follows:

1) It is considered that the effective data bit width after CSD encoding is half of the actual data bit width.

2) The multiway gate used for symbol extension and truncation in the CSD multiplier is converted to the two equally functioning gate in the Lyon multiplier, and the number of both is the same. Therefore, the advantage of MUX resources is reflected in the reduction of the number of gate for zero setting at the carry input and output of the adder. After the above convention is made, the hardware resource cost of the CSD multiplier and the Lyon multiplier is shown in Table 2 when the data bit width is n.

For example, with a bit width of 12 bits, a constant coefficient bit serial multiplier requires only 5 FAs and 19 MUXs, while an ordinary bit serial multiplier requires 11 FAs and 24 MUXs. The architecture proposed in this paper is a fully parallel FFT architecture, and a large number of constant coefficient multipliers save a lot of multiplier resource consumption.

V. BIT-SERIAL FFT WITH PARALLEL-SERIAL HYBRID ARCHITECTURE

Although the bit-serial designs can realize the flexible exchange of resources and performance, the traditional bit-serial FFT has a low throughput. To improve the throughput to meet the requirements of 5G-based applications,

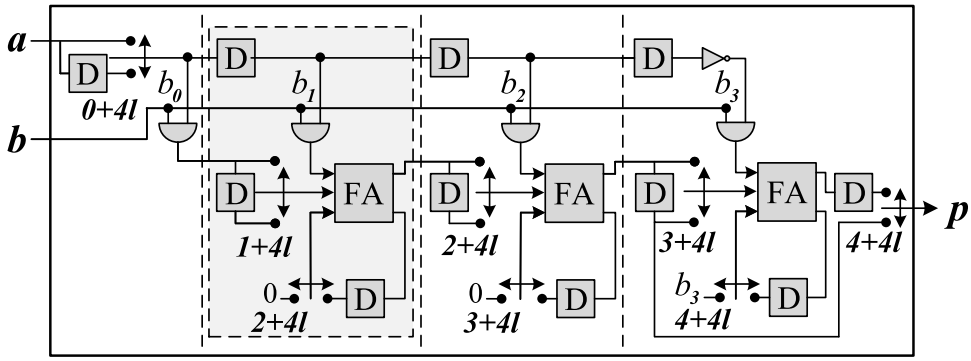

FIGURE 3. The diagram of the bit-serial multiplier paradigm with four bit width.

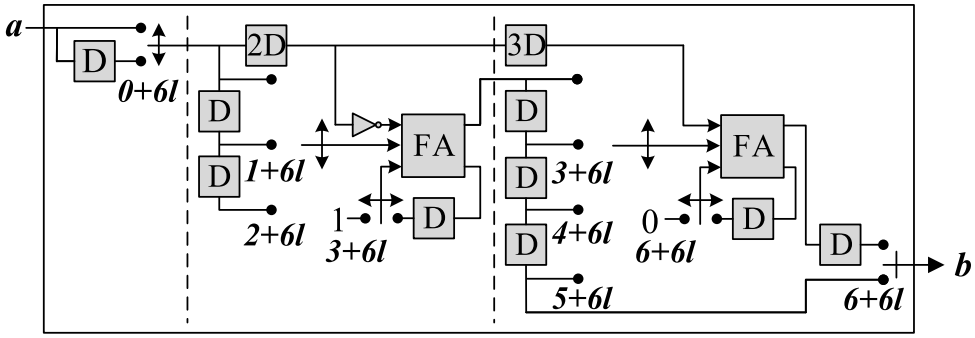

FIGURE 4. The diagram of the bit-serial CSD multiplier with six-bit constant coefficient "0 1 0 0 -1 0 1 0."

we propose a parallel-serial hybrid FFT architecture based on the CTA algorithm. According to the SNR, the architecture supports variable operation precision without changing the hardware design, achieving bit-level full-pipeline operations, and avoiding the worst-case design.

A. PARALLEL-SERIAL HYBRID FFT ARCHITECTURE

Based on the mixed-radix decomposition of the CTA, a large size FFT can be broken into smaller FFTs that can be combined arbitrarily with different architecture. Consequently, we propose a serial-parallel hybrid FFT architecture to improve the throughput of the whole FFT by high parallelism. Furthermore, the specific details are as follows.

As shown in Fig. 5, a N-point FFT is decomposed to N_1-point FFT with full-parallel architecture (pFFT in the figure) and a N_2-point SDF-based FFT with serial architecture (sFFT in the figure). In particular, the proposed N-point FFT processor can be divided into three stages. In *Stage I*, the N_1-parallel $\boldsymbol{x} = \left[x_0, x_1, \cdots, x_{N_1-1}\right]^T$ is as the input, and the corresponding output is $\dot{X} = \left[\dot{X}_0, \dot{X}_1, \cdots, \dot{X}_{N_1-1}\right]^T$ after being processed by the N_1-point FFT processor. And the *Stage II* consists of N_1 complex constant-coefficient multipliers (square multiplier shown in Fig. 5) for multiplication of the twiddle factors between the front-stage N_1-point FFT and the latter N_2-point FFT. In addition, a complex multiplier can be composed of three real bit-serial multipliers in practice. After the multiplications are performed, the parallel output $\dot{X}$ of the *Stage I* is converted to $\ddot{X} = \left[\ddot{X}_0, \ddot{X}_1, \cdots, \ddot{X}_{N_1-1}\right]^T$. In *Stage III*, the temporary output $\ddot{X}$ can be converted into $X = \left[X_0, X_1, \cdots, X_{N_1-1}\right]^T$ by N_1-parallel N_2-point SDF-based FFT. So far, the calculation of N-point FFT is completed. The above is the complete operation process of the N-point FFT processor with hyper-parallel architecture.

As shown in Fig. 5, the read-only memory (ROM) is used to store the twiddle factors. In particular, the twiddle factors for N-point FFT is stored in N-point *TW-ROM*, and the twiddle factors for N_2-point FFT are stored in N_2-point *TW-ROM*. The address generator module can generate the corresponding address signal to obtain the twiddle factor from the ROM. The control module generates control signals for each sub-module to finish the FFT operation.

In order to improve the throughput of the bit-serial design, the parallel architecture is applied into N_1-point FFT in *Stage I*. The high-parallel design ensures that internal twiddle factors are fixed and does not require ROM resources for storage. It also avoids complex data scheduling and achieves high-speed and low-latency operations. However, there is a sharp contradiction between the low hardware overhead and high parallelism. However, the conflict is well allevi-

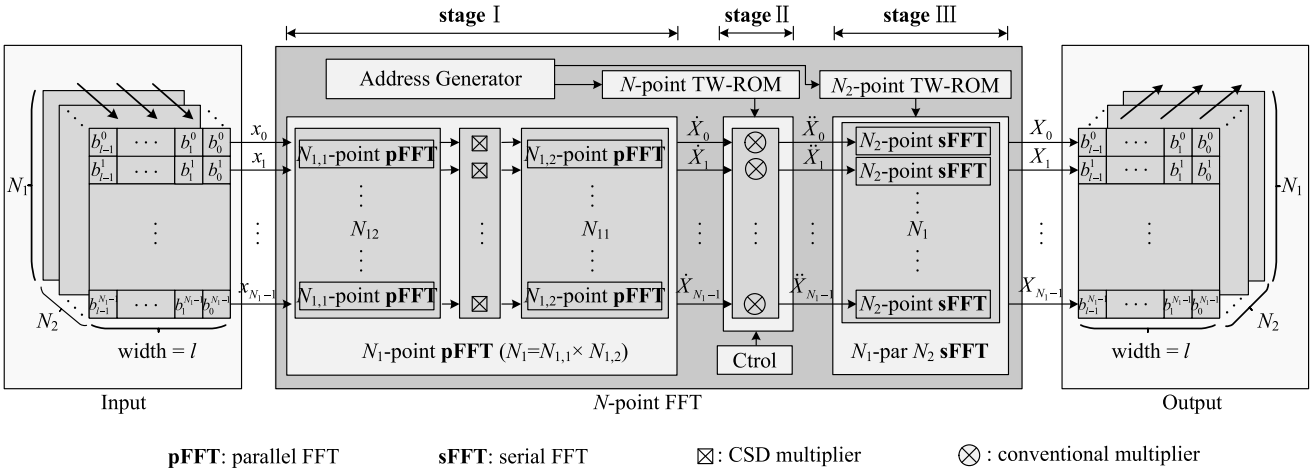

FIGURE 5. The block diagram for proposed serial-parallel FFT architecture based on bit-serial design.

TABLE 2. Resource comparison table of bit-serial Lyon multiplier and bit-serial CSD multiplier with constant coefficient.

	bit-serial Lyon multiplier	bit-serial CSD multiplier
D	$3 \times n - 1$	$2 \times n + n/2 + 1$
FA	$n - 1$	$n/2 - 1$
Mult2	$2 \times n$	$n - 1 + n/2 + 2$
And	n	0
Not	1	$n/4$

ated by bit-serial multipliers and adders in the proposed HPBS architecture. As a result, the proposed FFT architecture achieves a compromise of hardware overhead and throughput. Meanwhile, attributed to the bit-serial multiplier and adder, the proposed FFT processor can support variable arithmetic precision and efficiently exchange resources and performance.

The N_1-point parallel FFT in *Stage I* are N_1-parallel input and output, while the N_2-point SDF-based FFT in *Stage III* are single input and output. In order to make up for the mismatch in throughput, the N_2-point SDF-based FFT in *Stage III* is designed to be N_1-parallel. Thus the input and the output interfaces of the whole N-point FFT is also N_1-parallel. Moreover, the intermediate data in *Stage II* does not need to be temporarily stored, which avoids access conflict and complex data management modules.

In this article, a $N = 2048$ point FFT paradigm is designed and implemented to illustrate the proposed HPBS architecture. In particular, the $N = 2048$ point FFT processor can be decomposed into $N_1 = 64$ point parallel FFT and $N_2 = 32$ point FFT. And the specific details is shown below.

B. FULL-PARALLEL FFT

In the proposed N-point FFT architecture, only the front N_1-point FFT adopts a full-parallel design, where the input and output are N_1-parallel.

Like the decomposition of N-point FFT,the N_1-point FFT can be further decomposed into smaller $N_{1,1}$-point and $N_{1,2}$-point FFT based on the MR-CTA algorithm, where $N_1 = N_{1,1} \times N_{1,2}$. Likewise, the fully parallel N_1-point FFT operation process can also be divided into three parts. Here, the P_1 contains the $N_{1,2}$-parallel $N_{1,1}$-point FFT module, and the P_2 consists of $(N_{1,1} - 1)(N_{1,2} - 1) - 1$ complex multipliers (square multiplier shown in the Fig. 6). And the P_3 contains the $N_{1,1}$-parallel $N_{1,2}$-point FFT module. Due to the fully parallel structure, the circuit connection in the hardware is fixed, which avoids the conflict of data access and routing, and the twiddle factors required for the complex multiplication in the P_2 are fixed. Therefore, the multiplication in P_2 can be performed by the constant coefficients bit-serial multiplier. In addition, the N_1-point FFT does not require ROM resources and additional control circuits, which can reduce hardware resource overhead.

As shown in the Fig. 6, the input and output of the $N_1 = 64$ point FFT are 64-parallel respectively, where $N_{1,1} = N_{1,2} = 8$. In addition, Fig. 6 also shows the radix-8 full-parallel FFT can be further decomposed into a cascade of a radix-4 FFT and radix-2 FFT. According to the characteristics of twiddle factors, input data do not need to be rotated when the phase ϕ of the twiddle factor is 0. Since a half to the twiddle factors of the radix-8 is constant 1, the delay line is adopted to ensure that the parallel data can be aligned. Besides, the radix-4 FFT has only two twiddle factors $\{1, -j\}$. Thus the radix-4 FFT only needs to reverse and exchange the real and imaginary parts without consuming multipliers. The radix-4 FFT and radix-8 FFT have simple structures and have

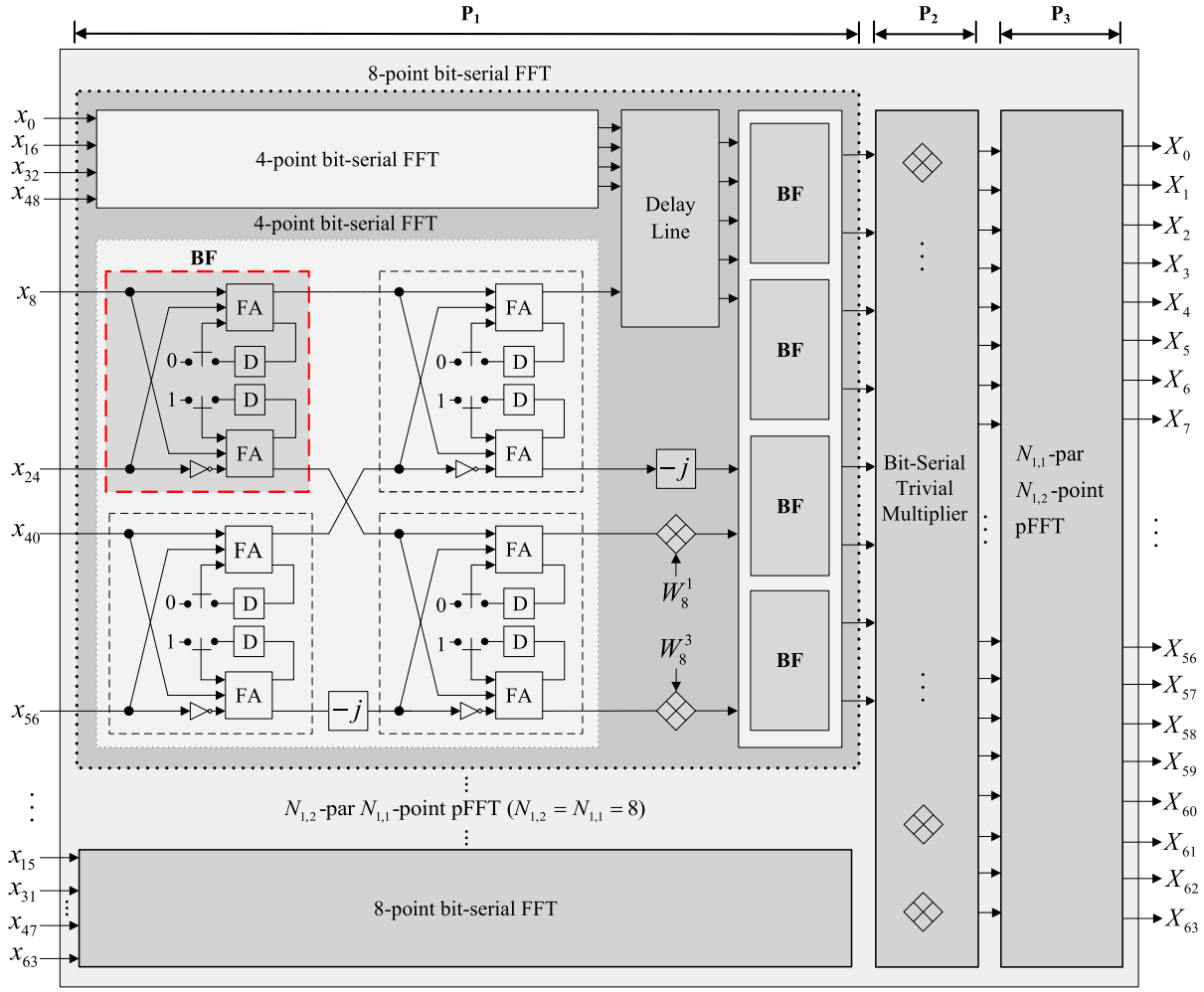

FIGURE 6. The diagram of the N_1-point FFT with full-parallel design.

less multiplication. Consequently, the proposed large FFT is decomposed into the mixed radix-4 and radix-8 FFT.

C. SDF-BASED FFT

As shown in the Fig. 7, a N-point SDF-based pipeline FFT contains $T = log_2(N)$ cascaded stages. Each stage consists of a radix-2 butterfly operation unit (BOU) based on bit-serial design, a first-in first-out (FIFO) and a complex multiplier for twiddle factor multiplication. The delay path in stages is performed by a FIFO with predetermined depth in the SDF architecture. And the depth L_t of the FIFO in t_{th}, $t \in \{1, 2, \cdots, T\}$ stage is associated with the corresponding delay 2^{t-1} and data width w, hence $L_t = 2^{t-1} \times w$. As the stage increases by one level, the depth of the FIFO will be doubled. And the total amount S of memory required can be given as

$$S = \sum_{t=0}^{T-1} L_t = w(N-1). \tag{30}$$

And in the delay feedback course, outputs from BOU are delayed to pair with the next coming inputs and then fed back to the BOU for next operation.

The detailed internal structure of the BOU is also shown at the bottom left of Fig. 7. In the first-level BOU, the first half data $\{x_i, i = 0, 1, \cdots, N/2-1\}$ is delayed by a FIFO with the depth $N/2$, then fed back to the BOU, finally performed the butterfly operation with the second half $x_{i+(N/2)}$. Moreover, there is a multiplexer to perform the output management in each BOU. Then the output of the first-level BOU is multiplied by the twiddle factor as the input of the second-level BOU. Similar to the first-level BOU, the input is delayed by $N/4$, then fed back to the second-level BOU. Besides, the operation process of subsequent BOUs can be deduced by analogy. Furthermore, the BOU can be bypassed to achieve variable-point FFT. In practice, the trivial multipliers and general multipliers are also bypassed when a former BOU is bypassed. The SDF-based FFT can support variable precision and power-of-two size combined with the bit-serial design.

There is a multiplier for the multiplication of twiddle factors between two adjacent stages. In particular, there is a trivial multiplier between every second stage as they only involve multiplication of $-j$ (where j is the imaginary unit). Thus the trivial multiplier only reverses the real and

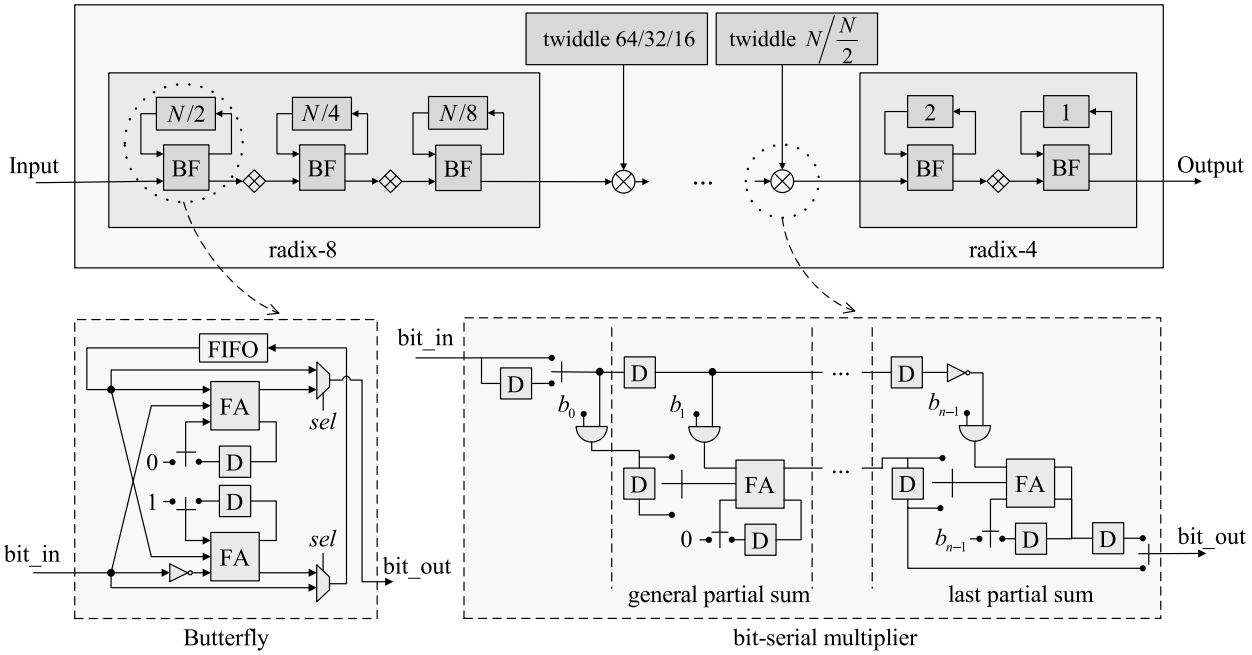

FIGURE 7. The diagram of the SDF-based bit-serial FFT.

imaginary components and negates the resulting imaginary part. In Fig. 7, the trivial multipliers are indicated with a rhombus, and the conventional multipliers are indicated with ⊗.

Based on the CTA algorithm, a large size FFT is decomposed into radix-8 and radix-4 FFT. Compared to the radix-2 counterpart, the radix-8 and radix-4 FFT multipliers can be greatly reduced. In this article, a 32-point FFT paradigm can be decomposed into a cascade of a former radix-8 (including 3-level BOUs) FFT and the latter radix-4 (including 2-level BOUs) FFT. In addition, a complex bit-serial multiplier is required to perform the multiplication of the twiddle factors of the above two stages. The twiddle factors are fixed and pre-stored in a ROM. According to the address generator, the corresponding twiddle factors are read and updated by the control module.

In summary, the characteristic of the SDF architecture is to multiplex the BOU and multiplier highly and fold multiple parallel FFT into a single pipeline input and output, and finally realize the data scheduling through FIFO. In addition, the input of the SDF-based FFT is in natural order, and the output is in bit-reverse order [32].

D. SEQUENCE ANALYSIS

As shown in Fig. 5, a N-point FFT can be divided into three stages: *Stage I*, *Stage II*, and *Stage III*, thus the time delay can be considered from three parts. In the proposed $N = 2048$ point FFT, the bit width of twiddle factors and data are set to M bits and K bits, respectively. The time delay in *Stage I* is $3M$ clock cycles (CC) since 3 complex multipliers are required in a data path of $N_1 = 64$ point FFT. The *Stage II* only contains a complex multiplier, thus the time delay is M CC. In the *Stage III*, every FIFO in SDF-based FFT leads to a different time delay that is up to the depth of FIFOs. In particular, a $N_2 = 32$ point SDF-based FFT has two complex multipliers, thus the total time delay is equivalent to the amount of memory in (30) as $2M + 31K$. Consequently, the sequence diagram of a $N = 2048$ point FFT can be illustrated as Fig. 8.

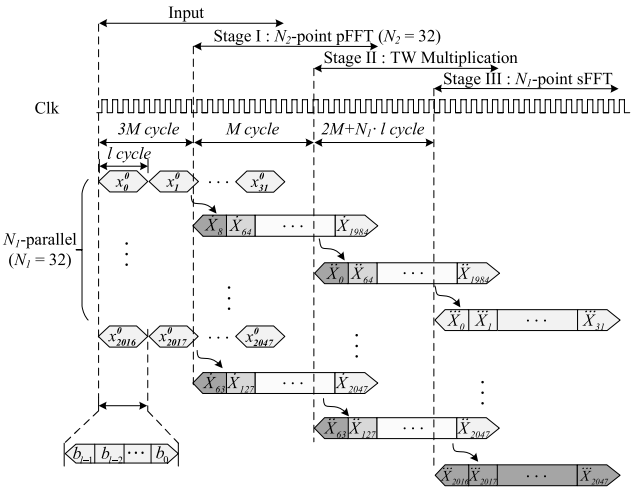

FIGURE 8. The sequence diagram of the proposed serial-parallel bit-serial FFT.

VI. PERFORMANCE ANALYSIS AND HARDWARE IMPLEMENTATION

A. PERFORMANCE ANALYSIS

This subsection mainly shows the performance of the MIMO-OFDM system based on the proposed bit-serial FFT

TABLE 3. The implementation comparison of FFTs on application-specific integrated circuit (ASIC).

	2018 TCAS-I [5]	2019 TCAS-I [7]	2017 VLSI [9]	2016 TCAS-II [14]	2018 TCAS-I [16]	2015 VLSI [20]	2017 TCAS-I [21]	This Work
FFT Size	2048	2048	2048	1024	1024	2048	2048	2048 (64-2048)[1]
Architecture	SDF	M.P.[3]	Memory-based	SDC	MDC	SDF	SDF	HPBS
Variable Size	Yes	Yes	Yes	No	No	Yes	Yes	Yes
Word Length (bit)	14	12	16	16	16	12	16	12 (8-16)[2]
Process (nm)	40	28	55	55	55	90	90	55
Data Path	1	2	2	1	4	1	1	64
Frequency (MHz)	500	300	122.88	200	320	40	188.67	502
Latency (CC)	2048	1200	1878	1024	265	2056	2187	840
Delay (μs)	4.096	4	15.28	5.12	0.83	51.4	11.59	1.673
Gata Count (k)	380	181	136	134	-	204.7	396	222
Area (mm^2)	0.36	0.08	0.615	0.15	0.212	0.783	0.87	0.689
Normalized Area (mm^2)	0.681	0.309	0.615	0.15	0.212	0.292	0.325	0.689
Throughput Rate (Gbps)	7.0	7.2	3.932	3.2	1.28	0.48	3.019	32.128
Normalized Efficiency (Gbps/mm^2)	10.279	23.301	6.394	21.33	6.038	1.644	9.289	46.630

[1] The proposed HPBS FFT size is power-of-two (at least 64-point).
[2] The quantization bit width supports dynamic variable, ranging from 8 bits to 16 bits. The typical quantization bit width is 12 bits, at this time, the BER performance loss is within 0.1 dB compared with the full precision.
[3] The modified pipelined architecture.

by the simulation of the bit error rate (BER) and normalized mean squared error (NMSE).

Consider the following point-to-point MIMO-OFDM communication system, where the number of antennas at the transmitter and receiver are both set to 64, hence $N_t = N_r = 64$. The number N_s of data streams is 4, hence $N_s = 4$. The data is applied with 16-QAM (quadrature amplitude modulation) constellation mapping, and the FFT size N is 256, hence $N = 256$. The channel model in the simulation is based on the CDL-B model in 3GPP [33]. Moreover, the carrier frequency is 3.5GHz. In order to illustrate the performance of the limited-precision bit-serial FFT and avoid the influence of the other factors, it can be assumed that the synchronization and channel estimation of the MIMO-OFDM system is perfect. Besides, the MMSE is applied to channel equalization.

The principle of the whole MIMO-OFDM system is as follows. The transmitter modulates the 16-QAM constellation-mapped signal to different sub-carriers through IFFT. Then the modulated multi-carrier signals are passed through the AWGN channel, then reach the receiver. Next, the received multi-carrier signals are demodulated by FFT into multiple sub-carrier signals for decoding.

1) THE NMSE OF MIMO-OFDM SYSTEM

The normalized MSE (NMSE) of a MIIMO-OFDM system can be defined as

$$NMSE = \frac{\|x - \tilde{x}\|^2}{\|x\|^2}, \tag{31}$$

which can indicate the performance of a whole system.

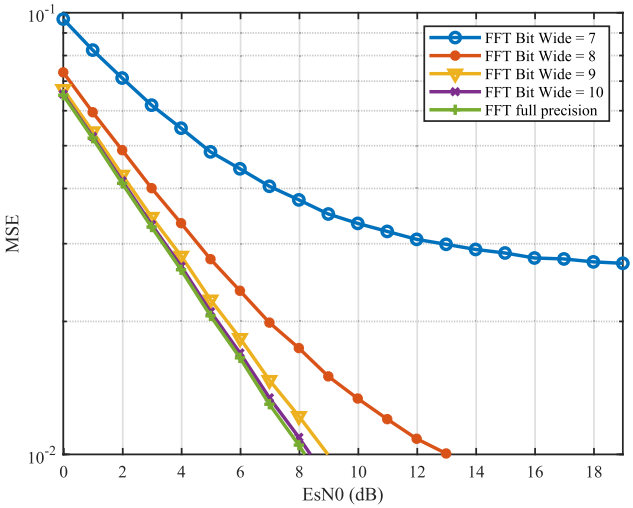

FIGURE 9. The NMSE simulation of the MIMO-OFDM system with a 2048-point bit-serial FFT.

As shown in Fig. 9, the horizontal axis $EsN0$ represents the SNR, and the vertical axis represents the NMSE of the whole MIMO-OFDM system based on the proposed FFT with different quantization bit width from 7 bits to 10 bits. Likewise, the NMSE in the Fig. 9 is obtained by averaging multiple data streams. The curves in different colors indicate the different bit-width w. The input bit width needs to be more than 7 bits; thus, the whole system can work normally. In order to facilitate the comparison of the performance loss under different quantization bit widths, the average NMSE of the MIMO-OFDM system under the full-precision FFT module is also supplemented in Fig. 9. As the quantization

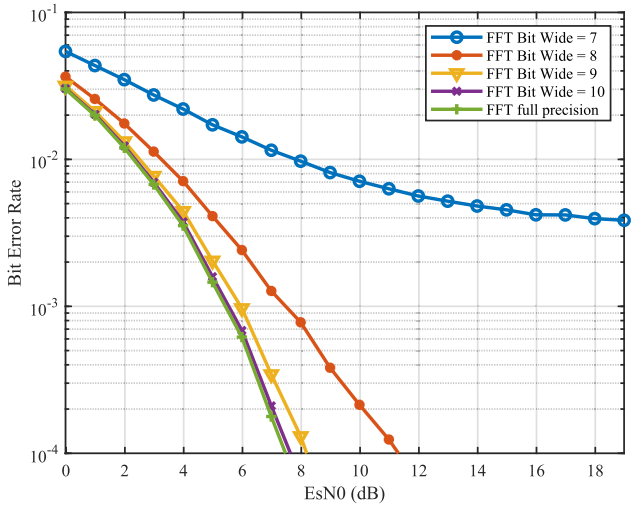

FIGURE 10. The BER simulation of the MIMO-OFDM system with a 2048-point bit-serial FFT.

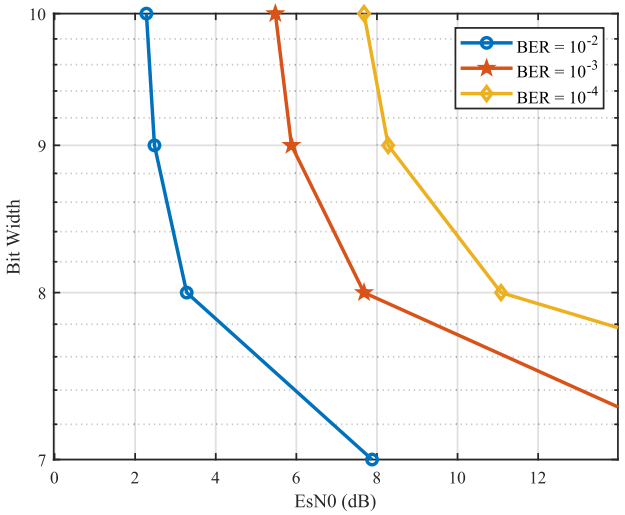

FIGURE 11. The tradeoff between quantization bit width and input SNR.

bit width increases, the operation precision can be improved while truncation error reduces. Thus, the performance is gradually approaching full precision.

2) THE BER SIMULATION

According to previous studies, the FFT modules with higher accuracy will lead to better BER with the presence of noisy channel. Previous studies have shown that in the 1024-point FFT fixed-point operation, when the fixed-point FFT internal word length is 8 bits, the signal-to-quantization noise ratio (SQNR) is about 40 $\sim$ 50db, which can meet the communication needs. As shown in Fig. 10, the horizontal axis is SNR $EsN0$, and the vertical axis represents the BER of the MIMO-OFDM system based on the proposed FFT with different quantization bit width from 7 bit to 10 bit. Since multiple data streams are considered in simulation, the BER in the Fig. 10 is obtained by averaging multiple data streams. In addition, the curves in different colors indicate the different bit-width w. In order to facilitate the comparison of the performance loss under different quantization bit widths, the average BER of the MIMO-OFDM system under the full-precision FFT module is also supplemented in Fig. 10. From the figure, if the bit width is more than 8 bits, the system performance is closer to the full-precision processing. As the bit width w increases, the precision of the data can be improved while truncation error reduces. Otherwise, the BER curves are gradually flat, which indicates the whole MIMO-OFDM system can not work properly. Therefore, the proposed bit-serial FFT can support variable precision and slightly degrade performance compared with the full-precision FFT.

3) THE FLEXIBLE EXCHANGE OF RESOURCES AND PERFORMANCE

Based on the proposed HPBS architecture, Fig. 11 illustrates the tradeoff between quantization bit width and input SNR. In order to achieve the identical BER, in the case of different input SNR, the BER performance of the whole MIMO-OFDM system can be maintained by dynamically changing the quantization bit width. As shown in Fig. 11, when the SNR of the received signal is 8dB and the quantization bit width is 7 bits, the system BER is 10^{-2}. In order to maintain the identical BER performance, the quantization bit width of the input signal needs to be increased to at least 8bit when the input SNR is about 3dB. In fact, the high-performance requirement for some applications is not always necessary. On the premise of satisfying application requirements, performance could be slightly sacrificed to reduce system resource and power consumption by relaxing the processing precision.

B. HARDWARE EFFICIENCY

We design and implement a 2048-point bit-serial FFT processor based on the above hybrid architecture. According to the BER performance simulation based on the proposed HPBS FFT, the equivalent average quantization bit width with 12 bits can ensure the identical BER performance as the traditional FFT designs [20], [21]. At this time, the BER performance loss of the whole MIMO-OFDM system is within 0.1dB compared with the full precision. In order to demonstrate the performance, the FFT processor was implemented in the 55 nm process in the *Design Compiler* to evaluate the hardware efficiency. Furthermore, the implementation results are shown in Table 3. Due to the bit-serial architecture, the processing delay of a critical path can be reduced largely. Thus the proposed FFT processor can be at a maximal working frequency of 502MHz. This innovative design work is competitive as compared to current state-of-the-art works, especially in terms of hardware efficiency. Compared with the traditional bit-parallel design, the normalized hardware efficiency of the proposed is more than 2X that of the conventional FFT designs.

VII. CONCLUSION

This article proposes an optimization strategy based on the compromise of processing and area from the performance perspective. In particular, the strategy achieves a tradeoff among performance, area, and throughput, then realizes the flexible exchange of resources and performance based on the bit-serial design. In particular, the proposed HPBS FFT processor can achieve variable precision by adjusting the number of processing cycles and matching the SNR of the input signal to satisfy the requirement of the MIMO-OFDM system. In addition, it does not require redundant storage resources and supports the full-pipeline operation. The design avoids the worst case, which saves the hardware resources and energy.

The HPBS FFT processor proposed in this article solves the following pain points:

1) The traditional hardware design cannot be adjusted in real time according to the performance of the communication system. The HPBS FFT architecture supports variable computing bit width to realize the flexible exchange of hardware resources and system performance. Different adjustments can be made in different communication scenarios to ensure the requirements of the communication system while saving hardware resources.

2) The traditional FFT architecture has low computational efficiency, while the normalized hardware efficiency of the HBPS FFT processor proposed in this article is high, at least twice that of the traditional FFT design.

Therefore, based on the above advantages, the proposed optimization strategy can be expected to become one of the significant directions for future hardware optimization.

REFERENCES

[1] J. Navarro-Ortiz, P. Romero-Diaz, S. Sendra, P. Ameigeiras, J. J. Ramos-Munoz, and J. M. Lopez-Soler, "A survey on 5G usage scenarios and traffic models," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 905–929, 2nd Quart., 2020.

[2] C. Zhang, Y.-H. Huang, F. Sheikh, and Z. Wang, "Advanced baseband processing algorithms, circuits, and implementations for 5G communication," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 7, no. 4, pp. 477–490, Dec. 2017.

[3] J. Gao, K. O. O. Agyekum, E. B. Sifah, K. N. Acheampong, Q. Xia, X. Du, M. Guizani, and H. Xia, "A blockchain-SDN-enabled Internet of Vehicles environment for fog computing and 5G networks," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4278–4291, May 2020.

[4] A. Ahad, M. Tahir, and K. A. Yau, "5G-based smart healthcare network: Architecture, taxonomy, challenges and future research directions," *IEEE Access*, vol. 7, pp. 100747–100762, 2019.

[5] X.-Y. Shih, H.-R. Chou, and Y.-Q. Liu, "VLSI design and implementation of reconfigurable 46-mode combined-radix-based FFT hardware architecture for 3GPP-LTE applications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 1, pp. 118–129, Jan. 2018.

[6] P. Aggarwal and V. A. Bohara, "A nonlinear downlink multiuser MIMO-OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 6, no. 3, pp. 414–417, Jun. 2017.

[7] M. Mahdavi, O. Edfors, V. Öwall, and L. Liu, "A low latency FFT/IFFT architecture for massive MIMO systems utilizing OFDM guard bands," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 7, pp. 2763–2774, Jul. 2019.

[8] M. Garrido, "Evolution of the performance of pipelined FFT architectures through the years," in *Proc. 35th Conf. Design Circuits Integr. Syst. (DCIS)*, Nov. 2020, pp. 1–6.

[9] K.-F. Xia, B. Wu, T. Xiong, and T.-C. Ye, "A memory-based FFT processor design with generalized efficient conflict-free address schemes," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 6, pp. 1919–1929, Jun. 2017.

[10] Y. Tian, Y. Hei, Z. Liu, Q. Shen, Z. Di, and T. Chen, "A modified signal flow graph and corresponding conflict-free strategy for memory-based FFT processor design," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 66, no. 1, pp. 106–110, Jan. 2019.

[11] M. Garrido, "A survey on pipelined FFT hardware architectures," *J. Signal Process. Syst.*, vol. 94, no. 11, pp. 1345–1364, Jul. 2021.

[12] M. Garrido, J. Grajal, M. A. Sanchez, and O. Gustafsson, "Pipelined radix-2^k feedforward fft architectures," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 1, pp. 23–32, Jan. 2013.

[13] M. Garrido, K. Möller, and M. Kumm, "World's fastest FFT architectures: Breaking the barrier of 100 GS/s," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 4, pp. 1507–1516, Apr. 2019.

[14] M. Garrido, S.-J. Huang, S.-G. Chen, and O. Gustafsson, "The serial commutator FFT," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 63, no. 10, pp. 974–978, Oct. 2016.

[15] M. Garrido, N. K. Unnikrishnan, and K. K. Parhi, "A serial commutator fast Fourier transform architecture for real-valued signals," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 65, no. 11, pp. 1693–1697, Nov. 2018.

[16] M. Garrido, S.-J. Huang, and S.-G. Chen, "Feedforward FFT hardware architectures based on rotator allocation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 2, pp. 581–592, Feb. 2018.

[17] A. X. Glittas, M. Sellathurai, and G. Lakshminarayanan, "A normal I/O order radix-2 FFT architecture to process twin data streams for MIMO," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 6, pp. 2402–2406, Jun. 2016.

[18] M. Garrido and P. Paz, "Optimum MDC FFT hardware architectures in terms of delays and multiplexers," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 68, no. 3, pp. 1003–1007, Mar. 2021.

[19] C.-H. Yang, T.-H. Yu, and D. Markovic, "Power and area minimization of reconfigurable FFT processors: A 3GPP-LTE example," *IEEE J. Solid-State Circuits*, vol. 47, no. 3, pp. 757–768, Mar. 2012.

[20] C. Yu and M.-H. Yen, "Area-efficient 128-to 2048/1536-point pipeline FFT processor for LTE and mobile WiMAX systems," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 9, pp. 1793–1800, Sep. 2015.

[21] X.-Y. Shih, Y.-Q. Liu, and H.-R. Chou, "48-mode reconfigurable design of SDF FFT hardware architecture using radix-3^2 and radix-2^3 design approaches," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 64, no. 6, pp. 1456–1467, Jun. 2017.

[22] X.-Y. Shih, H.-R. Chou, and Y.-Q. Liu, "Design and implementation of flexible and reconfigurable SDF-based FFT chip architecture with changeable-radix processing elements," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 11, pp. 3942–3955, Nov. 2018.

[23] J. Wang, C. Xiong, K. Zhang, and J. Wei, "A mixed-decimation MDF architecture for radix-2^k parallel FFT," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 1, pp. 67–78, Jan. 2016.

[24] S.-N. Tang, C.-H. Liao, and T.-Y. Chang, "An area- and energy-efficient multimode FFT processor for WPAN/WLAN/WMAN systems," *IEEE J. Solid-State Circuits*, vol. 47, no. 6, pp. 1419–1435, Jun. 2012.

[25] S.-G. Chen, S.-J. Huang, M. Garrido, and S.-J. Jou, "Continuous-flow parallel bit-reversal circuit for MDF and MDC FFT architectures," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 10, pp. 2869–2877, Oct. 2014.

[26] Y. Lu, T. J. Kazmierski, and L. Liu, "A bit-serial variable-accuracy FFT processor for energy-harvesting systems," in *Proc. IEEE Asia Pacific Conf. Circuits Syst. (APCCAS)*, Oct. 2018, pp. 299–304.

[27] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Comput.*, vol. 19, no. 90, pp. 297–301, 1965.

[28] L. Pang, A. Li, Y. Zhou, C. Yang, Y. Xie, and H. Chen, "Word length optimization method for radix-2^k fixed-point pipeline FFT processors," in *Proc. IEEE Int. Conf. Signal, Inf. Data Process. (ICSIDP)*, Dec. 2019, pp. 1–4.

[29] K. K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation*. New York, NY, USA: Wiley, 1999.

[30] C.-F. Hsiao, Y. Chen, and C.-Y. Lee, "A generalized mixed-radix algorithm for memory-based FFT processors," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 1, pp. 26–30, Jan. 2010.

[31] D. Pandey and H. Leib, "A tensor framework for multi-linear complex MMSE estimation," *IEEE Open J. Signal Process.*, vol. 2, pp. 336–358, 2021.
[32] C. Ingemarsson, P. Källström, F. Qureshi, and O. Gustafsson, "Efficient FPGA mapping of pipeline SDF FFT cores," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 9, pp. 2486–2497, Sep. 2017.
[33] *Evolved Universal Terrestrial Radio Access (E-UTRA); Further Advancements for E-UTRA Physical Layer Aspects (Release 9)*, document TS 36.814, V9.2.0, 3GPP, Mar. 2017.

TINGYONG WU (Member, IEEE) received the B.E., M.S., and Ph.D. degrees in communication systems from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 1998, 2001, and 2007, respectively. He is currently an Associate Professor with the National Key Laboratory of Wireless Communications, UESTC. His current research interests include signal processing in wireless communication, circuit-system designs, and artificial intelligence for communication.

YUXIN WANG (Student Member, IEEE) received the B.E. degree from the School of Information and Communication Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2022, where she is currently pursuing the M.Sc. degree in communications with the National Key Laboratory of Wireless Communications. Her current research interests include signal processing in wireless communication and circuit-system designs.

FUQIANG LI received the B.E. and M.Sc. degrees in communication systems from the University of Electronic Science and Technology of China (UESTC), Chengdu, China. He is currently a Senior Engineer with China Electronics Technology Group Corporation (CETC), 20th Institute, Xi'an, China. His current research interest includes communication networks.