# Fixed-Point Analysis and Parameter Selections of MSR-CORDIC with Applications to FFT Designs

Sang Yoon Park, *Member, IEEE* and Ya Jun Yu, *Senior Member, IEEE*

## Abstract

Mixed-scaling-rotation (MSR) coordinate rotation digital computer (CORDIC) is an attractive approach to synthesizing complex rotators. This paper presents the fixed-point error analysis and parameter selections of MSR-CORDIC with applications to the fast Fourier transform (FFT). First, the fixed-point mean squared error of the MSR-CORDIC is analyzed by considering both the angle approximation error and signal round-off error incurred in the finite precision arithmetic. The signal to quantization noise ratio (SQNR) of the output of the FFT synthesized using MSR-CORDIC is thereafter estimated. Based on these analyses, two different parameter selection algorithms of MSR-CORDIC are proposed for general and dedicated MSR-CORDIC structures. The proposed algorithms minimize the number of adders and word-length when the SQNR of the FFT output is constrained. Design examples show that the FFT designed by the proposed method exhibits a lower hardware complexity than existing methods.

## Index Terms

Fast Fourier transform (FFT), coordinate rotation digital computer (CORDIC), mixed-scaling-rotation (MSR)-CORDIC, fixed-point, error analysis.

## EDICS Category: DSP-FILT, DSP-QUAN

Sang Yoon Park is with the Institute for Infocomm Research, Agency for Science, Technology, and Research (A*STAR), 138632, Singapore (e-mail: sypark@i2r.a-star.edu.sg).

Ya Jun Yu is with the School of Electrical and Electronic Engineering, Nanyang Technological University, 639798, Singapore (e-mail: eleyuyj@pmail.ntu.edu.sg).

# I. Introduction

Discrete Fourier transform (DFT) is one of the most important algorithms in the digital signal processing systems, and thus there has been much research on its efficient calculation [1]–[4], approximation [5]–[10], and implementation [11]–[15]. The $N$-point DFT is defined as

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \qquad \text{for} \quad k = 0, 1, ..., N-1, \tag{1}$$

where $W_N = e^{-j(2\pi/N)}$. The direct calculation of (1) requires $O(N^2)$ complex multiplications. The radix-2 fast Fourier transform (FFT) algorithm can compute the DFT using only $O(N \log_2 N)$ operations [1], [2]. There are several variants such as radix-$2^p$ for $p = 2, 4, 8...$ or split-radix FFT for reducing the computational complexity further [1], [3]. Refer to [1]–[3] for descriptions on the FFT algorithms in detail.

The complex multiplication creates a bottleneck in the computation of FFT. Thus, its efficient design has been one of the major issues in FFT related literatures. All the coefficients (called twiddle factors) of the complex multipliers in an $N$-point FFT are represented in the form of $e^{j\theta}$, where $\theta = 2\pi k/N$ for an integer $k$. Let $x$ and $y$ be complex numbers. Then, the twiddle factor multiplication $y = e^{j\theta}x$ can be viewed as a complex rotation of a $2 \times 1$ vector as

$$y = \begin{bmatrix} 1 & j \end{bmatrix} \mathbf{R}(\theta) \begin{bmatrix} x_{re} \\ x_{im} \end{bmatrix} \tag{2}$$

where $x = x_{re} + j \cdot x_{im}$ with real $x_{re}$ and $x_{im}$, and

$$\mathbf{R}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \tag{3}$$

The direct calculation of (2) requires 4 real multiplications and 2 additions. The rotation (2) can also be expressed using 3 real multiplications and 3 additions as [9], [10]

$$\begin{aligned} y_{re} &= (\cos\theta - \sin\theta)x_{re} + \sin\theta(x_{re} - x_{im}) \\ y_{im} &= (\cos\theta + \sin\theta)x_{im} + \sin\theta(x_{re} - x_{im}) \end{aligned} \tag{4}$$

where $y = y_{re} + j \cdot y_{im}$. Another strategy is to decompose (3) into well-known lifting steps as [6]–[10]

$$\mathbf{R}(\theta) = \begin{bmatrix} 1 & \tan\frac{\theta}{2} \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\sin\theta & 1 \end{bmatrix} \begin{bmatrix} 1 & -\tan\frac{\theta}{2} \\ 0 & -1 \end{bmatrix}, \tag{5}$$

where 3 real multiplications and 3 additions are required as well. Since multiplications require higher computational complexity than additions in general, the representation of (4) or (5) is more efficient than that of (2).

In a practical implementation, the coefficients need to be quantized into finite precision. A direct method is to approximate the coefficients in (3), (4), or (5) using finite precision representations. Alternatively, the coordinate rotation digital computer (CORDIC) algorithm is applied for the approximation of (3) [16], [17]. The CORDIC algorithm is more hardware-friendly than the multiplication and accumulation (MAC) unit since it can be implemented by pipelined structures of submodules using only shift and add operations [18]. Many published articles applied the CORDIC algorithm to the FFT [5], [9]–[13], [18].

In FFT processor designs, the twiddle factors are known in advance. Modified CORDIC algorithms were proposed to improve the latency, accuracy, and complexity of the computation [19]–[22]. Mixed-scaling-rotation (MSR)-CORDIC algorithm proposed by Lin et al. [22] approximates the vector rotation with the smallest approximation error among existing CORDIC algorithms under the same hardware complexity. The conventional optimization methods based on Viterbi and greedy algorithms minimize the approximation error of single MSR-CORDIC processor in the course of parameter determination [21]. However, if the MSR-CORDIC is used as complex multiplier in FFT, all the MSR-CORDICs in the $N$-point FFT need to be optimized jointly so that the total mean squared error (MSE) of the FFT output is minimized. Furthermore, for the fixed point optimization of FFT, more practical metric is required to reflect the round-off and scaling errors as well as approximation errors. Keeping these in view, in this paper, we present a parameter optimization algorithm specially for the optimization of complex rotators when they are implemented using MSR-CORDIC. We also derive error analysis equations which are used as metric during optimization. The contributions of this paper are as follows:

First, the output MSE is estimated to achieve the optimal design of FFT with MSR-CORDIC. In [22], the MSE of approximation error of MSR-CORDIC has been analyzed. In this paper, the round-off error of MSR-CORDIC are also analyzed in terms of MSE. Our error model provides more accurate output error of MSR-CORDIC considering both of these two error terms.

Second, the MSE and the corresponding signal to quantization noise ratio (SQNR) of output of radix-2 decimation-in-time (DIT) FFT algorithm with MSR-CORDIC are derived. We prove that our error analysis closely matches actual simulation results by design examples. The analysis can be applied to FFTs with different radix or split-radix.

Third, a parameter optimization algorithm is proposed when twiddle factor multipliers of FFT are implemented using generalized MSR-CORDIC. In this application, the same MSR-CORDIC module is shared during the computation of a symbol FFT data to reduce the hardware cost by re-using the resources and to enhance the regularity of the overall structure. The reference implementation using generalized MSR-CORDIC is shown in [14]. The generalized MSR-CORDIC is implemented as a uniform structure for resource sharing, and designed to minimize the implementation cost while criteria such as the accuracy of FFT are satisfied.

Fourth, a parameter optimization algorithm is proposed when dedicated CORDICs are employed for the FFT. Each twiddle factor multiplication is designed to have its own dedicated CORDIC circuit for parallel processing. Recently, the design with dedicated multiplier has an increasing interest with the development of very large scale integrated (VLSI) technology in the applications such as digital filters [23]–[25], discrete cosine transform (DCT) [8], [26], and FFT [7]–[9] to satisfy the demand for high throughput rate. In the dedicated circuit, the number of adders is usually used as the measure that estimates the implementation cost since the shifter is hard-wired. The cost also depends on the word-length selection for intermediate registers in FFT. The design method which considers both the number of nonzero digits and word-length is proposed.

The rest of this paper is organized as follows. In Section II, the MSR-CORDIC is reviewed. In Section III, the fixed point errors of MSR-CORDIC algorithm are analyzed. Based on the analysis, the output MSE and the corresponding SQNR of radix-2 DIT FFT are estimated. Section IV presents the parameter determination algorithm to minimize the adder cost and word-length of FFT for a given accuracy constraint where the complex multipliers are implemented using generalized MSR-CORDIC. An alternative algorithm is proposed for FFT with dedicated MSR-CORDICs in Section V. A brief conclusion is given in Section VI.

## II. Review of MSR-CORDIC Algorithm

The original CORDIC algorithm proposed in [16], [17] approximates the real rotation matrix (3) as

$$[\mathbf{R}(\theta)]_Q = \frac{1}{S}\prod_{k=0}^{K-1}\begin{bmatrix} 1 & -\mu(k)2^{-k} \\ \mu(k)2^{-k} & 1 \end{bmatrix} \tag{6}$$

where $[\cdot]_Q$ is a nonlinear approximation operator, $\mu(k) \in \{-1, 1\}$, $S = \prod_{k=0}^{K-1}\sqrt{1+2^{-2k}}$, and $K$ is the number of rotations. The original CORDIC algorithm suffers from some drawbacks such as low approximation accuracy, long latency, and computational overhead due to the scaling factor $1/S$. The MSR-CORDIC overcomes these drawbacks with a few modifications of the original CORDIC equations as follows:

First, the MSR-CORDIC enhances the approximation accuracy by replacing $\mathbf{R}(\theta)$ with cascades of sub-rotations which include more flexible sum of signed powers-of-two (SPT) coefficients than (6), i.e.,

$$[\mathbf{R}(\theta)]_Q = \frac{1}{S}\prod_{k=0}^{K-1}\begin{bmatrix}\sum_{i=0}^{I(k)-1}\eta_i(k)2^{-p_i(k)} & -\sum_{j=0}^{J(k)-1}\mu_j(k)2^{-q_j(k)} \\ \sum_{j=0}^{J(k)-1}\mu_j(k)2^{-q_j(k)} & \sum_{i=0}^{I(k)-1}\eta_i(k)2^{-p_i(k)}\end{bmatrix} \tag{7}$$

where

$\eta_i(k), \mu_j(k) \in \{-1, 1\}$,

$0 \le p_i(k), q_j(k) \le B^c$ for the coefficient word-length $B^c$,

$$S = \prod_{k=0}^{K-1}\sqrt{\left(\sum_{i=0}^{I(k)-1}2^{-p_i(k)}\right)^2 + \left(\sum_{j=0}^{J(k)-1}2^{-q_j(k)}\right)^2}, \tag{8}$$

and $I(k)$ and $J(k)$ are nonnegative integers which represent the number of SPT terms. The approximated angle is given by

$$[\theta]_Q = \sum_{k=0}^{K-1}\tan^{-1}\left(\frac{\sum_{j=0}^{J(k)-1}\mu_j(k)2^{-q_j(k)}}{\sum_{i=0}^{I(k)-1}\eta_i(k)2^{-p_i(k)}}\right). \tag{9}$$

An appropriate search algorithm should be employed to obtain $[\theta]_Q$, and the approximation accuracy depends on how well the (sub)optimal parameters are found so as to minimize the residual angle, $|\theta - [\theta]_Q|$.

Second, the MSR-CORDIC attempts to reduce the latency by limiting the number of iterations $K$ ($K$ is usually set to be the internal register word-length in original CORDIC.). Even though the MSR-CORDIC has a smaller $K$, the approximation error is not more than that of original CORDIC due to more flexible representation of sub-rotations.

Third, the MSR-CORDIC obviates the postprocessing which multiplies the output with $1/S$. It can be attained by performing a search algorithm such that $|S - 1|$ as well as $|\theta - [\theta]_Q|$ is minimized. In that case, (7) is rewritten as

$$[\mathbf{R}(\theta)]_Q = \prod_{k=0}^{K-1}\begin{bmatrix}\sum_{i=0}^{I(k)-1}\eta_i(k)2^{-p_i(k)} & -\sum_{j=0}^{J(k)-1}\mu_j(k)2^{-q_j(k)} \\ \sum_{j=0}^{J(k)-1}\mu_j(k)2^{-q_j(k)} & \sum_{i=0}^{I(k)-1}\eta_i(k)2^{-p_i(k)}\end{bmatrix}. \tag{10}$$

In this paper, $[\mathbf{R}(\theta)]_Q$ is regarded as (10) instead of (7). Note that the number of additions required for the $2 \times 1$ vector rotation by MSR-CORDIC is represented as

$$A^c = \sum_{k=0}^{K-1}A^c(k) = \sum_{k=0}^{K-1}2\big(I(k) + J(k) - 1\big) \tag{11}$$

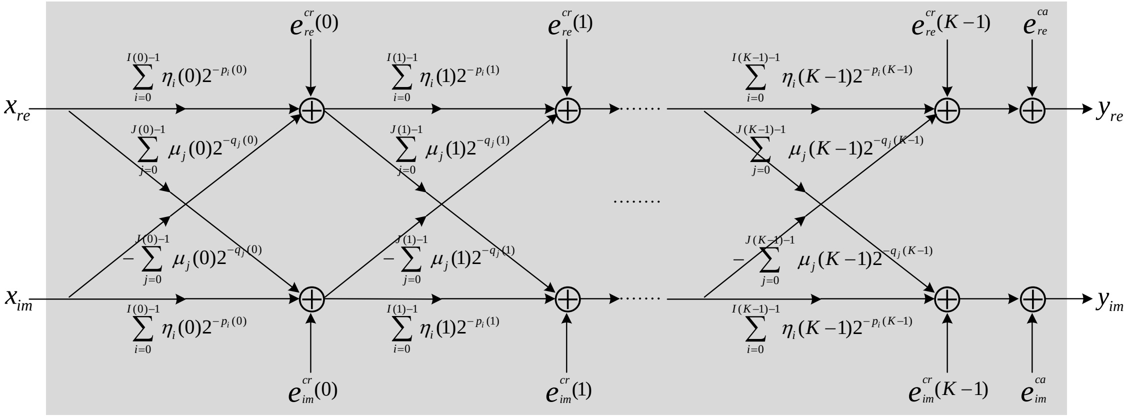

$e^{cr}_{re}(0)$
$e^{cr}_{re}(1)$
$e^{cr}_{re}(K-1)$
$e^{ca}_{re}$
$\sum_{i=0}^{I(0)-1} \eta_i(0)2^{-p_i(0)}$
$\sum_{i=0}^{I(1)-1} \eta_i(1)2^{-p_i(1)}$
$\sum_{i=0}^{I(K-1)-1} \eta_i(K-1)2^{-p_i(K-1)}$
$x_{re}$
$y_{re}$
$\sum_{j=0}^{J(0)-1} \mu_j(0)2^{-q_j(0)}$
$\sum_{j=0}^{J(1)-1} \mu_j(1)2^{-q_j(1)}$
$\sum_{j=0}^{J(K-1)-1} \mu_j(K-1)2^{-q_j(K-1)}$
$-\sum_{j=0}^{J(0)-1} \mu_j(0)2^{-q_j(0)}$
$-\sum_{j=0}^{J(1)-1} \mu_j(1)2^{-q_j(1)}$
$-\sum_{j=0}^{J(K-1)-1} \mu_j(K-1)2^{-q_j(K-1)}$
$x_{im}$
$y_{im}$
$\sum_{i=0}^{I(0)-1} \eta_i(0)2^{-p_i(0)}$
$\sum_{i=0}^{I(1)-1} \eta_i(1)2^{-p_i(1)}$
$\sum_{i=0}^{I(K-1)-1} \eta_i(K-1)2^{-p_i(K-1)}$
$e^{cr}_{im}(0)$
$e^{cr}_{im}(1)$
$e^{cr}_{im}(K-1)$
$e^{ca}_{im}$

Fig. 1. Error model for the MSR-CORDIC algorithm. $\mathbf{e}^{ca} = [e^{ca}_{re}, e^{ca}_{im}]^T$ is the approximation error, which is assumed to be added in at the last sub-rotation. $\mathbf{e}^{cr}(k) = [e^{cr}_{re}(k), e^{cr}_{im}(k)]^T$ is the round-off error introduced at the $k$-th MSR-CORDIC sub-rotation.

where $A^c(k)$ is the number of additions at the $k$-th sub-rotation. It should also be noted that the approximation ability of the MSR-CORDIC depends on $A^c$ and the coefficient word-length $B^c$. Specifically, as $A^c$ or $B^c$ increases, the approximation error decreases.

## III. Error Analysis of MSR-CORDIC and FFT

### A. *Error Analysis of MSR-CORDIC*

In this section, the output MSE of radix-2 DIT FFT algorithm where complex rotators are implemented as MSR-CORDIC is estimated. For this, the output MSE of MSR-CORDIC is first derived, and it is applied to the error analysis of FFT thereafter. The error sources and models are illustrated with the cascade structure of MSR-CORDIC in Fig. 1. There are two error sources in the fixed-point implementation of MSR-CORIC. One is the approximation error denoted by $\mathbf{e}^{ca} = [e^{ca}_{re}, e^{ca}_{im}]^T$ in Fig. 1, which results from the discrepancy between the ideal matrix $\mathbf{R}(\theta)$ and the approximated matrix $[\mathbf{R}(\theta)]_Q$. Superscript '$ca$' is the abbreviation for 'CORDIC approximation', and similar abbreviations are used for the other error sources of CORDIC and FFT. The intermediate data in the CORDIC computation is usually stored in registers with limited word-length to avoid an increase in the hardware size. Thus, the lower bits of the data are truncated for maintaining the word-length after the multiplication (shift) operation is performed. It introduces the round-off error $\mathbf{e}^{cr}(k) = [e^{cr}_{re}(k)\ e^{cr}_{im}(k)]^T$ for $0 \leq k < K$ at the $k$-th sub-rotation, which is the other error source of the MSR-CORDIC.

The output MSE of the approximation error of MSR-CORDIC was derived in [22]. It is rewritten using notations of this paper as follows:

$$E\{(\mathbf{e}^{ca})^T\mathbf{e}^{ca}\} \simeq E\{|\mathbf{x}|^2\}\big(\delta^2 + (1-S)^2\big) \tag{12}$$

where $\mathbf{x}$ is the $2 \times 1$ input vector of rotator and $\delta = \theta - [\theta]_Q$. It can be seen that the approximation error reduces as $\delta$ becomes smaller and $S$ closer to 1. As stated previously, it can be attained by using larger $A^c$ and/or $B^c$ although it is a trade-off with a larger chip area. It should also be noted that the approximation error depends on the energy of input $\mathbf{x}$.

The magnitude of the round-off error depends on the word-length of register, especially the number of fractional bits after the fixed point, which is denoted as $B^r$. The round-off error $\mathbf{e}^{cr}(k)$ introduced at the $k$-th sub-rotation is propagated through the subsequent CORDIC sub-rotations. Therefore, the sum of the accumulated round-off errors on the CORDIC output is given by

$$\mathbf{e}^{cr} = \sum_{k=0}^{K-1}\big(\prod_{l=k+1}^{K-1}\mathbf{P}(l)\big)\mathbf{e}^{cr}(k) \tag{13}$$

where $\mathbf{P}(l)$ is the matrix representing the $l$-th sub-rotation and represented as

$$\mathbf{P}(l) = \begin{bmatrix} \sum_{i=0}^{I(l)-1}\eta_i(l)2^{-p_i(l)} & -\sum_{j=0}^{J(l)-1}\mu_j(l)2^{-q_j(l)} \\ \sum_{j=0}^{J(l)-1}\mu_j(l)2^{-q_j(l)} & \sum_{i=0}^{I(l)-1}\eta_i(l)2^{-p_i(l)} \end{bmatrix}. \tag{14}$$

Now, we derive $E\{(\mathbf{e}^{cr})^T(\mathbf{e}^{cr})\}$ as

$$\begin{aligned} &E\{(\mathbf{e}^{cr})^T(\mathbf{e}^{cr})\} \\ &= E\{\sum_{k=0}^{K-1}(\mathbf{e}^{cr}(k))^T\prod_{l=K-1}^{k+1}\mathbf{P}(l)^T\sum_{i=0}^{K-1}\prod_{j=i+1}^{K-1}\mathbf{P}(j)\mathbf{e}^{cr}(i)\} \\ &= E\{\sum_{k=0}^{K-1}\text{Trace}\{\prod_{l=k+1}^{K-1}\mathbf{P}(l)\mathbf{e}^{cr}(k)(\mathbf{e}^{cr}(k))^T\prod_{l=K-1}^{k+1}\mathbf{P}(l)^T\}\} \\ &= E\{\sum_{k=0}^{K-1}\text{Trace}\{\mathbf{e}^{cr}(k)(\mathbf{e}^{cr}(k))^T\prod_{l=K-1}^{k+1}\mathbf{P}(l)^T\prod_{l=k+1}^{K-1}\mathbf{P}(l)\}\} \end{aligned} \tag{15}$$

assuming that $E\{\mathbf{e}^{cr}(k)^T\mathbf{e}^{cr}(l)\} = 0$ when $k \neq l$. Let $S = \prod_{k=0}^{K-1} S(k)$, where

$$S(k) = \sqrt{\Big(\sum_{i=0}^{I(k)-1}2^{-p_i(k)}\Big)^2 + \Big(\sum_{j=0}^{J(k)-1}2^{-q_j(k)}\Big)^2} \tag{16}$$

from (8). Then, (15) is rewritten as

$$E\{(\mathbf{e}^{cr})^T(\mathbf{e}^{cr})\} = \sum_{k=0}^{K-1}\prod_{l=k+1}^{K-1}S(l)^2E\{(\mathbf{e}^{cr}(k))^T\mathbf{e}^{cr}(k)\}. \tag{17}$$

When a two's complement data with $B^r$ fractional bits is truncated by $i$ right shift operation ($x >> i$), the error variance is represented as $\frac{2^{-2B^r}}{12}(1-2^{-2i})$ [27]. Thus,

$$\begin{aligned} & E\{\mathbf{e}^{cr}(k)^T\mathbf{e}^{cr}(k)\} = \\ & \frac{2^{-2B^r}}{12}\Big(\sum_{i=0}^{I(k)-1}(1-2^{-2p_i(k)}) + \sum_{j=0}^{J(k)-1}(1-2^{-2q_j(k)})\Big). \end{aligned} \tag{18}$$

Finally, assuming that $\mathbf{e}^{ca}$ is uncorrelated with $\mathbf{e}^{cr}$ and the means of all the errors are zero, the total MSE of MSR-CORDIC is expressed as

$$\begin{aligned} & E\{(\mathbf{e}^c)^T\mathbf{e}^c\} = E\{(\mathbf{e}^{ca})^T\mathbf{e}^{ca}\} + E\{(\mathbf{e}^{cr})^T\mathbf{e}^{cr}\} \\ & = E\{|\mathbf{x}|^2\}(\delta^2 + S^2 - 2S + 1) + \sum_{k=0}^{K-1}\prod_{l=k+1}^{K-1} \\ & S(l)^2\frac{2^{-2B^r}}{12}\left(\sum_{i=0}^{I(k)-1}(1-2^{-2p_i(k)}) + \sum_{j=0}^{J(k)-1}(1-2^{-2q_j(k)})\right). \end{aligned} \tag{19}$$

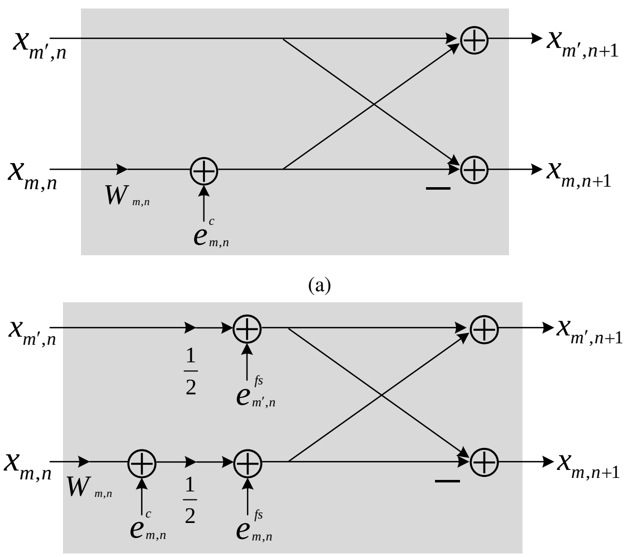
It should be noted that although the order of the sub-rotations is changed, the approximation error remains unchanged due to the same $S$ and $\delta$. However, the round-off error depends on the order of sub-rotations. Thus, we need to adjust the order of the sub-rotations after the searching process is completed such that (17) is minimized.

### B. Error Analysis of FFT with MSR-CORDIC

In the DIT FFT algorithm, the basic repetitive operation is the multiplication with twiddle factors followed by *Butterfly* operations as shown in Fig. 2 (a). The operation of the $m$-th and $m'$-th channels at the $n$-th FFT stage can be expressed as

$$\begin{aligned} & \begin{bmatrix} x_{m',n+1} \\ x_{m,n+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & W_{m,n} \end{bmatrix} \begin{bmatrix} x_{m',n} \\ x_{m,n} \end{bmatrix} \quad \text{for} \\ & 0 \le m, m' < N,\ 0 \le n < \log_2 N,\ \text{and} \\ & m = m' + 2^n \end{aligned} \tag{20}$$

where $x_{m,n}$ and $x_{m',n}$ are the input complex numbers of the $m$-th and $m'$-th channels at the $n$-th stage, respectively, and $W_{m,n}$ is the twiddle factor. Also, $\log_2 N$ is the number of stages for the $N$-point radix-2 FFT. When the twiddle factor multiplication $W_{m,n}x_{m,n}$ is implemented using MSR-CORDIC algorithm, the error $e^c_{m,n}$ is introduced as shown in Fig. 2 (a), and its MSE can be estimated using (19).



(a)

(b)

Fig. 2. The *Butterfly* operation of the radix-2 DIT FFT algorithm with error model at the $n$-stage when (a) the scaling process is not performed, (b) the down-scaling by the factor $1/2$ is performed.

Now, the propagation of $e^c_{m,n}$ into the last stage of FFT is examined in order to estimate the MSE of the FFT output. Let $\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ be denoted as $\mathbf{B}$, and $\begin{bmatrix} 1 & 0 \\ 0 & W_{m,n} \end{bmatrix}$ as $\mathbf{W}_{m,n}$ in (20). Since $\mathbf{B}^H\mathbf{B} = 2 \times \mathbf{I}$ where $[\cdot]^H$ is Hermitian of the matrix, the MSE of $e^c_{m,n}$ is doubled due to the *Butterfly* operation, and propagated into the $(n+1)$-th stage. Similarly, the MSE is doubled at every stage from the $(n+1)$-th stage to the last, because

$$(\mathbf{B}[\mathbf{W}_{m,n}]_Q)^H(\mathbf{B}[\mathbf{W}_{m,n}]_Q) \simeq 2 \times \mathbf{I} \quad \text{for all } m \text{ and } n. \tag{21}$$

In summary, the MSE introduced at the $n$-th stage is amplified by a factor of $2^{\log_2 N - n}$ after completing the computation of the last stage. It should also be noted that some twiddle factors having trivial values such as $\pm 1$ or $\pm j$ do not require any complex multiplications, and thus do not generate errors. Finally, the MSE per sample of FFT output can be represented as

$$\begin{aligned} &E\{(\mathbf{e}^f)^T\mathbf{e}^f\} = \\ &\quad \frac{1}{N}\left(\sum_{n=0}^{\log_2 N-1}\sum_{m=0}^{N-1} E\{(\mathbf{e}^c_{m,n})^T\mathbf{e}^c_{m,n}\}2^{\log_2 N-n}\right) \end{aligned} \tag{22}$$

where

$$
\begin{aligned}
&E\{(\mathbf{e}^c_{m,n})^T\mathbf{e}^c_{m,n}\} = \\
&\quad\begin{cases} 0 \quad \text{if } W_{m,n} = \pm 1, \pm j, \text{or } (m \;\mathbf{mod}\; 2^{n+1}) < 2^n, \\ \text{given in (19)} \quad \text{otherwise.} \end{cases}
\end{aligned} \tag{23}
$$

Meanwhile, the intermediate data is usually down-scaled by $1/2$ for preventing the overflow during the *Butterfly* calculation as shown in Fig. 2 (b) (The scaling factor is $1/p$ in the radix-$p$ FFT.). In this case, (22) is replaced with

$$
\begin{aligned}
&E\{(\mathbf{e}^f)^T\mathbf{e}^f\} = \\
&\quad \frac{1}{N}\Bigg(\sum_{n=0}^{\log_2 N-1}\sum_{m=0}^{N-1}\Bigg(E\{(\mathbf{e}^c_{m,n})^T\mathbf{e}^c_{m,n}\}\big(\frac{1}{2}\big)^{\log_2 N-n} \\
&\quad + E\{(\mathbf{e}^{fs}_{m,n})^T\mathbf{e}^{fs}_{m,n}\}2\big(\frac{1}{2}\big)^{\log_2 N-n-1}\Bigg)\Bigg).
\end{aligned} \tag{24}
$$

The first term of the summation in the right hand side in (24) is the modification of (22) based on the fact that $(\frac{1}{2}\mathbf{B})^H(\frac{1}{2}\mathbf{B}) = \frac{1}{2} \times \mathbf{I}$, and the second term of that is contributed by the down-scaling operation, where

$$
E\{(\mathbf{e}^{fs}_{m,n})^T\mathbf{e}^{fs}_{m,n}\} = 2\frac{2^{-2B^r}}{12}(1-2^{-2}) \;\text{ for all } m, n. \tag{25}
$$

Note that (24) is obtained as the weighted sum of MSEs by assuming that the errors generated at the different stages are uncorrelated. However, the assumption needs to be examined further. More specifically, let us rewrite (24) as

$$
\begin{aligned}
&E\{(\mathbf{e}^f)^T\mathbf{e}^f\} = \\
&\quad E\{(\mathbf{e}^{fa})^T\mathbf{e}^{fa}\} + E\{(\mathbf{e}^{fr})^T\mathbf{e}^{fr}\} + E\{(\mathbf{e}^{fs})^T\mathbf{e}^{fs}\}
\end{aligned} \tag{26}
$$

where superscripts '$fa$', '$fr$', and '$fs$' are abbreviations for 'FFT approximation', 'FFT round-off', and 'FFT scaling', respectively. It should be noted that $\mathbf{e}^{fa}$ and $\mathbf{e}^{fr}$ are originated from the MSR-CORDIC, and $\mathbf{e}^{fs}$ are from the down-scaling operation. According to (19) and (24), each term in (26) is represented as

$$E\{(\mathbf{e}^{fa})^T\mathbf{e}^{fa}\} = \frac{1}{N}\sum_{n=0}^{\log_2 N-1}\sum_{m=0}^{N-1}E\{(\mathbf{e}^{ca}_{m,n})^T\mathbf{e}^{ca}_{m,n}\}\big(\frac{1}{2}\big)^{\log_2 N-n}, \tag{27}$$

$$E\{(\mathbf{e}^{fr})^T\mathbf{e}^{fr}\} = \frac{1}{N}\sum_{n=0}^{\log_2 N-1}\sum_{m=0}^{N-1}E\{(\mathbf{e}^{cr}_{m,n})^T\mathbf{e}^{cr}_{m,n}\}\big(\frac{1}{2}\big)^{\log_2 N-n}, \tag{28}$$

$$E\{(\mathbf{e}^{fs})^T\mathbf{e}^{fs}\} = \frac{1}{N}\sum_{n=0}^{\log_2 N-1}\sum_{m=0}^{N-1}E\{(\mathbf{e}^{fs}_{m,n})^T\mathbf{e}^{fs}_{m,n}\}2\big(\frac{1}{2}\big)^{\log_2 N-n-1} \tag{29}$$

where $E\{(\mathbf{e}^{ca}_{m,n})^T\mathbf{e}^{ca}_{m,n}\}$, $E\{(\mathbf{e}^{cr}_{m,n})^T\mathbf{e}^{cr}_{m,n}\}$, and $E\{(\mathbf{e}^{fs}_{m,n})^T\mathbf{e}^{fs}_{m,n}\}$ are given as (12), (18), and (25), respectively. Many published articles have justified (28) and (29) by ignoring the second order effect among the errors, and they are proved through exhaustive simulations [8], [27], [28]. However, the equality in (27) often causes significant difference between the analysis and simulation, because the approximation errors introduced in the same data path are highly correlated. An alternative method for estimating the approximation error is to calculate the difference between the ideal and approximated matrix of DFT directly. To be more precise, let $\mathbf{D}_N$ be the true $N \times N$ DFT matrix and $\mathbf{x}$ be the $N \times 1$ input vector of the matrix. The output MSE then can be computed as

$$\begin{aligned} &E\{(\mathbf{e}^{fa})^H\mathbf{e}^{fa}\} \\ &\quad = \frac{1}{N}E\{\big((\mathbf{D}_N - [\mathbf{D}_N]_Q)\mathbf{x}\big)^H\big(\mathbf{D}_N - [\mathbf{D}_N]_Q\big)\mathbf{x}\} \\ &\quad = \frac{1}{N}\text{Trace}\{(\mathbf{D}_N - [\mathbf{D}_N]_Q)\mathbf{R}_{\mathbf{xx}}(\mathbf{D}_N - [\mathbf{D}_N]_Q)^H\} \end{aligned} \tag{30}$$

where $\mathbf{R}_{\mathbf{xx}} = E\{\mathbf{x}\mathbf{x}^H\}$. By replacing (27) with the more precise representation in (30), the output MSE of FFT can be rewritten as

$$\begin{aligned} &E\{(\mathbf{e}^{f})^T\mathbf{e}^{f}\} = \\ &\quad \frac{1}{N}\bigg(\text{Trace}\{(\mathbf{D}_N - [\mathbf{D}_N]_Q)\mathbf{R}_{\mathbf{xx}}(\mathbf{D}_N - [\mathbf{D}_N]_Q)^H\} \\ &\quad + \sum_{n=0}^{\log_2 N-1}\sum_{m=0}^{N-1}\big(E\{(\mathbf{e}^{cr}_{m,n})^T\mathbf{e}^{cr}_{m,n}\}(\frac{1}{2})^{\log_2 N-n} \\ &\quad + E\{(\mathbf{e}^{fs}_{m,n})^T\mathbf{e}^{fs}_{m,n}\}2(\frac{1}{2})^{\log_2 N-n-1}\big)\bigg). \end{aligned} \tag{31}$$

The MSE can also be represented as SQNR defined as

$$\text{SQNR (dB)} = 10 \log_{10} \frac{E\{\mathbf{x}^H \mathbf{x}\}}{N^2 E\{(\mathbf{e}^f)^T \mathbf{e}^f\}} \tag{32}$$

where the term $N^2$ is due to down-scaling operation. The output MSE of FFT with different radix or split-radix can also be estimated with a few modifications of the proposed analysis. It is observed that our analysis closely matches the simulation results obtained in next sections.

## IV. Design of FFT with Generalized MSR-CORDIC

In generalized MSR-CORDIC, complex multipliers are often reused to minimize the silicon area of the integrated circuits, and all the complex multiplications are designed to have a uniform structure. For this purpose, the parameters such as $A^c$, $K$, $I(k)$, and $J(k)$ should be fixed for all the multiplications. Meanwhile, the 'generalized MSR-CORDIC' module which has been proposed in [22] (see Fig. 9 in [22]) permits the flexibility of $I(k)$ and $J(k)$. In this section, we propose a novel determination algorithm of the parameters of MSR-CORDIC when the generalized MSR-CORDIC is employed for FFT. We minimize the number of adders and bit-width of complex multipliers when the SQNR of the FFT output is given as a design constraint. The twiddle factor multiplication and scaling in *Butterfly* operation shown in Fig. 2 (b) are the only sources of the MSE of the FFT output. Therefore, error analysis equations derived in the previous section are used for the proposed parameter determination algorithm.

In Subsection IV-A, the determination algorithm of the SPT coefficients $\sum_{i=0}^{I(k)-1} \eta_i(k) 2^{-p_i(k)}$ and $-\sum_{j=0}^{J(k)-1} \mu_j(k) 2^{-q_j(k)}$ based on the derived MSR-CORDIC error equation (19) is introduced for given parameters $B^c$, $K$, and $A^c(k)$. Then, the determination procedure of $K$ and $A^c(k)$ is developed in Subsection IV-B under the assumption that $A^c$ and $B^c$ are fixed. Finally, Subsection IV-C provides the flowchart for the determination of $A^c$ and $B^c$.

### A. Determination of SPT Coefficients in MSR-CORDIC

Since the parameters of the MSR-CORDIC in (10) cannot be directly obtained, it should be searched within a discrete coefficient space using a proper optimization technique. The accuracy of the fixed-point FFT depends largely on the ability of the employed searching method. Assume that the number of adders for each CORDIC sub-rotation $A^c(k)$ and coefficient word-length $B^c$ are fixed. The definitions for $A^c(k)$ and $B^c$ are given in (11) and (8), respectively, and the determination algorithm of $A^c(k)$ and $B^c$ is presented in next section. The optimization technique uses (19) as an objective function to search the two SPT coefficients $\sum_{i=0}^{I(k)-1} \eta_i(k) 2^{-p_i(k)}$ and $-\sum_{j=0}^{J(k)-1} \mu_j(k) 2^{-q_j(k)}$ in (10) for given $A^c(k)$ and $B^c$.

For this purpose, we need to collect all the possible SPT coefficients which satisfy (11). The following constraint is added to remove the duplicate candidates:

$$\{p_0(k), ..., p_{I(k)-1}(k), q_0(k), ..., q_{J(k)-1}(k)\} \text{ includes zero} \\ \text{for all } k \tag{33}$$

and it can be attained by adding the same integer to all the elements in the set of (33). It provides not only the reduced discrete candidate space, thus speeding up the searching process, but also the reduced round-off noise because the number of truncated bits is reduced during the bit-shift operation.

The direct approach is to create all the candidates of (10) using exhaustive combinations of possible SPT numbers which are consistent with (11) and (33), and then find the one which minimizes (19) in the candidate pool. For this purpose, the pools of SPTs, sub-rotations, and rotations are created one after another. Then, the candidate which has the smallest MSE is chosen in the pool of rotations. The exhaustive search is computationally acceptable if $A^c$ and/or $B^c$ are small. Otherwise, it requires high computational complexity and memory storage due to the vast number of candidates even if it can provide a global optimum.

Another method is to make use of the suboptimal approaches. They create only the pools of SPTs and sub-rotations during the optimization whereas the approximated rotation is obtained through specific strategies such as greedy or trellis based algorithms. Their computational complexity can be significantly reduced because the pool of rotations is not created. However, the MSE would be larger than that obtained by exhaustive search. The algorithm descriptions are omitted in this paper. Please refer to [25] for further details of trellis based algorithm, and [21] for those of greedy algorithm.

In $N$-point DIT FFT algorithms, the angle $\theta_{m,n}$ of the twiddle factor $W_{m,n}$, where $W_{m,n} = e^{j\theta_{m,n}}$, is represented as $-2\pi k/N$, where $k$ is an integer within $0 \leq k < N/2$. Thus, the twiddle factor angle falls within the range of $-\pi < \theta_{m,n} \leq 0$. When the twiddle factor is $\theta_{m,n} = 0$ or $-\pi/2$, the multiplication can be done without using multiplier circuitry. Moreover, the periodicity property of the twiddle factors can be exploited to reduce the complexity of the searching process further as follows:

$$\begin{aligned}
&\text{for } -\pi/2 < \theta_{m,n} < -\pi/4,\\
&[\mathbf{R}(\theta_{m,n})]_Q = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} [\mathbf{R}(-\theta_{m,n} - \pi/2)]_Q \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}\\
&\text{for } -3\pi/4 \leq \theta_{m,n} < -\pi/2,\\
&[\mathbf{R}(\theta_{m,n})]_Q = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} [\mathbf{R}(\theta_{m,n} + \pi/2)]_Q\\
&\text{for } -\pi < \theta_{m,n} < -3\pi/4,\\
&[\mathbf{R}(\theta_{m,n})]_Q = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} [\mathbf{R}(-\theta_{m,n} - \pi)]_Q \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}.
\end{aligned} \tag{34}$$

If $[\mathbf{R}(\theta_{m,n})]_Q$'s have already been obtained for the angle $-\pi/4 \leq \theta_{m,n} < 0$, those existing beyond the range can be obtained without any additional searching process using this property; thus the overhead of the searching process can be further reduced.

### *B. Determination of $K$ and $A^c(k)$*

$A^c(k)$ and $K$ can be determined for given $A^c$ and $B^c$ as follows: First, the set $\mathbf{S}(A^c)$ defined as

$$\begin{aligned}
\mathbf{S}(A^c) = \Big\{ \big(A^c(0), ..., A^c(K-1)\big) : A^c = \sum_{k=0}^{K-1} A^c(k),\\
A^c(k) \text{ is even, } 0 < A^c(k) \leq A^c(l) \text{ for } k < l \Big\},
\end{aligned} \tag{35}$$

TABLE I

THE ESTIMATED MSE AND SQNR (DB) OF THE 128-POINT RADIX-2 DIT FFT WHERE THE COMPLEX MULTIPLIERS ARE IMPLEMENTED IN THE STRUCTURE CORRESPONDING TO EACH ELEMENT IN THE SET $\mathbf{S}(8)$.

| | element $\in \mathbf{S}(A^c)$ | $K$ | $E\{(\mathbf{e}^f)^T\mathbf{e}^f\}$ | SQNR |
|---|---|---|---|---|
| $A^c = 8$ | $(8)$ | $1$ | $1.772080e{-5}$ | $29.45$ |
| | $(4,4)$ | $2$ | $1.131848e{-6}$ | $41.43$ |
| | $(2,6)$ | $2$ | $5.147796e{-7}$ | $44.86$ |
| | $(2,2,4)$ | $3$ | $3.533044e{-7}$ | $46.49$ |
| | $(2,2,2,2)$ | $4$ | $1.055369e{-6}$ | $41.74$ |

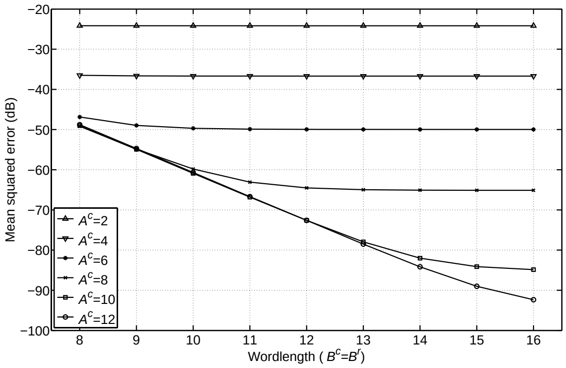


Fig. 3. The MSE (dB) versus word-length $B^c$ (=$B^r$) in the 128-point radix-2 DIT FFT algorithm.

is obtained for the given $A^c$. The element $\big(A^c(0), ..., A^c(K-1)\big)$ means that the rotation has $K$ sub-rotations and each of them has $A^c(0)$,...,$A^c(K-1)$ adders. From (11), we see that $A^c$ and $A^c(k)$ are even numbers for all $k$. For example, when $A^c = 8$, $\mathbf{S}(A^c)$ has five elements, $(8)$, $(4,4)$, $(2,6)$, $(2,2,4)$, and $(2,2,2,2)$. The element $(2,6)$ means that the rotation is implemented as the cascade of two sub-rotations ($K = 2$), and each sub-rotation includes 2 and 6 adders. In generalized MSR-CORDIC, only one element in $\mathbf{S}(A^c)$ should be selected and applied to all the twiddle factor multiplications in FFT. For this purpose, the searching process is performed for each case corresponding to each element in the set $\mathbf{S}(A^c)$ separately, and then the best element which minimizes the MSE of the FFT output is chosen. As an example, Table I shows the MSE and SQNR of a 128-point radix-2 DIT FFT estimated by (31) and (32), respectively when $A^c = 8$. In the simulation, we set $B^c = B^r = 12$, and use Gaussian random inputs with zero mean and standard deviation $0.1$ for the real and imaginary inputs. Also, the searching method presented in Subsection IV-A with the trellis-based algorithm is employed to obtain the SPT numbers. From the table, it can be seen that $(2,2,4)$ provides the minimum MSE among all the cases. Thus, we finally set $K = 2$, $A^c(0) = 2$, $A^c(1) = 2$, and $A^c(2) = 4$ for $A^c = 8$ and $B^c = 12$.

### C. Determination of $A^c$ and $B^c$

The MSE of FFT decreases with increasing $A^c$ because the approximation error reduces. The MSE also decreases as $B^r$ increases due to the reduction of round-off and scaling errors. However, this is not valid for specific $A^c$ and $B^r$. For more observations, the relationship between the MSE (dB) of the 128-point radix-2 DIT FFT and the word-length $B^c$ when $A^c$ ranges from 2 to 12 is shown in Fig. 3. $B^c$ should not be larger than $B^r$ since it is meaningless if the number of bits to be shifted is larger

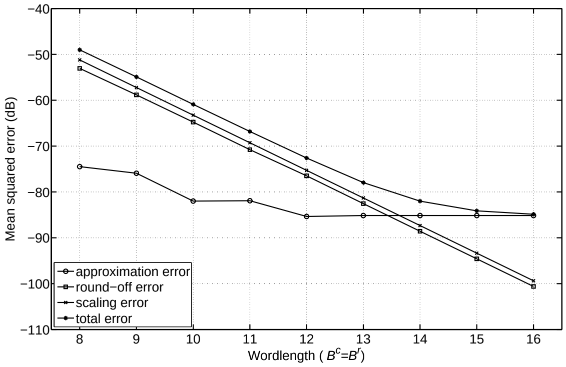


Fig. 4. The MSE (dB) of approximation, round-off, scaling, and total errors versus word-length $B^c$ (=$B^r$) in the 128-point radix-2 DIT FFT algorithm with $A^c = 10$.

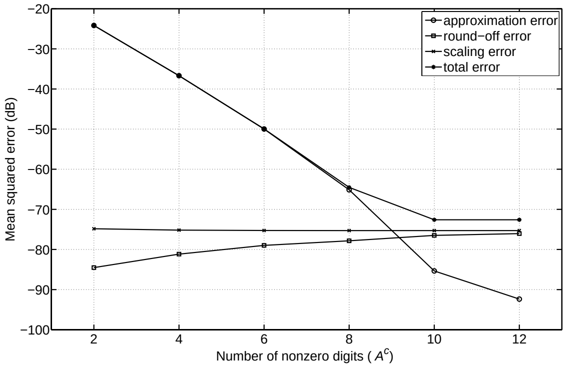


Fig. 5. The MSE (dB) of approximation, round-off, scaling, and total errors versus the number of nonzero digits $A^c$ in the 128-point radix-2 DIT FFT algorithm with $B^c = B^r = 12$.

than the word-length of the registers. Also, $B^c$ need not be smaller than $B^r$ for obtaining the higher approximation accuracy if the limitation in the ROM size is not strict. Thus, $B^c$ is set to have the same value as $B^r$. As shown in this figure, the MSE is not improved even though $A^c$ is increased from 10 to 12 if $B^r$ is smaller than 13. Also, if $A^c$ is small, larger word-length does not guarantee a smaller MSE. For better understanding, the MSEs (dB) of approximation, round-off, and scaling errors versus $B^r$ with $A^c = 10$ are illustrated in Fig. 4. The MSEs (dB) versus $A^c$ with $B^r = 12$ are also illustrated in Fig. 5. As shown in Fig. 4, when $B^r$ is small, the round-off and scaling errors become dominant error sources, and thus the total error is almost the same as the sum of these two errors. We can expect that if $A^c > 10$, the angle approximation error would be reduced further whereas the round-off error and

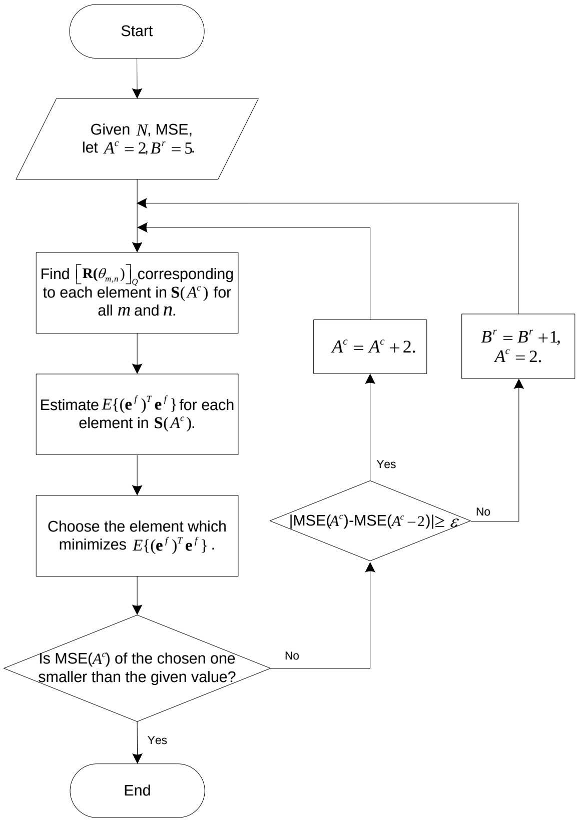

Start
Given $N$, MSE,
let $A^c = 2, B^r = 5.$
Find $[\mathbf{R}(\theta_{m,n})]_Q$ corresponding to each element in $\mathbf{S}(A^c)$ for all $m$ and $n$.
Estimate $E\{(\mathbf{e}^f)^T \mathbf{e}^f\}$ for each element in $\mathbf{S}(A^c)$.
Choose the element which minimizes $E\{(\mathbf{e}^f)^T \mathbf{e}^f\}$ .
Is MSE($A^c$) of the chosen one smaller than the given value?
Yes
End
No
|MSE($A^c$)-MSE($A^c - 2$)|≥ $\varepsilon$
Yes
No
$A^c = A^c + 2.$
$B^r = B^r + 1,$
$A^c = 2.$

Fig. 6. The flowchart for the determination of $A^c$ and $B^r$ in the FFT with generalized MSR-CORDIC.

scaling error would slightly increase with more SPT terms. Finally, the total error with $A^c > 10$ would be almost the same as that with $A^c = 10$ when $B^r$ is small, which could be seen in Fig. 3. Therefore, $B^r$ should be increased instead of $A^c$ in order to have a smaller MSE. On the contrary, when $A^c$ is small, the approximation becomes the dominant error source as shown in Fig. 5. Hence, a better MSE can not be obtained even if $B^r$ increases. In that case, $A^c$ should be increased.

Based on this observation, a procedure for determining $A^c$ and $B^r$ (=$B^c$) is proposed for a given MSE constraint. The flowchart of the detail design procedure is described in Fig. 6. $A^c$ and $B^r$ are initially set to small values (e.g. $A^c = 2$ and $B^r = 5$ in Fig. 6). For current $A^c$, all the elements in the set $\mathbf{S}(A^c)$ are obtained, and $[\mathbf{R}(\theta_{m,n})]_Q$ corresponding to each element are found for all $m$ and $n$ using the method presented. The element which provides the minimum MSE of FFT is chosen. If the minimum MSE for the current $A^c$ and $B^r$ is smaller than the given MSE constraint, the procedure is terminated, otherwise, more adders are allocated by setting $A^c = A^c + 2$, and the same procedure is repeated. If additional adders cannot reduce the MSE any more ($|\text{MSE}(A^c) - \text{MSE}(A^c - 2)| < \varepsilon$) in Fig. 6, the round-off and

TABLE II

THE MSE, SQNR (DB), AND PARAMETERS OBTAINED BY THE PROCEDURE IN FIG. 6 FOR THE $N$-POINT RADIX-2 DIT FFT WHEN GENERALIZED MSR-CORDIC IS EMPLOYED AND 50 DB IS GIVEN AS THE SQNR CONSTRAINT.

| | $N$ | estimated error | | | | | simulated error | $A^c$ | $\in \mathbf{S}(A^c)$ | $B^r$ |
|---|---|---|---|---|---|---|---|---|---|---|
| | | $E\{\lvert\mathbf{e}^{fa}\rvert^2\}$ | $E\{\lvert\mathbf{e}^{fr}\rvert^2\}$ | $E\{\lvert\mathbf{e}^{fs}\rvert^2\}$ | $E\{\lvert\mathbf{e}^{f}\rvert^2\}$ | SQNR | $E\{\lvert\mathbf{e}^{f}\rvert^2\}$ | | | |
| given SQNR constraint: 50 dB | 8 | $1.277e-8$ | $1.990e-7$ | $1.669e-6$ | $1.881e-6$ | 51.81 | $1.879e-6$ | 8 | $(2, 2, 4)$ | 9 |
| | 16 | $5.446e-7$ | $1.161e-7$ | $4.466e-7$ | $1.107e-6$ | 50.82 | $1.108e-6$ | 8 | $(2, 2, 4)$ | 10 |
| | 32 | $4.730e-7$ | $4.438e-8$ | $1.154e-7$ | $6.328e-7$ | 50.08 | $6.334e-7$ | 8 | $(2, 2, 4)$ | 11 |
| | 64 | $4.463e-9$ | $7.199e-8$ | $1.174e-7$ | $1.938e-7$ | 52.15 | $1.931e-7$ | 10 | $(2, 4, 4)$ | 11 |
| | 128 | $2.915e-9$ | $2.151e-8$ | $2.957e-8$ | $5.400e-8$ | 54.65 | $5.426e-8$ | 10 | $(4, 6)$ | 12 |
| | 256 | $2.520e-9$ | $2.231e-8$ | $2.969e-8$ | $5.452e-8$ | 51.58 | $5.543e-8$ | 10 | $(4, 6)$ | 12 |
| | 512 | $1.757e-9$ | $5.729e-9$ | $7.436e-9$ | $1.492e-8$ | 54.19 | $1.515e-8$ | 10 | $(4, 6)$ | 13 |

down-scaling are considered as the dominant error sources. Thus, $A^c$ is initialized and $B^r$ is increased. The MSE in Fig. 6 could be replaced with the SQNR without any modifications.

Table II shows the obtained MSE, SQNR, $A^c$, $B^r$ when $N$ (from 8 to 512)-point radix-2 DIT FFT is designed using the proposed procedure and 50 dB is given as the SQNR constraint. The simulated errors are also listed in the table to show that our estimates closely match the actual errors. Specially, the coefficient values for the case of 128-point are listed in Table III. $[\mathbf{R}(\theta_{m,n})]_Q$ can be obtained by substituting the listed sum of SPT coefficients into (10).

## V. DESIGN OF FFT WITH DEDICATED MSR-CORDIC

The generalized MSR-CORDIC processor is designed to perform multiplications of diverse twiddle factors. In that case, the parameters of the CORDIC are obtained via off-line optimization, and stored in ROM. The complex multiplier reads the appropriate parameters according to the twiddle factor angle from ROM. However, in the FFT with dedicated MSR-CORDIC where each complex multiplier is designed to have its own circuit, very high throughput rate can be achieved when it is implemented as the fully parallel structure with minimum memory storages but more CORDIC rotators. Also, the $N$-point FFT with dedicated MSR-CORDIC can be used as sub-blocks of radix-$N$ FFT. Since the reusability and regularity of the complex multiplier are not constrained in the design, the parameters such as $A^c$, $A^c(k)$, $K$, $I(k)$, and $J(k)$ are not necessary to be fixed for all the complex multipliers. In other words, each complex multiplier can have not only different number of adders but also different elements in the set $\mathbf{S}(A^c)$, and therefore, provides a smaller approximation error than that with the generalized MSR-CORDIC. A

TABLE III

THE COEFFICIENT VALUES OF THE 128-POINT RADIX-2 DIT FFT IN TABLE II. $[\mathbf{R}(\theta_{m,n})]_Q$ CAN BE OBTAINED USING THE SUM OF SPT COEFFICIENTS IN THE TABLE AND (10).

| twiddle factors | $A^c(0) = 4$ | | $A^c(1) = 6$ | |
|---|---|---|---|---|
| | $\sum_i \eta_i(0)2^{-p_i(0)}$ | $\sum_j \mu_j(0)2^{-q_j(0)}$ | $\sum_i \eta_i(1)2^{-p_i(1)}$ | $\sum_j \mu_j(1)2^{-q_j(1)}$ |
| $W_{128}^{1}$ | $2^{-4}$ | $2^0 - 2^{-9}$ | $2^{-6} - 2^{-9} - 2^{-12}$ | $-2^0$ |
| $W_{128}^{2}$ | $2^0$ | $2^{-5} - 2^{-8}$ | $2^0 - 2^{-7} - 2^{-11}$ | $-2^{-3}$ |
| $W_{128}^{3}$ | $2^0 + 2^{-7}$ | $-2^{-3}$ | $2^0 - 2^{-6}$ | $-2^{-5} + 2^{-7}$ |
| $W_{128}^{4}$ | $2^0 + 2^{-12}$ | $-2^{-2}$ | $2^0 - 2^{-5}$ | $2^{-4} - 2^{-6}$ |
| $W_{128}^{5}$ | $2^{-2} + 2^{-11}$ | $2^0$ | $0$ | $-2^0 + 2^{-5} - 2^{-10} - 2^{-12}$ |
| $W_{128}^{6}$ | $2^0 + 2^{-3} - 2^{-5}$ | $0$ | $2^0 - 2^{-3}$ | $-2^{-2} - 2^{-6}$ |
| $W_{128}^{7}$ | $2^0 - 2^{-5}$ | $-2^{-3}$ | $2^0 + 2^{-12}$ | $-2^{-2} + 2^{-5}$ |
| $W_{128}^{8}$ | $2^0 - 2^{-4}$ | $-2^{-5}$ | $2^0 - 2^{-9}$ | $-2^{-1} + 2^{-3}$ |
| $W_{128}^{9}$ | $2^{-1}$ | $2^0 + 2^{-5}$ | $2^{-7}$ | $-2^0 + 2^{-3} + 2^{-9}$ |
| $W_{128}^{10}$ | $2^{-6} - 2^{-10}$ | $-2^0$ | $2^{-1} - 2^{-6}$ | $2^0 - 2^{-3}$ |
| $W_{128}^{11}$ | $2^0 - 2^{-6}$ | $-2^{-7}$ | $2^0 - 2^{-3}$ | $-2^{-1} - 2^{-6}$ |
| $W_{128}^{12}$ | $2^0 + 2^{-8}$ | $-2^{-4}$ | $2^0 - 2^{-3} - 2^{-6}$ | $-2^{-1}$ |
| $W_{128}^{13}$ | $2^0$ | $-2^0 + 2^{-3}$ | $2^0 - 2^{-2}$ | $2^{-4} - 2^{-9}$ |
| $W_{128}^{14}$ | $2^{-2}$ | $2^0 - 2^{-3}$ | $-2^{-1} + 2^{-4}$ | $-2^0 - 2^{-7}$ |
| $W_{128}^{15}$ | $2^0 - 2^{-2}$ | $-2^{-6}$ | $2^0 + 2^{-7}$ | $-2^0 + 2^{-3}$ |
| $W_{128}^{16}$ | $0$ | $-2^0 + 2^{-2} - 2^{-8}$ | $2^0 - 2^{-4}$ | $2^0 - 2^{-4}$ |

dynamic adder allocation algorithm is proposed to minimize the total number of adders consumed by the FFT processor when the MSE of the FFT output is given as the design constraint.

Let the adder cost for the multiplication with $W_{m,n}$ be $A^c_{m,n}$. Thus, the total number of adders allocated to all the complex multipliers are represented as

$$A^f = \sum_{n=0}^{\log_2 N-1} \sum_{m=0}^{N-1} A^c_{m,n}. \tag{36}$$

In addition,

$$A^c_{m,n} = 0 \text{ if } W_{m,n} = \pm 1, \pm j, \text{or } (m \ \mathbf{mod} \ 2^{n+1}) < 2^n. \tag{37}$$

The detailed design procedure is described in Fig. 7 (a). $A^c_{m,n}$ is initially set to $0$ for all $m$ and $n$, and $B^r$ is initialized to a small value (e.g. $B^r = 5$ in Fig. 7 (a)). Then, we need to determine ($m$, $n$) which can achieve the largest reduction for the MSE of the FFT output when more adders are allocated. As described in **Algorithm 1**, we find $[\mathbf{R}(\theta_{m,n})]_Q$ when two more adders are allocated to the CORDIC in

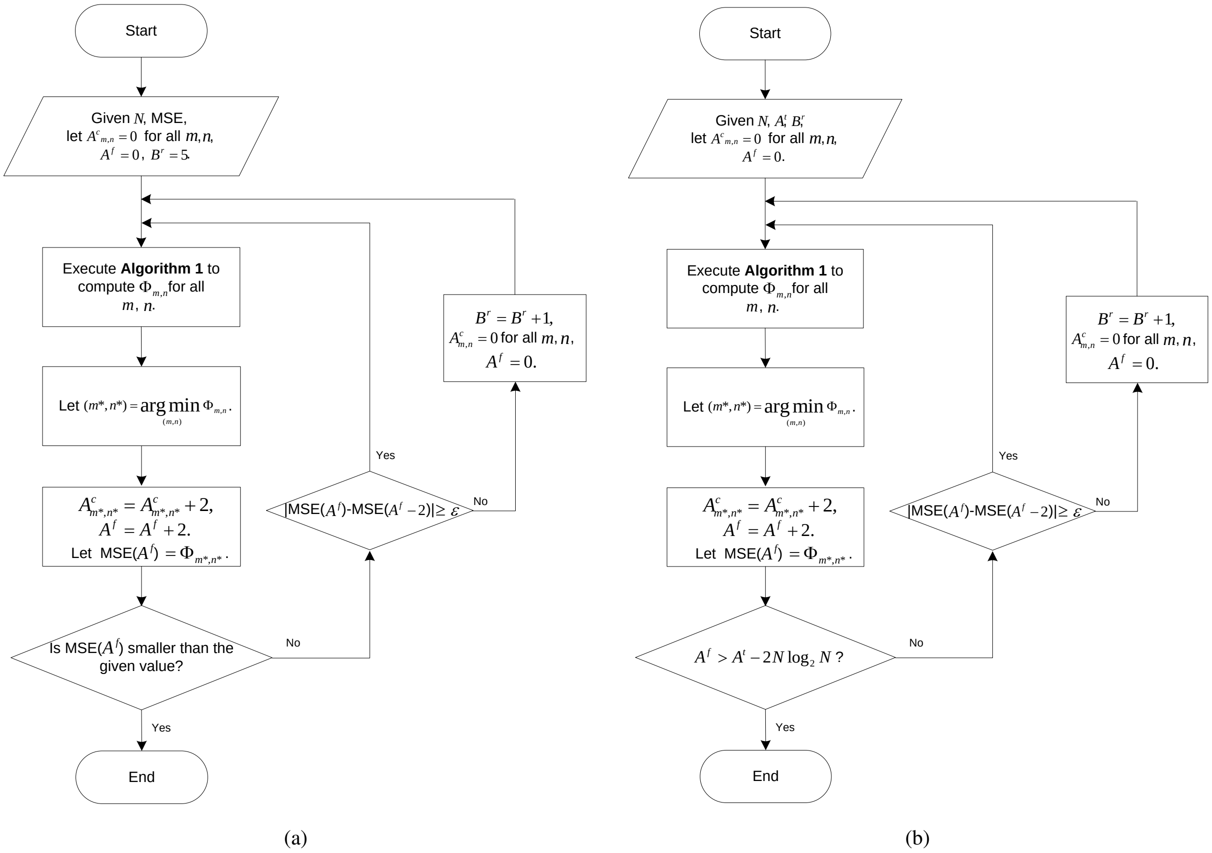

Start
Given N, MSE, let $A^c_{m,n} = 0$ for all m,n, $A^f = 0$, $B^r = 5$.
Execute Algorithm 1 to compute $\Phi_{m,n}$ for all m, n.
Let $(m^*, n^*) = \arg\min_{(m,n)} \Phi_{m,n}$.
$A^c_{m^*,n^*} = A^c_{m^*,n^*} + 2$, $A^f = A^f + 2$. Let MSE($A^f$) = $\Phi_{m^*,n^*}$.
Is MSE($A^f$) smaller than the given value?
Yes
No
End
|MSE($A^f$)-MSE($A^f - 2$)| ≥ ε
Yes
No
$B^r = B^r + 1$, $A^c_{m,n} = 0$ for all m, n, $A^f = 0$.
(a)
Start
Given N, $A^t$, $B^r$, let $A^c_{m,n} = 0$ for all m,n, $A^f = 0$.
Execute Algorithm 1 to compute $\Phi_{m,n}$ for all m, n.
Let $(m^*, n^*) = \arg\min_{(m,n)} \Phi_{m,n}$.
$A^c_{m^*,n^*} = A^c_{m^*,n^*} + 2$, $A^f = A^f + 2$. Let MSE($A^f$) = $\Phi_{m^*,n^*}$.
$A^f > A^t - 2N \log_2 N$ ?
Yes
No
End
|MSE($A^f$)-MSE($A^f - 2$)| ≥ ε
Yes
No
$B^r = B^r + 1$, $A^c_{m,n} = 0$ for all m, n, $A^f = 0$.
(b)

Fig. 7. The flowcharts for the determination of $A^c_{m,n}$ and $B^r$ in the FFT with dedicated MSR-CORDIC, when (a) the MSE is given as the design constraint, (b) the total number of adders consumed by FFT, $A^t$, is given as the design constraint.

the location of $(m, n)$. $E\{|\mathbf{e}^f|^2\}$ is estimated for a newly found $[\mathbf{R}(\theta_{m,n})]_Q$, and denoted as $\Phi_{m,n}$. We repeat the procedure until $\Phi_{m,n}$ are obtained for all $m$ and $n$, and the index $(m^*, n^*)$ that provides the minimum output MSE is chosen. If the minimum MSE for the current $A^f$ and $B^r$ is smaller than the given MSE constraint, the procedure is terminated. Otherwise, more adders are allocated, and the same procedure is repeated.

The logic depth of the complex multiplier may be constrained to reduce the critical path and maximize the throughput of the FFT. Note that we can control the critical path by limiting the number of adders in all the CORDICs. With a little modification of flowchart in Fig. 7 (a), the constraint for the maximum number of $A^c_{m,n}$ can be added.

It is well known that the number of non-trivial complex multiplications of $N$-point radix-2 DIT FFT algorithm is given as $(N/2)(\log_2 N - 3) + 2$. Whenever every 2 real adders are allocated, **Algorithm 1** is executed. Thus, (31) should be evaluated $(N/2)(\log_2 N - 3) + 2$ times whenever 2 adders are allocated.

**Algorithm 1** Calculates $\Phi_{m,n}$ for all $m$ and $n$.

```
for $n = 0, ..., \log_2 N - 1$ do
  for $m = 0, ..., N - 1$ do
    if $W_{m,n} \neq \pm 1, \pm j, \ \cap \ (m \textbf{ mod } 2^{n+1}) \geq 2^n$ then
      Find $[\mathbf{R}(\theta_{m,n})]_Q$ when $A^c_{m,n} = A^c_{m,n} + 2$.
      Set $\Phi_{m,n} = E\{|\mathbf{e}^f|^2\}$ when $[\mathbf{R}(\theta_{m,n})]_Q$ is deployed.
    end if
  end for
end for
```

If $N$ is large, the evaluation of (31) may require a significant amount of time. For example, if 512-point radix-2 DIT FFT is designed and $2 \times n$ adders on average are allocated to each complex multiplier, (31) should be executed $1,538 \times 1,538 \times n$ times for $1,538$ non-trivial complex multipliers. The evaluation of $E\{(\mathbf{e}^{fa})^H \mathbf{e}^{fa}\}$ term in (31), that is, (30) requires high computational complexity when $N$ is large. In that case, (24) would be more computationally efficient than (31) with a little degradation of accuracy. The second term in the summation in (24) associated with down-scaling does not vary with new allocations of adders since the term does not depend on the MSR-CORDIC parameters. Furthermore, the first term in the summation in (24) is obtained through (19), where $E\{(\mathbf{e}^{cr})^T \mathbf{e}^{cr}\}$ can be omitted because the procedure in Fig. 7 (a) always works in the state that the approximation error is dominant (As shown in Fig. 7 (a). When the round-off error becomes dominant, the algorithm increases $B^r$ and resets $A^c_{m,n}$, making the approximation error dominant again.). Finally, only the approximation error term $E\{(\mathbf{e}^{ca})^T \mathbf{e}^{ca}\}$ needs to be evaluated at every iteration. Meanwhile, since the input energy $E\{|\mathbf{x}_{m,n}|^2\}$ in (19) should be updated every iteration resulting in time-consuming. If $E\{|\mathbf{x}_{m,n}|^2\}$ is obtained using the ideal $W_{m,n}$ only once, and is used during the whole procedure, the computational time could be significantly reduced. Our design experience shows that all these simplifications do not cause severe deterioration.

In the design with dedicated MSR-CORDIC, the silicon area of the FFT highly depends on the total number of adders in FFT, denoted as $A^t$, since the area of hard-wired shifter is negligible. In some design problems, $A^t$ is constrained to minimize the silicon area. In this case, the design algorithm allocates the adders to each complex multiplier such that the total number of adders are not more than the given adder cost while the output MSE is minimized. As shown in Fig. 2, the *Butterfly* operation includes two complex additions and a complex multiplication. In addition, the $N$-point radix-2 DIT FFT algorithm

TABLE IV

The MSE, SQNR (dB), and parameters obtained by the procedure of Fig. 7 (a) for the $N$-point radix-2 DIT FFT when dedicated MSR-CORDIC is employed and 50 dB is given as the SQNR constraint. Av. $A^c$ means average $A^c$ per each CORDIC which is calculated as $A^f/(\frac{N}{2}(\log_2 N - 3) + 2)$.

| | $N$ | estimated error | | | | | simulated error | $A^f$ | Av. $A^c$ | $A^t$ | $B^r$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $E\{\lvert\mathbf{e}^{fa}\rvert^2\}$ | $E\{\lvert\mathbf{e}^{fr}\rvert^2\}$ | $E\{\lvert\mathbf{e}^{fs}\rvert^2\}$ | $E\{\lvert\mathbf{e}^{f}\rvert^2\}$ | SQNR | $E\{\lvert\mathbf{e}^{f}\rvert^2\}$ | | | | |
| given SQNR constraint: 50 dB | 8 | $1.277e{-8}$ | $1.990e{-7}$ | $1.669e{-6}$ | $1.881e{-6}$ | 51.25 | $1.858e{-6}$ | 16 | 8 | 64 | 9 |
| | 16 | $5.024e{-7}$ | $1.137e{-7}$ | $4.467e{-7}$ | $1.063e{-6}$ | 50.70 | $1.062e{-6}$ | 76 | 7.6 | 204 | 10 |
| | 32 | $4.517e{-7}$ | $4.487e{-8}$ | $1.154e{-7}$ | $6.120e{-7}$ | 50.09 | $6.076e{-7}$ | 258 | 7.6 | 578 | 11 |
| | 64 | $1.243e{-7}$ | $5.624e{-8}$ | $1.173e{-7}$ | $2.979e{-7}$ | 50.21 | $2.962e{-7}$ | 764 | 7.8 | 1532 | 11 |
| | 128 | $1.109e{-7}$ | $1.492e{-8}$ | $2.956e{-8}$ | $1.554e{-7}$ | 50.02 | $1.553e{-7}$ | 1976 | 7.7 | 3768 | 12 |
| | 256 | $3.195e{-8}$ | $1.609e{-8}$ | $2.969e{-8}$ | $7.773e{-8}$ | 50.02 | $7.750e{-8}$ | 4972 | 7.7 | 9068 | 12 |
| | 512 | $2.764e{-8}$ | $3.986e{-9}$ | $7.438e{-9}$ | $3.906e{-8}$ | 50.00 | $3.906e{-8}$ | 11678 | 7.6 | 20894 | 13 |

consists of $\log_2 N$ stages, and each stage includes $N/2$ ***Butterfly*** operations. If $A^t$ real adders are given as the design constraint, $A^t - 2N\log_2 N$ (=$A^f$) real adders can be allocated for the complex multipliers. Fig. 7 (b) describes the adder allocation procedure which is similar to that in Fig. 7 (a) except that $A^t$ is given as the design constraint. The number of non-trivial complex multiplications of $N$-point radix-2 DIT FFT algorithm is given as $(N/2)(\log_2 N - 3) + 2$. Therefore, the average number of real adders for each complex multiplier becomes $A^f/((N/2)(\log_2 N - 3) + 2)$.

Table IV shows the obtained MSE, SQNR (dB), $A^f$, and $B^r$ when $N$ (from $8$ to $512$)-point radix-2 FFT is designed with dedicated MSR-CORDIC using the proposed procedure and the SQNR constraint is $50$ dB. The coefficient values for the 64-point example are listed in Table V. As shown in this table, $A^c_{m,n}$, $A^c_{m,n}(k)$, and $K$ may have different values for different $(m, n)$. Specifically, the SPT coefficients when $(m, n) = (5, 40)$ or $(5, 56)$ are different from those when $(m, n) \neq (5, 40)$ or $(5, 56)$. It is because the design algorithm is terminated while the adders are allocated to the multipliers with angle $W_{64}^{8}$.

The proposed design methods consider both the approximation and round-off errors whereas the conventional methods [7], [9] minimize only the Frobenius norm (FN) related with the approximation error under the assumption that the register word-length $B^r$ is sufficiently large. The FN is defined as

$$\text{FN} = \sqrt{\text{Trace}\{(\mathbf{D}_N - [\mathbf{D}_N]_Q)(\mathbf{D}_N - [\mathbf{D}_N]_Q)^H\}}. \tag{38}$$

Note that (30) provides better results than (38) because (30) reflects the statistics of the FFT input. For the comparison with the results of [7], [9] in terms of FN, the FFTs which minimize only approximation

TABLE V

THE SPT COEFFICIENT VALUES OF THE 64-POINT RADIX-2 DIT FFT IN TABLE IV. $[\mathbf{R}(\theta_{m,n})]_Q$ CAN BE OBTAINED USING THE SPT COEFFICIENTS IN THE TABLE AND (10). $W_{64}^8$-A: $(m,n)=(5,40)$ OR $(5,56)$. $W_{64}^8$-B: $(m,n)\neq(5,40)$ OR $(5,56)$.

| twiddle factors | $k=0$ | | $k=1$ | | $k=2$ | |
|---|---|---|---|---|---|---|
| | $\sum_i \eta_i(0)2^{-p_i(0)}$ | $\sum_j \mu_j(0)2^{-q_j(0)}$ | $\sum_i \eta_i(1)2^{-p_i(1)}$ | $\sum_j \mu_j(1)2^{-q_j(1)}$ | $\sum_i \eta_i(2)2^{-p_i(2)}$ | $\sum_j \mu_j(2)2^{-q_j(2)}$ |
| $W_{64}^1$ | $2^0$ | $-2^{-4}$ | $2^0$ | $-2^{-5}$ | $-$ | $-$ |
| $W_{64}^2$ | $2^0$ | $-2^{-2}$ | $2^0-2^{-5}$ | $2^{-4}-2^{-6}$ | $-$ | $-$ |
| $W_{64}^3$ | $2^0$ | $-2^{-2}$ | $2^0$ | $2^{-6}$ | $2^0-2^{-5}$ | $-2^{-4}$ |
| $W_{64}^4$ | $2^0-2^{-6}$ | $2^{-3}$ | $2^0-2^{-3}$ | $-2^{-1}$ | $-$ | $-$ |
| $W_{64}^5$ | $0$ | $-2^0+2^{-4}$ | $2^{-1}$ | $2^0-2^{-4}$ | $-$ | $-$ |
| $W_{64}^6$ | $2^{-1}$ | $2^0-2^{-2}$ | $0$ | $-2^0-2^{-3}+2^{-6}$ | $-$ | $-$ |
| $W_{64}^7$ | $2^{-2}$ | $2^0-2^{-3}$ | $-2^{-1}+2^{-4}$ | $-2^0-2^{-7}$ | $-$ | $-$ |
| $W_{64}^8$-A | $2^0$ | $2^0$ | $0$ | $-2^0+2^{-4}$ | $2^0-2^{-2}$ | $0$ |
| $W_{64}^8$-B | $2^0$ | $2^0$ | $0$ | $-2^0+2^{-4}$ | $2^0-2^{-2}+2^{-8}$ | $0$ |

TABLE VI

COMPARISONS OF THE PROPOSED DESIGNS WITH OTHER DESIGNS IN TERMS OF NUMBER OF ADDERS AND FROBENIUS NORM (FN) IN DB.

| $N$ | radix-2 FFT | | method in [7] | | method in [9] | | FFT 1 (Sec. IV) | | | FFT 2 (Sec. V) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | mult. | add. | add. | FN | add. | FN | add. ($A^t$) | $B^c$ | FN | add. ($A^t$) | $B^c$ | FN |
| 8 | 4 | 52 | 84 | $-64$ | $-$ | $-$ | 64 | 9 | $-73$ | 64 | 9 | $-73$ |
| 16 | 24 | 152 | 252 | $-53$ | $-$ | $-$ | 208 | 9 | $-54$ | 208 | 9 | $-71$ |
| 32 | 88 | 408 | 756 | $-45$ | 616 | $-46$ | 592 | 9 | $-51$ | 592 | 9 | $-65$ |
| 64 | 264 | 1032 | 2094 | $-43$ | $-$ | $-$ | 1552 | 9 | $-49$ | 1552 | 9 | $-61$ |
| 128 | 712 | 2504 | 6727 | $-41$ | 4800 | $-41$ | 3856 | 9 | $-47$ | 3856 | 9 | $-60$ |

error are designed under fixed $B^c$ and large $B^r$, and (38) is computed. In the proposed design, the average number of adders is set to 8. Hence, the total number of adders consumed by FFT, $A^t$, is expressed as

$$A^t = 4N(\log_2 N - 3) + 16 + 2N\log_2 N. \tag{39}$$

The FNs and total numbers of adders of different FFT algorithms are summarized in Table VI. As shown in the table, the proposed design of FFT with MSR-CORDIC shows better accuracy as well as lower hardware complexity than other methods.

Some papers proposed the bit allocation algorithm which allows the register at each FFT stage to have different $B^r$ [8], [28]. Its design objective is to consume fewer bits for internal registers and memory storages while maximizing the system accuracy. However, this issue is beyond the scope of this paper. Interested readers may incorporate the proposed analysis and design with the algorithms introduced in [8], [28].

## VI. Conclusions

In this paper, the fixed-point error analysis of the FFT is presented when MSR-CORDIC is employed for the twiddle factor multiplier. Based on the analysis, total quantization error of the FFT including the approximation error, round-off error and scaling error, is derived in terms of SQNR. Parameter determination algorithms are proposed to maximize the SQNR and to minimize the total number of adders. The proposed method alleviates the impairment of the round-off error as well as approximation error in the course of design. The proposed design is helpful in low-cost and high accuracy design of FFT with the MSR-CORDIC.

## References

[1] A. V. Oppenheim and R. W. Schafer, *Discrete-time Signal Processing*. Englewood Cliffs, NJ: Prentice Hall, 1989.

[2] J. W. Cooley and J. Tukey, “An algorithm for the machine calculation of complex Fourier series,” *Mathmatics of Computation*, vol. 19, no. 90, pp. 297–301, Apr. 1965.

[3] H. Sorensen, M. Heideman, and C. Burrus, “On computing the split-radix FFT,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 1, pp. 152–156, Feb. 1986.

[4] S. G. Johnson and M. Frigo, “A modified split-radix FFT with fewer arithmetic operations,” *IEEE Transactions on Signal Processing*, vol. 55, no. 1, pp. 111–119, Jan. 2007.

[5] A. M. Despain, “Fourier transform computers using CORDIC iterations,” *IEEE Transactions on Computers*, vol. C-23, no. 10, pp. 993–1001, Oct. 1974.

[6] S. Oraintara, Y. J. Chen, and T. Q. Nguyen, “Integer fast Fourier transform,” *IEEE Transactions on Signal Processing*, vol. 50, no. 3, pp. 607–618, Mar. 2002.

[7] S. C. Chan and P. M. Yiu, “An efficient multiplierless approximation of the fast Fourier transform using sum-of-powers-of-two (SOPOT) coefficients,” *IEEE Signal Processing Letters*, vol. 9, no. 10, pp. 322–325, Oct. 2002.

[8] K. M. Tsui and S. C. Chan, “Error analysis and efficient realization of the multiplier-less FFT-like transformation (ML-FFT) and related sinusoidal transformations,” *Journal of VLSI Signal Processing*, vol. 44, no. 1, pp. 97–115, Aug. 2006.

[9] M. D. Macleod, “Multiplierless implementation of rotators and FFTs,” *EURASIP Journal on Applied Signal Processing*, no. 17, pp. 2903–2910, Jan. 2005.

[10] W. H. Chang and T. Nguyen, “Integer FFT with optimized coefficient sets,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’07)*, vol. 2, 2007, pp. 109–112.

[11] A. M. Despain, “Very fast Fourier transform algorithms for hardware implementation,” *IEEE Transactions on Computers*, vol. C-28, no. 5, pp. 333–341, May 1979.

[12] S. Y. Park, N. I. Cho, S. U. Lee, K. Kim, and J. Oh, “Design of 2K/4K/8K-point FFT processor based on CORDIC algorithm in OFDM receiver,” in *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, 2001 (PACRIM’01)*, vol. 2, Aug. 2001, pp. 457–460.

[13] R. Sarmiento, F. Tobajas, V. D. Armas, R. E. Chain, J. F. Lopez, J. A. M. Nelson, and A. Nunez, “A CORDIC processor for FFT computation and its implementation using gallium arsenide technology,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 6, no. 1, pp. 18–30, Mar. 1998.

[14] J. C. Kuo, C. H. Wen, C. H. Lin, and A. Y. Wu, “VLSI design of a variable-length FFT/IFFT processor for OFDM-based communication systems,” *EURASIP Journal on Applied Signal Processing*, no. 13, pp. 1306–1316, Dec. 2003.

[15] T. S. Chan, J. C. Kuo, and A. Y. Wu, “A reduced-complexity fast algorithm for software implementation of the IFFT/FFT in DMT systems,” *EURASIP Journal on Applied Signal Processing*, no. 9, pp. 961–974, Jan. 2002.

[16] J. Volder, “The CORDIC trigonometric computing technique,” *IRE Transactions on Electronic Computers*, vol. EC-8, no. 3, pp. 330–334, Sep. 1959.

[17] J. S. Walther, “A unified algorithm for elementary functions,” in *Spring Joint Computer Conference*, vol. 38, 1971, pp. 379–385.

[18] Y. H. Hu, “CORDIC-based VLSI architectures for digital signal processing,” *IEEE Signal Processing Magazine*, vol. 9, no. 3, pp. 16–35, Jul. 1992.

[19] C. S. Wu and A. Y. Wu, “Modified vector rotational CORDIC (MVR-CORDIC) algorithm and architecture,” *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 48, no. 6, pp. 548–561, Jun. 2001.

[20] C. S. Wu, A. Y. Wu, and C. H. Lin, “A high-performance/low-latency vector rotational CORDIC architecture based on extended elementary angle set and trellis-based searching schemes,” *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 50, no. 9, pp. 589–601, Sep. 2003.

[21] A. Y. Wu and C. S. Wu, “A unified view for vector rotational CORDIC algorithms and architectures based on angle quantization approach,” *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, no. 10, pp. 1442–1456, Oct. 2002.

[22] C. H. Lin and A. Y. Wu, “Mixed-scaling-rotation CORDIC (MSR-CORDIC) algorithm and architecture for high-performance vector rotational DSP applications,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 11, pp. 2385–2396, Nov. 2005.

[23] A. G. Dempster and M. D. Macleod, “Use of minimum-adder multiplier blocks in FIR digital filters,” *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 42, no. 9, pp. 569–577, Sep. 1995.

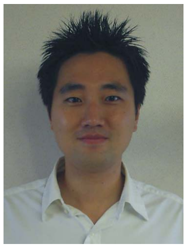
[24] Y. J. Yu and Y. C. Lim, “Design of linear phase FIR filters in subexpression space using mixed integer linear programming,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 10, pp. 2330–2338, Oct. 2007.

[25] S. Y. Park and N. I. Cho, “Design of multiplierless lattice QMF: Structure and algorithm development,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 55, no. 2, pp. 173–177, Feb. 2008.

[26] Y. J. Chen, S. Oraintara, T. D. Tran, K. Amaratunga, and T. Q. Nguyen, “Multiplierless approximation of transforms with adder constraint,” *IEEE Signal Processing Letters*, vol. 9, no. 11, pp. 344–347, Nov. 2002.

[27] S. Y. Park and N. I. Cho, “Fixed-point error analysis of CORDIC processor based on the variance propagation formula,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 3, pp. 573–584, Mar. 2004.

[28] W. H. Chang and T. Nguyen, “On the fixed-point accuracy analysis of FFT algorithms,” *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 4673–4682, Oct. 2008.



**Sang Yoon Park** (S'03-M'11) received the B.S., M.S., and Ph.D. degrees in Department of Electrical Engineering and Computer Science from Seoul National University, Seoul, Korea, in 2000, 2002, and 2006, respectively. He joined the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore as a Research Fellow in 2007. Since 2008, he has been with Institute for Infocomm Research, Singapore, where he is currently a Research Scientist. His research interests include architectures and algorithms for low-power/high-performance digital signal processing and communication systems.

**Ya Jun Yu** (S'99-M'05-SM'09) received both the B.Sc. and M.Eng. degrees in biomedical engineering from Zhejiang University, Hangzhou, China, in 1994 and 1997, respectively, and the Ph.D. degree in electrical and computer engineering from the National University of Singapore, Singapore, in 2004. From 1997 to 1998, she was a Teaching Assistant with Zhejiang University. She joined the Department of Electrical and Computer Engineering, National University of Singapore as a Post Master Fellow in 1998 and remained in the same department as a Research Engineer until 2004. She joined the Temasek Laboratories at Nanyang Technological University as a Research Fellow in 2004. Since 2005, she has been with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, where she is currently an Assistant Professor. Her research interests include digital signal processing and VLSI circuits and systems design. Dr. Yu has served as an Associate Editor for Circuits Systems and Signal Processing and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II: Express Briefs since 2009 and 2010, respectively.