Hybrid Word-Length Optimization Methods of Pipelined FFT Processors

Cheng-Yeh Wang, *Student Member*, *IEEE*, Chih-Bin Kuo, and Jing-Yang Jou, *Fellow*, *IEEE*

Abstract—Quickly and accurately predicting the performance based on the requirements for IP-based system implementations optimizes the design and reduces the design time and overall cost. This study describes a novel hybrid method for the word-length optimization of pipelined FFT processors that is the arithmetic kernel of OFDM-based systems. This methodology utilizes the rapid computing of statistical analysis and the accurate evaluation of simulation-based analysis to investigate a speedy optimization flow. A statistical error model for varying word-lengths of PE stages of an FFT processor was developed to support this optimization flow. Experimental results designate that the word-length optimization employing the speedy flow reduces the percentage of the total area of the FFT processor that increases with an increasing FFT length. Finally, the proposed hybrid method requires a shorter prediction time than the absolute simulation-based method does and achieves more accurate outcomes than a statistical calculation does.

Index Terms—Pipelined FFT processor, signal-to-quantization noise ratio, word-length optimization, statistical analysis, simulation-based analysis, upper-bound word-length, lower-bound word-length.

1 Introduction

The FFT is one of the most widely used algorithms for calculating the Discrete Fourier Transform (DFT) owing to its efficiency in reducing computation time [1]. Recently, the FFT requiring real-time processing has played a significant role in many communication systems based on Orthogonal Frequency Division Multiplexing (OFDM) technology such as high-definition TV (HDTV), xDSL modems, and wideband mobile terminals.

Pipelined FFT implementations are highly appropriate for real-time applications since pipelined FFT can be easily merged with the sequential nature of sampling. Several FFT architectures were developed, such as the Radix-2 Multipath Delay Commutator (R2MDC) [2], Radix-2 Single-path Delay Feedback (R2SDF) [3], $\text{Radix-}2^2$ Single-path Delay Feedback ($\text{R}2^2\text{SDF}$) [4], [5], Radix-4 Single-path Delay Feedback (R4SDF), and Radix-4 Multipath Delay Commutator (R4MDC) [2]. Among these architectures, delay feedback approaches are always more efficient than the corresponding delay commutator approaches in terms of the required memory size [4], [6]. The R4SDF requires fewer multipliers than those required by the R2SDF; however, the R2SDF architecture is simple and regular. The $\text{R}2^2\text{SDF}$ architecture is a compromise endowed with the R2SDF structure and the multiplicative complexity of the R4SDF. This study focuses on the R2SDF and $\text{R}2^2\text{SDF}$ architectures.

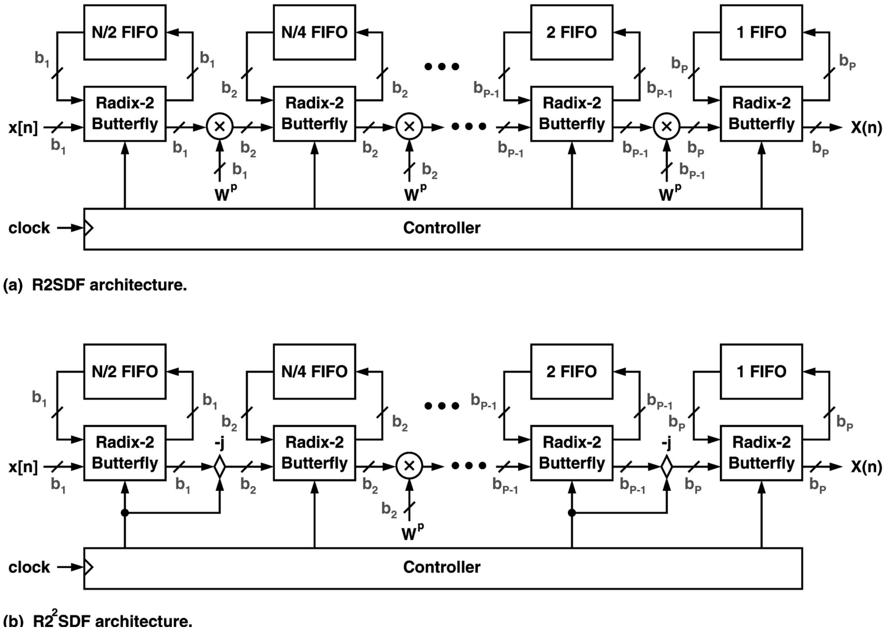
Since the pipeline FFT architecture is memory consuming, reducing its memory requirement will save a significant amount of chip area. Several studies have employed regular module implementations and have attempted to reduce the area-consuming elements in the FFT design. The design in [7] reduces the amount of memory used to store the twiddle factors by employing canonic signed digit (CSD) constant multipliers. A new FFT architecture, the radix-2 single deep delay feedback ($\text{R2SD}^2\text{SF}$) presented in [8], has smaller complex multipliers and adders than other FFT designs. Both of the designs in [7] and [8] have a fixed word-length for data and coefficients for each pipeline stage. The possibility of using varying word-lengths for these stages is frequently ignored when achieving modularized solutions. However, the increasing use of Intellectual Property (IP) makes the nonmodule implementation viable, allowing for the further exploitation of pipelined architectures.

In general, an FFT cannot be implemented exactly. Each multiplier and adder in the pipelined FFT architecture can introduce errors due to the rounding or truncation of arithmetic results. Errors typically accumulate successively over FFT stages. That is, errors from the early stages can affect performance in the later stages. The word-lengths of data and coefficients chiefly affect precision, quantization errors, and hardware complexity. Increased word-lengths increase the precision and reduce the quantization error at the cost of area and power. Conversely, to maintain a lower hardware cost, a shorter word-length can be chosen at the sacrifice of precision. Therefore, identifying an optimized solution of word-length is necessary.

Two conventional methods for the FFT error analysis of the signal-to-quantization-noise ratio (SQNR) and word-lengths are the statistical error analysis and the simulation-based analysis. Although the SQNR can be calculated efficiently by employing statistical models [9], [10], [11], the accuracy of the calculated result heavily depends on the model used. A more precise model yields more accurate results. The simulation-based method evaluates the FFT by comparing simulation results of the fixed-point computations with those obtained

- *The authors are with the Department of Electronics Engineering, National Chiao-Tung University, 1001 Ta Hsueh Road, Hsin-Chu, Taiwan 300, ROC. E-mail: cywang@eda.ee.nctu.edu.tw, back.kuo@gmail.com, jyjou@faculty.nctu.edu.tw.*

Manuscript received 5 May 2006; revised 23 Oct. 2006; accepted 16 Mar. 2007; published online 20 Apr. 2007.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number TC-0172-0506.
Digital Object Identifier no. 10.1109/TC.2007.1059.



(a) R2SDF architecture.

(b) $R2^2SDF$ architecture.

Fig. 1. Conventional R2SDF and $\mathrm{R2^2SDF}$ DIF implementations.

using the floating-point arithmetic [12]. Although simulation increases the accuracy of the evaluation results, it is time consuming.

According to error analysis, optimizing word-lengths of pipeline stages in FFT processors for given specifications is feasible. Optimization of an 8,192-point FFT processor using the simulation method has shown that progressive word-lengths and scaling in the early stages can achieve a good compromise between the SQNR and hardware cost [13]. However, this approach requires a long time to run the simulation.

This work presents a statistical model for error analysis at the stage level with varying word-lengths in the pipeline FFT processor. Furthermore, a hybrid method for reducing the required simulation time is introduced. The optimized word-length (OW) parameters at each stage are generated automatically according to design specifications of FFT processors such as the length of FFT, SQNR, and the real-time processing requirements. Finally, the optimization flow using the proposed error model and the hybrid method is demonstrated.

The rest of this paper is organized as follows: Section 2 gives a brief review of the FFT. Section 3 then introduces statistical and simulation-based error analyses and demonstrates the effectiveness of these methods. Section 4 describes the proposed method for word-length optimization step by step, whereas Section 5 summarizes the experimental results. Conclusions are finally drawn in Section 6.

2 Overview of FFT

An FFT based on structuring the DFT computation by forming increasingly smaller subsequences of the input sequence $x[n]$ is called a decimation-in-time (DIT) FFT. Alternatively, an FFT can also be decomposed using a first-half/second-half approach that divides the output sequence $X(r)$ into increasingly smaller subsequences; this procedure is called a decimation-in-frequency (DIF) FFT [14]. Since both of these schemes are similar in nature, their performance cannot be exactly compared without a given architecture [11]. In this work, the DIF algorithm is used to illustrate the architectural implementations.

This work examines the architectures of R2SDF and $\mathrm{R2^2SDF}$ for the fixed-point DIF pipeline FFT processor to demonstrate the effectiveness of the proposed optimization method. Their block diagrams are shown in Fig. 1, where N is the FFT length, b_k is the word-length of stage k, $k \in \{1, 2, \cdots, P\}$, and $P = \log_2 N$. Due to spatial regularity, both controllers in these architectures can be implemented by using simple P-bit counters [3], [5]. Since the valid output range of the $+/-$ operation of the FFT butterfly is double that of the valid input range, a scaling by 1/2 is applied to eliminate the overflow. Without scaling by 1/2, the overflow will cause excessive error. Therefore, scaling by 1/2 is employed for each stage in this work.

Although the area and power consumption in these pipeline architectures are dominated by memory (FIFOs) and multipliers, progressively adjusting the word-length of pipeline stages can reduce the area of memory and multipliers and, hence, the overall power consumption. To adjust word-lengths based on maintaining the SQNR requirement, an error analysis is required.

3 Error Analysis

3.1 Statistical Analysis

This section introduces the statistical error model for varying word-lengths of pipeline stages. The precision of the FFT processor is then discussed according to the SQNR derivation. This derivation has two major steps. The first step involves finding error sources based on the architectures adopted and the fixed-point arithmetic schemes, for example, truncation and scaling by 1/2 or not. The next step entails searching the paths of error propagation and combining all of these errors along paths to evaluate the variance of the errors propagating to the output. Moreover, the SQNR of the FFT output is given.

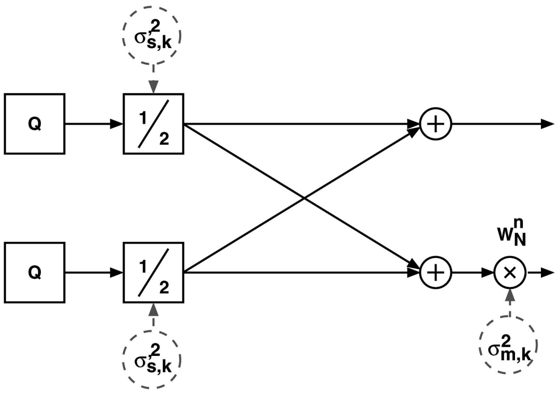


Fig. 2. Error model of a PE stage.

In the following analyses, two architectures, R2SDF and R2^{2}SDF, are illustrated for deriving the statistical error models. Assume a fixed-point arithmetic with (b_k+1)-bit word-lengths and a signed fraction, where k is the stage number of process element (PE) stage. The input to an N-point FFT, denoted by $x[n]$, where $n = 0, 1, 2, \cdots, N-1$, is a sequence of finite-valued complex numbers. The numbers consist of $2N$ real random variables that are uncorrelated and are uniformly distributed in $(-1/\sqrt{2}, 1/\sqrt{2})$. One dB is added to the SQNR constraint (iSQNR) to allow for the SQNR error in the statistical model. The effect of inaccuracy in the twiddle factor W^p is not addressed here. The truncation operations are modeled as uncorrelated.

3.1.1 Error Sources

Four major error sources must be addressed in an FFT processor. The first error source is the quantization error of the word-length difference between PE stages whose variance is $\sigma^2_{q,k}$. The second error occurs during scaling and its variance is represented by $\sigma^2_{s,k}$. Another error results from the complex multiplication of the twiddle factor and $\sigma^2_{m,k}$ is used to denote its variance. The last error, the insufficient output word-length error $\sigma^2_{q,o}$, is only considered for the last stage of the FFT processor.

Fig. 2 shows the error model of a PE stage with a stage-by-stage scaling of $1/2$. $\sigma^2_{q,k}$ occurs when the word-length b_{k-1} of stage $k-1$ is longer than that of stage k. $\sigma^2_{q,k}$ is the variance of the truncated bits from b_{k-1} to b_k. Scaling error is produced when $b_k < b_{k-1}+1$. Scaling by a factor of $1/2$ involves a one-bit right shift and truncation of the last significant bit (LSB). $\sigma^2_{s,k}$ is defined as the variance of this truncated bit. Both $\sigma^2_{q,k}$ and $\sigma^2_{s,k}$ can be combined as an error of directly scaling the output data of stage $k-1$ and truncating the scaled result to b_k bits. Since complex scaling can be achieved by separately scaling the real and imaginary parts of the data, the combined error $\sigma'^2_{s,k}$ (Fig. 2) can be expressed as

$$\sigma'^2_{s,k} = \begin{cases} 0; & b_{k-1}+1 \le b_k \\ 2 \times \left(\frac{1}{M} \cdot 2^{-2(b_{k-1}+1)} \cdot \sum_{v=0}^{M-1} v^2\right); & b_{k-1}+1 > b_k, \end{cases} \tag{1}$$

where $M = 2^{(b_{k-1}+1)-b_k}$.

If a complex multiplication is implemented by using four real multiplications and the results of the real multiplications are truncated individually, $\sigma^2_{m,k}$ is defined as the variance of the truncated bits of the result after finishing a complex multiplication. This variance can be represented by word-lengths b_{k-1} and b_k and is obtained as

$$\sigma^2_{m,k} = \begin{cases} 4 \times \left(\frac{1}{M} \cdot 2^{-2(b_{k-1}+1+b_k)} \cdot \sum_{v=0}^{M-1} v^2\right), \\ \quad M = 2^{(b_{k-1}+1+b_k)-b_k}; \quad b_{k-1}+1 \le b_k \\ 4 \times \left(\frac{1}{M} \cdot 2^{-2\cdot 2b_k} \cdot \sum_{v=0}^{M-1} v^2\right), \\ \quad M = 2^{2b_k-b_k}; \quad b_{k-1}+1 > b_k. \end{cases} \tag{2}$$

If the required output word-length b_o of the FFT processor is too short, the output of the last PE stage must be truncated. The quantization error is then generated and its variance $\sigma^2_{q,o}$ can be described by b_o and the word-length of the last stage, b_P; the formula is expressed as

$$\sigma^2_{q,o} = \begin{cases} 0; & b_P \le b_o \\ 2 \times \left(\frac{1}{M} \cdot 2^{-2b_P} \cdot \sum_{v=0}^{M-1} v^2\right); & b_P > b_o, \end{cases} \tag{3}$$

where $M = 2^{b_P - b_o}$.

3.1.2 Output SQNR

Since all error sources are assumed uncorrelated, the variance of the errors at the FFT output can be obtained by summing all contributions from the individual error sources that propagate to the output. For an N-point FFT processor employing either the Radix-2 algorithm or Radix-2^2 algorithm, the scaling error $\sigma'^2_{s,k}$ propagating to any output node can be given by

$$\sigma^2_S \approx N\left(\frac{1}{4}\right)^{P-1} \times \sigma'^2_{s,1} + \frac{N}{2}\left(\frac{1}{4}\right)^{P-2} \times \sigma'^2_{s,2} + \cdots + \frac{N}{2^{P-1}}\left(\frac{1}{4}\right)^{0} \times \sigma'^2_{s,P} \tag{4}$$

$$= \sum_{k=1}^{P} \frac{N}{2^{k-1}} \cdot \left(\frac{1}{4}\right)^{P-k} \times \sigma'^2_{s,k}, \tag{5}$$

where $(1/4)^{P-k}$ is the effect of scaling on the error propagating at stage k. The $\sigma'^2_{s,k}$ propagation can be illustrated using an 8-point DIF Radix-2 algorithm signal flow graph (SFG) (Fig. 3). The number of scaling errors $\sigma'^2_{s,k}$ propagating to any output node, for example, $X(0)$, from the first, second, and third stages are 8, 4, and 2, respectively. Thus, the error variance of $X(0)$ can be obtained from (4) and is given by $\sigma^2_{S|X(0)} = (1/2)\sigma'^2_{s,1} + \sigma'^2_{s,2} + 2\sigma'^2_{s,3}$.

To derive the variance of the FFT output due to multiplication errors, all multiplications are assumed noisy. Fig. 4 shows an 8-point DIF Radix-2 algorithm SFG. There are four, that is, half of eight, $\sigma^2_{m,k}$s in each stage and each $\sigma^2_{m,k}$ of the first $(k=1)$, second $(k=2)$, and last $(k=3)$ stages will propagate to 4, 2, and 1 output nodes, respectively. On the other hand, for a general case with an N-point FFT, there are half of the N $\sigma^2_{m,k}$s error sources in each stage and each $\sigma^2_{m,k}$ from the first stage to the last, Pth, stage propagates to $\frac{N}{2}, \frac{N}{4}, \cdots, \frac{N}{2^P}$ output data, respectively. Hence, one can easily derive the output variance caused by multiplication errors, σ^2_M, with the Radix-2 algorithm; the expression of σ^2_M is given by

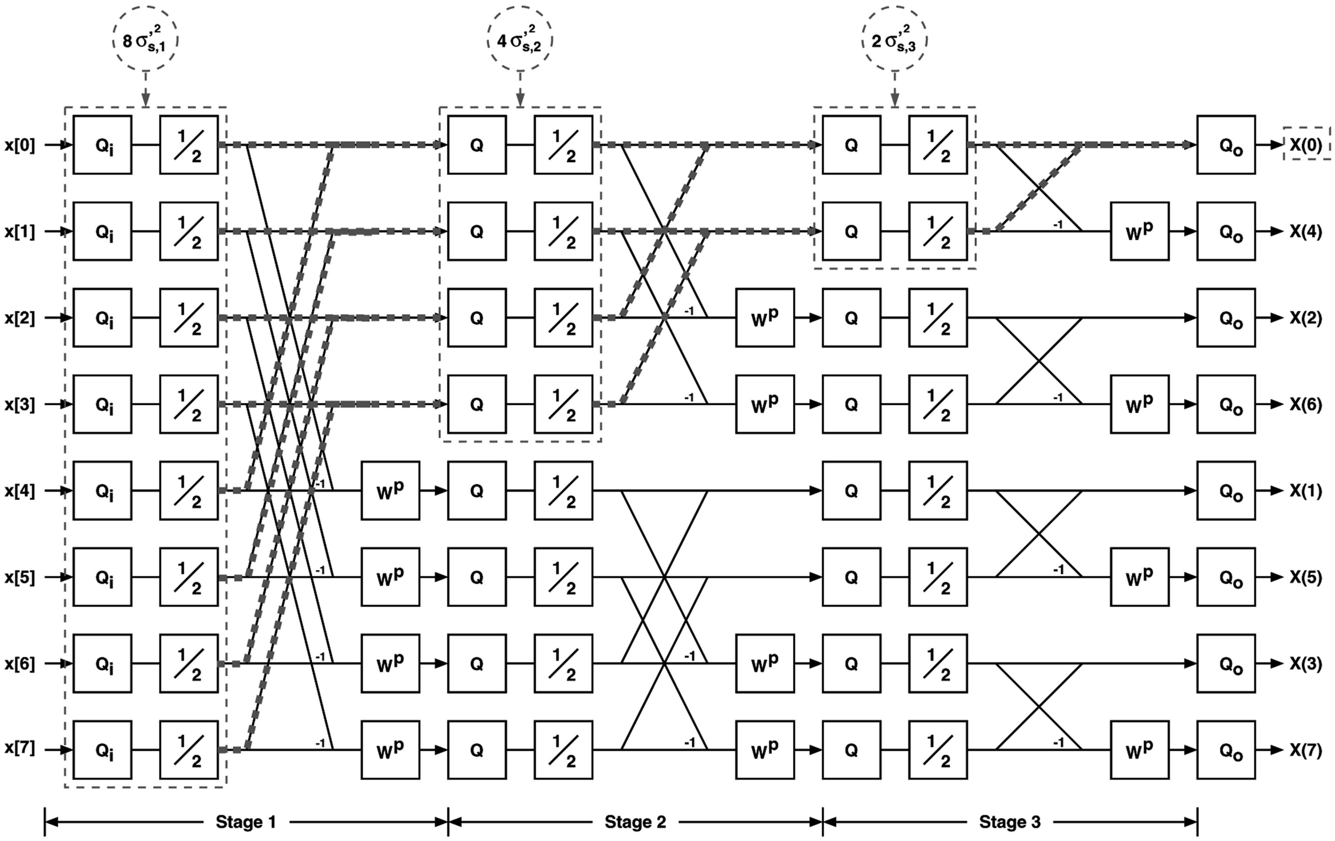


Fig. 3. Propagation of quantization and scaling errors.

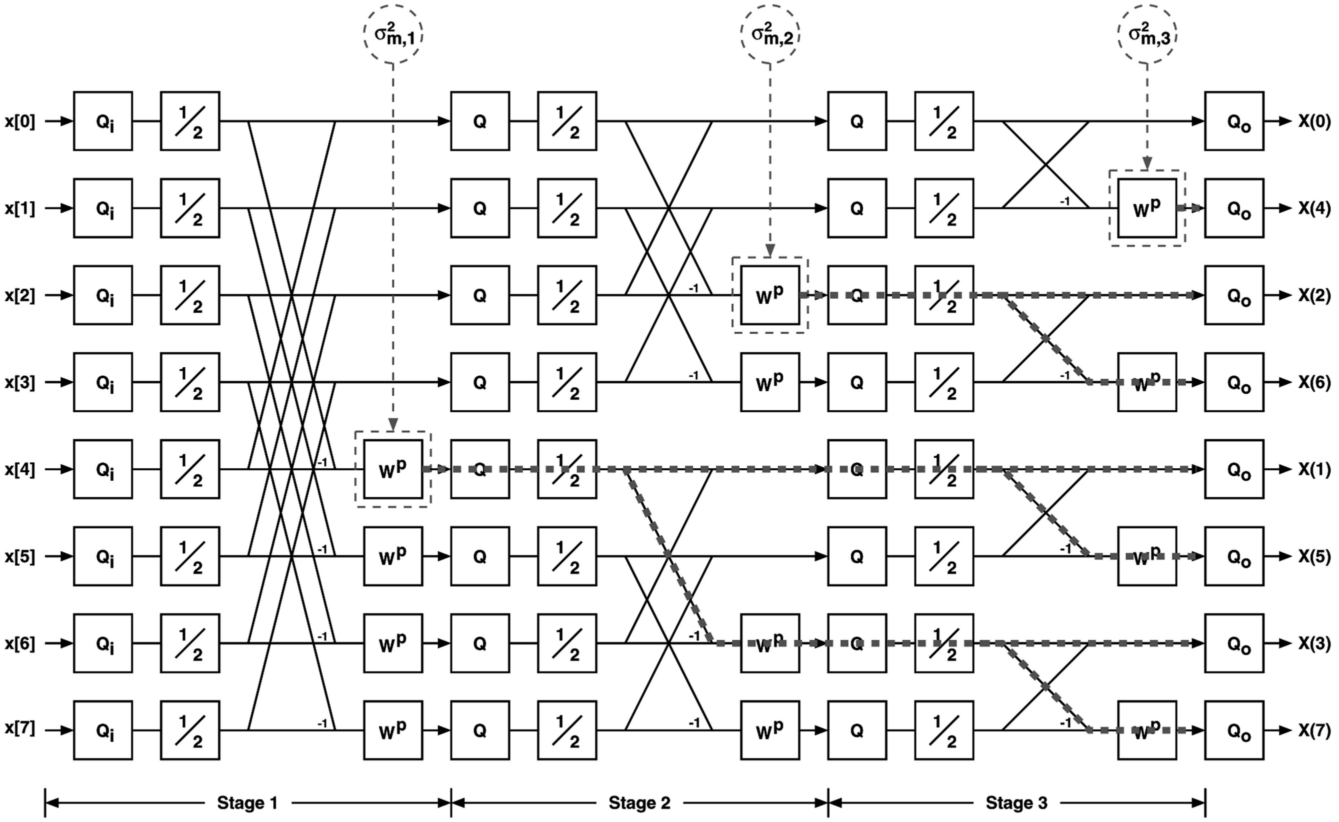


Fig. 4. Propagation of multiplication errors.

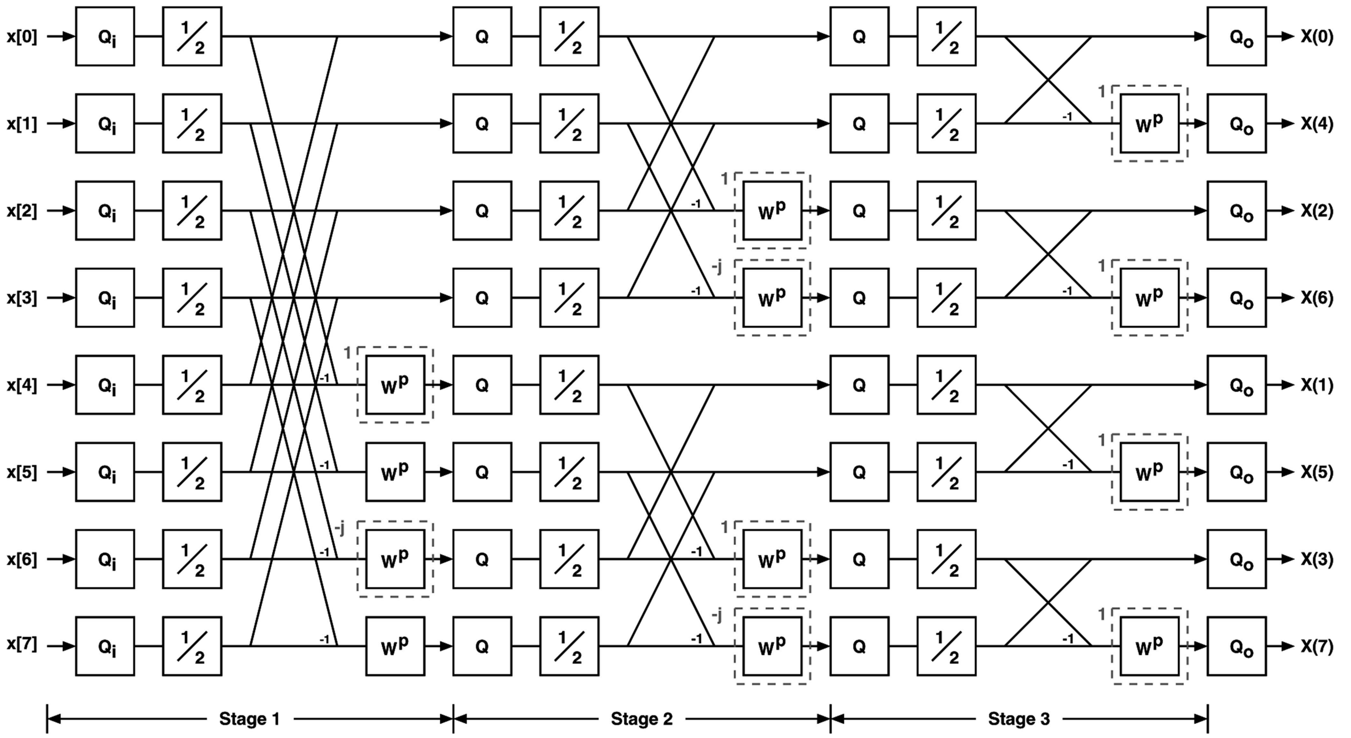


Fig. 5. Propagation of noiseless multiplications.

$$\sigma_M^2 \approx \frac{1}{N}\cdot\frac{N}{2}\left[\frac{N}{2}\left(\frac{1}{4}\right)^{P-1}\cdot\sigma_{m,1}^2+\frac{N}{2^2}\left(\frac{1}{4}\right)^{P-2}\cdot\sigma_{m,2}^2+ \cdots+\frac{N}{2^P}\left(\frac{1}{4}\right)^{0}\cdot\sigma_{m,P}^2\right] \tag{6}$$

$$=\frac{1}{2}\times\sum_{k=1}^{P}\frac{N}{2^k}\cdot\left(\frac{1}{4}\right)^{P-k}\times\sigma_{m,k}^2. \tag{7}$$

For the $\mathrm{Radix}\text{-}2^2$ algorithm, the corresponding expression of σ_M^2 is modified as

$$\sigma_M^2 \approx \frac{1}{N}\cdot\frac{3N}{4}\left[\frac{N}{4}\left(\frac{1}{4}\right)^{P-2}\sigma_{m,2}^2+\frac{N}{4^2}\left(\frac{1}{4}\right)^{P-4}\sigma_{m,4}^2+ \cdots+\frac{N}{4^{P/2}}\left(\frac{1}{4}\right)^{0}\sigma_{m,P}^2\right] \tag{8}$$

$$=\frac{3}{4}\times\sum_{k=1}^{P/2}\frac{N}{4^k}\cdot\left(\frac{1}{4}\right)^{P-2k}\times\sigma_{m,2k}^2. \tag{9}$$

Equations (6) and (8) are derived by assuming that all of the multiplications are noisy. However, the multiplications associated with twiddle factors $W^p=\pm 1$ or $W^p=\pm j$ introduce no errors. Fig. 5 shows the position of noiseless multiplications in an 8-point Radix-2 algorithm SFG. The variances of these noiseless multiplications, denoted as σ_C^2 in the Radix-2 algorithm SFG and the $\mathrm{Radix}\text{-}2^2$ algorithm SFG, are individually rederived and can be given by (10) and (12):

$$\sigma_C^2 \approx \frac{1}{N}\times\left[2\cdot\frac{N}{2}\left(\frac{1}{4}\right)^{P-1}\cdot\sigma_{m,1}^2+2^2\cdot\frac{N}{2^2}\left(\frac{1}{4}\right)^{P-2}\cdot\sigma_{m,2}^2 +\cdots+2^{P-1}\cdot\frac{N}{2^{P-1}}\left(\frac{1}{4}\right)\cdot\sigma_{m,P-1}^2\right] +\frac{1}{N}\times\left[\frac{N}{2}\times\frac{N}{2^P}\cdot\sigma_{m,P}^2\right] \tag{10}$$

$$=\left[\sum_{k=1}^{P-1}\left(\frac{1}{4}\right)^{P-k}\cdot\sigma_{m,k}^2\right]+\frac{1}{2}\times\frac{N}{2^P}\cdot\sigma_{m,P}^2, \tag{11}$$

$$\sigma_C^2 \approx \frac{1}{N}\cdot\frac{3N}{4}\times\left[\frac{N}{4}\cdot\left(\frac{1}{2^{P-2}}\right)\cdot\left(\frac{1}{4}\right)^{P-2}\sigma_{m,2}^2 +\frac{N}{4^2}\cdot\left(\frac{1}{2^{P-4}}\right)\cdot\left(\frac{1}{4}\right)^{P-4}\sigma_{m,4}^2+\cdots+\frac{N}{4^{P/2}}\cdot\sigma_{m,P}^2\right] \tag{12}$$
$$=\frac{3}{4}\times\sum_{k=1}^{P/2}\frac{N}{4^k}\cdot\left(\frac{1}{2^{P-2k}}\right)\times\left(\frac{1}{4}\right)^{P-2k}\cdot\sigma_{m,2k}^2.$$

According to the assumption of no correlations in these error sources, the total output error variance can be obtained by summing the variance of each error propagating to the output; this summation is expressed as

$$\sigma_T^2=\sigma_S^2+\left(\sigma_M^2-\sigma_C^2\right)+\sigma_{qo}^2. \tag{13}$$

Furthermore, the variance of output data in an N-point FFT processor is given in (14) [14]:

$$\sigma_X^2=\frac{1}{3N}. \tag{14}$$

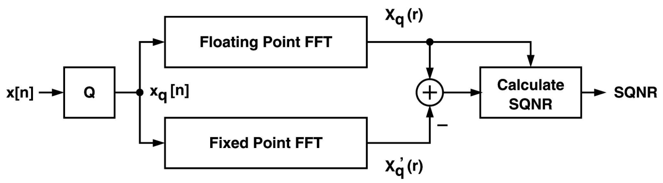


Fig. 6. Block diagram of the simulation analysis.

Hence, the output SQNR is obtained by

$$SQNR_1 = 10\log_{10}\left(\frac{\sigma_X^2}{\sigma_T^2}\right). \tag{15}$$

This $SQNR_1$ model is used as the performance index, whereas the statistical error analysis is employed in the FFT processor.

3.2 Simulation-Based Method

Fig. 6 presents a conceptual block diagram of the simulation-based analysis. To perform this simulation, floating-point and fixed-point C models with a given FFT algorithm were developed. According to system constraints, for example, the word-length of each stage, the rounding or truncation of stages, the number of scaling stages, and the input/output word-length, the C models obtain the proper fixed-point results. Then, SQNR can be evaluated by comparing the fixed-point output with the floating-point output; the formula of the calculation is given by

$$SQNR_2 = 10\log_{10}\frac{\sum_{r=0}^{N-1}\left[X_q(r)\right]^2}{\sum_{r=0}^{N-1}\left[X_q(r) - X'_q(r)\right]^2}. \tag{16}$$

During simulation, random patterns are generated as inputs and, then, the resulting $SQNR_2$s are averaged as an estimated average of the SQNR distribution. During simulation analysis, there is a trade-off between the accuracy of the $\overline{SQNR_2}$ and the required number of simulation times. The required simulation times are investigated as follows: First, the random variable $\bar{S}$ is used as an estimate of $\overline{SQNR_2}$. Then, according to the central limit theorem, the sampling distribution of $\bar{S}$ is approximately normally distributed. Therefore, we can be $(1-\alpha)100$ percent confident that the SQNR error will not exceed a specified amount e when the number of $SQNR_2$s equals

$$\left(\frac{z_{\alpha/2}\cdot\sigma}{e}\right)^2, \tag{17}$$

where σ is the standard deviation of the distribution of $SQNR_2$ and $z_{\alpha/2}$ satisfies the probability equation $\mathrm{Prob}(z_{\alpha/2} < Z) = \alpha/2$ when Z is a random variable with a standard normal distribution [16]. In this work, e is the constraint of the SQNR error (SQNR_Error).

3.3 Demonstration of the Statistical and Simulation-Based Analysis

The proposed error model was verified using the simulation-based error analysis and the result was compared with that obtained by the statistical error analysis. In this analysis, the SQNR is calculated using the statistical method and is also evaluated using a simulation for 8, 16, $\cdots$, 8,192-point DIF Radix-2 FFT and for 16, 64, $\cdots$, 4,096-point DIF $\mathrm{Radix}\text{-}2^2$ FFT with the freely chosen word-length from 8-32 bits for each stage.

Table 1 shows a summary of the comparison between the two methods with 20 randomly generated word-length sets in a 1,024-point DIF Radix-2 FFT, where the input word-length is set to be equal to that of the first stage and the output word-length is set to be the same as that of the last stage. The SQNR difference is obtained by subtracting the SQNR of the simulation analysis from that of the statistical analysis.

Fig. 7 shows the histogram of the SQNR difference with 10^4 randomly generated word-length sets for the 1,024-point FFT of Radix-2 and $\mathrm{Radix}\text{-}2^2$ algorithm. The difference in comparison is within ± 1.0 dB in Radix-2 FFT and within ± 1.1 dB for the $\mathrm{Radix}\text{-}2^2$ FFT.

Exhaustively comparing all word-length sets of 8-32 bits is impractical because the simulation time is unendurable. Therefore, partial exhaustive verification for word-lengths

TABLE 1
Example of Random Verification

Wordlength	Wordlength of PE Stage										SQNR (dB)		
Set no.	1	2	3	4	5	6	7	8	9	10	Simulation	Statistical	Difference
1	29	32	22	32	30	21	10	10	20	18	21.57151	21.225869	-0.345643
2	17	26	21	32	23	12	31	21	26	23	40.64704	40.568971	-0.078069
3	8	28	14	16	13	32	29	32	10	11	20.73324	20.836938	0.103696
4	9	12	11	11	12	12	16	16	10	22	20.55134	20.906414	0.355071
5	19	30	27	15	22	19	16	21	31	18	58.45886	59.02069	0.561833
6	29	15	23	25	13	32	17	14	11	8	5.981925	5.987078	0.005153
7	17	22	16	26	30	23	15	31	18	30	55.56245	55.547552	-0.014897
8	29	23	23	30	10	22	16	31	18	29	31.52884	31.443187	-0.085655
9	23	16	15	31	28	28	24	8	20	25	11.22512	11.082254	-0.142871
10	29	20	21	23	32	17	14	8	21	20	11.20543	11.081993	-0.123438
11	11	21	22	22	12	15	25	16	13	21	36.91202	37.570486	0.658464
12	28	13	13	24	12	27	12	10	30	14	22.67044	22.880391	0.209947
13	20	17	14	22	15	8	11	15	11	21	16.35159	16.105621	-0.24597
14	20	21	21	16	12	11	29	32	17	32	33.87208	34.053531	0.181455
15	27	20	21	10	19	13	29	25	18	9	12.01716	12.014605	-0.002551
16	32	15	25	24	8	8	9	8	21	11	9.187271	9.280194	0.092923
17	26	23	12	11	22	29	13	30	26	12	29.20325	29.517037	0.313784
18	20	29	16	28	31	13	25	20	22	14	40.35284	40.808381	0.455539
19	22	18	9	17	23	20	30	25	8	16	8.961997	9.005713	0.043716
20	11	25	27	19	24	14	8	29	31	9	9.633909	9.774141	0.140233

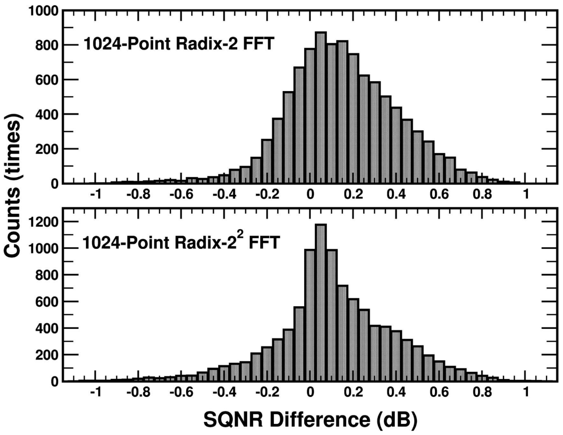


Fig. 7. Histogram of the SQNR difference with randomly generated word-lengths.

of 11-18 bits was employed in the comparison of the 64-point Radix-2 and Radix-2^2 FFT. This comparison required 130 hours. Fig. 8 shows the results of the comparison; the SQNR difference is within ± 1.1 dB.

Both Figs. 7 and 8 present a bias shift in the SQNR difference. This shift is produced because the noise model of multipliers in the statistical analysis is an approximation of the actual noise distribution of multipliers. However, this is not an important issue as the shift is much smaller than the maximum SQNR difference. On the other hand, a parameter Δ is introduced to indicate the maximum SQNR difference for the optimization process. The amount of SQNR difference is not analytically expressed and is obtained by an experiment. Thus, according to experimental results, the value of Δ is suggested to be 1 dB for the R2SDF architecture and 1.1 dB for the R2^{2}SDF architecture when the statistical analysis is mixed with the simulation-based analysis.

4 Word-Length Optimization

Fig. 9 presents the flow of the proposed automatic word-length optimization in the pipelined FFT processor. There are four major steps in the process. First, the upper bound word-length (UBW) for each PE stage is evaluated based on the operating frequency requirement and the SQNR constraint (iSQNR) of the processor. Next, the UBWs of stages are fed into the lower bound word-length (LBW) evaluation as an additional constraint for determining the LBWs. Both the UBW and LBW evaluations employ the statistical analysis. Then, we use the statistical analysis to determine optimized word-length candidates (OWCs) based on iSQNR-Δ. Finally, the OW evaluation is performed based on the two primary procedures: 1) If the SQNR_Error is ≤ 1 dB, a simulation analysis is used to select a solution with the smallest area. As the candidates are arranged in ascending order in area, the algorithm terminates after finding the first solution. 2) If the SQNR_Error is > 1 dB, a benefit function is introduced and the best benefit function is selected.

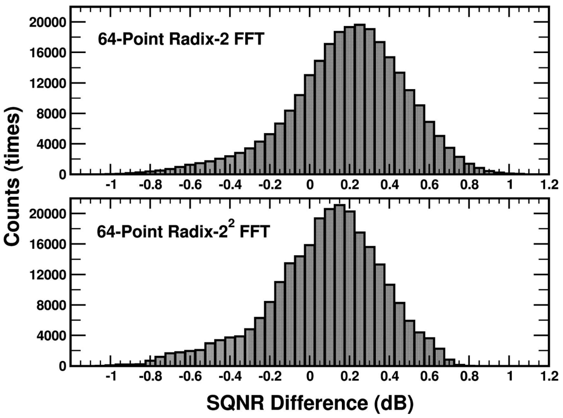


Fig. 8. Histogram of the SQNR difference with partial exhaustive verification.

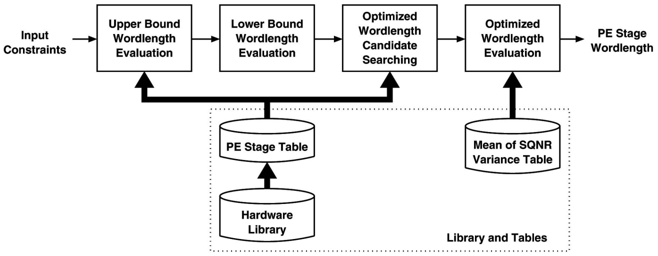

Input Constraints
Upper Bound Wordlength Evaluation
Lower Bound Wordlength Evaluation
Optimized Wordlength Candidate Searching
Optimized Wordlength Evaluation
PE Stage Wordlength
PE Stage Table
Mean of SQNR Variance Table
Hardware Library
Library and Tables

Fig. 9. Word-length optimization flow of a PE stage.

A hardware library and two tables, a PE stage table and a mean of SQNR variance table, are prepared prior to activating the optimizing process. To optimize the hardware cost, one hardware library such as the TSMC $0.25\mu m$ cell library is chosen to determine the area size and critical timing delay of a PE stage in the FFT processor by synthesizing versus different word-engths. The obtained data are recorded in the PE stage table to speed up automation. The mean of the SQNR variance table is used to derive lengths of simulation at different simulated confidences according to (17). This table is established by calculating the mean of 100 simulated SQNR variances of a PE stage with word-lengths of 8-32 bits versus distinct FFT lengths (N).

4.1 Evaluation of the Upper Bound Word-Length

The UBW of the kth PE stage, named $b_{U,k}$, is defined as the maximum possible word-length such that the critical path satisfies the timing constraint, which is the inverse of the operating frequency of the FFT processor. Since the upper bound is obtained based on the operating frequency and throughput, the lower throughput constraint and faster hardware library exactly increase the UBW. Conversely, the increased word-length will require increased time for the operation of the PE stage. That is, increasing the word-length reduces the allowed operating frequency and, thus, the operating frequency requirement can be violated. Additionally, a short word-length results in a poor SQNR.

Fig. 10 shows the process of UBW evaluation. First, the UBW corresponding to the operating frequency requirement is evaluated stage by stage. When the operating frequency requirement is achieved for each stage, $\{b_{U,k}\}$s are obtained. Otherwise, the maximum allowable operating frequency is reported. When all $\{b_{U,k}\}$s are given, they are

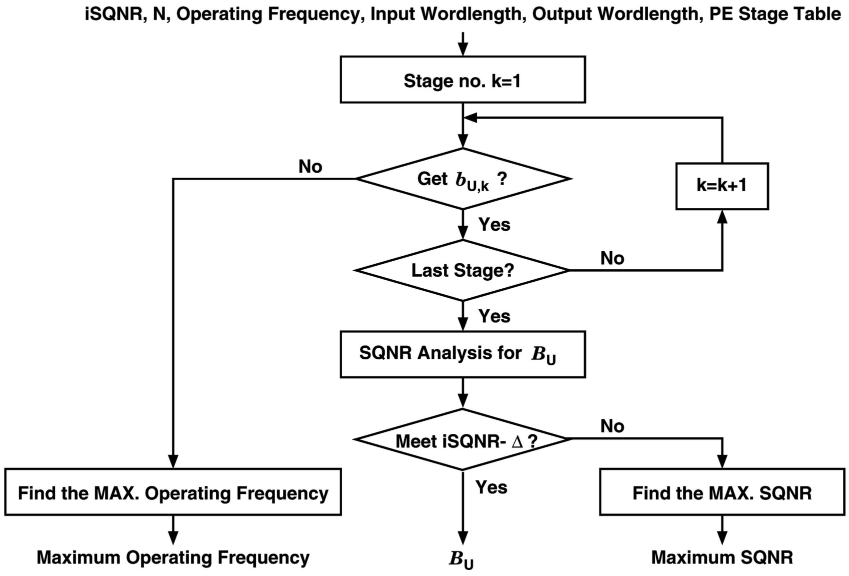

iSQNR, N, Operating Frequency, Input Wordlength, Output Wordlength, PE Stage Table
Stage no. k=1
Get $b_{U,k}$?
No
Yes
k=k+1
Last Stage?
No
Yes
SQNR Analysis for B_U
Meet iSQNR- Δ ?
No
Yes
Find the MAX. Operating Frequency
Find the MAX. SQNR
Maximum Operating Frequency
B_U
Maximum SQNR

Fig. 10. Evaluation of the upper bound word-length.

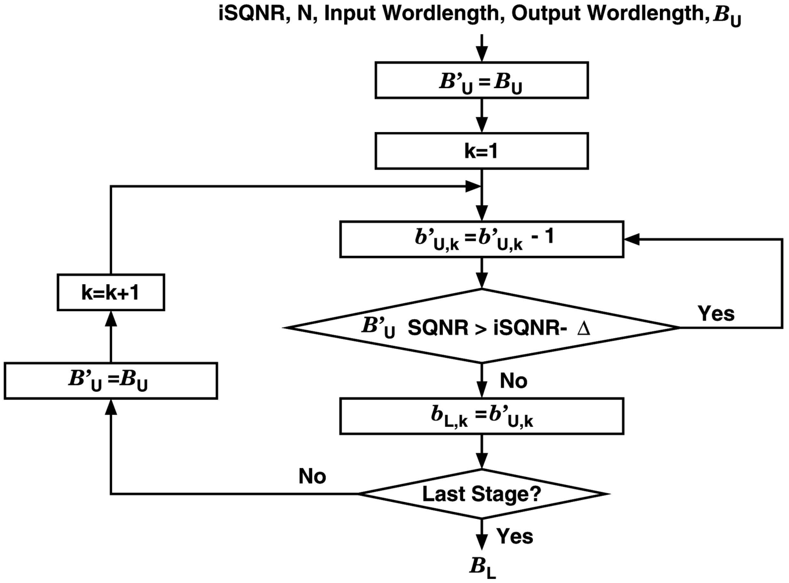


Fig. 11. Evaluation of the lower bound word-length.

used to analyze the SQNR of the FFT processor. If the evaluated SQNR meets the SQNR constraint, the UBW set denoted by B_U comprising these $\{b_{U,k}\}$s is output or the maximum achievable SQNR is reported.

4.2 Evaluation of the Lower Bound Word-Length

When the UBW set is obtained, it is used to support the evaluation of the LBW. The LBW set B_L is derived from B_U such that the optimized solution must be above B_L. That is, if the solution is not above B_L, the solution will not meet the SQNR constraint.

Fig. 11 shows the flow of the LBW evaluation. First, the UBW set B_U is assigned as the initial set B'_U. The evaluation is then conducted stage by stage. For the $k\mathrm{th}$ stage, the element of B'_U, $b'_{U,k}$, progressively decreases until the SQNR of its corresponding set is smaller than the SQNR constraint; this $b'_{U,k}$ is now stored and assigned as $b_{L,k}$. Notably, at the beginning of the evaluation of each stage, B'_U is reset as B_U. When the evaluation of the last stage is complete, the final LBW set B_L, consisting of $\{b_{L,k}\}$s, is reported and this LBW evaluation is terminated.

4.3 Optimized Word-Length Search

There are two chief considerations when searching the candidates of OW. First, since the DIF FFT processor requires a large amount of memory, especially in the early stages, it is better to keep the word-length of these stages short for area optimization. This is demonstrated in Fig. 12, which shows that the area increment of each PE stage in an 8,192-point FFT of Radix-2 when a word-length with 20 bits of each stage is increased by 1 bit.

The second issue is the output SQNR. The output SQNR of a pipelined FFT processor can be obtained by replacing these variances in (15) with the corresponding values obtained from (1) to (14) and rearranged. The final expression is obtained as

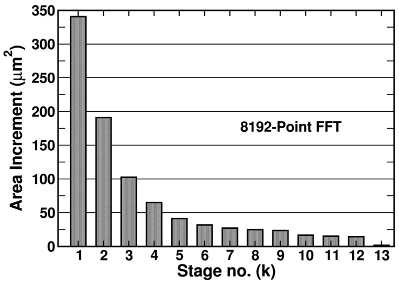


Fig. 12. Area increment of each PE stage as the word-length increases by 1 bit.

$$SQNR \approx 10\log_{10}\frac{c}{\left(a_1 2^{-2b_1+1} + a_2 2^{-2b_2+2} + \cdots + a_P 2^{-2b_P+P}\right)}, \tag{18}$$

where c is a constant, a_k is a constant for each PE stage, and $k \in \{1, 2, \cdots, P\}$. If there exists an mth PE stage such that $a_m 2^{-2b_m+m} >> a_k 2^{-2b_k+k}$, $\forall k$ and $k \neq m$, this stage will be a bottleneck degrading the output SQNR. That is, the value of $(a_1 2^{-2b_1+1} + a_2 2^{-2b_2+2} + \cdots + a_P 2^{-2b_P+P})$ is dominated by the term $a_m 2^{-2b_m+m}$. Thus, it is efficient to choose the sizes of the word-length to be close so that each stage has an approximate quantity of errors.

In addition to the consideration already mentioned, there are two major properties of the R2SDF and R2^{2}SDF architectures that must be considered when determining the format of the OW. The required FIFO memories decrease stage by stage (Fig. 1) and the errors degrading SQNR increase due to the amount of $a_k \cdot 2^k$ in each product term of (18) increasing along as k increases. According to these properties and considerations, the optimization format is in ascending order. This optimization format is true only for the architectures with the characteristics that the required memories decrease stage by stage. For other implementations, the optimization format needs to be modified and, then, the flow for optimization described in the following can be employed.

An OWC B_{OWC} of a pipelined FFT processor has three properties: 1) $B_L \leq B_{OWC} \leq B_U$, 2) B_{OWC} is in ascending order, and 3) the SQNR of the FFT processor with this word-length set meets the SQNR constraint. To find the OWC vector, all word-length sets from the LBW set B_L to the UBW set B_U are scanned and those sets that are in ascending order survive. These sets are then evaluated using the statistical analysis. If the evaluated SQNR meets the constraint, its corresponding set is chosen as an OWC and is stored in the OWC vector sorted by area. This process is iteratively performed until the word-length set reaches B_U and its evaluation is completed. Fig. 13 shows this search flow. The final output of the flow is the OWC vector consisting of B_{OWC} sets sorted by area size.

Fig. 14 presents the flow to determine the OW set, B_{OW}. First, choosing which method to employ is according to the constraint of the SQNR error. When the simulation-based analysis is used, the B_{OWC}s in the OWC vector are simulated starting with the B_{OWC} with the smallest area until the SQNR of the simulated B_{OWC} meets the SQNR constraint, iSQNR. The entire process of optimization is named a hybrid method, whereas the analysis employed is the statistical method before finding the B_{OW}.

When the SQNR_Error is > 1 dB, the whole optimized process using the statistical technique is called the pure statistical method. In this method, B_{OW} is determined based on the benefit function that is defined as

$$Benefit = \frac{\Delta_{SQNR}}{\Delta_{Area}}, \tag{19}$$

where Δ_{SQNR} is the difference in SQNR between the B_L and the analyzed B_{OWC} and Δ_{Area} is their area difference. The B_{OWC} with the best benefit is the OW B_{OW}.

As the statistical analyzer is less accurate due to the error in the analytical model, $Benefit$ is therefore introduced to

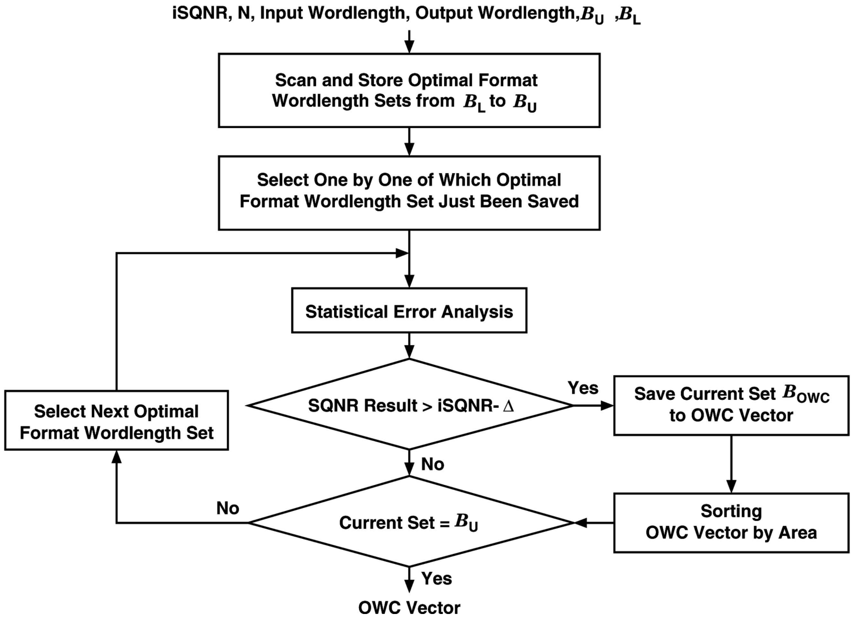

iSQNR, N, Input Wordlength, Output Wordlength,B_U ,B_L
Scan and Store Optimal Format Wordlength Sets from B_L to B_U
Select One by One of Which Optimal Format Wordlength Set Just Been Saved
Statistical Error Analysis
SQNR Result > iSQNR- Δ
Yes
Save Current Set B_{OWC} to OWC Vector
Select Next Optimal Format Wordlength Set
No
Sorting OWC Vector by Area
No
Current Set = B_U
Yes
OWC Vector

Fig. 13. The procedure to determine the OW set candidates.

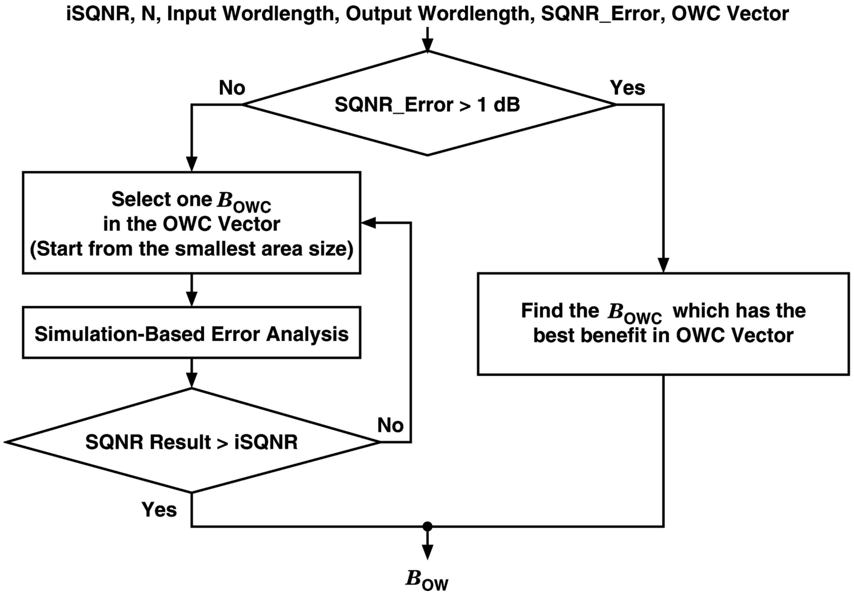
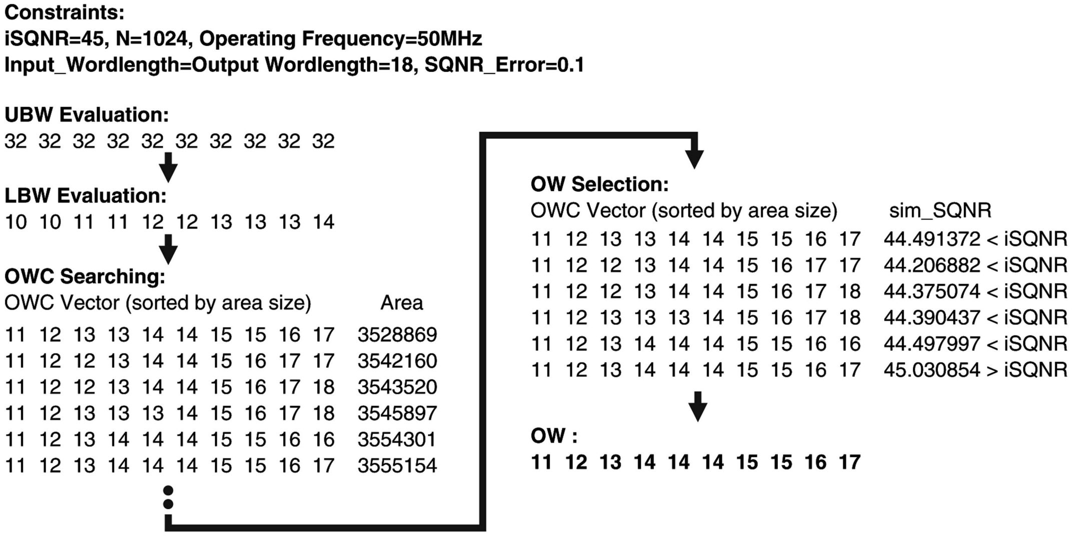

iSQNR, N, Input Wordlength, Output Wordlength, SQNR_Error, OWC Vector
SQNR_Error > 1 dB
No
Yes
Select one B_{OWC} in the OWC Vector (Start from the smallest area size)
Simulation-Based Error Analysis
SQNR Result > iSQNR
No
Yes
Find the B_{OWC} which has the best benefit in OWC Vector
B_{OW}

Fig. 14. Optimized word-length selection.


Constraints:
iSQNR=45, N=1024, Operating Frequency=50MHz
Input_Wordlength=Output Wordlength=18, SQNR_Error=0.1

UBW Evaluation:
32 32 32 32 32 32 32 32 32 32

LBW Evaluation:
10 10 11 11 12 12 13 13 13 14

OWC Searching:

OWC Vector (sorted by area size)	Area
11 12 13 13 14 14 15 15 16 17	3528869
11 12 12 13 14 14 15 16 17 17	3542160
11 12 12 13 14 14 15 16 17 18	3543520
11 12 13 13 13 14 15 16 17 18	3545897
11 12 13 14 14 14 15 15 16 16	3554301
11 12 13 14 14 14 15 15 16 17	3555154

OW Selection:

OWC Vector (sorted by area size)	sim_SQNR
11 12 13 13 14 14 15 15 16 17	44.491372 < iSQNR
11 12 12 13 14 14 15 16 17 17	44.206882 < iSQNR
11 12 12 13 14 14 15 16 17 18	44.375074 < iSQNR
11 12 13 13 13 14 15 16 17 18	44.390437 < iSQNR
11 12 13 14 14 14 15 15 16 16	44.497997 < iSQNR
11 12 13 14 14 14 15 15 16 17	45.030854 > iSQNR

OW :
11 12 13 14 14 14 15 15 16 17

Fig. 15. An example of hybrid word-length optimization.

finetune the solution to get an enhanced SQNR with little area overhead. In the trade-off between SQNR and area, $Benefit$ is a better cost function than the area cost function.

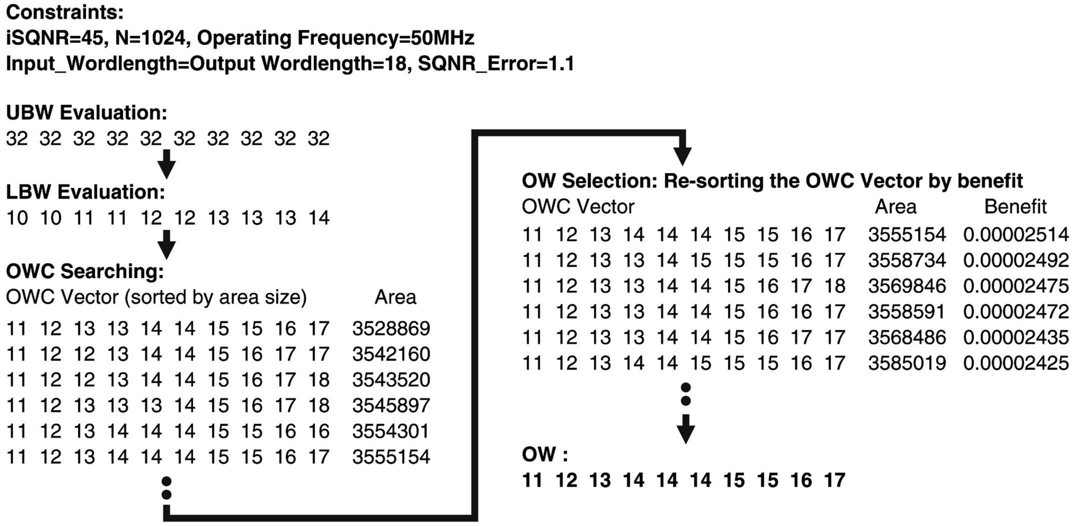
Fig. 15 shows an example of the steps in the word-length optimization by using the hybrid method when the SQNR error constraint is less than 1 dB. The system constraints are the same as those in Table 3. The sim_SQNR means the result of simulation and iSQNR is the SQNR constraint. After sorting the OWC vector with ascending area size, the B_{OW} selection is performed one by one until the sim_SQNR is greater than iSQNR and, then, the outcome of B_{OW} is obtained.

Fig. 16 shows the example of optimization employing the pure statistical method. Table 3 shows the system constraints, except for $\text{SQNR\_Error} = 1.1$ dB. Because the constraint of the SQNR error is > 1 dB, the pure statistical method is employed to identify B_{OW}. The B_{OWC}s in the OWC vector are resorted based on the benefit that is calculated according to (19). The B_{OWC} with the best benefit is B_{OW}.

To determine the OWC vector, it requires a full scan of the sets with the ascending order and it therefore has the complexity of $O(2^{P+\max\{b_{U,k}\}})$. To perform the OW evaluation, a greedy algorithm is used if the SQNR_Error is ≤ 1 dB. If the SQNR_Error is > 1 dB, the evaluation requires calculating the benefit function of the OWC vector. Hence, it has the same complexity as the determination of the OWC vector.

5 Experimental Results

In this work, FFT architectures DIF R2SDF and DIF R2^{2}SDF were implemented by using the C++ language with the SystemC [17], [18] library to demonstrate the proposed flow for optimization. The FFT length N can be adjusted from 8 to 8,192 and the word-length of each stage can be altered from 8 to 32 bits. Since the SQNR range is 40-60 dB and the specifications (Table 2) are



Constraints:
iSQNR=45, N=1024, Operating Frequency=50MHz
Input_Wordlength=Output Wordlength=18, SQNR_Error=1.1

UBW Evaluation:
32 32 32 32 32 32 32 32 32 32

LBW Evaluation:
10 10 11 11 12 12 13 13 13 14

OWC Searching:

OWC Vector (sorted by area size)	Area
11 12 13 13 14 14 15 15 16 17	3528869
11 12 12 13 14 14 15 16 17 17	3542160
11 12 12 13 14 14 15 16 17 18	3543520
11 12 13 13 13 14 15 16 17 18	3545897
11 12 13 14 14 14 15 15 16 16	3554301
11 12 13 14 14 14 15 15 16 17	3555154

OW Selection: Re-sorting the OWC Vector by benefit

OWC Vector	Area	Benefit
11 12 13 14 14 14 15 15 16 17	3555154	0.00002514
11 12 13 13 14 15 15 15 16 17	3558734	0.00002492
11 12 13 13 14 14 15 16 17 18	3569846	0.00002475
11 12 13 13 14 15 16 16 17	3558591	0.00002472
11 12 13 13 14 14 15 16 17 17	3568486	0.00002435
11 12 13 14 14 15 15 15 16 17	3585019	0.00002425

OW :
11 12 13 14 14 14 15 15 16 17

Fig. 16. An example of the pure statistical analysis.

typically used in most OFDM-based systems [19], the SQNR of 45 dB and operating frequency of 50 MHz were adopted as constraints in these experiments.

Synthesis was conducted without any constraints using the Synopsys Design Analyzer [20]. The models of the hardware functional blocks, including adders, multipliers, and multiplexers, employ Synopsys DesignWare [21] and TSMC $0.25\mu m$ cell library, as well as the memory models, including the shift registers and ROMs [22], [23]. The fast carry look-ahead synthesis model for adders, the booth-encoded Wallace tree synthesis model for multipliers, and the universal multiplexer synthesis model for multiplexers are adopted. The area and timing reports of the Synopsys Design Analyzer are used for these models.

To verify the proposed optimization flow of the FFT processor, the C++ language with SystemC library is used to build the fixed-point model of the FFT hardware. In this experiment, the quantization mode is always truncation, which is SC_TRN in the SystemC library, and the overflow mode is saturation, which is SC_SAT in the library.

Finally, the platform is built on a PC with an Intel 2.4 GHz CPU and 768 Mbyte RAM. The operating system (OS) is Microsoft Windows 2000.

5.1 Variant FFT Lengths

Tables 4 and 5 present experimental results for area optimization versus FFT lengths of 8 to 8,192 points. Table 4 shows the experimental results for R2DSF and Table 5 shows the experimental results for R2^2SDF. Table 3 presents the optimization constraints in which the SQNR_Error is the value of e in (17) and the SQNR simulation confidence level is the amount of $(1-\alpha)100$ percent. Since the constraint of the SQNR error is smaller than 1 dB, the hybrid method is employed. In both tables, the first column lists the FFT length. The second column indicates whether the choice of word-length is optimized: w/o indicates that the word-length choice is based on using the same word-length for each stage, which is the minimum length such that all of the constraints are met, and w/ indicates that the word-length is decided using the proposed optimization method. The column "Area Reduction" presents the reduction rate of the area calculated by

$$\frac{Area_{\{\text{w/o optimization}\}} - Area_{\{\text{w/ optimization}\}}}{Area_{\{\text{w/o optimization}\}}} \times 100\%. \qquad (20)$$

The last column, "Time," is the required time of the used computer for performing optimization. The maximum and minimum area reduction rates are 24 percent and 9 percent for R2SDF and 23 percent and 6 percent for R2^2SDF. In summary, these tables show that, as N increases, the area reduction rate increases.

5.2 Output SQNR Requirement

Fig. 17 shows the achieved area reduction rate versus different SQNR constraints for R2SDF and R2^2SDF. In the conventional designs with the same word-length for all pipeline stages, the SQNR increases by about 6 dB when the word-length increases by 1 bit. In Fig. 17, a similar phenomenon can be observed when 6 dB is the cycle area reduction rate for different SQNR requirements. The reduction rate varied from 12 percent to 20 percent.

5.3 Loose SQNR Error Requirement

Table 6 shows the word-length optimization results based on the constraints presented in Table 3, except

TABLE 2
Common Specifications of FFT for OFDM

	FFT Length	Operating Frequency	I/O Format
Short Length	16~256 points	50 MHz	Complex, Word-Sequential
Long Length	256~8192 points	20 MHz	Complex, Word-Sequential

TABLE 3
Constraints for Optimization

SQNR	SQNR_Error	Operating Frequency	I/O Wordlength	SQNR Simulation Confidence Level
45 dB	0.1 dB	50 MHz	18 bits	95%

TABLE 4
Area Reduction of R2SDF with Different FFT Lengths Using the Hybrid Method

I/O Wordlength = 18 bits																	
FFT Length	w/ or w/o	Wordlength of PE Stage													Area	Area	Time
(N)	Optimization	1	2	3	4	5	6	7	8	9	10	11	12	13	(μm^2)	Reduction	(sec)
8	w/o	12	12	12											258831	16%	8
	w/	11	11	12											216184		
16	w/o	12	12	12	12										402693	11%	11
	w/	11	11	12	13										359378		
32	w/o	13	13	13	13	13									733434	9%	18
	w/	11	12	12	13	13									664339		
64	w/o	14	14	14	14	14	14								1152506	14%	20
	w/	11	12	13	13	13	14								993360		
128	w/o	14	14	14	14	14	14	14							1550048	10%	41
	w/	11	12	13	13	14	14	15							1388215		
256	w/o	15	15	15	15	15	15	15	15						2249617	16%	46
	w/	11	12	13	13	14	14	15	15						1882859		
512	w/o	15	15	15	15	15	15	15	15	15					2988883	14%	67
	w/	11	12	13	13	14	15	15	15	16					2569228		
1024	w/o	16	16	16	16	16	16	16	16	16	16				4474445	20%	139
	w/	11	12	13	13	14	14	15	16	17	17				3568487		
2048	w/o	16	16	16	16	16	16	16	16	16	16	16			6396581	19%	310
	w/	11	12	13	14	14	15	15	15	16	17	18			5184678		
4096	w/o	17	17	17	17	17	17	17	17	17	17	17	17		10255423	23%	616
	w/	11	12	13	13	14	15	15	16	17	17	17	18		7858489		
8192	w/o	17	17	17	17	17	17	17	17	17	17	17	17	17	16668224	24%	971
	w/	11	12	13	13	14	15	15	16	17	17	18	18	19	12676846		

TABLE 5
Area Reduction of $\mathrm{R2^2SDF}$ with Different FFT Lengths Using the Hybrid Method

I/O Wordlength = 18 bits																
FFT Length	w/ or w/o	Wordlength of PE Stage												Area	Area	Time
(N)	Optimization	1	2	3	4	5	6	7	8	9	10	11	12	(μm^2)	Reduction	(sec)
16	w/o	12	12	12	12									214156	11%	10
	w/	11	11	12	13									189706		
64	w/o	13	13	13	13	13	13							760033	6%	20
	w/	11	12	12	13	13	14							717914		
256	w/o	15	15	15	15	15	15	15	15					1732185	16%	31
	w/	11	12	12	13	13	14	15	16					1461204		
1024	w/o	16	16	16	16	16	16	16	16	16	16			3695209	20%	65
	w/	11	12	13	13	14	14	15	15	16	17			2973710		
4096	w/o	17	17	17	17	17	17	17	17	17	17	17	17	9271212	23%	317
	w/	11	12	12	13	14	14	15	16	17	18	18	18	7120428		

$\mathrm{SQNR\ error} = 1.1$ db. Since the constraint of the SQNR error equals 1.1 dB, the statistical analysis method is employed for optimization. The achieved area reduction rate versus the FFT lengths drifts toward the same direction as the results shown in Table 4. The SQNR corresponding to the OWs is evaluated by simulation to verify its accuracy. Table 6 lists these evaluated results in the last column. The obtained SQNRs of FFT lengths 128, 512, and 2,048 points violate the SQNR constraint.

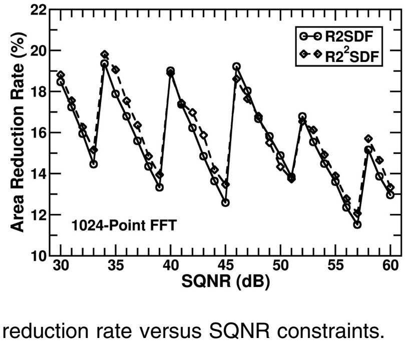


Fig. 17. Area reduction rate versus SQNR constraints.

5.4 Comparisons of the Three Methods

Fig. 18 shows the comparisons of the experimental results by employing the pure statistical method, the pure simulation-based method, and the hybrid method for the R2SDF optimization. The pure simulation-based method is the proposed optimization without statistical error analysis and applies a greedy algorithm to find the local optimum. One dB is added to the SQNR constraint (iSQNR) to allow for the SQNR error in the statistical model. This is done for overdesign to cover the SQNR error of the statistical model. These experimental results suggest that overdesign with extra 1 dB has larger hardware costs than the other two methods. This difference is reflected by the comparisons of the area reduction rate versus FFT lengths shown in the top frame. The bottom frame presents that the required time for performing optimization can be dramatically reduced by using the hybrid methods compared with that when using the pure simulation-based analysis with approximately the same optimized area. In summary, employing the hybrid method for the word-length optimization is faster than the pure simulation-based method and provides a more accurate determination of word-lengths than the pure statistical method.

TABLE 6
Area Reduction of R2SDF with Different FFT Lengths Using Statistical Analysis

I/O Wordlength = 18 bits																		Post
FFT Length	w/ or w/o	Wordlength of PE Stage													Area	Area	Time	SQNR
(N)	Optimization	1	2	3	4	5	6	7	8	9	10	11	12	13	(μm^2)	Reduction	(sec)	(dB)
8	w/o	12	12	12											258831	16%	1	46.43
	w/	11	11	12											216184			
16	w/o	12	12	12	12										402693	11%	1	45.44
	w/	11	11	12	13										359378			
32	w/o	13	13	13	13	13									733434	9%	1	45.27
	w/	11	12	12	13	13									664339			
64	w/o	14	14	14	14	14	14								1152506	14%	1	45.58
	w/	11	12	13	13	13	14								993360			
128	w/o	14	14	14	14	14	14	14							1550048	12%	1	44.94
	w/	11	12	12	13	14	14	15							1370066			
256	w/o	15	15	15	15	15	15	15	15						2249617	16%	1	45.64
	w/	11	12	13	13	14	14	15	16						1884288			
512	w/o	15	15	15	15	15	15	15	15	15					2988883	15%	1	44.62
	w/	11	12	13	13	14	14	15	15	16					2546349			
1024	w/o	16	16	16	16	16	16	16	16	16	16				4474445	21%	1	45.00
	w/	11	12	13	14	14	14	15	15	16	17				3555154			
2048	w/o	16	16	16	16	16	16	16	16	16	16	16			6396581	19%	1	44.84
	w/	11	12	13	13	14	15	15	15	16	17	18			5153721			
4096	w/o	17	17	17	17	17	17	17	17	17	17	17	17		10255423	23%	2	45.30
	w/	11	12	13	13	14	15	15	16	17	17	18	19		7875389			
8192	w/o	17	17	17	17	17	17	17	17	17	17	17	17	17	16668224	24%	2	45.04
	w/	11	12	13	13	14	15	15	16	17	17	18	18	19	12676846			

6 Conclusions

This work presented a modified statistical error model for varying word-lengths of individual stages of a pipelined FFT processor with Radix-2 and Radix-2^2 algorithms. The error model is used to investigate the hybrid method for word-length optimization at the stage level. The proposed hybrid method combined with statistical and simulation-based error analyses performs the word-length optimization based on minimizing the area size cost. A generator of pipelined FFT processors was developed for experiments to validate the proposed method and error model. The generator gives the suggested value of the maximum available SQNR or operating frequency at the UBW evaluation when constraints cannot be met. Experiments indicate that the hybrid method can obtain optimized results faster than the conventional simulation-based method, thereby reducing the design time for the FFT processor. Furthermore, the area size of the processor is also minimized.

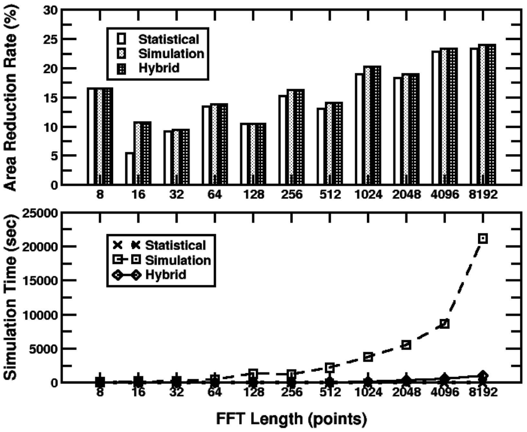


Fig. 18. Comparisons of results using different analytical methods.

References

[1] J.W. Cooley and J.W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," *Math. Computation*, vol. 19, pp. 297-301, 1965.

[2] L.R. Rabiner and G. Gold, *Theory and Application of Digital Signal Processing*. Prentice Hall, 1975.

[3] E.H. Wold and A.M. Despain, "Pipelined and Parallel-Pipeline FFT Processors for VLSI Implementation," *IEEE Trans. Computers*, vol. 33, no. 5, pp. 414-426, May 1984.

[4] S.-S. He and M. Torkelson, "A New Approach to Pipeline FFT Processor," *Proc. 10th Int'l Parallel Processing Symp. (IPPS '96)*, pp. 766-770, 1996.

[5] S.-S. He and M. Torkelson, "Designing Pipeline FFT Processors for OFDM (De)Modulation," *Proc. URSI Int'l Symp. Signals, Systems, and Electronics (ISSSE '98)*, pp. 256-262, 1998.

[6] J.-C. Choi, W.-C. Choi, S.-G. Hwang, M.M.-O. Lee, and K.-R. Cho, "Design of a New IFFT/FFT for IEEE 802.11a WLAN Based on the Statistics Distribution of the Input Data," *Proc. IEEE Int'l Conf. High-Speed Networks and Multimedia Comm. (HSNMC '04)*, pp. 589-597, June 2004.

[7] J.-Y. Oh and M.-S Lim, "Area and Power Efficient Pipeline FFT Algorithm," *Proc. IEEE Workshop Signal Processing Systems Design and Implementation*, pp. 520-525, 2005.

[8] L. Yang, K. Zhang, H. Liu, J. Huang, and S. Huang, "An Efficient Locally Pipelined FFT Processor," *IEEE Trans. Circuits and Systems II: Express Briefs*, vol. 53, pp. 585-589, July 2006.

[9] P.D. Welch, "A Fixed-Point Fast Fourier Transform Error Analysis," *IEEE Trans. Audio Electroacoustics*, vol. 17, pp. 151-157, June 1969.

[10] A.V. Oppenheim and C.J. Wenstein, "Effects of Finite Register Length in Digital Filtering and the Fast Fourier Transform," *Proc. IEEE*, vol. 60, pp. 957-976, Aug. 1972.

[11] R. Meyer, "Error Analysis and Comparison of FFT Implementation Structures," *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, vol. 2, pp. 888-891, May 1989.

[12] A. Pomerleau, H.L. Buijs, and M. Fournier, "A Two-Pass Fixed Point Fast Fourier Transform Error Analysis," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 25, pp. 582-585, Dec. 1977.

[13] S. Johansson, S. He, and P. Nilsson, "Wordlength Optimization of a Pipelined FFT Processor," *Proc. 42nd Midwest Symp. Circuits and Systems*, pp. 501-503, 1999.

[14] A.V. Oppenheim and R.W. Schafer, *Discrete-Time Signal Processing*, second ed. Prentice Hall, 1999.

[15] M. Sundaramurthy and V.U. Reddy, "Some Results in Fixed-Point Fast Fourier Transform Error Analysis," *IEEE Trans. Computers,* vol. 26, no. 3, pp. 305-308, Mar. 1977.

[16] R.E. Walpole, R.H. Myers, and S.L. Myers, *Probability and Statistics for Engineers and Scientists,* sixth ed. Prentice Hall, 1998.

[17] *SystemC Version 2.0 User's Guide,* http://www.systemc.org, 2002.

[18] J. Bhasker, "A SystemC Primer," *Star Galaxy,* 2002.

[19] W.C. Yeh, "Arithmetic Module Design and Its Application to FFT," PhD dissertation, Nat'l Chiao Tung Univ., Taiwan, R.O.C., July 2001.

[20] *Synopsys Design Analyzer,* http://www.synopsys.com, 2000.

[21] *Synopsys Design Ware,* http://www.synopsys.com, 2000.

[22] *Artisan TSMC 0.25μm Process High-Density Dual-Port SRAM (HD-SRAM-DP) Generator User Manual, Release 1.0,* http://www.artisan.com, June 2000.

[23] *Artisan TSMC 0.25μm Process High-Speed Single-Port SRAM (HS-SRAM-SP) Generator User Manual, Release 3.0,* http://www.artisan.com, June 2000.

Cheng-Yeh Wang received the BS and MS degrees in electronics engineering from the National Chiao Tung University, Hsinchu, Taiwan, Republic of China, in 1999 and 2000, respectively. He is currently working toward the PhD degree at the National Chiao Tung University. His current research interests include CAD, VLSI design, network on chips, and electronic system level (ESL) design. He is a student member of the IEEE.

Chih-Bin Kuo received the MS degree in electronics engineering from the National Chiao Tung University, Hsin-Chu, Taiwan.

Jing-Yang Jou received the BS degree in electrical engineering from the National Taiwan University, Taiwan, Republic of China, in 1979 and the MS and PhD degrees in computer science from the University of Illinois, Urbana-Champaign, in 1983, and 1985, respectively. He is currently the director general of the National Chip Implementation Center, National Applied Research Laboratories, Taiwan. He is a full professor and was the chairman of the Electronics Engineering Department from 2000 to 2003 at the National Chiao Tung University, Hsinchu, Taiwan. Before joining the National Chiao Tung University, he was with GTE Laboratories from 1995 to 1996 and with AT&T Bell Laboratories, Murray Hill, New Jersey, from 1986 to 1994. He received the distinguished paper award of the IEEE International Conference on Computer-Aided Design in 1990, the Outstanding Academy-Industry Cooperation Achievement Award granted by the Ministry of Education (MOE), Taiwan, in 2002, and the Outstanding Electrical Engineering Professor Award from the Chinese Institute of Electrical Engineering (CIEE) in 2006. His research interests include logic and physical synthesis, design verification, CAD for low power, and network on chips. He has published more than 160 technical papers. He is a fellow of the IEEE. He is the elected president of the Taiwan Integrated Circuit Design Society (TICD) 2007-2008. He serves as an associate editor for the *IEEE Transactions on Very Large Scale Integration Systems*. He was the technical program chair of the 2007 International Symposium on VLSI Design, Automation, and Test (VLSI-DAT), the 12th VLSI Design/CAD Symposium (2001), and the Fourth Asia-Pacific Conference on Hardware Description Languages (APCHDL '97). He was the honorary chair of the International Workshop on Multi-Project Chip (IWMC '06) and the executive chair of the Second Taiwan-Japan Microelectronics International Symposium (2002).