Fine Grain Word Length Optimization for Dynamic Precision Scaling in DSP Systems

Seogoo Lee and Andreas Gerstlauer
Department of Electrical and Computer Engineering
The University of Texas at Austin
{sglee,gerstl}@utexas.edu

***Abstract*—Dynamic precision scaling is a promising technique to reduce power consumption in Digital Signal Processing (DSP) systems. Power savings are achieved by dynamically adapting word lengths to a time-varying environment. Typical applications are wireless communication systems that operate under different wireless channel conditions. One of the obstacles of such techniques is that they require an optimization of word lengths for all intermediate values at all possible operation points. This makes traditional simulation-based fixed-point optimization infeasible. In this paper, we study efficient heuristics to find optimal sets of word lengths for all variables in a system under a range of operating conditions. We exploit statistical analysis of quantization noise coupled with Additive White Gaussian Noise (AWGN) models of the channel environment. Applied to an example of a Fast Fourier Transform (FFT) block in an Orthogonal Frequency Division Multiplexing (OFDM) system, a more than 5,000x improvement in optimization time compared to an efficient simulation-based word length optimization method can be achieved.**

***Index Terms*—Power reduction, word length optimization, FFT**

I. INTRODUCTION

Power consumption continues to be a critical aspect of Digital Signal Processing (DSP) systems. Among various power reduction techniques, dynamic precision scaling is a technique aimed at reducing internal system precision and hence dynamic power consumption in reaction to changing operating conditions [1], [2], [3] and [4]. In traditional design of DSP systems, fixed-point word lengths are determined to support the worst case. However, the system is not always in this situation and hence a power reduction opportunity exists. The key idea is to control precision of a system adaptively according to current signal quality.

In [1], the authors introduce the concept of dynamic word length scaling by forcing lower significant bits to zero if the current signal quality is better than a predetermined minimum requirement. There are two drawbacks in their work: the authors use the same word length across the whole design and their method requires dedicated training symbols to find the best word lengths at run time in a self-adjusting scheme. However, a self-adjusting process introduces additional overhead that negates some of the power savings. Morover, since fine-tuning of word lengths would result in even more overhead in a run-time approach, their method is limited to a single word length over all variables in the system. In [2], the authors use both precision and voltage scaling to maximize power reduction. They first optimize word lengths according to the channel environment and then use these word lengths to find the optimal voltage that still satisfies a required error rate. Their method is robust to process variation, but it also incurs run time overhead. In [3] and [4], word lengths are optimized at design time to avoid the run time overhead. In [3], the authors target software-defined implementations of wireless systems, and use simulations to support fine-grain optimization of all variables but only consider powers of 2 as word lengths. In [4], optimal word lengths are also determined by simulation, where precision is instead allowed to decrease when it can be absorbed in increasing base noise under varying bit error requirements. In both cases, given the large number of variables and operating conditions to optimize for, simulation-based design methods are time-consuming, which is a main reason why dynamic precision scaling is not widely used.

In this paper, we investigate novel analytical techniques that resolve some of the drawbacks of previous works. Our method calculates the optimal set of word lengths at design time using statistical analysis. Compared to the methods that require long simulations, our approach can dramatically reduce the design time. Moreover, fine tuning of word lengths reduces overhead and improves power consumption compared to run-time methods in [1], [2].

The paper is organized as follows: after a brief summary of related work and the system model, our optimization process is described in Section II. Result are shown in Section III. Finally, Section IV concludes and discusses future work.

A. Related Work

Fixed-point conversion and word length optimization has a long history of research [5], [10]. In a fixed-point representation, integer bits are related to the dynamic range of a signal and fractional bits are related to precision. For determining the optimal number of both integer and fractional bits, analytical or simulation-based methods have been introduced.

Simulation-based methods are widely used to estimate fixed-point performance. For example, Sung et al. [5] add a Signal to Quantization Noise Ratio (SQNR) block to quantify the finite word-length effects when the word-length for the implementation of the system changes. In [6], various word length search methods are summarized and compared. The efficiency of simulation-based methods is analyzed to determine the number of simulations needed to reach optimum word

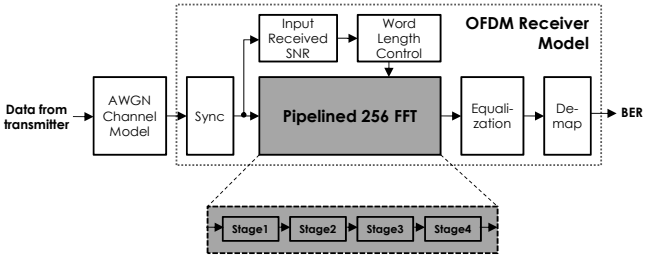

Fig. 1: OFDM receiver.

lengths. The complexity of a full search is $O(N^s)$, where N is number possible word lengths for each decision variable, and s is number of variables. It is shown that the complexity can be dramatically reduced to $O(s)$ by using efficient search methods that rely on sensitivity information but may run into local minima.

Among the various analytical techniques, in [9] the authors adopt Affine Arithmetic (AA) to model the min/max error propagation of quantization noise. However, static min/max approaches are not appropriate for dynamic precision scaling. They are known to be overly conservative. Furthermore, in applications such as communication systems, Additive White Gaussian Noise (AWGN) sources from the outer channel environment are hard to characterize in a min/max form. The research done by Shi and Brodersen [10] analyzes quantization noise with perturbation theory instead. They measure the sensitivities of input word lengths to output noise by simulations and use this information in their constraint function. Constantinides et al. find optimal word lengths by evaluating the variance of quantization noise through the system transfer function [7]. Finally, in [11] a variance propagation method is applied to quantization noise analysis in a FFT block. Menard et al. [8] propose a similar method for generalized Data Flow Graphs (DFGs). Because their method can be used both for linear and nonlinear systems and is suitable for general DSP systems, we use it for statistical word-length analysis and optimization in this paper.

B. System Model

We use a basic OFDM receiver to verify our idea (Figure 1). The receiver consists of a synchronization block, a 256 point FFT, an equalizer and a symbol de-mapper. An AWGN channel model is assumed to exist between transmitter and receiver. The FFT is used as an example to be designed in dynamically scaled fixed point form. Without loss in generality, among many implementation schemes, we assume that a pipelined radix-4 FFT is used. The 256-point FFT has four radix-4 stages and each stage contains a radix-4 butterfly and a twiddle multiplication. Since we change the SNR of the system by adding quantization noise, the targeted SNR of the FFT is defined as a desired SNR at its output, which is affected both by a given input SNR and internal quantization noise sources. At design time, statistical analysis determines multiple sets of word lengths for all internal variables in FFT and at all input SNRs defined through floating point simulations. At run time, a SNR block measures the FFT's input SNR and a word length controller selects the best set of word lengths that is suitable for the current input SNR to maintain a pre-defined output SNR. We assume that perfect SNR measurement is possible. We only use fixed point numbers with a round-to-nearest rounding method.

The final performance metric for a wireless communication system is usually the coded Frame Error Rate (FER). In this paper, however, we use uncoded Bit Error Rate (BER) instead. Every FER has a corresponding BER, which is not affected by frame length and coding scheme. Our goal is to find FFT word lengths that satisfy a desired BER for any given input SNR. BER is closely related to SNR. However, BER is decision error and the relationship between SNR and BER is not linear but a function of the noise's Probability Density Function (PDF). It is hard to find the exact PDF of noise for a general DSP system that has quantization noises. In this paper, we assume that propagated noise at the output of an FFT stage is Gaussian distributed. From the central limit theorem, it follows that the noise of the output of a radix-4 butterfly can be approximated as Gaussian. Our simulation results also show that the output noise from twiddle multiplications, since additive, can be approximated to be Gaussian. Furthermore, the input signal is assumed to be Gaussian. This is true considering the time domain signal of an OFDM system. Hence, although we use an SNR metric in our analysis, under the above assumptions we can estimate BER from SNR.

Our approach is heuristic because 1) we use a Pseudo Quantization Noise (PQN) model instead of exact distribution functions of noise, and 2) we consider the quantization of system coefficients, such as the twiddle factors or filter coefficients as additive noise injection, which ignores associated changes of the transfer function. In contrast to other approaches [11], this allows us to consider coefficient quantization noise. However, the impact on the transfer function and hence corresponding inaccuracies in our method are specific to a given application.

II. STATISTICAL ANALYSIS

Our approach only applies to fractional bits. As such, we assume that the number of integer bits has already been determined by range analysis. Our word length optimization procedure for fractional bits is shown in Figure 2.

Our optimization problem is to minimize power consumption subject to output SNR constraints under given input and quantization noises. We start by building a floating point model of our system, which is simulated to obtain a targeted output SNR and a set of possible input SNRs. These are the inputs to our optimization problem. From the floating-point model, we also extract the DFG of the system. With the DFG and targeted output SNR we build cost and constraint functions, which are functions of input SNR and word lengths. Then, we apply our optimization problem to each input SNR in order to determine an optimal set of word lengths that minimizes power consumption while satisfying the targeted output SNR. This process is repeated for all possible input SNRs.

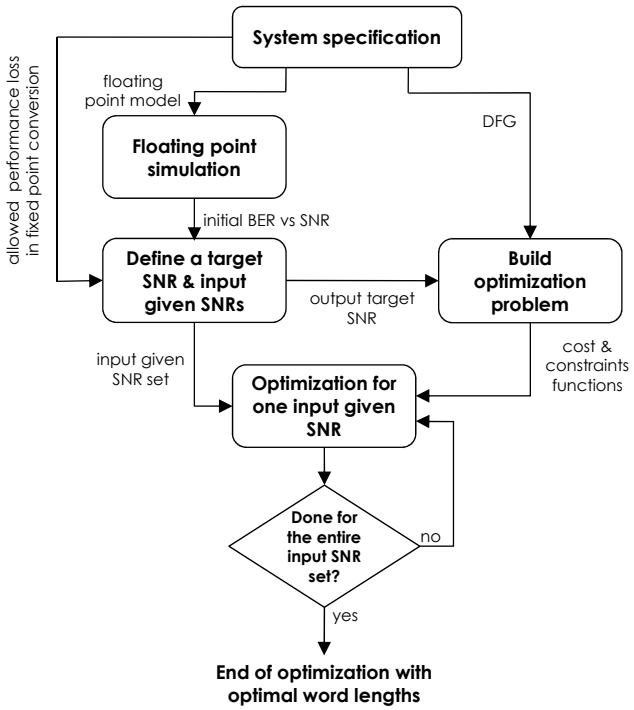

Fig. 2: Optimization procedure.

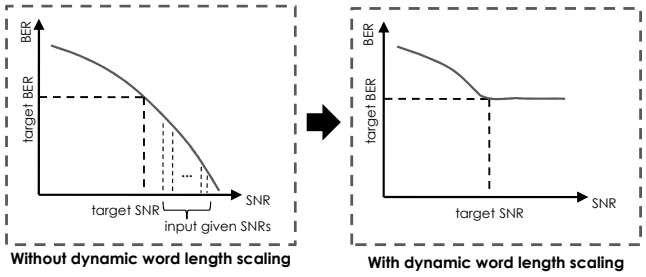

Fig. 3: Performance change after optimization.

A. Quantization Noise Model

We assume that quantization noise sources as well as AWGN are independent. It is well known that we can get the variance σ^2 after addition and multiplication of two independent random variables as follows:

$$\begin{aligned} \text{Addition: } \sigma^2 &= \sigma_1^2 + \sigma_2^2 \\ \text{Multiplication: } \sigma^2 &= \mu_1^2\sigma_2^2 + \mu_2^2\sigma_1^2 + \sigma_1^2\sigma_2^2, \end{aligned}$$

where μ_i is the expectation and σ_i^2 is the variance of input random variable i. The output variances after subtraction or division are also available in a similar way.

Quantization noise is modeled as additive noise. If we add quantizers to two independent inputs of an adder, $s1$ and $s2$ with signal variances σ_{s1}^2 and σ_{s2}^2, respectively, we add noise sources σ_{n1}^2 to σ_{s1}^2 and σ_{n2}^2 to σ_{s2}^2. At the output of the addition, the noise and signal variances therefore become $\sigma_n^2 = \sigma_{n1}^2 + \sigma_{n2}^2$ and $\sigma_s^2 = \sigma_{s1}^2 + \sigma_{s2}^2$, respectively. Hence, the total variance is $\sigma^2 = \sigma_s^2 + \sigma_n^2 = \sigma_{s1}^2 + \sigma_{s2}^2 + \sigma_{n1}^2 + \sigma_{n2}^2$.

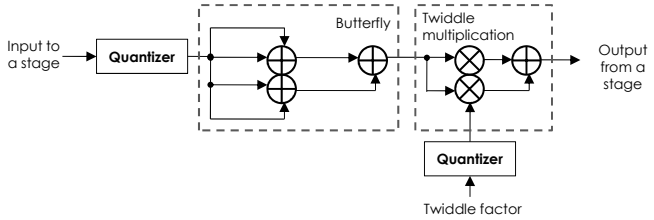

Fig. 4: DFG for one radix-4 stage of the FFT.

B. Variance Propagation for one Decision Variable

The DFG for one stage of the FFT is shown in Figure 4. The first two additions are for the butterfly and the following multiplication and addition represent the complex twiddle multiplication. Only calculation of the real phase is shown. The same computation is performed for the imaginary phase. There can be two quantization points in each stage of the FFT that affect the number of fractional bits: quantization of the input and of the twiddle factor. For simplification, we use a single word length for those two quantization points.

The input to the FFT can be modeled as a sum of the error-free input with variance $\sigma_{s_{in}}^2$ and additive input noise with variance $\sigma_{n_{in}}^2$. Furthermore, the variance of quantization noise with F fractional bits and uniform distribution under a round-to-nearest rounding is $\sigma_{n_{quan}}^2 = \frac{1}{3}2^{-2F-2}$. The variance at the output of the butterfly is then also a sum $\sigma_{butterfly}^2 = \sigma_{s_{butterfly}}^2 + \sigma_{n_{butterfly}}^2$ of the error-free output variance $\sigma_{s_{butterfly}}^2 = 4\sigma_{s_{in}}^2$ and the noise variance $\sigma_{n_{butterfly}}^2 = 4(\sigma_{n_{in}}^2 + \sigma_{n_{quan}}^2) = 4\sigma_{n_{in}}^2 + \frac{1}{3}2^{-2F}$, including input quantization noise.

The butterfly output becomes the input to twiddle multiplication. The other input, the twiddle factor is a sum of the ideal, sinusoidal twiddle factor with variance $\sigma_{s_{twiddle}}^2 = 1/8$ (for a 4-stage, 256-point FFT) and additive quantization noise $\sigma_{n_{quan}}^2$. We assume that all signals and variables have zero mean ($\mu = 0$) and that $\sigma_{s_{in}}^2$ is gain-controlled to 1. Therefore, the output variance after one FFT stage is

$$\begin{aligned} \sigma_{out}^2 &= 2(\sigma_{s_{butterfly}}^2 + \sigma_{n_{butterfly}}^2)(\sigma_{s_{twiddle}}^2 + \sigma_{n_{quan}}^2) \\ &\approx 1 + \sigma_{n_{in}}^2 + (\frac{2}{3} + \frac{2}{3}\sigma_{n_{in}}^2 + \frac{1}{12})2^{-2F} \\ &= 1 + \sigma_{n_{out}}^2 \end{aligned}$$

With this, we can use the allowed performance loss in the floating to fixed-point conversion process for our optimization. For example, if the allowed performance loss in SNR is 0.2dB for fixed point conversion, given an input SNR of 11.6dB and the signal power of 1 ($\sigma_{n_{in}}^2 = 0.069$), the minimum word length to get a 11.4dB output SNR ($\sigma_{n_{out}}^2 = 0.072$) becomes $F = 5$. We can find the same value through simulation. By contrast, min/max propagation using affine arithmetic [9] with a $\pm 3\sigma$ min/max of the input AWGN would result in $F = 9$. This reflects that static analysis with min/max propagation is conservative.

C. Extension to Multiple Stages

We extend the above analysis to an FFT with multiple stages and hence decision variables. The output noise from the first stage becomes the input noise to the second stage and propagates through the whole FFT. At the end of the FFT, a noise constraint function can be represented as a function of the variance of the input AWGN (σ^2_{AWGN}) and a set of word lengths F_i for each stage i. For our four-stage FFT example, the output noise variance from the first stage is a function of σ^2_{AWGN} and F_1 as shown in the previous subsection:

$$\sigma^2_{out,1} = f(\sigma^2_{AWGN}, F_1),$$

where $f()$ is defined as

$$f(\sigma, F) = \sigma + (\frac{2}{3} + \frac{2}{3}\sigma + \frac{1}{12})2^{-2F}.$$

Similarly, the output noise variance from the i-th stage ($i > 1$) is a function of $\sigma^2_{out,i-1}$ and F_i:

$$\sigma^2_{out,i} = f(\sigma^2_{out,i-1}, F_i).$$

Combining the above output noise formulations, the output noise variance from the last FFT stage is a function of σ^2_{AWGN} and F_1 through F_4, and the constraint function becomes:

$$f(f(f(f(\sigma^2_{AWGN}, F_1), F_2), F_3), F_4) \leq \sigma^2_{n_{out}}.$$

Under different input SNRs, the targeted output noise variance should remain constant. For an example of QPSK modulation with BER = 0.01%, the SNR of the wireless channel can be any value larger than 11.4dB, but the output noise SNR should always be close to (but not below) 11.4dB.

Our cost function is the same for all input SNRs. We assume that all the bits in our design have the same transition rate of 0.5. With this assumption, dynamic power consumption is linearly proportional to the area of the circuit that is toggling. Hence, our power cost function can be represented as the number of unit hardware blocks. For combinational logic such as adders and multipliers, the cost for each stage is the same and can be represented as the number of 1-bit full adder equivalents. For one stage of the FFT, I_a is the number of integer bits at the input to the FFT and I_b is the number of integer bits for the twiddle factor. Then, the cost for one butterfly is

$$c^{'} = 2 \times (I_a + F) + (I_a + 1 + F)$$

and the cost for one twiddle multiplication is

$$c^{''} = 2 \times (I_a + 2 + F) \times (I_b + F) + (I_a + I_b + 2 + 2F).$$

$I_a = 3$ including the sign bit is enough not to affect decoding performance. Also, I_b is 1 since the range of twiddle factor is within ± 0.5.

For sequential logic, the power consumption of a 1-bit D Flip-Flop (DFF) is compared to that of a 1-bit full adder using a TSMC 0.18um library, which is our target technology for validation. In each FFT stage, two intermediate values are stored: $(I_a + F)$ bits of data after input quantization and $(I_a + F + 2)$ bits of data after the butterfly. Hence, the number of DFFs used in one FFT stage becomes $(2F + 8)$. According to synthesis results, the ratio in power consumption between a 1-bit DFF and a 1-bit full adder is 8.4, and this is used as a weight of the normalized sequential logic cost:

$$c^{'''} = 8.4(2F + 8).$$

With this, the total cost of one FFT stage becomes:

$$C(F) = c^{'} + c^{''} + c^{'''} = 2F^2 + 33.4F + 32.$$

Note that for large FFTs, intermediate data is usually stored in SRAMs. However, since scaling is only performed for DFFs and combinational logic, the power consumption of SRAMs is not included in our analysis.

Since our optimization problem is not linear nor convex, we apply Adaptive Simulated Annealing (ASA) [12] as in [9]. ASA adapts to changing sensitivities and has faster convergence compared to traditional simulated annealing.

D. Overhead Analysis

Dynamic precision scaling requires an input SNR measurement block and a mapping table between measured input SNRs and word lengths for all decision variables. Also, combinational gates are added in front of the FFs to control clock gating. Most wireless communication systems already include SNR measurement capabilities for various uses such as channel state information feedback. Hence, it is assumed that our approach uses the existing SNR measurement block and we do not include it in overhead analysis. By contrast, with binary on/off decisions stored in the mapping table, its size becomes $N_s \times N_d$, where N_s is the number of SNR steps and N_d is the number of DFFs to control. For example, $N_s = 11$ if there are 11 input SNR steps from 6dB to 16dB, and $N_d = 24$ if there are 4 stages, and each stage has 6 DFFs to be controlled. The overhead in power consumption of the mapping table and additional clock gating logic is included in our power analysis shown in the results.

III. RESULTS

In the following, we validate our optimization model and demonstrate the achievable gains in word length optimization times. We apply our approach to a 256-point FFT example in a QPSK OFDM receiver with a cyclic prefix length of 64 assuming perfect synchronization. We perform power estimation of the generated gate-level netlists using Synopsys Power Compiler with a TSMC 0.18um library at a 40MHz clock. We include both dynamic and leakage power consumption in all reported results. Our optimization is only targeted at dynamic power, and leakage is less than 1 uW for our FFT example in 0.18um. For more advanced technology nodes with a larger fraction of leakage power, design techniques such as power gating can be combined with dynamic word length scaling.

The method presented in this paper, which we call dynamic scaling by variance propagation (DS-VP), is compared against four conventional methods: 1) non-scaling by full simulation (NS-FS), which only finds one set of word lengths for the

TABLE I: Optimized word lengths for various target SNRs (BERs).

	Channel SNR	NS-FS $\{F_i\}$	NS-FS Power	CS-FS $\{F_i\}$	CS-FS Power [mW]	DS-FS $\{F_i\}$	DS-FS Power [mW]	DS-ES $\{F_i\}$	DS-ES Power [mW]	DS-VP $\{F_i\}$	DS-VP Power [mW]
7.25dB (1%)	<8dB	{4,4,4,3}	2.52 mW	{4}	2.53 (0.4%)	{4,4,4,3}	2.57 (2.0%)	{4,4,4,3}	2.57 (2.0%)	{4,4,4,3}	2.57 (2.0%)
	8dB			{3}	2.27 (-9.9%)	{3,3,3,2}	2.20 (-12.7%)	{3,3,3,2}	2.20 (-12.7%)	{3,3,3,2}	2.20 (-12.7%)
	9dB			{3}	2.27 (-9.9%)	{3,3,3,1}	2.13 (-15.5%)	{3,3,3,1}	2.13 (-15.5%)	{3,2,3,2}	2.16 (-14.3%)
	10dB			{3}	2.27 (-9.9%)	{3,3,2,1}	2.05 (-18.7%)	{3,2,3,1}	2.06 (-18.3%)	{3,2,2,1}	1.97 (-21.8%)
	11dB			{2}	1.94 (-23.0%)	{3,2,2,1}	1.97 (-21.8%)	{3,2,2,1}	1.97 (-21.8%)	{2,2,3,1}	1.99 (-21.0%)
	12dB			{2}	1.94 (-23.0%)	{2,2,2,1}	1.90 (-24.6%)	{2,2,2,1}	1.90 (-24.6%)	{2,2,2,1}	1.90 (-24.6%)
9.7dB (0.1%)	<10dB	{4,5,4,4}	2.63 mW	{5}	2.94 (11.8%)	{4,5,4,4}	2.66 (1.1%)	{4,5,4,4}	2.66 (1.1%)	{4,5,4,4}	2.66 (1.1%)
	10dB			{4}	2.53 (-3.8%)	{5,5,4,3}	2.60 (-1.1%)	{5,5,4,3}	2.60 (-1.1%)	{4,4,4,4}	2.57 (-2.3%)
	11dB			{3}	2.27 (-13.7%)	{3,3,3,2}	2.20 (-16.3%)	{3,3,3,2}	2.20 (-16.3%)	{3,3,3,2}	2.20 (-16.3%)
	12dB			{3}	2.27 (-13.7%)	{3,3,3,2}	2.20 (-16.3%)	{3,3,3,2}	2.20 (-16.3%)	{3,3,3,1}	2.13 (-19.0%)
	13dB			{3}	2.27 (-13.7%)	{3,3,2,2}	2.16 (-17.9%)	{3,3,2,2}	2.16 (-17.9%)	{3,2,3,2}	2.16 (-17.9%)
	14dB			{3}	2.27 (-13.7%)	{3,3,2,2}	2.16 (-17.9%)	{3,3,2,2}	2.16 (-17.9%)	{3,2,2,2}	2.06 (-21.7%)
11.4dB (0.01%)	<12dB	{5,4,5,4}	2.75 mW	{5}	2.94 (6.9%)	{5,4,5,4}	2.83 (2.9%)	{5,4,5,4}	2.83 (2.9%)	{5,4,5,4}	2.83 (2.9%)
	12dB			{4}	2.53 (-8.0%)	{4,5,3,3}	2.49 (-9.5%)	{4,5,3,3}	2.49 (-9.5%)	{4,4,4,3}	2.57 (-6.5%)
	13dB			{3}	2.27 (-17.5%)	{4,3,3,2}	2.29 (-16.7%)	{4,3,3,2}	2.29 (-16.7%)	{4,3,3,2}	2.29 (-16.7%)
	14dB			{3}	2.27 (-17.5%)	{3,3,3,2}	2.20 (-20.0%)	{3,3,3,2}	2.20 (-20.0%)	{3,3,3,2}	2.20 (-20.0%)
	15dB			{3}	2.27 (-17.5%)	{3,3,3,2}	2.20 (-20.0%)	{3,3,3,2}	2.20 (-20.0%)	{3,3,2,3}	2.22 (-19.3%)
	16dB			{3}	2.27 (-17.5%)	{3,3,3,2}	2.20 (-20.0%)	{3,3,3,2}	2.20 (-20.0%)	{3,3,2,2}	2.16 (-21.5%)
Optim. time		3.6 hours		6 min.		21.6 hours		1-2 min.		1.2-1.8 msec.	

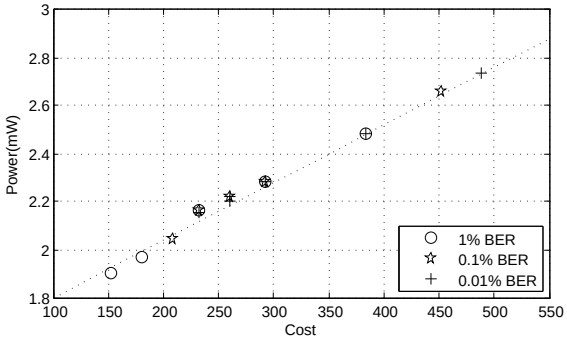

Fig. 5: Accuracy of cost function.

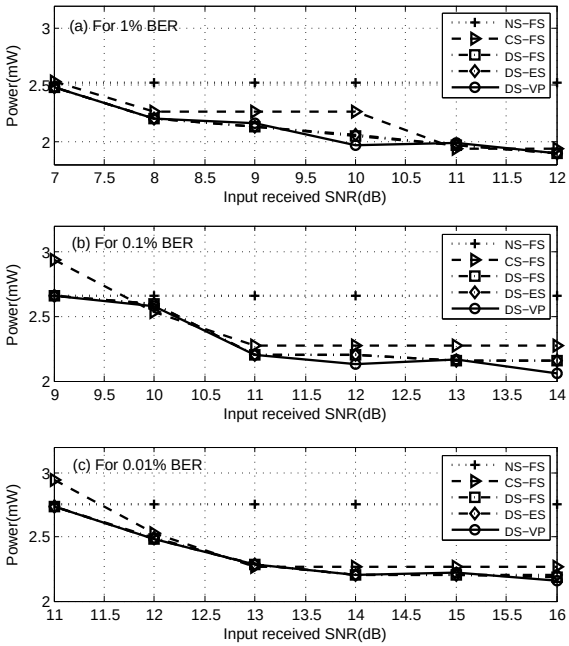

Fig. 6: Power comparison.

worst-case operation point, 2) coarse dynamic scaling by full simulation search (CS-FS), which finds multiple sets of word lengths by full simulation, but using a single word length for all variables in a set, 2) dynamic scaling by full simulation search (DS-FS), which finds optimal sets of word lengths using full simulation, and 4) dynamic scaling by efficient simulation search (DS-ES), which finds multiple sets of word lengths using the efficient simulation approach from [6].

Table I shows the sets of word lengths found by the different methods across different target BERs and corresponding input SNRs. The sets of word lengths in Table I are the word lengths for Stage 1 to Stage 4 of the FFT, i.e. $\{F_1, F_2, F_3, F_4\}$. The table also includes estimated power consumption and optimization runtime for each approach. All experiments were performed on an Intel Core i7 workstation running at 2.7ĠHz. The sets of word lengths from DS-FS are optimal and used as word length and power reference.

Our method shows a significant gain in design time compared to simulation based methods, which makes dynamic scaling feasible even for large systems with many variables and operation points. For one operation point in our FFT example, the number of simulations using a full search is 6^4 (4 decision variables and with a range from 1 to 6 bits each). For each simulation trial, we run 10,000 OFDM symbols corresponding to 5 million bits in order to achieve enough simulation accuracy. Each such simulation takes about 10 seconds. To find the optimal word lengths using an exhaustive search requires 3.6 hours. With the preplanned simulation method from [6], the number of trials can be significantly reduced. For example, if the search starts from {2,2,2,2}, and the optimal word length set is {4,3,3,2}, optimal word lengths can be obtained with only 4 simulations. However, for dynamic scaling, a search is required for each operating condition and total optimization time increases linearly with the number of operating points. Thus, even efficient simulation-based methods may still not be suitable for design-time optimization in the presence of dynamic scaling.

By contrast, our analysis method requires only about 2ms to find a set of word lengths for one operating point, which

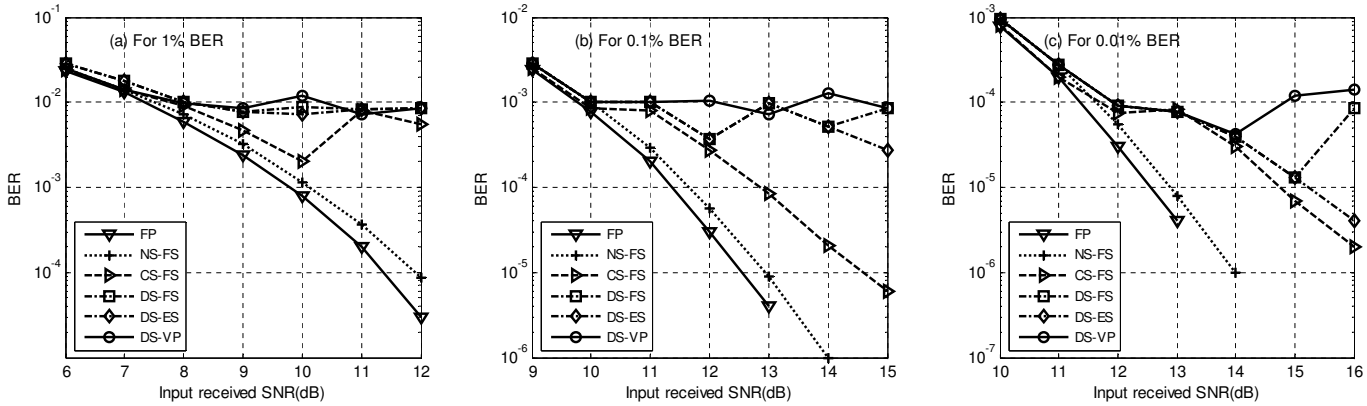

Fig. 7: QPSK BER performance.

is 5,000 times faster than the time for one simulation trial. Considering that word length optimizations can take up to 50% of design time with conventional simulation-based approaches [6], this represents a significant improvement in productivity.

To validate the optimality and accuracy of our approach, achievable power figures using various methods are compared to those of the reference DS-FS approach. Figure 5 shows that our cost function used for optimization correlates well with the final gate-level power numbers. Nevertheless, the DS-VP method results in up to a 5% difference in power consumption, which is a downside of achieving large gains in design time. The DS-ES method also exhibits a small 0.5% difference in some isolated cases where it is not able to guarantee the optimal solution. We also compared fine-grain DS-based methods against dynamic scaling with coarse optimizations, i.e. using a single word length for all variables (CS-FS). Power numbers using fine-grain scaling are always the same or smaller with a reduction of up to 13.6% even considering additional overhead for control at finer granularity.

In terms of overhead, compared to a method with no scaling (NS-FS), the extra power consumption for dynamic scaling is less than 3% according to our synthesis results. This overhead is small compared to the average 17% power reduction that can be achieved by dynamic scaling across varying input SNR levels. At SNR levels that are lower than the required SNR, the power numbers are larger than those for NS-FS due to the overhead of finely tuned dynamic scaling. The system, however, is not usually in such a poor environment. Hence, on average, large power savings can be expected.

Finally, Figure 6 and Figure 7 plot the results of performance simulations. Using DS-type methods, the system is able to maintain the targeted output BER over the full input SNR range leading to a large power reduction at higher SNR values. In Figure 7, the BER of floating point model (FP) is also plotted as a reference. The measured BER for a targeted BER of 0.01% ranges from 0.004% to 0.014% using our DS-VP method. Note that while in some cases the power consumption can be lower than in other DS-based methods, this comes at the cost of violating the BER constraint for those operation points. This mismatch is caused by the heuristic nature of our optimization approach.

IV. SUMMARY AND CONCLUSIONS

In this paper, we introduce a statistical analysis scheme using variance propagation for word length optimization. A fine-grain optimization of word lengths for dynamic scaling is possible and results in significant power savings. A static design-time approach avoids run-time overhead and the need for time-consuming exhaustive simulations. In the future, we plan to generalize our method to other types of operations and blocks in DSP systems, including optimization for other metrics, such as coded BER. Furthermore, we plan to automate the approach, including generation of optimized HDL code and clock-gating logic within our flow.

ACKNOWLEDGMENTS

The authors would like to thank Kamran Saleem and Jingwen Li of The University of Texas at Austin for their help and support in making this work possible.

REFERENCES

[1] J. Kim and S. Yoshizawa, and Y. Miyanaga, "Dynamic wordlength calibration for energy reduction FFT processors in wireless LAN," *ISCAS*, 2011.

[2] M.M. Nisar and A. Chatterjee, "Environment and process adaptive low power wireless baseband signal processing using dual real-time feedback", *Conf. on VLSI Design*, 2009.

[3] D. Novo, at al., "Scenario-based fixed-point data format refinement to enable energy-scalable software defined radios", *DATE*, 2008.

[4] H.-N. Nguyen and D. Menard, and O. Sentieys, "Energy reduction in wireless system by dynamic adaptation of the fixed-point specification," *DASIP*, 2008.

[5] K. Kum and W. Sung, "Combined word-length optimization and highlevel synthesis of digital signal processing systems," *IEEE TCAD*, vol. 20, no. 8, 2001.

[6] K. Han and B.L. Evans, "Optimum Wordlength Search Using Sensitivity Information," *EURASIP Journal on ASP*, vol. 2006, pp. 1-14, 2006.

[7] G. Constantinides, P. Cheung, and W. Luk, "Wordlength Optimization for Linear Digital Signal Processing," *IEEE TCAD*, vol. 22, no. 10, 2003.

[8] D. Menard, O. Sentieys, "Automatic evaluation of the accuracy of fixed-point algorithms," *DATE*, 2002.

[9] D.-U. Lee, et al., "Accuracy-guaranteed bit-width optimization," *IEEE TCAD*, vol. 20, no. 10, 2006.

[10] C. Shi and R.W. Brodersen, "Automated fixed-point data-type optimization tool for signal processing and communication systems," *DAC*, 2004.

[11] B. Widrow and I. Kollar, *Quantization Noise*, Cambridge University Press, 2008.

[12] ASA 25.15, 2004. http://www.ingber.com/#ASA