A Polynomial Time Algorithm for Solving the Word-length Optimization Problem

Karthick N. Parashar
INRIA, University of Rennes-1,
Imperial College, London
Email: k.parashar@imperial.ac.uk

Daniel Menard
INRIA, University of Rennes-1,
INSA, IETR, UEB
Email: daniel.menard@insa-rennes.fr

Olivier Sentieys
INRIA, University of Rennes-1
Email: olivier.sentieys@inria.fr

***Abstract*—Trading off accuracy to the system costs is popularly addressed as the word-length optimization (WLO) problem. Owing to its NP-hard nature, this problem is solved using combinatorial heuristics. In this paper, a novel approach is taken by relaxing the integer constraints on the optimization variables and obtain an alternate *noise-budgeting* problem. This approach uses the quantization noise power introduced into the system due to fixed-point word-lengths as optimization variables instead of using the actual integer valued fixed-point word-lengths. The *noise-budgeting* problem is proved to be convex in the rounding mode quantization case and can therefore be solved using analytical convex optimization solvers. An algorithm with linear time complexity is provided in order to realize the actual fixed-point word-lengths from the noise budgets obtained by solving the convex *noise-budgeting* problem.**

I. Introduction

Fixed-point representations of numbers are used ubiquitously in modern electronic gadgets. Efficient design of fixed-point operations have profound impact on the quality of design. A well designed system in fixed-point arithmetic is characterized by minimal cost of implementation while not compromising on the quality of computation beyond certain acceptable limits. This trade-off between accuracy and cost has been played in order to achieve multiple design objectives such as minimizing energy dissipation, reducing total area or improving throughput of the system.

In spite of good understanding of the quantization effects, designers often spend about 25% to 50% [1] of the design time in striking a good compromise between the implementation cost and the degradation of performance of the system. Ad hoc optimization strategies are considered the main reason for such delays. Moreover, the fixed-point refinement process is an iterative optimization problem which is combinatorial in nature and is considered NP hard [2].

As the focus of fixed-point refinement shifts from mere utilization of available fixed-point platforms to the use of fixed-point formats for improving system costs, several attempts to solve the word-length optimization (WLO) problem have been attempted. Solutions proposed in order to solve them have been predominantly based on heuristics. With growing capacity to crunch more transistors per unit area, the complexity of systems being realised on silicon and embedded systems continues to increase. The WLO of such systems requires that each of the variables (of which there may be thousands) be assigned the right word-lengths. Indeed, existing algorithms do not scale up to the challenge. The best known fast *Min* $+1$ *bit* WLO algorithm [3] does not scale up with growing problem size [4] and are in general regardless of achieving the most optimal solution. Relevant background work and their limitations are discussed in Section II.

In this paper, an attempt to address both scalability and optimality of the solution to a WLO problem is made and a near-optimal solution to the word-length optimization (WLO) problem is obtained. The original WLO problem is transformed to an alternative formulation: the *noise-budgeting* problem by relaxing the integer constraint on the number of bits assigned to fixed-point numbers. In Section III, conditions for convexity of this problem are defined. The convex problem thus obtained is solved using a convex solver to obtain noise budgets to each of the optimization variables. In Section IV, the result obtained from solving the *noise-budgeting* problem is used to realise the fixed-point word-lengths that generates just as much quantization noise as budgeted. In Section V, the proposed technique is applied on several examples and the results obtained are compared against the classical greedy algorithm: *Min* $+1$ *bit* algorithm and concludes with a summary in Section VI.

II. Background and Previous Work

Solving the word-length optimization problem requires estimation of degradation in the accuracy of the system and the total cost of the system measured by functions $\lambda(\mathbf{w})$ and $C(\mathbf{w})$ respectively. A system with m fixed-point operations has a word-length vector $\mathbf{w}$ of length m whose values correspond to the fixed-point word-lengths of each of the operations. Therefore, the cost and performance estimation functions essentially map the m-dimensional vector to a positive real number, that is $\lambda(\mathbf{w}), C(\mathbf{w}) : \mathbb{N}^m \rightarrow \mathbb{R}^+$. The word-length optimization problem of a fixed-point system targeting minimization of the total cost of implementation is formally stated as

$$\min\left(C\left(\mathbf{w}\right)\right) \quad \text{subject} \quad \text{to} \quad \lambda\left(\mathbf{w}\right) \leq \lambda_{obj}, \tag{1}$$

where, λ_{obj} is the objective accuracy, i.e. maximum quantization noise power above which the performance of the system under consideration is not acceptable. To be precise, the WLO problem described in Eq. 1 is the cost minimization problem under specified performance constraint. It is equivalently possible to describe a performance maximization problem under a given cost constraint. In this paper, the cost minimization problem is chosen to emphasise the importance of meeting certain performance criteria.

The authors would like to thank Dr. Prasad Sudhakar for helping them to get acquainted with *CVX* package for Matlab. He is currently a post-doctoral research associate at *Universite catholique de Louvain*, Belgium

978-1-4799-1071-7/13/$31.00 ©2013 IEEE

The WLO problem defined in Eq. 1 presents a vast combinatorial optimization space ($O(\mathbb{N}^m)$) that needs to be explored before arriving at optimal word-lengths. It may be noted here that the solution space increases exponentially with increasing size of the problem. The NP-hard nature [2] of the WLO problem makes it difficult to guarantee optimality unless the entire search space is explored.

The use of heuristics has been the primary technique for solving the word-length optimization problem thus far. These heuristics include greedy approaches [3] such as the popular *Min* $+1$ *bit* and *Max* -1 *bit* techniques. Genetic algorithms and simulated annealing approaches [5], [6] have also been experimented with. One of the main problems with greedy heuristics is that they are susceptible to be stuck in local minimas. In [13], a mix and match of greedy heuristics are explored to address this problem. The optimization algorithm is based on simulated annealing procedure. This algorithm has a good average case performance in terms of quality of the solution but could be very time consuming in comparison with classical greedy approaches.

In case of all such heuristic-driven algorithms including the simulated annealing approach, increase in the number of variables is known to cause an increase in the time taken for solving the optimization problem. Also, it gets more difficult to comment about the optimality with growing problem size. Therefore, this problem is more visible when complex systems with a large number of variables participate in the word-length optimization problem.

In [7], an approximate but an analytical framework for optimizing word-lengths are presented. The approach here is to trace a *Pareto-optimal* trade-off curve and hence be able to specify bounds on the achievable minimum system cost. In [8], an analysis of the gradient-based greedy approaches is made by applying the popular *Lagrange Multipliers* and the *Marginal Analysis* technique is proposed for word-length optimization. This technique is similar to the classical *Min* $+1$ *bit* algorithm except that the precision bits are incremented starting from 0 bits instead of minimum number of bits. In [9], the same problem is solved using Geometric Programming in order to avoid negative bit assignments to word-lengths.

A common trait in the above said analytical techniques is the assumption made to derive their performance and cost estimation models. By relaxing the integer value constraint on the word-length optimization variables, the classical word-length optimization problem is transformed to a convex optimization problem. The cost minimization problem is then written as

$$\min\left(\sum_{i=1}^{M} c_i w_i\right) \quad \text{subject to} \quad \frac{1}{3}\sum p_i 2^{-2w_i} \leq \lambda_{obj}, \quad (2)$$

where w_i is the number of bits corresponding to the i^{th} signal, p_i is the path gain from the i^{th} signal to the output and c_i is the cost of using w_i bits for the i^{th} signal. Clearly, the performance function measures the mean square error with the zero mean assumption and the cost function is proportional to the number of bits assigned to every fixed-point operation.

Another common trait in the analytical approaches is that the integer constraint on the number of bits w_i assigned to the i^{th} signal is relaxed and are allowed to take on real values. Due to this relaxation, the cost estimation function (objective function) of the minimization problem in Eq. 2 is convex when the weights are kept constant throughout the optimization process. Also, due the monotonicity and the convexity of the exponential function (2^{-w_i}), the constraint function is also convex.

It is important to consider the conditions under which the objective and constraints of the problem described in Eq. 2 are indeed convex.

A. The Objective Function

The proportionality to the number of fixed-point bits of the cost function is based on the premise that each signal emanates from a fixed-point operation and that each of them is assigned one fixed-point format. This is a fairly simplistic approach and the actual cost function is not so simple always. In a more practical scenario, it is not possible to express the cost function by scalar multiplication of weights as expressed in the minimization problem in Eq. 2.

Even if this was to be possible, the cost weights (ω_i) need not be the same for all possible word-length assignments. To illustrate this, consider the cost of a binary operator such as a two input adder. One way of implementing such a fixed-point adder is to perform addition by using a full adder circuit with as many bits as the maximum bits assigned to either inputs and then discard the resulting bits either by truncation or rounding. Suppose the average energy dissipated as a function of number of bits is used as the cost metric. The total cost C_a of using such an adder circuit depends on the number bits assigned to each of the inputs signals and it is written as

$$C_a \propto \max(w_1, w_2), \quad (3)$$

where $\propto$ represents proportionality and w_1, w_2 are the word-lengths assigned to the inputs of the adder. The objective function is the sum of $max()$ or in other words a convex function of a convex function. So, although the cost function in Eq. 2 is convex, it does not represent a realistic cost function.

B. The Constraint Function

There can be more than one quantization noise source along a given path to the output. The noise contribution by each fixed-point quantization noise source is not just a function of that particular fixed-point word-length. Consider the data path shown in Figure 1, the quantization noise power added by two quantizers is shown in the statistically equivalent additive PQN model.

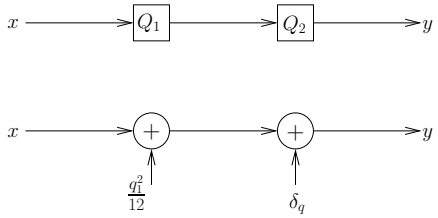


Fig. 1. Quantization noise sources along a data-path

The amount of quantization noise added by the second quantization in quantizer Q_2: δ_q, does not depend on the number of bits assigned at the output of Q_2 alone. It is in

deed a function of both quantization step sizes q_1, q_2 and is given as

$$\delta_q(q1, q2) = \begin{cases} \frac{q_2^2}{12} - \frac{q_1^2}{12} & q_2 > q_1 \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

Only when $q_1 << q_2$, the value of δ_q approaches the value of $\frac{q_2^2}{12}$.

Clearly, it is only under such conditions that the constraint function in the minimization problem formulation in Eq. 2 can be approximated to be convex by ignoring the noise contribution by the quantizer Q_1. In practice, this can happen in scenarios where the difference between two quantizers is very large such as a data-path where a multiplier is followed by an adder. In cases where the difference between the two step-size is as small as 1 bit, ignoring the quantization noise contribution by the first quantizer introduces an error of nearly 25% in the estimation.

In summary, the problem formulation as given in Eq. 2 would only work well if the quantization step sizes between successive quantization are relatively large and the cost function was as simple as a bit-count. It fails to capture all the nuances of the quantization dynamics with respect to both the cost objective and the performance constraint functions.

III. The Noise Budgeting Problem

Each fixed-point operation can potentially be a source of noise in the system. The word-length optimization problem can be thought of as an attempt to budget the total quantization noise power to each fixed-point operation. Using the noise-power q_i introduced into the system at the i^{th} fixed-point operation as the optimization variable, the noise budgeting problem for minimizing the cost of implementation can be written as

$$\min(C(\mathbf{q})) \quad \text{subject to} \quad \lambda(\mathbf{q}) \leq \lambda_{obj}, \quad (5)$$

where $\mathbf{q} = [q_1, q_2 \dots q_M]$ corresponds to the noise-power injected into the system by M different operations used to implement the fixed-point design.

The total cost is a simple sum of all the individual fixed-point operator costs. Since the cost of each operation is obtained as a function of the injected quantization noise power, it becomes necessary to study the trade-off behavior cost and noise-power of all types of operations in the sub-system for various choices of fixed-point configurations. As the cost of every fixed-point operation is non-negative, minimum cost of the total system is obtained when the cost of each operation is also minimized.

A. Operator-Level Trade-Off

In order to choose the *Pareto-optimal* cost, it is required to profile both the given operation under consideration for all possible fixed-point configurations. All basic operations used in the design and implementation practices of signal processing applications are simple with few inputs and outputs. Therefore, an exhaustive exploration of basic operations is not costly. Moreover, this exercise of characterizing operators needs to be done only once for a given library (such as a standard cell library) of fixed-point operations.

Energy dissipation cost of a fixed-point operator is obtained by looking up from its hardware library. The total energy dissipation cost of the given implementation E is obtained as

$$\mathrm{E} = \sum_{i=1}^{D} \mathrm{E}_{op}^{i} \cdot n_{op}^{i}, \quad (6)$$

where D is the number of different types of operations qualified by their fixed-point word-lengths and E_{op}^{i} is the corresponding energy dissipation obtained by looking up the library and n_{op}^{i} is the number of the i^{th} type of fixed-point operation.

The energy dissipation of binary adders and multipliers with various fixed-point configurations is build on the targeting the ASIC platform of $130nm$. The energy consumption estimates are obtained by using *Prime Time* from Synopsys [10] and the estimates obtained are in Joules. A large, random input test vector set which is uniformly distributed and which spans the entire range of the assigned binary fixed-point range is used. Therefore, the energy dissipation values corresponds to the average energy dissipated.

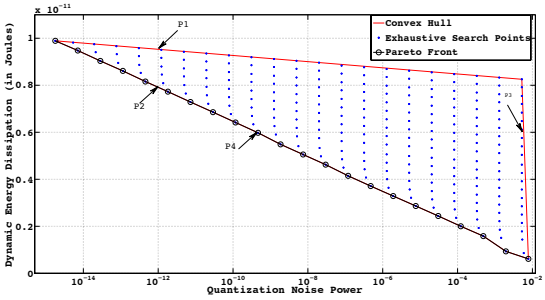


Fig. 2. Binary adder: cost vs. accuracy trade-off in the semilog scale

In Figure 2, all the points considered during an exhaustive search of a binary adder are shown. The y-axis shows the energy dissipation cost and the x-axis shows the quantization noise introduced by the operation for each of the points considered. An exhaustive search for all possible fixed-point operations are performed in the range of 2 to 24 bits assigned to the inputs and outputs of the adder.

B. Identifying Pareto front

The *convex-hull* is a convex polygon encompassing all the points considered during the exhaustive search. This polygon is drawn around the points plotted on a graph with linear scale on either axes. In Figure 2, the logarithmic scale for x-axis is chosen for the purpose of illustration and for clear visibility of all points.

The choice of points that form the Pareto front should be chosen such that they do not make sub-optimal choices. For example, consider four points P_1, P_2, P_3 and P_4 as shown in Figure 2. Although points P_1 and P_2 have the same noise power, P_2 is a Pareto point due to its lower cost. Similarly, between P_3 and P_4 that have the same cost, P_4 is the Pareto point due to lower quantization noise-power. Hence, those points that are closer to the origin are the ones representing the minimum cost for a given quantization noise-power. The line joining all such points mark the Pareto-front of the operation under consideration.

C. Relaxation for convexity

The *Pareto-front* boundary marked in Figure 2 is a continuous line defined by $\Phi(q)$ as a function of quantization noise power q. It is obtained by connecting successive points along the convex-hull with straight lines. These lines also happen to be the edges of the convex polygon. By definition of the convex hull, the function $\Phi(q)$ which traces a piece-wise linear curve is convex.

The value of quantization noise is discrete as the word-lengths can only be assigned integer values in practice. The idea here is to relax this constraint on q such that it can be assigned any real value on the *Pareto-front*. Then, if κ is the minimum cost of the operator for any given q. The value of κ can be obtained by looking up the *Pareto-curve* and is given as

$$\kappa = \Phi(q) \tag{7}$$

The cost function in the minimization problem in Eq. 8 is essentially the sum of individual operator costs. This cost metric is applicable to the energy dissipation cost of hardware designs in general. Given that the word-length optimization exercise is performed at a very high level, it is impossible to capture the impact of decisions such as scheduling and resource binding that are taken very late in the design cycle during high-level synthesis. The focus here is to use a first-cut estimate of the general trends to arrive at optimal word-lengths.

D. Convexity of the noise budgeting problem

In order to use convex optimization techniques to solve the minimization problem in Eq. 5, it is important to check if the problem considered is indeed a convex optimization problem. In the light of the previous section, minimization problem of i^{th} sub-system with N quantization noise sources can now be written as

$$\text{minimize} \left(\sum_{j=1}^{N} \Phi_i(q_i) \right) \text{ subject to } \quad \lambda(\mathbf{q}) \leq \quad \lambda_{obj}, \tag{8}$$

where $\lambda(\mathbf{q})$ is the total quantization noise at the output of the system, obtained from Eq. 10 and λ_{obj} is the target accuracy objective. The minimization problem is a convex optimization problem if both objective and constraint functions are convex.

From first principles, a function $f(x)$ is convex if the domain $dom(f)$ is a convex set and for all $m, n \in dom(f)$ and $\eta \in (0, 1)$ the following relation holds [11]

$$\eta \cdot f(m) + (1-\eta) \cdot f(n) \geq f(\eta \cdot m + (1-\eta) \cdot n) \tag{9}$$

1) Performance Evaluation Function: Using the linear noise propagation model described in [12], the total noise power at the output of the system is the sum of all quantization noise powers scaled by their respective path gains. The total noise at the output of the system as a function of the noise power vector $\mathbf{q}$ is given as

$$\lambda(\mathbf{q}) \quad = \quad \underbrace{\sum_{i=1}^{N} \beta_i \sigma_i^2}_{\sigma^2} + \underbrace{\left(\sum_{i=1}^{N} \alpha_i \mu_k \right)^2}_{\mu^2}, \tag{10}$$

where σ_i and μ_i are the variance and the mean of the quantization noise power q_i generated by the i^{th} operation. β_i and α_i are constants derived from the path function between i^{th} operation and the system output. The function $\lambda(\mathbf{q})$ is a function of all the noise-power generated within the system.

Let q_i^{σ} be the standard deviation of the quantization errors and $q_i^{\mu} = \sqrt{\mu_i}$ of the i^{th} operation. The noise source q_i consists of contribution from the respective variance and mean components as

$$\begin{aligned} q_i &= q_i^{\sigma} + q_i^{\mu} \\ &= \sigma_i^2 + \mu_i^2 \end{aligned} \tag{11}$$

The total quantization noise power function $\lambda(\mathbf{q})$ at the output of the system can also be split into two and expressed as the sum of two functions $\lambda^{\sigma}(\mathbf{q}^{\sigma})$ and $\lambda^{\mu}(\mathbf{q}^{\mu})$, where $\mathbf{q}^{\sigma} = [q_1^{\sigma}, q_2^{\sigma} \ldots q_N^{\sigma}]$ is a vector of the noise source components corresponding to contribution by noise variance component and $\mathbf{q}^{\mu} = [q_1^{\mu}, q_2^{\mu} \ldots q_N^{\mu}]$ is a vector of noise contribution by the mean component. The total noise-power as a function of the mean and variance components of the fixed-point operation noise-sources is written as

$$\begin{aligned} \lambda(\mathbf{q}) &= \lambda^{\sigma}(\mathbf{q}^{\sigma}) + \lambda^{\mu}(\mathbf{q}^{\mu}) \\ &= \sum_{i=1}^{N} \beta_i q_i^{\sigma} + \left(\sum_{i=1}^{N} \alpha_i \sqrt{q_i^{\mu}} \right)^2 \\ &= \sum_{i=1}^{N} \beta_i q_i^{\sigma} + \sum_{i=1}^{N} \sum_{k=1}^{N} \alpha_i \alpha_k \sqrt{q_i^{\mu} q_i^{\mu}} \end{aligned} \tag{12}$$

If the function $\lambda(\mathbf{q})$ is convex, it has to satisfy the condition for convexity in Eq. 9. The expression corresponding to the right-hand-side (RHS) of the convexity condition is written as

$$\begin{aligned} \lambda(\eta\mathbf{m} + (1-\eta)\mathbf{n}) \quad = \quad & \sum_{i=1}^{N} \beta_i (\eta \cdot m_i^{\sigma} + (1-\eta) \cdot n_i^{\sigma}) + \\ & \sum_{i=1}^{N} \alpha_i \sqrt{(\eta \cdot m_i^{\mu} + (1-\eta) \cdot n_i^{\mu})} \cdot \\ & \sum_{k=1}^{N} \alpha_k \sqrt{(\eta \cdot m_k^{\mu} + (1-\eta) \cdot n_k^{\mu})}, \end{aligned} \tag{13}$$

where $\mathbf{m} = [m_1, m_2, \ldots, m_N]$ and $\mathbf{n} = [n_1, n_2, \ldots, n_N]$ are two combinations of the quantization noise power source vector $\mathbf{q}$ such that it is an optimal solution to the noise budgeting problem for two corresponding accuracy constraints λ_{obj}^m and λ_{obj}^n respectively. The gain $\beta_i = \mathrm{E}[h_i^2]$ is the expectation of the impulse response of the path function from the i^{th} source to the output. Therefore, β_i cannot be negative. The *square-root* operation for the mean part makes the noise estimation function non-convex. Therefore, in general the constraint function is not convex. In the case of convergent rounding mode, the mean of the quantization noise is exactly zero. It is very close to zero even if it is the case of simple rounding mode. That is, the noise contribution due to the mean is either zero or is negligibly small in the rounding. In such a scenario, continuing with evaluation of the expression in Eq. 13 it can be written as

$$\begin{aligned} \lambda(\eta\mathbf{m} + (1-\eta)\mathbf{n}) &= \eta \left(\sum_{i=1}^{N} \beta_i m_i^{\sigma} + \right) + (1-\eta) \left(\sum_{i=1}^{N} \beta_i n_i^{\sigma} \right) \\ &= \eta \cdot \lambda(\mathbf{m}) + (1-\eta) \cdot \lambda(\mathbf{n}) \end{aligned} \tag{14}$$

Therefore, it can be concluded that the accuracy evaluation function in the rounding case is not only convex but is also affine. Considering the result in Eq. 14, the technique described henceforth is strictly applicable only to quantization carried out in the convergent rounding mode. However, as the magnitude of rounding is very small in the simple rounding case, this result approximately holds true even for simple rounding mode. In the truncation mode, the noise power contribution by the mean component is comparable to the variance component. Therefore, this proposal is not applicable for truncation mode quantization.

2) Cost Function: In this section, the cost function defined in Eq. 8 will be proved to be a convex function. The function $\Phi(x)$ is obtained by an exhaustive profiling of the operator and considering the Pareto-front obtained by constructing a convex polygon around the points thus obtained on the cost vs. accuracy axes. Therefore, it is convex by definition. The convexity of the function $\kappa_i = \Phi_i(q_i)$ for every operation i implies that

$$\Phi_i(\eta \cdot q_i^m + (1-\eta)\cdot q_i^n) \quad \leq \quad \eta \cdot \Phi_i(q_i^m) + (1-\eta)\cdot \Phi_i(q_i^n) \tag{15}$$

Now, consider evaluating the right-hand-side of Equation 9 with respect to the cost estimation function $C(\mathbf{q})$.

$$\begin{aligned} C(\eta\mathbf{m} + (1-\eta)\mathbf{n}) &= \sum_{i=1}^{N} \Phi_i(\eta q_i^m + (1-\eta) q_i^n) \\ &\leq \sum_{i=1}^{N} \eta\Phi_i(q_i^m) + (1-\eta)\Phi_i(q_i^n) \\ &= \eta \sum_{i=1}^{N} \Phi_i(q_i^m) + (1-\eta)\sum_{i=1}^{N} \Phi_i(q_i^n) \\ &= \eta C(\mathbf{m}) + (1-\eta) C(\mathbf{n}) \end{aligned} \tag{16}$$

Therefore, the cost function at the sub-system level is convex. Summarizing results obtained in Eq.13 and Eq.16, it can be concluded that when rounding mode quantization is used, the *noise-budgeting* problem defined in Eq.8 is convex.

IV. Word-length Optimization Algorithm

Using the relaxation technique discussed in the previous section, the problem of cost minimization subject to accuracy constraint of a fixed-point system is relaxed to obtain a convex optimization problem. In this section, a word-length optimization algorithm described in Algorithm 1 that captures the various steps of the relaxation process in order to use standard convex optimization solvers and apply the result thus obtained to determine the actual word-lengths is presented.

The first step in solving the word-length optimization problem is to obtain the data flow graph $S(V, E)$ consisting of V nodes and E edges by calling the function **GetSystemGraph()**. The graph S is a directed graph with one node corresponding to every operation, the edges connect them and point in the direction of the data-flow in the algorithm. The various types of operation are enumerated for studying the cost and accuracy trade-off behavior. The function **ExtractOperatorPoints()** conducts an exhaustive search of the operator and returns all the feasible operating points to be stored in the operator database *DB*. As described in Section III-C, the convex

Algorithm 1 : Word-length Optimization

1: $S_i(V, E)$ = GetSystemGraph()
2: N_t = GetOperatorTypes($S_i(V, E)$)
3: **for all** n_i Operator types $n_i \in N_t = [n_1, n_2, \ldots, n_t]$ **do**
4: DB_j = ExtractOperatorPoints(n_i, Wd_{Min}, Wd_{Max})
5: Φ_j = GetConvexParetoFront(DB_j)
6: **end for**
7: C_i = GetCostExpression($S(V, E)$) $\backslash*$ *Sum of operator costs* $*\backslash$
8: λ_i = GetNoisePowerExpression($S(V, E)$)
9: $\mathbb{P}$ = ConstructNoiseBudgetingProblem(DB,λ_{obj},C_i, λ_i)
10: $\bar{q}_{opt}$ = Solve($\mathbb{P}$)
11: $\bar{Wd}_{opt}$ = GetFinitePrecisionWordlengths($\bar{Wd}_{in}$, $\bar{q}_{opt}$, $\bar{q}_{opt}$, P_{obj},DB)

Pareto-front Φ_i of every type of operation is deduced from the trade-off points by the function **GetConvexParetoFront()**.

Analytical expressions for the sub-system cost and noise power are determined by calls to functions **GetCostExpression()** and **GetNoisePowerExpression()** respectively. Noise-budgeting problem $\mathbb{P}$ is expressed using the standard optimization modeling language such as *"CVX"*. The procedure **Solve()** uses a standard solvers such as *"SeDumi"* or *"SDPT3"* (used in this paper) to solve the minimization problem $\mathbb{P}$. Solving the convex optimization problem, the minimum cost of implementation is obtained such that the performance constraint is satisfied. The values of noise powers of each operation which gives the minimum cost are obtained in the vector $\bar{q}_{opt}$. The individual operation cost can be deduced by performing an inverse of the operator level Pareto curve (i.e. $\Phi_i(q_i)$ for the i^{th} operation). In the final step, the procedure **GetFinitePrecisionWordlengths()** realises the budgeted optimal noise-power using fixed-point operations.

A. Realising Noise-power Using Fixed-Point Operators

The value of q_i obtained as a solution to $\mathbb{P}$ lies on the pareto-front of each operation. If the optimal noise-power suggested by the convex optimization solver coincides with one of the feasible points in the operator data-base, the operator precision at its output is precisely found. If that is not the case, the nearest feasible point is chosen such that it does not affect the sub-system output accuracy constraint greatly.

The process of resolving the solution lying on the *Pareto-optimal curve* to obtain the actual word-length in case of an adder is shown in Figure 3.

Consider a particular case of the adder where the quantization noise assigned by the convex optimization process is 3×10^{-4}. The minimum cost as attained by the convex optimization algorithm is obtained by calculating the inverse of the convex-Pareto curve as $C_{cvx} = \Phi(3 \times 10^{-4})$. Clearly, this is not a feasible point. Further, suppose the input constraints to this operation is $(7, 7)$ (i.e. both inputs are quantized by 7 bits). With the input constraints applied, it is clear that the trade-off point must lie on the vertical line corresponding to the input constraints $(7, 7)$. After translation, the cost of the operation is $C_{constraint}$. The point is still non-feasible as fractional word-length assignment is not possible. Therefore, the nearest feasible point satisfying the noise-power constraint

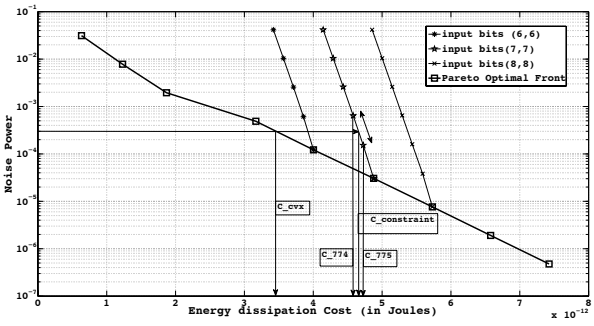


Fig. 3. Finding feasible operating points

is chosen. Two points $(7, 7, 4)$ and $(7, 7, 5)$ exist in the vicinity corresponding to this quantization noise-power.

Here, the cost $C_{cvx} < C_{774} < C_{constraint} < C_{775}$. Clearly, C_{cvx} is not a feasible choice. The choice of the point C_{775} satisfies the budgeted quantization noise power but the cost incurred in the process is higher. On the other hand, the choice of point C_{774} does not satisfy the quantization noise-power constraint.

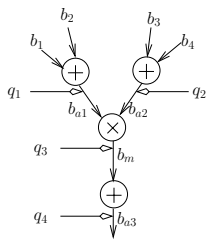
Among the two options, it is safe to take a conservative approach keeping in mind the accuracy satisfiability condition. An algorithm which always makes the conservative choice for word-length determination is presented in Procedure 1a.

Procedure 1a : GetFinitePrecisionWordlengths()

1: $T(V)$ = TopologicalSort($S_i(V, E)$)
2: **for all** $v_i \in T(V)$ **do**
3: DB_i = GetOperatorDB(v_i)
4: $[Wd_i]$ = LookupDB(DB_i, q^i_{opt}, $\bar{W}d^i_{in}$)
5: **end for**

In order to satisfy the input constraints and the propagation of bit-widths across operators, it is important that all operations in the sub-system graph is topologically sorted such that the operation whose input constraints are already known are considered first. Resolving the output fixed-point bit-widths of these operations in turn generates the input constraints for the successive operations.

The function **GetOperatorDB()** returns the database corresponding to the given operator type. The conservative choice of word-lengths is made using the function **LookupDB()**.



b_{a1}	b_{a2}	b_m	
b^l_{a1}	b^l_{a2}	b^1_m	$q_{b_1,b_2,b^l_{a1}} < q_1$
b^l_{a1}	b^h_{a2}	b^2_m	$q_{b_1,b_2,b^h_{a1}} \geq q_l$
b^h_{a1}	b^l_{a2}	b^3_m	$q_{b_3,b_4,b^l_{a2}} < q_2$
b^h_{a1}	b^h_{a2}	b^4_m	$q_{b_3,b_4,b^h_{a2}} \geq q_2$

Fig. 4. Propagating bits across noisy operators

In the light of Procedure 1a, consider determining the word-lengths of the operators in the graph shown in Figure 4. The input constraints to the sub-system are user defined. Thus, the bits b_1, b_2, b_3, b_4 in the Figure 4 are user given. The values of the noise contribution q_1, q_2, q_3 of three operations are obtained by solving the convex optimization problem. In order to realise the noise-power q_1, two feasible points $q_{b_1,b_2,b^l_{a1}}$ and $q_{b_1,b_2,b^h_{a1}}$ around the value of q_1 represents the greater and smaller fixed-point word-length assignments possible respectively. The actual choices made is marked along the tree shown in Figure 5. In this figure, the word-length option chosen are indicated by taking the left edge at any node. On the contrary, if the right edge were to be chosen consistently, the right most node would have been the choices. It is clear that the optimum choice for fixed-point word-lengths lies somewhere between choosing the left edge or the right edge.

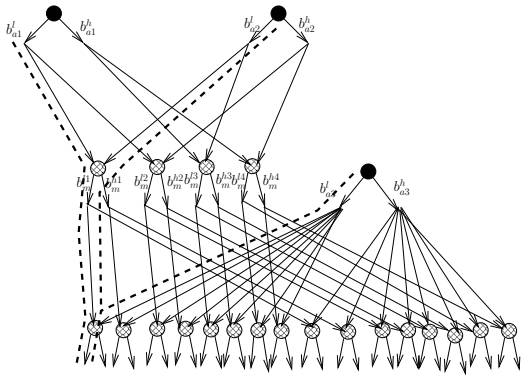


Fig. 5. Choosing the left-edge always

B. Complexity Analysis

In the classical *min* $+1$ *bit* WLO algorithm, the fixed-point operation is free to take on any of the N fixed-point word-lengths. So, the combinatorial search space defined by m variables consists of as many as N^m different unique combinations. In the proposed *noise budgeting* problem, the relaxed cost vs. performance curve for every fixed-point operation is used to determine the optimal noise-budgets such that the cost of the system on the whole is minimum. Therefore, the search space of the *noise-budgeting* problem is reduced to an m-dimensional real space: $\mathbb{R}^m$. The convex solver uses one of several algorithms with polynomial time complexity such as [14] for solving the convex optimization problem. The solution thus obtained provides a technique to propagate the word-lengths from input to the output of the system while satisfying the budgeted noise. In Algorithm 1, every operation node, there are two choices available dictated by the input constraints and the noise-budget assigned. Thus, while performing the *near-optimal* word-length optimization procedure, the complexity of the search space is reduced to $O(2^m)$.

This reduction is by a factor of $\frac{N}{2}^m$.When both N and m are large, the reduction in search space is of several orders in magnitude. The choice of a conservative word-length at every stage corresponds to the *left-edge* traversal of the graph. The complexity of the word-length assignment needs to count the time required for a topological sort and a one time traversal on such a sorted list. Therefore, the proposed word-length optimization algorithm has a complexity of $O(m)$. This complexity

is lesser than the non-linear complexity [14] of the convex solvers. Therefore, it can be concluded that the proposed technique for WLO has a polynomial time complexity.

C. Comparison with previous work

In the proposed approach, the *Pareto-front* of the fixed-point operation is relaxed instead of relaxing the variables of the original problem to construct the convex *noise-budgeting* problem. As a consequence, the actual noise introduced into the system is decoupled with the actual generation of such errors. It is therefore possible to genuinely work with a noise model abstraction without making approximations to the performance estimation function. The performance function in the case of rounding quantization is shown to be linear (and hence convex). The inclusion of non-zero mean errors makes the problem non-convex. This limitation is also present in previous works discussed in Section II and is reflected in the way the performance function is constructed in those.

The *Pareto-front* is convex by definition as it is obtained by constructing a convex hull around all feasible trade-off points for a given fixed-point operation. The cost function is also convex by the way it is defined (as a sum of individual operator costs). Therefore, as long as the actual noise introduced into the system is zero mean, and it is not expensive to invest in a one time effort for deriving the *Pareto-front* of fixed-point operations, the generality of the proposed technique is not compromised.

V. Results

In order to illustrate the efficacy of the proposed word-length optimization method, it is imperative to showcase its scalability and usability in different scenarios. In this paper, the radix-2 FFT algorithm is chosen to illustrate the scalability of the algorithm and the QR decomposition of a given matrix using CORDIC rotations is considered to illustrate the usability of the algorithm. It may be noted that these signal processing blocks are used in a number of popular communication algorithms such as MIMO/ OFDM based systems. The time taken and the quality of results obtained by application of the proposed convex optimization and the *Min* $+1$ *bit* greedy heuristic are compared.

A. Characterizing Operations

In order to apply the proposed convex optimization technique, different fixed-point operations need to be exhaustively characterized and the *Pareto-front* of each of these operations are to be obtained. The binary adder and the binary multiplier are most commonly used in the design of signal processing systems. In case of LTI (linear, time-invariant) systems, a number of constant multipliers are encountered. So, it is important to characterize each constant multiplier for different input and output fixed-point formats. The range of fractional bits assigned to the adder and multipliers is chosen to be between 2 and 24 bits for these experiments.

The QR decomposition consists of several CORDIC operations in vector and rotation modes [15]. The CORDIC operation consists of several additions and scaling operations and is also followed by decision in both modes. The presence of decision operation makes it difficult to arrive at an analytical formula for the error due to quantization at the output of one CORDIC operation. However, the simulation of CORDIC operations in both vector and rotation modes is not complex. Therefore, CORDIC operations are considered to be one among basic operations such as adders and multipliers in this work.

B. Optimization with CVX

The main task here is to write the routines for evaluating performance and cost evaluation as a function of the vector $\mathbf{q}$ in each case. It has to be written such that an analytical expression can be constructed by the CVX environment as it parses the convex optimization problem. A detailed guide for this coding style is given by the authors of CVX in [16].

Once the optimal noise power distribution is available, the assigned quantization noise is realised using the procedure described in Section IV. The total quantization noise and the cost is evaluated after all the operations have been assigned a fixed-point format. These experiments were realised using MATLAB. While the *Min* $+1$*bit* algorithm takes of the order tens of minutes time, the proposed approach finishes in less than a minute even in the case of FFT-64 (with 576 variables). The results corresponding to FFT and QR decomposition algorithms are presented below.

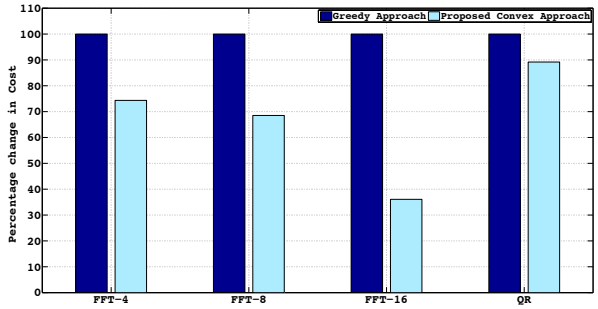


Fig. 6. Comparisons: cost of implementation

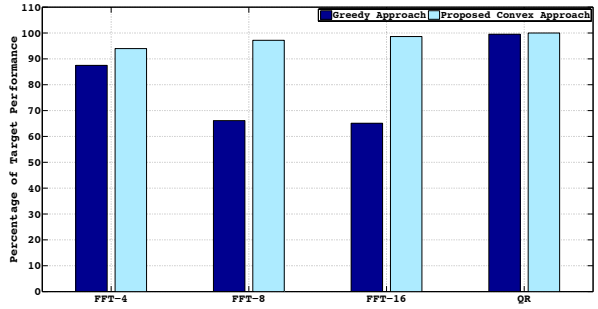


Fig. 7. Comparisons: performance achieved

The relative cost of fixed-point system obtained using the *Min* $+1$ *bit* algorithm and by application of the proposed noise-budgeting framework are depicted in Figure 6. Clearly, the proposed approach out-performs the *Min* $+1$ *bit* algorithm. The cost improves with the system size in case of the FFT algorithm. The QR algorithm is executed on a 4×4 matrix (typically representing 4×4 MIMO channel coefficients) and the

optimization is performed using Pareto-fronts of the CORDIC operator. Therefore, the number of optimization variables in the QR algorithm is reduced to 18 which is very small and hence the minimum cost achieved is comparable with that of the *Min* $+1$ *bit* algorithm.

The actual quantization noise achieved by both the processes is shown in Figure 7. The desired quantization noise power generated by the fixed-point design is set to 100%. The actual quantization noise power is achieved and it is usually lower than the desired noise power. In all the cases shown in Figure 7, the quantization noise power target is nearly reached by the convex optimization framework. Whereas, the classical approach tends to over-optimize. This is due to the fact that the classical approach moves from one fixed-point format to the next and thereby accruing sub-optimality in every iteration.

The use of *Pareto-front* for optimization and then searching for the fixed-point solution in the locality of the solution plays a key role in achieving the improvement in cost. The greedy heuristic latches onto a feasible point in every iteration. Some choices made during early iterations can cause a cascading effect and the choice of word-lengths could become grossly sub-optimal. It is impossible to determine whether a suboptimal choice during one of the iterations would help approach a better choice closer to optimality. On the other hand, in the convex optimization approach, the *Pareto-front* is used and the actual fixed-point determination is carried out as a final step. By then the actual quantization noise power from each of the sources is well established. It has to be noted here that if fractional word-length assignments were to be possible, then the solution obtained by the convex optimization problem solver is truly optimal. The sub-optimality factor creeps into the proposed convex optimization framework when the noise power is realised as fixed-point operators. In this paper, a conservative approach is chosen to determine the word-lengths as suggested in Section IV.

VI. Conclusion

In this paper, a convex optimization based approach is proposed for solving the word-length optimization (WLO) problem. An alternate problem fomulation: the *noise-budgeting* problem corresponding to the given WLO problem is obtained by relaxing the integer constraint on word-length assignments. By using an exhaustive search of the basic operations involved, a convex Pareto-front is obtained for each of the basic operations used in the system. The *noise-budgeting* problem is expressed using the CVX [16] tool. A general purpose convex solver such as *"SeDumi"* or *"SDPT3"* is used to arrive at optimal noise budgets for each fixed-point operation. An algorithm whose complexity is as low as $O(n)$ is proposed for realising the assigned quantization noise budgets for corresponding fixed-point operations. The proposed approach for solving the WLO problem derives advantage from fast convex solvers which are way faster than iterative combinatorial solvers. Therefore, the proposed technique scales well with growing optimization problem size. Realising the noise budgets obtained by solving the *noise-budgeting* problem requires to explore a search space of $O(2^m)$. This is a huge reduction in comparison to the search space of the original WLO problem which is $O(N^m)$. The shrinkage in search space is several orders of magnitude in practical cases. Therefore, this approach tends to perform better in comparison to greedy heuristics with growing system sizes (large m). Indeed, a branch-and-bound algorithm instead of the conservative approach could potentially lead to a better choice which could be much closer to optimality. In this paper, the proposed technique for WLO is applied on two algorithms: FFT and QR decomposition used commonly in wireless communication and signal processing applications. The results obtained suggest improvement in the cost achieved measured in terms of the systems energy efficiency. The time spent on optimization is also reduced due to the use of analytical convex solvers and the low complexity algorithm for translating the noise budgets to fixed-point word-lengths.

References

[1] M. Clark, M. Mulligan, D. Jackson, and D. Linebarger, "Accelerating Fixed-Point Design for MB-OFDM UWB Systems," *CommsDesign*, Jan. 2005.

[2] G. A. Constantinides and G. Woeginger, "The complexity of multiple word-length assignment," *Applied Mathematics Letters*, vol. 15(2), pp. 137–140, 2002.

[3] M.A. Cantin, Y. Savaria, D. Prodanos, and P. Lavoie, "An automatic word length determination method," in *The 2001 IEEE Int. Symp. on Circuits and Systems, ISCAS* , vol. 5, 2001, pp. 53 –56

[4] K. Parashar, R. Rocher, D. Menard, and O. Sentieys, "A hierarchical methodology for word-length optimization of signal processing systems," in *VLSI Design, 2010. VLSID '10. 23rd Int, Conf. on*, 2010, pp. 318–323.

[5] T. Arslan and D. Horrocks, "A genetic algorithm for the design of finite word length arbitrary response cascaded iir digital filters," in *Int. Conf. on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA.)* , Sep. 1995, pp. 276 –281.

[6] G. Baicher, "Optimization of finite word length coefficient iir digital filters through genetic algorithms - a comparative study," in *Advances in Natural Computation*, ser. Lecture Notes in Computer Science, vol. 4222. Springer Berlin / Heidelberg, 2006, pp. 641–650.

[7] P. Fiore, "Efficient Approximate Wordlength Optimization," in *IEEE Transaction on Computers*, vol. 57, no. 11, 2008, pp. 1561 –1570.

[8] S. C. Chan and K. M. Tsui, "Wordlength determination algorithms for hardware implementation of linear time invariant systems with prescribed output accuracy," in *IEEE Int. Symp. on Circuits and Systems, ISCAS* , May 2005, pp. 2607 – 2610 Vol. 3.

[9] S. C. Chan and K. M. Tsui, "Wordlength optimization of linear time-invariant systems with multiple outputs using geometric programming," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 4, pp. 845 –854, april 2007.

[10] Synopsys, "Synopsys prime time suite," *http://www.synopsys.com*.

[11] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. , Cambridge Univ. Press, UK, 2004.

[12] D. Menard, R. Rocher, and O. Sentieys, "Analytical fixed-point accuracy evaluation in linear time-invariant systems," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 55, no. 10, pp. 3197 – 3208, nov. 2008.

[13] H.-N. Nguyen, D. Menard, and O. Sentieys. Novel algorithms for word-length optimization. In *19th European Signal Processing Conference (EUSIPCO 2011)*, pages 1944–1948, 2011.

[14] K. C. Toh, M. J. Todd, and R. H. Ttnc, "Sdpt3 a matlab software package for semidefinite programming, version 1.3," *Optimization Methods and Software*, vol. 11, no. 1-4, pp. 545–581, 1999.

[15] P. Meher, J. Valls, T.-B. Juang, K. Sridharan, and K. Maharatna, "50 years of cordic: Algorithms, architectures, and applications," *IEEE Transactions on Circuits and Systems I: Regular Papers* , vol. 56, no. 9, pp. 1893–1907, 2009.

[16] M. C. Grant and S. P. Stephen P. Boyd Boyd, *The CVX Users Guide*. Cambridge Univ. Press, UK: CVX Research, Inc., 2012.