

A Symbolic Noise Analysis Approach to Word-Length Optimization in DSP Hardware

Arash Ahmadi Mark Zwolinski

Electronic System Design Group

School of Electronics and Computer Science University of Southampton

{aa03r,mz}@ecs.soton.ac.uk

Abstract—This paper addresses the problem of choosing different word-lengths for each functional unit in fixed-point implementations of DSP algorithms. A symbolic-noise analysis method is introduced for high-level synthesis of DSP algorithms in digital hardware, together with a vector evaluated genetic algorithm for multiple objective optimization. The ability of this method to combine word-length optimization with high-level synthesis parameters and costs to minimize the overall design cost is demonstrated by example designs.

I. INTRODUCTION

The main objective of High Level Synthesis (HLS) is to find the optimal design in terms of area, latency, throughput, and power consumption. Data Word-Length (WL) is one of the parameters that influences these metrics. In custom hardware implementations there is freedom for the WL to be chosen optimally for different points of the hardware. Despite the simplicity of the idea, designers face difficulties choosing the best WL in complicated systems, thus 50% of the design time may be spent on WL determination [1].

Optimization approaches based on Linear Programming (LP) have execution times that increase exponentially with design complexity. In general, WL optimization is an NP-hard problem [2], making exact methods impractical in the case of real designs.

The objective of this work is to introduce a new method of WL optimization. In this approach, a Symbolic Noise Analysis (SNA) method is used to analyze the computational error at every point of the hardware, without restrictive assumptions about the statistical model of the signals. This model is applied to a Multi-Objective Optimization (MOO) method to find the minimal WL at each point in the hardware implementation of Digital Signal processing (DSP) algorithms.

The paper is organized as follows: section II provides a review of related work; the proposed computational error model is presented in section III; section IV is devoted to a very brief review of the cost functions the implementation of the synthesizer and synthesis results are reported in section V.

II. BACKGROUND

In [3] a heuristic WL optimization method is introduced to tradeoff system area against Signal Quantization-Noise Ratio (SQNR). This is a stimuli based method which utilizes a reference floating point computation of the algorithm while the final

WL optimization is conducted using the synthesized hardware models. In [4], a combined method of static and dynamic analysis is proposed which employs an interval propagation analysis for range width determination and a simulation-based method for precision bit-width optimization. Nayak et al. in [5] present a compiler that takes high-level signal processing algorithms described in MATLAB and generates optimized hardware in which data range optimization is performed by a data range propagation technique. Their results show significant reductions in hardware costs. In [6] and [7] methods based on analytical digital noise analysis are proposed which are more suitable for Linear Time Invariant (LTI) systems; however, these methods are extended to WL optimization for nonlinear systems in [8] and [9]. These methods exploit this fact that the fixed-point implementation of an algorithm is a weak perturbation of its high precision specification.

Several works report applications of symbolic analysis in computational error analysis. A basic implementation of this method is known as Interval Arithmetic (IA) and Affine Arithmetic (AA) which perform a symbolic error analysis on the algorithm [10]. In this method, dependencies of the noise sources are taken into account in a parametric representation of the error at different points in the Data Flow Graph (DFG). In [11] Lee et al. implemented an AA-based method which categorizes the problem into two parts, range analysis and precision evaluation. The former gives the integer part of the data whereas the latter provides the fractional part of the numbers in every point on the DFG. Similar to this work a study is reported by Pu and Ha in [12] which applies AA with a different heuristic. In the later work, inspired from [13], by applying the central limit theorem the first and second moments of the output noise are approximated from the symbolic representation of the output noise.

Our work introduces a static analysis which is a combination of the symbolic data range analysis and noise analysis, called SNA. This accuracy evaluation method is combined with a multi-objective optimization method in which the objectives are circuit area, latency, power consumption and digital noise, all integrated in a Vector Evaluated Genetic Algorithm (VEGA). The contributions of this work are: 1) Merging noise based analysis with symbolic range analysis to characterize the computational noise analytically as well as statistically; 2) Correcting the round-off noise model for multiple WL implementations in shared hardware designs; 3) Introducing

a GA method for WL which integrates the HLS with WL optimization for a variety of nonlinear applications; and 4) Combining WL with area, power consumption and delay in a multi-objective optimization and design method.

III. WORD-LENGTH: CAUSES AND EFFECTS

In the digital representation of data, reducing the data bit-width has a direct effect on the accuracy, which is construed as computational error or noise. From this viewpoint, WL optimization methods can be categorized as error range analysis or noise analysis. The former approach considers how the maximum/minimum values of the signals propagate through the system from inputs to the output(s). Accordingly, the result of the analysis is the range of the output error. Several methods are introduced in this category such as IA [1], AA [11] and the Taylor Model [14]. These sub-categories are altered in the way of their range representation and approximation. In the noise analysis approach, on the other hand, the outcome of the accuracy reduction is represented as a random process, which also called computational noise. Different characteristics of the computational noise has been inspected so far and they commonly assumed to be White Sense Stationary (WSS) signals [15]. Inspired by analogue signal processing, most of the existing works utilize the Signal-to-Noise Ratio (SNR) error criterion as the accuracy cost.

In our proposed method a partially known quantity x is represented in SNA form as in Equation(1).

$$x = T_N(\vec{E}), \quad (1)$$

where $T(\cdot)$ is a polynomial of order N with M known coefficients (x_1, x_2, \dots, x_M) ; and \vec{E} is an array as in Equation (2).

$$\vec{E} = [x, \varepsilon_1, \varepsilon_2, \dots, \varepsilon_m], \quad (2)$$

where ε_i are symbolic representation of random values.

This model, called algebraic representation [14], covers a big range of nonlinear relationships which can be expressed as an algebraic relation. By eliminating x from \vec{E} in Equation (2), Equation (1) will be reduced to a Taylor Model. Furthermore, the AA representation can be achieved with a first order Taylor Model as in Equation (3).

$$x = x_0 + \sum_{i=1}^m x_i \cdot \varepsilon_i, \quad (3)$$

where the x_0 is the original value, x is the rounded value, $x_i \in R$ are constants and $-1 \leq \varepsilon_i \leq +1$ are noise symbols.

As represented by the AA analogy, noise symbols are random variables in the range $[-1, +1]$. Every noise symbol has a known Source (S) in the computation DFG and a known Probability Density Function (PDF). Accordingly, in this study, any noise symbol is defined by two other symbols $\varepsilon_i = (S, P)$, in which S represents the noise source and P indicates the PDF type. Extending symbol variables ε_i into two symbols provides more information about noise at every point of the system, however it necessitates more computational effort during the optimization process.

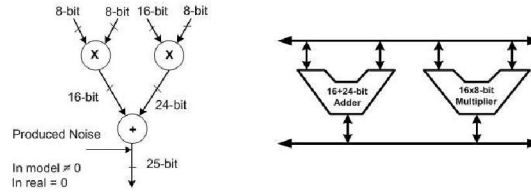


Fig. 1. Mapping a multiple-WL DFG to shared resource hardware.

In the proposed method, modeling the effects of WL manipulation takes place in two basic steps: the first is a noise symbol of the computational errors for every operation node in the DFG, and the second is propagation of the noise symbols through the DFG. The noise model, that is presented in [2], is the commonly accepted model in the multiple WL paradigm. Then this model is embodied in the form of affine symbol variables in [11]. Equation (4) gives the variance (σ_k) of the noise.

$$\sigma_k^2 = \frac{2^{2p}}{12} (2^{-2n_2} - 2^{-2n_1}), \quad (4)$$

where p represents the decimal point position, n_2 represents the required WL and n_1 represents the available WL for data representation ($n_1 > n_2$). According to this model, the values of noise sources are specified by the WL of the current FU and its preceding (parent) node(s). Despite its clarity, this model can mislead the optimization search in some cases, especially in stochastic search methods.

Figure (1) shows the maximum required WL in a sample DFG assuming 8-bit input data. The intermediate WLs are calculated based on the type of the operations and the input signals WLs. Since the maximum required WL propagates through the DFG, WL in each point in the DFG is a function of the parent nodes of that point. Accordingly, a noise source in the model in [2] is also dependent on all its preceding nodes. This example shows that in noise source evaluation by Equation (4), the WL for every node in the DFG must be calculated from data range propagation through all its preceding parent nodes. Especially in stochastic search methods or HLS integrated methods this data range analysis must be repeated at every iteration.

To provide a noise propagation model, it must be recalled that many DSP algorithms can be considered (or approximated [9]) as LTI systems. This assumption is very useful in simplification of the noise symbols propagation through DFG, however it is not the case in all applications. In our method, noise propagation through the DFG is evaluated by polynomial algebra. Accordingly, by a Taylor approximation of the nonlinear operations, it is possible to formulate the noise symbols in the output. This rules and methods are explained comprehensively in related works such as [11] and [14]. Another problem regarding noise symbol propagation is noise symbol combination. Unlike the AA method, here noise symbols are random variables with a known PDF, thus they can be merged to form new noise symbols. This is useful especially when the number of symbols increases explosively

in iterative algorithms. The expectation and variance of a sum of independent random variables such as ε_i in Equation (3) can be calculated by:

$$E(\sum_{i=1}^m x_i \cdot \varepsilon_i) = \sum_{i=1}^m x_i \cdot E(\varepsilon_i), \quad (5)$$

$$Var(\sum_{i=1}^m x_i \cdot \varepsilon_i) = \sum_{i=1}^m x_i^2 \cdot Var(\varepsilon_i) + 2 \sum_{i < j} x_i \cdot x_j \cdot Cov(\varepsilon_i, \varepsilon_j), \quad (6)$$

where $E(\cdot)$, Var and Cov stand for expectation, variance and covariance respectively. In addition, based on Central Limit Theorem the symbolic noises in Equation (3) be merged the distribution of the replacement symbolic noise is approximately normal for large m .

IV. OPTIMIZATION METHOD

From HLS viewpoint, costs of the design can be divided into three parts: those of datapaths; controllers; and interconnections. Since WL is the optimization parameter, its effect must be evaluated on each part individually. The controller and interconnections parts are not dependent on the WL and so they can be considered as constant values in the cost function. It is shown in [16] and [2] that accuracy, area and power consumption costs are dramatically dependent on the WL and execution delay is a function of the WL in the case of sequential FUs. Therefore, the cost model is as given in Equation (7),

$$F_{Total}(\vec{X}) = F_C + F_I + F_D(\vec{X}), \quad (7)$$

where F represents the cost function, \vec{X} is the set of synthesis parameters including WLs array for functional units (FUs) and C , I and D indices stand for Controller, Interconnection and Datapath respectively. All the relations and values are derived from basic cells in the ST 1.2 μm technology using the Synopsys tools as in [17] and [16].

The implemented design method starts from a high-level specification of the system and produces a set of synthesizable RTL-VHDL files. This tool is based on a target architecture with a multiple-shared bus. Actually this target structure restricts the implementation space but in return reduces the search time dramatically [16]. The utilized genetic operators (including weighted roulette wheel, crossovers and mutation [18]) are extracted from a standard GA procedure for variable length, integer array genomes. The synthesizer then employs an elite-preserving, VEGA optimization algorithm with a fitness function of a weighted Chebyshev combination of the basic design costs (area, delay, energy and noise) to find the optimal points in the constrained feasible space [18].

From this experience, genetic search does not converge in a reasonable time for complicated designs (more than 50 nodes in the DFG) because of the size of the feasible space ($> 50^{32}$). Thus a biased generation of the individuals is employed to speed up the GA optimization. Accordingly, before optimization search, a design with uniformly chosen WL is found to have the closest costs to the constraint values as the bias point, then the GA search for optimal points is

performed around this preliminary found solution. All our optimization results are achieved in reasonable execution times using this biased search method on an AMD-Opteron CPU.

V. RESULTS

Four case studies were implemented in ST 1.2 μm technology using the proposed method and tools. Design I is an order-18 difference equation, Design II is a Filter (FIR-25), Design III is an 8-point FFT and Design IV is a DCT 4x4.

Since, in practical implementations, there are pre-defined constraints which must be satisfied and therefore, other costs must be optimized with respect to them, an exhaustive set of synthesis optimizations are performed to show the design costs dependency to WL as a synthesis parameter along with the other classical synthesis parameters such as binding, allocation and scheduling.

Table (I) provides the results of design optimizations with fixed WL. This table gives the basic costs (design area, power consumption, delay and digital noise variance in the output node) for different assumptions of uniform WL ($W=8, 16, 24$ and 32) in all the design points. This set of information is used as the basis for comparison with other optimization results and also as constraints for them.

In the second step, constrained optimizations are applied for designs considering WL as a synthesis parameter. Since there are four different costs in this study, four different cases of constraints are considered. Table (II) shows the synthesis results for the same systems where design area is constrained. The constraint values for the area cost function in Table (II) are the area costs results in the Table (I).

Similarly Tables (III), (IV) and (V) show synthesis results with optimization constraints for energy consumption, output noise and latency respectively. Again the constraint values for each column and row of these tables can be found in the corresponding column and row in the Table (I). In these tables A, E, N and D stand for: area cost (in μm^2), energy consumption cost (in $\mu\text{Watt/Hz}$), digital noise variance in the output and latency cost of the design respectively (number of clock cycles).

VI. CONCLUSION

This study presents a new method for minimizing the hardware implementation of DSP algorithms by optimizing the word-length of the data in each functional unit. Symbolic noise analysis is used in combination with models of power consumption, circuit area and delay. Results from four example designs demonstrate a considerable saving in costs when these optimizations are applied.

REFERENCES

- [1] H. Keding, M. Willems, M. Coors, and H. Meyr, "FRIDGE: A fixed-point design and simulation environment," in *DATE'98*, 1998, pp. 429–435.
- [2] G. A. Constantinides, P. Y. K. Cheung, and W. Luk, *Synthesis and Optimization of DSP Algorithms*. Kluwer Academic Publishers, 2004.
- [3] K.-I. Kum and W. Sung, "Combined word-length optimization and high-level synthesis of digital signal processing systems," *IEEE Trans. on CAD*, vol. 20, no. 8, pp. 921–930, 2001.

TABLE I
DIFFERENT FIXED-UNIFORM WL FOR DESIGNS

Designs	Cost	Uniform Fixed WL for all the FUs			
		W=8	W=16	W=24	W=32
Design I	A	4152	8304	12456	16608
	E	5672.73	20779.2	45753.7	80602.9
	N	1.03E-2	4.04E-5	1.58E-7	6.16E-10
	D	168	311	454	598
Design II	A	22184	44368	66552	88736
	E	7143.11	26929.2	59584.1	105061
	N	1.28E-2	5.09E-5	1.99E-7	7.62E-10
	D	58	82	106	130
Design III	A	14456	44368	89736	119648
	E	9631.03	35813.4	78909	138029
	N	2.95E-2	1.15E-4	4.50E-7	1.76E-9
	D	100	110	121	145
Design IV	A	29912	111344	174744	222688
	E	18256.5	71076.6	156085	273138
	N	3.26E-2	1.27E-4	4.97E-7	1.94E-9
	D	121	130	152	178

TABLE II
AREA CONSTRAINED SYNTHESIS

Designs	Cost	Area costs are constrained as in Table (I)			
		#1	#2	#3	#4
Design I	E	4478.08	18350.9	42218.3	75706.9
	N	1.03E-2	4.04E-5	1.09E-7	6.16E-10
	D	150	293	436	580
	E	6295.61	25751.8	58066.5	101421
Design II	N	1.15E-2	4.80E-5	1.77E-7	6.82E-10
	D	53	79	102	126
	E	9095.21	34298.4	77773.5	136270
	N	2.01E-2	5.68E-5	3.14E-7	1.05E-9
Design III	D	100	107	113	137
	E	17341.5	70962.8	152568	271351
	N	2.62E-2	1.22E-4	4.71E-7	1.63E-9
	D	119	126	143	168

TABLE III
ENERGY CONSTRAINED SYNTHESIS

Designs	Cost	Energy costs are constrained as in Table (I)			
		#1	#2	#3	#4
Design I	A	3633	7785	11937	16483
	N	1.03E-2	4.04E-5	1.58E-7	4.27E-10
	D	150	293	436	580
	A	21987	44243	66105	87967
Design II	N	1.14E-2	4.29E-5	1.66E-7	6.92E-10
	D	55	78	102	127
	A	14456	44118	82724	118754
	N	2.73E-2	6.31E-5	2.46E-7	1.40E-9
Design III	D	100	106	119	135
	A	27086	100468	164315	222438
	N	2.79E-2	1.15E-4	4.69E-7	1.86E-9
	D	119	126	144	169

- [4] R. Cinar, L. Rijnders, P. Schaumont, S. Vernalde, and I. Bolsens, "A methodology and design environment for DSP ASIC fixed point refinement," in *DATE'99*, 1999, p. 56.
- [5] A. Nayak, M. Haldar, A. Choudhary, and P. Banerjee, "Precision and error analysis of MATLAB applications during automated hardware synthesis for FPGAs," in *DATE'01*, 2001, pp. 722 – 728.
- [6] G. Caffarena, G. A. Constantinides, P. Y. Cheung, C. Carreras, and O. Nieto-Taladriz, "Optimal combined word-length allocation and architectural synthesis of digital signal processing circuits," *IEEE Trans. on Circuits and Systems II: Express Briefs*, vol. 53, no. 5, pp. 339–343, 2006.

TABLE IV
NOISE CONSTRAINED SYNTHESIS

Designs	Cost	Noise costs are constrained as in Table (I)			
		#1	#2	#3	#4
Design I	A	3633	7785	11937	16089
	E	4478.08	18350.9	42091.6	75706.9
	D	150	293	436	580
	A	20646	43349	64495	87323
Design II	E	6074.11	24996	55273.2	100628
	D	53	79	101	126
Design III	A	12649	41595	88645	116178
	E	7572.32	31676.3	76467.4	128521
	D	99	107	114	135
Design IV	A	26889	100915	163027	219987
	E	16856.9	69308.4	147610	265048
	D	119	126	145	168

TABLE V
LATENCY CONSTRAINED SYNTHESIS

Designs	Cost	Delay costs are constrained as in Table (I)			
		#1	#2	#3	#4
Design I	A	3633	7785	11937	16089
	E	4478.08	18350.9	42091.6	75706.9
	N	1.03E-2	4.04E-5	1.58E-7	6.16E-10
	A	21665	44243	65783	88736
Design II	E	6936.68	26541.3	57530	104750
	N	1.20E-2	4.46E-5	1.87E-7	6.96E-10
Design III	A	14331	36909	79951	119201
	E	8879.05	30696	73861.1	136681
	N	2.81E-2	9.87E-5	3.29E-7	1.32E-9
Design IV	A	26245	106908	165353	221919
	E	16355.9	70709.7	151477	269987
	N	2.06E-2	1.10E-4	4.58E-7	1.84E-9

- [7] G. A. Constantinides, P. Y. K. Cheung, and W. Luk, "Optimum and heuristic synthesis of multiple word-length architectures," *IEEE Trans. on VLSI*, vol. 13, no. 1, pp. 39–57, 2005.
- [8] C. Shi and R. W. Brodersen, "Automated fixed-point data-type optimization tool for signal processing and communication systems," in *DAC'04*, 2004, pp. 478–483.
- [9] G. A. Constantinides, "Perturbation analysis for word-length optimization," in *FCCM'03*, 2003, pp. 81–90.
- [10] J. Stolfi and L. de Figueiredo, *Self-Validated Numerical Methods and Applications*. Institute for Pure and Applied Mathematics (IMPA), 1997.
- [11] D.-U. Lee, A. A. Gaffar, R. C. C. Cheung, O. Mencer, W. Luk, and G. A. Constantinides, "Accuracy-guaranteed bit-width optimization," *IEEE Trans. on CAD*, vol. 25, no. 10, pp. 1990–2000, 2006.
- [12] Y. Pu and Y. Ha, "An automated, efficient and static bit-width optimization methodology towards maximum bit-width-to-error tradeoff with affine arithmetic model," in *ASP-DAC'06*, 2006, pp. 886–891.
- [13] C. F. Fang, R. A. Rutenbar, and T. Chen, "Fast, accurate static analysis for fixed-point finite-precision effects in DSP designs," in *ICCAD'03*, 2003, pp. 275–282.
- [14] N. S. Nedialkov, V. Kreinovich, and S. A. Starks, "Interval arithmetic, affine arithmetic, taylor series methods: Why, what next," *Numerical Algorithms*, vol. 37, no. 1-4, pp. 325–336, 2004.
- [15] A. B. Sripad and D. L. Snyder, "A necessary and sufficient condition for quantization errors to be uniform and white," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 25, no. 5, pp. 442–448, 1977.
- [16] A. Ahmadi and M. Zwolinski, "Word-length oriented multiobjective optimization of area and power consumption in dsp algorithm implementation," pp. 614–617, May 2006.
- [17] —, "Area word-length trade off in dsp algorithm implementation and optimization," pp. 16/1–6, 2005.
- [18] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley and Sons, 2001.