2009 Eigth IEEE/ACIS International Conference on Computer and Information Science

Floating-point to Fixed-point Transformation using Extreme Value Theory

Linsheng Zhang, Yan Zhang, Wenbiao Zhou
Department of Electronic and Information Engineering
Harbin Institute of Technology Shenzhen Graduate School
Shenzhen 518055, China
Email: zhanglinsheng@gmail.com, ianzh@hit.edu.cn, zhouwenbiao@hit.edu.cn

Abstract

Efficient hardware implementation of digital signal processing and communication algorithms requires using fixed-point arithmetic. However, most of these algorithms are developed in floating-point arithmetic. The automation of floating-point to fixed-point transformation is a key step for high-level synthesis. Based on Extreme Value Theory, this paper presents a novel approach to find both the optimal integer and fractional bit-widths for signals. Compared to traditional full simulation-based methods, this approach uses lightweight simulations to study the characteristics of extreme conditions. With theoretical probabilities to prevent overflows and meet the output error constraint, the proposed approach provides more practical solutions than those by analytical methods. Case studies and experimental results demonstrate its efficiency.

1. Introduction

For fast algorithm verification and system prototyping, most of digital signal processing (DSP) and communication algorithms are developed in floating-point arithmetic. After that they are transformed to fixed-point arithmetic for more efficient hardware implementation, such as higher speed, smaller area and lower power. The manual transform from floating-point arithmetic to fixed-point arithmetic has always been a time-consuming and error-prone task for designers. As a key step for high-level synthesis, automatic floating-point to fixed-point transformation has aroused great attention from both academic and industry researchers over the past few years.

Floating-point to fixed-point transformation, or bit-width optimization, consists of range analysis and precision analysis. Range analysis is to find the minimum integer bit-widths for signals to prevent overflow. Precision analysis is to allocate optimal fractional bit-widths, which maintain the accuracy at output while minimizing implementation cost.

Generally speaking, bit-width optimization approaches can be separated into two categories. The first one is simulation-based methods [1]–[3], which feed large amount of input stimuli signals into the system and perform Monte Carlo simulations to find the extreme values. Although their solutions commonly work fine, these methods are time-consuming and the correctness for not simulated conditions is unknown. The second category is analytical methods [4]–[6], which employ range arithmetic (Affine Arithmetic) to analyze the ranges and precisions of signals. They are fast and the accuracy at output can be guaranteed. However, their results are often too conservative and they still have difficulty to handle systems with feedback loops, especially when the number of loops is unknown [6].

In very recent years, Extreme Value Theory (EVT) has been employed to analyze the ranges of integer variables in embedded systems [7], [8]. It does not require excessively long run-time like traditional full simulation-based methods. Besides, it provides theoretical probability to prevent overflows. To the best of our knowledge, there still has no literature discussing how to apply EVT to precision analysis. This paper proposes a novel approach to optimize both integer and fractional bit-widths using EVT. This approach uses lightweight simulations to study the characteristics of extreme conditions and it provides solutions with theoretical probabilities to prevent overflows and meet the output error constraint.

The rest of this paper is organized as follows. Section 2 introduces the necessary background on bit-width analysis and EVT. Section 3 presents our proposed automatic floating-point to fixed-point transformation. Section 4 gives case studies and comparisons with a state-of-the-art analytical method. Conclusions are summarized in Section 5.

2. Background

2.1. Bit-width analysis

A fixed-point signal's bit-width (BW), is composed of two parts. Integer bit-width (IB, including the sign bit for signed arithmetic) is used to cover dynamic range of the signal, in case of overflow. Fractional bit-width (FB) is used to prevent signal underflow and sustain the accuracy at output.

$$BW = IB + FB$$

If the dynamic range of signal x is found to be $[x_{min}, x_{max}]$, where the two real numbers x_{min} and x_{max} respectively denote its minimum and maximum values, the

978-0-7695-3641-5/09 $25.00 © 2009 IEEE
DOI 10.1109/ICIS.2009.38

required integer bit-width for signal x can be computed by the following [4]

$$IB_x = \lceil \log_2 (\max(|x_{min}|, |x_{max}|)) \rceil + \alpha,$$
$$\text{where } \alpha = \begin{cases} 2, & \mathrm{mod}(\log_2(x_{max}), 1) = 0 \\ 1, & \text{otherwise.} \end{cases}$$

where "$\lceil \, \rceil$" means the equal or nearest bigger integer.

When fractional bit-width is assigned to a signal, the signal is quantized. There are mainly two types of quantization: truncation and round to nearest. Truncation directly truncates the bits lower than least significant bit, while round to nearest uses an small adder for the rounding operation. They cause a maximum error of 2^{-FB} and 2^{-FB-1} to the signal respectively. The quantization error will take part in the computation and cause an inaccuracy at the output. For simplicity, round to nearest is considered as the default quantization type in the following discussion.

2.2. Extreme value theory

Similar to the central limit theorem, which describes that the probability density function (pdf) of the mean of large number of random variables approaches a gaussian distribution, Extreme Value Theory (EVT) reveals that the pdf of extreme values also approaches a limiting form [9]. EVT is a powerful statistical analysis theory for rare events. It has been used to variety of applications for quantitative risk management.

Extreme values of a signal are its maximum and minimum in a sample. A sample consists of M number of data points or values of the signal. If the extreme values of these M data points are observed, they are extracted from this sample. For N samples, N maxima and N minima of this signal are collected. According to the analysis of EVT, maxima and minima distributions converge to the Gumbel distribution. The Gumbel density function for maxima of signal x follows [9]

$$f(x) = \frac{1}{\sigma} \cdot \exp(-\frac{x-\mu}{\sigma}) \cdot \exp(-\exp(-\frac{x-\mu}{\sigma})), \quad (1)$$

and its cumulative distribution function (cdf) is

$$F(x) = P(X \leq x) = \exp(-\exp(-\frac{x-\mu}{\sigma})), \quad (2)$$

where μ and σ are the location and scale parameters. The location parameter shifts the distribution to left or right and the scale parameter scales its shape.

The method of moments estimators of μ and σ parameters are given by

$$\sigma = \frac{s\sqrt{6}}{\pi}, \quad \mu = \overline{x} - \sigma\lambda, \quad (3)$$

where λ is the Euler's constant (0.5772), s and $\overline{x}$ represent the standard deviation and sample mean respectively.

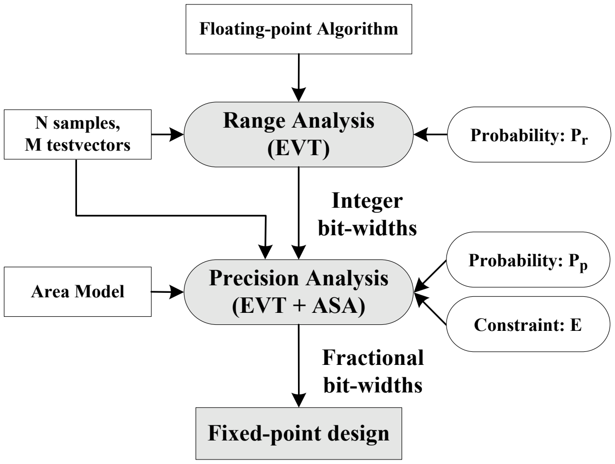


Figure 1. Overview of proposed approach

By simply changing the sign of evaluated variable, the minima can be changed into maxima. Then, the maxima converted from minima fit into the Gumbel distribution in (1).

3. Proposed approach

The overview of proposed automatic floating-point to fixed-point transformation is shown in Fig. 1. Its two phases, range analysis and precision analysis, are described in detail in the following subsections.

3.1. Range analysis

Firstly, N samples input stimuli, M testvectors for every sample, are generated and fed into the floating-point design to perform simulations. During the process, range analysis takes the responsibility to record the N maxima and N minima for every signal, whose integer bit-width needs to be determined.

Then, a default or user-specified probability P_r to prevent overflow is handed to range analysis. P_r can be different for different signals, depending on their importance. Then, by the Gumbel cdf for maxima in (2), the expected maximum value x_{max} for signal x should satisfy

$$P(x \leq x_{max}) = \exp(-\exp(-\frac{x_{max}-\mu}{\sigma})) = P_r.$$

Solving the desired x_{max} from above equation, we can get

$$x_{max} = \mu - \sigma \ln(\ln(\frac{1}{P_r})), \quad (4)$$

where μ and σ can be computed by (3) with x's statistics of N maximum values.

The expected minimum value x_{min} can be calculated by changing the signs of collected minima and computing the statistics of resulted maxima. By (4), the negative of x_{min} can be obtained. Thus, we can find the range of signal x to be $[x_{min}, x_{max}]$ with theoretical probability P_r to prevent its overflow. For two's complement representation, saturation arithmetic can be used to prevent the numerical wrap-around.

As long as N and M are large enough, the extreme values will fit the Gumbel distribution very well. In practical DSP applications, we find that N and M can be very small to stabilize the important parameters μ and σ. So, compared to fully simulation-based methods, it only requires lightweight simulations to find the ranges of signals.

Another important factor is probability configuration: P_r. The higher P_r is set, the lower the chance of overflow occurs. However, as (4) shows, P_r can never be set to 1. So, there will always be a possibility that rare event happens, which is the essence of stochastic methods. We can set P_r very high, such as 99.9999999% (9 nines), to make the possibility to overflow very low. Fortunately, we find that the increase of maximum value is really very small with the increase of P_r configuration in actual applications.

3.2. Precision analysis

If a set of fractional bit-widths are provided for the inputs and intermediate signals, whose FBs are to be allocated, their quantization errors will propagate in the resulted fixed-point design and take part in the computation. At the final output, there will be an error at the fixed-point result compared to floating-point one. We denote the absolute value of this error caused by fractional bit-widths allocation as e. After performing $M \times N$ times simulations, N maxima of e can be extracted. Provided with the guaranteed probability P_p and utilizing EVT, we can find the expected maximum absolute error at output e_{max} caused by this set of FBs.

An error constraint E for floating-point to fixed-point transformation is a precision requirement to be satisfied at the output. For example, $E = 0.01$ means that the maximum absolute error can be tolerated is 0.01 at the final output. So, $e_{max} \leq E$ should be fulfilled for the chosen FBs.

In order to minimize the implementation cost, an area model of the circuit as function of signals' bit-widths is needed to conduct the precision analysis. Different area models can be adopted targeting different logic synthesizers and device technologies. For example, we take the same area model as [4]: area model of $x \pm y$ is taken as $\max(IB_x + FB_x, IB_y + FB_y)$ and the area for $x \times y$ is modeled with $(IB_x + FB_x)(IB_y + FB_y)$. Because the required IBs for signals have been determined after range analysis, all we need is to find an optimal set of FBs to meet the error constraint E while minimizing the circuit area. Because of the nonlinear property of quantization errors, it is a constrained nonlinear optimization problem.

As Fig. 1 shows, we employ Adaptive Simulated Annealing (ASA) [10] to assist the precision analysis. ASA is an efficient global optimization algorithm. It is believed to be faster and more robust than other simulated annealing techniques for most complex problems [10]. For ASA, user supplies a constraint function and a cost function to find the minimum cost results and meet the constraint function. In our application, the constraint function is built by $e_{max} \leq E$ and the cost function is a function of the FBs. Thus, utilizing EVT and ASA, precision analysis optimizes the FBs to fulfill the output error constraint, while minimizing the total area cost.

3.3. Example of application

An example to demonstrate the usage of proposed approach is presented in this subsection. The floating-point algorithm needs to be transformed to fixed-point is

$$f(x) = (x_1 - 1)(x_1 + 2)(x_2 + 1)(x_2 - 2)x_3,$$

where inputs x_1, x_2 and x_3 are over [-2, 2]. Define the intermediate signals a, b, c and output d for the design as follows:

$$a = (x_1 - 1)(x_1 + 2), \qquad b = a(x_2 + 1),$$
$$c = b(x_2 - 2), \qquad d = cx_3.$$

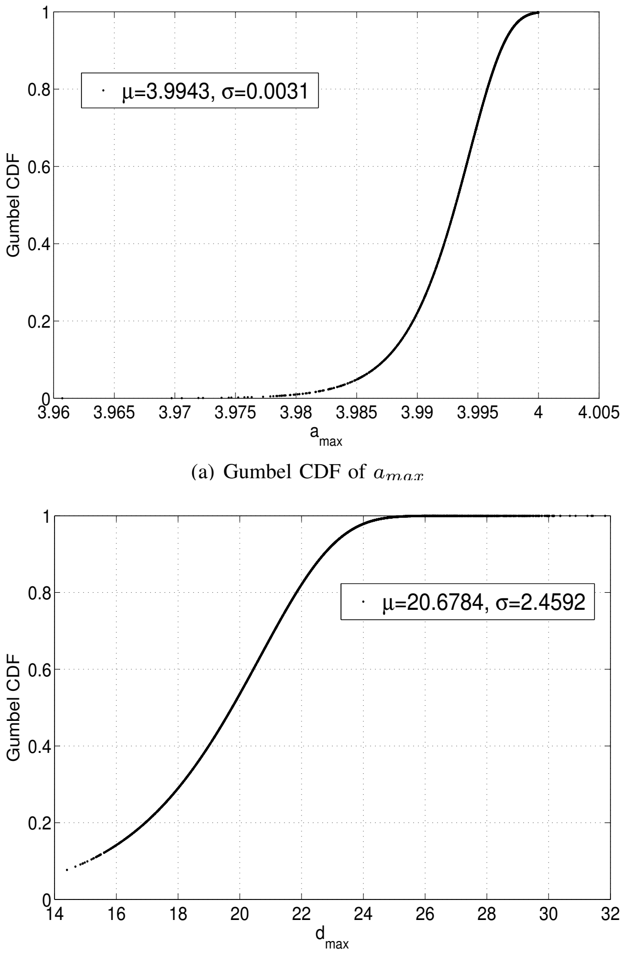
As reference, their accurate ranges are calculated by numerical method. Range results obtained by 10^8 random generated simulations and analytical method in [4], which uses Affine Arithmetic (AA) to analyze integer and fractional bit-widths, are listed in Table 1. The simulation-based method underestimates the IBs of a, c, d and AA-based analytical method overestimates the IBs of b, c, d.

Table 1. Accurate ranges and range results obtained by 10^8 simulations and AA

Sig.	Accurate		Simulation		AA	
	range	IB	range	IB	range	IB
a	[-2.25, 4]	4	[-2.25, 3.99]	3	[-8, 4]	4
b	[-6.75, 12]	5	[-6.71, 11.05]	5	[-24, 20]	6
c	[-9, 16]	6	[-8.75, 15.82]	5	[-80, 88]	8
d	[-32, 32]	7	[-25.76, 21.91]	6	[-176, 176]	9

Table 2. Range analysis results by proposed approach

Sig.	$P_r = 0.9999$	$P_r = 1 - 10^{-6}$	$P_r = 1 - 10^{-9}$
a	[-2.25, 4.02]	[-2.25, 4.04]	[-2.25, 4.06]
b	[-6.89, 13.33]	[-6.99, 14.32]	[-7.13, 15.85]
c	[-9.29, 19.31]	[-9.49, 21.86]	[-9.79, 25.57]
d	[-42.92, 43.33]	[-55.01, 54.63]	[-72.73, 71.68]



(a) Gumbel CDF of a_{max}

(b) Gumbel CDF of d_{max}

Figure 2. Gumbel CDF examples of maxima

With the same generated $N = 10^4, M = 10^4$ total 10^8 input testvectors (x_1, x_2, x_3) over [-2, 2], we perform the range analysis as described in Section 3.1. The Gumbel cdf for maximum values of signal a and d are depicted in Fig. 2. The range analysis results for different P_r are listed in Table 2.

In Tabel 1 and Table 2, it is shown that the proposed approach can predict the extreme values which cannot be found by simulation-based methods even in 10^8 simulations. When the computation chain is long and the number of input variables is large, fully simulation-based methods may never be able to reach the extreme regions. For example, the 10^8 simulations only get extreme values -25.76 and 21.91 for output d, which are far away from the true values -32 and 32. Compared to AA-based analytical method, which provides faithful ranges to cover these signals, our proposed approach find much narrower range intervals with very high guaranteed probabilities. In fact, for $P_r \geq 0.9999$, all the range results have covered their true ranges. Except one redundant bit of IB_d for the highest $P_r = 1 - 10^{-9}$, other IBs obtained by proposed approach are the same as the accurate results.

Table 3. Precision analysis results: FBs of signals and total area cost

Method	x_1	x_2	x_3	a	b	c	**Area**
$P_p = 0.9999$	14	14	13	13	12	12	1172
$P_p = 1 - 10^{-6}$	15	14	13	13	14	12	1242
$P_p = 1 - 10^{-9}$	15	14	14	14	15	13	1311
AA	15	15	17	13	13	12	1390

Then we come to the precision analysis phase. Suppose the output fractional bit-width is set to 8 and the precision at this bit needs to be guaranteed, which means the error constraint $E = 2^{-8}$. Perform the precision analysis as discussed in Section 3.2. The optimized fractional bit-widths and total area cost is compared with the AA-based analytical method in Table 3. It should be noted that, the overestimation of IBs by AA-based method also contributes to the increase of total area. In this application, 15.7%, 10.6% and 5.7% area are saved for $P_p = 0.9999, 1 - 10^{-6}$ and $1 - 10^{-9}$ respectively. That is because analytical methods always consider the worst-case for every uncertainty, which may never happen in reality.

In order to verify the correctness of results, 10^{10} new generated testvectors are simulated in floating-point and resulted fixed-point designs. A counter is used to calculate the probability of rare event when the absolute error between fixed-point output and floating-point output is larger than 2^{-8}. For these three resulted fixed-point designs with $P_p = 0.9999, 1 - 10^{-6}$ and $1 - 10^{-9}$, the computed possibilities of breaking the error constraint are 2×10^{-6}, 1×10^{-8} and 0 respectively. That demonstrates the reliability of proposed approach.

4. Case studies and comparisons

A C++ routine for the proposed automatic floating-point to fixed-point transformation is developed. For comparison, four cases taken from [4] are studied. Referring to an Affine Arithmetic C++ library [11], the AA-based analytical method described in [4] is reimplemented. All the case studies are performed on an Intel Pentium 4 3GHz PC with 512-MB DDR2 SDRAM.

Area results optimized by proposed approach for different E, P_r and P_p configurations and by AA-based analytical method are shown in Table 4. Their area cost ratios compared to the AA-based method for $E = 2^{-8}$ are illustrated in Fig. 3.

Because RGB to YCbCr and 8×8 DCT are linear systems, AA performs very well to find the accurate ranges for

Table 4. Case studies and comparisons between AA-based analytical method and our approach ($N = M = 500$; $P_1 : P_r = P_p = 0.9999$; $P_2 : P_r = P_p = 1 - 10^{-6}$; $P_3 : P_r = P_p = 1 - 10^{-9}$)

Case studies		**Total area cost**			
Application	**E**	P_1	P_2	P_3	**AA**
Degree 4 Polynomial	2^{-8}	281	290	312	385
	2^{-16}	934	962	1036	1257
RGB to YCbCr	2^{-8}	1962	2022	2084	2103
	2^{-16}	4285	4338	4460	4487
2×2 **Matrix Multiplication**	2^{-8}	791	834	899	978
	2^{-16}	2279	2353	2506	2747
8×8 **DCT**	2^{-8}	12542	13066	13859	14039
	2^{-16}	23112	24233	25968	26743

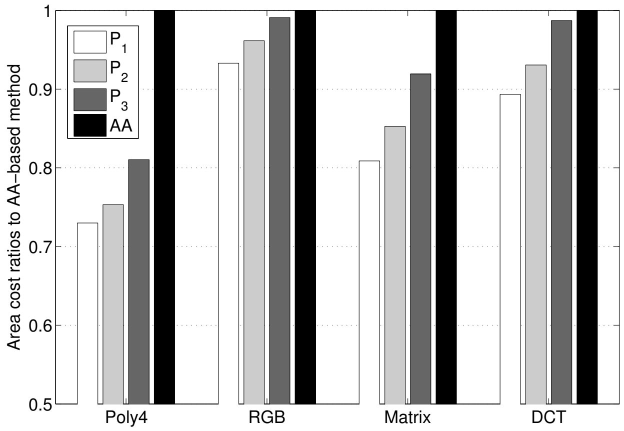


Figure 3. Area cost ratios compared to AA-based method in [4] for $E = 2^{-8}$

their signals. The area cost saved by proposed approach are primarily on fractional bit-widths. So, their area reductions for these two cases are small. However, for the two nonlinear algorithms, degree 4 polynomial approximation and 2 × 2 matrix multiplication, the area cost reductions are around 10% to 20%, even if P_r and P_p are all set to 9 nines. The area cost ratios for $E = 2^{-16}$ are similar to Fig. 3.

The CPU time consumed by proposed approach is mainly dependent on the total number $M \times N$ of testvectors for simulations. The CPU run-time for total optimizations are listed in Table 5. For $S_3 : M_3 = N_3 = 1000$, the optimization time consumed by proposed approach is 52 to 63 times of that by AA-based analytical method. If M and N are decreased to $M_1 = N_1 = 300$, the optimization time is only about 5 times comparing with AA-based method. However, the bit-widths and area cost results for $M_1 = N_1 = 300$, $M_2 = N_2 = 500$ and $M_3 = N_3 = 1000$ are very close to each other: bit-widths vary at most 1 bit and area cost varies between $\pm 3.5\%$. That benefits from the quick stabilizations of Gumbel distribution parameters μ and σ. So we can perform lightweight simulations to find the solutions.

Table 5. CPU time consumed for case studies, where $S_1 : M_1 = N_1 = 300; S_2 : M_2 = N_2 = 500; S_3 : M_3 = N_3 = 1000$

Case studies		**Optimization time (s)**			
Application	**E**	S_1	S_2	S_3	**AA**
Degree 4 Polynomial	2^{-8}	12	37	156	3
	2^{-16}	11	38	159	3
RGB to YCbCr	2^{-8}	103	314	1203	19
	2^{-16}	106	315	1207	21
2×2 **Matrix Multiplication**	2^{-8}	172	415	1879	34
	2^{-16}	173	416	1884	36
8×8 **DCT**	2^{-8}	1151	3519	13839	223
	2^{-16}	1159	3523	13855	224

Besides, because of its statistical essence, the proposed approach can be applied to any kind of systems. However, AA-based analytical methods still have problem at dealing with feedback loops contained systems, especially when the number of loops is unknown [6]. For these applications, the proposed approach can be used as an efficient substitution for traditional full simulation-based methods.

5. Conclusions

This paper presents a novel approach to automate the transformation from floating-point to fixed-point DSP systems. Based on Extreme Value Theory, the proposed approach uses lightweight simulations to study the characteristics of extreme conditions and provides more practical solutions than those by analytical methods, with theoretical probabilities to prevent overflows and meet the output error constraint.

Finally, we note some limitations of this approach that are the focus of future work. First of all, although the proposed approach does not require excessively long run-time like fully simulation-based methods, the optimization is still slow compared to analytical methods, especially when the design is large. Partition large design into parts maybe a choice, although it may lead to non-optimal solution. Secondly, there always has a possibility to overflow or break the error constraint. Combining analytical methods, which provides the probability 1 configurations for P_r and P_p, maybe a choice to resolve very important signals.

References

[1] W. Sung and K. Kum, “Simulation-based word-length optimization method for fixed-point digital signal processing systems”, *IEEE Trans. Signal Process.*, vol. 43, no. 12, pp. 3087–3090, Dec. 1995.

[2] K. I. Kum and W. Sung, “Combined word-length optimization and high-level synthesis of digital signal processing systems”, *IEEE Trans. Computers*, vol. 20, no. 8, pp. 921–930, Aug. 2001.

[3] S. Roy and P. Banerjee, “An algorithm for trading off quantization error with hardware resources for MATLAB-based FPGA design”, *IEEE Trans. Computers*, vol. 54, no. 7, pp. 886–896, Jul. 2005.

[4] D.-U. Lee, A. A. Gaffar, R. C. C. Cheung, O. Mencer and W. Luk and G. A. Constantinides, “Accuracy-guaranteed bit-width optimization”, *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 10, pp. 1990–2000, Oct. 2006.

[5] W. G. Obsborne, R. C. C. Cheung, J. G. F. Coutinho, W. Luk and O. Mencer, “Automatic accuracy-guaranteed bit-width optimization for fixed and floating-point systems”, *Proc. FPL 2007*, pp. 617–620, Aug. 2007.

[6] J. A. López, G. Caffarena, C. Carreras and O. Nieto-Taladriz, “Fast and accurate computation of the roundoff noise of linear time-invariant systems”, *IET Circuits Devices Syst.*, vol. 2, no. 4, pp. 393–408, Oct. 2008.

[7] E. Ozer, A. P. Nisbet and D. Gregg, “Stochastic bit-width approximation using extreme value theory for customizable processors”, *Proc. CC 2004*, pp. 250–264, Mar. 2004.

[8] E. Ozer, A. P. Nisbet and D. Gregg, “A stochastic bitwidth estimation technique for compact and low-power custom processors”, *ACM Trans. on Embedded Comput. Syst.*, vol. 7, no. 3, pp. 1–30, Apr. 2008.

[9] R. D. Reiss and M. Thomas, *Statistical analysis of extreme values*, Birkhauser, Basel, Switzerland, 1997.

[10] L. Ingber, “Adaptive Simulated Annealing (ASA)”, [Online]. Available: http://www.ingber.com/#ASA.

[11] O. Gay, “Libaffa - C++ affine arithmetic library for GNU/Linux”, [Online]. Available: http://www.nongnu.org/libaffa.