Bayesian Source Localization using Stochastic Computation

Adithya Krishna, and Chetan Singh Thakur
Indian Institute of Science, Bengaluru, India
Email: {adithyaik,csthakur}@iisc.ac.in

***Abstract*—Bayesian models are challenging to implement on hardware with the conventional design methodologies due to their high computational complexity. Conventional digital architectures are designed for deterministic computation and are not optimal for implementing probabilistic algorithms on hardware. In this work, we propose an alternative method to implement the probabilistic algorithms such as Bayesian models on hardware using a stochastic computation (SC) framework. This framework leverages on the probabilistic nature of the Bayesian models and facilitates the implementation of complex probabilistic models using simple logic gates. From an application standpoint, we propose a novel Bayesian source localization model (BSLM) that estimates a source's position in a noisy environment by solving the Bayesian recursive equation implemented on Field Programmable Gate Array (FPGA) with low resource utilization. The proposed SC design framework will pave the way to build complex probabilistic algorithms for real-time edge computing applications.**

***Index Terms*—Bayesian Inference, Field Programmable Gate Array, Unmanned Ground Vehicle, Hardware Architectures.**

I. Introduction

Bayesian systems are widely adopted in various state estimation applications such as object tracking [1]–[3], simultaneous localization and mapping (SLAM) [4], [5], inertial navigation [6], etc. These systems are computationally expensive, and real-time implementation using a conventional software based model is not feasible. Hence, there is a need to develop real-time computing hardware implementation, and many existing works have attempted the same [7]–[9]. Vigoda [7] proposed an analog VLSI circuits for probabilistic message passing. Shapero [8] developed configurable analog hardware for neuromorphic Bayesian inference. However, it isn't easy to scale and build large systems using analog circuits due to their non-programmable nature. Moreover, due to the lack of standard design and test flows, the development and testing of large analog implementations are challenging relative to digital systems. Additionally, the implementation of probabilistic Bayesian models on deterministic digital circuits has drawbacks such as, the area, and performance inefficiencies.

In this work, we propose a circuit based on the paradigm of Stochastic Computation (SC) originally suggested by Gaines [10], which is ideally suited for the implementation of probabilistic algorithms with the low area and low power requirements. In the SC framework, the digital values are represented by bit streams and can be processed using simple digital logic gates. For instance, multiplication of two-bit streams can be implemented using a simple AND gate in the SC framework. Our Bayesian system design using SC primitives is developed using simple logic gates and supports massive parallelism. Since the mathematical framework of all the Bayesian systems remains same, our design can serve as a potential building block to implement other complex Bayesian models.

From the application standpoint, we attempt to solve the source localization problem in real-time using the hardware architecture proposed. In this model, we try to estimate the position of a source based on noisy measurements by computing the Bayesian recursive equation online using SC.

II. Algorithmic Framework

In this work, an experimental framework similar to the one presented in [11] is used. The major contribution of this work is, we propose a novel Bayesian source localization model (BSLM) and demonstrate its implementation using SC.

A. Bayesian source localization model

Fig. 1 shows an overview of the Bayesian source localization in the 2D arena. The entire arena is divided into $K \times K$ grids, assuming that the source is present in one of the grids. As a proof of concept, we employ an omnidirectional light source as the source to be localized. The space around the UGV is divided into eight sectors with 45° angular separation and an array of eight photodiodes, each having 45° field of view is mounted atop the UGV to sense and localize the source. Photodiodes provide the bearing information of the source, which is used by the UGV to traverse and eventually converge at the source location. Depending on the light incident on each photodiode, we consider the photodiode's output to be either 0 or 1. Reflective sources and other stray light sources are potential sources of noise picked up by the sensor, producing false detections. The challenge is to estimate the source's location or, in other words, the co-ordinate of the grid with the highest likelihood of having the source using noise corrupted sensor measurements. For this, we employ a triangulation concept wherein we triangulate the region around the UGV into eight sectors, and depending on the output of photodiodes, we assign the probabilities to each grid in the arena using Bayes theorem. In our study, we assume that there is only one source present in the arena spanned by $K \times K$ grids, as shown in Fig. 1 and the direction of the UGV movement is restricted to a 45° angle in 2D arena.

Some probabilities relevant to the BSLM are discussed below.

- The probability of noise due to reflective or stray light sources is: $P(n) = \beta$.

- The probability of the j^{th} photodiode output being 1 i.e., $(z_j = 1)$ either due to a light source $(s_{(k,l)})$ in grid (k, l) or noise (n) is: $p(z_j|s_{(k,l)}, n) = \alpha$.
- If there is a light source in grid (k, l) of sector j, then the j^{th} photodiode output will be 1 with a probability of α irrespective of noise. The likelihood of photodiode output being 1 or 0 in the presence of source in grid (k, l) is:

$$L_{1k} = p(z_j|s_{(k,l)}) = \begin{cases} \alpha, & \text{for } z_j = 1. \\ 1 - \alpha, & \text{for } z_j = 0. \end{cases} \quad (1)$$

- If there is no source in the sector j, then there is a noise source (distractor) with a probability β. The likelihood of photodiode output being 1 or 0 in the absence of source is:

$$L_{0k} = p(z_j|\widetilde{s}_{(k,l)}) = \begin{cases} \alpha\beta, & \text{for } z_j = 1. \\ 1 - \alpha\beta, & \text{for } z_j = 0. \end{cases} \quad (2)$$

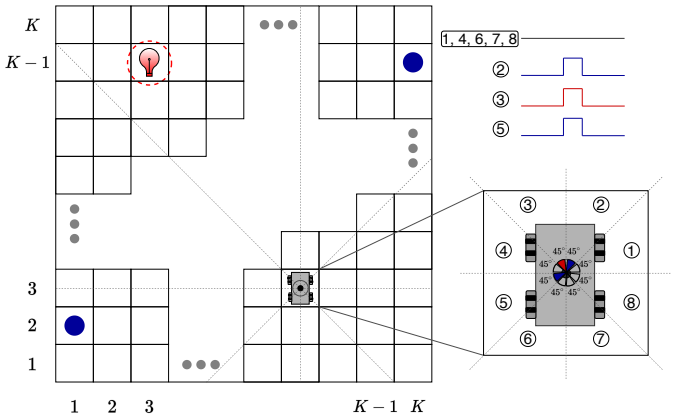

Fig. 1: An overview of the Bayesian source localization model. The entire arena is divided into $K \times K$ grids with the source positioned at $(3, K-1)$. The region around the UGV is divided into eight sectors with $45°$ angular separation, and a set of eight photodiodes is mounted on the UGV. The photodiode may fire (i.e., output 1) due to the presence of a distractor or light source. For instance, photodiodes in sector 2 and 5 fires due to the presence of distractors at positions $(K, K-1)$ and $(1, 2)$ respectively, whereas the photodiode in sector 3 fires due to the light source positioned at $(3, K-1)$. There is no activity in other sectors, and photodiode output remains 0.

B. Bayesian Inference

The BSLM aims to estimate the posterior probability of the source in grid (k, l) given the measurement z_j:

$$P(s_{(k,l)}|z_j) = \frac{P(z_j|s_{(k,l)}).P(s_{(k,l)})}{P(z_j)} = \frac{L_{1k}.P(s_{(k,l)})}{L_{1k}.P(s_{(k,l)}) + L_{0k}.P(\widetilde{s}_{(k,l)})} \quad (3)$$

$P(s_{(k,l)}|z_j)$ represents the required posterior probability, $P(s_{(k,l)})$ represents the prior probability that the source is present in grid (k, l), L_{0k} and L_{1k} represents the likelihood functions discussed in Section II-A. The posterior probabilities of all K^2 grids are evaluated for T time steps and the grid with maximum posterior probability gives an estimate of the source location. The pseudocode for the BSLM model is provided in Pseudocode 1.

Pseudocode 1 : BSLM pseudocode

Method: X_s = BSLM(X_{UGV}, z)

for $t = 1$ to T do
  for k, l in $((a, b)$ for a in range(K) for b in range$(K))$:
  Triangulation: ind = sector_index$(X_{UGV}, X_g(k, l))$
    ▷ $X_g(k, l)$ represents the position of the grid (k, l).
    $z_j = z$[ind]
  Posterior Update:
    $P_t(k, l) = \frac{L_{1k}.P_{t-1}(k, l)}{L_{1k}.P_{t-1}(k, l) + L_{0k}.(1 - P_{t-1}(k, l))}$
    ▷ Refer Eq. 1 & Eq. 2 for estimating L_{1k} & L_{0k}.
  end for
end for
Max computation and X_s estimation:
P_{max} = max$(P_t(:, :))$
X_s = pos_of(P_{max})

III. Hardware Implementation

The top-level architecture of the proposed Bayesian source localization model on hardware is shown in Fig. 2. Since there are $K \times K$ grids (possible source locations), we divide the network into $K \times K$ parallel inference paths, one corresponding to each grid. Each inference path consists of a triangulation module, a stochastic Bayesian module, and a register. A set of eight photodiode measurements (z) and the instantaneous position of the UGV (X_{UGV}) are fed to the triangulation module, which estimates the sector in which a particular grid is located. A photodiode measurement associated with the particular grid sector is selected and fed as the input to the stochastic Bayesian module. The register's output is the prior probability of the grid, which is fed back to the Bayesian module. The Bayesian module estimates the grid's posterior probability using Bayes' theorem (cf. Eq. 3). The posterior probability computed in each path is fed to a Max module to estimate the position of the source (X_s).

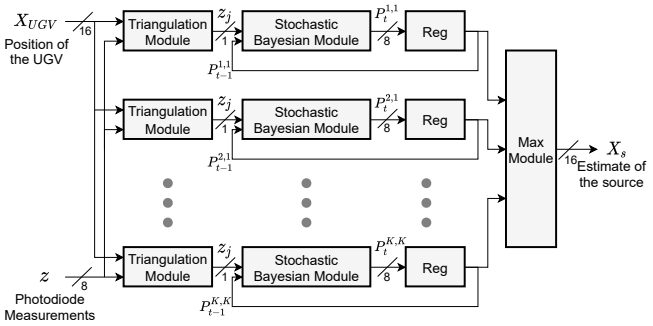

Fig. 2: Overview of BSLM hardware architecture.

A. Triangulation Module

A triangulation module is used to triangulate the region around the UGV into eight sectors and identify the sector in which a particular grid is located. This is achieved by knowing the instantaneous position of the UGV ($X_{UGV} = (x_{UGV}, y_{UGV})$) and position of the grid ($X_g = (x_g, y_g)$) in 2D arena. A set of eight photodiode measurements ($z = \{z_1, z_2, \cdots z_8\}$) are provided as the input. After identifying the sector index of a particular grid, we output the photodiode measurement corresponding to that sector. For instance, if a particular grid is located in sector 2 relative to the UGV, then z_2 is selected from z and given as the output.

B. Stochastic Bayesian Module

A stochastic Bayesian module calculates the posterior probability of each grid from the prior probability and the input measurement using the Bayesian equation given below:

$$P_t = P(s_{(k,l)}|z_j) = \frac{L_{1k}.P_{t-1}}{L_{1k}.P_{t-1} + L_{0k}.(1 - P_{t-1})} \quad (4)$$

where, P_t and P_{t-1} represents the posterior and prior probability of the grid respectively.

The Bayesian equation presented in Eq. 4 is implemented using the SC framework which gives the capability to perform complex division and normalization operations using simple logic gates. The circuit implementation of Eq. 4 using the SC is shown in Fig. 4. The basic computational elements that we have used to implement the Bayesian equation are:

1) Stochastic number generator (SNG): In the SC framework, we first convert the binary input into a stochastic domain wherein a binary number is represented by a bit stream, using a stochastic number generator (SNG) [3]. A bit stream is a sequence of binary digits (1s and 0s) in which information is contained in the primary statistics of the bit stream or the probability of any given bit in the stream being a logic 1. The conversion from binary to stochastic bit stream involves generating an n-bit random binary number (r_i) in each clock cycle employing a pseudo-random number generator and comparing it to the n-bit input binary number (X). For each time instant, if $r_i \leq X$, the comparator outputs 1, otherwise 0 is generated. The circuit of a stochastic number generator is shown in Fig. 3. Stream of random numbers is generated by a Linear Feedback Shift Register (LFSR) circuit [3], [12].

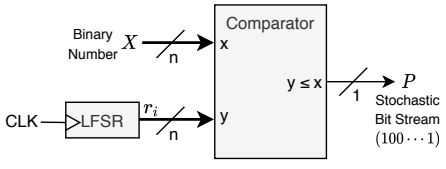

Fig. 3: Digital implementation of stochastic number generator.

2) Likelihood and Normalization Circuit: Calculation of the posterior probability using the Bayesian equation (cf. Eq. 4) mainly involves likelihood estimation and normalization operation. For this, we employ a likelihood module and a normalization module, as shown in Fig. 4. To begin with, the probabilities α, $\alpha\beta$, and prior P_{t-1} are converted into a stochastic domain by utilizing SNGs. Their equivalent stochastic versions are denoted by α_s, $\alpha\beta_s$ and P_{t-1_s}. A likelihood module estimate the likelihoods L_{1k} and L_{0k} as per Eq. 1 and Eq. 2 respectively in the stochastic domain. It is evident that the likelihood function's behavior is similar to a multiplexer, which selects one of the inputs at a time depending on the input measurement z_j which acts as a select line of the multiplexer. $1-\alpha_s$ and $1-\alpha\beta_s$ are computed using an inverter in the stochastic domain. Further, the multiplication of likelihoods L_{1k} and L_{0k} with the prior probability is implemented using AND gate in the stochastic domain. Thus, the output of two AND gates P_1 and P_2 in the stochastic domain are given by:

$$P_1 = L_{1k_s}.P_{t-1_s}\,;\; P_2 = L_{0k_s}.(1 - P_{t-1_s})$$

P_1 and P_2 are then fed to a normalization module to estimate the posterior.

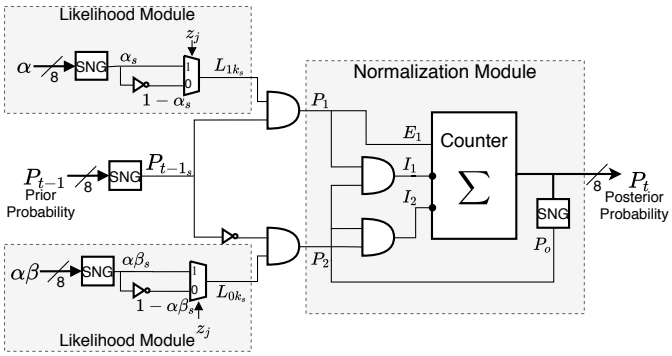

Fig. 4: Stochastic Bayesian module architecture.

The normalization module shown in Fig. 4 uses two AND gates to perform multiplication in the stochastic domain and a counter as an accumulator. The counter has one excitatory input, E_1, and two inhibitory inputs, I_1 and I_2. Two AND gates inside the normalization module performs the multiplication of P_o with P_1 and P_2, respectively. The counter increments by one unit when E_1 is logic 1, and both I_1 & I_2 are logic 0, decrements by one unit when E_1 is 0 and either of the inhibitory inputs is 1 or if E_1 is 1 and both the inhibitory inputs are 1, decrements by two units if E_1 is 0 and both the inhibitory inputs are at logic 1 and remains unchanged otherwise. Thus, the counter accumulates the value $P_1 - P_1P_o - P_2P_o$, and its output is given by,

$$P_o = \int P_1 - P_1P_o - P_2P_o\,;\; \dot{P}_o = P_1 - P_1P_o - P_2P_o$$

At equilibrium, the change in output probability is zero, i.e., $\dot{P}_o = 0$, thus we have:

$$P_1 = P_1P_o + P_2P_o$$

$$P_o = \frac{P_1}{P_1 + P_2} = \frac{L_{1k_s}.P_{t-1_s}}{L_{1k_s}.P_{t-1_s} + L_{0k_s}.(1 - P_{t-1_s})}$$

Thus, the architecture successfully implements the Bayesian equation in the stochastic domain. The counter's output provides the posterior probability (P_t) as an 8 bit binary value.

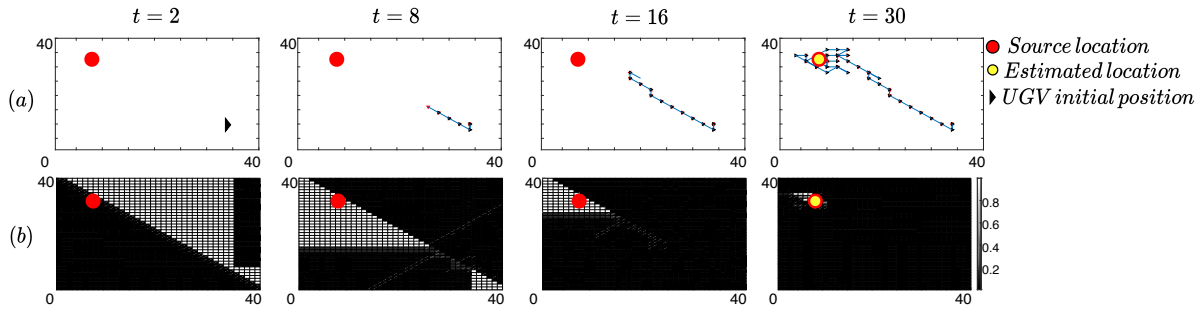

Fig. 5: 2D source localization experimental result with the source positioned at $[8, 33]$ marked by a 'red' circular dot. (a) Trajectory of the UGV at different time instants t. At the start, UGV is positioned at $[34, 10]$. Estimated position of the source (X_s) at time instant $t = 30$ obtained by BSLM is marked by a 'yellow' circular dot. (b) Corresponding posterior probability maps of 40×40 grids at different time instants.

C. Max Module

As the name suggests, the max module determines which of the $K \times K$ inference paths or grids have the maximum posterior probability and outputs the grid's co-ordinate, which most likely contains the source. Thus, providing an estimate of the source position (X_s). We employ digital comparators arranged in a tree structure [13], [14] to evaluate the maximum posterior probability with minimum computational time.

IV. Experimental Results

To study the performance of the proposed BSLM framework, we first modeled the entire system in MATLAB, and the experimental result is shown in Fig. 5 with $\alpha = 0.8$ and $\beta = 0.4$ in a 2D area spanned by 40×40 grids. We have simulated the system for 30 time steps, and the results demonstrate the effectiveness of the proposed framework by successfully localizing the source. We then implemented the model on Zynq UltraScale+ (xczu3eg-sbva484-1-e) FPGA using the SC framework and verified the FPGA results with the MATLAB version. The FPGA implementation was able to estimate the source location accurately and matched the results with the software version. Verilog HDL (Hardware Description Language) was used for the implementation on FPGA. Further, resource utilization of the implemented system on FPGA is provided in Table I. All the internal variables were converted from a floating-point to an 8 bit fixed-point representation. The counter used in normalization module (cf. Fig. 4) is 8 bit wide. Table. II shows the time steps (T) required to localize the source as a function of β for $\alpha = 0.8$, and compares the fixed point SC implementation with the MATLAB floating-point implementation. Here, the source is assumed to be localized if the Euclidean distance between the true and estimated source positions is below the specified threshold which is 1.5 in our case. Each value of T in the Table. II is obtained by taking the mean of 100 different simulations. It is evident that as β increases T also increases and the SC framework has relatively higher T due to the quantization and stochastic computation. However, the latency (L) of the single time step (t) for MATLAB implementation on Intel Core i7-7700 CPU with 8 cores clocking at 3.60 GHz is significantly greater than the SC implementation on FPGA using 100 MHz clock. Thus, the total computational time on FPGA using SC is very low compared to the MATLAB version. For instance, with $\beta = 0.6$, the total computational time for MATLAB and SC implementations are 6.6 s and 0.76 ms respectively.

TABLE I: Device utilization summary on Zynq UltraScale+ (xczu3eg-sbva484-1-e) FPGA.

Resource	Used	Available	Utilization(%)
LUTs	8,974	70,560	12.72
Registers	5,773	141,120	4.09

TABLE II: Mean time steps (T) required to localize the source as a function of β and comparison of latency (L) for MATLAB and SC implementations for single time step (t).

Implementation	Time steps (T)				Latency
	β=0.2	β=0.4	β=0.6	β=0.8	(L)
MATLAB	20	23	30	52	0.22 s
SC Framework	23	27	37	54	20.66 μs

V. Conclusions and Outlook

This paper presents a novel implementation of the Bayesian Source Localization Model on an FPGA using the SC framework. Simple logic gates are used in our design for recursive Bayesian computation. Our architecture supports massive parallelism for Bayesian implementation, which is otherwise not feasible with conventional architectures. The system was validated for solving the source localization problem in real-time. The experimental results indicate the efficacy of the model proposed, and the hardware implementation reaffirms the feasibility of Bayesian computation in real-time using SC. The model was implemented on Zynq UltraScale+ FPGA with 12.72% LUT utilization. The architecture is modular and thus, can be easily scaled for building large systems with minimum design and programming effort. Our real-time implementation of BSLM on hardware finds numerous applications in robotics for developing autonomous systems that can perform localization tasks without any human intervention. The system would also help implement other Bayesian probabilistic algorithms on hardware efficiently using the SC framework for edge computing applications.

REFERENCES

[1] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 425–437, 2002.

[2] K. Li, F. He, H. Yu, and X. Chen, "A parallel and robust object tracking approach synthesizing adaptive Bayesian learning and improved incremental subspace learning," *Frontiers of Computer Science*, vol. 13, no. 5, pp. 1116–1135, Oct 2019. [Online]. Available: https://doi.org/10.1007/s11704-018-6442-4

[3] C. S. Thakur, S. Afshar, R. M. Wang, T. J. Hamilton, J. Tapson, and A. van Schaik, "Bayesian Estimation and Inference Using Stochastic Electronics," *Frontiers in Neuroscience*, vol. 10, p. 104, 2016. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnins.2016.00104

[4] R. Sim, P. Elinas, and J. J. Little, "A Study of the Rao-Blackwellised Particle Filter for Efficient and Accurate Vision-Based SLAM," *International Journal of Computer Vision*, vol. 74, no. 3, pp. 303–318, Sep 2007. [Online]. Available: https://doi.org/10.1007/s11263-006-0021-0

[5] Z. Zhang, P. Dong, H. Tuo, G. Liu, and H. Jia, "Robust Variational Bayesian Adaptive Cubature Kalman Filtering Algorithm for Simultaneous Localization and Mapping with Heavy-Tailed Noise," *Journal of Shanghai Jiaotong University (Science)*, vol. 25, no. 1, pp. 76–87, Feb 2020. [Online]. Available: https://doi.org/10.1007/s12204-019-2146-7

[6] T. Nguyen, Y. Zhang, and M. Griss, "Probin: Probabilistic inertial navigation," in *The 7th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2010)*, 2010, pp. 650–657.

[7] B. Vigoda, "Analog Logic: Continuous-Time Analog Circuits for Statistical Signal Processing," *Doctoral dissertation, Massachusetts Institute of Technology*, Aug 2007.

[8] S. A. Shapero, "Configurable analog hardware for neuromorphic Bayesian inference and least-squares solutions," *Doctoral dissertation, Georgia Institute of Technology*, May 2013.

[9] A. T. Tharakan, D. Bhaskar, and C. S. Thakur, "Implementation of Bayesian Fly Tracking Model using Analog Neuromorphic Circuits," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020, pp. 1–5.

[10] B. R. Gaines, "Stochastic computing," in *Proceedings of the April 18-20, 1967, Spring Joint Computer Conference*, ser. AFIPS '67 (Spring). New York, NY, USA: Association for Computing Machinery, 1967, p. 149–156. [Online]. Available: https://doi.org/10.1145/1465482.1465505

[11] A. Krishna, A. van Schaik, and C. S. Thakur, "Source localization using particle filtering on FPGA for robotic navigation with imprecise binary measurement," *arXiv*, 2020.

[12] P. Kitsos, N. Sklavos, N. Zervas, and O. Koufopavlou, "A reconfigurable linear feedback shift register (LFSR) for the Bluetooth system," in *ICECS 2001. 8th IEEE International Conference on Electronics, Circuits and Systems (Cat. No.01EX483)*, vol. 2, 2001, pp. 991–994 vol.2.

[13] G. Xiao, W. Ahmad, S. A. A. Zaidi, M. R. Roch, and G. Causapruno, *High Speed VLSI Architecture for Finding the First W Maximum/Minimum Values*. Cham: Springer International Publishing, 2016, pp. 35–41. [Online]. Available: https://doi.org/10.1007/978-3-319-20227-3_5

[14] L. G. Amaru, M. Martina, and G. Masera, "High Speed Architectures for Finding the First two Maximum/Minimum Values," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 12, pp. 2342–2346, 2012.