ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 12, December 2014

Survey the stochastic computing and probabilistic transfer matrix (PTM) in logic circuits

Bahram Dehghan[1], **Naser Parhizgar**[2]

Department of Electrical Engineering, College of Engineering, Shiraz Branch, Islamic Azad University, Shiraz, Iran[1,2]

Abstract: Stochastic computing is a new alternative approach to conventional real arithmetic. Stochastic computation has been considered for different applications such as implementation of real-time motor controller and Error control coding and artificial neural networks. We have shown how to construct classic gates using probabilistic equations. In this paper, we focus on several logic circuits based on Probabilistic Transfer Matrix (PTM). In addition, this paper provides a survey of stochastic logic circuits with generating arbitrary probabilities. Also, we demonstrate PTMs is a methodology for representing probabilistic behaviour in logic circuits. In addition, our purpose has been to study the applicability of PTM in logic circuits. Results demonstrate that different kinds of logic circuits architecture can be verified with PTM methodology.

Keywords: StochasticComputing, Probabilistic Transfer Matrix, Logic Circuits

I. INTRODUCTION

Stochastic computing (SC) was proposed in the 1960s as a low-cost alternative to conventional binary computing. It is unique in that it represents and processes information in the form of digitized probabilities[1]. Its key feature is the use of long random bit-streams called stochastic numbers (SNs) to represent the data being processed[2]. The idea of probabilistic computing with digital logic dates back to von Neumann [3] and Gaines [4]. Gaines introduced stochastic modules to perform simple arithmetic operations such as addition, multiplication, division and integration.

The growing interest in stochastic computing is attributed to the simplicity and error tolerance of the stochastic modules. Consider two independent random bit streams A and B as inputs to a two-input AND gate where,

$$\mathrm{Prob}(A=1)=a, \mathrm{Prob}(B=1)=b \qquad (1)$$

The output random bit stream is then given by,

$$\mathrm{Prob}(C=1)=\mathrm{Prob}(A=1, B=1)= \mathrm{Prob}(A=1)\ \mathrm{Prob}(B=1) = ab \qquad (2)$$

Stochastic multiplication is thus performed by one AND gate as shown in Fig.1.[5]

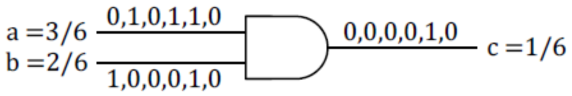


Figure1. Stochastic Multiplication, c=ab

Also, PTM is one of the probabilistic error modeling tools which performs simultaneous computation over all possible input combinations and calculates the exact probabilities of failures, it is based on matrix representation where column indices represent input values and row indices represent output values. For an example, PTM for a standard NOR gate is shown in (3) where p represents probability for incorrect output value[6].

$$\mathrm{PTM} = \begin{bmatrix} p & 1-p & 1-p & 1-p \\ 1-p & p & p & p \end{bmatrix} (3)$$

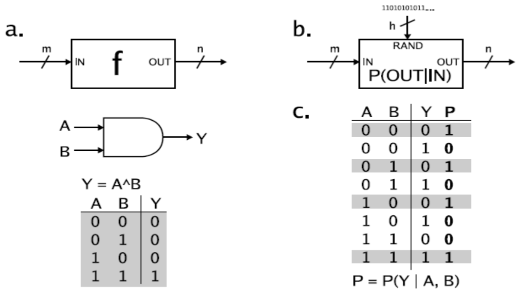
In this paper, we investigate this approach which directly models probability in digital circuits.

II. STOCHASTIC LOGIC CIRCUITS

Structured stochastic processes play a central role in the design of approximation algorithms for probabilistic inference and nonlinear optimization. Markov chain[7, 8] and sequential [9] Monte Carlo methods are classic examples. However, these widely used algorithms - and approximate Bayesian inference in general - can seem unacceptably inefficient when simulated on current general-purpose computers.

A Boolean gate has input bit lines and output bit lines, and puts out a Boolean function of its inputs on each work cycle. Each gate is representable by a set of truth tables, one for each output bit; the abstraction and an AND gate example are show in Figure 2a.

Figure 2b shows a combinational stochastic logic gate, which adds random bit lines. On each cycle, the gate puts a sample from P(OUT|IN) on its output lines, using the random bits[10].Consider the JK flip-flop shown in Figure 3 with input sequences of {ai} and {bi} representing the probabilities of P_a and P_b, respectively. The output bit c_i is equal to ‘1’ with the probability of Pc and is equal to ‘0’ with the probability of $1 - P_c$.



Y = A^B

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

A	B	Y	P
0	0	0	**1**
0	0	1	**0**
0	1	0	**1**
0	1	1	**0**
1	0	0	**1**
1	0	1	**0**
1	1	0	**0**
1	1	1	**1**

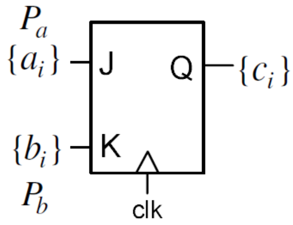
P = P(Y | A, B)

(a) (b)

Figure2. Combinational stochastic logic. (a) The combinational Boolean logic abstraction, and one example: the AND gate and its associated truth table. (b) The combinational stochastic logic abstraction[10].

A random output transition from ‘1’ to ‘0’ occurs with probability of $(1-P_c)P_a$ and the reverse transition occurs with the probability of P_cP_b. Since the expected occurrence of random transition in both directions must be equal, we have

$$P_cP_b=(1-P_c)P_a \rightarrow P_c = P_a/(P_a + P_b) \qquad (4)$$



$P_c = P_a/(P_a+P_b)$

Figure3. Stochastic division[11]

TABLE I. EVALUATION JK FLIP-FLOP

J	K	Q
1	0	1
0	1	0
0	0	Hold
1	1	Reverse

Table I shows the evaluation of the mentioned circuit. Table II shows the probabilistic equation for several basic logic gates.

TABLEII. PROBABILISTIC EQUATION

Gate	Probabilistic equation
NOT	$1-x_1$
AND	x_1x_2
NAND	$1-x_1x_2$
OR	$1-(1-x_1)(1-x_2)$
NOR	$(1-x_1)(1-x_2)$
XOR	$x_1(1-x_2)+x_2(1-x_1)$

III.PROBABILISTIC TRANSFER MATRIX

Every logic circuit has a PTM representing its error-free function. This is a matrix of size$2^m \times 2^l$where*m* and *l* are the numbers of inputs and outputs of the circuit, respectively[2].

PTM of an OR gate implementing z = x ∨y is

$$\begin{array}{c} \\ \\ xy \\ \\ \end{array}\begin{array}{c} \text{Z} \\ \begin{array}{cc} 0 & 1 \end{array} \\ \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \end{array} \qquad (5)$$

Multiplying an input PTM by a circuit PTM yields an output PTM, which for a single-output gate is a vector $[o_0\ o_1]$ where o_0 and o_1 are the probabilities of 0 and 1, respectively, appearing at the output. For example, the SC operation performed by the OR circuit of Fig. 4 is described by the PTM calculation[2].

Any event, which is associated with some of the random experiments and is happen if the occurrence of any one of the elementary event is its outcome. Hence, in this gate with mentioned inputs probability of occurrenceis 7/8.

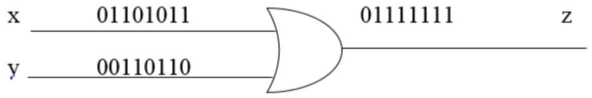


Figure4. OR gate

$$[1/8 \quad 2/8 \quad 3/8 \quad 2/8]\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} = [1/8 \quad 7/8] \qquad (6)$$

A relationship between input and output values for the operation of an integrated nano-based electronic circuit can be characterized by its truth table. Under certain circumstances, an incorrect output value can be generated by its input value. If we can identify how frequent this phenomenon is likely to occur, then we can demonstrate this occurrence using Probabilistic Transfer Matrix (PTM). PTM is represented in matrix form where column and row indicate input and output values respectively[15].

The number of different combinations of input and output vectors is limited by 2^m and 2^n, respectively. Probabilistic Transfer Matrix (PTM) is defined as a matrix P where the(j, k)th entry represents the probability of output vector value k given input vector value j, i.e., p(k | j). Therefore, P has 2^m rows and 2^n columns and the complexity of a function PTM with m inputs and n outputs is of order $O(2^{m+n})$. Fig. 5 gives an example of PTM for the NOT gate and other logic functions with m = 2 and n = 1. From this figure, we see that p is obviously the probability of an error occurrence and so q = 1 − p is the correct value probability[16].

$$T_{NOT}=\begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad P_{NOT}=\begin{bmatrix} p & q \\ q & p \end{bmatrix} \qquad T_{AND}=\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$P_{AND}=\begin{bmatrix} q & p \\ q & p \\ q & p \\ p & q \end{bmatrix} \qquad T_{XOR}=\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \qquad P_{XOR}=\begin{bmatrix} q & p \\ p & q \\ p & q \\ q & p \end{bmatrix}$$

Figure 5. Truth table matrix (T) and probabilistic transfer matrix (P) for NOT, AND, and XOR logic gates.

IV.PTM FOR OPERATIVE PART OF HALF-SUBTRACTOR

The purposes of PTM representation and label them in_0, . . . in_{n-1}; similarly, the m outputs are labeled out_0, ... out_{m-1}. The circuit C can be represented by a $2^n \times 2^m$ PTM M. The rows of M are indexed by an n-bit vector whose values range from 00….0 to 11….1. The row indices correspond to input vectors, i.e. 0/1 truth assignments of the circuit’s input signals[17]. PTM can be utilized to any logic gates. In PTM computation, an assumption has been made that gate errors happen independently and gate’s PTM is already recognized. In this stage we implement the

PTM representation and the sample to see the evaluation tools. For example, architecture of half subtractor (HS) circuit is shown in Fig.6.

The PTM for the complete half-subtractor design can be defined with matrix. We have two stage in recent example, S_0 and S_1 respectively.

Put be the addressing of columns in P_{S0} from 0 to 15. As can be seen through this addressing, only the columns 2, 7, 8, 13 (in bold) are non-nulls and they correspond to the rows extracted from the P_{S1} to form the total circuit probability transfer matrix P_{HS}.

Also, P_{G1} and P_{G2} are combined in parallel structure. We can achieve $P_{S1} = P_{XOR} \otimes P_{AND}$. PTM model that is applied to a parallel of XOR and AND gates must understand the differences at each stage. To determine the desired P_{S0} we deal with first stage. Hence, recent matrix equation implies that:

$$P_{S0} = \begin{array}{c} \begin{array}{cccccccccccccccc} & & (2) & & & & & (7) & (8) & & & & & (13) & & \end{array} \\ \begin{bmatrix} 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & \mathbf{0} & \mathbf{0} & 0 & 0 & 0 & 0 & \mathbf{0} & 0 & 0 \\ 0 & 0 & \mathbf{0} & 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{0} & 0 & 0 & 0 & 0 & \mathbf{0} & 0 & 0 \\ 0 & 0 & \mathbf{0} & 0 & 0 & 0 & 0 & \mathbf{0} & \mathbf{1} & 0 & 0 & 0 & 0 & \mathbf{0} & 0 & 0 \\ 0 & 0 & \mathbf{0} & 0 & 0 & 0 & 0 & \mathbf{0} & \mathbf{0} & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \end{bmatrix} \end{array}$$

The corresponding Matrix for operative part of half-subtractoris shown in the following:

$$P_{S1} = \begin{bmatrix} q^2 & qp & pq & p^2 \\ q^2 & qp & pq & p^2 \\ \boldsymbol{q^2} & \boldsymbol{qp} & \boldsymbol{pq} & \boldsymbol{p^2} \\ qp & q^2 & p^2 & pq \\ pq & p^2 & q^2 & qp \\ pq & p^2 & q^2 & qp \\ pq & p^2 & q^2 & qp \\ \boldsymbol{p^2} & \boldsymbol{pq} & \boldsymbol{qp} & \boldsymbol{q^2} \\ \boldsymbol{pq} & \boldsymbol{p^2} & \boldsymbol{q^2} & \boldsymbol{qp} \\ pq & p^2 & q^2 & qp \\ pq & p^2 & q^2 & qp \\ p^2 & pq & qp & q^2 \\ q^2 & qp & pq & p^2 \\ \boldsymbol{q^2} & \boldsymbol{qp} & \boldsymbol{pq} & \boldsymbol{p^2} \\ q^2 & qp & pq & p^2 \\ qp & q^2 & p^2 & pq \end{bmatrix}$$

PTM for operative part of half-subtractor

In the following matrix,we can see that P_{HS} is a submatrix consisting of the useful rows in the matrix P_{S1}.

$$P_{HS} = \begin{bmatrix} \boldsymbol{q^2} & \boldsymbol{qp} & \boldsymbol{pq} & \boldsymbol{p^2} \\ \boldsymbol{p^2} & \boldsymbol{pq} & \boldsymbol{qp} & \boldsymbol{q^2} \\ \boldsymbol{pq} & \boldsymbol{p^2} & \boldsymbol{q^2} & \boldsymbol{qp} \\ \boldsymbol{q^2} & \boldsymbol{qp} & \boldsymbol{pq} & \boldsymbol{p^2} \end{bmatrix}$$

PTM for the complete half-subtractor.

We cananalyse the Probabilistic Transfer Matrix (PTM) methodology which is based on matrix representation of errors in logic circuits.

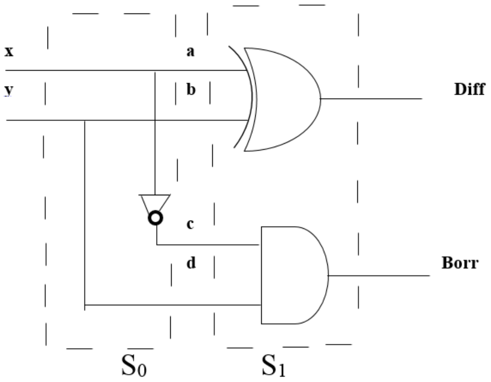


Figure 6. Half-subtractor circuit

V. PTM FOR ONE BIT COMPARATOR($A \leq B$)

In digital system, comparison of two numbers is an arithmetic operation that defines if one number is greater than, equal to, or less than the other number. The comparator compares two numbers, A and B, and defines their relative magnitudes. The result of comparison is determined by three binary variables that indicate whether A>B, A=B, or A<B. The matrixrepresentation of corresponding figure is shown.

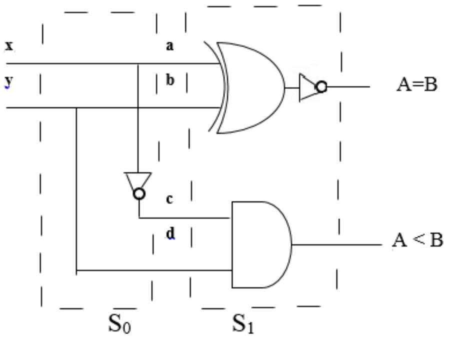


Figure 7. One bit comparator($A \leq B$)

$$P_{S0} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

The corresponding matrix for operative part of one bit comparator ($A \leq B$) is shown in the following:

$$P_{S1} = \begin{bmatrix} pq & p^2 & q^2 & pq \\ pq & p^2 & q^2 & pq \\ \boldsymbol{pq} & \boldsymbol{p^2} & \boldsymbol{q^2} & \boldsymbol{pq} \\ p^2 & pq & pq & q^2 \\ q^2 & pq & pq & p^2 \\ q^2 & pq & pq & p^2 \\ q^2 & pq & pq & p^2 \\ \boldsymbol{qp} & \boldsymbol{q^2} & \boldsymbol{p^2} & \boldsymbol{qp} \\ \boldsymbol{q^2} & \boldsymbol{pq} & \boldsymbol{pq} & \boldsymbol{p^2} \\ q^2 & pq & pq & p^2 \\ q^2 & pq & pq & p^2 \\ pq & q^2 & p^2 & pq \\ pq & p^2 & q^2 & pq \\ \boldsymbol{pq} & \boldsymbol{p^2} & \boldsymbol{q^2} & \boldsymbol{pq} \\ pq & p^2 & q^2 & pq \\ p^2 & pq & pq & q^2 \end{bmatrix}$$

PTM for operative part of one bit comparator(A ≤ B)

A matrix operation can accomplish this easily.

$$P_{CS}(A \leq B) = \begin{bmatrix} \boldsymbol{pq} & \boldsymbol{p}^2 & \boldsymbol{q}^2 & \boldsymbol{pq} \\ \boldsymbol{qp} & \boldsymbol{q}^2 & \boldsymbol{p}^2 & \boldsymbol{pq} \\ \boldsymbol{q}^2 & \boldsymbol{pq} & \boldsymbol{pq} & \boldsymbol{p}^2 \\ \boldsymbol{pq} & \boldsymbol{p}^2 & \boldsymbol{q}^2 & \boldsymbol{pq} \end{bmatrix}$$

PTM for the complete one bit comparator(A ≤ B)

VI. PTM FOR ONE BIT COMPARATOR(A ≥ B)

According to logic function achieved from truth table, logic diagram for one bit comparator (A ≥ B) is shown in Fig.8:

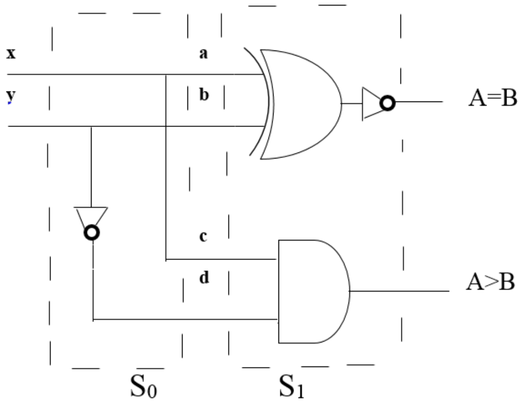


Figure 8. One bit comparator(A ≥ B)

$$P_{S0} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

The corresponding matrix for operative part of one bit comparator (A ≥ B) is shown in the following:

$$P_{S1} = \begin{bmatrix} pq & p^2 & q^2 & pq \\ \boldsymbol{pq} & \boldsymbol{p}^2 & \boldsymbol{q}^2 & \boldsymbol{pq} \\ pq & p^2 & q^2 & pq \\ p^2 & pq & pq & q^2 \\ \boldsymbol{q}^2 & \boldsymbol{pq} & \boldsymbol{pq} & \boldsymbol{p}^2 \\ q^2 & pq & pq & p^2 \\ q^2 & pq & pq & p^2 \\ qp & q^2 & p^2 & qp \\ q^2 & pq & pq & p^2 \\ q^2 & pq & pq & p^2 \\ q^2 & pq & pq & p^2 \\ \boldsymbol{pq} & \boldsymbol{q}^2 & \boldsymbol{p}^2 & \boldsymbol{pq} \\ pq & p^2 & q^2 & pq \\ pq & p^2 & q^2 & pq \\ \boldsymbol{pq} & \boldsymbol{p}^2 & \boldsymbol{q}^2 & \boldsymbol{pq} \\ p^2 & pq & pq & q^2 \end{bmatrix}$$

PTM for operative part of one bit comparator(A ≥ B)

Hence, the Pcs for (A ≥B) can be identified with recent matrix:

$$P_{CS}(A \geq B) = \begin{bmatrix} \boldsymbol{pq} & \boldsymbol{p}^2 & \boldsymbol{q}^2 & \boldsymbol{pq} \\ \boldsymbol{q}^2 & \boldsymbol{pq} & \boldsymbol{pq} & \boldsymbol{p}^2 \\ \boldsymbol{pq} & \boldsymbol{q}^2 & \boldsymbol{p}^2 & \boldsymbol{pq} \\ \boldsymbol{pq} & \boldsymbol{p}^2 & \boldsymbol{q}^2 & \boldsymbol{pq} \end{bmatrix}$$

PTM for the complete one bit comparator(A ≥ B)

Considering that by putting q = 1and p = 0 in the matrix, the output which is obtained based on the truth tablethat is true. For example, if the q and p parts of half-subtractor were equal to 1 and 0 respectively, then the following matrix can appear. The accuracy of the results is confirmed by the truth table.

$$P_{HS} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

PTM for the complete half-subtractor(q=1, p=0)

CONCLUSION

In stochastic computation, probabilities are represented as streams of random digital bits. The main objective of this work is to establish a comparison between three architectures in probabilistic transfer matrix(PTM). The PTM for NOT, AND, and XOR logic gates was depicted. Also, we illustrated the PTM concepts using a half-subtractor and one bit comparator with (A ≥B, A ≤ B) characteristic using matrices. This paper focuses on the effect of PTM computation on digital circuits. The other circuits with over than two states can be constructed easily.

REFERENCES

[1] A.Alaghi, John P. Hayes," Survey of Stochastic Computing", ACM Transactions on Embedded Computing Systems, Vol. 12, No. 2s, Article 92, 2013.

[2] A.Alaghi , John P. Hayes," Exploiting Correlation in Stochastic Circuit Design", IEEE 31st International Conference on Computer Design (ICCD), 2013.

[3] J.vonNeumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," Automata Studies, pp.43-98, Princeton University Press, 1956.

[4] B.R. Gaines, "Stochastic Computing Systems," Advances in Information Systems Science, J.F.Tou, ed., vol.2, chapter 2, pp.37-172, New York: Plenum, 1969.

[5] N. Saraf , K.Bazargan," Sequential Logic To Transform Probabilities", IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2013.

[6] N. S. S. Singh, N. H. Hamid, V. S. Asirvadam, U. Khalid and J.Anwer," Sensitivity Analysis of Probability Transfer Matrix (PTM) on same Functionality Circuit Architectures", 8th International Colloquium on Signal Processing and its Applications, IEEE 2012.

[7] N. Metropolis, AW Rosenbluth, MN Rosenbluth, AH Teller, and E. Teller. Equations of State Calculations by Fast Computing Machines. Journal of Chemical Physics, 1953.

[8] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. Readings in Computer Vision: Issues, Problems, Principles, and Paradigms, pages 564–584, 1987.

[9] A. Doucet, N. de Freitas, and N. Gordon. Sequential Monte Carlo in Practice, 2001.

[10] V. Mansinghka, E. Jonas, J.Tenenbaum," Stochastic Digital Circuits for Probabilistic Inference", massachusetts institute of technology, Cambridge, 2008.

[11] S. Sharifi Tehrani, S.Mannor and Warren J. Gross," Survey of Stochastic Computation on Factor Graphs," Proceedings of the 37th International Symposium on Multiple-Valued Logic, IEEE 2007.

[12] V. HamiyatiVaghef, A.Peiravi," A graph based approach for reliability analysis of nano-scale VLSI logic circuits", Microelectronics Reliability Journal, Volume 54, Issues 6–7, 2014, pp. 1299–1306.

[13] A. Khazamipour and K.Radecka," Adiabatic Implementation of Reversible Logic", 48th Midwest Symposium on Circuits and Systems, IEEE 2005.

[14] B.Dehghan, "Design Multipurpose Circuits with Minimum Garbage Outputs Using CMVMIN Gate," Chinese Journal of Engineering, Hindawi Publishing Corporation, Volume 2014.

[15] N. S. S. Singh, N. H. Hamid, V. S. Asirvadam," Error Threshold for Individual Faulty Gates Using Probabilistic Transfer Matrix (PTM)", 2014 AASRI Conference on Circuits and Signal Processing PP.138 – 145, 2014.

[16] Lirida A. B. Naviner, Ma´ı C. R. de Vasconcelos, Denis T. Franco and Jean-Franc, oisNaviner," Efficient Computation of Logic Circuits Reliability Based on Probabilistic Transfer Matrix", 3rd International Conference on Design & Technology of Integrated Systems in Nanoscale Era, IEEE 2008.

[17] http://www.springer.com/cda/content/document/cda_downloaddocumen t/9789048196432 -c2.pdf?SGWID=0-0-45-1349849-p174021170