Design To Implement Arithmetic Logical Bit Operations and Synthesizing Polynomial Functions Using Stochastic Computation

GATTU ASA PRASANTHI[1], CH.SRI GIRI[2]

[1]*PG student, Dept. of ECE,* GODAVARI INSTITUTE OF ENGINEERING AND TECHNOLOGY, Rajahmundry, East Godavari, *India.*

[2]*Asst.PROFESSOR, Dept. of ECE,* GODAVARI INSTITUTE OF ENGINEERING AND TECHNOLOGY, Rajahmundry, East Godavari, *India.*

(E-mail: asagattu7799@gmail.com)

Abstract — stochastic computing is arbitrary likelihood dissemination; it performs on ease, low power and low blunder resilience. A large portion of the computerized frameworks work on double portrayal of information i.e., parallel radix. In this double radix portrayal higher request bits weighted more than bring down request bits. In proposed approach we actualize legitimate piece number juggling tasks utilizing stochastic rationale for better execution and the exactness length of bit stream increments for better accuracy and increment in calculation time. Contrasted with traditional paired radix portrayal stochastic rationale is anything but difficult to execute and they possesses less territory ,control diminishment and equipment cost is low, Complex tasks can be performed in basic stochastic rationale.

Keywords— *stochastic rationale, traditional paired radix representation, synthesizing polynomial functions, Bernstein equations.*

I.INTRODUCTION

Presently a day, current registering is building with the prerequisites like little size, low power utilization and high unwavering quality. So in this paper we proposed a stochastic registering number-crunching activity. Stochastic registering works on low zone; low power and high dormancy. Stochastic registering worldview is a minimal effort strategy contrasting option to traditional twofold portrayal. The calculation performed on information speaks to in likelihood appropriation normally randomized piece streams. Stochastic registering can be connected in the fields like picture preparing (edge discovery, picture thresholding, mean separating… etc.,), neuron systems, blunder remedy techniques(Ldpc interpreting), Nano innovation and flag handling little variances of mistakes endured yet vast mistakes are cataclysmic. Delicate mistakes caused by ionizing radiation while expanding a semiconductor is a primary concern in circuits work in unforgiving situations. Arbitrariness is a principle worry in stochastic calculation. Arbitrariness accessible in communicating zones like correspondence and cryptography can perform with low many-sided quality. Great pseudo haphazardness reproduction gets in irregular material science, quantum physical science and science.

We propose another strategy for calculation is stochastic rationale. Stochastic rationale outlines change unmistakable sources of info and yields Boolean numbers and partial whole numbers are same. Orchestrating the circuits from 4 to 8 fragment need expanding fragment length from 2^4 =16 range to 2^8 = 256 bican be perform on rationale to create likelihood esteems input and output. The real approach is stochastic processing is material for randomized calculations. The proposed approach we give better mistake resilience, low region and high energy of configuration to implement8-bit math tasks utilizing stochastic rationale calculation. To actualize this design present another 8 bit precise stochastic circuits expansion, duplication, subtraction and division number-crunching activities performed. Stochastic registering requires increment in the accuracy requires an exponentional increment in bit stream length and comparing exponential increment in calculation time to change the numerical exactness of stochastic calculation ts.

II. RELATED WORK

Development of SC is started in 1956's by von Neumann; they defined fundamental concepts of probabilistic logic design. Modular redundancy technique is majorly used for fault tolerance. Memory based subsystems and communication purpose error correcting codes are used for both on chip and off chip. Probabilistic methods are appearing in system design and circuit level synthesizing. The main goal to applying probability is characterizing the unceratinity.For better performance of error tolerance through software mechanisms

statistical timing analysis used. Error tolerance is mainly considered in application field. The proposed approach is mainly for less power consumption in hardware design. It is also applied to computing applications like Monte Carlo simulations and they are tolerated errors. Presented a specific architectural design for data path computations, the main contribution is design a stochastic architecture is to detect the fault errors in previous approach. The new logic synthesis methodology the computation is performed in statistical probability distributions. Randomness process performed in serial or parallel bit streams in bit level. Accurate results depend on the statistics probabilities not their bit level. In the real valued based applications computation perform in analog character but they are implemented in digital components. This is for hardware based methods for error tolerance and memory based subsystems (error correcting codes).the proposed methodology is synthesizing the arbitrary polynomial functions implemented through Stochastic operation and also arbitrary continuous non polynomial expressions can be obtain through stochastic logic. Complex operations performed through probabilistic methods of stochastic operation.

III.PRELAMANARIES

a) *Stochastic logic*:

Stochastic operations performed at logic level in irregular bit streams in two ways either continued or side by side. Continued bit streams signals are performed in probabilistic in time and side by side bit streams they are performed in space. The bit streams carrying binary values zeros and ones and they are proceed to binary logic gates AND, OR. The signal is obtain through statistical distribution through logical values. Computation in Boolean domain represent probabilistic in real domain. Randomized bit streams the stream 'N' bits containing N1 ones and N-N1 zeros and stochastic number denoted as N1/N treated as probability. The main advantage of a stochastic computation is highly stood for fault errors**.**

A) EXISTING SYSTEM**:**

Traditional paired radix portrayal performs paired arithmetic operations addition, subtraction, multiplication they contain to specific rules. Division they didn't perform on any rule and little bit tougher than other operations.

Traditional paired radix addition performs on the logic 0+0=0, 0+1=1, 1+0=1,1+1=1 the carry will be applied to next digit.

For example traditional paired radix portrayal addition of inputs (0.001)2, (0.101)2 the output will be (0.101)2.The same inputs taken (0.001),(0.110)2 and when one single bit flip change the result will be(0.111)2the output will be high. Other binary arithmetic operations perform with their logic and occupies more area and hardware cost, single bit flip will be given high error.

IV.PROPOSED SYSTEM

Stochastic logical calculation process the logical synthesizing arithmetic operations are obtain in probabilistic domain. The arithmetic operations are simple logical multiplication, addition. Stochastic computing there are two representations available, they are unipolar describing a real-valued number x $(0 \leq x \leq 1)$ is defining by stream in which each bit has probability x of being one and probability $1- x$ of being zero. In the "bipolar "devices, a real-valued number y $(-1 \leq y \leq 1)$ is defining by a stream in which each bit has probability $(y + 1)/2$.The main advantage of using Stochastic bit streams is the lack of a place value as compared to the traditional paired MSB and LSB.The environment gets corrupted by bit flips and suppose that the significant bit gets flipped with double radix encrypting. Here, the relative error will be 2^m-1/2^m =1/2. However, in stochastic encoding, the data is represented as the fractional weight on a bit stream of length 2^m. Thus the single bit flip changes the result only by ½^m.

Traditional paired radix describing is a positional encoding and maximally compressed. Stochastic is a uniform encoding and uncompressed.

a) *Stochastic number generation*:

To obtain stochastic process first generate the stochastic number generator with the help of linear feedback shift register and constant binary number given to the AND switches the resulted outputs are given to OR switch to generate SNG (stochastic number generator) probabilities.

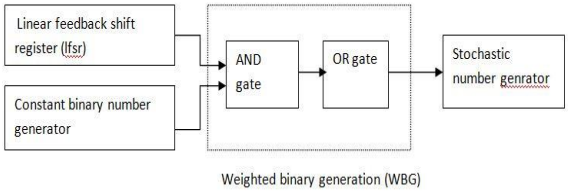


Fig 1:block diagram for stochastic number generator

A. *Multiplication process in stochastic rationale*

Logical arithmetic operation of a stochastic multiplication requires only one AND switch. Compared to traditional paired radix stochastic calculation is easy and less hardware needed. Consider two input AND switch input g,h and resulted output S.

The stochastic product of

S=P (S) =P (g=1 and h=1) then

S=P (g=1). (h=1)

S=g.h

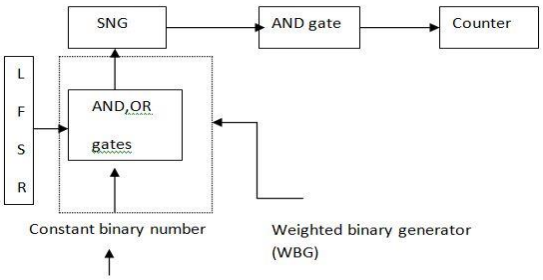


Fig 2: block diagram of stochastic multiplication

For example consider stochastic product of g=11011011, h=01010110 the probabilistic representation of p (g)=6/8,p(h)=4/8 then the resulted output p(S)=p(g=1).p(h=1)

P(S) =6/8.4/8=3/8

P(S) =01010010

Stochastic probabilistic defining = number of one's / total no. of bit stream length.

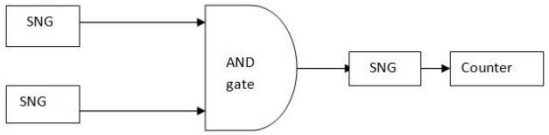


Fig 3: logical diagram of stochastic multiplication

Compared to traditional representation stochastic productive is easy because each arithmetic operation in traditional must contain the decrypting operands, the weighted number of MSB's is more and LSB's are less so the result is must be re-encrypt in weighted form. In stochastic form is unique and no decrypting and encrypting are need to operate on the values all bits are equally weighted.

Stochastic rationale of one input bit stream bit flip is g=11011011=6/8, h=01010111=5/8 then the result output is S=01010011=4/8. When bit flip changes the output is 3/8≈4/8 or near the output value.

B. *Scaled addition process in stochastic rationale*:

Common addition is not practical to add two probability values precisely. The resulted value is greater than one; they cannot be performing in probability values. Stochastic scaled adder operating on real valued numbers in the stochastic representation. It contains a multiplexer (MUX), it will be with selects one of its two input independent values to be the output value, based on a third "selecting" input value

Assuming two independent and correlated input bit streams g, h and selecting input bit stream S the output bit stream k is

K=p(K=1) =p(S=1 and g=1) + p(S=0 and h=1)

=p(S=1) .p (g=1) + p(S=0). P (h=1)

=S.g + (1-S).h

Stochastic addition performed independent inputs.x1=01001101, x2=10110111, s=00101000 output x3=10010110.

P(k) =p(S=1) p(g=1) +p(S=0) p(h=1)

=p (2/8) p (4/8) +p (2/8) p (6/8)

P (k) =p (6/8)

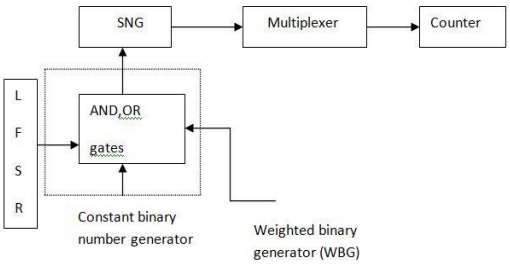


Fig 4: block diagram of stochastic addition.

Same process when x1=01001101, x2=00110111, s=00101000 output x3=10010110.

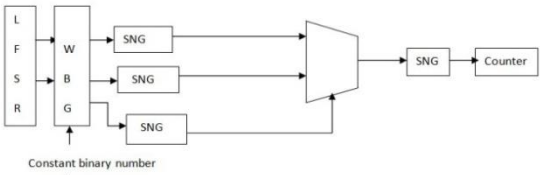


Fig 5: logic diagram of stochastic scaled addition

P (k) =p(S=1) p (g=1) +p(S=0) p(h=1)

=p (2/8) p (4/8) +p (2/8) p (5/8)

P (k) =p (7/8) a single bit flip change the output will be 7/8≈6/8.

C. *Stochastic subtraction*:

Scaled subtraction in stochastic logic implementation using MUX and not gate, subtraction can be obtained in bipolar format because negative value representation can be done in bipolar format. Unipolar format negative representation bit streams cannot be done. The scaled subtraction inputs are bipolar format and selective line input will be unipolar format. Scaled subtraction is similar to scaled addition only one input stream NOT gate will be connect to MUX based on this logic subtraction performs as

P (K) =p (g =1) p (S=1) – p (h=0). P(S=0)

=g.S-(1-S).h

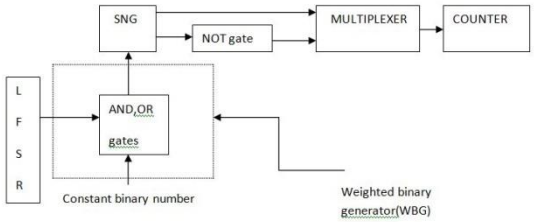


Fig 6: block diagram of stochastic subtraction

Assuming stochastic subtraction of independent input bit streams

p(g)=01011111,p(h)=10110001,p(s)=10010000output p(k)= 01010000

P (K) =P(S=1) P (g=1)-p (S=0) p (h=1)

=p (2/8) p (6/8)-p (2/8) p (4/8)

=p (2/8)

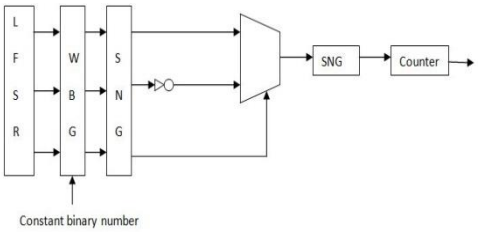


Fig 7: logic diagram of stochastic scaled subtraction

Same process when single bit flip give p(s) =10001000
P (g) =01011111, p (h) = 00110001, p (s) =10010000output
P (k) = 01010000
P (K) =P(S=1) P (g=1)-p(S=0) p (h=1)
=p (2/8) p (6/8)-p (2/8) p (3/8)
=p (3/8)
A single bit flip error result will be near the exact result 3/8≈2/8

D. *Stochastic divider*:

In this paper proposes a new division technique is CORDIV (correlated division) correlated bit streams are inputs. Area and cost is low in CORDIV divider other than conventional dividers. CORDIV operate on fallowing two keys
1. The division probability px1/x2 defines the x1, x2 division probability px1, x2/px1
2. Correlated inputs x1, x2 efficiently transform this division operation px1/px2.

This process is achieving better accuracy.x1, x2 are maximally correlated bit streams if px1<px2 thenpx1x2=px1.Itroduces to Px1/x2=px1/px2.This process is achieving better accuracy.x1, x2 are maximally correlated bit streams if px1<px2 then px1.x2=px1 It reduces toPx1/x2=px1/px2.

Correlated division process only one random number generator is used so the area and cost is low. If correlated division condition Z=x1/x2.the probability representation is Pz=px1/px2.the selective line l=1 MUX is controlled by x2, when x2=1.same way x2=0 the output result stored in D-flip-flop previous bit. This probability is PDFF=px1/px2.the outputMUX

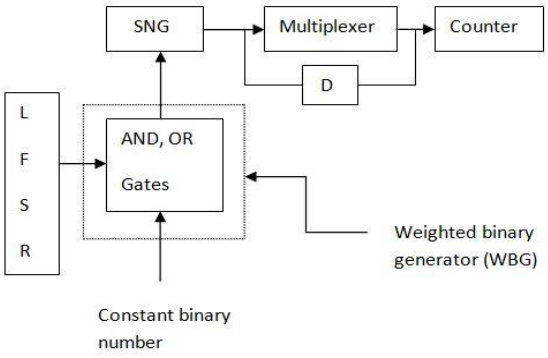


Fig 8: block diagram of stochastic divider

Pz=px2.px1/px2+ (1px2).pDFF=px1/px2. When l >1padding memory can be realized in several ways. It can simply be an l bit lfsr that stores l consecutive zeros in z. D-flip flop is feedback in MUX and x2 is 0 MUX select the line from D-flip flop. z padded memory is continuously stored consecutive 0s in shift register. To avoid such repetition, the divider must have large l bit padding memory when x2 have many consecutive zeros. Large Padding memory adds significantly l increase. In some cases for short stochastic numbers l is not usually preferable .l clock cycles needed to warm up to fill the effective bits in padding memory.

Assuming correlated inputs x1=1001 0100 0000 0000=3/16, x2=1001 1100 0001 0000=5/16, the output pz=px1/x2=3/5=10/16=1111 0111 1110 0000.

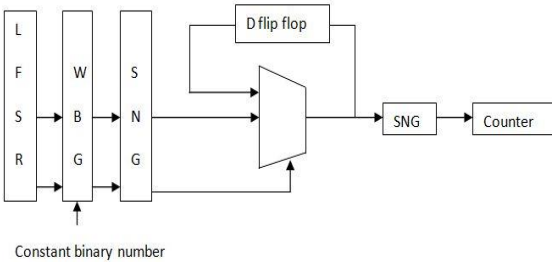


Fig 9: logic diagram of stochastic divider

The advantage of division circuit is conditional probability of x1, x2 and quotient x1x2 and the probability of x2.

b) Synthesizing polynomial functions:

Stochastic computation performs in synthesizing polynomials by using adder, Mux operations. The proposed method is for implementing arbitrary polynomial functions. Synthesizing polynomial coefficients must be less than zero or greater than one. The polynomial function maps from the unit interval to values in the unit interval. Large and low coefficients are implemented by stochastic logic. The first step is implement synthesizing polynomial is transforming a power-form polynomial into Bernstein polynomial. Its degree n is of the form

$$B_n(t) = \sum_{t=0}^{n} b_{i,n}\ B_{i,n(t)} \quad (1)$$

Where $b_{i,n}$is Bernstein polynomial co-efficient
Adapting power form polynomial of degree n

$$g(t) = \varepsilon_{i=0}^{n} t^i a_{i,n} \quad (2)$$

In Bernstein polynomial degree n

$$g(t) = \varepsilon_{i=o}^{n}\ b_{i,n} B_{i,.n}(t) \quad (3)$$

The conversion from $a_{i,\,n}$ polynomial into $b_{i,n}$ polynomial the closed form is

$$b_{i,n} = \varepsilon_{i=o}^{n} \frac{\binom{i}{j}}{\binom{n}{j}} a_{j,r} \quad (4)$$

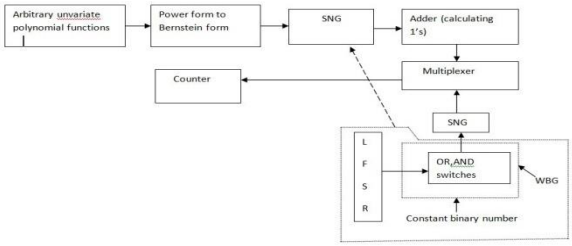


Fig 10: block diagram of synthesizing functions.

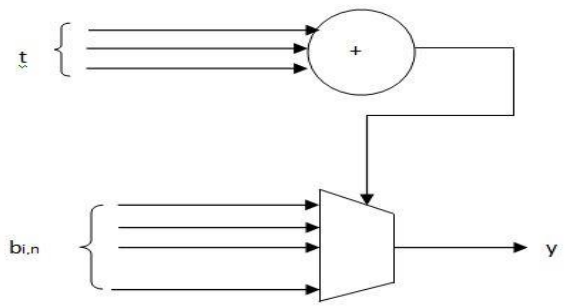


Fig11: logical diagram of synthesizing polynomial functions.

The stochastic consist of adder block and multiplexer. And inputs of the adder sets as t, t1 ...tn, the data inputs of MUX is z0, z1 ...zn.The output of the adder is selective line input to the MUX. The input bit streams are independent and P (Xi=1) = xi ε (0, 1) for $1 \leq i \leq n$ and P (Zi=1) =Zi ε (0, 1) for $0 \leq i \leq n$.

For example: g(t)=3/4-t+3/4t^2 maps the unit interval itself. It can be converted polynomial of degree 2.

Using eq(1),(2),(3),(4),(5)

$$g(t) = {}^{3}/_{4} B_{0,2}(t) + {}^{1}/_{4} B_{1,2}(t) + {}^{1}/_{2} B_{2,2}(t)$$

$$b_{i,m+1} = \begin{cases} b_{0,m} & i=0 \\ (1-i/m+1)b_{i,m}+i/m+1\ b_{i-1,m} & 1 \leq i \leq m \\ b_{m,m} & i=m+1 \end{cases} \quad (5)$$

degree of elevation.

Commonly a power-form polynomial of degree n can be adapted into an equivalent Bernstein polynomial of degree greater than or equal to n. The coefficients of a Bernstein polynomial of degree m + 1 ($m \geq n$) can be derived from the Bernstein coefficients of an equivalent Bernstein polynomial of degree m.

Synthesizing polynomial functions in traditional portrayal have large calculation and hardware area required. For example z=x+xy-(1+y) the function required multiplication, subtraction and addition operands. The operations perform large area.

VI.RESULTS

Stochastic implementation of area $A_{(n)}, D_{(n)}$delay product circuits compute the Bernstein polynomial degree n=2, 3, 4, and 5.then

$$A_{(n)} D_{(n)} 2^{M}$$

Where 2^M accounts for length of the bits stream.

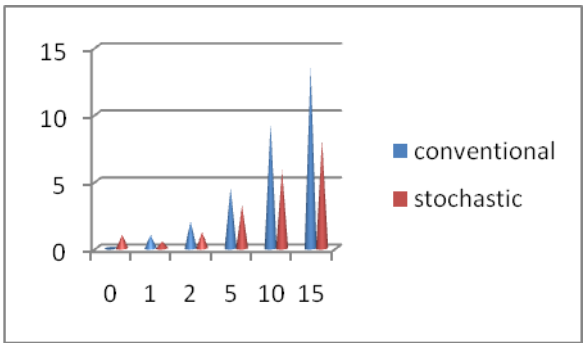
Table 1: area and delay product of circuit compute Bernstein polynomials of degree 2,3,4,5 and 6.

degree n of Bernstein polynomial	Area A(n)	Delay D(n)
2	13	7
3	20	12
4	38	16
5	46	19
6	55	19

Table 2: shows the area delay product for conventional and stochastic implementation for polynomials of degree n= 3, 4 and 5 resolutions 2^-M. M=7, 8, 9, 10, the last column shows the ratio of the two. We can see that for $M \leq 8$.the area and delay product of stochastic is always less than that of conventional implementation

n	M	Area-delay product		Stochastic product/conventional product
		conventional	stochastic	
3	7	97407	26120	0.268
	8	144752	52240	0.360
	9	211526	104480	0.493
	10	291099	208960	0.717
4	7	105012	83050	0.739
	8	193430	170040	0.879
	9	269860	340080	1.260
	10	395436	680160	1.720
5	7	158765	123404	0.777
	8	238975	246808	1.032
	9	355050	493616	1.238
	10	505982	987232	1.951
6	7	187632	138450	0.737
	8	294520	276900	0.940
	9	434280	553800	1.275
	10	612964	1107600	1.806

Fig 12: average output error rate of stochastic and conventional implementations different error ratios



Simulation results:

Fig 13: simulation result for stochastic multiplication

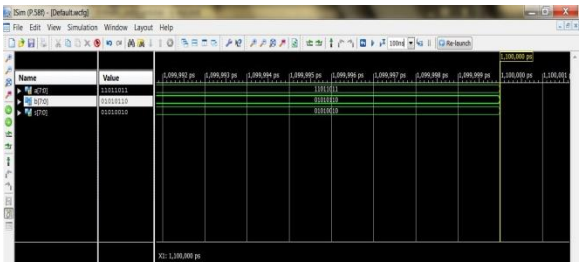


Fig 14: simulation result for stochastic addition

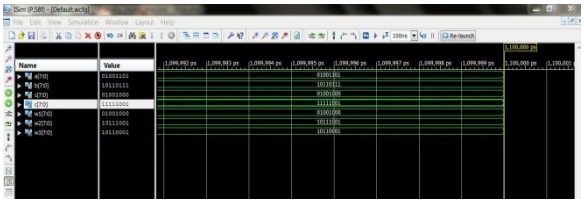


Fig 15: simulation result for stochastic subtraction

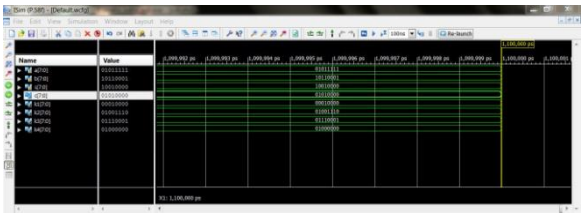
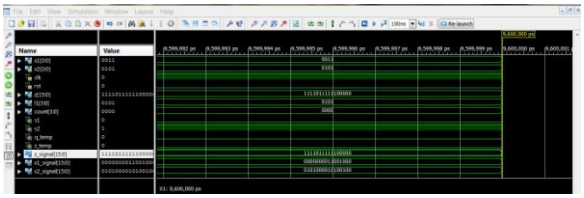


Fig 16: simulation result for stochastic division



V.CONCLUSION

In this paper we mainly discuss about stochastic rationale. Compared to prior approaches its performances speed, cost and area are less and better power consumption. It's reduces the computation time. Why we choose this concept? Now a day's integration of devices are more impact they offer low power and running time, speed characteristics so this project is best use full for future scope. In this paper we conclude the logical arithmetic operations with simple logic. Usage of gates is less and area, delay product is less. Polynomial operations perform with simple usage of AND, MUX logic switches.

VI. REFERENCES

[1]. Nepal, K., Bahar, R. I., Mundy, J., Patterson, W. R., and Zaslavsky, A. 2005. "Designing logic circuits for probabilistic computation in the presence of noise". In *Proceedings of the Design Automation Conference*. 485—490

[2]. Oppelbaum, W. J., Afuso, C., and Esch, J. W. 1967. "Stochastic computing elements and systems". In *Proceedings of the AFIPS Fall Joint Computer Conference*. 635—644.

[3]. M. Mitzenmacher and E. Upfal, Probability and Computing: Randomized Algorithms and Probabilistic Analysis. Cambridge University Press, 2005

[4]. B. Gaines, "Stochastic computing systems," in Advances in Information Systems Science. Plenum, 1969, vol. 2, ch. 2, pp. 37—172

[5]. S. Narayanan, J. Sartori, R. Kumar, and D. Jones, "Scalable stochastic processors," in Design, Automation and Test in Europe,2010, pp. 335—338

[6]. Naderi, A. et al., "Delayed stochastic decoding of LDPC codes," *IEEE Trans. Signal Proc.*, vol. 59, pp. 5617-5626, 2011

[7]. P. Li and D. Lilja, "Using stochastic computing to implement digital image processing algorithms," in *Proc. Int. Conf. Comput. Design*, 2011, pp. 154—161.