A Design Of Unsupervised Learning Of Weight Changing Synaptic Memory BIST

V.Saravanakumar, D. Dolphin Kiruba, S.Tamil Vendan

Abstract— Design for low power testing is a primary concern in modern circuits. In this paper a novel test pattern generator (TPG) is proposed which is more suitable for memory built in self test (BIST) architecture, used for testing of circuits. The objective of the BIST is to reduce power consumption during testing of circuits. The proposed memory BIST requires test pattern generator, Device under test (DUT) output measurement sensor and Artificial Neural Network (ANN). Design approach hinges on the ability to meet strict area and power constrains of the circuits. In this work, we design a high efficient test pattern generator using an unsupervised leaning Artificial Neural Network(ANN) .High precision RAM storage of weighted coefficients during operation or standby ,using this platform we design an HEBBIAN learning algorithm and it is used to train Artificial Neural Network(ANN) ,which generate BIST with high fault coverage and low overhead.

Index Terms—Memory BIST, ANN, DUT, HEBBIAN learning.

1 INTRODUCTION

An integrate on-chip dedicated circuitry for deciding pass/fail based on simple on-chip measurements to achieve a Memory Bulid in self test solution that can be used for low-cost production test as well as in-field periodic test. This dedicated circuitry comes in the form of a neural classifier, which is configured and trained after production to effectively distinguish compliant from non-compliant functionality in the low-cost measurement multi-dimensional space. To investigate this approach, it is designed and fabricated an integrated circuit which serves as a neural network platform. This platform allows us to experiment with various neural classifier topologies and training algorithms. Thereby, it is demonstrate not only the ability of an on-chip neural classifier to accurately produce pass/fail labels in silicon, but also the technology and implementation details that make this approach suitable for on-die integration for fully stand-alone BIST purposes.

In this work, we propose to integrate on-chip dedicated circuitry for deciding pass/fail based on simple on-chip measurementsto achieve a fully stand-alone analog/RF BIST solution that can be used for low-cost production test as well as in-field periodic test. This dedicated circuitry comes in the form of a neural classifier, which is configured and trained after production to effectively distinguish compliant from non-compliant functionality in the low-cost measurement multi-dimensional space. To investigate this approach, we have designed and fabricated an integrated circuit which serves as a neural network platform. This platform allows us to experiment with various neural classifier topologies

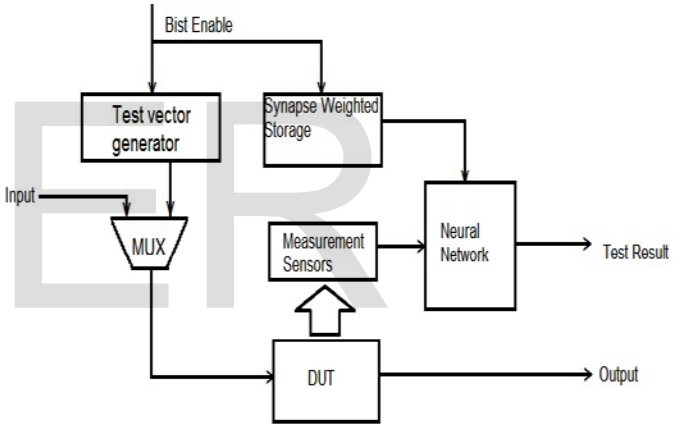


Fig 1. Basic Block Diagram of Memory BIST

In this work, we propose to integrate on-chip dedicated circuitry for deciding pass/fail based on simple on-chip measurements to achieve a memory BIST shown in Fig 1 that can be used for low-cost production test as well as in-field periodic test. This dedicated circuitry comes in the form of a neural classifier, which is configured and trained after production to effectively distinguish compliant from non-compliant functionality in the low-cost measurement multi-dimensional space. To investigate this approach, we have designed and fabricated an integrated circuit which serves as a neural network platform. This platform allows us to experiment with various neural classifier topologies and training algorithms [6]. Thereby, we can demonstrate not only the ability of an on-chip neural classifier to accurately produce pass/fail labels in silicon, but also the technology and implementation details that make this approach suitable for on-die integration for fully stand-alone BIST purposes.

- *V.Saravanakumar is currently working as Assitant Professor in SMK Fomra Institute of Technology ,India.*
- *D.Dolphin Kriubha is currently working as Assitant Professor in Tagore Engineering College. India*
- *S.Tamil Vendan is currently pursuing his PG degree in SMK Fomra Institute of Technology ,India.*

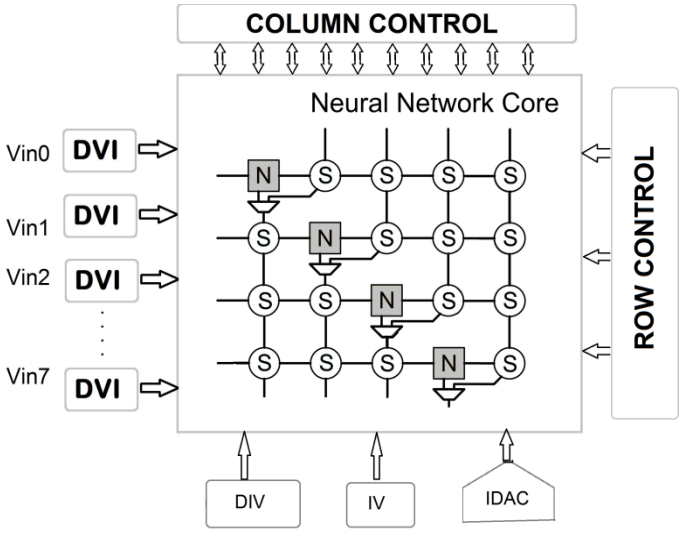
The proposed analog/RF BIST architecture, shown in Figure1, was designed, along with our first neural network experimentation chip. That chip served as proof of-concept that we can adequately learn how to separate passing and failing populations in silicon, and its effectivenesswas demonstrated on synthetic test data from a simple Low Noise Amplifier (LNA). Furthermore, it was a mixed-signal implementation with volatile SRAMs serving as the storage mechanism for holding the learned synapse weights of the neural network. As such, neural classifiers based on this technology would be rather bulky for on-die integration and would require an on-chip EEPROM for holding the synapse weights to support long-term in-field test.

2 Overview Of Design

The block diagram of our ANN is presented in Figure 2. The core of the design is a 30_20 array of synapses (S); each row is locally connected to a corresponding neuron (N). Global connectivity is programmable by means of multiplexors inserted between rows. This allows the core to be configured into several learning structures, including a multilayer perceptron [6] and an ontogenic neural network [9]. The former is a three-layer network of fixed topology with programmable number of neurons in each layer. In contrast, the ontogenic configuration allows for the network topology to be learned dynamically in parallel to its weights. The information processing inside the core is analog; the signals and weights are represented by balanced differential currents. The current signal domain and the translinear principle offer a wide variety of mathematical functions, including multiplication and tanh-like transformation [10], whereas the differential coding allows for four-quadrant multiplication.

A single weight value requires two current sources for differential current storage. It appears that the overall learning ability depends – to a great extent – on the "quality" of these sources. The ideal implementation should have the following characteristics: high precision, non-volatile storage and fast bidirectional update. To this end, we designed a novel current storage cell (CSC) featuring two modes of weight storage: dynamic, for rapid bidirectional update, and non-volatile, for long-term storage of learned weights. The dynamic mode is engaged during training, when the weight values undergo multiple changes. Once the best set of weights is found, their values are copied onto the floating gate transistors for permanent storage. Surrounding the core are the peripheral circuits providing support for fast programming, on figuration storage, and interfacing with the external world. In particular, the "DVI" blocks convert voltage-encoded input signals into balanced differential currents required by the core. Not only does it simplify the interface with the off-chip stimuli generator, but it also allows a direct connection of on-chip sensors with voltage output. The row and column controls isolate individual CSC cells from the array for weight programming. Finally, the circuits at the bottom facilitate network training by transferring some of the programming related tasks on-chip. In particular, a digitally-controlled current source IDAC" generates target currents for dynamic programming of the CSC.

Fig 2 System architecture



Both the "IV" and the "DIV" blocks convert the output current supplied by the core into voltage, which is captured by an off-chip ADC. For high accuracy we use the single-ended current to voltage converter "IV", which is necessary for floating-gate transistor (FGT) programming. This block constitutes a part of a fast current measurement system; the current values are derived from the measured voltages using characterization data of the converter. The "DIV" block converts differential currents produced by the network output into differential voltages. Although less accurate, this is useful for quick network output evaluation in run mode. During training, however, accurate estimation of the error between the network output and the target value is necessary, which can only be furnished by the "IV" converter.

One of the simplest learning method of synaptic weight change is hebbian learning, in which two cells fires simultaneously (Have strong response). Their connections strength or weights increases, where the weight increases between two neurons is proportional to the frequency at which they fire together. Since weights are adjusted according to the correlation of neural inputs

$$\Delta W_{ij}(t) = \gamma * x_j * xi \qquad (1)$$

A general equation of a hebbian learning rule is

$$\Delta W_{ij}(t) = F(x_j, x_i, \gamma, t, \theta) \qquad (2)$$

in which time period and learning node thresholds can be taken into account.

$$wij = \gamma \sum_p y_{ip} y_{jp} \qquad (3)$$

2.1 Storage Cell

The circuit of current storage cell is illustrated in Fig. 3. We use a numerous input FG transistor (FGT) CP1 to store the incoming drain current Iw representing one of the weight value components. The drain current is modulated by the voltage on the FG node, which is itself determined by the FG node charge and the voltages on two control gates. The global voltage vgate1 of the first control gate is shared among all FGTs, while vgate2 is stored locally in the dynamic sample-and-hold (S/H) circuit which consists of the switch transistor Q3 and the MOS capacitor CP3. The low-coupling capacitor CP2 makes Iw much less sensitive to charge leakage and other parasitic effects of the sample-and-hold circuit. The tunneling capacitor CP4 is implemented as a minimum size PMOS transistor with its source, drain and well terminals connected to vtun. The details of non-volatile and dynamic programming are described in [7].

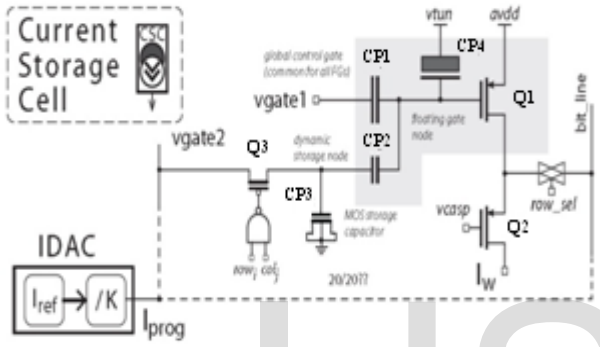


Fig 3 Current Storage Cell

We selected two mechanisms for non-volatile programming of FGTs. Hot-electron injection is used to add electrons to the FG, thus, lowering its voltage and increasing the drain current. Conversely, Fowler-Nordheim (FN) tunneling is used to remove electrons from the FG. Although the two mechanisms allow for bidirectional charge transfer, ue to the difficulties in on-chip routing of high voltages and poor controllability we use FN tunneling for global erase. Injection, on the other hand, is used to program individual FGTs to a target current with high accuracy. First, the FGT of interest is isolated from the containing circuitry by raising the global vcasp and connecting its drain to the bit line. Next, we ramp up the a vdd and apply a series of short pulses to the bit line, measuring the drain current with the "IV" circuit after each pulse. The amount of charge injected during each pulse depends on both the source to-drain voltage and the duration of the pulse. For accurate injection we adopt the algorithm described in [11], however,using a pulse-width instead of a drain voltage modulation.

2.2 Synapse Circuit

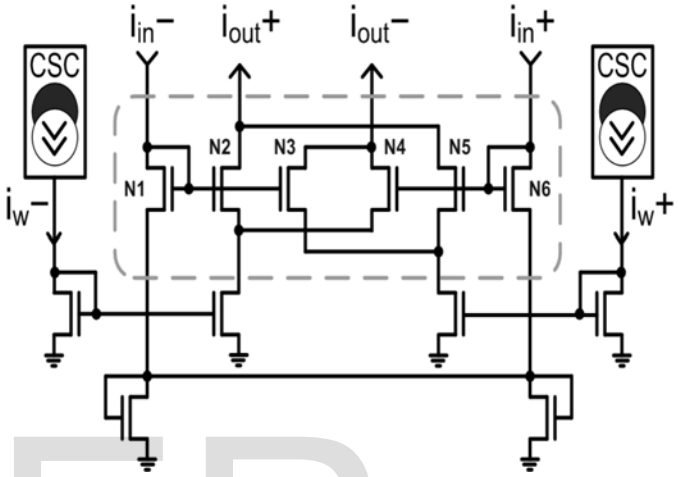
The synapse circuit, illustrated in Figure 4, implements a four-quadrant multiplication function [12]. The circuit features two CSC cells for differential weight components storage and a six-transistor core, enclosed by the dashed box. Provided the core transistors are identical and there is no mismatch, the output differential and common mode currents are obtained by

$$I_{out}^{+} + I_{out}^{-} = \frac{I_{in}^{+} - I_{in}^{-}}{I_{in}^{+} + I_{in}^{-}} \left(I_{w}^{+} - I_{w}^{-}\right) \quad (4)$$

$$I_{out}^{+} + I_{out}^{-} = I_{w}^{+} + I_{w}^{-}$$

where in are the differential components of the input signal and are the differential components of the weight value. The core results in a very compact layout, while most of the area is occupied by the CSC cells due to the dynamic capacitors.

Fig 4 Synapse circuit



2.3 Neural Network

Neural networks have an appealing silicon implementation. Synapses and computational elements can be densely interconnected to achieve high parallel distributed processing ability, which enables them to successfully solve complex cognitive tasks. Neural networks also provide a high percent of healthy, strong and fault tolerance since they comprise numerous nodes that are locally connected, distributing knowledge among the numerous synapses. Thus, intuitively, damage to a few nodes does not impair performance. We are interested primarily in analog implementations of neural networks as, in comparison to a digital implementation, they have superior time response and computational density in terms of silicon mm2 per operations per second and, in addition, they consume extremely low power.

In designing an analog neural network one has to consider a number of important factors. Appropriate connectionist topologies, training algorithms, long-term weight storage are among the most crucial. Furthermore, one has to consider implications of the technology in which a network is to be implemented. Digital CMOS processes, which are becoming more popular for analog/RF circuits, are plagued by process variation, mismatch, noise, environmental factors, etc. The nodes present in the neural network will be giving the each information of the dut , which is given to the neural network.

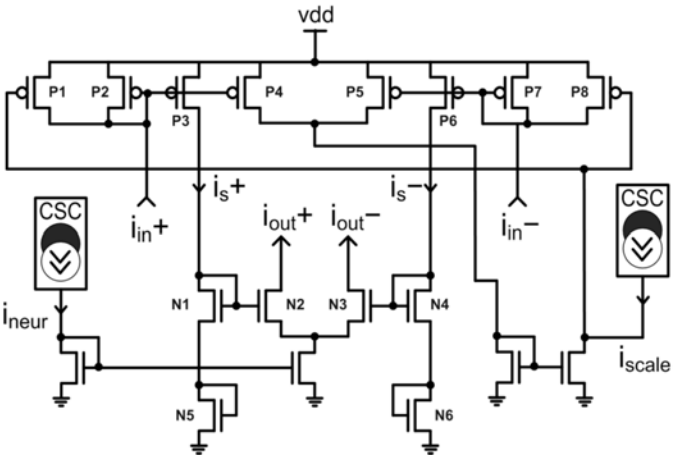


fig 5 neuron circuit

the main function of a neuron circuit is to convert the sum of differential currents from its synapses into a differential voltage. two issues need to be taken into account when designing this circuit. then if the output voltage is passed to the next node, it should be compatible with the input requirements of the synapses, i.e. it should have high common mode. second, the circuit should handle relatively large dynamic range of input currents. while the useful information is contained in the difference, the common mode current may vary significantly depending on the number of connected synapses, as well as on their weight values. in our design, the common mode current ranges from 90 na to 30 μa. a circuit satisfying these requirements is shown in fig. 5.

the central part of the circuit is responsible for common mode cancellation by subtracting the input currents from each other and producing a positive difference. the output currents of the transistors n0 and n7 can be expressed as max $(0,(I_{in}^{-}-I_{in}^{+}))$ and max $(0,(I_{in}^{+}-I_{in}^{-}))$ respectively. thus, only one of the transistors can sink non-zero current at a time. the second stage is a simple current-to-voltage converter composed of two p-channel mosfets. it can be shown that, when the transistors are identical, such circuit exhibits a linear to the first degree characteristic of the following form

$$V = V_{dd} - \frac{I}{2K_p(V_{dd} - 2V_{TP})} \qquad (2)$$

where kp is the transconductance coefficient, vtp is the threshold voltage, and vdd is the supply voltage. the circuit also provides a limiting function when the input current exceeds the internal current flowing through the circuit, thus introducing nonlinearity to the neuron characteristic. notice from the formula above that the slope of the characteristic depends on the kp , which is set at the design stage by specifying transistor sizes. finally, the output of the converter is shifted upwards to meet the requirements of the high common mode input voltage for the synapses in the following layer. this level shifter is a simple source follower circuit where the amount of shift is controlled by vbias. a shift of 1v is used in this design. fig. 8 shows the simulated transfer characteristic of the entire circuit and represents the activation function of the neuron

3 Neural Network Training

Hebbian learning algorithm is a general ethic that states that when two neurons are 'simultaneously' active then the synaptic efficacy between two neurons will be increase, and decrease if not. [12] defines a neural motivated learning algorithm appropriate for the spike response model. In this paper, the learning algorithm was using to simplify it , but maintain the following qualities: Two neurons are 'positively coordinate' if the pre-synaptic neuron spikes before the post-synaptic neuron. If any change in the weightage either increased or decreased will be computed using the spike time and the difference will be correlated using the neural network training.The differences of two neurons are negatively coordinate if the pre-synaptic neuron spikes after the post-synaptic neuron, excepting for a few brief milliseconds around the when both neurons spike at exactly the same instant. The weight is decreased for negatively correlated spike times. The weights are not allowed to grow or shrink without bound. An upper bound and lower bound are determined heuristically. A 'window' of time around the 'equal correlation point' is analyzed. In the cortex or the hippocampus, the learning window probably has a width of 50 - 200ms [12]. A 100ms window was implemented in this project (50ms on either side of the 'equal correlation point is accounted for). This window corresponds to the time period over which chemical activity in real neurons takes place to change synaptic efficacy. The learning rule is executed at intervals of time greater than or equal to the 'window' size. A 100ms interval was chosen as the default for this project, which is the smallest usable value (accounting for all spike activity).The learning rule is the same for both the pyramidal cell and the inhibitory neuron, although this may not be true for real neurons.

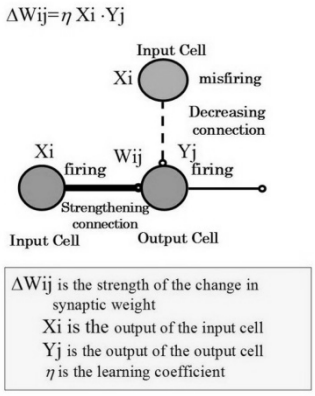


Fig 6 Training Node

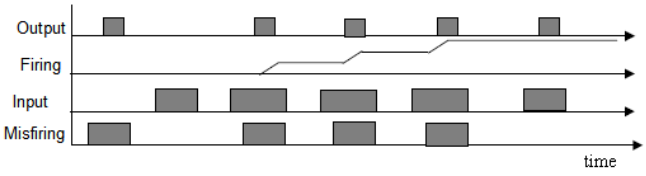


Fig 7 Hebbian Learning Principle

4 DEVICE UNDER TEST

The Device under test used here is an 8 bit multiplier which follows booth's algorithm. Booth's multiplication algorithm is a multiplication algorithm that multiplies two signed binary numbers in two's complement notation. Booth's algorithm examines adjacent pairs of bits of the *N*-bit multiplier *Y* in signed two's complement representation, including an implicit bit below the least significant bit, $y_{-1} = 0$. For each bit y_i, for *i* running from 0 to *N*-1, the bits y_i and y_{i-1} are considered. Where these two bits are equal, the product accumulator *P* is left unchanged. Where $y_i = 0$ and $y_{i-1} = 1$, the multiplicand times 2^i is added to *P*; and where $y_i = 1$ and $y_{i-1} = 0$, the multiplicand times 2^i is subtracted from *P*. The final value of *P* is the signed product.

The representation of the multiplicand and product are not specified; typically, these are both also in two's complement representation, like the multiplier, but any number system that supports addition and subtraction will work as well. As stated here, the order of the steps is not determined. Typically, it proceeds from LSB to MSB, starting at $i = 0$; the multiplication by 2^i is then typically replaced by incremental shifting of the *P* accumulator to the right between steps; low bits can be shifted out, and subsequent additions and subtractions can then be done just on the highest *N* bits of *P*. There are many variations and optimizations on these details. The algorithm is often described as converting strings of 1's in the multiplier to a high-order +1 and a low-order –1 at the ends of the string. When a string runs through the MSB, there is no high-order +1, and the net effect is interpretation as a negative of the appropriate value. Suppose we multiply a*b where x is multiplicand and y is multiplier. The key to Booth's insight is to divide the groups bit of multiplier into 3 parts: the beginning, the middle, or the end of a run of 1s. More specific, the table1 explains in detail

TABLE 1.
Booth Multiplication Operation

y_i	y_{i-1}	*Operation*
0	0	Do nothing
0	1	Add x
1	0	Subtract x
1	1	Do nothing

5 EVALUATION

The BIST designed here is trained by hebbian learning principle. The errors are manually inserted in to the DUT. These errors are found using BIST which was designed. The Altera Quartus II design software provides a complete, multiplatform design environment that easily adapts to your specific design needs. It is a comprehensive environment for system-on-a-programmable-chip (SOPC) design. The Quartus II software includes solutions for all phases of FPGA and CPLD design.

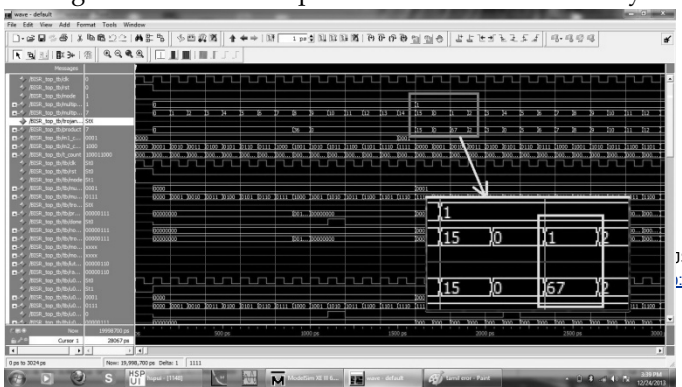


Fig 8. DUT output with error

The above Fig 8 shows the simulation result of error in the booth multiplier.The Neural Network is trained along with DUT to define the rich test vectors with the help of memory BIST. This faults in the DUT is located with these test vectors from Neural Networks. The graph shows the analysis of fault detection using trained neural network. Fig 9 shows the synapse weighted storage data for 100 inputs which is trained to neural network to find rich test vector and Fig 10 true and faulty outputs. Like this we can train N- no. of input test cetors and rich test vectors are stored in ROM.

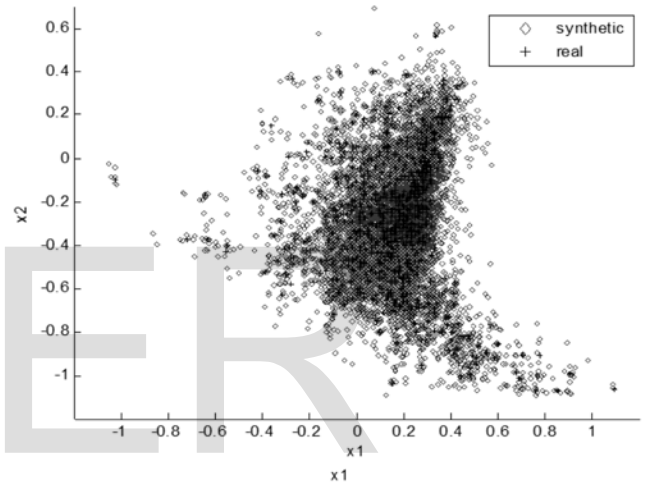


Fig. 9 Original and synthetic devices shown together. One hundred ofsynthetic devices comprise the validation set

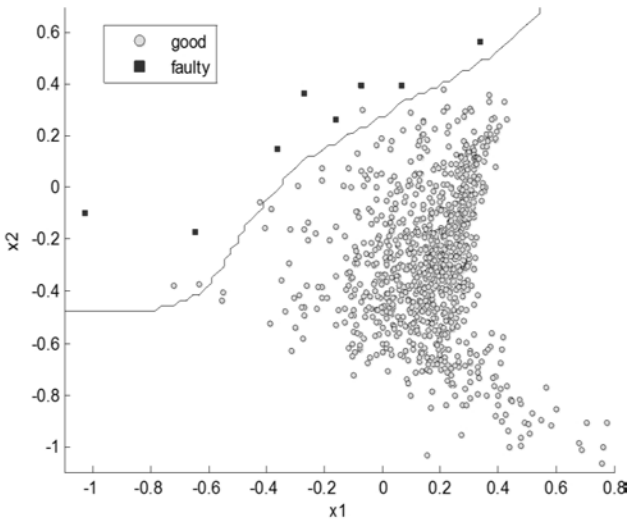


Fig. 10. Original data set consisting of 100 devices generated

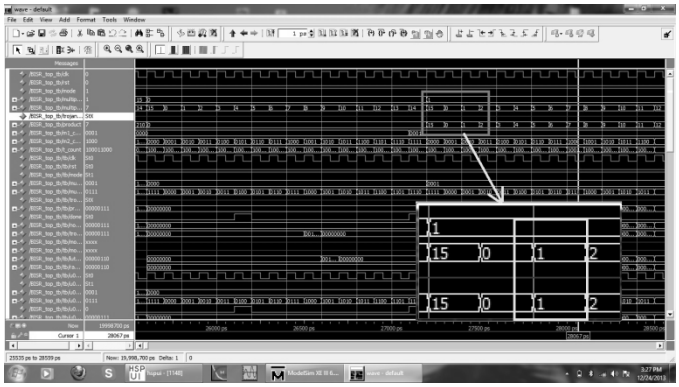


Fig 11 DUT output without error

The above Fig 11 shows the simulation result of error-free booth multiplier. Using this tool the power dissipation of the circuit is found. The total power dissipation is 74.36mW with dynamic power dissipation and static power dissipation is 6.62mW and 46.14mW.

6 CONCLUSION

The memory BIST design with artificial neural network using floating gate technology is used in used in neural classifier. The design of memory BIST with neural networks takes to realization. With this design it is capable to produce an pattern reconfigurable of an memory BIST. The DUT is tested by using training data set produced by the BIST which is able to produce pass/fail during the testing. In the point of improvement using neural network which made the cost effective in the testing.

REFERENCES

[1] H. A. Castro, S. M. Tam, and M. A. Holler, "Implementation and performance of an analog nonvolatile neural network," *Analog Integrated Circuits and Signal Processing*, vol. 4, no. 2, pp. 97–113, 1993.

[2] C.R. Schlottmann and P.E. Hasler, "A highly dense, low power, programmable analog vector-matrix multiplier: The FPAA implementation,"*IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 1, no. 3, pp. 403–411, 2011.

[3] D. Maliuk, H.-G. Stratigopoulos, H. He, and Y. Makris, "Analog neural network design for RF built-in self-test," in Proceedings of the IEEE International Test Conference (ITC), 2010, pp. 23.2.1– 23.2.10.

[4] S. Haykin, Neural Networks: A Comprehensive Foundation (2nd Edition).Prentice Hall, 1998.

[5] S. C. Liu, J. Kramer, G. Indiveri, T. Delbr¨uck, and R. Douglas, Analog VLSI: Circ uits and Principles. MIT Press, 2002.

[6] D. Maliuk and Y. Makris, "A dual-mode weight storage analog neural network platform for on-chip applications," in IEEE International Symposium on Circuits and Systems, 2012, pp. 2889–2892.

[7] A. Bandyopadhyay, G. J. Serrano, and P. Hasler, "Adaptive algorithm using hot-electron injection for programming analog computational memory elements within 0.2% of accuracy over 3.5 decades," IEEE Journal of Solid-State Circuits, vol. 41, no. 9, pp. 2107–2114, 2006.

[8] M. Valle and F. Diotalevi, "A dedicated very low power analog VLSI-architecture for smart adaptive systems," in Applied Soft Computing 4,2004, pp. 206–226.

[9] W. Gerstner, R. Kempter, J.L. van Hemmen, and H. Wagner. *Pulsed Neural Networks*, Hebbian Learning of Pulse Timing in Bradford Books, MIT Press, 1999.

[10] M. Cimino, H. Lapuyade, M. De Matos, T. Taris, Y. Deval, and JB. B´egueret, "A robust 130nm-CMOS built-in current sensor dedicated to RF applications," in IEEE European Test Symposium, 2006, pp. 151–158.

[11] A. Montalvo, "Toward a general-purpose analog VLSI neural network with on-chip learning," IEEE Transactions on Neural Networks, vol. 8, no. 2, pp. 413–423, 1997.

[12] T.-S. Gotarredona B.-L. Barranco and R.-S. Gotarredona, "Compact low-power calibration mini-DAC for neural arrays with programmable weights," IEEE Transactions on Neural Networks, vol. 14, no. 5, pp. 1207–1216, 2003.

[13] M. Jabri and B. Flower, "Weight perturbation: An optimal architecture and learning technique for analog VLSI feedforward and recurrent multilayer networks," IEEE Transactions on Neural Networks, vol. 3, no. 1, pp. 154–157, 1992.

[14] Y.-C. Huang, H.-H. Hsieh, and L.-H. Lu, "A low-noise amplifier with integrated current and power sensors for RF BIST applications," in IEEE VLSI Test Symposium, 2007, pp. 401–408.

[15] E. Acar and S. Ozev, "Defect-oriented testing of RF circuits," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 27, no. 5, pp. 920–931, 2008