ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 8, August 2017

Implementation of Fault Tolerant Embedded Signature Analyzer

[1] Amol Gulab Patil, [2] Usha Jadhav
[1][2] Dept. of Electronics and Telecommunication Engg.
[1][2] D. Y. Patil College of Engineering, Akurdi, Pune, India

***Abstract* - To test and verify arithmetic and logic operations performed by digital circuits an arithmetic and algebraic codes are used. Residue generator is an important unit of hardware implementation of arithmetic code which generates residue of number with respect to check base. The proposed system uses residue generator with arbitrary check base. It is shown that to reduce the probability of error escape, when proposed residue generator is used for detecting arithmetic errors. The proposed generator is embed into a microprogrammable finite state machine to test its operation without adding hardware overhead. The proposed method can be used in arithmetic/algebraic error-control and fault-tolerant digital designs.**

Index Terms: Built-in self-test, cyclic redundancy check codes, design for testability, digital circuits, error correction codes, fault detection, fault tolerant system.

I.INTRODUCTION

The output signals of a sequential circuit depend on the input signals as well as the internal states as shown in fig 1. Sequential circuit is a combination of combinational logic and flip-flop. The inputs to the combinational logic x1,…,xn, and the flip-flop outputs y1,…,ym, are respectively called the primary inputs and the present states (pseudo primary inputs). And the outputs of the combinational logic, z1,...,zm, and the flip-flop inputs, y1,….,Ym, are called the primary outputs and the next states (pseudo primary outputs).
An FSMs is constructed in two ways, One type of construction is designed from logic gates with the intention of minimizing the hardware and maximizing the operational speed.
The second type of construction is combinational logic unit is comprised of a programmable read only memory (PROM). The main advantage of a programmable FSM it uses the same hardware for any stable state, making the design procedure simple and flexible in terms of debugging or upgrading an existing design.

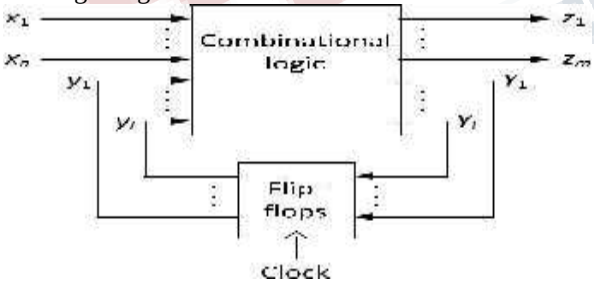

Fig.1. The Huffman model of a sequential circuit.

Both hardwired and programmable FSMs can be executed in application specific integrated circuits (ASICs) or programmable logic devices (PLDs). There are two sorts of PLDs: complex programmable logic device (CPLDs) and field programmable gate arrays (FPGAs). The real distinction between these two is that CPLDs contain combinational logic gates, while FPGAs contain memory units that are ordinarily found as look-up tables. Thus, CPLDs can be effective while executing hardwired FSMs, though FPGAs can be more applicable for programmable FSMs.

II. TESTING CONCEPT

The idea of built in self- test (BIST) suggests that a circuit under test (CUT) is encouraged by input test and the output reactions of this circuit are compressed into a signature that is then compared with the fault-free circuit's signature. The test decision is depends on the comparison result. If the two signatures match, the circuit under test (CUT) is considered to be fault free.

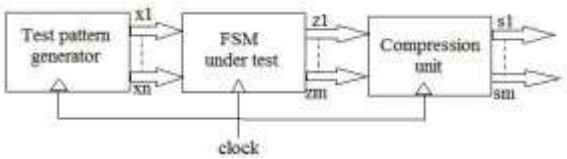

Figure 2. A microprogrammable FSM with built-in self-test

The logic circuit that executes this pressure is appeared in Figure 3. Here, the requested combine, (S1, S2), means the following condition of the signature analyzer. It can be effortlessly confirmed

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 8, August 2017

that after 5 shifts, the remainder left in the circuit is 11.

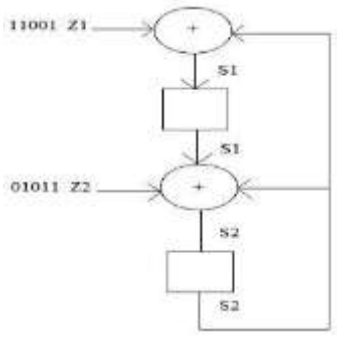

Figure 3. A 2-bit parallel signature analyzer

In the circuit is of an arithmetic nature, better error location capacities are accomplished by applying arithmetic codes. The arithmetic error display was ordinarily received for arithmetic devices and safeguards more productive usage of error control hardware.

III. ARITHMETIC COMPRESSION

The majority part of testing uses of error control coding ideas depend on block codes. In this class of error control coding, the data arrangement is partitioned into blocks. A block is represented by the k-tuple, u = (u_{k-1},…,u0), called a message. The encoder changes each message into a codeword, v = (v_{n-1},.....,v0), and transmits it through a noisy channel. The symbols of u and v are q-ary, where $q = 2^m$, m is the width of the channel. The decoder changes the got message ~v = ($\sim v_{n-1}$,…,$\sim v_0$) into an expected message that must be an imitation of u, if there were no errors present in the transmission channel.

At the point when a error control system is connected to testing, the "noise" (in the form of errors) is created by the inadequate equipment. The decoder is just required to detect these errors without attempting to recover them. The error recognition depends on a syndrome calculation, and the syndrome here is a arithmetic residue. On the off chance that the syndrome is "zero", it is accepted that there are no errors. We will signify the probability of error escape as PNDA.

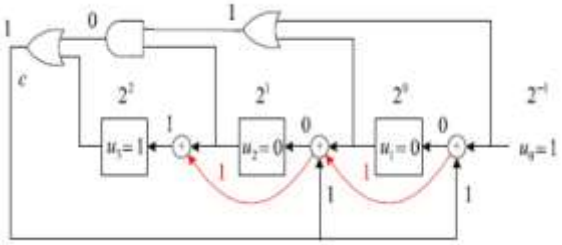

Figure 4. The mod 5 serial residue generator.

The circuit that implements the parallel (3-bit) mod 5 division is presented in figure 5.

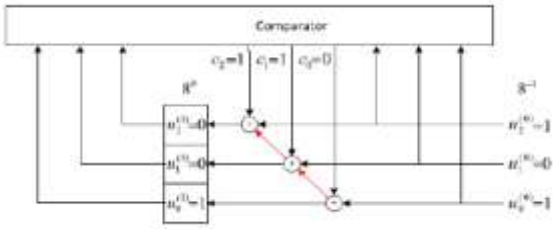

Figure 5. The mod 5 parallel residue generator

The logic expression for the signals c2, c1, c0 that initiate the addition of 8' s complement are more complex;

C2 = (u01 & (u20 ^ u10)) + (u11 & ~u01 & u20) +(u21 &u10 & u00)+(u21 & u20) + (~u11 & u01 & u10 & ~u00)+(u11 & u01 & ~u10);

C1 = (u21 & ~u20 & (~u10 + ~u00)) + (u11 & u01 &u20)+(u01 & u10 & ~u00)+ (~u21 & ~u11 & ~u01 & u20&(u10 +u00))+ (~u11 & & u01 &~u10) +(u01 & ~u20+u00);

C0 = (~u11 & ~u01 &u20 &(u00 +u10))+(~u11 & u01 &~u20 &~u10) + (~u11 & u20 & u10 & u00) +(u11 & ~u20&(~u01 +u10 +u00))+(u11 & ~u20 & u10) +(u11& u01 & u20 & ~u10) + (u21 & u10 & u00)+ (u21 & u20);

IV. CHECK BASE SELECTION

On the off chance that specific conditions are forced on the check base (modulus), the base of the number system, at that point the complexity of the

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 8, August 2017

configuration can be decreased. The modulo g = br-1 residue generator shown in Figure 6 is considered to be a low cost circuit.

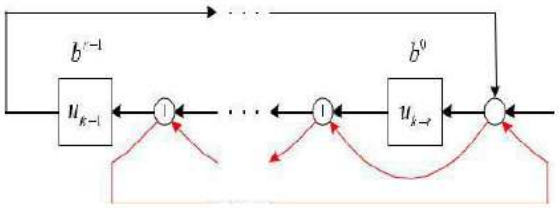

Fig 6. A low cost residue computing circuit.

V. APPLICATION TO MICROPROGRAMMABLE FSM

As talked about already, the microprogrammable usage of a FSM has certain advantages. In addition to known benefits, an especially helpful property of this FSM is that it is in a perfect suited for applying error control coding standards. Many microprogrammable FSMs have a lot of unused memory cells. These repetitive cells can be used to frame a decoder of a error recognizing code (appeared as a pressure unit in Figure 2). Such a decoder would identify operational error in the FSM. This approach is displayed in Figure 7. It is the Moore type FSM (the Mealy FSM can be considered similarly).

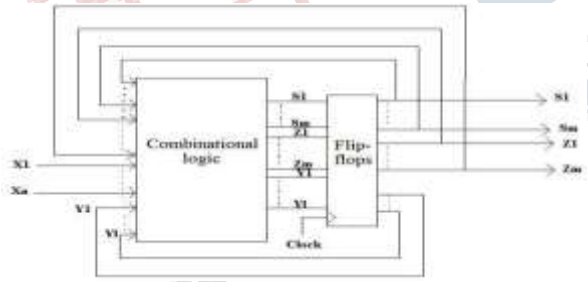

Fig 7. The FSM with the embedded signature analyzer.

Depending upon the idea of the FSM (i.e. regardless of whether it is an arithmetic or non-arithmetic devices), we will recognize two types of error identifying codes: arithmetic and mathematical ones. The FSM memory content is dependent upon the type of code that is selected. However the size of the memory (test hardware overhead) and the speed of the operation are independent of the code type.

VI. EXPRIMENTAL RESULT

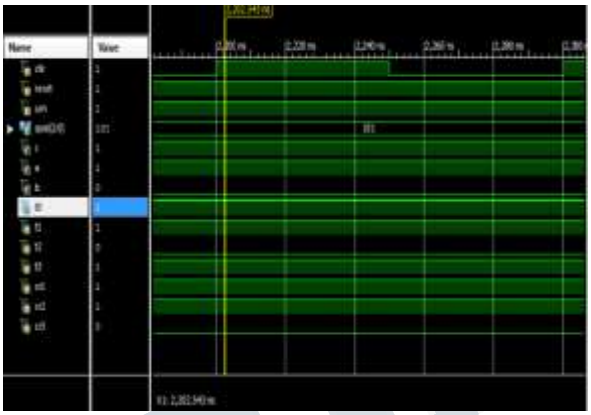

Figure 8. Simulation of the 5 mod serial residue generator

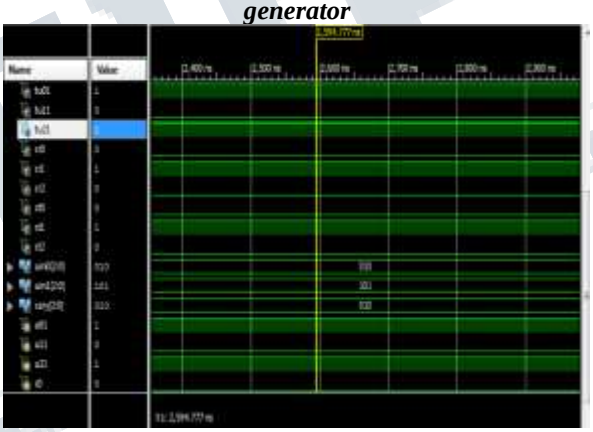

Figure 9. Simulation of the mod 5 parallel residue generator.

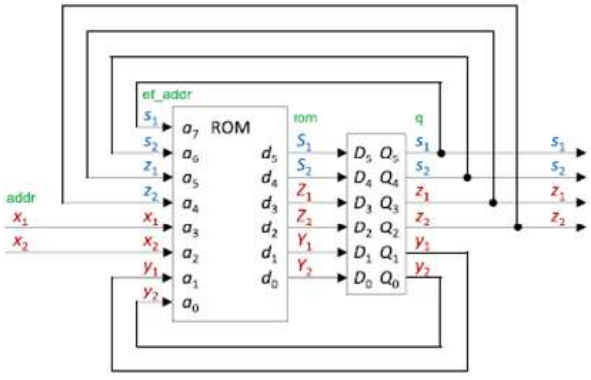

Figure 10. A microprogrammable FSM with 2 bit signature analyzer

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 8, August 2017

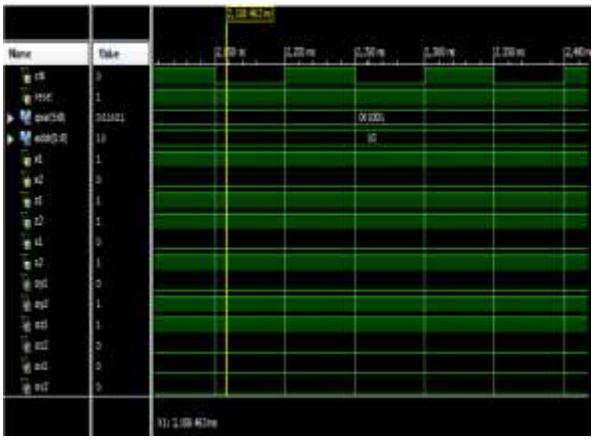

Figure 11. Simulation of the microprogrammable 2 bit signature analyzer

VII. CONCLUSION

Method of designing serial and parallel residue generators with a arbitrary check base and inside an arbitrary number system. Demonstrated how to reduce the probability of error escape, in regard to when these generators are utilized for detecting arithmetic errors. Showed how to use the generators (and also algebraic compactors) for testing micro programmable finite state machines without including additional test hardware (in this way making the modulo generators reconfigurable). And, finally, we simulated the proposed techniques to justify their validity. The created techniques can be utilized in arithmetic error-control coding and in fault-tolerant system designs.

REFERENCES

[1] N. Kehl and W. Rosenstiel, “Circuit level concurrent error detection in FSMs," J. Electron. Test., vol. 29, no. 2, pp. 185_192, Apr. 2013.

[2] J. C.-M. Li, H.-M. Lin, and F.-M. Wang, “Column parity row selection (CPRS) BIST diagnosis technique: Modeling and analysis," IEEE Trans. Comput., vol. 56, no. 3, pp. 402_414, Mar. 2007.

[3] H.-M. Lin and J. C.-M. Li, “Column parity and row selection (CPRS): A BIST diagnosis technique for multiple errors in multiple scan chains," in Proc. IEEE Int. Test Conf. (ITC), Nov. 2005, pp. 1009_1117.

[4] A. Chandra and K. Chakrabarty, “System-on-a-chip test-data compression and decompression architectures based on Golomb codes," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 20, no. 3, pp. 355_368,
Mar. 2001.

[5] M. Ottavi, G. C. Cardarilli, D. Cellitti, S. Pontarelli, M. Re, and A. Salsano, “Design of a totally self checking signature analysis checker for finite state machines," in Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Syst., Oct. 2001, pp. 403_411.

[6] U. Sparmann and S. M. Reddy, “On the effectiveness of residue code checking for parallel two's complement multipliers," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 4, no. 2, pp. 227_239, Jun. 1996.

[7] J. Duran and T. E. Mangir, “A design approach for a microprogrammed control unit with built in self test," ACM SIGMICRO Newslett., vol. 14, no. 4, pp. 55_60, Dec. 1983.

.