ISSN:2249-5789
Ch.Vinaykumar et al , International Journal of Computer Science & Communication Networks,Vol 2(3), 404-406

ARCHITECTURE

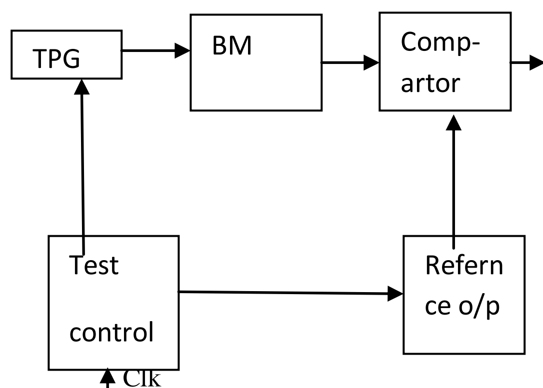

Fig 6: Architecture of testing circuit of bench Mark circuit

The essential Testing circuitry, shown in Figure 1 typically includes a test pattern generator (TPG), reference output analyzer, and test controller. The TPG

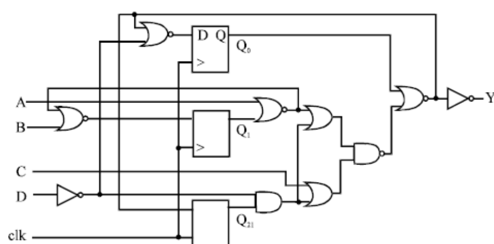

Fig7 : s27 Bench mark circuit

LFSR

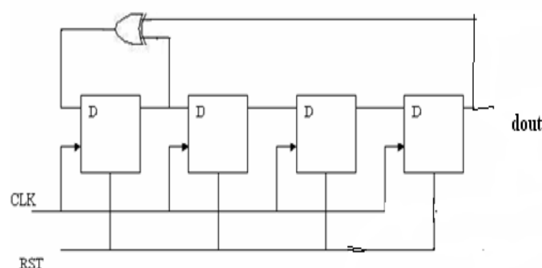

Fig 8 Block Diagram of Pn-sequence generator

The LFSRs represent the simplest and most commonly used pseudo-random TPG hardware, the efficiency of an LFSR is far from optimum in terms of fault coverage and testing runtimes. The test generated by an LFSR is usually up to several orders of magnitude longer than a similar ATPG test. In general, PRPG fault coverage trend is characterized by such peculiarities like fast initial growth and too long time to complete. Figure8 illustrates this fact clearly showing the corresponding sections of the curve. The slow growth section is mostly caused by existence of pseudo-random pattern resistant faults a.k.a. hard-to-test faults (HTTF) which are usually very difficult to handle by PRPG-based methods. In case of large numbers of HTTFs in the DUT, the maximum fault coverage cannot be achieved by LFSR sequence in realistic time at all. Due to this difficulty, as well as great importance of testing circuitry, there is a huge amount of works that target improvement of PRPG efficiency. Each of these works has certain advantages and disadvantages over the others.

A big portion of research on testing circuitry is devoted to study of alternative PRPG types that have better saturation properties compared to the one of LFSR due to better fulfilling randomness criteria for generated sequences.These are, for instance, Cellular Automata and GLFSR. However, the randomness has only been empirically proven to improve the quality of testing. On the contrary, larger designs, especially those that contain random pattern resistant HTTFs, need special treatment. As the result, mixed-mode methods that combine PRPG and ATPG patterns gained a major attention.

From the implementation point of view, the available methods can be classified into two major categories accordingly to the way they handle ATPG patterns: a) memory-based methods and b)special encoding and embedding hardware based methods

CONCLUSION

We showed that in full scan benchmark circuits, undetectable single stuck at faults tend to cluster in certain areas. In order to provide a target for an extended set of faults we introduced double stuck-at fault to improve the coverage of these areas. Based on undetectable faults and detectable faults that are adjacent to them we defined the double stuck – at faults. We showed through experiments consisting of test generation and fault simulation that the coverage areas with undetectable faults can be extended.