VHDL Implementation of GCD Processor with Built in Self Test Feature

Rekha Devi
University Institute of Engineering and Technology, Panjab University, Chandigarh

Jaget Singh
University Institute of Engineering and Technology, Panjab University, Chandigarh

Mandeep Singh
Centre for Development of Advanced Computing (CDAC), Mohali, India

ABSTRACT
The Very Large Scale Integration (VLSI) has a dramatic impact on the growth of digital technology. VLSI has not only reduced the size and the cost, but also increased the complexity of the circuits. Due increase there is a problem of circuit testing, which becomes increasingly difficult as the scale of integration grows. One solution to this problem is to add logic to the IC so that it can test itself. In this paper we have design GCD (greatest common divider) processors in VHDL with BIST capability and compared the area overhead of with and without BIST.

Keywords
Built in Self Test, VLSI Testing, Greatest common Divisor, Finite State Machine

1. INTRODUCTION
Testing of integrated circuits (ICs) is of crucial importance to ensure a high level of quality in product functionality in both commercially and privately produced products. The impact of testing affects areas of manufacturing as well as those involved in design. This desire to attain a high quality level must be tempered with the cost and time involved in this process. These two design considerations are at constant odds. It is with both goals in mind (effectiveness vs. cost/time) that Built-In-Self Test (BIST) has become a major design consideration in Design -For-Testability (DFT) methods. As digital systems become more complex, they become much harder and more expensive to test. One solution to this problem is to add logic to the IC so that it can test itself. This is referred to as "Built in self Test" (BIST) .BIST approach is beneficial in many ways. First, it can reduce dependency on external Automatic Test Equipment (ATE). This aspect impacts the cost/time constraint because the ATE will be utilized less by the current design and can be used elsewhere or on the other Devices [1].

GCD stands for greatest common divisor. Computation of the GCD of long integers is heavily used in computer algebra systems because it occurs in normalization of rational numbers and other important sub algorithms. While performing experiments, half of the time is spent for calculating GCD of long integers. There are various fields where this division is used e.g. channel coding, cryptography, error correction and code construction. There are various algorithms to calculate GCD. [4].

The most basic algorithm to calculate GCD is Euclid's algorithm Many practical algorithms have dynamic dependency structure in their computation which is undesirable for VLSI hardware implementation. e.g. in case Euclid's algorithm has dynamic dependency, so it can be implement that algorithm which has dynamic dependency on Euclid's algorithm. So the resulting algorithm is mapped to a linear systolic array which is area efficient and achieves maximum throughput with pipelining.

A representative architecture of the BIST circuitry as it might be incorporated with the GCD(as CUT part) is illustrated in the block diagram of Figure 1. This BIST architecture includes two essential functions as well as two additional functions that are necessary to facilitate execution of the self-testing feature while in the system. The two essential functions include the test pattern generator (TPG) [10] and output response analyzer (ORA). While the TPG produces a sequence of patterns for testing the GCD. The other two functions needed for system-level use of the BIST include the test controller (or BIST controller) and the input isolation circuitry [5].

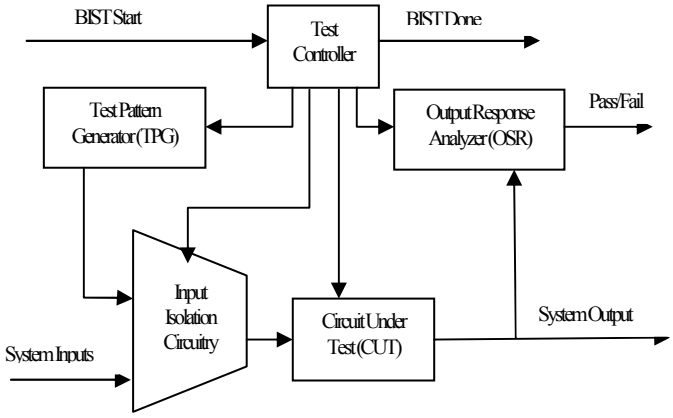

Figure 1. Basic BIST Architecture

Aside from the normal system I/O pins, the incorporation of BIST may also require additional I/O pins for activating the BIST sequence (the BIST Start control signal), reporting the results of the BIST (the Pass/Fail indication), and an optional indication (BIST Done) that the BIST sequence is complete and that the BIST results are valid and can be read to determine the fault-free/faulty status of the CUT.

2. RELATED WORK

Many researchers have been working on the BIST implementation and GCD related research some of them are given below:

Crouch et al. [8] has discussed the main point is Built-In-Self Test (BIST) architecture for sequential circuits based on cellular Automata (CA).Yamani et al. [2][11], the BIST technique incorporated into the UART design before the overall design is synthesized by means of reconfiguring the existing design to match testability requirements. Shalanwa et al [7] designed GCD processor with FSM using Mealy Machines with NAND gates for the computation of two numbers. the method employed here is utilized purely state diagrams, state table, excitation table and Karnough Map and although the Machine is capable of the GCD of any number provided the inputs is stated correctly Chen et al. [9], this paper shows the extended GCD algorithm is very useful for data dependence tests, for example, the power test, on supercomputers. , they improved the sequential extended GCD algorithm.

3. GCD PROCESSOR IMPLEMENTATION WITH BIST FEATURE

3.1 FSM Based design of GCD Processor

We have first write algorithms for calculation of GCD, after this data paths and control unit is designed for GCD processor. In this model, both the FSM and the data path circuits are manually constructed as separate units. The FSM and the data path are connected together in an enclosing unit using the control and status signals.

```
Input X
Input Y
While (X! =Y)
{IF (X>Y)
THEN X=X-Y
ELSE Y=Y-X
END IF}
OUTPUT X
```

Figure 2: Euclid's algorithm

The algorithm shown in Figure 2 has five data manipulation statements in lines 1, 2, 5, 7, and 10. There are two conditional tests in lines 3 and 4. We can conclude that the data path requires two 8-bit registers, one for variable X, and one for variable Y, and a subtractor. The dedicated data path is shown in

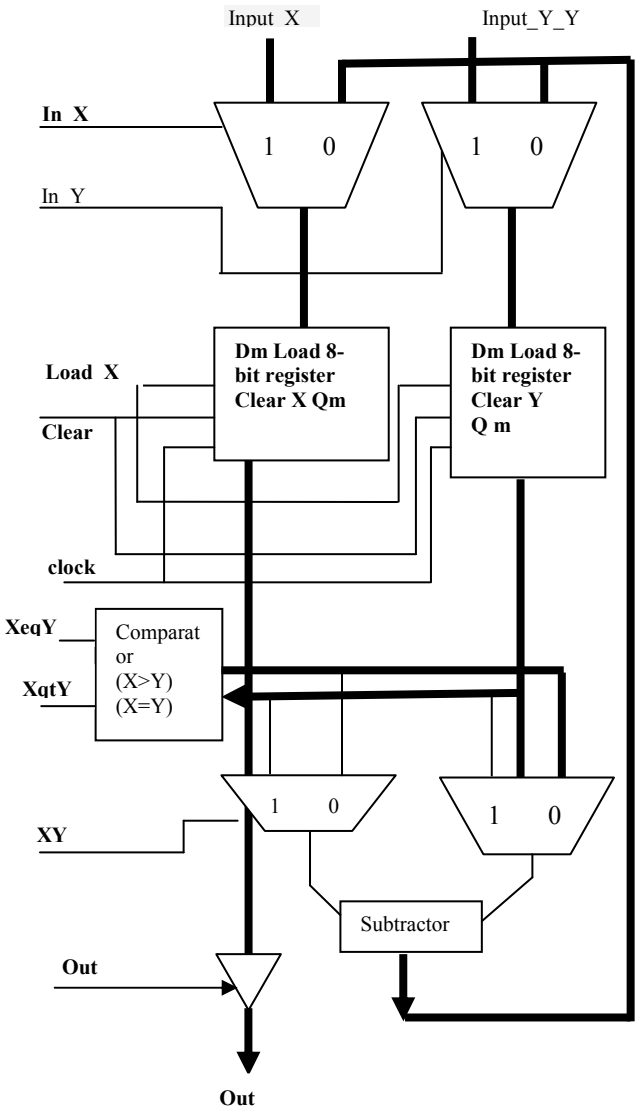

Figure 3: Data paths of GCD processor.

In Figure 3, We have a 2-to-1 mux for the input of each register because for each register, we need to initially load it with an input number, and subsequently load it with the result from the subtractor. The two control signals, In_X and In_Y, select which of the two sources are to be loaded into the registers X and Y respectively. The two control signals, Load_X and Load_Y, load a value into the respective register.

3.2 GCD Processor implementation

The VHDL implementation was done in Xilinx's 9.2 and the RTL View of GCD Processor is shown below in Figure 4.

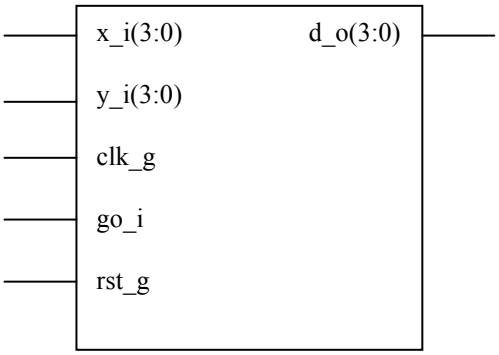

Figure 4: Top Level RTL view of GCD processor

In the above figure the GCD having x_i, y_i, rst_g, clk_g, go_i, as input signals and d_o as output signal which is assigned required values to check its final functioning.

3.3 BIST Implementation for GCD Processor

The RTL synthesis of GCD Processor with BIST Features is shown below in figure 5

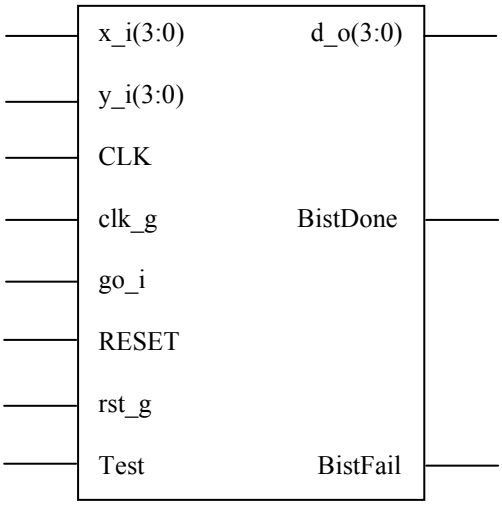

Figure 5: Top Level RTL view of GCD processor with BIST Features

The figure shows the GCD Processor with BIST Features having x_i, y_i, rst_g, clk_g, go_i, CLK,Test as input signals and d_o, BistDone and BISt Fail as output signal which is assigned required values to check its final functioning.

4. SIMULATION AND RESULTS

The Spartan-3 Field Programmable Gate Arrays (FPGA) from Xilinx is used for realization of the simple GCD and GCD processor with BIST capability. The processor is designed and implemented in VHDL using FSM .the simulation results are discussed in 4.1 and 4.2 sections. Comparison of area overhead is given in table 1 and shown graphically in figure 8 and 9

4.1 Simulation Result of GCD processor

The figure 6 shows the simulation results of GCD. GCD is having x_i, y_i, rst_g, clk_g, go_i, as input signals and d_o as output signal which is assigned required values to check its final functioning. x_i and y_i are assigned to 0XC & 0X4 respectively whose GCD is 0X4 as appears at d_o after 9 clock cycles which are used to calculate intermediate values as appears in the simulation results.

4.2 Simulation results of GCD processor with BIST Application

The figure 7 shows the simulation results of GCD processor with BIST, having two signals when Test signal is high BIST section is activated, and its gives two output signals BIST fail and BIST done. When Test is Low then this circuit operates in its normal mode of GCD generation whose output signals are Dout .

Table 1 : Comparison of area overhead of GCD processor with and without BIST

Hardware Components	GCD with BIST feature	GCD Processor
Number of Slice Registers	67	27
Number of occupied Slices	71	70
Number of 4 input LUTs	106	42
Number of bonded IOBs	20	15
Total equivalent gate count	1179	653
Additional JTAG gate count for IOBs	960	720

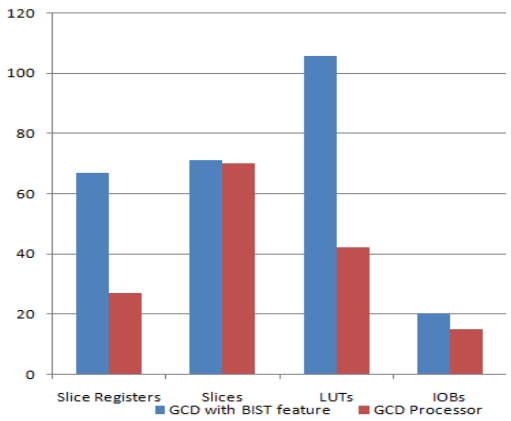

Figure 8: Comparison of Slice, FF, LUTs for GCD processor with and without BIST

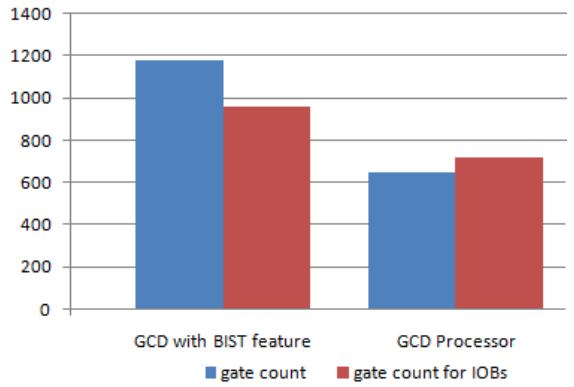

Figure 9: Comaprision of Gate count and IOB with and without BIST

5. CONCLUSION

In this paper, the analysis of area overhead for implementing BIST technique in GCD Processor is carried out. Firstly Simulation is done for the verification of functionality and then BIST module is integrated with GCD processor for self testing. Hardware comparison was made as shown in figure 9 and 10 .it is concluded that BIST implementation for GCD has increases area overhead BIST can be used to perform these especial tests with additional on-chip test circuits, eliminating the need to acquire such high-end testers

6. REFERENCES

[1] Bushnell M. L. and Agrawal V. D.(2000) Essentials of Electronic Testing. Kluwer Academic Publishers.

[2] Mohd Yamani Idna Idris "A VHDL Implementation Of UART design with BIST capability" Malaysian Journal of Computer Science, Vol. 19 (1), 2006

[3] S. Wang, "Generation of low power dissipation and high fault coverage patterns for scan-based BIST", Proceeding of International Test Conference, 2002, pp. 834 –843.

[4] Z. Navabi, "VHDL Analysis and Modeling of Digital Systems", McGraw-Hill Inc., 1991.

[5] Charles E. Stroud "A Designer's Guide to Built-in Self-Test" Kluwer Academic Publishers 2002

[6] C. H. Roth " Digital System Design Using VHDL" PWS Publishing Company, 1998

[7] D.A. Shalangwa and S. Samaila , "design of a mealy state machine to realized the GDC of Two Numbers", journal of engineering and Applied Sceinces 4(5-6) pp 338

[8] Crouch, A. (1999), "Design-for-test for Digital IC's and Embedded Core Systems", Prentice Hall, 347 pp

[9] Pou-Yah Wu & Julian Chuen-Liang Chen, "Parallel Extended GCD Algorithm" Parallel Processing Symposium, 1994.

[10] Singh, B.; Khosla, A.; Bindra, S.; , "Power Optimization of Linear Feedback Shift Register (LFSR) for Low Power BIST," Advance Computing Conference IACC IEEE International., vol., no., pp.311-314, 6-7 March 2009

[11] Shikha Kakar, Balwinder Singh, and Arun Khosla" Implementation of BIST Capability using LFSR Techniques in UART" International Journal of Recent Trends in Engineering (IJRTE), Volume 1, Number 3, May 2009 Issue on Electrical & Electronics Page(s): 301-304

.

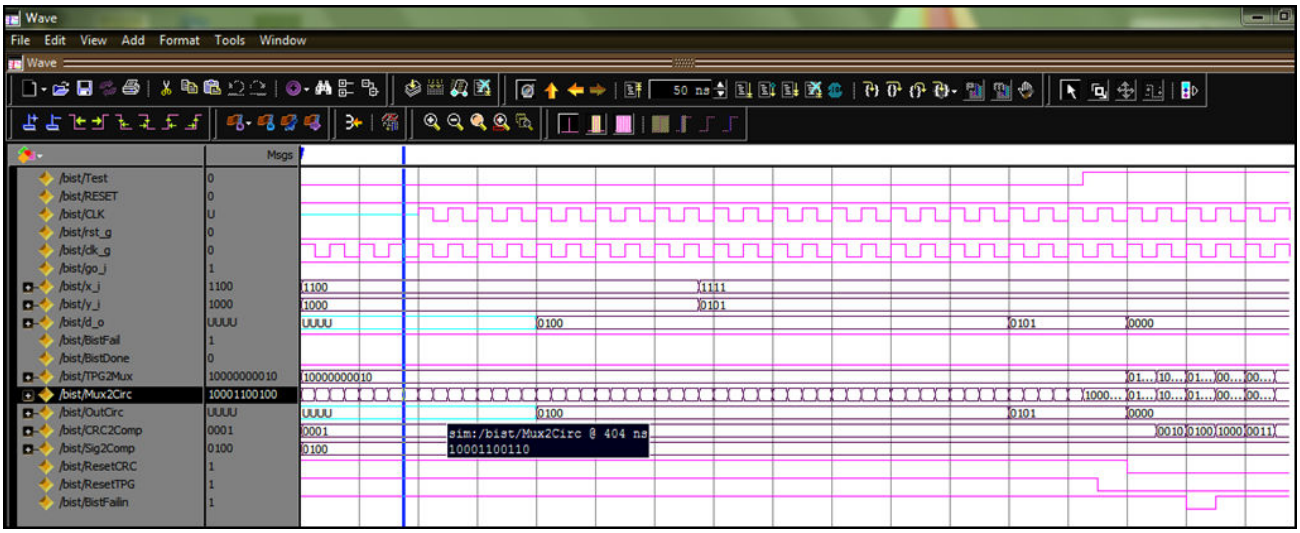

Figure 6: Simulation results of GCD Processor

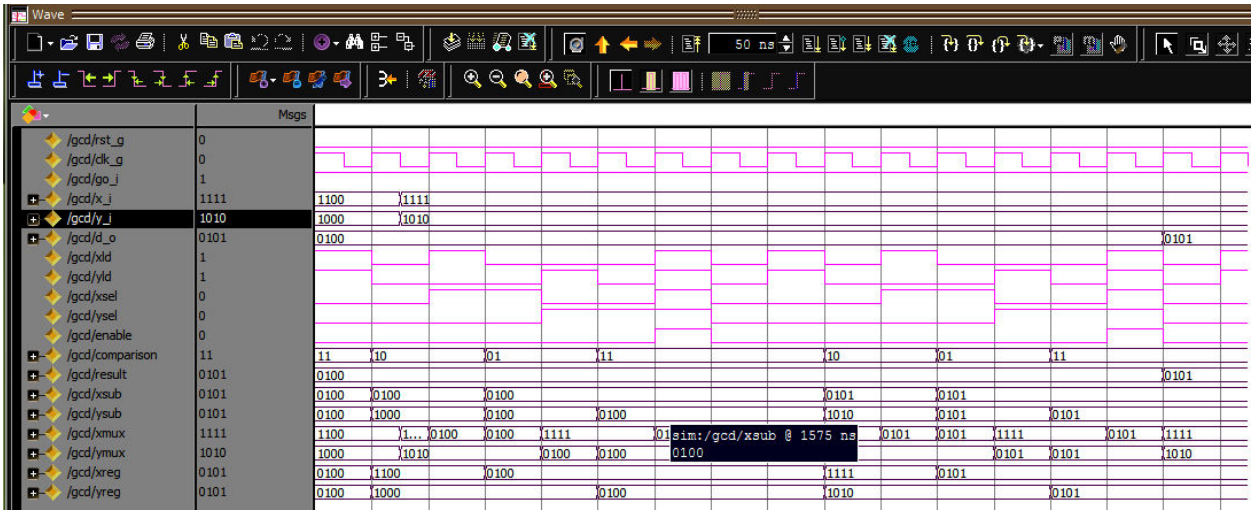

Figure 7: Simulation Result of GCD processor with BIST feature