Comparative Analysis of Stein's and Euclid's Algorithm with BIST for GCD Computations

[1]Sachin D.Kohale , [2]Ratnaprabha W.Jasutkar

[1] Post Graduate Student , G. H. Raisoni College of Engineering
Nagpur , Maharashtra , India

[2]Assistant Professor, Department of Computer Science and Engineering ,G. H. Raisoni College of Engineering
Nagpur , Maharashtra , 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 cost, but also increased the complexity of the circuits. Due to increase in complexity, it is difficult to test circuits. To reduce this problem of testing, it is advantageous to add another IC along with it which will test and correct errors by itself. This IC is known as Built in Self Test(BIST).In this paper , we are particularly concentrating upon finding the comparative parameters of Euclid's and Stein's Algorithm , which is used to find greatest common divisor(GCD) of two non negative integers. Thus, the best parameters to be found can be used effectively for finding gcd , This indirectly reduces time for calculating greatest common divisor , which is being used very frequently in communication applications.

Keywords: *Built In Self Test(BIST), Euclid's Algorithm, Linear Feedback Shift Register, Stein's Algorithm ,VLSI testing.*

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. As the complexity of circuits continues to increase, high fault coverage of several types of fault models becomes more difficult to achieve with traditional testing paradigms. 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). Secondly , it can be used elsewhere or on the other Devices.

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 algorithms to calculate Greatest Common Divisor(gcd).

In this paper , we are using Euclid's and Stein's algorithm for calculating Greatest Common Divisor(gcd) of two non-negative integers. Theoretically , Stein's algorithm is better than Euclid's algorithm for gcd calculations.

1.1 Built in Self Test(BIST)

Built-In Self Test (BIST)[3] is a technique of integrating the functionality of an automatic test system onto a chip. It is a Design for Test technique in which testing is accomplished through built in hardware features. The general BIST architecture has a BIST test controller which controls the BIST circuit, test generator which generates the test address sequence ,response verification as a comparator which compares memory output response with the expected correct data. The BIST controller can be implemented by either hardwired logic in the form of finite state machine(FSM) , microcode controller or based on processor.

2. Linear Feedback Shift Register(LFSR)

Linear Feedback Shift Register(LFSR)[2] is a circuit consisting of flip-flops connected in series with each other. The output of one flip-flop is connected to the input of the next flip-flop and so on. The feedback polynomial which is also known as the characteristics polynomial is used to determine the feedback taps which in turn determines the length of the random pattern generation.

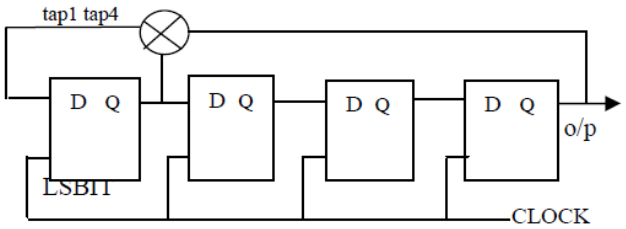

Fig. 1 Conventional LFSR or XOR LFSR.

In Computing , a Linear Feedback Shift Register(LFSR) is a shift register whose input bit is a linear function of its previous state. The most commonly used linear function of single bits is XOR. Thus, an LFSR is most often a shift register whose input bit is driven by the exclusive-or(XOR) of some bits of the overall shift register value.

The initial value of the LFSR is called seed, and because the operation of the register is deterministic, the stream of values produced by the register is completely determined by its current(or previous) state. Likewise, because the register has a finite number of possible states, it must eventually enter a repeating cycle. However, an LFSR with a well-chosen feedback function can produce a sequence of bits which appears random and which has a very long cycle. Applications of LFSR including generating pseudo-random numbers, pseudo-noise sequences, fast digital counters and whitening sequences. Both hardware and software implementations of LFSR are common.

3.Euclid'sAlgorithm

In Mathematics, the Euclidean algorithm or Euclid's algorithm[3], is an efficient method of computing the greatest common divisor(gcd)[7] of two integers, also known as greatest common factor(gcf) or highest common factor(hcf). It is named after the Greek Mathematician , Euclid.

In its simplest form, Euclid's algorithm starts with a pair of positive integers and forms a new pair that consists of the smaller number and the difference between the smaller and larger numbers. The process repeats until the numbers are equal. That number then is the greatest common divisor of the original pair.

Basically Euclid algorithm [3] can be described as

$$gcd(a , 0) = a \quad (1)$$
$$gcd(a , b) = gcd(b , a \bmod b) \quad (2)$$

If arguments are both greater than zero, then

$$gcd(a , a) = a \quad (3)$$
$$gcd(a , b) = gcd(a - b , b) ; \text{if } b < a \quad (4)$$
$$gcd(a , b) = gcd(a , b - a) ; \text{if } a < b \quad (5)$$

For ex. gcd(20 , 0) is 20 [1]. Similarly, gcd(20 , 10) [4] is same as gcd((20-10) , 10) = gcd(10 , 10) = 10.

4.Stein'sAlgorithm

This algorithm is also known as binary gcd algorithm. It is algorithm that computes the greatest common divisor of two nonnegative integers. It gains a measure of efficiency over the ancient Euclidean algorithm by replacing divisions and multiplications with shifts, which are cheaper when operating on the binary representation used by modern computers. This is particularly critical on embedded platforms that have no direct processor support for calculations of divison. .

Basically Stein's algorithm[4] can be described as

$$gcd(0 , v) = v \quad (6)$$

$$gcd(u , 0) = u \quad (7)$$

$$gcd(0 , 0) = 0 \quad (8)$$

If u and v are both **even**, then

$$gcd(u , v) = 2.gcd(u/2 , v/2) \quad (9)$$

If u is **even** and v is **odd,** then

$$gcd(u , v) = gcd(u/2 \;\; v) \quad (10)$$

Similarly u is **odd** and v is **even** then

$$gcd(u , v) = gcd(u , v/2) \quad (11)$$

If u and v are both **odd** and **u is ≥ v**, then

$$\gcd(u, v) = \gcd((u - v)/2, v) \quad (12)$$

If both are **odd** and **u < v** , then

$$\gcd(u, v) = \gcd((v - u)/2, u) \quad (13)$$

For ex. gcd(0 , 22) is 22 [6]. Also, gcd(33 , 0) is 33 [7]. Similarly , gcd(21 , 22) is same as gcd(21 , 11) [11]. Also, gcd(21 , 41) is same as gcd((41 - 21) / 2 , 21) is again same as gcd(10 , 21)[13].

5. Comparison of Euclid's Vs. Stein's Algorithm

Table 1: Euclid's Algorithm for BIST with 8-bit input data

Device	***XC3S50***	***XC3S200***	***XC3S400***	***XC3S1000***
No. of Slices	807	825	825	825
No. of Slice Flip Flops	16	16	16	16
No. of 4 input LUT's	1613	1613	1613	1613
No. of bounded IOB's	35	35	35	35
Total Equivalent gate count for design	13503	13503	13503	13503
Additional JTAG gate count for IOB's	1680	1680	1680	1680
Power Consumption (in milli-Watts)	Cannot be calculated due to large design	37	56	92

Table 2:Stein's Algorithm for BIST with 8-bit input data

Device	***XC3S50***	***XC3S200***	***XC3S400***	***XC3S1000***
No. of Slices	74	74	74	74
No. of Slice Flip Flops	48	48	48	48
No. of 4 input LUT's	131	131	131	131
No. of bounded IOB's	22	22	22	22
Total Equivalent gate count for design	1395	1395	1395	1395
Additional JTAG gate count for IOB's	1056	1056	1056	1056
Power Consumption (in milli-Watts)	24	37	56	92

Table 1 and Table 2 shown above shows different parameters that has been calculated after selecting Spartan 3 Device, using Euclid's and Stein's Algorithm for BIST with 8-bit input data. Under Spartan 3 Device, different parameters is being calculated for different families such as XC3S50 , XC3S200 , XC3S400 and XC3S1000.

From the above tables (1 and 2), it has been observed that number of LUT's is more in case of Euclid's Algorithm as compared to number of LUT's in Stein's Algorithm. Also, total equivalent gate needed for designing hardware is more in Euclid's Algorithm as compared to total equivalent gate needed in Stein's Algorithm. Power consumption is also one of the factor in both algorithms.

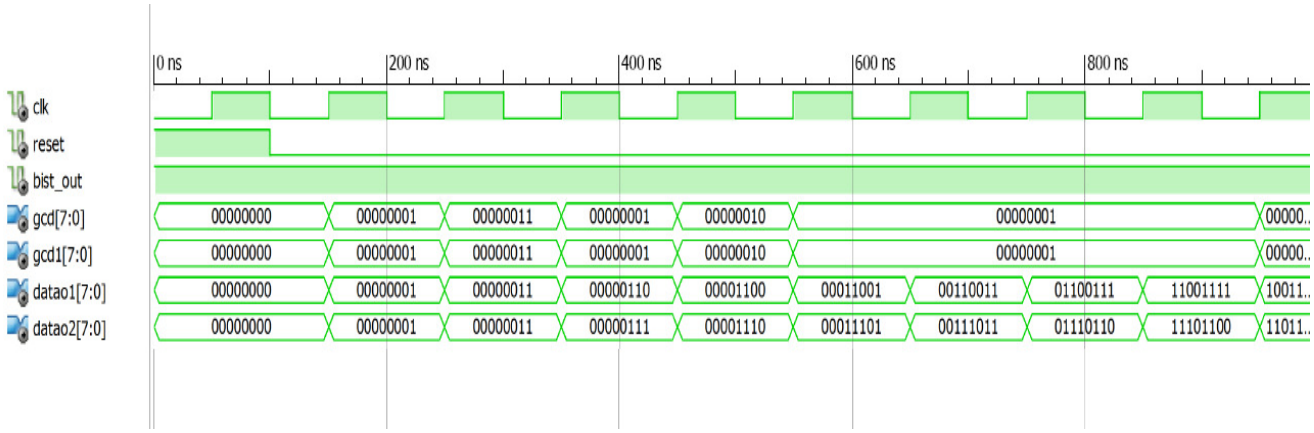

Fig. 2. Euclid's 8 bit data input with BIST.

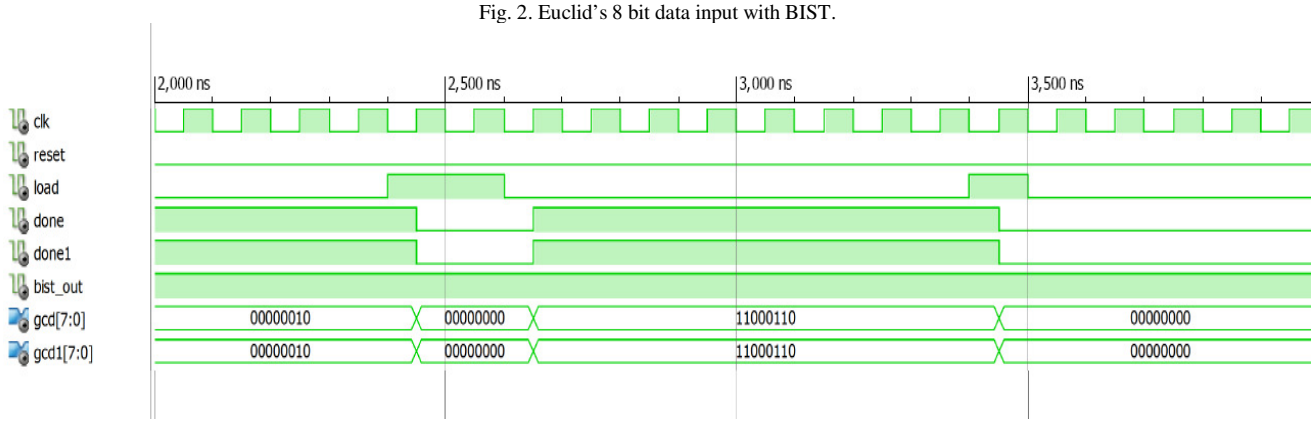

Fig. 3. Stein's 8 bit data input with BIST.

6. Conclusion

Thus, After comparing Euclid's and Stein's algorithm , we can conclude that Stein's algorithm is better than Euclid's Algorithm practically. Although , theoretically it is already being proved. From table 1 and table 2 , it is been observed that power consumption is less for XC3S50 family of Spartan 3 device. Also, as the number of Look-Up Tables(LUT's) needed is less in Stein's Algorithm as compared to that of Euclid's Algorithm, less function generators is being implemented in CLB's. Also, gate count needed is less in Stein's Algorithm as compared to that of Euclid's Algorithm. Thus, due to less number of gates needed, gcd processor implemented using Stein's Algorithm is better for calculating greatest common divisor(gcd) of two non-negative integers. Fig. 2 and Fig. 3 shown above shows comparison of output waveforms of Euclid's and Stein's Algorithms with BIST considering 8-bit data input

Appendix

Slice : Two slices form a CLB within Spartan®-II and Virtex® families. This is a specific example of a comp type that corresponds to the basic fabric of logic in all FPGA's.

Look-Up Table (LUT) : Look-up tables (LUTs) are used to implement function generators in CLBs. Four independent inputs are provided to each of two function generators (F1-F4 and G1-G4). These function generators can implement any arbitrarily defined Boolean function of four inputs. The H function generator can implement any Boolean function of four inputs.

IOB (input/output block) : A collection or grouping of basic elements that implement the input and output functions of an FPGA device.

Gate : An integrated circuit composed of several transistors and capable of representing any primitive logic state, such as AND, OR, XOR, or NOT inversion conditions. Gates are also called digital, switching, or logic circuits.

Gate Array : It is a Part of the ASIC chip. A gate array represents a certain type of gate repeated all over a VLSI-type chip. This type of logic requires the use of masks to program the connections between the blocks of gates.

References

[1] Jamuna. S , and V . K. Agrawal , "VHDL Implementation of BIST Controller", in Proc. of int. Conf on Advances in Recent Technologies in Communication and Computing 2011, pp. 188-190.

[2] Prathyusha Navineini and S.K.Masthan , "Power Optimization of BIST circuit using Low Power LFSR", International Journal of Computer Trends and Technology-volume 2 Issue 2-2011, pp-5-8.

[3] Rekha Devi, Jaget Singh , and Mandeep Singh , "VHDL Implementation of GCD Processor with Built in Self Test Feature", International Journal of Computer Applications (0975 – 8887) Volume 25– No.2, July 2011, pp-50-54.

[4] G. Purdy, A carry-free algorithm for finding the greatest common divisor of two integers, Computers and Math. with Applications, 9 (2), pp. 311-316, 1983.

[5] Douglas Densmore, "Built-In-Self Test (BIST) Implementations An overview of design tradeoffs", M.S. thesis, EECS 579 – Digital Systems Testing, University of Michigan, USA,2001.

[6] Zhu Hongyu and LI Huiyun , “A BIST Scheme to Test Static Parameters of ADCs”, IEEE Symposium on Electrical & Electronics Engineering (EEESYM) ,2012.

[7] Pavel Emeliyanenko ,” High-performance Polynomial GCD Computations on Graphics Processors” , 978-1-61284-383-4/11/$26.00 ©2011 IEEE , pp-215-224.