IJSHRE

ISSN NO: 2347-4890
International Journal of software & Hardware Research in Engineering
Volume 1 Issue 2, October 2013

BIST ARCHIETECTURE AND IMPLEMENTATION OF A HIGH SPEED 32-BIT SIGNED/UNSIGNED PIPELINED MULTIPLIER

Deepak Parashar

M.Tech VLSI

Suresh Gyan Vihar University, Jaipur

Email ID-dpklive_4u@yahoo.com

ABSTRACT

In this paper a High speed 32-bit signed/unsigned pipelined multiplier is implemented. A BIST test pattern generator for signed/unsigned pipelined multiplier is proposed. A booth multiplier algorithm is also used. Linear feedback shift registers are used to generate the test pattern. A comparator is used to compare the output response and the expected response. For the circuit to work correctly the output response must be the same as the expected response. Xilinx ISE is used to synthesize the circuit and ModelSim is used for simulation purpose.

1. INTRODUCTION

BIST is a technique to test a circuit through built-in hardware functions. In BIST a part of the circuit is used to test the circuit itself. BIST helps in the testing and verification of the circuit without the requirement of any hardware verification language. Every device is required to go through testing to ensure proper working. The devices that are produced nowadays consist of heterogeneous components like processors, memories etc. So it's not easy to test them. Also the design of the devices is mostly core based. Therefore the internal structures cannot be accessed. The technology used to manufacture devices is deep submicron technology. If there is a fault in such a device, it's very complicated to detect.

BIST provides solution to all these problems

Well if we see both type of numbers whether its signed or unsigned numbers are generally used in and important in Digital Signal Processing (DSPs) and various special type of computers and circuits. However the multipliers are generally made for either signed or unsigned multiplier type. For our best knowledge the multiplier which are made with operation for performing signed/unsigned operation together are 1)earlier a 2-bit signed/unsigned booth multiplier ,2) newly a 32-bit signed/unsigned multiplier pipelined multiplier. But there are certain problems in there such as the low speed and also the need of hardware verification and testing of circuit in which we use a special technique BIST technique for it.

To solve this problem BIST architecture is used in the circuit. BIST architecture randomly generates a test pattern and applies it to the input of the device. Also the BIST architecture produces its own output, known as expected output, of those inputs. Both these outputs are compared with each other. To ensure proper working the outputs must match each other. If the outputs are same then a message is displayed regarding correct working of the floating point multiplier unit. In case the outputs are different, a message is displayed showing that the floating point multiplier is not working properly.

2. 32-BIT SIGNED/UNSIGNED PIPELINED MULTIPLIER

As we have discussed mostly the multiplier generally deals with either the signed operation or the unsigned operation. So we have developed a multiplier that can perform operation with both signed and unsigned

ISSN NO: 2347-4890
International Journal of software & Hardware Research in Engineering
Volume 1 Issue 2, October 2013

data simultaneously with correctness with higher inputs so we uses 32-bit signed/unsigned pipelined multiplier. The multiplier represents to us two different modes of operations, namely 32-bit 2's complement number operand and unsigned 32-bit binary number operand. This thing for its speeding up process instead of array and serial/parallel multiplier uses Booth multiplier technique as it is simple and fast in performance so it decreases the time complexity.

3) BOOTH MULTIPLIER

Booth Multiplier is a powerful algorithm for signed-number multiplication, which treats both positive and negative numbers uniformly and equally in the same manner.
For the standardised add-shift operation, each multiplier bit generates one multiple of the multiplicand which to be added to the partial product. If the multiplier is very large, then a large number of multiplicands needed have to be added.

Booth algorithm is a method that helps to reduce the number of multiplicand multiples which increases speed. For a given range of numbers to be represented, a higher representation radix leads to fewer digits. Since a k-bit binary number can easily be interpreted as K/2-digit off radix-4 number, a K/3-digit radix-8 number, and so on, it can easily deal with more than one bit of the multiplier in each round cycle by using high radix multiplication.
The Booth's multiplication algorithm is a multiplication algorithm that multiplies two signed or unsigned binary numbers in two's complement numerological and notation. The algorithm was invented by a scientist Andrew Donald Booth back in 1950.

The Booth's algorithm mostly examines adjacent pairs of bits of the various N-bit multiplier Y in signed 2's complement representation, including an implicit bit below at the least significant bit, $y_{-1} = 0$. For each bit y_i, there for i running from 0 to N-1, the different bits y_i and y_{i-1} are considered. Where these two bits are equal, the product accumulator P is left unchanged. This is 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 last value of P is then the signed product.

The representation of the multiplicand and which of product are not specified; typically, they are both also in two's complement representation, like the multiplier, but in any number system that supports addition and subtraction will together work as well. As it is written here, the order of the steps is not determined. mainly, 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 which is to be the right between steps; low bits can be easily shifted out, and normal additions and subtractions can then be done just on the highest N bits of P. Actually, There are many variations and optimizations on these details.

One of the important thing which is used by the Booth Multiplier is that it uses the radix-4 format to reduce the number of steps which are involved in the input and output operations .
The radix-4 technique method generally divides the given input data which is to be multiplied into the group of 3 digits and thus forms an arbitrary binary digit for that particular group of data and for that purpose there is a special table which is been provided and which is to be used. For example a group of "010" is been replaced by simply +1.

4) BIST ARCHIETECTURE

The general BIST architecture consists of a test pattern generator, a device under test (DUT) and an optical response analyzer (ORA). The test pattern generator generates several patterns of inputs to test the device. The patterns generated by the test pattern generator are provided to the device under test (DUT) and its response to the test patterns is analyzed by the optical response analyzer (ORA). The optical response analyzer compares the output of the DUT with the expected output and provides the results accordingly.

The test patterns generator generates patterns using a gate-level representation of the design netlist. These patterns are stored in tester memory and scanned into the circuit using parallel scan chains. The chip input/output plays a major role in determining the number of scan chains. Other than this there are also some factors like the tester channels and on-chip routing congestion which can have an impact on the number of scan chains.

ISSN NO: 2347-4890
International Journal of software & Hardware Research in Engineering
Volume 1 Issue 2, October 2013

BIST adds an on-chip pattern generator and an on-chip result compressor to the circuit. The former helps in feeding the scan chains and latter helps in compressing the scanned out responses. All those responses are compressed into a final signature.

When a test pattern is applied to a circuit, the pattern data is scanned out firstly. After scanning, clock cycles are applied to it. The clock cycles can be one or more than one. Finally the resultant data is scanned out.

4.1) TEST PATTERN GENERATOR

The test pattern used to test the circuit can be generated by a number of techniques. Basically the test pattern generator techniques helps in defining and testing the different hardware and software circuit implementation and testing. Some of the main techniques used to generate test pattern are

a) Exhaustive testing
b) Pseudorandom testing
c) Pseudoexhaustive testing

When a test pattern is applied to a circuit, the pattern data is scanned out firstly. After scanning, clock cycles are applied to it. The clock cycles can be one or more than one. Finally the resultant data is scanned out.

TEST PATTERN GENERATOR

↓

DEVICE UNDER TEST (DUT)

↓

OPTICAL RESPONSE ANALYZER (ORA)

Fig 1 BIST Architecture

This diagram provides us the basic concept about the BIST architecture and the working steps.

5) SYNTHESIS AND SIMULATION RESULTS

The BIST architecture is implemented on 32-bit high speed pipelined multiplier using XILINX ISE . It has been mapped and routed on XILINX FPGA circuit of family Spartan 3E XC3S500E. ModelSim is used for simulation purpose. The implementation results are shown below. Fig.2 shows the RTL schematic of the BIST architecture. Fig.3 shows the RTL schematic of the circuit without implementing BIST architecture.

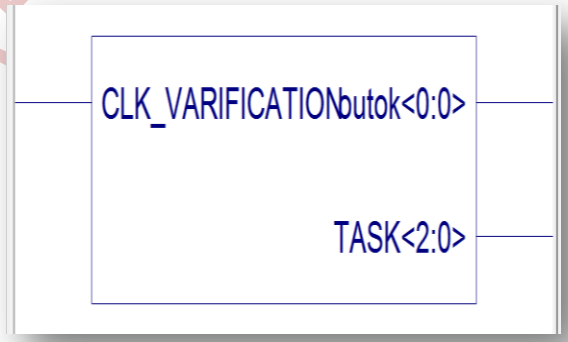

Fig.2 RTL schematic of BIST architecture

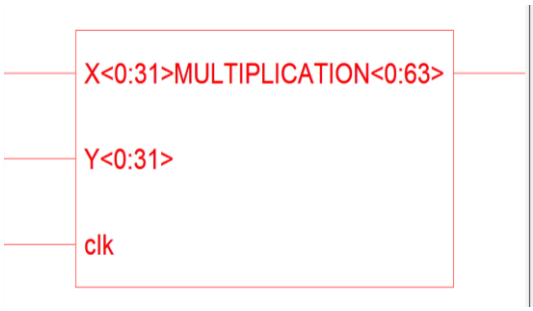

Fig.3 RTL of 32-bit pipelined multiplier

ISSN NO: 2347-4890
International Journal of software & Hardware Research in Engineering
Volume 1 Issue 2, October 2013

Fig.4 shows the simulation waveform of signed/unsigned pipelined multiplier unit without implementing the BIST architecture and Fig.5 shows the simulation waveform of the complete unit with BIST architecture

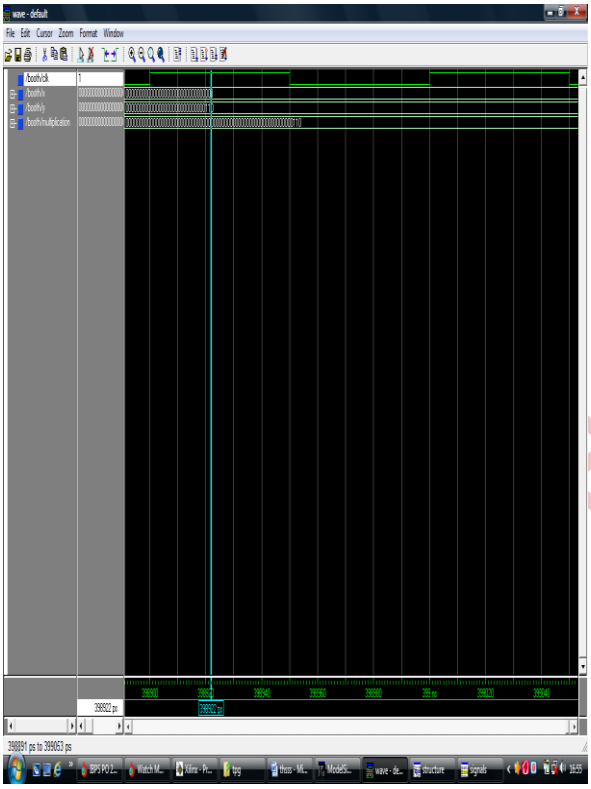

Fig.4 Simulation result of signed/unsigned pipelined multiplier

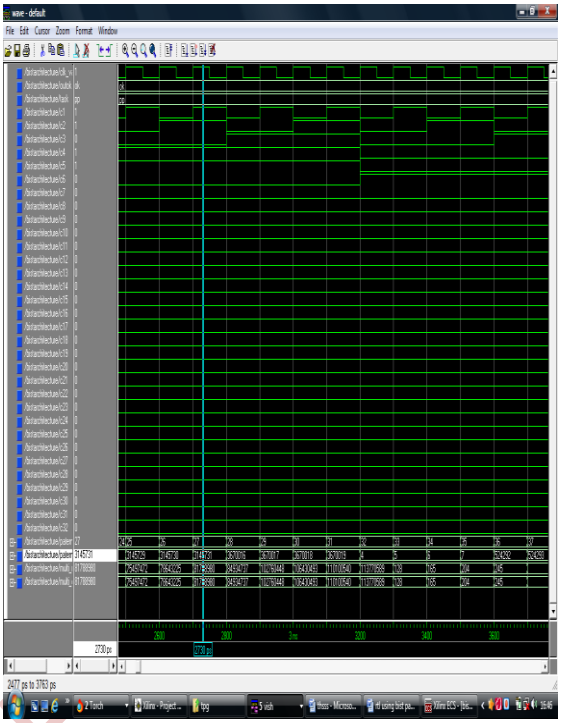

Fig.5 Simulation result of BIST architecture

6) CONCLUSION

A high speed 32-bit signed – unsigned pipelined multiplier has been successfully implemented using VHDL. The code is synthesized using Xilinx ISE tool and Model Sim is used for simulation purpose. The output of the signed – unsigned pipelined multiplier has been verified with the help of BIST architecture. The output of the device under test (DUT) is the same as the expected output and the signed – unsigned pipelined multiplier unit is working correctly. This signed/unsigned multiplier has been very successfully optimized in various areas such as speed, power consumption, latency etc. by including several different types of adders and various regular compressors.

REFERENCES:

[1]. John L Hennesy & David A. Patterson "Computer Architecture A Quantitative Approach" Second edition; A Harcourt Publishers International Company

[2]. C. S.Wallace, "A suggestion for fast multipliers," *IEEE Trans. Electron. Comput.*, no. EC-13, pp. 14–17, Feb. 1964.

[3]. M. R. Santoro, G. Bewick, and M. A. Horowitz, "Rounding algorithms for IEEE multipliers," in *Proc. 9th Symp. Computer Arithmetic*, 1989, pp. 176–183.

[4]. Kuang S. R., Wang J. P., guo C. Y.,"Moodified Booth Multipliers with a regular partial product Array",IEEE Transactions on circuits and circuitsII"vol. 56 issue 5,pp 404-408,may 2009.

[5]. Naofumi Takagi, Hiroto Yasuura, and Shuzo Yajima. High-speed VLSI multiplication algorithm with a redundant binary addition tree. *IEEE Transactions on Computers, C-34(9),* Sept 1985.

[6]. "IEEE Standard for Binary Floating-point Arithmetic", ANSUIEEE Std 754-1985,New York, The Institute of Electrical and Electronics Engineers, Inc., August 12, 1985.

[7]. Morris Mano,"Digital Design" Third edition; PHI 2000

[8.] J.F. Wakerly, *Digital Design: Principles and Practices*, Third Edition, Prentice-Hall, 2000.

[9] J. Bhasker, *A VHDL Primer*, Third Edition, Pearson, 1999.

[10]. M. Morris Mano,"Computer System Architecture", Third edition; PHI, 1993.

[11]. John. P. Hayes, "Computer Architecture and Organization", McGraw Hill, 1998.

[12].Wang, G" A Signed/unsigned binary multiplier", The thirty-eight Assilomar conference on signals, systems and computers,2004,vol 1.pp:513-516,nov7-10 ,2004