DESIGN AND IMPLEMENTATION OF MODIFIED CLOCK GENERATION

Bibi Hajira[1], Nayana M[2], Siva Yellampalli[3], Mujthaba[4]

[1]*Department of VLSI & Embedded Systems, UTL Technologies, Bangalore*
[2]*Department of VLSI & Embedded Systems, UTL Technologies, Bangalore*
[3]*Department of VLSI & Embedded Systems, UTL Technologies, Bangalore*
[4]*Jain College of Engineering, Belgaum*

Abstract

Performing delay test needs the automatic test equipment (ATE) which is used to provide the high-speed clocks, which is then used to generate at-speed test. ATE has some limitations such as it has limited number of clock pins, and is limited in supplying maximum clock frequencies. Expensive ATE has number of pins that works in high frequencies but that will be very expensive to go with to avoid that in this project at speed pulses is generated by a logic that is given to STUMP based LBIST

Keywords *— ATE, LBIST*

----------***----------

1. INTRODUCTION

In olden days' people used to generate at speed clock pulses using ATE. For at speed testing the number of external pins requirement will increases but ATE has limited number of pins. Some ATE are available with more number of external pins but they become expensive for generating clock pulses in Giga Hertz.

As the technology is scaling down the frequency to detect delay defects has to be high in frequency. Initially ATE used for detecting the delay defects but when the frequency of operation is in GHz the ATE becomes very complicated and costly. So clock generation on chip was the best idea to come up with. The on chip clock generation without increasing the complexity will generate the speed clock by using the logic. These generated clocks is then given to LBIST.

2. LITERATURE SURVEY

In paper [11], the ATPG was done manually for insertion of patterns for on chip clock generation they used Delay test ATPG. Since there is an issue with sequential nature of the ATPG, delay test in sequential ATPG is far complex as compare to stuck-at ATPG. In stuck-at ATPG, the use of more than one clock cycle during ATPG is known. Additional clock cycles are required to initialize sequential elements for example non-scan cells and RAM, they are also known as "clock sequential" or "RAM sequential" ATPG, respectively. Due to this addition of non scan cells it adds still more complexity to the problem of delay fault ATPG. Since access to RAMs is very essential for delay testing as critical functional timing paths mostly involves functional paths to and from RAMs.

In paper [12], they had come up with programmable opcg which is compatible with any design. In this no cut points known as pseudo primary inputs is required to find, which is very complicated if it has to be apply it for complex circuit. The use of cut-points and pseudo-PIs requires lot of efforts manually. Below are the procedure, the user has to do to identify the cut-points and utilise them in programmable test sequences:

• Cut-point is placed where an opcg control or clock comes in a picture.
• Make a relation between the cut-points and the pseudo-PI which is abbreviated as PPI, that is given in the test patterns as it acts as a stimulus point. The cut points which is given by the programmer are set in a single clock which work in a same clock domain, will be set together.
• Initialize the sequence which has to be programmed and start with the PLL.
• All legal clocking sequences must be given such that they are utilized when the PPI activity needs it and all the real events are necessary to make the OPCG programming correct for that particular sequence.
• Verify all the steps mentioned above correctly.

In paper [10], at speed testing is done using the logic which is built in chip. When test compression technique is applied within the chip then it affects also on the pattern count. This is an approach which has utilise the OPCG which actually enables high-speed testing, ``and is very compatible with test compression. It also enables at a time multiple domain in OPCG mode which reduces again the data volume and test.

3. PROPOSED ARCHITECTURE

The below figure 1 shows the on chip clock logic
The PLL will generate functional frequency clocks. PLL output and Trigger macros is then inserted on to the OPCG Domain. When the TESTMODE = 0 then the PLL bypass the OPCG logic.

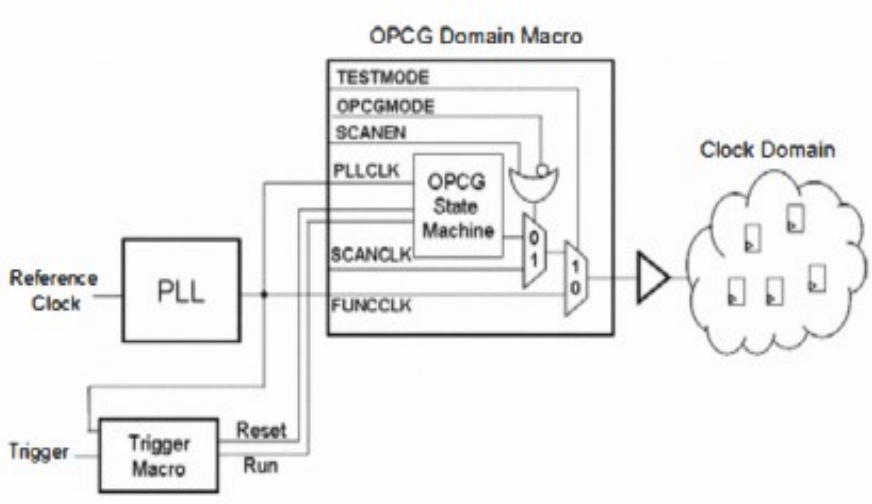

Fig 1: architecture of on chip clock generation

When test mode is given as 1 and the OPCGMODE is also equal to 1, then the OPCG state machine starts generating the high speed launch and capture pulses and when the scan shift operation is done, the SCANCLK is used. A Trigger Macro is inserted to remove glitch and to produce run and reset signals which is then applied to the OPCG macros. Once the Trigger signal is inserted, it Reset all the signals and then reload the On Product Clock Generation state machines to the programmed state. After few cycles Run signal is given from trigger macro and clock is generated.

The Fig 1 shows an on product clock generation logic, the logic is implemented in this paper with a name clock chopper. This clock chopper is a logic for OPCG. The figure 2 shows clock chopper.

3.1 Clock Chopper

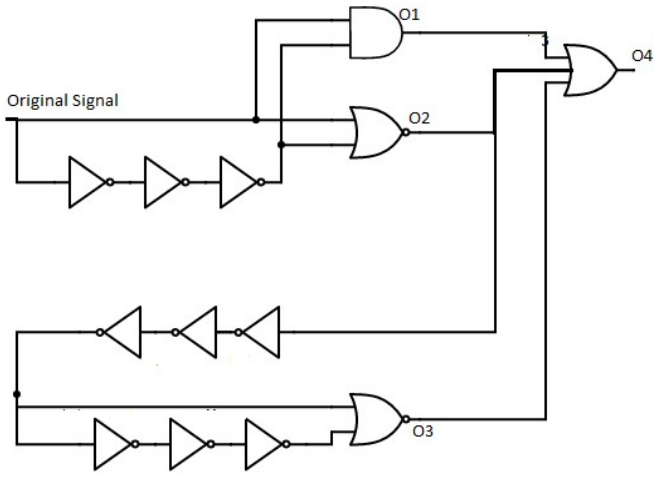

Fig 2: Clock Chopper

Clock chopper is a OPCG logic which is used to change the clock frequency of the original signal [9]. When the original signal is passed through the circuit, there will be delay in the circuit due to chains of inverters, this will help in chopping the original signal.

These invereters will make a delays for the original clocks and adding up all these clocks with different delays will make a new clock generation which is used for functional clock. This pulses reference clock is given by PLL which is working on 20 MHz frequency and it produces 500MHz frequency which is then given to clock chopper.

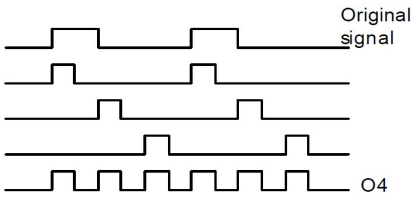

Fig 3: Waveform of clock chopper

4. STUMPS BASED LBIST

The figure 4 shows the architecture of STUMPS based LBIST. This LFSR is used for generating pseudo random code generator. Which is passed to scan chain here in this multiplier is used for programming and with that scan chains are generated

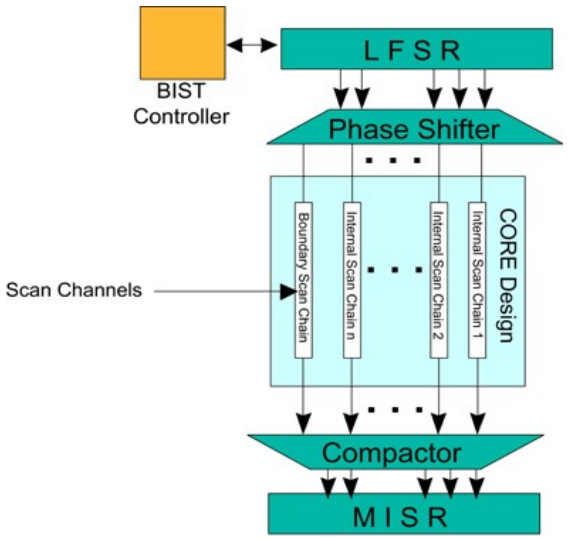

Fig 4: Architecture of LBIST

In this it works when test mode is 0 then the clock will be given from PLL and when test mode is 1 then the clock will be from clock chopper.

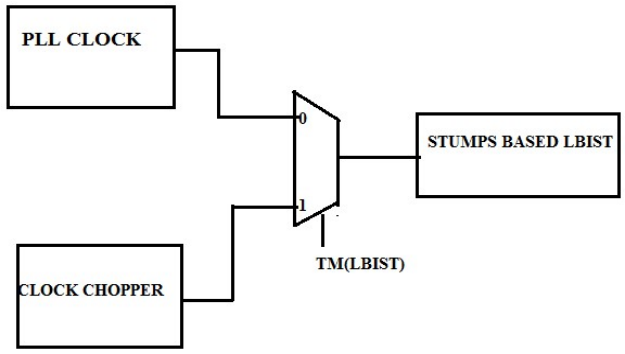

Fig 5: Architecure of the design implemented

5. RESULTS

Fig 6 shows the waveform of PLL output when it is in bypass mode, the PLL output frequency is same as reference clock.

Figure 7 shows PLL in power down mode. This will generate no signal for the PLL output. The PLL will be in OFF mode.

when Reset = 1 there will be some free running cycles of about 10MHz frequency will run as PLL output as shown in figure 8.

The figure 9 shows the PLL output of high frequency. The input given is 20 MHz and the output is of 400MHz. this value can be still more increased depending on the divider values. This mode is called locked.

Fig 6: Bypass mode

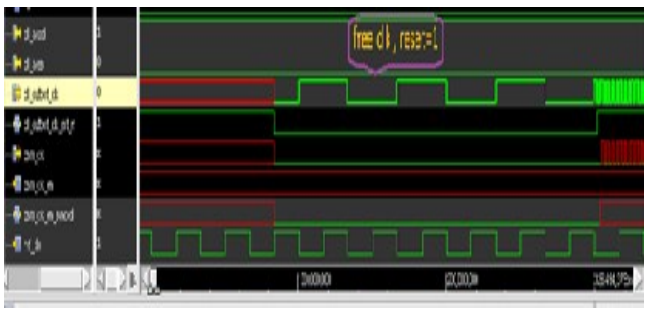

Fig 7: Power down mode.

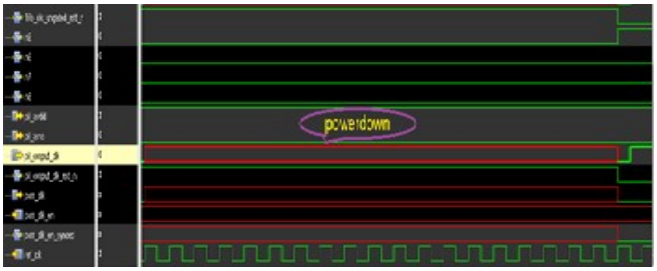

Fig 8: PLL in reset mode.

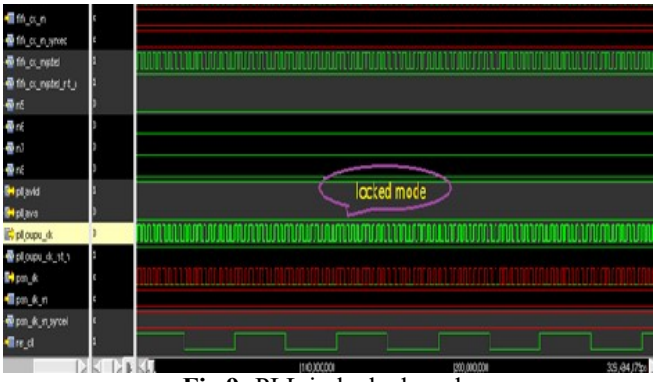

Fig 9: PLL in locked mode

When the LBIST uses PLL when the TM is 0 the PLL has to be in locked mode, and the LBIST takes clock from clock chopper when TM is 1.

Fig 10: STUMPS based LBIST

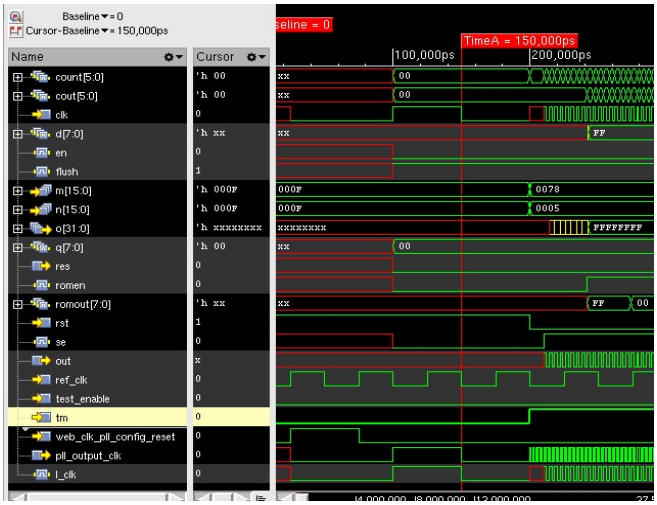

Fig 11: Proposed architecture waveform

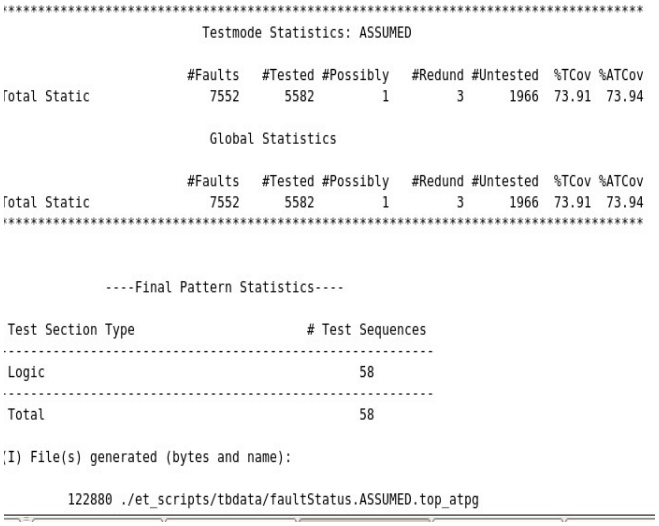

Fig 12: fault coverage of stumps based LBIST

6. CONCLUSION

The PLL is implemented with the four modes, here in this project, we need the locked mode for functional clock, which will be given to theclock chopper for generating the clock for LBIST when it is in testmode and PLL 20 MHz clock frequency will be taken as reference clock, which is used in PLL.

The clock chopper is implemented with a clock defined by the PLL in locked mode that is 500 MHZ. this locked mode PLL can generate different clock frequencies by changing the divider values and with that clock, got the output of clock chopper. The ouput clock of PLL and output clock of clock chopper is then given to Need to generate CCLK and the basic stumps is generated.

REFERENCES

[1] Matthias Beck, Olivier Barondeau, Xijiang Lin, Ron Press ,Martin Kaibel, and Frank Poehl ,"Logic Design for On-Chip Test Clock Generation–Implementation Details and Impact on Delay Test Quality," IEEE J. of Solid-State Circuits, No.3, pp.1530-1536,March 2005.

[2] Brion Keller, Anis Uzzaman, Bibo Li, and Tom Snethen," Using Programmable On-Product Clock Generation (OPCG) for Delay Test ," 16th IEEE Asian Test Symposium, pp.69-72,2007.

[3] B. Keller, K. Chakravadhanula, B. Foutz, V. Chickermane, R. Malneedi, T. Snethen and V.Iyengar, D. Lackey, G. Grise ," Low Cost At-Speed Testing using On-Product Clock Generation Compatible with Test Compression," international test conference ,pp 1-10,2010.

[4] Laung-Terng Wang, Cheng-Wen Wu, and Xiaoqing Wen, "VLSI Test Principles and Architectures: Design for Testability", Morgan Kaufmann Publishers

[5] Miron Abramovici, Melvin Breuer, Arthur Friedman, "Digital systems Testing and testable design", Jaico Publishers.

[6] B. Dervisoglu and G. Stong, "Design for Testability: Using Scanpath Techniques for Path-Delay Test and Measurement," in Proc. Int. Test Conf. (ITC'91), pp. 365–374,1991.

[7] K. Usami, M. Igarashi, F. Minami, T. Ishikawa, M.Kanazawa, M. Ichida, and K. Nogami, "Automated low-power technique exploiting multiple supplyvoltages

[8] applied to a media processor," IEEE J. of Solid-State Circuits, Vol.33, No.3, pp.463-472,March 1998.

[9] Delay Defect Characteristics and Testing Strategies,K.S. Kim, S. Mitra, and P.G Ryan, Design & Test of Computers, Vol. 20, No. 5, pp. 8–16,2010.

[10] Evaluating ATE Features in Terms of Test Escape Rates and Other Costs of Test Culprits, J.Gatej et al., Proc. Int'l Test Conf (ITC02), IEEE Press, 2002, pp.1111–1119

[11] Design For Test in Encounter RTL Compiler,"Inserting On-Product Clock Generation Logic", pp.368-415,2010.

[12] Test Article, "On Product Clock generation",pp.1-3,2005.

[13] Anis Uzzaman and Carl Barnhart," Delay test generation with on-product clock generation",pp.1-4,2007.