DESIGN-FOR-DELAY-TESTABILITY TECHNIQUES FOR HIGH-SPEED DIGITAL CIRCUITS

Herman Vermaak

Samenstelling van de promotiecommissie:

Voorzitter:	Prof. dr. ir. T. Mouthaan	UT
Secretaris:	Prof. dr. ir. T. Mouthaan	UT
Promotor:	Prof.dr.ir. T. Krol	UT
	Prof. G.D. Jordaan	SUT
Assistent- Promotor:	Dr.ir. H.G. Kerkhoff	UT
Leden:	Prof.dr. H. Wallinga	UT
	Prof.ir. A.J.M. van Tuijl	UT
	Prof. dr. ir. M.T.M. Segers	TUE

UT :	Universiteit Twente,
TUE:	Technishe Universiteit Eindhoven
SUT:	Sentrale Universiteit voor Techniek, Vrijstaat, Zuid-Afrika

Titel:	Design-for-Delay-Testability Techniques for High-Speed Digital Circuits
Auteur:	Herman Vermaak
ISBN:	90-365-2259-5

Deze onderzoek was gesteuen door de Nationale Onderzoek Stichting van Zuid Afrika (NRF)

DESIGN-FOR-DELAY-TESTABILITY TECHNIQUES FOR HIGH-SPEED DIGITAL CIRCUITS

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof. dr. W.H.M. Zijm,
volgens besluit van de College voor Promoties,
in het openbaar te verdedigen
op 7 december 2005 om 13.15 uur

door

Hermanus Jacobus Vermaak
Geboren op 20 Februari 1963
te Uitenhage, Zuid Afrika

Dit proefschrift is goedgekeurd door:

de promoteren
prof.dr.ir. T. Krol
prof. G.D. Jordaan

de assistent-promotor dr.ir. H.G. Kerkhoff

Aan mijn vrouw, Linda en mijn dochters, Lin-Mari en Mari-Louise

Contents

Chapter 1

Introduction to the Testing of High-Speed CMOS Digital Circuits

"*The key to developing engineering confidence is the rigorous identification of the cause for ALL failures encountered for ALL phases of testing and providing tests to detect ALL these faults* ", Dr. Joseph F. Shea the Deputy Director of Manned Space Flight.

1.1 Introduction

The main purpose of the test process, as it is applied to the manufacturing of semiconductor products, is to provide a measure of the quality and/or reliability of a finished semiconductor product, whether that product is a stand-alone die or a packaged part. The purpose of design-for-testability (DfT) is to place the "hardware hooks" on the die to enable the ability to conduct the quality-reliability measurement. If done correctly, the DfT will:

- enable the quality goals to be met with a high degree of confidence (high test coverage) during testing
- allow the coverage measurement to be done efficiently and economically to meet the cost-of-test goals
- and enable some form of test-vector automation, such as automatic test pattern generation (ATPG) to help meet the aggressive time-to-market or time-to-volume manufacturing requirements.

Currently, the semiconductor industry is changing on many fronts:

- smaller geometric features on silicon

- the reduction of supply voltage levels
- the use of new physical processing techniques such as copper interconnect, low-K dielectrics, and silicon-on-insulator
- the use of complex mega cells and macro cells referred to as cores
- the use, and reuse, of existing simple and complex cores
- the integration of huge amounts of memory and the integration of large amounts of logic on system-on-a-chip (SoC) devices.

One of the most important changes currently in the SoC industry is the rapid increase in device speed. SoCs have surpassed 3.5 GHz. It is expected that the speed of such circuits will continue to increase for future technology generations. In the last two decades, however, the clock frequencies of SoCs have improved at an average rate of 30% per year.

These industry changes make quality and reliability requirements more important than ever, but at the same time are creating aggressive challenges in the ability to measure the quality level, and to measure the quality level economically [1].

In the past, the test process has been characterized as an “over-the-wall” event that occurred when the design team completed the design and passed it to a dedicated team of test and/or verification professionals. This test process was largely the translation and reuse of the functional simulation verification vectors to the target tester platform, and the manual creation of new verification vectors to get more coverage. The “post design” time to provide a set of vectors to meet high quality expectations was measured in months and even years.

The structural tests nowadays require that test structures be designed into the device to assist with the test process, quality measurement, and the vector generation. These changes are being driven by market pressures to:

- provide high-quality parts
- meet time-to-market or time-to-volume manufacturing windows
- meet product cost goals by meeting cost-of-test goals

1.2 Why Testing Digital Integrated Circuits?

For more than 30 years, MOS device technologies have been improving at a dramatic rate. A large part of the success of the MOS transistor is due to the fact that it can be scaled to increasingly smaller dimensions, which results in higher performance. The ability to improve performance, like speed, consistently while decreasing power consumption has made CMOS the dominant technology for integrated circuits. The scaling of the CMOS transistor has been the primary factor of driving improvements in e.g. microprocessor performance. Transistor delay times have decreased by more than 30% per technology generation resulting in a doubling of microprocessor speed every two years [2].

In the manufacturing test area, there are significant challenges rapidly approaching and this is exactly the objective of the test industry to keep up with these challenges. Whatever Moore's Law dictates must get to manufacturing and shipping with minimal testing costs [1]. Test costs are highly dependent on the application of the product. In space or safety-critical applications it is factors higher than all production and design costs. It is predicted that testing a transistor will cost the same in 2007 as design and manufacturing together [3].

If one designs a product, fabricate and test it, and it fails the test, then there must be a cause for the failure. Either (1) the test was wrong, or (2) the fabrication process was faulty, or (3) the design was incorrect, or (4) the specifications were incorrect. The role of testing is to detect whether something went wrong and the role of diagnosis is to determine exactly what went wrong, and where the process needs to be altered [4]. Therefore, correctness and effectiveness of testing is most important for quality products. The process of providing good chips to the customer is shown in Figure 1.1.

If the test procedure is correct and the product fails, then we suspect the fabrication process, the design, or the specifications to be faulty. A well-thought -out test strategy is crucial to economical realization of products. The trade-offs in testing are quality and costs. Quality indicates the probability that a customer receives a faulty device. Normally, a high quality product has a high cost. It will be impossible for an engineer to design a quality product without a profound understanding of the physical principles underlying the processes of manufacturing and test.

A long-held axiom of the semiconductor industry states that the number of transistors on a chip doubles every 18 months. Known as Moore's Law, it has driven the industry until now and should continue for the next 15 years. The result is an increase in performance, increase in complexity, as well as a reduction in the cost of the chips.

The performance of chips is often indicated in terms of speed and power dissipation. The first is a combined result of circuit design and decreased minimum dimensions. Power dissipation is influenced by circuit and system design and technology. It is known that new technology is reducing the power supply specifications (dielectrics), decreased leakage currents and reducing the parasitics that are present.

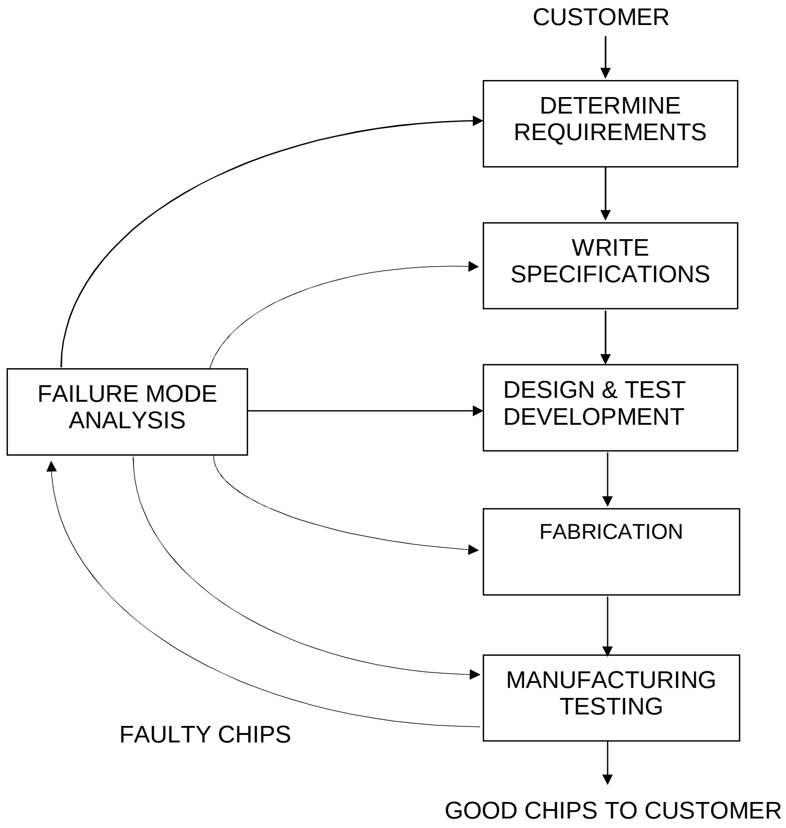


***Figure 1.1**: A flowchart showing the process of supplying the customer with good chips [1].*

1.3 Developments in High-Speed Digital Circuits

Tremendous progress in scaling of process technologies down to the deep-submicron (DSM) domain has paved the way for a significant increase in the level of integration and performance of modern SoCs. The integration of complex SoCs is now a reality. Achieving acceptable reliability levels for modern SoC chips is a critical issue, making testability a significant factor that could limit scaling trends if not addressed adequately. This is especially significant since defect sizes have not scaled in a manner commensurate with shrinking geometries [1]. Process and Application induced variations can have a significant influence on a chip's failure to meet specified performance [5].

1.3.1 Delay-Fault Testing of High-Speed Digital Circuits

Delay testing is essential as chip geometries continue to shrink. The manufacturing procedures and process technologies used to fabricate below 100 nm SoCs can unintentionally introduce hard-to-detect defects including wire and via opens (complete and resistive), bridges between wires, and parametric failures which can slow the transition of a signal within a design. This problem was relatively benign in slower technologies, but as device speeds have increased the likelihood of seeing device failures due to delay-faults is growing exponentially.

Delay tests can be generated automatically using ATPG tools [6]. Engineers have been using transition-fault delay tests extensively to detect delay defects and path delay-tests to characterize device performance [6]. Compared to using functional vectors, the use of ATPG reduces the test development effort significantly. In addition, designers can employ scan-based delay tests using scan chains running at lower speeds, as long as there is a method to exercise the system clock at-speed (typically provided by on-product clock-generation logic). This enables the use of a lower cost tester.

These tests must keep up with the ever-increasing speed and bandwidth of high-performance SoCs. In many cases, one is pushing the performance limits of the silicon processes. Delay-fault testing is therefore necessary to ensure product quality and help us sort the production chips into the rated speed categories during test. This technique is commonly known as speed binning.

1.3.2 Delay-Fault Testing of Embedded Cores

The question is how do the existing test methods adapt to core-based SoCs? This adaption is an open issue, since, in addition to the standard test-quality problems, core-based SoCs present new challenges, in particular in terms of test development for providers and test-access mechanisms (TAM) for integrators. The basic issue is high-speed access to the core. One way to deliver at-speed tests to embedded cores is to exploit the SoC architecture-specific information for providing sources and sinks during testing and to reuse on-chip functional interconnect for TAMs.

Regardless of their potential benefits in the long term, unless implemented automatically using a reliable test-tool flow, these architecture-specific DfT methodologies do not provide reusability, flexibility and inter-operability. To address these problems in a well-structured modular way, the practical state-of-the-art SoC test approaches, such as the recently adopted IEEE P1500 standard, use separate test and functional communication architectures [7-8].

The adoption of a core-based design methodology includes the consideration of at least three different major test problems (and many minor ones). One test problem comes from the ability to create extremely large designs in a short period of time by reusing existing design elements: the system-on-a-chip design methodology. Another problem involves the reuse of several cores within one design where they each may have existing but different DfT strategies. The third problem is having the ability to access and test complex cores if they are embedded. Currently a lot of research [9-13] is focussing on adaption of standard test wrappers and cells to accommodate test functionality to be able to test these embedded cores for delay-faults.

The chips made by reusing existing large amounts of logic may include separated (existing) test strategies, which leads to the test methodology of testing the individual cores in isolation. The virtual test socket is being able to test each core individually in a self-contained manner [14]. The types of test strategies chosen for these designs may adversely impact other chip design budgets such as power, area, timing, packaging, and package pins. The trade-offs involving the type of test strategy chosen may also result in the adoption of the type of core and whether or not the core has pre-existing DfT logic.

1.4 Trends in Delay Testing using Automatic Test Equipment

The most significant changes in Automatic Test Equipment (ATE) requirements are the result of stitching together, on a single piece of silicon, circuits (cores) that historically have existed only as individual devices. As a result, we are seeing the need to quickly generate and debug high-speed digital patterns obtained from several sources. Pattern sources include a variety of simulation environments, vectors supplied as part of the cores, ATPG tools and sometimes, vectors ported from other ATE platforms. The ATE can be modelled or represented as a number of channels with memory depth (a channel basically representing the memory behind a package pin), a number of clock generators, and a number of power supplies. It can be noted that the big ATE companies have developed SoC testers, which specifically deal with SoC designs (e.g. Advantest & Agilent).

The basic capabilities of ATE must keep up with trends towards lower voltages, larger pin count, extended accuracy, larger memory depth, and increase in speed of digital signals. The International Technology Roadmap for Semiconductors (ITRS) illustrates this trend in detail [2].

1.4.1 Can ATE keep At-Speed?

Modern ATE, while providing the required timing and waveform flexibility at tester-base data rates, generally has fallen short if base rates are exceeded. ATE data rates can be increased by constructing multiple circuit-under-test (CUT) waveforms in a single tester cycle. Historically, this is accomplished either by adding to the number of data bits available per tester cycle or by channel multiplexing with the corresponding loss of available tester channels. These techniques require that test engineers and ATPG software fully comprehend the extremely complicated programming model needed to successfully create and debug the resulting timing and vectors.

1.4.2 Limitation of ATE regarding Complexity

The problems faced with the present ATE-based testing include exploding test data, high cost of equipment and inability to verify chips at speed. It must be remembered that ATE has to test future devices using existing hardware. It is predicted that tester accuracy will improve in the near future but at a much lower pace than the increase in clock speeds. ATE has not been able to achieve effective fault coverage with access basically only to input and output pins. The main issues seen by test groups are as follows:

- Potential yield losses as the cycle time of manufactured devices become comparable to the timing accuracy (speed problem) of ATE, as shown in Figure 1.2. Such yield losses are unacceptable and can further increase the testing costs.
- Increasing cost of capital equipment, driven by increasing pinning and higher frequencies.
- Requirements of test to support yield learning and defect detection, and failure analysis.

Most of the technology problems causing major yield losses and cost increase are related to the slower growth of ATE speeds versus the ever improving device speed as shown in Figure 1.2 [2].

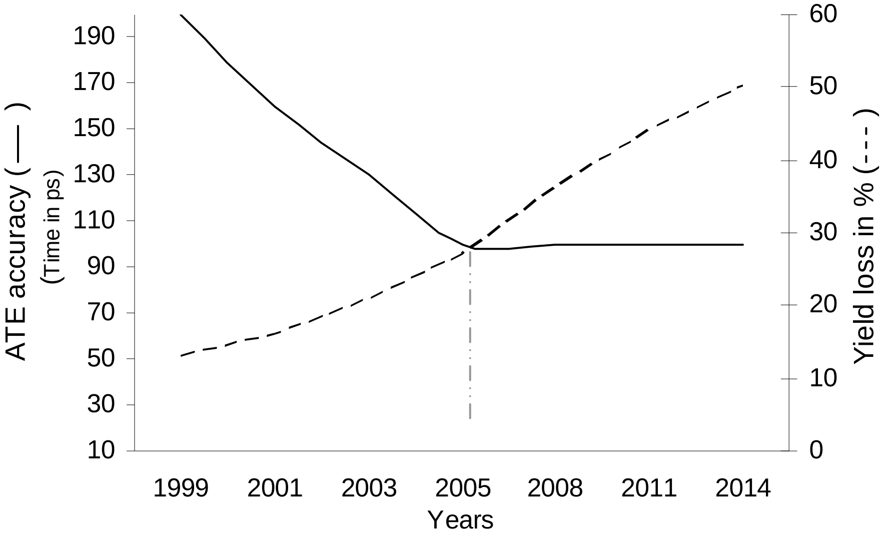


***Figure 1.2**: Trends in ATE accuracy and yield loss [2].*

The recent increase in Built-In Self-Test (BIST) hardware, one form of DfT, in SoC design is expected to cause a change in test equipment requirements and a reduction of cost in the future. Relaxing the requirements of

ATE by introducing an internal test infrastructure is therefore a very important issue. This trade-off is referred to as test-resource management.

1.4.3 ATE and Cost Considerations

The main component of the “cost-of-test” is the time a chip spends “in-socket” on the ATE. There is a minimum cost based on the minimum time that it takes an automated handler to insert and remove a chip from the tester socket (production throughput). Aggressive handler time is generally in the 1-3 second range. Cost reductions to bring tester time below the handler threshold require testing multiple SoCs in parallel with the same tester, using multi-site stations [14].

If the test programs applied on the tester are very complex or contain an excessive amount of vector data, then a vector reload may be required. A reload adds a large amount of time to the overall test process, which increases the “in-socket” time, and therefore, the cost of test. The ways to minimize the test program complexity and data size are to apply simplifying techniques, such as single-edge-set testing, and to apply vector compression to the vectors sets [14]. A simpler test program also reduces the complexity of the ATE that is required. A less complex ATE requirement means that a less expensive tester can be used.

Although the speed capability of the SoC ATE continues to rise, the cost is a major concern. The cost of ATE, in general, is a function of speed, pin-count and vector memory. In addition to the high clock rate, complex SoC devices also have high pin-count and long vector sequences. One may, therefore, have to compromise speed in favour of other characteristics in selecting an ATE. By the time the 1 GHz ATE finds its way into the large-scale chip manufacturing, one will begin to see even higher device speeds. This is to be expected since it is the device speed that drives the development of the ATE. One has to realize that today’s ATE uses today’s most advanced devices. However, they must be able to test tomorrow’s devices. Experience also shows that above 20 MHz, test system noise can become a source of inaccuracy [15].

So what are the trends that have such a major influence on the test cost of digital circuits?

- *Rising chip clock rates*: Microprocessors represent the leading edge in the Silicon technology trend. The exponentially rising clock rate indicates several changes in testing over the next few years.

  - *At-Speed Testing*: It has been established that stuck-at-fault tests are more effective when applied at the circuit's rated clock speed, rather than at a lower speed. Stuck-at-fault testing covers many circuit signals assuming that a faulty signal may be permanently stuck-at logic 0 or 1. For a reliable high-speed test, the ATE must operate as fast as, or faster than, the CUT.

- *Increasing transistor density*: Transistor feature sizes on a SoC, e.g. processors, reduce roughly by 10.5% per year, resulting in a transistor density increase of roughly 22 % every year. An almost equal amount of increase is provided by wafer and chip size increases and circuit design and process innovations. The doubling of transistors on an integrated circuit every 18 to 24 months was stated by Moore's Law [2], as indicated earlier. Although many have predicted its end, it continues to hold. This leads that testing difficulty increases as the transistor density increases. This occurs because the embedded cores become increasingly difficult to access.

1.4.4 Problems with regard to Load Boards

The ATE can be modelled or represented as a number of channels with memory depth, a number of clock generators and a number of power supplies. These resources are applied to the chip through a load board that houses the chip socket.

A load board is a circuit board designed to serve as an 'interface' circuit between the ATE and the CUT. Load boards contain the necessary components to:

- set up the CUT for correct testing by the ATE
- route the test and response signals between the CUT and the ATE
- provide additional test capabilities that the ATE may not be able to provide itself.

There are also load boards designed for the purpose of testing or calibrating the ATE itself. The load board must also reflect the performance of the device it is intended to test. If it is a high-frequency design, then board routing must observe the rules of transmission lines for length, width and impedance matching. The load board should not set the frequency limit of testing, the tester's internal limitation should apply.

The key point to be made here is that the tester only has access to the chip by its pins and the load board may be the limiting factor for the level of test that can be applied.

An ideal load board introduces no distortion, noise, delays, nor errors to the testing process of the CUT. This means that an ideal load board is one that does not seem to exist at all, i.e. as if the CUT is directly connected to the ATE. To implement a load board as close as possible to this ideal is the challenge to every engineer who designs load boards.

Load board design takes into consideration many factors, one of which is reflections. Reflection of the signal occurs internally in the load board between the signals and different discontinuities. Even in the simple example of a single impedance load board, multiple reflections are present due to connector and socket discontinuities and may decrease the accuracy of the measurements.

Another consideration with load board design is power supply routing. It is good practice to assign a separate power plane for every supply voltage needed by the CUT, even if two or more supplies will be tied up to the same nominal voltage. This has two advantages: 1) noise immunity between power supplies and 2) the ability to assign each power supply to a different voltage later on. Adding sense lines as close as possible to the CUT to each power plane for output monitoring would also be helpful. Putting decoupling capacitors between each power supply plane and the ground plane will help to reduce power supply noise [16].

Signal routing is another consideration in the design of a load board. There are two types of CUT signals, and guidelines for handling them in a load board differ. The first type, the low-speed digital signal, does not require much as far as load board design goes. However the lengths of their traces should be the same. The second type of CUT signal is the high-performance signal. This signal type requires high-performance instrumentation for measurement, because the speed and accuracy of these signals cannot be handled by the ATE. To avoid disturbances, the length of the cable connecting the CUT to the load board must be kept as short as possible. Also, parallel runs of mixed signals should be avoided as well, to avoid noise coupling [16].

Innovative, high-performance ATE timing-system architectures are now possible thanks to technology that was unavailable until very recently. If properly used, high-density submicron CMOS and SiGe high-performance mixed-signal capabilities support an unprecedented combination of speed, flexibility, and accuracy. Unfortunately even with these new architectures the ATE still has to test tomorrow’s devices with today’s technology and at an extremely high cost. This is even made worse by the fact that the costs of high-speed processors are non-linear.

1.5 Problem Definition

The rapid increase in clock frequencies of digital circuits due to sub-micron technology has created an increased concern about cost-effective detection of delay-faults. The major industrial players in the manufacturing and testing of digital high-speed circuits acknowledge the importance of detection of delay-faults [2, 15].

Several approaches have been proposed for improving the delay-fault detection of high-speed digital circuits. The identification of the critical paths to minimize the paths that have to be tested for delay-faults has been proposed, e.g. in reference [17]. Also the aspect of insertion of test points to improve the controllability and observability of a circuit to improve the detection of delay faults has been suggested [18]. Many other methods for improving delay-testability have also been presented [19-21]. One approach is introducing a Design-for–Delay-Testability (DfDT) structure to help with the detection of faults and especially delay-faults.

Today incredible performance of new complete systems that are integrated into one chip is being offered. What it means, firstly, is that one will see one or multiple of these SoC stacked with memory into a System-in-Package (SiP) [22].

The testing of cores is receiving a lot of interest nowadays due to the popularity of SoCs. The integrated circuit (IC) design community is divided into two groups: the core providers and the core users. The providers supply the correctly functioning cores to the user who is only concerned with the design integration, manufacturing and testing of his own system. The cores are provided as soft, firm or hard-core, not manufactured and therefore not tested [7]. This leaves the testing of the core to the user after manufacturing of the system. The user would therefore require assistance from the provider by delivering pre-defined tests with the core.

In this environment the IEEE 1500 standard for Embedded Core Test (SECT) was developed. It intends to facilitate the testing of embedded cores as separate entities [23]. The IEEE 1500 does not cover the core's internal test methods or test integration and optimisation. The IEEE 1500 is targeted at testing "black-box" third party cores. The implementation details of the cores are hidden from the core user and it is mandatory that the core providers deliver delay tests information with these cores to ensure that testing of these cores can be carried out with sufficient fault coverage.

The purpose of this research work is to develop and present new structures that can be used as DfDT structures but also to provide information on the susceptibility of these structures to variations and solutions to problems arising from using these structures. The thesis also addresses the detection of delay-faults in embedded cores and provide enhancements to standard test structures to be able to accommodate these test methodologies.

1.6 Outline of the Thesis

The structure of this thesis is as follows:

- Chapter 2: This part provides an introduction to the concept of DfDT. The different test methodologies used for delay-fault testing are discussed and proposed structures and techniques for delay-fault testing are treated.

- Chapter 3: A new type of DfDT structure and associated BIST architecture for detecting delay-faults in digital high-performance circuits are discussed. It circumvents the requirement of an expensive high-speed tester. This chapter deals with the susceptibility of the proposed structure to process- and application-induced variations. Due to the critical timing necessary when detecting small delay faults it is crucial to know what to expect from these variations and subsequently reduce their influence. This chapter provides solutions to many of these questions regarding the implementation of the DfDT structure.

- Chapter 4: One way to overcome the critical test setup used in many test techniques is to make use of the digital oscillation test method. In this part it will be shown that by introducing a varying supply voltage, in combination with the oscillation test method, it is possible to detect delay faults as well as stuck-at faults in the particular sensitized critical path in the digital circuit under test. This technique is also proven by an experiment by applying a varying supply voltage to a discrete circuit. The results of this experiment are also discussed in this chapter.

- Chapter 5: Continued advances in the manufacturing processes of integrated circuits provide designers with the ability to create more complex and denser architectures and increased functionality on a single chip. The increased usage of embedded cores in SoC necessitates a core-based test strategy in which cores are also tested separately or in parallel. The IEEE 1500 SECT aim is to improve the testing of core-based SoCs. This chapter deals with the enhancement of the test wrapper and wrapper cells to provide a structure to be able to test embedded cores for delay faults. This approach allows delay-fault testing of cores by using the digital oscillation test method and the help of the enhanced elements while staying compliant to the IEEE 1500 standard.

- Chapter 6: The last chapter provides conclusions and a summary of the work in this thesis. It also highlights the contributions of the thesis and gives recommendations for future work.

Delay testing has been a topic of extensive research both in industry and in academia for more than a decade. As a result, several delay-fault models and numerous testing methodologies have been proposed. This work is contributing towards the drive from test engineers and industry to provide techniques that address some of the current problems in delay-fault testing.

1.7 References

[1] K.-T. Cheng, S. Dey, M. Rodgers, and K. Roy, “Test Challenges for Deep Sub-Micron Technologies”, *Design Automation Conference (DAC*), ACM, June 2000, pp. 142-149.

[2] G.E. Moore, “Progress in Digital Integrated Electronics,” *Proceedings of the International Electron Device Meeting*, December 1975, pp. 11- 13.

[3] http://www.itrs.net Semiconductor Industry Association. *International Technology Roadmap for Semiconductors:* 2003 edition, Austin, SEMATECH, 2003.

[4] S. Thompson, P. Packan, and M. Bohr, “MOS Scaling: Transistor Challenges for the 21st Century,” *Intel Technology Journal*, Third Quarter 1998.

[5] H.J. Vermaak and H.G. Kerkhoff, “Reducing the Susceptibility of Design-for-Delay-Testability Structures to Process- and Application-Induced Variations“, *Proceedings of IEEE European Test Workshop (ETW),* May 2001, Stockholm, Sweden, pp. 35-41.

[6] M.L Bushnell and V.D. Agrawal, “*Essentials of Electronic Testing for Digital, Memory & Mixed-Signal VLSI Circuits*”, ISBN 0-7923-799-1-8, Kluwer Publishers, Massachusetts, 2001.

[7] E.J Marinissen, R. Kapur, M. Lousberg, T. McLaurin, M. Ricchetti, and Y. Zorian, ”On IEEE P1500’s Standard for Embedded Core Test”, *Journal of Electronic Testing: Theory and Applications (JETTA)*, vol. 18, no. 4-5, October 2002, pp. 365–383.

[8] IEEE 1500 SECT Task Forces. IEEE 1500 Web Site: http://grouper.ieee.org/groups/1500/.

[9] E.J. Marinissen, S.K. Goel, and M. Lousberg, “Wrapper Design for Embedded Core Test”, *Proceedings IEEE International Test Conference (ITC*), 2000, Atlantic City, USA, pp. 911–920.

[10] E. Larsson and Z. Peng, “An Integrated System-on-Chip Test Framework”, *Proceedings of Design, Automation, and Test in Europe Conference (DATE*), 2001, Munich, Germany, pp. 138–144.

[11] E.J Marinissen, Y. Zorian, R. Kapur, T. Taylor, and L. Whetsel, ”Towards a Standard for Embedded Core Test: An Example”, *Proceedings of International Test Conference (ITC),* 1999, Atlantic City, USA, pp. 616-627.

[12] S. Goel and E.J. Marinissen, “Effective and Efficient Test Architecture Design for SoCs”, *Proceedings IEEE International Test Conference (ITC*), 2002, Baltimore, USA, pp. 529–538.

[13] V. Iyengar, K. Chakrabarty, and E.J. Marinissen, “Co-Optimization of Test Wrapper and Test Access Architecture for Embedded Cores,” *Journal of Electronic Testing: Theory and Applications (Jetta)*, vol. 18, no. 2, April 2002, pp. 213–230.

[14] A.L. Crouch, “Design-*for-Test for Digital IC’s and Embedded Core Systems*”, ISBN 0-13-084827-1, Prentice Hall, USA, 2000.

[15] D. Herrell, “Power to the Package”, *IEEE Spectrum*, vol. 36, no. 7, July 1999, pp. 46 – 53.

[16] http://www.semiconfareast.com/

[17] K.T. Lee and J.A. Abraham,” Critical Path Identification and Delay Tests of Dynamic Circuits”, *Proceedings of International Test Conference (ITC)*, 1999, pp. 421-430.

[18] N. Mukherjee, T.J. Chakraborty, S. Bhawmik, "A BIST Scheme for the Detection of Path-Delay Faults*," Proceedings of International Test Conference (ITC)*, 1998, pp. 422-431.

[19] M.Favalli, P. Olivo, M. Damiani and B. Ricco, ”Novel Design for Testability Schemes for CMOS IC’s*”, IEEE Journal of Solid-State Circuits, Vol. 25, No. 5, October 1990.*

[20] V. D. Agrawal and T. J. Chakraborty, "High-Performance Circuit Testing with Slow-Speed Testers", *Proceedings of International Test Conference (ITC)*, 1995, pp. 302-310.

[21] M. Shashaani and M. Sachdev, "A DFT Technique for High Performance Circuit Testing," *Proceedings of International Test Conference (ITC)*, 1999, pp. 276-285.

[22] R.C. Aitken, ”Designs are Changing – Can Test Keep up*?*”, *Agilent Design-for-Testability Library of Presentation*, ITC 2000.

[23] Y. Zorian, “Test Requirements for Embedded Core-Based Systems and IEEE P1500,” *Proceedings of International Test Conference (ITC)*, Washington, USA, November, 1997, pp. 191–199.

Chapter 2

Testing for Delay-Faults in High-Speed Digital Circuits

While devices in excess of 3 GHz clock rates are currently being designed [1], even higher frequencies will be possible in the near future. It must also be mentioned that the original quest for higher frequencies is slowing down and an example of this is that INTEL has stopped a design, not because of speed, but because of the relating excessive power dissipation. Test inputs for these devices are developed either to exercise the implemented functions at full speed, or to detect modelled faults (e.g., stuck-at faults) in the structure [2-3]. In the past decade, much work has been reported on deriving tests based on delay-fault models (e.g. [4-11]). Irrespective of how tests are obtained for this category of faults, they must be applied to the circuit under test at the rated speed to guarantee the correct operation.

2.1 Testing High-Speed Digital Circuits

A major roadblock in testing high-speed digital circuits are the high pin count probes and test sockets. Research and development is urgently required to enable cost-effective solutions with reduced parasitic impedance. High-speed serial interface speed and port count trends will continue to drive high-speed analog source/capture and jitter analysis instrument capability for characterization. There is therefore a great need for the development of effective DfT techniques for testing these circuits. It is important that device interface circuitry must not degrade equipment bandwidth and accuracy, or introduce noise, especially for high-frequency differential I/O and analog circuits.

DfT techniques like scan, and to a lesser extend BIST have been the mainstay for over 20 years and new test methods for control and observation are needed. Tests will need to be developed utilizing the design hierarchy. Analog DfT and BIST techniques must mature to simplify test interface, Logic BIST techniques must evolve to support new fault models, failure analysis, and deterministic tests. Electronic Design Automation (EDA) tools for DfT insertion must support DfT selection with considerations for functionality, coverage, cost, circuit performance and ATPG performance.

2.1.1 Test Methodologies for High-Speed Digital Circuits

Automatic test equipment in advanced production environments of today can test SoC devices at 1 GHz clock rate [12]. However, current semiconductor technology allows devices that can operate at much higher clock rates. ATE speeds, as determined by the pin electronics, transmission lines, and test fixtures, have always lagged behind device speeds. As mentioned in chapter 1, the speed capability of the SoC ATE continues to rise, but the cost is a major concern. One may, therefore, have to compromise speed in favour of other characteristics in selecting an ATE as stated earlier.

BIST has been often cited as a solution to the high-speed test problem [13]. In this method, hardware is added to the device for test-data generation and response analysis. A high-speed clock may be applied from an external ATE which would not require a large vector memory and high pin-count capability. Because of the response compaction used in conventional BIST, the method does not provide a high diagnostic or debug capability. It must, however be mentioned that diagnostic BIST approaches nowadays are providing some enhanced capabilities for use. In addition, BIST adds substantial hardware to the device. For these reasons, conventional ATE-based testing is still preferred in many applications. BIST can have the advantage of staying within the chip, thereby circumventing the limited bandwidth of I/O pins, and all loading issues from the board and ATE.

Several approaches addressing issues that are to be considered when developing a SoC test solution have been proposed. One method that is very often utilized is that of Test-Resource Partitioning (TRP). Zorian proposed a scheduling technique within TRP that minimizes the test time while considering test-power consumption [14].The technique assumes that each testable unit has its pre-determined and dedicated BIST resource. The partitioning of test sets into on-chip test (BIST) and off-chip test using an ATE was also investigated [15].

What is popular TRP then? Each testable unit is tested by applying at least one set of test vectors (test stimuli) where each test set is stored or generated at a *test source* and the test response is stored or analyzed at a *test sink*. Each testable unit may have its dedicated test source and test sink or may share either test source or test sink with other testable units or may share both test source and test sink. The *test resources* (test source and test sink) can be placed on-chip or off-chip where an ATE is a typical off-chip test resource and a Linear-Feedback Shift-Register (LFSR) is a typical on-chip test source. In general, any combination of test resources is possible [14]. The test stimuli at a test source can be generated off-chip and analyzed on-chip. It is also possible to store the test stimuli in an on-chip memory making the test source on-chip and storing the test response in a test sink placed off-chip. A testable unit can also be tested by several test sets. One test set can be for stuck-at fault testing, one for at-speed testing and one for functional testing. Furthermore, it is also possible to test a testable unit with one test set stored at an ATE, one test set generated by an LFSR, and one test set stored in a memory. The test solution is highly dependent on the partitioning of the test sets for the testable units. A limited test set at an ATE often results in the same fault coverage as a larger test set generated by an LFSR [15]. Within TRP the test problem is seen as a task that can be divided between DfT and ATE. The problem is the limited bandwidth of integrated circuit pins. Or stated differently, solve high-speed issues on-chip and apply ATE for low-speed problems.

It is possible to test a *scan* circuit at slow speed and yet verify its high-speed performance. Nowadays it is even possible to do high-speed testing making use of scan. This is accomplished by generating skewed clocks to allow propagation times between flip-flops that are smaller than the test-clock period. Scan flip-flops can be modified for delay testing and require more hardware than normal scan flip-flops. As a result of the speed limitations of inexpensive ATE, many high-speed digital components (SoC chips, printed circuit boards and multi-chip modules) are tested at clock rates that are slower than the full-speed specification.

High-speed test methods can be classified into two categories, being, direct and indirect methods [16]. In a direct method, the delay of the functional logic is directly observed by propagating a test signal through it. In indirect methods, one assumes that timing characteristics of all logic gates on a SoC chip are correlated since these logic gates were produced by the same fabrication steps. Thus, it is not necessary to test each gate once the fabrication process has been characterized. The two different methods will now be explained in more detail.

2.1.2 Indirect Test Methods

One of the major categories of defects that make a SoC circuit to fail is spot defects. These spot defects can seriously damage transistors or interconnects within the circuit. These defects are normally detected by slow-speed tests, such as the ones designed to cover stuck-at faults. In addition, some parametric variations in the fabrication process may not seriously damage any device, but will affect all or most devices on the chip. Such parametric variations can reduce the operating speed of the chip and may not be detected by slow-speed tests. In this situation, a simple test is to characterize the fabrication process. This is done by adding a separate test circuit either on every chip or on a few sites, called Process Control Modules (PCM), on the wafer.

The delay of a single logic gate or even that of a few gates which form a path in a high-speed test circuit may be too small to detect via a measurement. However, a long string of gates can be laid out on chips such that the total delay of the path is within the measurable range. A possible test circuit for this purpose is a ring oscillator. This circuit consists of a series of inverters where the last inverter feeds into the first inverter. If the number of inverters in the circular chain is odd, the circuit will oscillate. The frequency is determined by the delay of a single inverter and the number of inverters used [17].

The indirect test methods of testing high-speed devices also include correlation techniques. Although the correlation methods tend to be probabilistic in nature, they are still widely used in SoC testing. An example is that the correlation between a small-sized ring oscillator and a relatively big circuit may not be appropriate. Moreover, as technology is scaled, the supply voltage is also scaled down to reduce power consumption and to enhance reliability. As a consequence, there may not be sufficient possibilities to reduce the supply voltage for low-voltage testing. In [16] it is pointed out that the routing delays remain largely unchanged as the supply voltage is lowered. Hence, critical paths at a low voltage may not be the same as at a nominal power supply voltage [16].

It is of interest to observe the changing nature of delays, from transistor delays to interconnect delays. Pin-to-Pin (Pn-Pn) and Point-to-Point (Pt-Pt) delays are more terms for intrinsic and extrinsic delays respectively. A Pn-Pn delay is measured between a transition occurring at the input of a logic gate and a corresponding transition occurring at the output from that gate. On the other hand a Pt-Pt delay is measured between the output from a driving gate and the input of a load gate. As the geometries of structures on the silicon shrink and the number of gates in a device increase, interconnect delays assume a greater significance. Increasingly sophisticated algorithms are required to accurately represent the effects of the interconnect lines. From "pure RC" (lumped-load) calculations, through distributed RC calculations, to more complex RLC formula that also take input-switching threshold values into account.

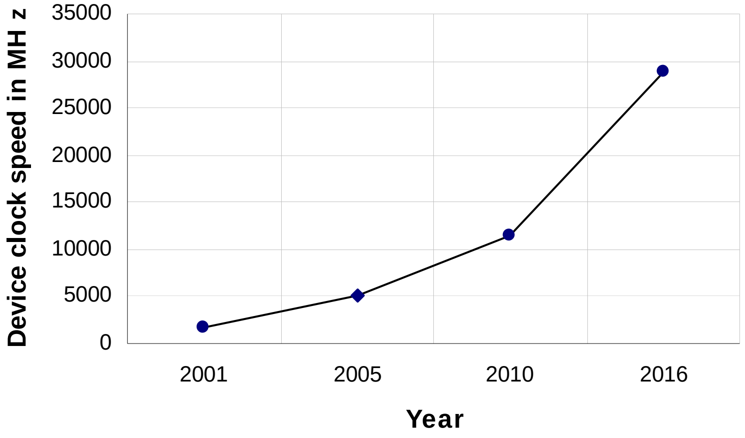


Figure 2.1: *Anticipated speed of devices according to ITRS [17].*

To mention one problem associated with the interconnect lines, is sending signals over longer distances. Since the resistivity of metal is a constant, increasing current density J means that the electric field E along a wire must increase. But meanwhile, the maximum voltage differences that are encountered scale as V, and for long wires (whose resistance we noted earlier scales upwards) the wires will be carrying significant fractions of our voltage drops [16]. But if the voltages themselves are dropping as V, this means that the length L of a wire must scale as V/E. The wire length as a multiple of L must hence decrease. In other words, long wires should be avoided in our devices! If the length of the wire does not scale down *faster* than transistor lengths, wire delays will dominate. However, dies are getting larger, and hence global interconnect lines are longer. Also capacitive loading due to fringing fields are relatively increasing.

In Figure 2.1 the anticipated speed, as given by the ITRS, for devices in the next 10 years are shown [18]. It is therefore obvious from Figure 2.1 that the above mentioned problems are not in the distance future any more.

2.1.3 Direct Test Methods

In general, a manufacturing defect can damage several logic gates on a chip. Many real situations are modelled neither by spot defects, nor by the distributed parametric variation. An ATE approach, where test signals are propagated through all or most logic gates, is frequently used for high-confidence testing. In spite of the problems stated in the Section 2.1, ATE are being used to test very high-speed devices.

A brief look into the past shows different methods by which the ATE was enhanced to be able to test high-speed devices. The following is a brief summary of the most important methods. Barton [19] has described the test of 500MHz GaAs and Silicon devices using special test fixtures. The length of cables that carry signals to and from the device under test has been minimized and proper terminations are carefully designed. The device output is picked up by an electro-optic sensor placed external to the ATE and close to the device.

For various reasons the timing accuracy is found to change with the length of tests. Tests are therefore repeatedly applied, with measurements conducted each time on different vectors. Calibration of the test system is critical and influences the reliability of the result. It is important to mention that this calibration is of utmost importance and very crucial.

Ackner and Barber [20] have shown the generation of clocks of frequencies that are up to four times higher than that produced by the tester. They multiplex tester pins through high-speed GaAs circuitry to produce these clock signals. The basic idea of pin multiplexing is illustrated in Figure 2.2. Because of the high accuracy that ATE offers in placing signal edges within a clock period, two properly shaped signals can be OR-ed to produce a signal that has twice the frequency. Notice that two ATE pins are used to produce one test signal. This extra hardware to produce the signal with twice the frequency of the ATE is added to form part of the ATE.

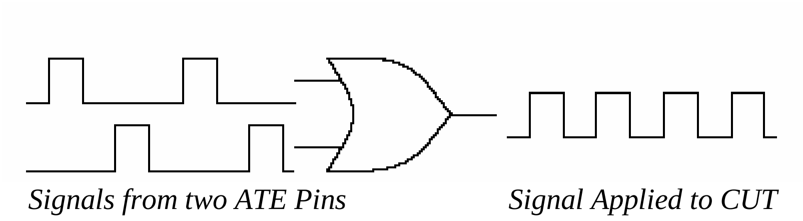


Figure 2.2: *An example of ATE pin multiplexing.*

To combat the tester-bandwidth problem, Ackner and Barber also use external GaAs comparators. However, they point out several limitations: (1) multiplexing reduces the number of available signal pins; (2) certain restrictions apply to signal waveforms; (3) restricted timing accuracy and (4) high costs.

Keezer [21] has described a system in which the high-speed clock is generated via multiplexing. The bit stream for a data pin of the device under test is loaded by the ATE into an external shift register and stored. The bit stream is then applied to the device pin at a much higher clock rate than is possible within the ATE. He also describes a per-pin external comparator for response analysis [22]. The restricted length of the data stream and resolution of output comparison are two of the limitations in such procedures. Despite limitations, the above techniques have been used to test devices at 500 MHz where the rated speed of the ATE was only 100 MHz [23].

Complex SoC devices contain many levels of flip-flops with intervening combinational logic. Timing tests require that flip-flops be clocked at the rated frequency while long vector sequences are applied to primary inputs. At the same time, primary outputs should be continuously strobed and checked for correctness. The required flexibility and accuracy from an ATE designed to operate at much lower frequencies, if at all possible, can be difficult and expensive to attain. In an alternative method, the test is conducted using a slow-speed test mode. If the circuit can be slowed down in a predictable manner, then timing tests can be applied by a slow-speed ATE. For testing high-bandwidth memories by slower testers, the use of on-chip test circuitry has been suggested in the literature [24]. In logic circuits, since the memory elements are embedded in combinational logic, the design of on-chip test circuitry can be quite different than that in memory chips.

As deep-submicron technologies emerge, however, many of these higher order delay effects are beginning to manifest themselves across technologies with scant regard for traditional boundaries.

2.1.4 Problems relating to DSM processes

Over the last decade, rapid strides have been made in electronic systems design and integration. From the several million transistor chips of the 1990s, we now have industry prototypes of 200 million transistor chips, operating at 4.5 GHz frequency. Soon, billion transistor chips will be available. This trend is expected to continue well into the next decade with the use of ultra high-performance, deep submicron devices operating at below 1 V supply voltages. In parallel, research in nano and quantum electronics will push digital circuit designs into new realms.

Scaling in the deep submicron regime has fundamentally altered the primary issues affecting SoC design. The emergence of DSM-related problems has resulted in a proliferation of design techniques that attempt to alleviate these newer effects in current flows. However, future design methodologies would be required to undergo a paradigm shift to comprehensively address these problems. A few of these anticipated problems are listed below:

- Interconnections have become a vital bottleneck in the speed performance of a circuit.
- As supply voltage levels go down, problems due to noise and leakage will become paramount.
- The design of global supply networks and clock networks will become more difficult with the increased chip complexities, and more vital to the correct functioning of the circuit.

The problems of design verification and testing of integrated SoCs will assume hitherto unseen proportions. This complicates manufacturing economics and the global competitiveness of the industry.

2.2 Testing for Delay Faults

The arrival of 90 nm designs has created new problems in clock skew and power delivery, while the latest nanometer technologies have demonstrated that defects are located predominantly in routing. Failure analysis conducted by major silicon manufacturers reveals that most of part failures are timing related, and delay is the biggest culprit. As a result, manufacturers are demanding dynamic defect models that are more sophisticated than the traditional models.

From the late 1990's there is an extremely high and still increasing logic-to-pin ratio on the chips [25]. This makes it increasingly harder to accurately observe signals at the device level, which is essential for testing.

DSM defects are typically physical deformations caused by missing or extra material and manifest in the form of shorts or opens. Depending on their impact, defects are typically classified as global and local defects. Global defects generally affect large areas or even entire wafers. On the other hand, local defects impact a smaller area on a die. Local defects are more difficult to detect/control, and require rigorous test practices for their detection. Local defects causing timing-only failures are becoming increasingly important in scaled DSM technologies. These defects often result in parametric failures and can cause delay faults that are difficult to detect.

2.2.1 Origin of Delay Faults

Delay faults are a major concern in nanometer technologies. Process variations are perhaps the most obvious problem in the behaviour of silicon structures.

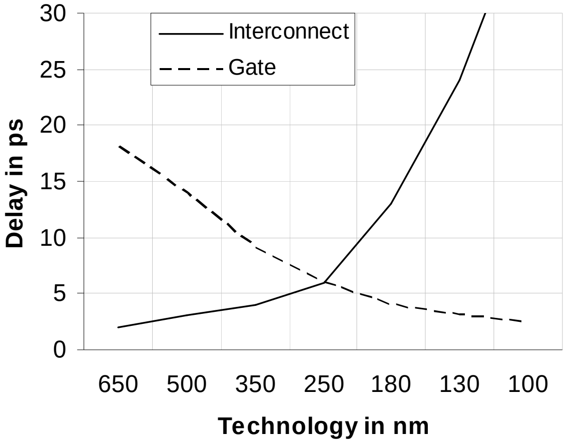


Figure 2.3*: Time delays caused by interconnect and logic gate at different feature sizes [26].*

Commonly observed failure mechanisms such as excessive leakage currents, gate-oxide faults, and resistive open defects in vias and interconnects can all lead to excessive switching delays on signal paths. Unfortunately, CMOS circuits display vastly different switching delays depending on the inputs. Also, the actual delay observed can also significantly depend on the capacitive coupling between internal nodes. Gates may vary according to width, length, threshold voltages, oxides and doping. Parametric delay faults may occur because of random particles on chips.

Interconnects, which are connecting devices, will decrease in size and increase in density due to the DSM technology. The scaling of interconnections poses several serious metallization problems including the increase in interconnection delay and the reliability of circuits. All estimates project that at gate lengths of about 0.13 µm and below, the interconnections will dominate the communication speed through the circuits (see Figure 2.3). Interconnects vary according to line width, metal thickness and interlayer dielectric thickness. The interconnect evolution is shown in Table 2.1.

Table 2.1*: The interconnect evolution [26].*

Year of first product shipment	*1997*	*1999*	*2001*	*2003*	*2006*	*2009*	*2012*
Technology generation	*0.25µm*	*018µm*	*0.15µm*	*0.13µm*	*0.1µm*	*0.07µm*	*0.05µm*
Number of metal level for logic	6	6-7	7	7	7-8	8-9	9
Total length (m/chip) for logic	820	1480	2160	2840	5140	10000	24000
Chip area (mm^2)	280	400	480	560	790	1120	1580
Metal density (m/mm^2)	2.9	3.7	4.5	5.0	6.5	8.9	15.2
Local clock frequency (MHz)	750	1250	1500	2100	3500	6000	10000

Copper with close to half the resistivity (1.7 µΩ.cm compared to 3.0 µΩ.cm for Al/0.5% Cu alloys) and with electro migration in the order of ten times better appears to be the most appropriate material for future interconnect. Copper is a clear choice at technology of 0.13 mm and smaller because it provides speed enhancement with no sacrifice of device reliability [26].

2.2.2 Delay-fault Models

As chip manufacturing processes become more sophisticated, the major failures shift from logic to delay origin. It should be noted that delay has an analogue nature. Traditional manufacturing tests include vectors for stuck-at faults with very high fault coverage (> 99.99 %). However, tests for stuck-at faults do not check explicitly for timing errors. These are often modelled as delay faults [27]. This class of faults causes the chip to malfunction at the required speed. Test technology has to be adjusted to cope with this class of faults.

In order to test such high-speed circuits, advanced test methods are necessary to verify their timing behaviour in an efficient and cost-effective manner. Already in 1974, Breuer [28] indicated the existence of delay faults. He stated that this would require a new fault model which was subsequently developed so that the faults could be detected. A fault model is a description of the behaviour and assumptions of how elements in a defective circuit behave. The goal of fault modelling is to model a high percentage of the physical defects that can occur in the device at the highest possible level of abstraction. The high level of abstraction reduces the number of individual defects that must be considered and lowers the complexity of the device description used in generating the test. The result is that test generation can occur earlier in the design cycle with less expensive computing resources.

Although more models exist, two models are popular, the gate-delay fault model and the path-delay fault model [2]. These models will now be reviewed in detail.

2.2.3 Gate-Delay Fault Model

The gate-delay fault model (also called the transition fault model) implies that only logic gates might be affected by delay faults. This model is a logical model for a defect that delays either a rising or falling transition on a specific line in the circuit.

In this model, all possible delays in a circuit are thought of to be concentrated in the logic gates. Hence, the interconnection lines are considered to be delay-fault free. Each gate can have a Slow-to-Rise and a Slow-to-Fall fault, and hence the total expected faults are equal to twice the gate count. The advantage of this model is that the number of test patterns is small and generation is rather conventional. If a delay-fault is sufficiently large, it behaves as a temporary stuck-at fault, and single stuck-at fault testing techniques can be applied.

The disadvantages, however, are that cumulative small delays do not add up to timing errors and two patterns are required for detection initialization and transition detection (propagation) and the minimum delay-fault size that can be detected is difficult to determine.

2.2.4 Path-Delay Fault Model

The path-delay fault model assumes that the distributed delay-faults are present in paths rather than individual logic gates. The path-delay fault model considers the cumulative delay of all logic gates and interconnections in a path, in other words 0-to-1 and l-to-0 transitions are propagated from an input to an output. The basic problem is the enormous amount of possible paths which tend to explode (exponentially). Some methods have been suggested to reduce the number of paths. Usually only critical paths are considered: if the cumulative delay exceeds e.g. the clock period in a sequential circuit in these paths, a timing error has occurred. The same holds for combinational circuits if it exceeds the maximum allowed propagation times.

The path-delay model overcomes a possible problem with the gate-delay model, in which other faster gates in the path may compensate for the delay of a faulty gate. In this case a faulty device is one that has exceeded the maximum allowed propagation time. The path-delay model can eliminate this problem by considering the contribution of the entire chain of possible delays. This delay model can be used with an aggressive design philosophy that pushes the process to the limit by recognizing that all gate delays are almost never simultaneously at maximum levels. This technique makes it possible to specify a greater clock speed by statistically determining the highest level of delays that are likely to occur in the real world. A problem with this model is that the number of possible paths grows exponentially with the number of nets.

2.2.5 Influence of Cross-Talk on Delay-Faults

A very disturbing and problematic effect is the change in delays of neighbouring interconnects lines under influence of cross-talk. Unfortunately, CMOS circuits in addition display vastly different switching delays depending on the choice of inputs.

For complex SoC designs, the incremental effects of cross-talk can exert a profound impact on signal integrity- and on timing itself. Depending on the nature of the cross-talk noise waveform, cross-talk can alter a victim signal waveform so that the signal reaches thresholds later than it normally would. The result is effectively a delay in the victim signal. Because cross-talk can alter the delay of a victim signal by more than 100 percent, the effect can be fatal in critical paths. For example, if the victim signal is in a critical maximum-delay path, the extra cross-talk-induced delay can result in a setup failure because the signal arrives too late at a latch or a flip-flop [29].

Also, the actual delay observed can also significantly depend on the capacitive coupling between internal nodes. In practice, delay due to coupling is strongly correlated to the switching direction of signals and is a strong function of input patterns [30]. In DSM circuits, cross-talk has a more significant effect and may result in functional or delay faults. It is unlikely for functional faults to appear in static CMOS circuits due to the high noise margin of gates. Signal propagation, however, can be delayed due to cross-talk resulting in increased delay.

A major limitation of many techniques that attempt to generate delay tests for a circuit considering cross-talk effects is that the effect of cross-talk is analyzed for only one targeted victim line. However, in reality, the worst-case delay of a circuit may be exhibited when cross-talk effects are distributed along a sensitized path. Hence, a delay test-generation methodology should consider the cross-talk effect on the delay of all signal lines along a sensitized path, rather than a single victim line. This is a complicated problem for static CMOS circuits because the delay of a victim line is dependent upon the transition of lines physically adjacent to the victim line.

To account accurately for cross-talk effects, special techniques must be used to deal with the complex cross-talk waveforms, complex timing and logic relationships between aggressor and victim signals, and the large amount of parasitic data that exists in SoC designs. Still, cross-talk analysis cannot occur in isolation. Just as cross-talk affects timing of signals, timing is of course a primary factor in the nature of the cross-talk noise waveforms caused by tight coupling at DSM densities [31]. Due to this mutual interdependence of cross-talk and timing, the correct approach for timing verification is a method that accounts for both of these effects. The most promising approach is noise-aware timing analysis, which can alternate between cross-talk noise analysis and timing analysis, feeding the results of each analysis to each other. Because cross-talk and signal-integrity issues wield such dramatic impact on timing, noise-aware timing analysis is emerging as a critical element in SoC verification. As process technologies continue to evolve, this new breed of analysis will be increasingly vital for achieving timing closure, meeting design performance goals and realizing time-to-market objectives for SoC designs [31].

2.2.6 The Two-Pattern Test Method

As a delay-fault is inherent part of the timing characteristics of a circuit one should therefore examine the signal transitions of a circuit to be able to detect delay faults. To be able to propagate a transition it is necessary to work with vector-pairs.

Delay faults, unlike stuck-at faults, require a set of two test patterns (a vector-pair) to excite and propagate a fault effect. Two test vectors which are sufficiently independent of each other have to be applied at consecutive clock cycles in order to activate a signal transition at the inputs of a circuit.

The detection of a path-delay fault requires path activation and fault-effect propagation conditions in a circuit, and therefore, is much harder to be detected by random test vectors. Conventional test-pattern generators for stuck-at faults do not serve those purposes as there is a correlation between the successively generated vectors.

A test for a delay-fault requires the application of two test patterns to create the necessary transitions on the sensitized critical path. A path is said to be sensitized if the effect of a fault is propagated along the path from the fault site to an observable output. The first pattern, called the initialization pattern, initializes the logic to a known state. The second pattern, called the launch pattern, launches a transition at the circuit line and propagates the fault effect to primary output(s) and/or scan flip-flop(s). It activates the targeted fault, causing a transition to propagate along the path under test. The first pattern is applied, and the transients are given time to settle. The second pattern is then applied, and the outputs are sampled after the cycle time. Consequently, several solutions for two-pattern generators for detecting delay faults have been presented in the literature. The effectiveness of such a test pattern generator can be determined either by the transition coverage [32] or by the number of two-pattern tests that actually appear at the inputs of a circuit.

2.2.7 Test-Patterns for Delay-Fault Detection

Stuck-at ATPGs have provided more challenges than fault simulation. ATPG algorithms work with many heuristics to efficiently determine tests for modelled failures. The notable algorithms have been the D-Algorithm, Path-Oriented Decision Making, and Fan-out-Oriented Test Generation. While technology has changed significantly, most changes have not affected ATPG solutions in a dramatic way. The scaling of process technology, however, has led to a significant change [2]. The calculation of timing delays, historically a function of delays occurring at the gate, is now dominated by delays occurring in the interconnections between the gates.

The main difference between TPG for stuck-at and delay faults is that delay testing requires two input vectors (a test pair) which are referred to as the initialization vector and the propagation vector. The first vector initializes the lines in the circuit to the desired initial value. The propagation vector is then applied to launch the desired transitions on the inputs and propagate them along the circuit under test. The response of the circuit is captured at normal circuit speed after the application of the propagation vector.

The first step of the test generation system consists of generating N+l random test patterns representing N test pairs. Next, a quality function characterizing the ability of a test pair to activate and detect delay faults is calculated for each test pair and compared with a threshold value. From the result of the comparison, test pairs are classified as positive or negative test pairs. A positive (negative) test pair is a test pair with a high (low) probability to detect delay faults. A relevant feature is a combination of the values of some bits which allows to predict with a good confidence level if any given test pair is positive or negative. A directed random test-pattern generation is then carried out. Details on the test generation process are given in the third subsection. The last step of the process is to simulate the generated test pairs so as to define delay-fault coverage and evaluate the CPU time required for the complete test generation.

A feature missing in stuck-at detection is that delay faults are analogue and have a property called delay size. Hence also some care in fault coverage of delay-faults in general must be made. However, gross delay-faults are easier to detect.

2.2.8 High-Speed ATE

The evolution of devices to be tested, interfaces jumping in speed in one generation from several hundred megahertz to gigahertz has increased the need for faster and more accurate high-speed testing. This leaves semiconductor manufacturers with the challenge of balancing cost versus technology innovation. Meanwhile, in an uncertain environment created by the convergence of technologies, manufacturers find it increasingly difficult to predict what must be tested in the future.

As the industry make transitions to deep sub micron / nanometer SoCs, with geometry sizes of 90 nm or below, the fault model changes. Different forms of new failures occur, such as unexpected timing behaviour, cross-talk and parametric failures, clock skew and synchronization issues, or parametric failures of high-speed I/Os. As a result, semiconductor manufacturers are beginning to encounter schedule slip, recalls and other problems that can be traced back to the proliferation of new failure mechanisms of these sub micron process technologies. In many situations, at-speed functional and parametric testing of nanometer devices is the only solution. Ideally, ATE gives manufacturers sufficient flexibility to adapt test strategies during design and manufacturing, allowing them to scale between different speed and memory ranges to support different test methodologies. Yet, the rigid configurations of traditional ATE solutions, with different digital cards for different speed ranges, fail because they lack needed flexibility.

2.2.9 What Drives the Cost of High-Speed ATE

One of the most compelling arguments put forth to stimulate efforts to reduce the cost of test is illustrated in the now famous Semiconductor Industry Association (SIA) Test Capital trend graph shown in Figure 2.4 [17]. The curve indicates that the cost of making a transistor will soon fall below the cost of testing that transistor if nothing is done to change the cost trend of testing devices.

Basically there are three, more or less interrelated characteristics that represent the difference between expensive and low-cost test equipment. These characteristics are:

- Speed – the rate at which vectors can be applied. In general, the cost of increasing the vector application rate goes up exponentially with the frequency
- Precision – both voltage and time. As with vector rates, an increase in the precision in both time and voltage typically will cause an exponential increase in the cost of the equipment

- Flexibility – this is a more abstract concept, but in this context it is meant to represent such things as the number of independent complex waveform generators available, the number of tester pins available, etc.

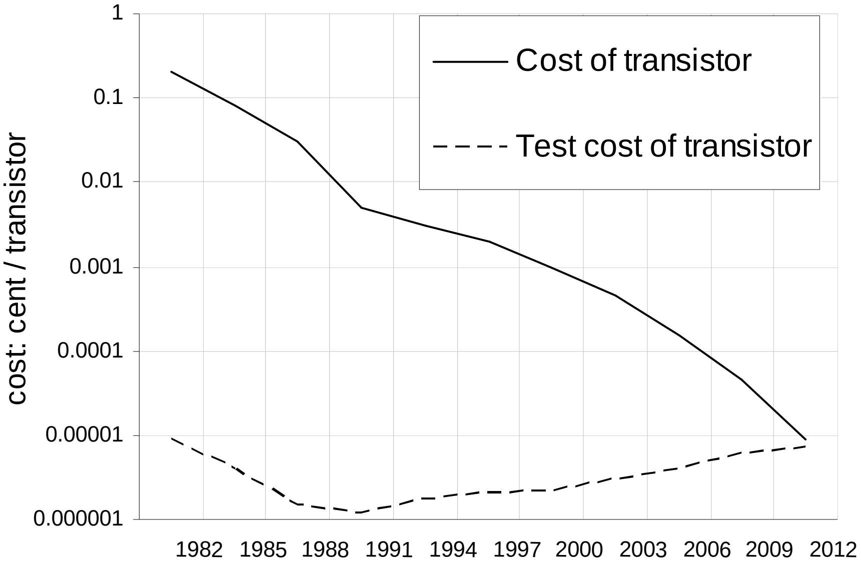


***Figure 2.4**: SIA test capital trend graph [17].*

While it is more difficult to define a “flexibility variable” that relates cost to system features, it is clear that the cost of test equipment increases significantly with flexibility. Due to problems such as power supply regulation, temperature variation, and electrical parasitics, tester timing inaccuracies continue to rise as a function of the shrinking clock periods of high-performance designs. The graph in Figure 2.5 shows trends for device period, overall tester timing accuracy (OTA), and the resulting percentage yield loss.

The use of DfT is showing a rapid growth trend across the semiconductor industry. The reasons for this growth are many. DfT can greatly shorten test development cycle times, improve fault coverage, access multiple internal circuits in a SoC through a common subset of pins, test high performance circuits with medium performance interfaces, facilitate massively parallel testing, and more. Except for a few isolated point solutions, most structural (DfT) testing to-date has been performed on conventional digital ATE. This has the unfortunate consequences of exceeding the test requirements in some areas and being sub optimal in others. This means that many devices utilizing DfT are paying a higher cost of test than necessary. Therefore, a need has arisen for the development of specialized ATE, targeted at this “new” class of DfT savvy devices [33].

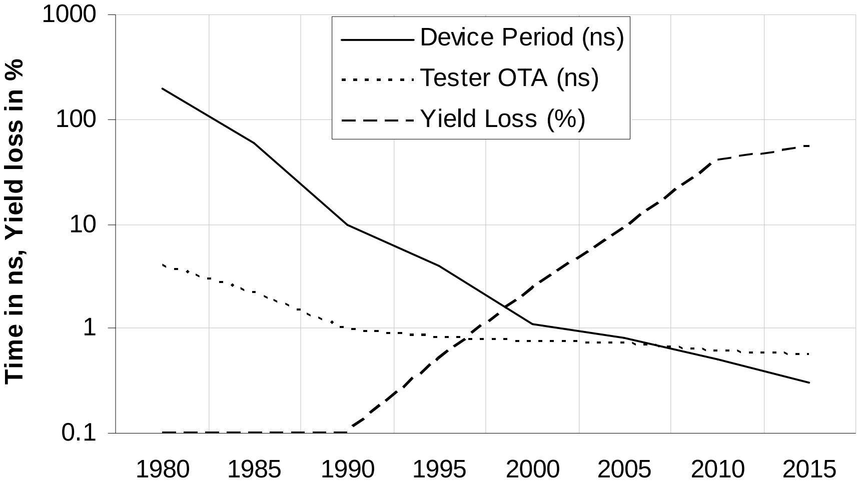


Figure 2.5: Tester accuracy and projected yield loss trends [17].

2.2.10 DfT Testers

The price of a digital ATE depends on its channel count, vector memory depth, and accuracy. While SoCs seemed to require large, and expensive ATEs, companies have started to realize that for many of their (structural) tests, at most 10% of the capabilities of their expensive ATEs is really used [33]. This led to a wave of interest into low-cost *‘DfT Testers’*. Such testers do work, provided that one is willing to rely on DfT-based structural testing only, and one is willing to add some on-chip DfT hardware to compensate for the reduced capabilities of the low-cost ATE. As already indicated in the section on DfT, a main change in production test solutions is to find solutions via advanced DfT techniques. For example loop back techniques for high speed I/O testing [17] reduce the need for special high end capabilities on the ATE. Not only are these capabilities expensive, especially since they are in general active less than 1% of the total test cycle, they also have the risk of being outdated in a very short time. Using proper DfT techniques reduces the needs for high-end performance, enabling real low-cost production ATE. A prerequisite for this is that a thorough evaluation and characterization of the device is still possible with a clear correlation to the DfT technique used.

2.2.11 The Road Forward for High-Speed ATE

Although the future seems difficult to predict, there are some ATE trends that are clearly in motion already. It is reasonable to expect that the number of new tester platforms will be far less than in the past. With the increasing importance of test subcontractors in tester purchase decisions, the key features of tester platforms will be flexibility and scalability. It will be increasingly difficult for semiconductor companies to predict their test requirements due to fast rate changes in the industry. Successful future ATE companies must develop tester platforms with built-in flexibility and scalability so that the platform has a longer asset life even in light of unpredictable and increasing test requirements.

Focused testers that are very narrowly defined may have a very limited customer base. Many recently introduced DFT-only testers could experience difficulty finding wide acceptance in the marketplace because of this. A successful tester platform must have the capability to reconfigure itself as customers' test requirements evolve [34].

For example, a customer may purchase a low cost digital tester initially and expect later reconfiguration to a high-performance mixed-signal tester with minimal transition cost. Such a tester requires an investment by the manufacturer and presents both economic and technical challenges to ATE companies.

Some in the industry have claimed that the open architecture standard [12] will be a solution to this problem. Open architecture makes the claim that the promised flexibility and scalability of an open architecture platform can be delivered, and risks inherent in building such a platform can be distributed among its participants. This reasoning seems flawed. The key challenge, and therefore the key innovation required in creating successful open architecture is to provide the modularity necessary for flexibility and scalability. This must be done without sacrificing throughput, efficiency and cost while minimizing the integration burden as future instruments are added to the platform. An adopted architecture must comprehend the integration requirements for not just hardware, but also software, applications, and service [34].

2.3 DfT for Delay-Fault Detection

Digital IC performance has followed Moore's law, improving annually by 30%. However, ATE performance has improved by only 12% annually. The discrepancy between ATE edge placement accuracy and circuit under test performance will make at-speed logic testing increasingly difficult for future deep-submicron technologies. This scaling causes a three- to fourfold increase in transistor off-state leakage current per μm and IC background leakage for every technology generation. Consequently, the deep-submicron regime is eroding the effectiveness of current-based test techniques and stress testing (burn-in). In addition, the number of parametric defects that cause timing-only failures as opposed to catastrophic failures is increasing. These difficult-to-detect defects are causing an increasing number of test escapes. This poses a serious obstacle to the long-term reliability of future digital ICs [35].

Two types of direct test techniques are therefore receiving increasing attention: ATE with improved capabilities and higher frequencies, and DfT and BIST for improved circuit testability. Some of these methods require incorporating additional DfT structures.

2.3.1 DfT Methods for Testing High-Speed Digital Circuits

Defect mechanisms in deep sub-micron designs are often manifested as speed failures under very specific conditions. Most commercial ATPG tools, which are based on the stuck-at and transition fault models [2], are not equipped to handle these complex failure modes. Although ATPG technology has progressed during this time, the success of these tools is predicated on providing a high degree of access, controllability, and observability to the internal nodes of the design by using DfT techniques.

Scan design, the best-known structured DfT technique, comes at the cost of both performance and area, although some trade-off is possible. In order to meet tight timing requirements, high-performance designs tend to have very few gates between storage elements, which results in a high latch-to-logic ratio. Therefore, implementing scan DfT generally translates into sacrificing considerable silicon real estate.

Another DfT technique that is gaining acceptance in the industry is BIST, which incorporates mechanisms to generate stimuli and compress responses for later off-chip comparisons into the design. BIST allows a large number of patterns to be applied at speed in a short time, with very little tester support. However, most logic BIST techniques that enjoy commercial success today require full scan, or close to it. In addition, they need design changes to enhance random-pattern testability, to allow at-speed test application, and to prevent the system from getting into an unknown state that can corrupt the compressed response.

2.4 Built-In Self-Test Approach for DfDT

Test and diagnosis must be quick and have high fault coverage. One way to ensure this is to specify test as one of the system functions, so it becomes self-test. The fault coverage and the diagnostic resolution of those software-implemented tests are not always as high as required. The diagnostic resolution may be poor because the software must test parts that are difficult to test, and therefore it may not effectively determine which part is at fault. Also, software tests can be long, slow, and expensive to develop. Therefore, it becomes increasingly attractive to build the self-test function into the hardware [36]. It is also most effective to consider testing as early in the design cycle as possible. Otherwise, costly prototyping turns (cycles of redesign and refabrication of the prototype) result, and these lead to schedule slip for product introduction.

Systems designed without an integrated test strategy covering all levels from the entire system to components can best be described as chip-wise and system-foolish. With properly designed BIST, the cost of added test hardware will be more than balanced by the benefits in terms of reliability and the reduced maintenance cost. Thus, these benefits, rather than the cost, can be passed on to the customers. The savings from BIST include reduced test generation effort at all levels, reduced test effort at chip through system levels, improved system-level maintenance and repair, and improved component repair. Gordon and Nadig [37] described the economic impact of signature analysis and BIST on the first systems that used BIST: Hewlett-Packard digital voltmeters, one of them being the HP 3455A. Development in time and costs rose roughly 1%. There was a 1% increase in parts cost due to added jumpers and extra ROM space required in the electronics for signature analysis, but total factory costs dropped, because of a 5% decrease in other materials costs.

For example, it was no longer necessary to divide the product into small, replaceable modules. Also, BIST reduced service-module inventory at the factory, and decreased administrative and handling costs for failing units returned to the factory.

One unfortunate property of SoC circuits is that testing cannot be easily partitioned. Consider two cascaded devices. There is frequently no simple way to obtain tests for the complete system from tests for the individual parts. In fact, even though each part is fully testable and has a test set that gives 100% stuck fault coverage, the cascaded connection of the two parts will often have untestable and redundant hardware and much lower stuck-fault coverage. In other words, testing is a global problem.

BIST efficiently tests embedded components and interconnect, thus reducing the burden on system-level test, which now only needs to verify the synergy among the functional components [36]. If faults occur, the BIST hardware should be designed to indicate via an error signal or bus which sub-assembly is faulty. This greatly reduces repair costs. Typical quality requirements are 99.99% single stuck-fault coverage or 100% interconnect fault coverage.

Another advantage of BIST is that the testing capability grows with the technology, whereas with external testing, the test capability always lags behind the SoC technology capability. Logic gates and transistors are relatively cheap compared to the labour needed to develop test programs, the cost of automatic test equipment and the cost of real time for the test to be run on production chips with ATE. It has the disadvantage that deterministic test with BIST is very difficult and expensive, and the cheap versions only provide pseudo-random tests which very often do not cover the faults sufficiently [38].

A delay-fault BIST testing system has the standard BIST architecture, but with a hybrid pattern-generator optimized to test both stuck-faults and delay-faults replacing the standard LFSR pattern generator. For delay-fault BIST, one of the problems that must be addressed is hazards in the circuits. One way to avoid hazards is to use a hardware pattern generator that creates single-input changing patterns, so only one input changes during each clock period.

Figure 2.6 shows a possible BIST arrangement in which a test vector generator produces the test vectors that must be applied to the circuit under test.

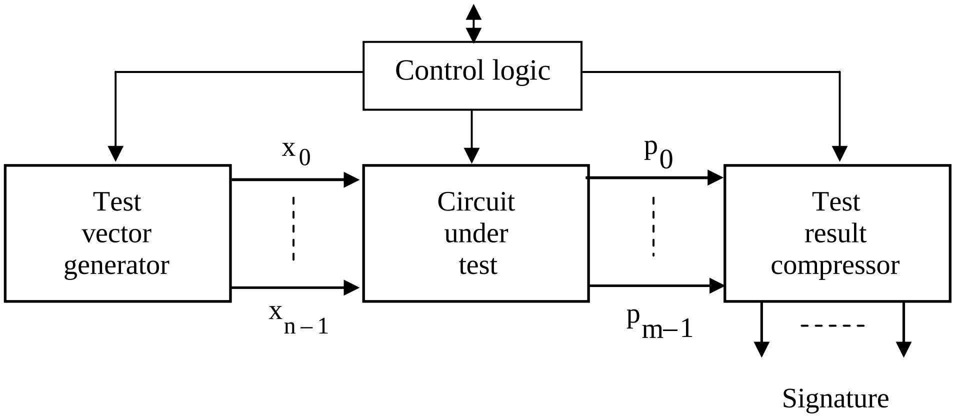


***Figure 2.6**: Possible BIST arrangement.*

2.4.1 Different BIST Options

Modern SoCs integrate digital, analog, and memory modules into a single IC. Testing such heterogeneous ICs requires either the use of multiple, dedicated testers or a single, complex tester for testing the digital, analog, and memory parts. The use of Logic BIST for the digital modules, analog BIST for the analog modules, SoCBIST for System-on-Chip, provides that a single, low-cost external tester is sufficient.

As with most of these BIST options, Logic BIST requires some extra circuitry and, because it is based on a final signature, carries rules that must be followed. In other words, logic BIST compels one to design following strict guidelines which, of course, ought to be standard procedure, anyway to avoid unknown states, or values, that might corrupt the signature. The alternative is that one may have to add circuitry to mask off such states. The end result is a better chip, and a better board later, because erroneous inputs will be detectable at both stages.

BIST is rapidly emerging as a robust alternative for new intellectual property (IP) designs. It is a design-for-testability methodology where testing is accomplished by built-in hardware and software features. A SoC performs BIST by generating test patterns and evaluating test responses on chip.

What is remarkable about BIST is that it embeds test into the circuit. As a result, the amount of test data stored and transferred across the chip boundary is insignificant. Consequently, I/O throughput is not a problem for BIST, even for extremely dense SoCs. Moreover, the test application time is not limited by the throughput. Most cores can be tested in parallel in a SoC, as long as the testing does not exceed the device's maximum power consumption and heat dissipation requirements. Since BIST reuses test capabilities analogous to design reuse, it offers similar advantages, the reduction of product development cycle, while reducing the cost of manufacturing testing. As a result, BIST is an extremely attractive component of an IP DfT strategy.

2.5 Conclusions

The critical issue with sub-90nm processes is yield and yield loss is no longer primarily due to random defects as it was in the past. Today yield loss is the result of not only random defects but also printability problems and systematic defects, e.g. signal integrity.

With DSM technology, new fault models and testing mechanisms are required. For example, contact defects require at-speed delay testing, and metal defects require resistive shorts testing. Open vias, which are becoming more of a problem, also require delay tests. Large chips introduce relatively greater proportions of wiring, which means that wire-related faults (such as bridging) become more important in considering the delay test strategy. A model to look at the effect of cross-talk needs to be developed. The most promising approach to deal with cross-talk is noise-aware timing analysis, which can alternate between cross-talk noise analysis and timing analysis, feeding the results of each analysis to each other. Because cross-talk and signal-integrity issues wield such dramatic impact on timing, noise-aware timing analysis is emerging as a critical element in SoC verification. As process technologies continue to evolve, this new breed of analysis will be increasingly vital for achieving timing closure, meeting design performance goals and realizing time-to-market objectives for SoC designs

Inevitably, the dramatic growth in design complexity has implications for the number of test vectors now required to test chips. More vectors imply that more time is required to test each device, with the consequent impact on cost. There is a need for a new approach to test.

The cost saving associated with the use of BIST manifests itself throughout the complete life-cycle of the device; it much reduces or in some cases eliminates the task of test-vector generation and fault simulation, while reducing the costs of fabrication testing, board and system test and the in-service testing by reducing test times and ATE costs. It is accepted that these factors must be weighed against the disadvantage of reduced yield and higher design and fabrication costs through the use of increased silicon real-estate. Digital IC performance has followed Moore's law, improving annually by 30%. However, ATE performance has improved by only 12% annually. For this reason, many companies have expressed an interest in using testers designed to focus on DfT test methods. Since these machines can cost as much as an order of magnitude (or more) less than a traditional tester, moving tests from the more expensive environment to the less expensive environment can save significant amounts of money. This all makes a lot of sense, of course; but is it possible to move tests from the more expensive machines to the less expensive systems? The answer lies in the implementation of DfT structures in the devices to be tested.

This chapter has given an insight into the need to test for delay-faults and the origin thereof. Also, attention was paid to the difficulties related to delay-fault testing and the use of DfT to ease this difficulty. It is therefore inevitable that attention has to be paid to the development of structure to test for delay-faults that are reliable and to extend these techniques to be able to test embedded cores for delay-faults.

2.6 References

[1] http://www.intel.com/technology/computing/archinnov/platform2015/

[2] M.L Bushnell and V.D. Agrawal, "*Essentials of Electronic Testing for Digital, Memory & Mixed-Signal SOC Circuits*", ISBN 0-7923-799-1-8, Kluwer Publishers, Massachusetts, 2001.

[3] A.L. Crouch, "*Design for Test for Digital IC's and Embedded Core Systems*", ISBN 0-13-084827-1, Prentice Hall, New Jersey, 2000.

[4] H. Yan and A.D.Singh, “Experiments in Detecting Delay Faults using Multiple Higher Frequency Clocks and Results from Neighboring Die”, *Proceedings of International Test Conference (ITC)*, September 2003, Charlotte, pp. 105-111.

[5] S. Padmanaban and S. Traqoudas, “A Critical Path Selection Method for Delay Testing”, *Proceedings of International Test Conference (ITC)*, October 2004, Charlotte, pp. 232-241.

[6] R.C. Tekumalla and S. Davidson, “On Identifying Indistinguishable Path Delay Faults and Improving Diagnosis”, *Proceedings of International Test Conference (ITC)*, October 2001, Baltimore, pp. 993 - 1002.

[7] M. Sharma and J.H. Patel, “Testing of Critical Paths for Delay Faults”, *Proceedings of International Test Conference (ITC)*, October 2002, Baltimore, pp. 634 – 664.

[8] N. Mukherjee, T.J. Chakraborty, S. Bhawmik, "A BIST Scheme for the Detection of Path-Delay Faults," *Proceedings of International Test Conference (ITC)*, 1998, Washington DC, pp. 422-431.

[9] K.T. Lee and J.A. Abraham,” Critical Path Identification and Delay Tests of Dynamic Circuits”, *Proceedings of International Test Conference (ITC)*, 1999, Atlantic City, pp. 421-430.

[10] H.G. Kerkhoff, K. van Nee and H. Speek, "DfT for Delay-Fault Diagnosis in Digital High-speed ICs", *Proceedings of IEEE European Test Workshop (ETW)*, Constance, Germany, 1999, pp. SP4/5 9-10.

[11] T.J. Chakraborty, V.D. Agrawal, and ML. Bushnell, “Delay Fault Models and Test Generation for Random Logic Sequential Circuits,” *Proceedings of the 29th Design Automation Conference (DAC)*, June 1992, Anaheim, pp. 165-172.

[12] http://www.advantest.com/aac/Products/product.html

[13] V.D. Agrawal, C.J. Lin, P.W. Rutkowski, S. Wu, and *Y*. Zorian, “Built-In Self-Test for Digital Integrated Circuits,” *AT&T Technology Journal*, Vol. 73, March/April 1994, pp. 30-39.

[14] S. Mourad and Y.Zorian, “*Principles of Testing Electronic Systems*”, ISBN 0-471-31931, John Wiley &Sons, 2000.

[15] M. Sugihara, H.Date, and H.Yasuura, “Analysis and Minimization of Test Time in a Combined BIST and External Test Approach”, *Proceedings of Design, Automation and Test in Europe Conference (DATE)*, Paris, March 2000, pp. 134-140.

[16] V.D. Agrawal and T.J. Chakraborty, “High-Performance Circuit Testing with Slow-Speed Testers,” *Proceedings of International Test Conference (ITC)*, 1995, Washington DC, pp. 302-310.

[17] M. Shoji, “*CMOS Digital Circuit Technology*”, ISBN 0-13-138850-9, Prentice Hall, New Jersey, 1988.

[18] Semiconductor Industry Association. *International Technology Roadmap for Semiconductors (ITRS)*. SIA, Austin, December 2003. (Available via http://public.itrs.net/).

[19] S. Barton, “Characterization of High-Speed (Above 500 MHz) Devices Using Advanced ATE-Techniques, Results and Device Problems,” *Proceedings of International Test Conference (ITC),* 1989, Washington, DC, pp. 860-868.

[20] L. Ackner and M.R. Barber, “Frequency Enhancement of Digital SOC Test Systems,” *Proceedings of International Test Conference (ITC),* 1990, Washington DC, pp. 444-451.

[21] D.C. Keezer, “Multiplexing Test System Channels for Data Rates Above 1Gb/s,” *Proceedings of International Test Conference (ITC),* 1990, Washington DC, pp. 362-368.

[22] D.C. Keezer, “Real-Time Data Comparison for Gigahertz Digital Test,” *Proceedings of International Test Conference (ITC),* 1991, Nashville, pp. 790-797.

[23] D. Wimmers, K. Sakaitani, and B. West, “500 MHz Testing on a 100MHz Tester,” *Proceedings of International Test Conference (ITC),* 1994, Washington DC, pp. 273-278.

[24] L.A. Gasbarro and M.A. Horowitz, "Techniques for Characterizing DRAMS with a 500MHz Interface," *Proceedings of International Test Conference (ITC),* 1994, Washington, DC, pp. 516-525.

[25] J.Rajski and J. Tyszer, "Arithmetic Built-In-Self-Test for Embedded Systems", Upper Saddle River, New Jersey: Prentice-Hall, 1998.

[26] V.H. Nguyen, H van Kranenburg and P.H. Woerlee, "Copper for advanced interconnect", *Proceedings of Third International Workshop on Materials Science (IWOMS '99),* November 1999, Hanoi, pp. 90-99.

[27] C.J. Lin and S.M. Reddy, "On delay fault testing in logic circuits", *IEEE Transactions on CAD*, vol. CAD-6 (5), September 1997, pp. 694-703.

[28] M.A. Breuer, "The effects of Races, Delays, and Delay Faults on Test Generation" *IEEE Transactions on Computers*, vol. C-23, no. 10, October 1974, pp. 1078-1092.

[29] B.C. Paul and K. Roy, "Testing Cross-Talk Induced Delay Faults in Static CMOS Circuit Through Dynamic Timing Analysis", *Proceedings of International Test Conference (ITC)*, October 2002, Baltimore, pp. 384-390.

[30] W. Chen, S.K. Gupta and M.A. Breuer, "Test Generation for Crosstalk-Induced Delay in Integrated Circuits", *Proceedings of International Test Conference (ITC)*, 1999, Atlantic City, pp. 191-200.

[31] C. Huang," New Problems May Complicate SOC Design", CadMOS *Design Technology – Electronic News*, October 2002, San Jose.

[32] K. Furuya and E.J. McCluskey, "Two-pattern test capabilities of autonomous TPG circuits", *Proceedings of International Test Conference (ITC),* October 1991, Nashville, pp. 704-711.

[33] S. Comen, "DfT-Focused Chip Testers: What Can They Really Do?", *Proceedings of International Test Conference (ITC)*, October 2000, Atlantic City, page 1120.

[34] T. Lee, "Future ATE: Perspective & Requirements", *Proceedings of International Test Conference (ITC)*, September 2003, Charlotte, page 1300.

[35] M. Sachdev, "Current-Based Testing for Deep-Submicron VLSI," *IEEE Design & Tes*t, vol. 18, no. 2, Mar.-Apr. 2001, pp. 76-84.

[36] V. D. Agrawal, C. R. Kime, and K. K. Saluja, "A Tutorial on Built-In Self-Test, Part 1: Principles," *IEEE Design & Test of Computers*, vol. 10, no. 1, March 1993, pp. 73 – 82.

[37] G. Gordon and H. Nadig, "Hexadecimal signatures Identify Troublespots in Microprocessor Systems," *Electronics*, vol. 50, no. 5, March 1977, pp. 89-96.

[38] H. Vranken, F. Meister and H. J. Wunderlich, "Combining Deterministic Logic BIST with Test Point Insertion", *Proceedings of European Test Workshop (ETW)*, May 2002, Corfu, pp. 389-394.

Chapter 3

Controllable Scan Flip-Flop as Design-for-Delay-Testability Structure

The recent increase in BIST and DfT in IC design is expected to cause a change in test-equipment requirements and a reduction of test-cost in the future. This, together with the rapid increase in clock frequencies of digital circuits due to sub-micron technology, has created an increased concern about cost-effective detection of delay faults [1-3]. Several approaches have been proposed for improving the delay-fault detection of high-speed digital circuits. Many different methods for improving delay-testability have been presented [4]. The quest therefore is for providing a DfDT structure to decrease testing costs by testing the high-performance digital circuit at a significantly lower speed than its operating speed while still guaranteeing its high-speed behaviour [5]. This has led to a growing interest into the development of special flip-flops to introduce controlled delay in a circuit, which circumvents the use of at-speed tests to determine delay faults.

This chapter will present the controllable scan flip-flop as a DfDT structure. The different components of the structure will be handled as well as the problem of clock-skew. The first section (3.1) of this chapter will introduce critical path and in section 3.2 the controllable scan flip-flop will be handled. In section 3.3 the different components of the DfDT structure will be discussed in detail. The susceptibility of the DfDT structure to process and application variables are described in the next section (3.4). In the fifth section (3.5) the use of the DfDT structure within a low-speed BIST environment will be dealt with. Finally, the last section (3.6) will present the conclusions with respect to the use of the controllable scan flip-flop as a DfDT structure within a low-speed BIST environment.

3.1 Critical Paths

Delay defects in a manufactured integrated circuit that normally do not change the logic behaviour of the circuit, but only affect the speed at which the circuit can operate, require a special testing methodology, referred to as delay-fault testing. Basically it verifies the timing specifications of the circuit. However, at the required speed, its functional behaviour can result in stuck-at like faults in the behaviour. The path-delay fault model [6] is the most accurate delay-fault model, since it models distributed as well as localized delay defects. The paths with the largest delay, called critical paths, are the most important for delay-fault testing since a delay defect in the circuit is most likely to cause a timing violation in these paths. Testing the critical paths in a circuit is essential to cover distributed delay and small delay defects in a manufactured circuit. However, in most circuits only a small percentage of functionally irredundant critical paths are robustly testable [7]. Functional irredundant critical paths refer to those paths in a circuit that can affect the cycle time and are functionally sensitizable. A robust test guarantees the detection of a fault on the target path regardless of the delays on all other signals in the circuit [8]. In almost all circuits, a very large number of paths are functionally irredundant [9].

Hence, in practice, structural delay-based approaches are used to select those functionally irredundant paths for testing that are most likely to fail in the presence of delay defects. The simplest structural approach is to select all those paths in the circuit whose estimated delay, as determined during static timing analysis, is longer than a certain threshold, because a delay defect in the circuit is most likely to cause a timing violation in such a path. The technique used by Reddy [10] selects a set of paths such that for each line, the path with the largest delay passing through is selected. More sophisticated procedures [11] either exploit the fact that path delays are related to each other because paths in a circuit share many lines, or use statistical manufacturing-process information [11-13].

The first step in any delay-fault test strategy is the identification of the critical paths as well as establishing the test patterns to sensitize these critical paths. Much work has been published on this matter - for instance in reference [14]. It must be stated that it is not part of our study. It will be assumed that all critical paths have been identified and test patterns to sensitize them are available. The assumption is a realistic assumption as a part of industry is doing it already.

3.2 The Controllable Delay Scan Flip-Flop

The delay in a normal flip-flop can result from its combinational logic and interconnection-line parasitics. An additional test-mode clock can also control a flip-flop delay. The basic idea for the controllable delay flip-flop [15] has its origin from a (single-clock) pulse-triggered flip-flop that has different operational modes.

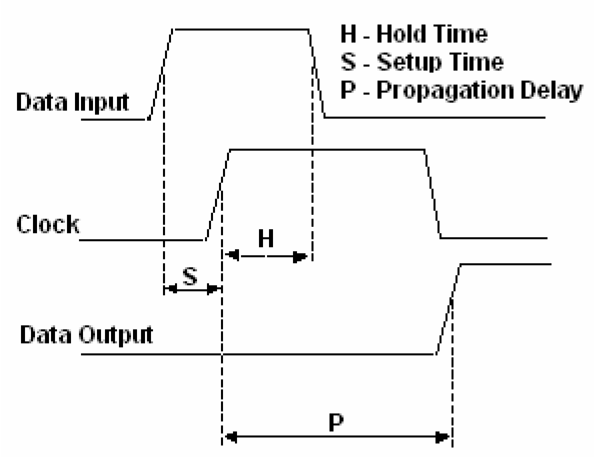


Figure 3.1: *Timing Parameters for Master-Slave Flip-Flops.*

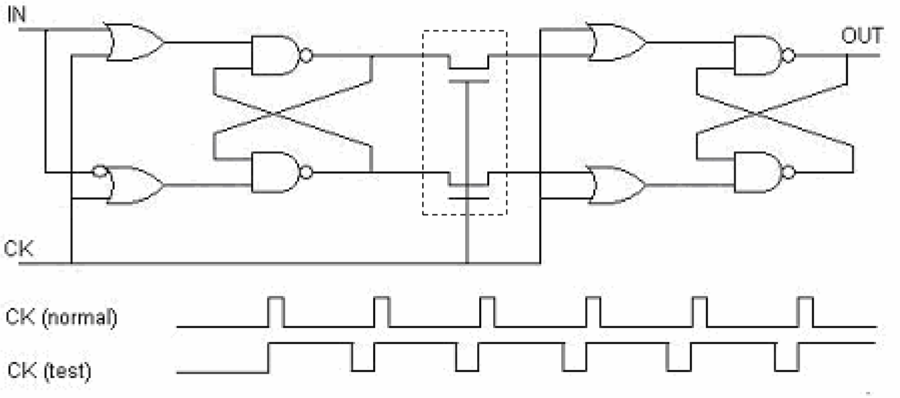


Figure 3.2: *Pulse-triggered flip-flop with the inserted dynamic latch and its clock waveform in normal and test modes.*

The timing parameters of the traditional flip-flop are shown in Figure 3.1. Basically, a dynamic latch has been introduced inside a traditional master-slave flip-flop. The two pass transistors act as the dynamic latch (enclosed within the dotted-line box in Figure 3.2) and must hold data while both static latches on either side remain transparent.

This two-latch arrangement allows for the modulation of the flip-flop delay with the test-mode pulse width [16]. The pulse-triggered flip-flop with the inserted dynamic latch is shown in Figure 3.2. In a sequential circuit, as shown in Figure 3.3, all signal paths must satisfy the timing relationship given in Equation (3.1). Here, T denotes the clock period, PD_{FFi} the propagation delay through the flip-flop (FFi), PD_{CL} the propagation delay through the combinational logic circuit (CL), and T_{Setup} the set-up time for the flip-flop. The controlled flip-flop can operate in normal as well as test mode. In normal mode, an increased pulse width of the clock will increase the propagation delay of the flip-flop.

$$T \geq PD_{FF} + PD_{CL\,+}\,T_{Setup} \qquad \textbf{(3.1)}$$

Considering that PD_{CL} and T_{Setup} remain unchanged, the clock period, T, must become larger for Equation (3.1) to remain valid. In other words, a slower clock frequency as compared to the operating frequency is able to test critical paths with the same timing specifications as when testing at normal frequency.

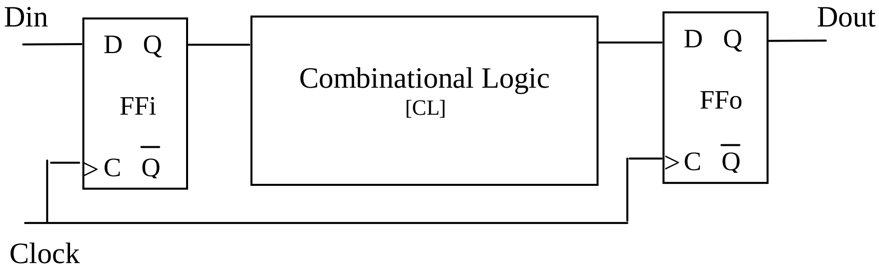


Figure 3.3: *Simple sequential circuit.*

One of the most significant implementation issues in the pulse-triggered flip-flop is the realization and propagation of a precise pulse width at the chip level. A small pulse width needed for high-speed normal mode operation may appear significantly distorted due to the interconnect impedance. The dynamic latch in the controllable-delay flip-flop, in addition, is rather time critical.

However, an additional test-mode clock can control flip-flop delay. These flip-flops are referred to as controlled delay flip-flops. As scan testing is often used for stuck-at testing in these flip-flops anyhow, it is called controllable delay scan flip-flops (CDSFF) and is based on a conventional scan flip-flop. This has the additional advantage that also the delay between embedded blocks can be detected and shifted out in a serial, scan-like manner.

The CDSFF allows master-to-slave data transfer on the rising edge of the test clock. A schematic diagram of the CDSFF is depicted in Figure 3.4.

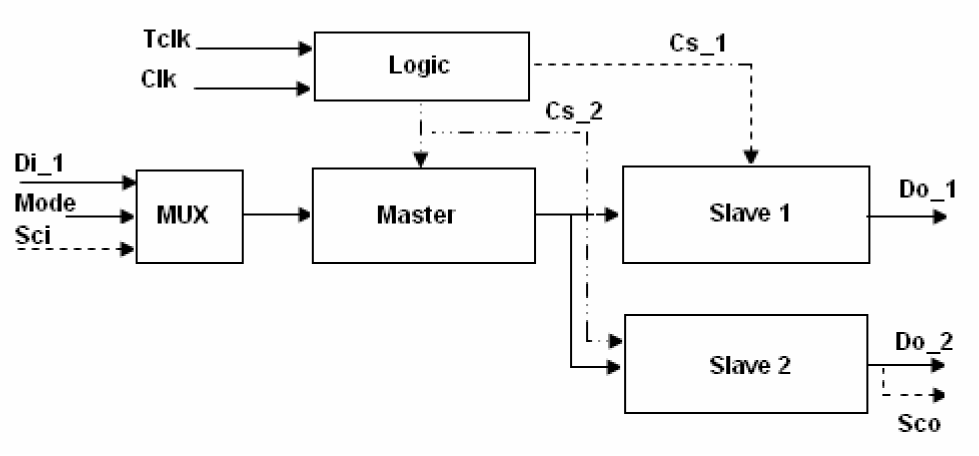


Figure 3.4: *Block diagram of the Controllable-Delay Scan Flip-Flop (CDSFF*).

The scan-input, *Sci*, and scan-output, *Sco*, become active with the signal labelled "*Mode*". *Di_1* and *Do_1* are the respective data input and output while *Clk* denotes the regular clock and *Tclk* the additional test clock. In the test mode, the propagation delay *Tp* of the flip-flop is controlled by the test clock *Tclk* because master to slave data transfer is on the rising edge of the test clock *Tclk*. This relation of the propagation delay to the test clock is shown in Equation 3.2. T_{Clk} denotes the period of the clock signal and t_{offset} the offset of the test clock signal with regard to the clock signal (see Figure 3.5).

$$Tp = T_{Clk} - t_{offset} \qquad \textbf{(3.2)}$$

The significance of the additional test clock is illustrated in Figure 3.5. This figure depicts the test-mode timing diagrams of detecting an interconnection delay-fault (labelled "del" in Figure 3.5) between CDSFFs. The data input to the next CDSFF is labelled "*Di_2*".

In the normal mode, the test clock has no function and is held high ensuring normal (scan) flip-flop operation. However, during the testing of the IC, it operates as a clock with tester-programmed time offset ("*offset*", in Figure 3.5) with regard to the regular clock, *Clk*. It is a tester-programmed offset because the tester is able to vary the offset of the test clock, *Tclk* with regard to the regular clock, *Clk*. When the test clock is active, it controls via C*s_1* the data transfer from the master to the (top) slave latch, *Slave 1* (Figure 3.4), in the CDSFF. In other words, depending on the timing relationship between the clock and test clock, a delay is introduced between master and slave latch, Slave 1, of the CDSFF (Figure 3.4). The net effect is that the CDSFF data "*Di_1*" appears at its output *Do_1* after the additional delay at the rising edge of *Tclk* and the internal propagation-delay *Tp*. After the delay of the interconnection line ("*del*") it arrives at the input *Di_2* of the next CDSFF. Before the set-up time *Ts* of the normal scan flip-flop (SCFF), the data can be read in the master part of the next CDSFF.

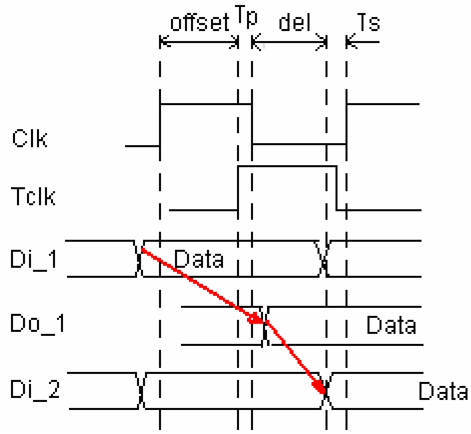


Figure 3.5: *Timing diagram of the CDSFF in test mode and assuming a 50% duty cycle of clock and test clock.*

The test-mode clock period should be sufficiently large to accommodate the delay time in the interconnect lines that interface the different parts of the CDSFF. It is obvious that as the offset *toffset* is increased, the period of the clock, when in test-mode, is also increased or the clock frequency *fclk* is reduced. In other words, the clock frequency *fclk* can be reduced while the logic circuits within the CDSFF are tested with the same delay margins as when the CDSFF is operating under normal conditions. The realisation of the CDSFF requires additional transistors and an additional test-mode clock input as compared to a normal scan flip-flop as indicated in Figure 3.4. The extra components to implement our complete structure consist of 3846 CMOS transistors for which a chip area of 0.008mm^2 is needed. The choice where to introduce CDSFFs is guided (and limited) by critical delay-path analysis.

3.3 Components in the DfDT structure

The DfDT structure referred to above avoids the requirement of an expensive high-speed tester. In Figure 3.6, the set-up of the DfDT architecture is shown [17], enabling a high-speed chip to be effectively tested at low speed. The approach makes use of the special controlled-delay scan flip-flops, and support circuits for manipulating duty cycles and delay between the system clock "clk" and the additional test clock "tclk".

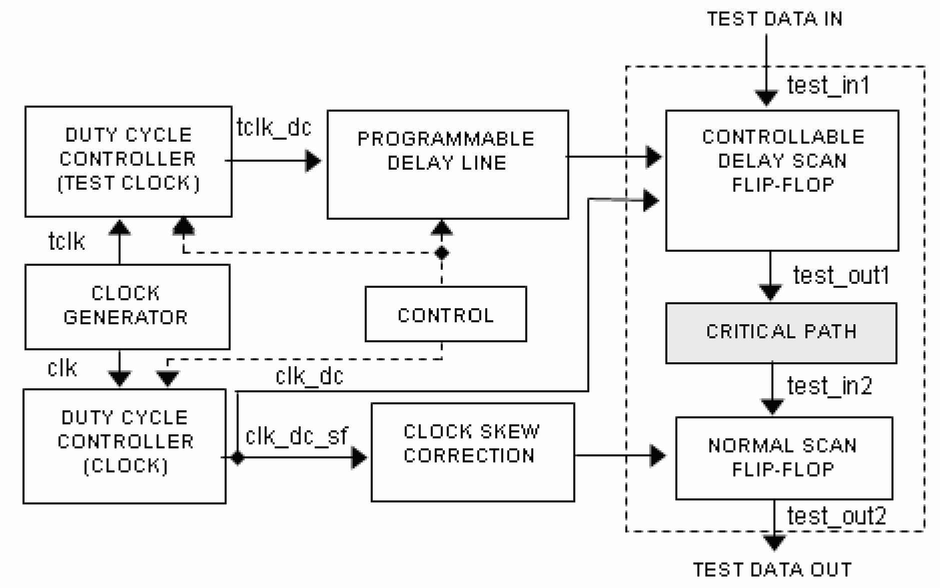


Figure 3.6: *The architecture for detecting small delay faults in a critical path and the DfDT structure used.*

3.3.1 The Programmable Delay Line

The most crucial part of the DfDT circuitry in Figure 3.6 is the programmable delay line (PDL). There are several ways of implementing this structure, e.g. such as discussed in [18]. Inverter chains are tapped at uneven locations by means of NAND gates (Figure 3.7).

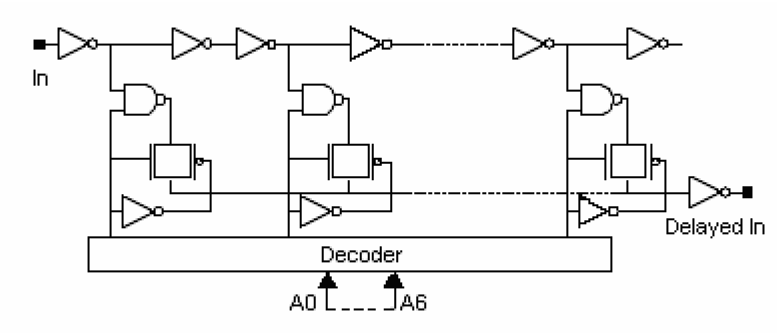


Figure 3.7: *Design of a programmable delay line.*

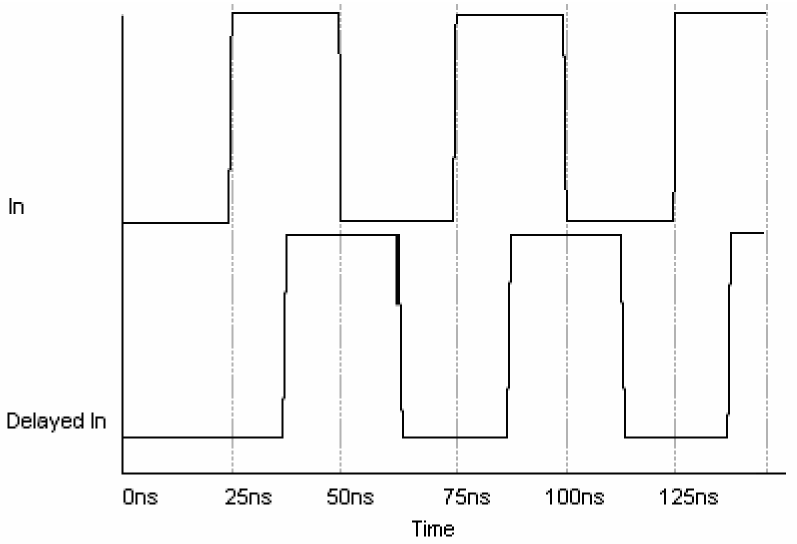


Figure 3.8: *HSPICE circuit simulation of the designed PDL (TSMC 0.35 μm [19]) in the low-speed mode (20 MHz).*

The transmission gates have been inserted for security reasons with regard to the OR-wired (buffered) output line to avoid that more than one tap can be connected to the output line at the same time. The decoder with addresses A0…A6 is a standard implementation of a 7-line-to-127-line decoder using NAND gates. A HSPICE circuit simulation was carried out to verify the behaviour of the PDL circuit.

The minimal timing resolution of the design, meaning the smallest obtainable delay, was set to around 500ps. The maximum obtainable delay between the input and output becomes in this case: 127 * 500ps = 63.5 ns. Figure 3.8 shows the low-speed case (fclk = 20 MHz) with a programmed delay of 12.5 ns. Also in the faster case (fclk = 130 MHz) with little delay (500 ps), the circuit worked satisfactory.

3.3.2 Duty-Cycle Control Element

The second important component in the DfDT architecture of Figure 3.6 is the duty-cycle control (DCC) block. There are two DCC blocks required in the architecture: one for the clock (clk) and one for the test clock (tclk). The simplest implementation consists of the previously discussed programmable delay line in combination with a logic block as shown in Figure 3.9.

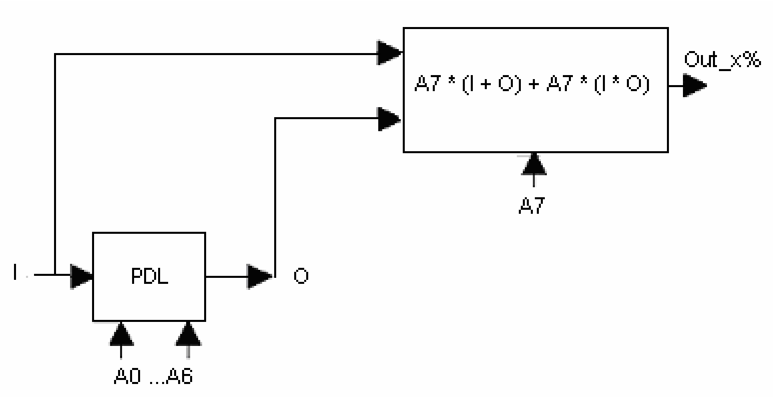


Figure 3.9: *Design of the programmable duty-cycle control block, using the programmable delay line.*

The frequency range is between 10 MHz and 50 MHz, using the previous PLD design and the duty cycle can be varied between 5 % and 95 % with minimum pulse duration of 500 ps. It requires 7 bits (A0 up to A6) to control this duty cycle.

A HSPICE circuit simulation of the DCC block (TSMC 0.35 µm [19]) for the low-speed mode (20 MHz) and a duty cycle of 95 % are shown in Figure 3.10. On top in Figure 3.10 is the 50 % duty-cycle clock signal, and at the bottom the generated 95 % duty-cycle test clock. It is clear that conventional stuck-at testing can test the PDL and DCC circuits. It is also obvious that the PDL is the most complex part of the DfDT structure due to the required delay and duty-cycle resolution.

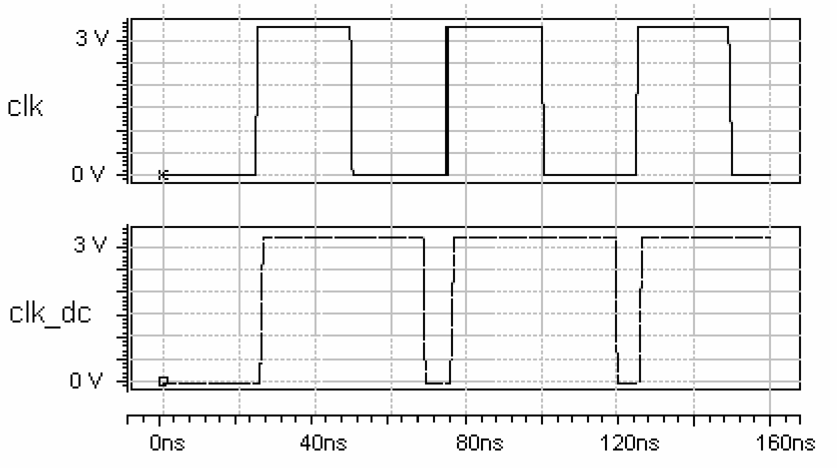


Figure 3.10: *Circuit simulation of the DCC in low-speed mode (20 MHZ) and a duty-cycle of 95%.*

3.3.3 Simulation Results of the DfDT Structure

The used DfDT structure avoids the requirement of an expensive high-speed tester. In Figure 3.6, the set-up of the DfDT architecture was shown, enabling a high-speed chip to be effectively tested at low speed. The approach makes use of the controllable-delay scan flip-flops, and support circuits for manipulating duty cycles and delay between the system clock and the additional test clock.

This architecture and its components that have been described in detail have been simulated at circuit level, making use of HSPICE and using the 0.35 μm TSMC CMOS technology from MOSIS [19].

Figure 3.11 shows the HSPICE simulation results of the structure in the fault-free case at 20 MHz. The critical path was present in the combinational logic part of the chip and has a fault-free delay of 4.8 ns. Next, an extra delay of 620 ps was inserted in the critical path to emulate a possible delay-fault. The critical path has now an overall delay of 5.42 ns.

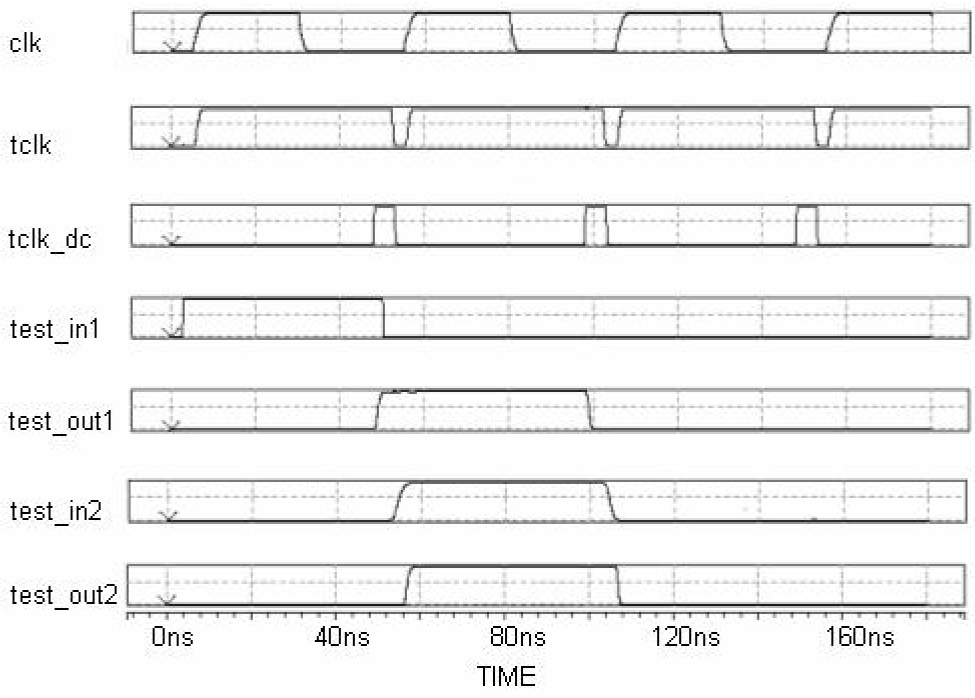


***Figure 3.11**: HSPICE simulation results of the fault- free operation of the circuit in Figure 3.6 (TSMC 0.35 μm technology).*

The HSPICE simulation of the latter is shown in Figure 3.12. The comparison of the result of the fault-free simulation depicted in Figure 3.11 with the result of the simulation with the extra inserted delay of 620 ps shown in Figure 3.12 is proving that the DfDT structure was successful in detecting the induced delay-fault of 620 ps.

These HSPICE simulations at transistor level therefore indicate the viability of our strategy to detect small delay faults at low tester speeds. The required simulation time was 1597 seconds (26 minutes) on an HP9000/J5600 UNIX machine.

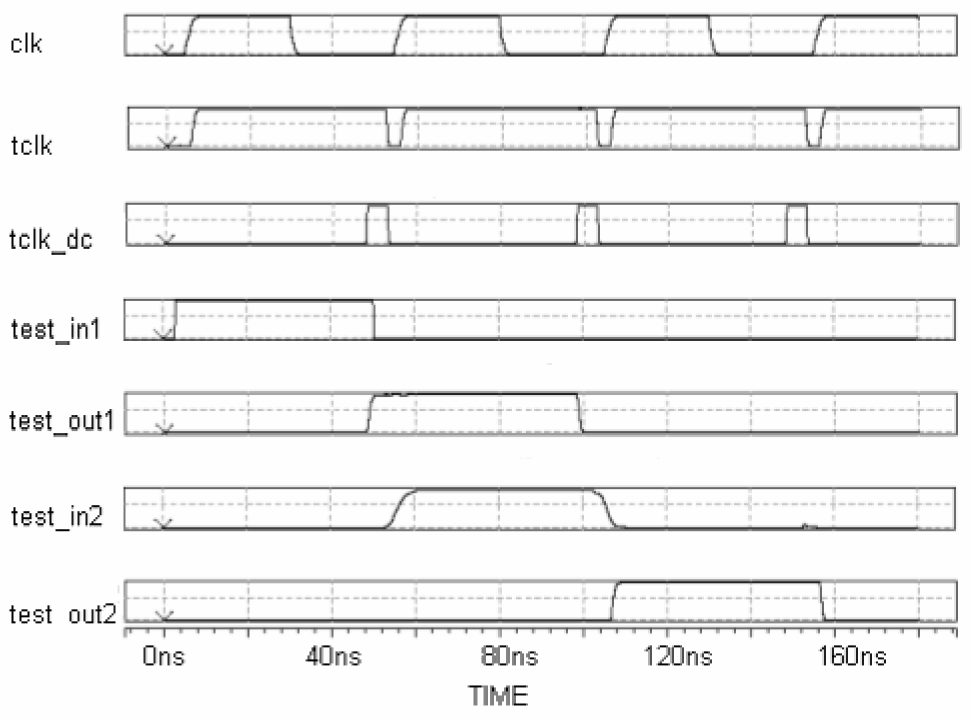


Figure 3.12: *The HSPICE simulation results of the circuit in Figure 3.6 (TSMC 0.35 μm technology) with an inserted delay of 620 ps in the critical path.*

3.3.4 Clock-Skew Correction

Tolerance to process-induced skew remains one of the major concerns in the design of large-area and high-speed clock-distribution networks [20]. Indeed, despite the availability of some efficient exact-zero skew algorithms that can be applied during circuit design [21-22], clock-skew remains an important performance-limiting factor after chip manufacturing. It is of increasing concern for sub–micron high-speed technologies. Excessive clock-skew can give rise to race conditions as well as cycle-time violations. Clock-skew can be positive or negative. Positive skew is the amount of time that must be added to the minimum clock period to reliably transfer data through a system. Positive skew increases the path delay between any two points in the circuit. This effect will decrease the maximum operating frequency and therefore decrease the performance of the device. Negative skew on the other side, can improve the performance of a synchronous system. However, negative skew decreases the amount of time that remains for local data operations within a clock cycle. This imposes a constraint on the minimum path delay of the system. The advantage of negative skew is to reduce the critical-path delays. The two options that exist are to either do clock-skew compensation or to design assuming clock-skew. The latter will be discussed first and subsequently clock-skew reduction.

When designing while keeping skew clock in mind, there are different options to consider. First, there is the option to design for skew. The data-path timing information must be calculated with a suitable timing analysis tool and then be used to generate a range of permissible skew [23]. This permissible range of skew will then be used to determine a feasible clocking schedule. The clock scheduling together with the register locations and buffer description will be used to determine a buffered clock-tree topology. This topology must ensure that the clock will be delivered according to the clock schedule and will then be used to implement the layout of the clock network. This layout will include all the routing of the elements that form part of the clock network. The last part of this process will be to verify the clock timing. The clock tree can be analysed by extracting the parasitic resistivity and capacity and calculate the skews of the clock network. Next it is possible to compare the results for correct operation and to ensure that no race conditions exist whereby signals to the same logic gate have different path lengths. The problems with this process are that it is not simple to implement a set of specific skews. Also it is not easy to accurately calculate the logic path delays and interconnect delays. It is therefore clear that if skew is to be used it must fit into the design and the clock design must follow the placement of logic cells and registers.

Clock-skew reduction is generally accomplished by having a good clock distribution. Several distribution architectures can be implemented such as H-tree, trunk and clustering [18, 24]. These methods, together with frequency stabilization by means of Phase-Locked Loops (PLL) or Delay-Locked Loops (DLL) are normally used to reduce the clock-skew of a particular circuit. For delay-fault detection with our DfDT, the clock-skew is limiting the accuracy of our measurements. Having a high clock-skew, there will be a large uncertainty about the time the detection is carried out. Hence, the time delay cannot be measured accurately. Both PLLs and DLLs check the phase and delay for skew, upon detection, a correction procedure is started which uses feedback and normally takes a few clock cycles to be accomplished.

It is not difficult to mention elements that can help to reduce clock-skew, like over-designing, wide interconnects, extra metal layers to carry clock signals, shielding and much more, but looking at the practical side of these methods it is not that simple to implement it. The following components are critical when looking at practical aspects to minimize clock-skew:

- Make use of extra buffers to equalize delays in all lines and try to keep initial buffers small
- Keep the routing load balanced between different networks of the clock
- Make use of an effective clock distribution network
- The routing of the clock lines are very critical and use a complete clock distribution layout
- The wire width and length of the clock network is also critical in minimizing the clock-skew

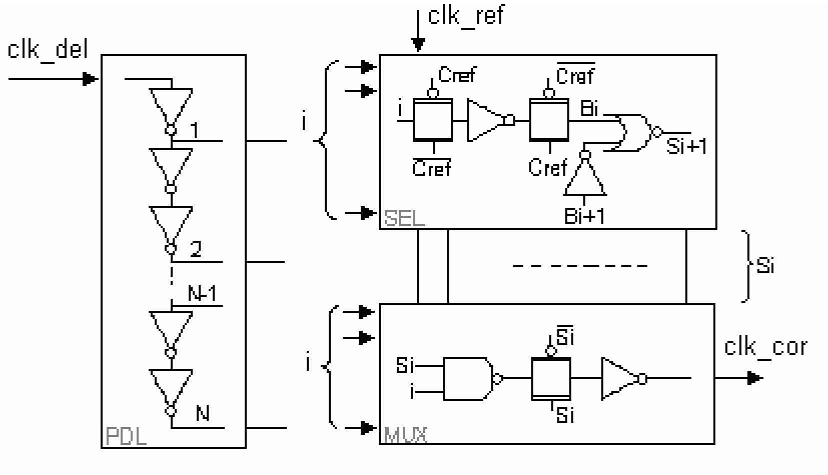
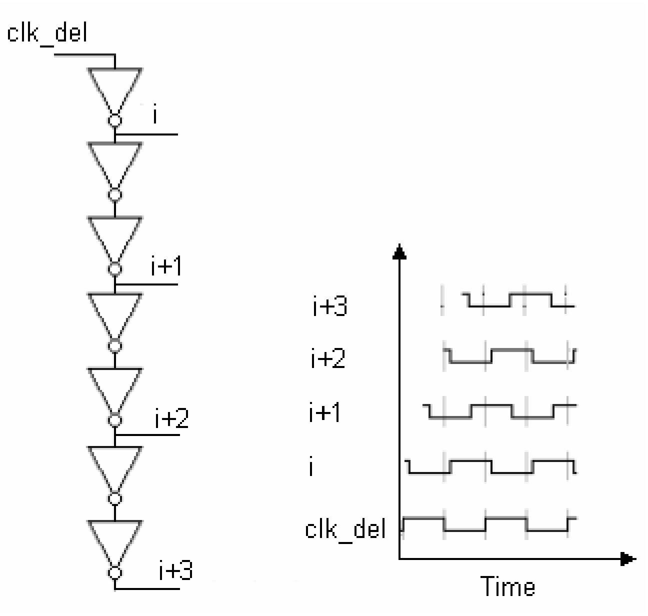


Figure 3.13: *The skew-reduction circuit used. clk_ref denotes the reference clock and clk_del is the skewed clock.*

It should be remembered that the area occupation of the clock-skew reduction circuitry must be looked at as well as the influence of the process variations on the clock-skew of the system.

We have implemented a skew-reduction circuit to give a constant skew-corrected signal to be used in our structure. This circuit will now be discussed in detail. The circuit, which was used in our DfDT structure, is shown in Figure 3.13. This skew-correction circuit does not use feedback and will only be activated during the test cycle.



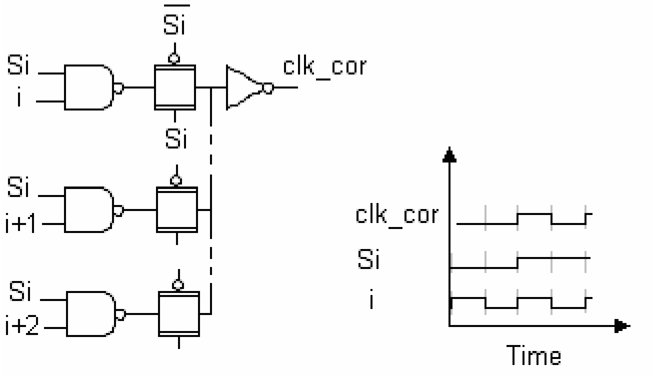
(a) (b)

Figure 3.14: *(a) Schematic of the PDL and (b) its logic signals.*

The method makes use of a PDL (the same design as within our DfDT architecture), which provides a number of delayed replicas of the skewed clock signal. The replicas are then evaluated to select the one that is closest to the reference clock signal. The maximum clock-skew correction possible is equal to the maximum delay of the PDL. The maximum clock-skew correction possible should be less than T_{ref} (the clock period of the reference clock) but more than T_{ref} - t_a (clock-skew tolerance of the circuit). If the maximum delay would be larger than T_{ref}, two different delayed clock signals might be selected at the same time. If the maximum delay of the PDL is less than T_{ref}- t_a, then the clock-skew of the output clock signal cannot be guaranteed to be within its tolerance. The clock-skew tolerance is determined by the delay of the inverters placed between two output clock signals.

Figure 3.14 shows the schematic of the PDL and its clock signals. The inputs of the MUX in Figure 3.15 are the delayed clock signals *i* (outputs of the PDL) and the selection signals *Si* (output selection). The output of the MUX is the input clock signal corrected for clock-skew. The multiplexer consists of several transmission gates. A transmission gate conducts if its selection signal *Si* is high. Thus, a delayed clock signal passes to the output if its corresponding selection signal *Si* is high. The outputs of all the transmission gates form the output of the MUX. By using transmission gates it is possible to connect several inputs to one output without the inputs affecting one another.

To prevent the influence of a large output load on the clock signals, a NAND gate is placed in front of each transmission gate and at the output an inverter is inserted. This inverter is a different inverter than that used in the PDL and especially prepared for the output.

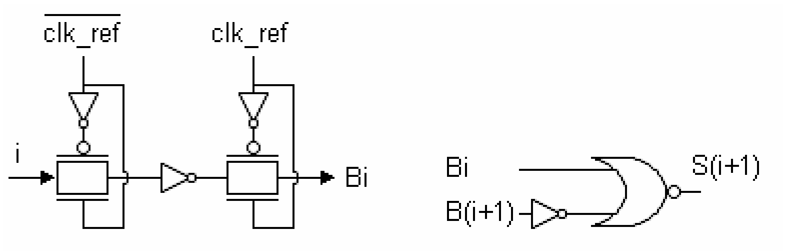


(a) (b)

Figure 3.15: *(a) Schematic representation of the MUX and (b) its logic signals.*

In Figure 3.15 a schematic representation of the MUX is seen. The construction of the selection signals can be divided into two parts. The first part compares the logic value versus the time of the delayed clocked signals just after the rising edge of the reference clock.

A schematic representation of the SEL part of the structure is seen Figure 3.16 (a). Node Bi takes the complementary value of the delayed clock signal at i just after the rising edge of the reference clock. In Figure 3.16b, the scheme of the NOR operation is shown which produces the selection signals *S*i from the nodes *Bi*. If a delayed clock signal *clk_del*, applied to *i*, has a high value just after the rising edge of the reference clock *clk_ref*, the corresponding node *Bi* has a low value. There will be only one value of *i* for which *Bi* has a low value and *B(i+1)* has a high value.



(a) (b)

Figure 3.16: *Scheme of the selection part, (a) charging nodes Bi and (b) NOR-operation.*

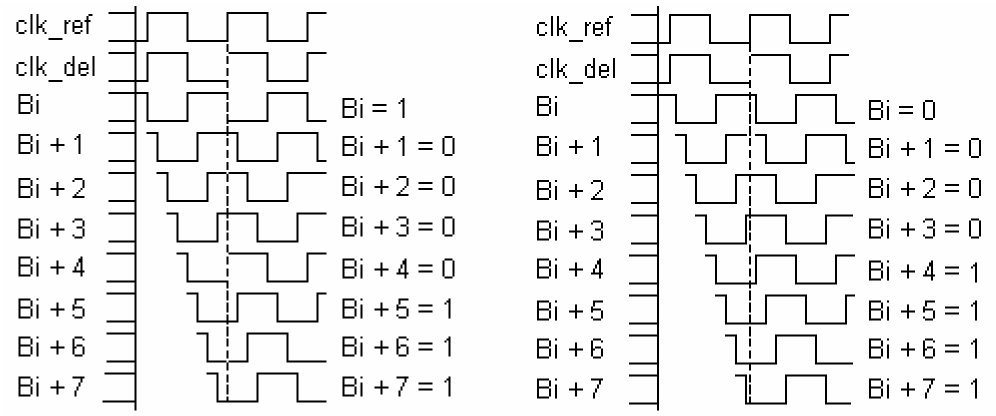
From Figure 3.17, where simple timing diagrams are shown, it can be seen that if *Bi=0* and *B(i+1) =1* then the selected delayed clock signal *clk_del* is *(i+1)*. This corresponds to a NOR-operation:

$$\mathrm{Bi} + \overline{\mathrm{B}(\mathrm{i}+1)} = \mathrm{S}(\mathrm{i}+1). \qquad \mathbf{(3.3)}$$

In the article by Metra [25], a different NOR-operation was used:

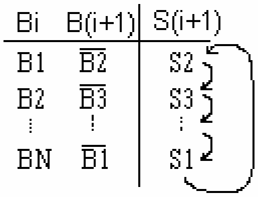
$$\mathrm{Bi} + \overline{\mathrm{B}(\mathrm{i}+1)} = \mathrm{Si} \qquad \mathbf{(3.4)}$$

This is only valid in case the skew of the input clock signal can exactly be compensated for. It actually means that there is no clock-skew. Normally if one wants to implement clock-skew correction then one can be certain that there will always be some clock-skew present. The clock-skew must therefore be corrected to an acceptable level so that it does not influence the testing technique.



(a) (b)

Figure 3.17*: Timing diagrams of delayed clock signals. (a) NOR operation:* $Bi + \overline{B(i+1)} = S(i+1)$ *(b) NOR operation:* $Bi + \overline{B(i+1)} = Si$.



Bi	B(i+1)	S(i+1)
B1	$\overline{B2}$	S2
B2	$\overline{B3}$	S3
⋮	⋮	⋮
BN	$\overline{B1}$	S1

Figure 3.18*: The shifting of selection signals Si in the NOR-operation. (See Figure 3.16(b))*

By shifting the selection signals, as can be seen in Figure 3.18, it is possible to change the delay of the output clock signal *clk_out*. It is thus possible to provide the output clock signal with a required delay that is within the restrictions of the architecture and hardware.

For the previously explained selection mechanism it has been assumed that between two successive output signals of the PDL there are two inverters. If only one inverter is assumed between two output signals, the second part of the selection will be different. Figure 3.19 shows the simple timing diagrams in case only one inverter is placed between two delayed clock signals.

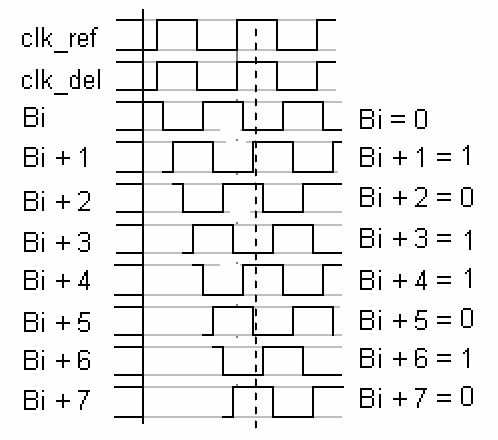


Figure 3.19*: Simple timing diagrams of delayed clock signals, using one inverter between two delayed clock signals.*

From the above timing diagram it can be seen that for only one signal *i*, nodes *Bi* and *B(i+1)* have both a high value. In this case a NAND gate and an inverter construct the selection signals. If the delay of the calibration parts itself is not taken into account, the NAND-operation should be:

$$\mathrm{Bi} + \mathrm{B}(\mathrm{i} + 1) = \overline{\mathrm{S}(\mathrm{i} + 1)} \qquad \textbf{(3.5)}$$

Adjusting the delay of the output clock signal is not as easy as that of the NOR-operation. The selection signal cannot be shifted that easily, because two delayed clock signals next to each other are not just delayed but also inverted. If using this last method, the clock-skew tolerance is smaller because it is determined by only one inverter and not two. But for the same reference clock signal there will be more selection signals and thus more transistors are used. This method that is explained above takes three clock cycles to produce a corrected signal while other methods, like a PLL or DLL, may take a very long time (e.g. hundreds of clock periods) [26] and it does not need feedback. The number of replicas required depends on the delay of the PDL elements and the period of the clock signals. The multiplexer (MUX) ensures that the correct signal is routed to the output.

Our design which has been described in detail in the previous part has been verified by simulation with HSPICE. All the timing analyses have been carried out at 50 % of the high value of the reference clock signal. The reference clock signal that is used has a frequency of 40 MHz with a rise and fall time of 0.5 ns. The reference clock signal is shown in Figure 3.20.

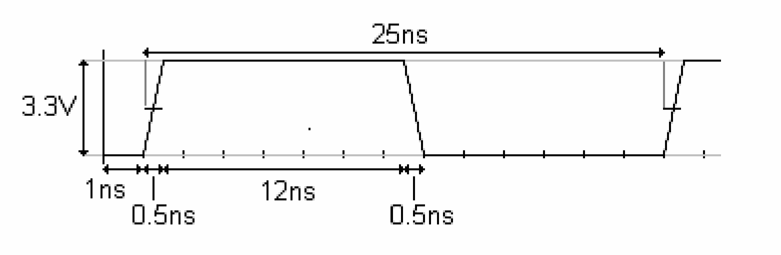


Figure 3.20: *The reference clock signal clk_ref.*

A capacitive output load of 200 pF was used. If the MUX only consists of transmission gates, the load influences the clock signals. The effect of the load on the delayed clock signal (output PDL) can easily be seen if no NAND gates are used. By placing a NAND in front of the transmission gates, the delayed clock signals are not much influenced by the load. The average delay of two inverters in our process is 0.6 ns; this was used to calculate that 83 inverters are needed in the PDL. To determine the real number of required inverters, a simulation of the design with a PDL with 83 inverters has been performed. The delayed clock that showed a delay just smaller than *Tref* indicated the number of required inverters. This delay is checked again after removing excessive inverters. The number of inverters in the PDL for the used reference clock is 75. The behaviour of the design has been verified by simulation with different input (skewed) clock signals. First, the selection signal *S1* has a high value; after ~52 ns *S1* becomes low and *S75* high. The change in the active selection signal is due to a (small) change in the delay of the delayed clocks. This change of the active selection signal indicates that a variation of delay is immediately corrected. Input signals where taken with: a) no skew b) 6 ns skew, c) 12 ns skew and d) 18 ns skew. The output clock signals show that the delay of the calibration part itself is also compensated for. Here the following NOR-operations were used:

$$Bi + \overline{B(i+1)} = S(i-1) \qquad \textbf{(3.6)}$$

More simulations have been performed with input signals with a different skew of around 12 ns; it showed a correct selection of delayed clocks. Simulations with loads of 100 pF and 300 pF showed not any remarkable difference in the output clock signals.

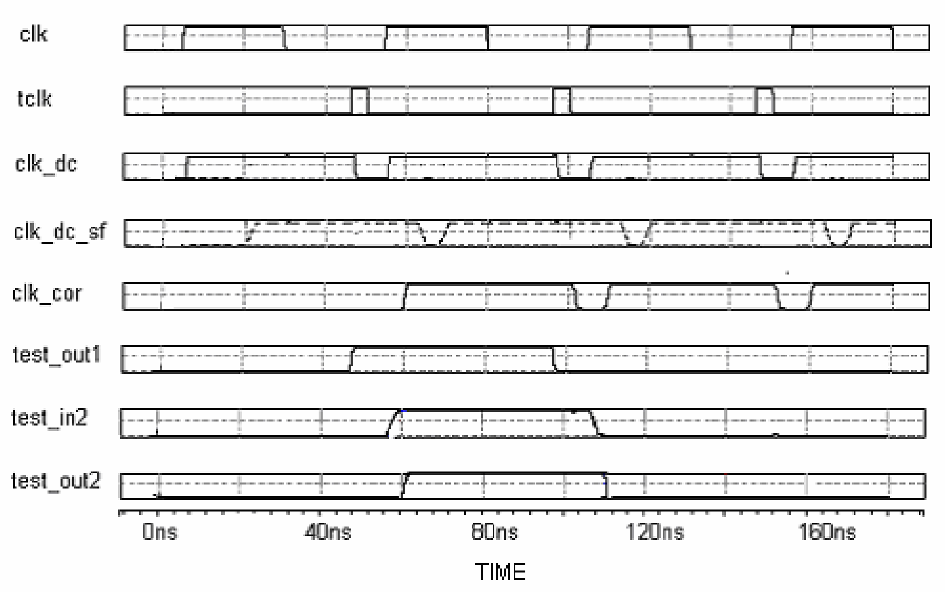


***Figure 3.21**: HSPICE simulation results of fault- free operation with skew correction in the DfDT structure of Figure 3.6 (TSMC 0.35 μm Technology). In the simulation 32% skew was assumed.*

As the maximum delay of the PDL can be influenced by its load and input (skewed) clock signal, the maximum delay should be checked for each result. If this maximum delay is not between *Tref*-t_a and *Tref*, either the selection of a delayed clock signal can go wrong or the clock-skew in the output clock signal can be larger than the allowed clock-skew tolerance. Simulations of the design with different load capacitances showed that as long as the NAND and output inverters are used in the MUX the output is not much affected by its load. For the skew correction time a value of three clock periods has been used although most of the time two clock periods suffices. The maximum clock-skew is 0.8 ns, but often 0.6 ns can be obtained. Simulation showed that with this new clock-skew design a fast correction of clock-skew is possible. The selection of delayed clock signals makes it easy not only to compensate clock-skew but also to produce clock signals with any delay one wants. The design with the NAND-operation produces a smaller clock-skew in the output clock signal; a disadvantage is the use of more transistors and a more difficult change of delay in the output clock signal.

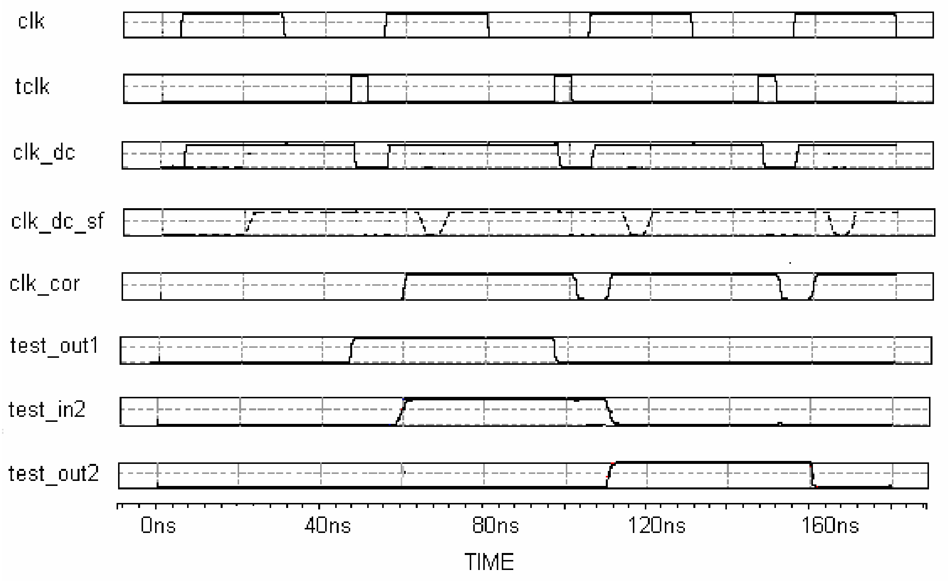


Figure 3.22: *HSPICE simulation results with skew correction in the DfDT structure of Figure 3.6 (TSMC 0.35 μm technology). In the simulation 32% skew of the clock period was assumed and a delay of 620 ps was inserted in the critical path.*

Simulations were carried out in HSPICE with this skew correction included in our DfDT architecture and the results are shown in Figure 3.21. The constant clock-skew is 4.2 % of the clock period and a clock-skew up to 32 % (skew clock signal is 0.32*clock period later) of the clock period could still be handled successfully. The result in Figure 3.22 shows that the skew-reduction circuit is operating successfully and a constant skew clock signal is provided to the normal scan flip-flop in the DfDT architecture. This HSPICE simulation therefore indicates the viability of our strategy to compensate for the possible clock-skew induced by change in some of the earlier mentioned variables.

3.4 Susceptibility of the DfDT structure to Variations

The DfDT structure in Figure 3.6 has been designed to detect small delay faults, of which the minimum detectable delay size is determined by design and processing parameters. Changes in process and application-induced parameters can hence lead to detection problems of existing delay faults. When assuming variations in process and application-induced parameters it is important to define them accurately. In the most preferred scenario, a process is configured during its development to target electrical parameters required by the specific design style intended to be used. However, designers are faced with the challenge of providing a product that must be portable among batches with a variety of parameter ranges. This is normally referred to that a design should be robust. In industry, one has to show that the design works within the occurring process variations. In this case more attention must be paid to tolerances of the variation in process conditions. It is required to look at the main contributors to performance variation, which will affect the critical components in our DfDT structure.

3.4.1 The Influence of Process Variations

Looking at recent trends in process variations, the following variables are of importance to our structure. The *gate-oxide thickness* (*tox*), having a very small dimension in the fabrication of CMOS devices, has a first–order impact on device performance. It directly affects the threshold voltage and the gain factor ß. A change in *threshold voltage* (*Vth*) can have different effects on the electrical behaviour of the chip. For instance a high threshold voltage will result in slower operation of transistors. In contrast, a low threshold voltage results in somewhat faster circuits. In the case of having a low threshold voltage, also the input noise margin may become critical. One usually strives to minimize the channel length to achieve the best performance since both “on” resistance and gate capacitance are largely determined by this parameter. The channel width (*W*) variation has a second-order impact on the performance relative to that of the *channel length* (*L)* variation. In general, the effect of process variations manifests itself first in the most critical paths in the design, being those with maximum delays. In order to investigate the robustness of the delay-detection architecture to these variations, it was decided to look at the variation in gate-oxide thickness, threshold voltage and the relation between channel length and width (*W/L*). This was done in order to establish the susceptibility of the circuit to these process variations. Although other parameters exist, like e.g. the electron mobility, we only looked at these specific ones as they are the most important parameters for our circuits.

Extensive Monte-Carlo HSPICE simulations were carried out to analyze the effect of the different variables on the Programmable Delay Line (PDL) and Duty Cycle Controller (DCC), which are the most critical parts of the DfDT architecture. The PDL is also part of the DCC and therefore the analysis was primarily carried out with respect to the PDL. The delay times of the PDL (64-stages) as function of Vth, tox and W/L were determined subsequently. The results with a 10 % variation in parameter value of Vth, tox and W/L, assuming a relation of changes in terms of a Gaussian distribution, are shown in Figure 3.23, Figure 3.24 and Figure 3.25 respectively.

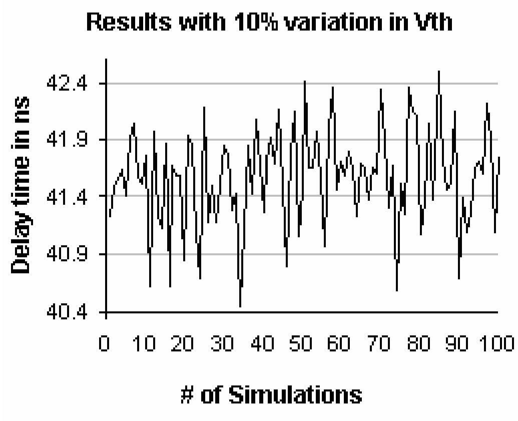


Figure 3.23: *Monte Carlo HSPICE simulation results of the delay time of the PDL with a 10 % change in threshold voltage. (Vth = 0.51 V ± 0.051 V for NMOS and Vth = -0.69 V ± 0.069 V for PMOS.)*

A Gaussian distribution was used because the variation of parameters is given in literature [21] as 2 % to 5 % with the exception of a maximum variation of 10 %. The vertical axis shows the propagation delay time of the PDL. Hundred simulation runs were carried out (42 minutes CPU time on an

HP9000/J5600). The simulations show that a 10% variation in *Vth* results in a 2,1 % change in delay time, the 10% variation in *tox* gives a 0,9 % change and the *W/L* variation is resulting in a 1,4 % change in delay time.

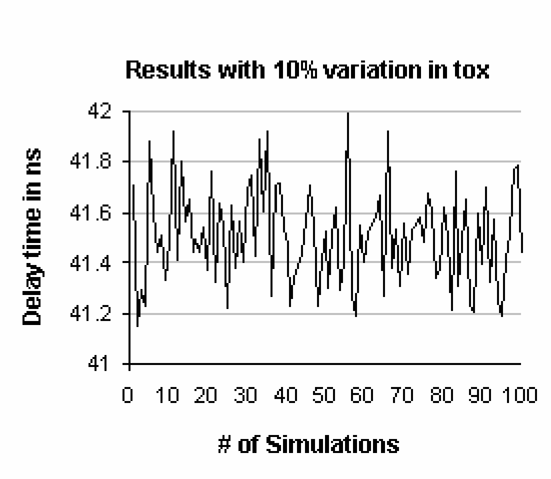


Figure 3.24**: Monte Carlo HSPICE simulation results of the delay time of the PDL with a 10 % change in gate oxide thickness. *(tox = 7.7nm ± 0,077nm)*

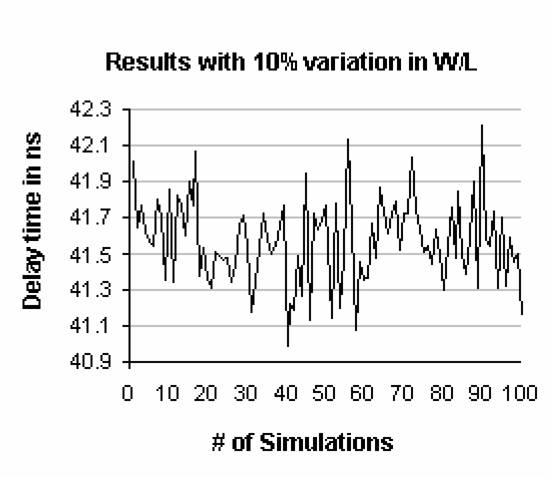


***Figure 3.25**: Monte Carlo HSPICE simulation results of the delay time of the PDL with 10 % change in the W/L relation. (W/L = 0.6 ± 0.06)*

3.4.2 The Influence of Application-Induced Variations

Application-induced factors are defined as factors that are not part of the manufacturing process, but depend on the environmental conditions. When looking at application-induced variations, there are two major contributors being the supply voltage (*Vdd*) and temperature (*T*). Upgrading the process capabilities by means of raising the supply voltage and lowering the temperature can increase the speed of a CMOS circuit [19].

Changing the external power supply varies the overall performance of the chip. Local voltage drops in the internal power-distribution system will create variations in delay across the chip. The distribution of power to an active circuit changes the tolerances of delay and noise margin in all circuits.

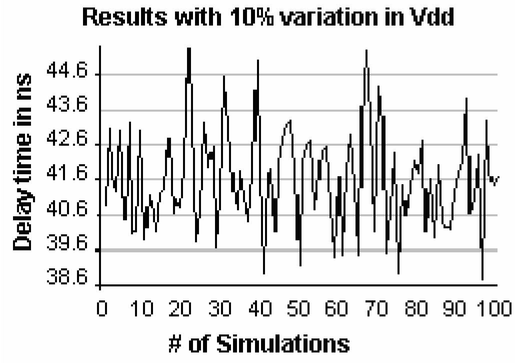


Figure 3.26*: Monte Carlo HSPICE simulation results of the delay time of the PDL with a 10 % change in the supply voltage Vdd. (Vdd = 3.3 V ± 0.33 V)*

The power grid can induce both AC and DC voltage variations across the chip. In Figure 3.26, the effect of a 10 % change in supply voltage is clearly visible in the Monte Carlo simulations that were carried out on the critical element of the DfDT architecture, being the PDL. Since the movement of electrons and holes decrease with an increase in temperature and the current in a CMOS device will decrease, the circuits that operate at a high temperature will be slower than circuits operating at low temperature. Also, a reduction in temperature will reduce the resistance of the metal interconnects. Therefore, a reduced temperature can be used to one's advantage up to certain limits. The temperature gradient will also result in additional skew in the clocking distribution, most notably due to the variation in the change of the interconnect resistance. HSPICE Monte-Carlo simulations were done to investigate the percentage change of the delay time of the PDL under a 25 % variation in ambient temperature. The Monte-Carlo analysis shown in Figure 3.27 depicts the influence of a six degree Celsius (25 %) variation in temperature on the delay time of the PDL [27].

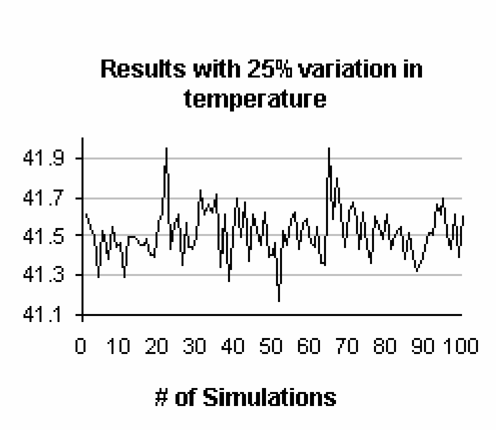


***Figure 3.27**: HSPICE Monte-Carlo simulation results of the delay time of the PDL with a 25 % variation in ambient temperature. (T = 24 ^{o}C ± 6 ^{o}C [27])*

From the results given in Table 3.1, showing the variation in delay times from the HSPICE Monte-Carlo simulations, it becomes clear that the major factor in delay-time variations is *Vdd*. Hence, an additional support circuit will be suggested to circumvent the influence of an unstable supply voltage and more likely, local IR drops [28]. The percentual variation as shown in Table 3.1 can therefore be a combination of more than one parameter under certain conditions. It is clear from Table 3.1 that the variation in supply voltage has a major effect on the delay time of the system [29]. It is therefore essential that this effect of a varying supply voltage must be looked at. It is however also very important to investigate carefully the design of power-lines keeping in mind the effect IR drops can have on local power supply.

***Table 3.1**: Delay-time variation as result of process and application-induced variations as percentage of the clock period.*

	Vth	tox	W	L	Vdd	T
Δ delay	2.1	0.9	1.4	1.4	7.2	1.8

3.4.3 Introducing a Stable Voltage Reference

In order to reduce the noise on the supply distribution, a decoupling capacitance can be added to the power supply. However, the decoupling capacitance must be electrically close to the source of the noise to be able to suppress it. On-board decoupling capacitances have little effect on the internal localized supply noise since the inductance and the series resistance will shield and delay the charge that the decoupling capacitance could provide [26]. The increased number of on-chip (simultaneously) switching circuits can cause high voltage drops due to fast current fluctuations. A solution might be to integrate decoupling capacitors on the chip, which reduce the on-chip voltage fluctuations. However the introduction of capacitances on the power lines in chips is not common practice. The first thought is that the values of the capacitors will be unacceptably high (in the region of 250 nF) in terms of area needed to implement the capacitors.

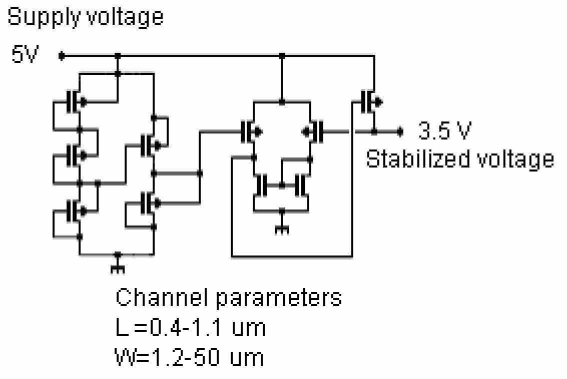


***Figure 3.28**: The voltage-reference circuit as used in HSPICE simulations.*

One possible approach is to look at regulated voltage-reference circuits. While the magnitude of a reference voltage is often not significant, because it can be compensated by system design, it is important for the reference to exhibit long-term stability and insensitivity to temperature variations. In Figure 3.28, the used voltage reference is shown. It is based on a circuit described in [30] and occupies 45 µm^2 in our 0.35 µm CMOS process [19]. The circuit was adapted to the CMOS process indicated above and the simulation results show its ability in present technologies.

The simulation result of our PDL, including the voltage reference circuit, is shown in Figure 3.29. The external supply voltage (Vsup) was taken as 5 V with a 10 % variation. The Δ delay has now been reduced from 7.2 % to 1.9 %. This result has proven that if the supply voltage can be kept stable, then the Δ delay can be controlled within acceptable limits. When one uses the complete DfDT structure one has to ensure that the voltage stabilizing circuit that is used is able to provide the necessary power for the complete DfDT structure.

The voltage-reference circuit that we used was developed to be able to provide the necessary power for the complete DfDT structure. Once the structure is implemented in silicon it will be possible to obtain physical results.

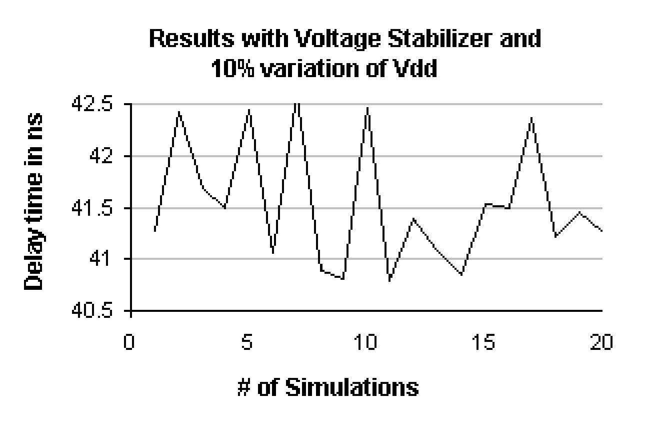


***Figure 3.29**: HSPICE simulation results with the voltage stabilizing circuit included in the PDL circuit. An initial supply voltage variation of 10 % was assumed. (Vdd = 5 V ± 0.5 V)*

3.5 Low-Speed BIST for DfDT

It is possible to test a scan circuit at slow speed and yet verify its high-speed performance. This is done by generating skewed clocks to allow a propagation time between the flip-flops that are smaller than the test-clock period. Scan flip-flops are modified to be able to handle delay test and require more hardware than the normal scan flip-flop. As a result of possible speed limitations of the ATE, many high-speed digital components (VLSI chips, printed circuit boards and multi-chip modules) are tested at clock rates that are lower than the one stated in the specifications.

As the ever changing technology develops, it is nowadays even possible to do high-speed scan [31]. This is a major change in the scan technology and will have to be looked at in future work.

3.5.1 Different BIST Approaches

Until the last decade test engineers have assumed that testing of logic circuits is done by externally applying the test inputs and comparing the results with the expected simulated behaviour of the circuit. This requires connecting external equipment to the circuit under test. The quest was then to investigate if it will be possible to incorporate the testing capability within the circuit itself so that no or less external equipment is needed. Such built-in capability would allow the circuit to be self-testable. This testing scheme that provides the on-chip testing capabilities has been referred to as BIST.

In the past, a number of methods with regard to BIST for detecting delay faults have been published. Most efforts have been confined to generating the required two-pattern tests for detecting delay faults as was stated in Chapter 2. The "adjacent testing" approach uses test pairs which only differ at a single bit position. As a result, only simple hardware is required [32]. A more sophisticated technique starts with a predetermined set of test pairs. In this case, hardware is designed which generates sequences in which the test pairs (initialisation and propagation vector) are embedded [1]. The key elements in these generators are special Linear Feedback Shift Registers (LFSR) [33] or Multiple Input Signature Registers (MISR) [34]. Although rarely discussed, the evaluation of the responses can be carried out in a similar way as in "conventional" digital BIST approaches using signature analysers. A completely different approach makes use of controlled oscillation of the block involved [35]. However, the problem is measuring the high oscillation frequency either internally or externally. In our approach, additional hardware has been designed to carry out the BIST at low clock speeds [36-37].

3.5.2 Using the PDL and DCC in a BIST Environment for Detecting Delay Faults

In Figure 3.30, the set-up of our suggested BIST architecture is shown. With this option the two-pattern generation and response evaluation is done externally by means of a slow-speed (< 50 MHz) ATE. This option is referred to as partial-BIST as already explained in Chapter 2.

The full-BIST option employs LFSRs for two-pattern generation and response evaluation, the key elements in the low-speed approach are the Programmable Delay Lines (PDL) and Duty-Cycle Control (DCC) blocks. The extra hardware needed for this option is shown in Figure 3.31.

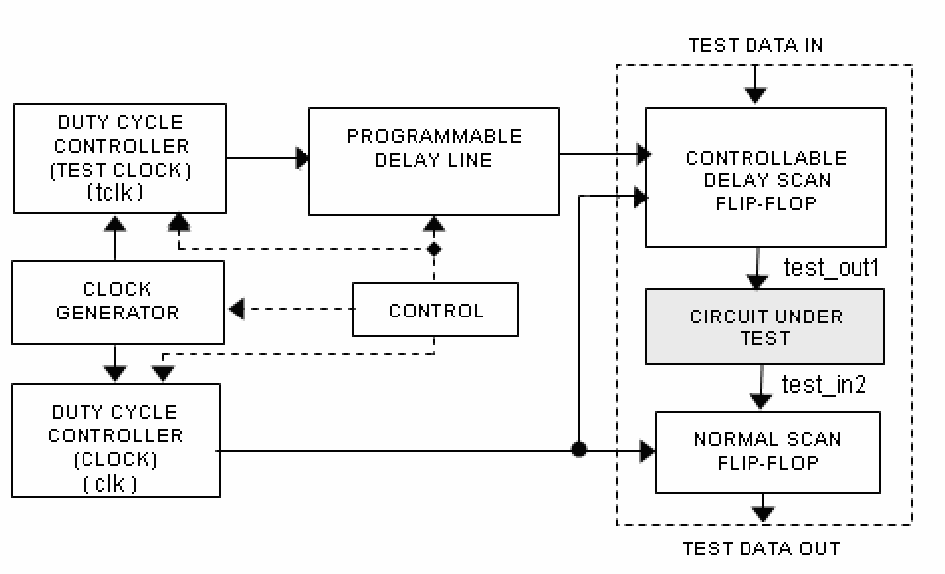


Figure 3.30: *Overall set-up of the BIST architecture for the partial BIST option.*

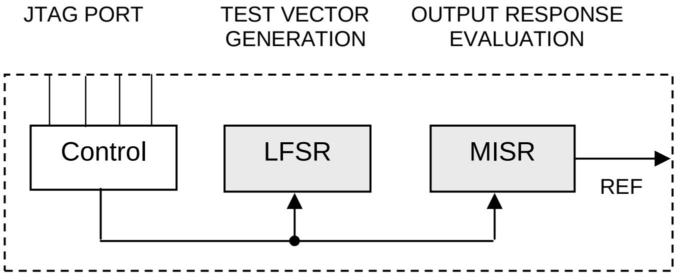


Figure 3.31: *Extra hardware on chip for implementing the full-BIST option.*

3.5.3 Simulation Results from the PDL and DCC

The previously described parts have been combined into the BIST architecture as depicted in Figure 3.30.

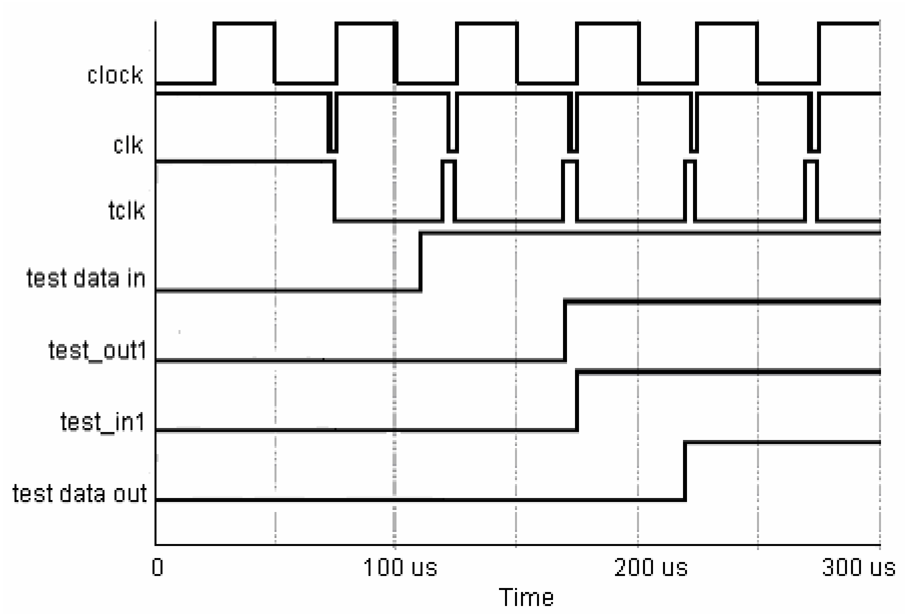


Figure 3.32: *Simulation of the BIST architecture in the case of low-speed testing and fault-free behaviour. Normal delay of the circuit: 4750 ps.*

For the sake of simplicity and insight of the structure, the control parts, LFSR and MISR for signal generation and evaluation have been omitted in the simulations that have been carried out. HSPICE circuit simulations were carried out to verify the behaviour of the PDL circuit. The minimal timing resolution of the design was set to around 500 ps. The maximum obtainable delay between the input and output of the PDL becomes in this case 63.5 ns. The values obtained from the above simulations were used in the overall BIST simulation.

Figure 3.32 shows the low-speed BIST in the fault-free case. The critical path has a delay of 4750 ps. Next, in Figure 3.33, the low-speed operation is simulated with values determined from simulations for the required frequency, duty-cycles and delay. For this delay-fault of 600 ps, the frequency are 20 MHz, duty cycles of clock and test clock 95.8 % and 10.8 % respectively, and a delay of 45 ns between clock and test clock is assumed. As can be seen from the simulations, the delay-fault is detected in the last case. The simulations show that our concept of low-speed BIST works.

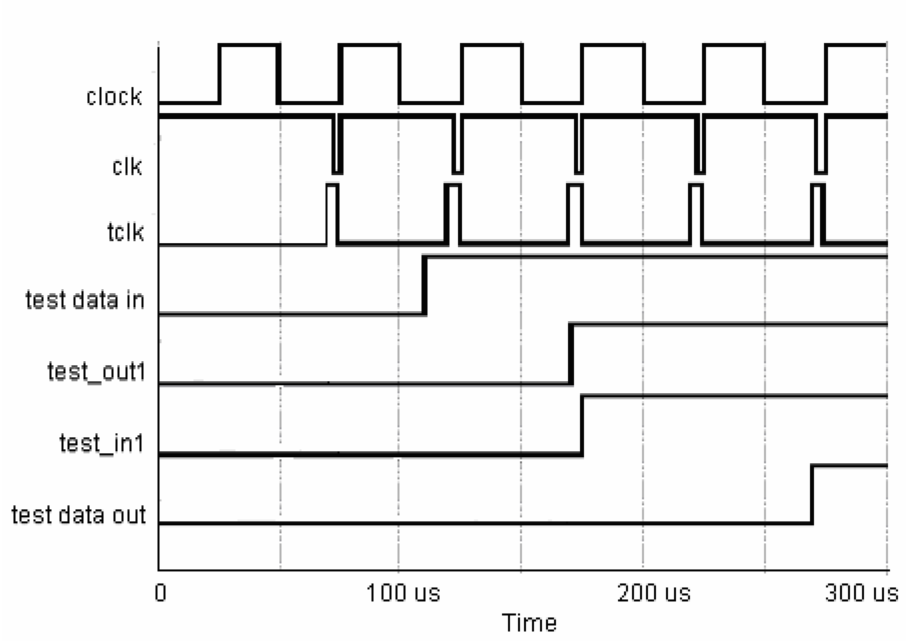


Figure 3.33: *Simulation of the BIST structure in the low-speed mode (20 MHz) detecting the delay-fault. The delay after adding a 600 ps delay-fault: 5350 ps.*

3.6 Conclusions

While virtually every step in the chip fabrication process introduces tolerances, a subset thereof is capable of creating substantial changes in the electrical response of circuits. Due to the critical timing necessary when detecting small delay faults it is crucial to have precise timing in the structure.

A Controlled Delay Scan Flip-Flop (CDSFF) has been proposed for detecting small delay faults in digital high-speed circuits. The manipulation of delay between and duty cycles of the original clock and an additional test clock as well as the use of controlled delay scan flip-flops were crucial in this method.

The extra components to implement our complete structure, including the skew-reduction and voltage-reference circuits consist of 3846 CMOS transistors for which a chip area of 0.008mm^2 is needed in our 0.35 μm CMOS process. The area overhead of each component is shown in Table 3.2. The increase in chip area is acceptable looking at the complexities envisioned.

Table 3.2*: Area overhead needed for DfDT components in our process.*

BIST Part	Silicon area
PDL and DCC	0.005
Clock-skew correction & Voltage reference circuits	0.003
TOTAL	0.008 mm^2

The complete DfDT structure was shown as part of a BIST structure for detecting small delay faults in digital high-speed circuits. The method avoids the requirement of an expensive high-speed tester. Depending on the application, a full BIST or a partial BIST approach employing a low-speed tester can be used. Simulations of the DfDT parts, as well as the overall DfDT architecture indicate the feasibility of the methods. The DfDT structure will have a very small impact on the chip performance because it is only active during the test sequence.

The susceptibility of our DfDT architecture to process and application-induced variations was shown by means of Monte Carlo analysis. The variation in Vdd was identified as a possible problem area for our structure. A practical solution was presented to eliminate this factor influencing our structure and verified by means of simulation. Inaccuracy in the position of the clock timing edges can be due to process variations in the clock generation circuit. The timing inaccuracy causes undesirable clocks skew in the DfDT structure. A clock-skew correction circuit was added to the DfDT structure to be able to solve the clock-skew problem. Simulations showed that the circuitry can handle unwanted clock-skew of up to 32% of the clock period with success.

We have therefore not only showed the viability of our DfDT structure and possible BIST architecture but also enhanced our structure to be able to provide constant response in an unstable environment. The main factors that were influencing the environment were the supply voltage and clock-skew which were both compensated for in the enhancements.

3.7 References

[1] R.C. Aitken,” Designs are Changing – Can Test Keep up?”, *An Invited address to the International Test Conference (ITC)*, 2000. Atlantic City, USA.

[2] Semiconductor Industry Association. *International Technology Roadmap for Semiconductors (ITRS)*. SIA, Austin, December 2003. (Available via http://public.itrs.net/).

[3] R.D. Isaac, “The future of CMOS Technology”, *IBM Journal of Research Development,* Vol. 44, No. 3, May 2000, pp 369-378.

[4] K.T. Lee and J.A. Abraham,” Critical Path Identification and Delay Tests of Dynamic Circuits”, *Proceedings of International Test Conference (ITC)*, 1999, Atlantic City, USA, pp. 421-430.

[5] V. D. Agrawal and T. J. Chakraborty, "High-Performance Circuit Testing with Slow-Speed Testers", *Proceedings of International Test Conference (ITC)*, 1995, Washington, DC, USA, pp. 302-310.

[6] G. L. Smith, “Model for delay faults based upon paths”, *Proceedings of International Test Conference (ITC)*, 1985, Philadelphia, USA, pp. 342-349.

[7] M. Sharma and J.H. Patel, “Testing of critical paths for delay faults.” *Proceedings of International Test Conference (ITC)*, 2001, Baltimore, USA, pp. 634 – 641.

[8] K.-T. Cheng, H.-C. Chen. “Classification and Identification of Non robust Untestable Path Delay Faults”, *lEEE Transactions on Computer-Aided Design, CAD-15*, 1996, pp.845-853.

[9] T.Xiao, and M. Marek-Sadowska, “Functional Correlation Analysis in Crosstalk Induced Critical Path Identification”, *Proceedings of Design Automation Conference (DAC)*, 2001, Las Vegas, USA, pp. 653-656.

[10] W. Li, S. M. Reddy, and S. K. Sahni, “On path selection in combinational logic circuits”, *IEEE Transactions on Computer-Aided Design of Integrated Circuits & System*s, vol. 8, no. 1, January 1989, pp. 56–63.

[11] S. Tani, M. Teramoto, T. Fukazawa, and K. Matsuhiro, “Efficient path selection for delay testing based on path clustering”, *Journal of Electronic Testing: Theory & Applications (Jetta),* vol. 15, no. 1, pp. 75–85, 1999.

[12] J. Liou, K. Cheng, and D. Mukherjee, “Path selection for delay testing of deep sub-micron devices using statistical performance sensitivity analysis”, *Proceedings of VLSI Test Symposiu*m, 2000, Montreal, Canada, pp. 97-104.

[13] M. Luong and D.M.H Walker, “Test generation for global delay faults”, *Proceedings of International Test Conference (ITC),* 1996, Washington, DC, USA, pp. 433–442.

[14] K.T. Lee and J.A. Abraham,” Critical Path Identification and Delay Tests of Dynamic Circuits”, *Proceedings of International Test Conference (ITC),* 1999, Atlantic City, USA, pp. 421-430.

[15] H. Speek and H.G. Kerkhoff, "Bridging the Testing Speed Gap: Design for Delay Testability*", Proceedings of IEEE European Test Workshop (ETW),* 2000, Cascais, Portugal, pp.15-20.

[16] V.D. Agrawal and T.J. Chakraborty, “High-Performance Circuit Testing with Slow-Speed Testers”, *Proceedings of International Test Conference (ITC),* 1995, Washington, DC, USA, pp. 302-310.

[17] H.J. Vermaak and H.G. Kerkhoff, “Reducing the Susceptibility of Design-for- Delay-Testability Structures to Process- and Application-Induced Variations“, *Proceedings of IEEE European Test Workshop (ETW),* May 2001, Stockholm, Sweden, pp.35-41

[18] H.G. Kerkhoff, K. van Nee and H. Speek, "DfT for Delay-Fault Diagnosis in Digital High-speed ICs", *Proceedings of IEEE European Test Workshop (ETW)*, Constance, Germany, 1999, pp. 9-10.

[19] http://www.mosis.org: MOSIS TSMC 0.35um CMOS process data.

[20] M. Shoji, ”Elimination of Process-Dependent Clock Skew in CMOS VLSI”, *IEEE Journal of Solid-State Circuits,* Vol. Sc-21, No. 5, October 1986, pp 875-880.

[21] K. Bernstein et al., ”High Speed CMOS Design Styles,” *IBM Microelectronics*, 1998, pp.2-272.

[22] K. Yip, “Clock tree distribution design in deep sub-micron ASIC design”, *M.Sc. Thesis,* University of Waterloo, Canada, 1998.

[23] http://www.model.com/products/se.asp: MODELSIM SE.

[24] M. Nekili, Y. Savaria and G. Bois, “Design of clock distribution networks in the presence of process variations”, *Proceedings of IEEE Symposium on VLSI,* Los Amamitos, CA, USA, 1998, pp. 95-102.

[25] C. Metra et al., “Self-Checking Scheme for Very Fast Clocks’ Skew Correction”, *Proceedings of the International Test Conference (ITC),* Atlantic City, USA, 1999, pp. 652-661.

[26] A. Efendovich, Y. Afek, C. Sella, and Z. Bikowsky, “Multi-frequency Zero-Jitter Delay-Locked Loop”, *IEEE Journal of Solid-State Circuits,* January 1994, Vol. 29, pp 67-70.

[27] http://vlsi.wpi.edu/technology/AMS/ams.html: AMS 0.8µm CMOS process data.

[28] P.Gupta and A.B. Kahng, “ Manufacturing-Aware Physical Design”, *Proceedings of International Conference on Computer-Aided Design*, November 2003, San Jose, USA, pp. 681-687.

[29] H.J. Vermaak and H.G. Kerkhoff, "The Ability of Design-for-Delay-Testability Structures to function in an environment with Process- and Application-Induced Variation”, *Proceedings of the Dutch Technology Foundation Conference on Circuits, Systems and Signal Processing (ProRisc),* November 2001, Eindhoven, The Netherlands, pp. 694-699.

[30] K.D. Wagner and E. McCluskey, “ Effect of Supply Voltage on Circuit Propagation Delay and Test Application,” *Proceedings of ICCAD,* Las Vegas, USA, 1985, pp. 42-44.

[31] K. Filliter, “Boundary-Scan Testing of High-Speed Differential Signals”, *Proceedings of International Test Conference (ITC)*, 2001, Baltimore, USA, p. 1187.

[32] S. Thompson, P. Packan and M. Bohr, ”MOS Scaling: Transistor Challenges for the 21st Century”, *Intel Technology Journal*, Q3’, 1998, pp 1-19.

[33] N. Mukherjee, T.J. Chakraborty, and S. Bhawmik, "A BIST Scheme for the Detection of Path-Delay Faults*," Proceedings of International Test Conference (ITC)*, 1998, Washington, DC, USA, pp. 422-431.

[34] K.T. Lee and J.A. Abraham,” Critical Path Identification and Delay Tests of Dynamic Circuits”, *Proceedings of International Test Conference (ITC)*, 1999, Atlantic City, USA, pp. 421-430.

[35] M. Favalli, P. Olivo, M. Damiani and B. Ricco, ”Novel Design for Testability Schemes for CMOS IC’s*”, IEEE Journal of Solid-State Circuits,* Vol. 25, No. 5, October 1990, pp 1239-1246.

[36] H. Speek, H.G. Kerkhoff, M. Shashaani and M. Sachdev, “A Low-speed BIST Framework for High-Performance Circuit Testing”, *Proceedings of VTS*, 2000, Montreal, Canada, pp. 349-355.

[37] M. Shashaani and M. Sachdev, "A DFT Technique for High Performance Circuit Testing," *Proceedings of International Test Conference (ITC)*, 1999, Atlantic City, USA, pp. 276-285.

Chapter 4

Delay-Fault Detection by Pulsed Supply Voltage

Electro Magnetic Compatibility (EMC) issues, together with process variations in DSM technologies, lead to timing-related defects which can be modelled as delay faults. Delay faults cause digital integrated circuits to malfunction at the desired clock rates and hence violate timing specifications. Testing for these delay faults is therefore becoming critical in DSM CMOS digital circuits. Testing these high-performance digital logic circuits is a difficult and expensive task. As the minimum feature size is decreasing, the gap in speed resolution issues between the ATE and the device is increasing [1]. Therefore devices cannot be tested in the near future at their operational speed, unless proper DfT is provided. A special approach of DfT to test for delay faults in these high-performance digital circuits is Design-for-Delay-Testability (DfDT) structures [2-3]. In this chapter, the fundamental issues with regard to the digital oscillation test method will be investigated, which will form the basis of our new approach to test for delay faults.

4.1 Introduction

Over the past decade, many delay-fault models have been proposed, both in the functional and structural domains (transition, gate, path, segment, etc.) [4-6]. The fault models vary one from another by trading off the test-coverage and computational requirements.

In new DSM technologies, the interconnect delay exceeds the gate delay. The interconnect delay varies as a function of placing-and-routing efficiency and process variations and is not predictable by simulations at the gate level.

However, if one has a logic-gate net list, one can carry out place-and-routing and make an estimation of the expected delay of interconnects.

The above mentioned research includes the development of fault models like the gate-delay fault model, path-delay faults, robust delay faults, non-robust delay faults and hazard-free delay faults [7-8]. For all these approaches, the test setup is rather complicated; for example it requires two sets of latches with precisely controllable test clocks and test-pattern pairs. These approaches are therefore time consuming and occupy a lot of silicon area. One way to overcome this critical test setup is to make use of the digital oscillation-test method.

4.2 The Digital Oscillation Test Environment

The testing task classically consists of verifying that a manufactured device correctly performs its function according to the data-sheet specifications. A faulty device is therefore one with at least one direct parameter outside of the data-sheet specifications. Here, a direct parameter is defined as a device property (e.g. sensitivity, bandwidth…) that must range between a lower and an upper bound. For a given device, numerous parameters are generally specified, some of which are not always easy to measure. Hence, a complete device characterization is a very expensive process. To reduce test costs, an interesting alternative is the Oscillation-based Test Methodology (OTM) [9]. It is based on the reconfiguration of the CUT into an oscillating device and on the measurement of indirect parameters of the oscillating frequency and/or amplitude [10].

In a combinational circuit, the path that has the longest propagation time from a primary input to a primary output, called critical path, determines the operating speed of the circuit. Other paths may have much shorter propagation times and therefore a parametric variation in their delay value may not affect the circuit operating speed unless the changes make their propagation time longer than the critical-path delay. However, even a very small increase in the critical-path delay will slow down the operating speed of the circuit. Also in a sequential circuit, the system is free of timing failures if every combinational path between two memory elements propagates its signal in less time than the interval of the operating system clock. In other words, the input signal of every memory element in the system should have a stable signal before the arrival of the active clock edge. In our approach the OTM will be used as part of the solution to be able to detect delay faults in high-speed digital circuits.

4.2.1 Basic Principles of the Digital Oscillation Test Techniques

The oscillation test method is concerned with sensitizing critical paths and then test them individually for delay faults. Critical paths are those paths that have the longest propagation delay from primary input to primary output. Therefore, the critical path is the most likely path for a delay-fault to cause the circuit to malfunction. The specific critical path that must be tested must first be found and subsequently sensitized. To sensitize a path, all off-path logic values inputs must be set to non-controlling values [11].

In the oscillation test method for digital circuits one starts from the well-known digital ring oscillator in which an oscillation occurs when there is an odd number of inverting elements in the ring. The earlier mentioned sensitized path in the CUT is then incorporated into a ring oscillator to test for delay faults. The oscillation frequency is determined by the propagation delay through the sensitized path. Delay-faults or stuck-at-faults that may alter or stop the oscillations can be detected by observing the frequency of oscillation.

As an example a simple combinational circuit, with a critical path going from input B through gates 1, 2 and 4 to output Q is shown in Figure 4.1 (dark line). In Figure 4.2, the same critical path is sensitized with the output Q connected to the input B to be used in the oscillation-test method.

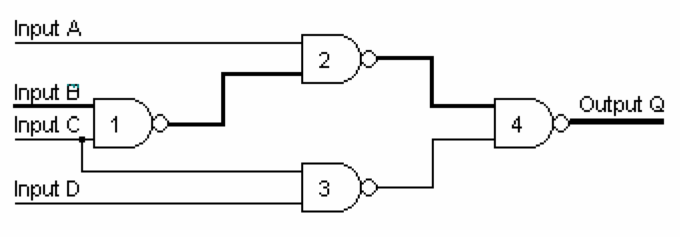


Figure 4.1: *Critical path in a CUT.*

This is shown as the feedback path in Figure 4.2 and will require a multiplexer to enable it for test purposes only. The zeros and ones indicated in Figure 4.2 represent the logic levels required for sensitization of this particular critical path. The X represents a changing state due to the oscillation in the critical path. Therefore, a given delay increase in a critical path may result in a malfunction in the circuit but the same delay increase in another path may not effect the circuit functionality and performance.

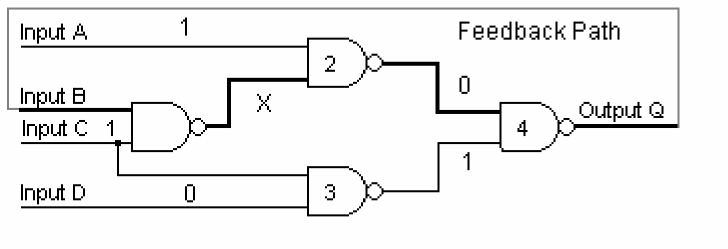


Figure 4.2*: The sensitized critical path in the CUT.*

The oscillating frequency (f_{osc}) is a function of the propagation delay through the critical path. Therefore, the loss or the deviation of the oscillating frequency from its nominal fault-free value can be used to detect delay faults in the circuit.

$$f_{osc} = 1 / (t_{PDg} + t_{PDp)} \qquad \textbf{(4.1)}$$

In which t_{PDg} denotes the sum of the propagation delay of all the gates in the critical path and t_{PDp}, the sum of the propagation delay of all the paths (interconnect wires) in the critical path

A 4-bit digital adder was used to experiment with the digital oscillation of a circuit. As shown in Figure 4.3, for a 4-bit digital adder, the critical path that determines the operational speed of the circuit is the path between the input B1 and the output C4 via the logic gates 1 to and including 7.

The sensitized path is non-inverting and therefore in order to establish an oscillator, the output C4 should be connected to the input B1 via an additional inverter. One must realize the determination of the critical path is not always the longest logic depth. The critical path depends also on the used logic blocks and interconnects. The proper way to establish the critical paths is by using delay calculators, such as the one used in Cadence (CeltIC™ nanometer delay calculator (NDC)). Cadence has introduced SignalStorm® NDC which is a new-generation, fast hierarchical delay calculator for cell-based designs. It accurately computes delays to within a few percent of SPICE for all nets including long interconnects and multi-driven nets such as clock meshes [12].

The delay of the critical path being tested is the main parameter that influences the oscillation frequency. This delay is dependent on the propagation delay of the path as well as any other delays in the path. The causes of delays are numerous and can range from manufacturing flaws to crosstalk and application-induced delays. The maximum operating frequency can be determined by looking at the lowest oscillating frequency of all critical paths through the circuit. The presence of any extra unintentional delay will be enlightened by the change in oscillating frequency.

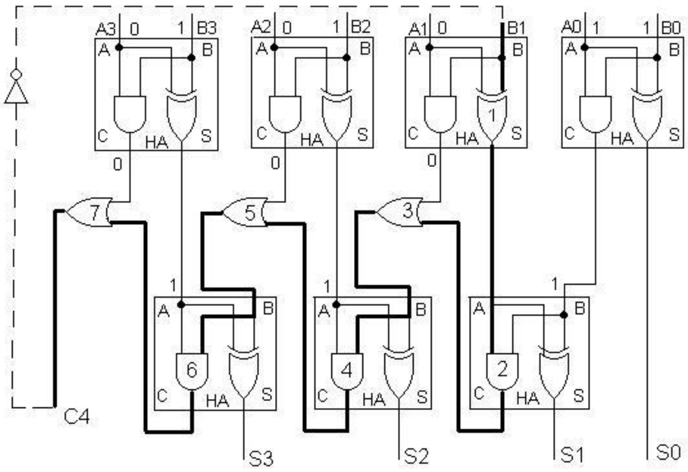


***Figure 4.3**: Critical path in a 4-bit digital adder.*

4.3 The Principles of Low-Voltage Testing

Hao and McCluskey showed in reference [13] that very-low-voltage (VLV) testing can detect delay flaws in CMOS integrated circuits. A delay flaw is a defect that causes a local timing failure but the failure is not sufficiently severe to cause malfunctioning of the IC. A delay-fault is a timing failure that causes the circuit to fail to operate at the designed speed but will be functional at a slower speed. Most studies on testing timing failures focus on the detection of delay faults. However, some timing failures that are embedded in short paths may not cause delay faults at normal operating conditions.

Considering the principles of low-voltage testing (LVT) that some delay faults will not be detected at normal operating conditions and the principles of the OTM it is clear that both try to detect delay faults with special methodologies. This encourages the investigation into combining the two methodologies to develop a test technique that can detect delay faults that would not be detectable at normal operating conditions while using the advantage of the OTM that reduce the test cost as earlier explained.

The supply voltage for VLV testing for detection of delay faults should be set in the region where the changing rate of the propagation delay of a CMOS gate starts to increase significantly. One therefore has to ensure that the circuit is still able to operate correctly when the supply voltage is changed to a low value. The supply voltage is critical, as it is also a parameter that influences many other characteristics of a particular circuit.

The operating speed of a CMOS circuit can be increased by raising the supply voltage (V_{DD}) and lowering the temperature. Changing the supply voltage varies the overall performance of the chip or circuit. Voltage drops on the power-distribution system can result in variations in delay across the chip. The variation of power applied to an active circuit changes the tolerance of the delay and noise margin on the circuits as well as the threshold-voltage (V_T) specifications.

While investigating the variation of the supply voltage, one first has to consider the implications of such variations on the circuit performance. The transistor drive current (I_{DRC}) and therefore circuit-speed performance is proportional to the gate overdrive (V_{OD}) raised to the power n where n is between 1 and 2.

$$V_{OD} = (V_{DD}-V_T) \quad \textbf{(4.2)}$$

$$I_{DRC} \, \alpha \, ((V_{DD}-V_T)^{\,n}). \quad \textbf{(4.3)}$$

If the supply voltage is lowered, the gate overdrive rapidly decreases for DSM devices, thereby strongly degrading device performance if a normal operating temperature is assumed. This decreasing gate overdrive is increasing the sub-threshold leakage, which is important for the CMOS operation and is set by the device threshold voltage. It is therefore clear that one cannot lower the circuit supply voltage without carefully considering the above mentioned issues.

It has been investigated to which extent one can decrease the supply voltage without causing the fault-free circuit to malfunction. In our tests, we made use of SPICE and using the 0.35 µm TSMC CMOS technology from MOSIS [14]. A combinational circuit was used as test vehicle and the supply voltage was lowered and the operation of the circuit verified. This test shows that the optimal minimum value for this specific test circuit is 1.75 Volts. The normal supply voltage of the circuit was 3.3 Volts. This test was also repeated on the same combinational circuit making use of a 0.8 µm technology of AMS [15] with a normal supply voltage of 5 Volts.

The results show that 1.75 Volts was again the optimal minimum value for the supply voltage that constantly ensures correct operation of the circuit. It was therefore decided that 1.75 Volts would be used as the minimum value for our varying supply voltage. This closely matches the result from previous work on VLV testing [12, 16] in which it was stated that the ideal supply-voltage range for VLV testing is between 2 and 2.5 times the threshold voltage (PMOS) of the specific technology that is used.

$$V_{DDMIN} = V_{TH} * 2.5 \qquad \textbf{(4.4)}$$

Our technology has a minimum threshold voltage value of 0.69 Volt and this converts therefore to a supply voltage of 1.73 Volts if Equation (4.4) is considered. This varying supply voltage will only be applied during the testing phase. It is clear that VLV testing can be used and that our results confirm previous work.

4.4 Using a Pulsed Supply Voltage on the Circuit Under Test

4.4.1 The Chosen Test Strategy

Taking the results from VLV testing and the principles of the OTM into account, it was decided to develop a new technique to test for delay-faults by combining the principles of the two methods. The VLV testing suggests using a lower supply voltage and the OTM uses oscillation to detect a delay-fault. Therefore combining the methods gives us the option to change the shape of the supply voltage of the CUT in the test phase. The supply voltage will be taken as a signal with pulse width and period equal to that of the normal fault-free oscillating frequency of the CUT. This is to ensure that the circuit is still operating normally and that the oscillating frequency will still be as with a constant supply voltage. The lower part of the pulsed supply voltage must be equal to the minimum value that was calculated in the previous section. The important question of how close the supply voltage must follow the shape of the normal oscillating frequency of the CUT will be dealt with in a later section. The normal oscillating frequency has a square wave shape with 50 % duty cycle under ideal conditions. It was decided to start by applying a square wave with 50% duty cycle and frequency equal to that of the normal fault-free oscillating frequency of the CUT as the supply voltage. The oscillation frequency will differ slightly from chip to chip and this aspect will be discussed later when tolerances are discussed.

Our method is based on varying the supply voltage of the CUT and observing the output, which is also connected to the input of the specific critical path that is being dealt with. This test must be carried out with every critical path that was identified in the CUT. This can be significant, as in optimized designs there will be many critical paths. Typically only a small number of these critical paths are testable. In other words, many critical paths in a circuit are often untestable. The industry today relies upon ad-hoc at-speed functional testing to cover defects in these paths. However, this is becoming very expensive and impractical with increasing circuit speed and pin-counts.

As mentioned earlier, each circuit will have many critical paths. The number of critical paths grows exponentially with circuit size and is making it impossible to test all critical paths in the circuit. To overcome this problem, several approaches have been proposed to select only the important critical paths for testing. These approaches can be roughly classified into structural delay based and functional approaches.

Functional schemes [17-19] exclude paths that need not to be tested based on logic relations between various signals in a circuit. The authors in [17,19] have defined a set of paths called *robust dependent,* such that if all paths not in this set are delay-fault free, then paths in this set need not to be tested. However, this theoretical finding is not useful in practice since in almost no circuit one can robustly test all the paths not in the robust dependent set. A more useful technique has been proposed in [18, 20]. This technique is based on the fact that some paths in a circuit can never affect circuit timing. If there is a path *P* such that for all possible two-vector input patterns, there exists a gate *g* on the path such that the final value on the on-path input to *g*, is non-controlling while the final value on some side input to *g* is controlling or there exists a gate *h* such some side input to *h* is a steady controlling value then a delay-fault on *P* cannot cause a timing violation. Thus *P* can be ignored for the purpose of delay-fault testing. Such paths are called *functionally redundant* paths as explained earlier in Chapter 3.

The remaining paths in the circuit are *functionally irredundant* (explained in Chapter 3) and defects in these paths can cause timing violations. In almost all circuits, a very large number of paths are functionally irredundant. Hence, in practice, structural delay-based approaches are used to select those functionally irredundant paths for testing that are most likely to fail in the presence of delay defects. The simplest structural approach is to select all those paths in the circuit whose estimated delay, as determined during static timing analysis, is greater than a certain threshold because a delay defect in the circuit is most likely to cause a timing violation on such a path. The technique in [21] selects a set of paths such that for each line, the path with the largest delay passing through it is selected.

More sophisticated procedures [22] either exploit the fact that path delays are related to each other because paths in a circuit share a lot of lines, or use statistical manufacturing process information [23-24].

In Figure 4.4 one can see the flow diagram of the subsequent steps of the test technique that is proposed [25]. According to Figure 4.4 one first has to identify the critical paths in the circuit. Next, the chosen critical path has to be sensitized and one has to make sure that there are an uneven number of inverting elements to ensure oscillation and then the oscillation frequency has to be measured. This can be done, for example by using a digital counter to capture the characteristics. Next, a square-wave supply voltage is applied to the CUT.

The next section will be focussed on establishing the characteristics of the circuit due to the square-wave supply voltage. The last part of the test will be to observe the output waveform of the CUT whilst applying a square-wave supply voltage. The frequency characteristics of the output waveform will show if any delay or stuck-at faults are present. One has to remember that a stuck-at fault can be detected as a delay-fault with infinite delay. Later it will be shown how to interpret the output waveform to be able to prove whether or not delay faults are present in our CUT.

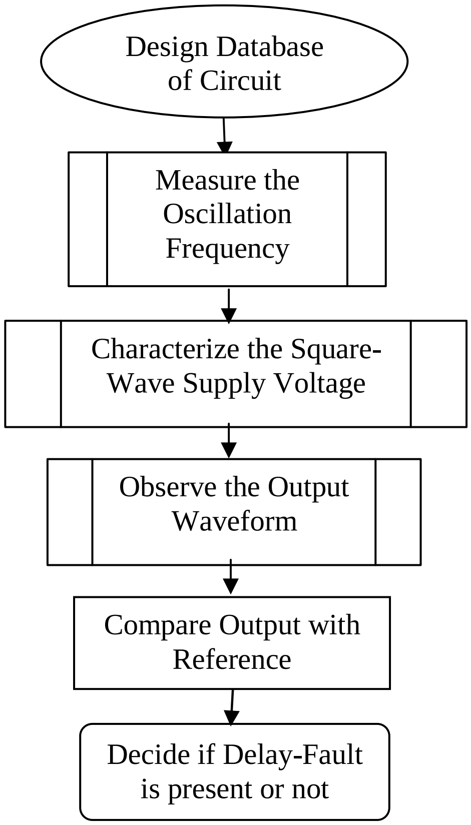

Design Database of Circuit
Measure the Oscillation Frequency
Characterize the Square-Wave Supply Voltage
Observe the Output Waveform
Compare Output with Reference
Decide if Delay-Fault is present or not

Figure 4.4*: Subsequent steps in the pulsed Vdd oscillation test technique.*

4.4.2 Pulsing the Supply Voltage

In the section on VLV it was explained that it is possible to use lower power-supply voltages to detect faults in CMOS circuits. Now, a supply voltage with the same pulse width as the natural oscillating frequency of the circuit is being applied. As a square waveform is the closest waveform shape to the natural oscillating frequency waveform shape it was chosen as the shape of the supply voltage with 50 % duty cycle and period equal to that of the normal oscillating frequency of the CUT. One, therefore, has to ensure that the circuit is still able to operate correctly if the supply voltage is changed to a low value [16]. The supply voltage is critical, which is also a parameter that influences many other characteristics of a particular circuit. The square wave will not go to zero volts during the 50 % off-time but to a certain value. This value is calculated by taking two times the threshold voltage and results in 1.36 Volts in our case. This value of 1.36 volts was explained earlier in the section on VLV testing (Section 4.3). This square-wave supply voltage will only be applied during the testing phase. The only synchronization that is needed is that the square-wave supply voltage must have a logic high initial start condition. The shape of the square-wave supply voltage signal is shown in Figure 4.5. It will be difficult to produce an identical signal to that of the oscillation. The fundamental question - to which extend this will degrade the detection capability if the frequency, phase and rise and fall times will differ - will be answered later in this chapter.

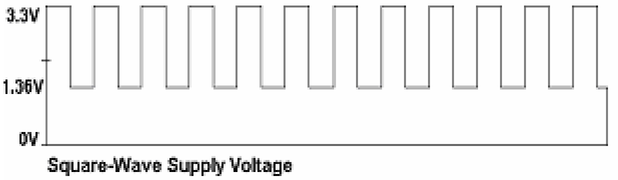


Figure 4.5: *The used square-wave supply voltage for testing.*

4.5 The Fault-Detection Process

4.5.1 Test Setup

A simple possible implementation for the proposed test techniques is shown in Figure 4.6 [26]. The control logic block controls the input logic and output logic as well as the signal comparison. The fault-free circuit response is compacted into the fault-free signature and compared to the output response of the CUT. The stored response will be with respect to either the number of transitions in a given time period or the frequency of the output waveform of the circuit. The purpose of the input logic is to apply the test patterns to the CUT. The output logic is to condition the signal that the comparator will compare with the stored signatures of the fault-free circuit.

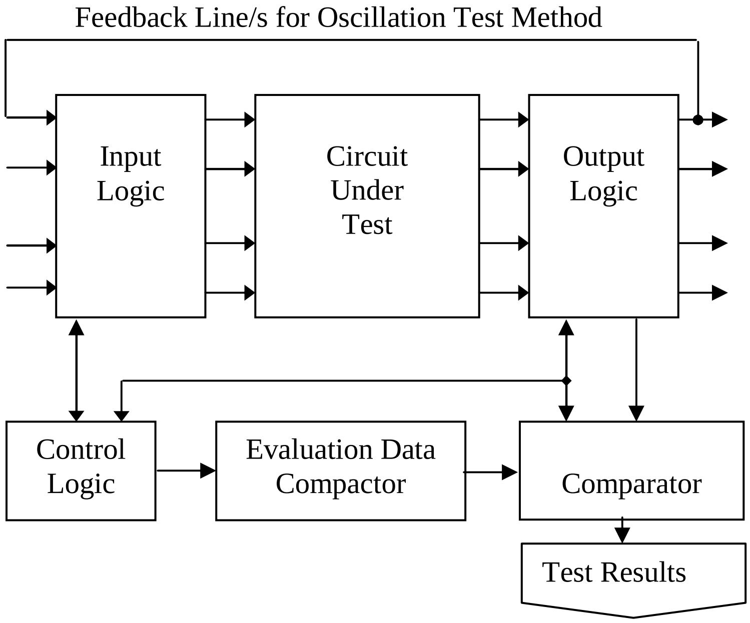


Figure 4.6: *Block diagram representation of a typical oscillation test organization.*

There is an option to include the above-mentioned evaluation on-chip as BIST. Care must be taken for introducing faults into the test setup due to the extra circuitry needed. The extra circuitry must be self-checking or be included in the test process as an initial test. An example of the above is if a Multiple-Input-Shift-Register (MISR) is used to compact the output response for evaluation.

Finally, the control logic directs the whole operation. It controls the input and output logic as well as the evaluation of the oscillating frequency. It must be noted that this test organization assumes that there is only one path sensitized at a given time and converted to oscillation at a given time. To sensitize the path in the circuit, off-path inputs of all gates directly involved in the path should be set to non-controlling values by properly setting the primary inputs.

Compared to conventional delay testing methods, the OTM requires less test vectors as each path can be sensitized and tested using only one test vector instead of two test vectors. One will have to make use of a design tool to generate these test vectors. Two main techniques are classically used to generate test vectors for circuits, namely, the deterministic approach and the simulation-based approach. The simulation-based approach may use either random or genetic generation. In some cases both techniques are found to be combined in the same tool, in others they are separated. Commercially available ATPG tools that can be used include tools such as "DFT-PRO™" from SynTest. These tools will operate on scan-inserted net lists and will include tools for testing DfT rules violations, automatic test pattern generation, as well as test pattern formatting to directly link to ATE from popular vendors such as Agilent, Credence and Teradyne [27] .

4.5.2 Simulation Results

The first set of simulations was carried out on a number of simple combinational circuits. The simulations were carried out using SPICE and a combined net list, based on net lists of primitive functions in the 4000 series. A 4-bit digital adder as was shown in Figure 4.3 was used as vehicle to carry out the initial simulations.

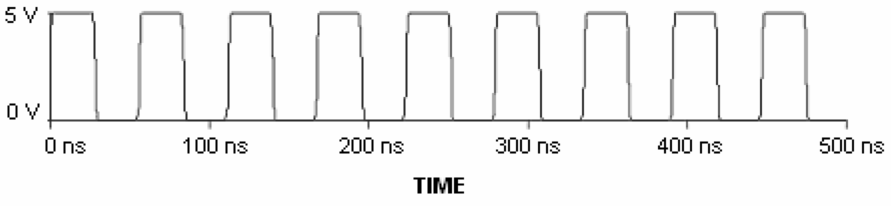


Figure 4.7: *Oscillating output of a sensitized path in a fault-free circuit under test (Figure 4.3) with a constant supply voltage of 5 volts.*

This simulation was carried out because this adder circuit would be built with discrete components to test the technique on a prototype circuit at a later stage. The varying supply voltage must have a logic high initial start condition. Figure 4.8 shows the result of the output if a varying supply voltage is applied. This oscillation occurs if a normal supply voltage is applied to the fault-free CUT as shown in Figure 4.7**.**

First, stuck-at faults were introduced into our adder circuit. A stuck-at 0 and then a stuck-at 1 fault were introduced into the sensitized critical path. The results show clearly that no oscillation occurred and they were therefore easily detected. The overall propagation delay of the critical path in the 4-bit adder was calculated to be 73 ns.

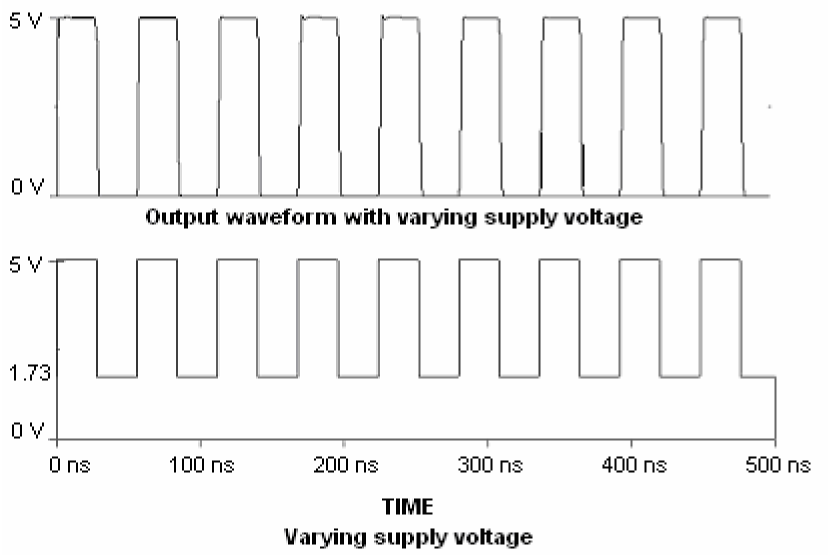


Figure 4.8*: Output waveform of the sensitized path under a varying supply voltage.*

The next part of the test simulations was concerned with inserting extra delay into the critical path and observing the output waveform. The same adder circuit as before was used. The first graph of Figure 4.9 shows the output in the case of an extra delay of 10ns in the critical path and the second if the delay is 4ns. The waveforms show that the normal oscillation characteristics are absent.

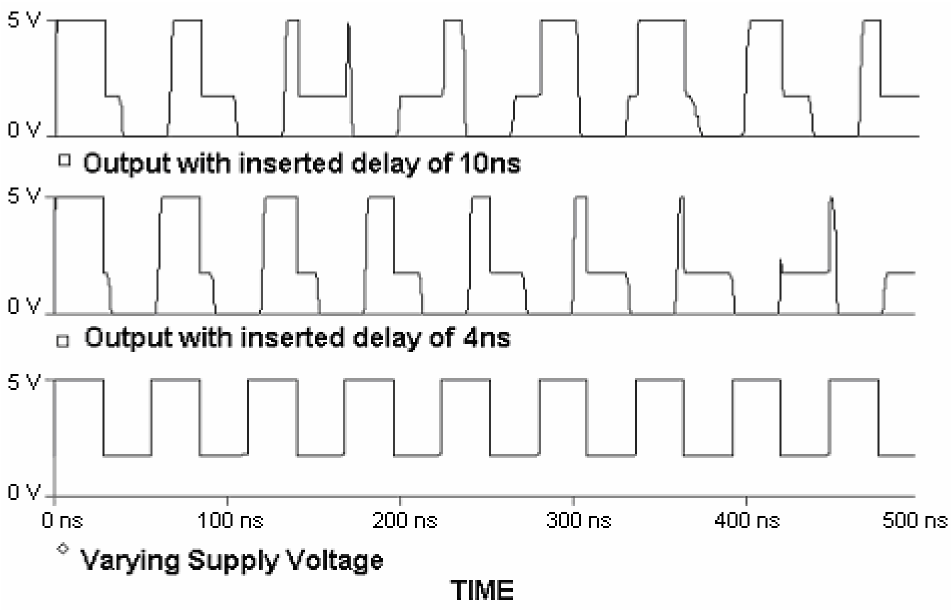


Figure 4.9: *Output waveform of sensitized path with different values of extra inserted delays: (a) extra inserted delay of 10ns and (b) an extra inserted delay of 4 ns.*

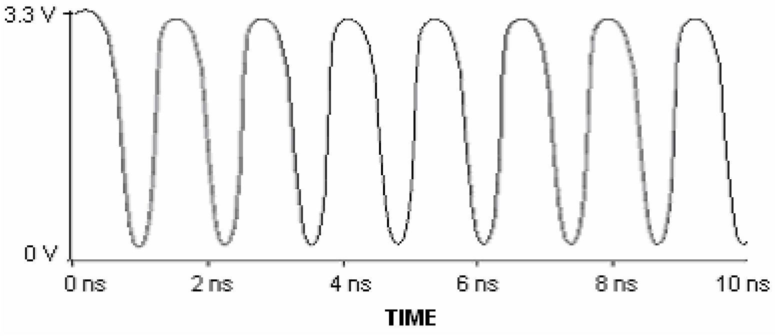


Figure 4.10: *Simulated oscillating output of sensitized path in CUT (see Figure 4.3) using the 0.35 μm TSMC CMOS technology from MOSIS with a constant supply voltage of 3.3 volts.*

The same simulations were now carried out on a 4-bit digital adder using SPICE and using the 0.35 μm TSMC CMOS technology from MOSIS being a more recent process [12]. Figure 4.10 shows the nominal oscillating frequency of the CUT with a constant supply voltage of 3.3 volts for the circuit that was shown in Figure 4.3. In Figure 4.11, the shape of the square-wave supply-voltage signal is shown together with the output waveform of the 4-bit digital adder as was shown in Figure 4.3. This result shows the normal unique oscillating frequency while a square-wave supply voltage is applied to the CUT.

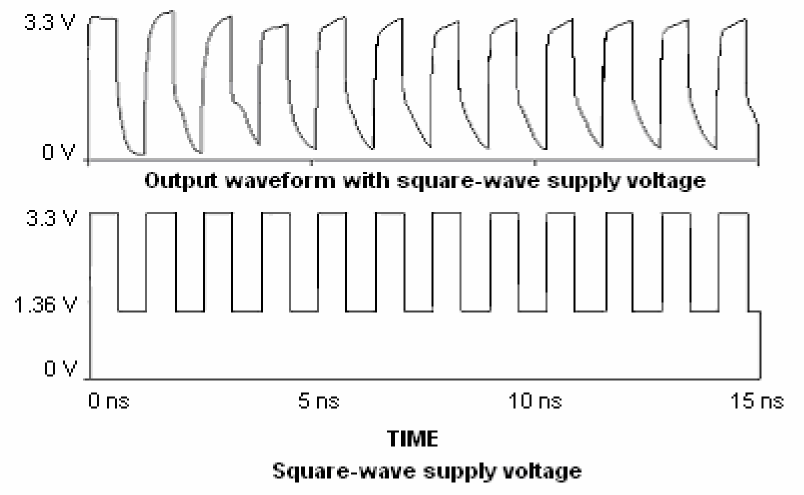


Figure 4.11*: Output waveform using the 0.35 μm TSMC CMOS technology from MOSIS with a square-wave supply voltage.*

In Figure 4.11**,** the characteristic oscillating output waveform is clearly visible. This result shows the normal unique oscillating frequency while a varying supply voltage is applied to the CUT.

First, stuck-at faults were introduced into our 4-bit adder circuit. A stuck-at-0 and then a stuck-at 1 fault were introduced into the sensitized critical path. The results again show clearly that no oscillation occurred and it was therefore easily detected. This was expected as it was in the previous simulations. The overall propagation delay of the critical path in the 4-bit adder was calculated to be 1.28 ns. Figure 4.12 shows the result if an extra delay of 390 ps was added to the critical path.

The result suggests that also the amplitude of the oscillation may reveal information about the presence of a delay-fault, as has been indicated in literature [10].

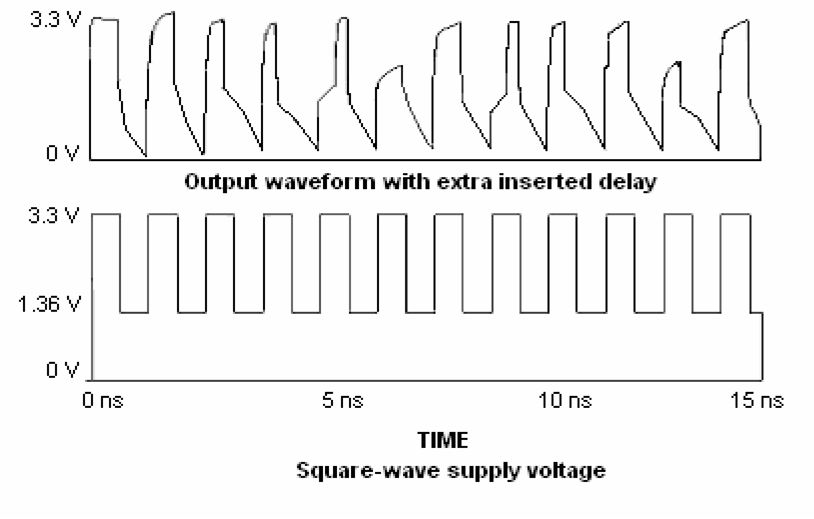


Figure 4.12: *Output waveform with extra inserted delay of 390 ps in the critical path.*

4.5.3 The Effect of the Duty Cycle of the Pulsed Supply Voltage

Up to now, the test simulations were concerned with inserting extra delay into the critical path and observing the output waveform. Indeed a change in the output waveform characteristics was observed if an extra delay was inserted. The idea is to be able to detect very small delay faults and therefore one has to look at inserting very small delays in our simulations. It was therefore necessary to look at parameters that are influencing the size of delay faults that can be detected with this method.

First the effect of the duty cycle of the supply voltage is investigated. This was done in an effort to make the detection of even smaller delay faults possible. Simulations show that the best results for this specific circuit were obtained with a 10% duty cycle of the supply voltage. This can be explained by the relation of the duty cycle and the size of the delay-fault. The same size of delay-fault will be easier detected with a lower duty cycle as the delay will interrupt the normal oscillation completely as with a higher duty cycle were it will only interrupt the output waveform partially. This interruption of the oscillation is visible when the output waveform is on the lower voltage level of the pulsed supply voltage. As with the higher duty cycle sometimes only some of the peaks are malformed and clipped at lower levels than the logic high level. This is due to the fact that the circuit only has the time period available during the high part of the duty cycle to start the oscillation. If a delay-fault is present, it delays the operation of the circuit and therefore interrupts the oscillation of the circuit. The circuit has much less time to start the oscillation when a smaller duty cycle is used.

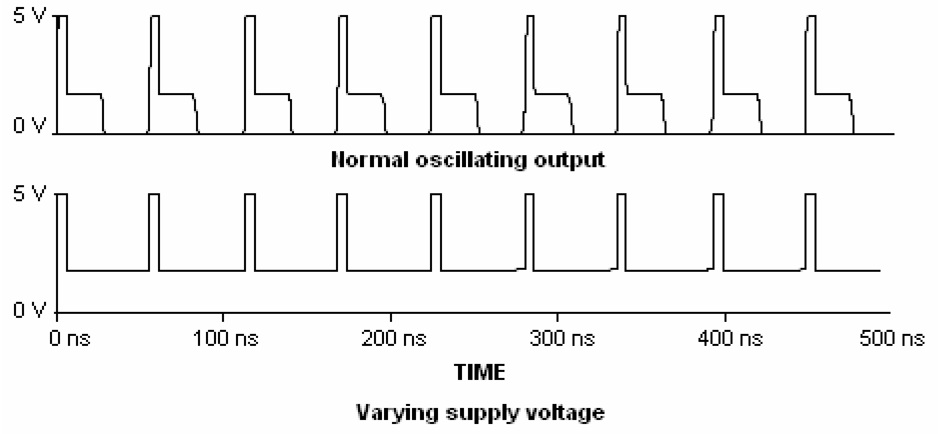


***Figure 4.13**: Output waveform with square-wave supply voltage and duty cycle of 10% using the 4000 CMOS series circuits.*

Results of simulations carried out in SPICE with the 4000 High-Speed CMOS series circuits with a square-wave supply voltage with duty cycle of 10% are shown in Figure 4.13. Delay data was obtained from the circuit's data sheet rated at a load of 50 pF. The waveforms show that the normal oscillation characteristics are present. The results are shown in Figure 4.14 if using the 0.35 μm TSMC CMOS technology from MOSIS [12].

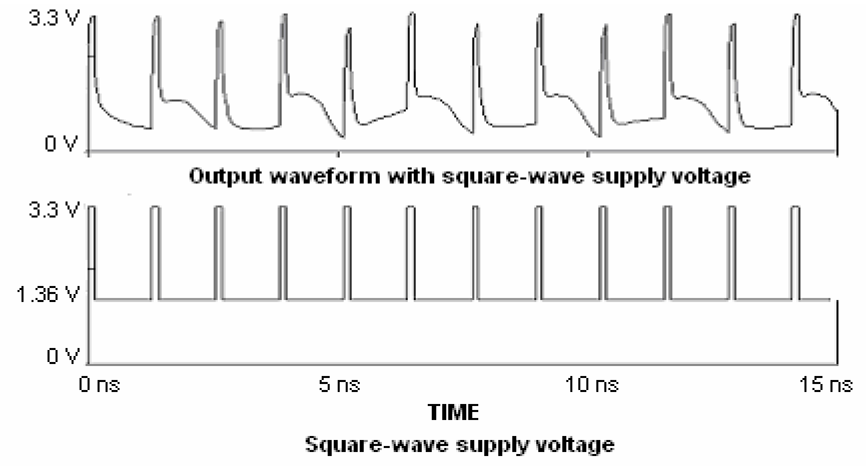


Figure 4.14: *Output waveform with square-wave supply voltage and duty cycle of 10% using the 0.35 μm TSMC CMOS technology from MOSIS.*

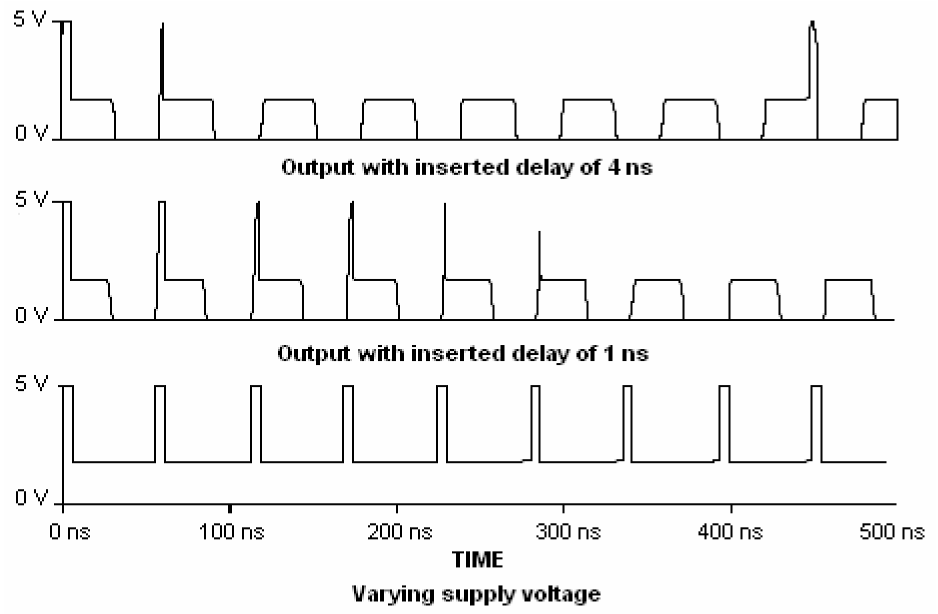


Figure 4.15: *Output waveform with extra inserted delays and duty cycle of 10% using the 4000 CMOS series circuits.*

The overall simulation results show that it is possible to change the duty cycle and still get an oscillation output waveform that is correct. The effect of the duty cycle on the *size* of the *delay-fault* that can be detected is investigated next. One must also look if the *size* of the *propagation* delays of the critical path that is being tested plays a role in what the most effective duty cycle is to be used. More simulations were done with other combinational circuits by applying a square-wave supply voltage with a duty cycle of 10%. It was noted that as the overall propagation delay of the critical path in the circuits are getting smaller, in the region of 92ps, then better results were obtained with a 20% duty cycle.

The results confirm therefore that by altering the duty cycle to a lower value does influence the ability of this testing technique to detect even smaller delay faults than that discussed earlier. As was mentioned before it is, however, necessary to ensure that the circuit does operate correctly at the chosen duty cycle if no extra delay is introduced in the critical path that is being tested.

Results of simulations carried out in SPICE with the same circuit as above with an inserted delay of 31 ps into the critical path are shown in Figure 4.16. The waveforms show that the normal oscillation characteristics are absent.

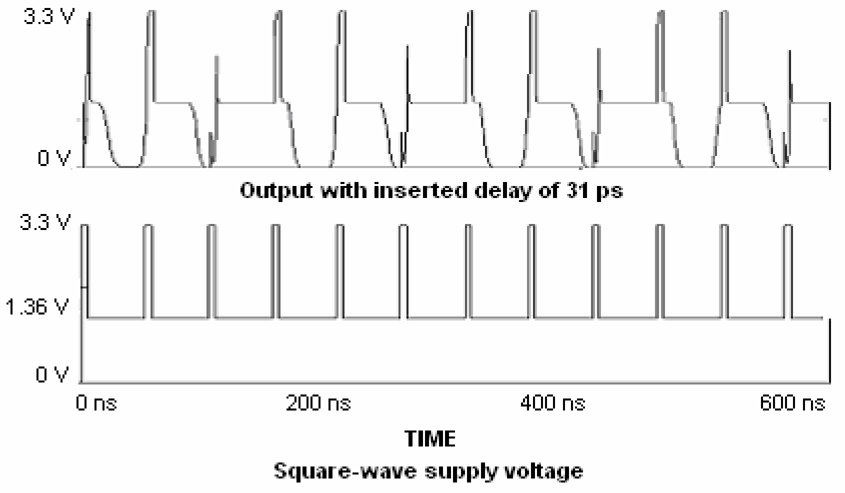


Figure 4.16: *Output waveform with extra inserted delays and duty cycle of 10% using the 0.35 μm TSMC CMOS technology from MOSIS.*

4.5.4 Signature Analysis in the Frequency Domain

To be able to observe and evaluate the abnormal oscillation output due to the inserted delay, fault signature waveforms were created in the frequency domain by performing a Fast Fourier Transform on the raw time-domain waveforms that are shown in Figure 4.14 and Figure 4.16. The waveforms in the frequency domain are shown in Figure 4.17. By mathematically subtracting the waveform of the circuit with inserted delay-fault from the waveform of the fault-free circuit one is able to capture signal differences between devices. The resultant of the subtractions is shown in Figure 4.18 [28].

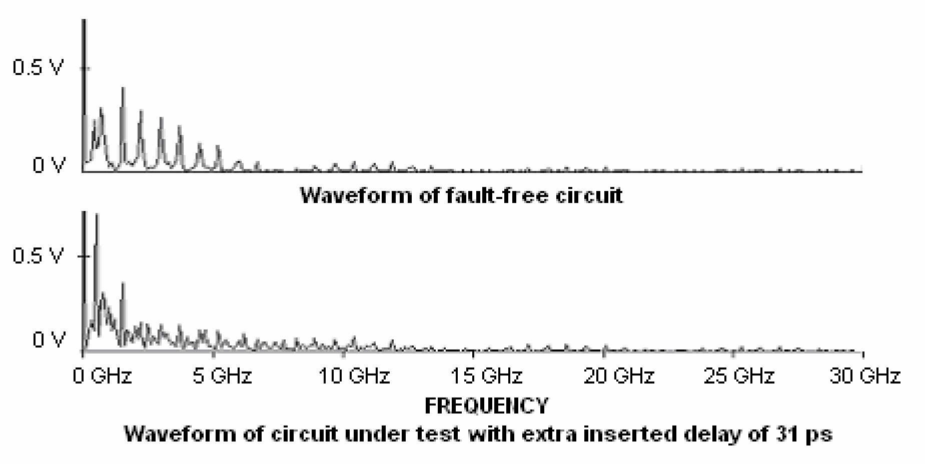


Figure 4.17: *Frequency domain output waveforms.*

The resultant waveform shows that the inserted delay-fault is changing the frequency composition of the output waveform of the test device. This shows that the inserted delay-fault can therefore be detected with this technique. For this fault evaluation one will need an advanced mixed-signal tester.

Although the resultant waveform shows clear differences in the simulations, it is not revealing more information than in the time domain. Simulations show that the magnitude of the resultant waveform is not related to the size of the inserted delay. Fault-free circuits will probably have a whole bunch of slightly different frequency signatures.

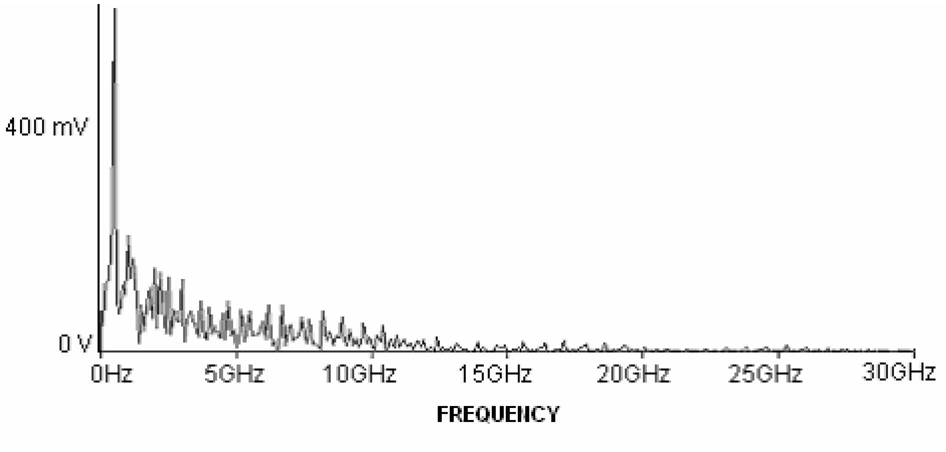


***Figure 4.18**: Resultant waveform after subtraction of the waveforms of Figure 4.17.*

Another obstacle was the question of how these simulations could be verified in practice. Advanced equipment would be needed for the simulations to be done in the frequency domain and then it would still be very difficult, if possible, to measure at the required location in the circuit. The results in the frequency domain are confirming the detection of the extra inserted delay but do not add any new information with regard to the detection of delay faults.

4.5.5 Tolerance Issues in the Pulsed-Supply Voltage Approach

The period of the square-wave supply voltage method is an important parameter. The question is now raised how close to the period of the oscillating frequency the supply-voltage period must be to ensure good results. Simulations were carried out to establish the tolerance that is allowed with respect to the period of the supply voltage; the results show that up to ± 5 % deviation from the oscillating frequency period is still providing good results. This outcome shows that although precise measurements to measure the period of the oscillating frequency are a prerequisite, it does allow for sufficient tolerance so that no expensive equipment is required for this task.

It will therefore be necessary to have a range of frequencies that is acceptable for the fault-free circuit. One would therefore need a digital "larger than" and "smaller than" circuit, and combine the outputs to get the non-allowed values. This will then form part of the evaluation circuitry that will be discussed later.

The question arises if it will be possible to generate this needed pulsed supply voltage. There are products available on the market that can provide this pulsed pattern for our application. A Pulse Pattern Generator that covers a frequency range from 1 MHz to 3.35 GHz and an output amplitude range from 50 mV to 100V is available from Agilent Technologies [29]. It provides full control over all pulse parameters like timing, levels and edges. This device will be able to provide the required pulsed supply voltage for our technique.

4.6 Physical Experiments

4.6.1 The Discrete Circuit used for the Experiments

The circuit that was chosen as actual test vehicle was the same as the one that was used in the simulations. A 4-bit digital adder was used as shown in Figure 4.19.

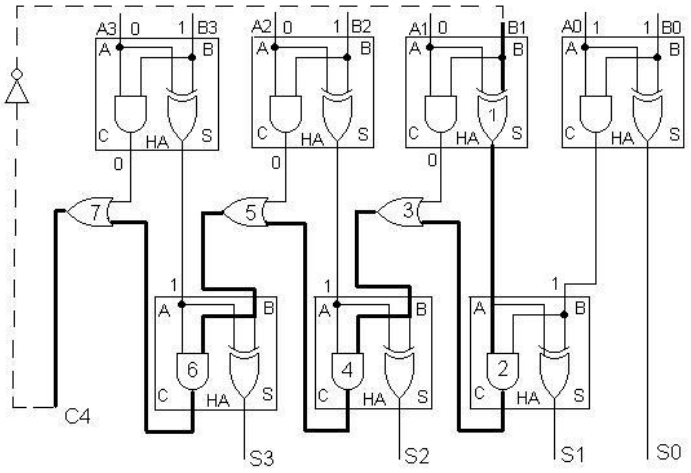


Figure 4.19: *The circuit diagram of the 4-bit adder.*

One of the critical paths that determine the operational speed of the circuit is the path between the input B1 and the output C4 via logic gates 1-6, and 7. This critical path is non-inverting and therefore in order to establish an oscillator the output C4 should be connected to the input B1 via an additional inverter (dashed). The circuit was built on a breadboard and the discrete CMOS components as shown in Table 4.1 were used to design the circuit. The specifications of the propagation delay of the different components that were used were taken from the datasheets and are shown in Table 4.2.

Table 4.1*: CMOS components used in discrete circuit.*

Component	Manufacturer and Part Number
Inverter	Fairchild 4069
AND gate	Fairchild 4081
OR gate	Fairchild 4071
XOR gate	Fairchild 4070

Table 4.2*: Typical propagation delays of components used.*

Component	Propagation Delay from Input to Output @ 50 pF load Typical Rising / Falling Delay
Power Supply	5 Volt
Inverter	± 50 ns
AND gate	± 110 ns
OR gate	± 95 ns
XOR gate	± 105 ns

4.7 Measurements with a Fixed Power Supply

First, several measurements were carried out using a fixed power supply. This was to establish the normal operating conditions of the circuit. While working with discrete circuits, it is a given fact that results will vary due to environmental factors. This is also predicted in the specifications and therefore it is not a surprise that temperature does influence the delay. Electromagnetic fields generated by human presence could occasionally trigger the circuit into instability. The circuit on the breadboard is therefore relatively more sensitive to environmental disturbances than an IC.

4.7.1 Critical Path Delay

The longest delay path in the circuit consists of one inverter, three AND gates, three OR gates, and one XOR gate. So, in theory the typical total delay time (tdt) of the path for the circuit at 5 V, neglecting wiring delay is:

$$\text{tdt} = 1*50 \text{ ns} + 3*110 \text{ ns} + 3*95 \text{ ns} + 1*105 \text{ ns} = 770 \text{ ns}$$

Because, during one oscillation period, this longest delay path is traversed twice, once for the low voltage and once for high voltage output, the period length (Tt) is twice the total delay time, therefore 1540 ns. The typical oscillation frequency (fosc) can therefore be calculated as 649 kHz.

4.7.2 Wire-Delay Measurement

In order to determine the influence of long wires, a measurement was carried out for determining the “average delay” for the wires used in the breadboard. The supply voltage for this experiment was a 5V voltage source. An extra long wire was placed between gate number 4 and gate number 5 in Figure 4.19. The oscillation frequency without and with the extra long wire were measured. The results with a wire of 0.5 mm diameter are shown in Table 4.3.

***Table 4.3**: Oscillating frequencies with different wire lengths.*

Power Supply	Wire length	Oscillation Frequency
5.00 V	3 cm	911.1 kHz
5.00 V	30 cm	900.9 kHz

This results in an additional delay per centimetre wire length of:

((1/900.9 kHz) – (1/911.1 kHz) /2) / 27 cm = 0.22 ns/cm delay at 5 V.

Using the above result the expected delays for the wires used in the breadboard are as follows:

***Table 4.4**: Expected delays for wires used in the breadboard.*

	Wire length	Expected delay
Minimum length	1 cm	0.22 ns
Maximum length	4 cm	0.88 ns

Comparing the values from Table 4.4 with the other delays in the circuit it is less than 3 % of the other delays and is therefore negligibly small. It is therefore noted that relatively long wires have some influence on the oscillation frequency. However in this experiment the delay due to the “long” wires were negligibly small.

4.7.3 The Oscillation Frequency

The identified critical path as shown before was sensitized and the inverter added to get an uneven amount of inverting elements; in this case oscillation of the circuit will occur if the output is fed into the input of the critical path.

The measured output waveform shown in Figure 4.20 shows the oscillation of the circuit. The oscillation frequency was calculated to be 848 kHz. Hence the previous measurements seem to be in the same order of magnitude, but are 200 kHz higher than calculated from the typical delay data. This can be explained by the fact that the “typical” values from the datasheets were used and are an average value. Therefore it can be assumed that the ICs were probably not under the exact same conditions (e.g. different load) than that of the given data.

Measurements of the oscillation frequency of the circuit at 5V varied according to environmental factors and application-induced variations between ±650 kHz and ±920 kHz. The environmental factors, including the temperature, were monitored to be aware of any drastical changes. These measurements were reproducible and were always within the above range.

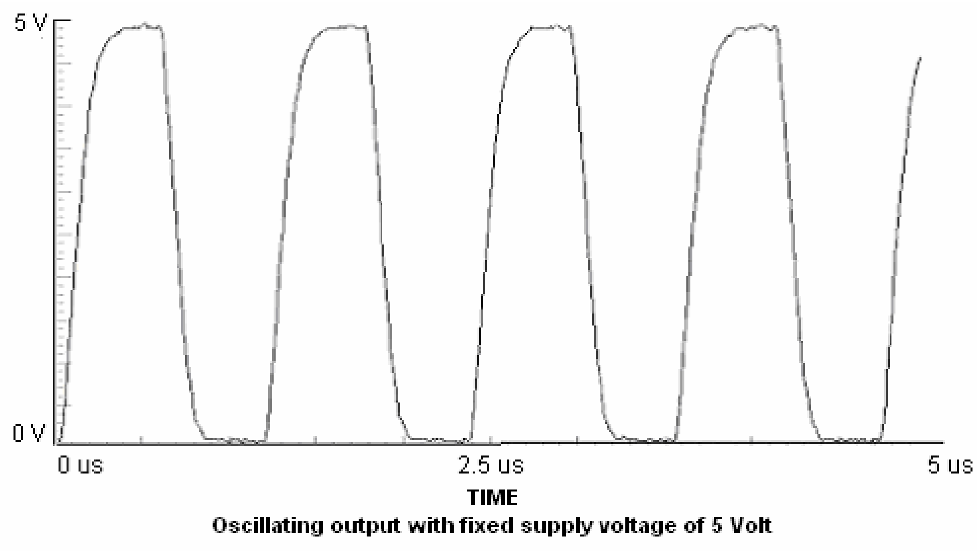


Figure 4.20*: Oscillating frequency of circuit with a fixed 5 Volt power supply.*

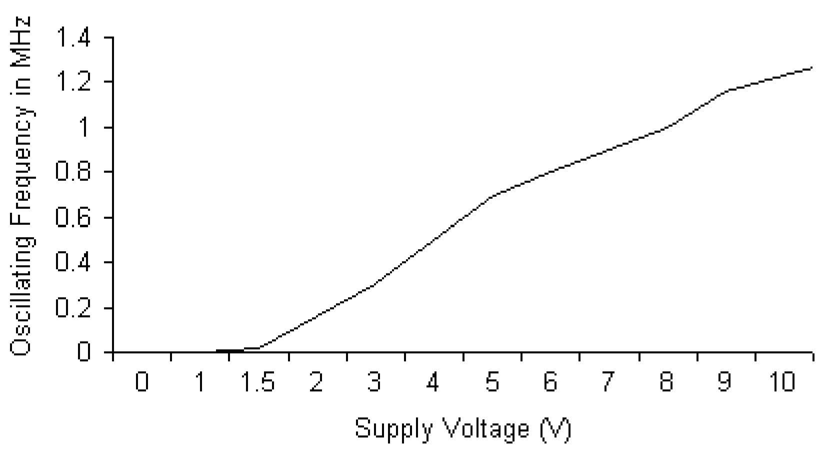


Figure 4.21: *Relationship between the oscillating frequency and the supply voltage.*

Measurements were carried out in order to investigate the relation between the oscillation frequency of the circuit and the power-supply voltage. The relation between the supply voltage and the oscillation frequency is shown in Figure 4.21 and appear to be roughly linear. The minimum voltage at which the circuit still worked was 1.45 V. This is in line with what has been derived earlier, that the minimum operating voltage for a circuit is between 2 and 2.5 times the threshold voltage.

4.8 Measurements in the Case of a Pulsed Power Supply

The circuit did not operate correctly at voltages below 1.45 volt as indicated from measurements of the circuit. The threshold voltage for these integrated circuits is 0.7 volts and the minimum value of the pulsed supply voltage must therefore be between 2 x 0.7 = 1.4 volt and 2.5 x 0.7 = 1.75 volt to be within the range. The minimum value of the pulsed supply voltage was set to 1.75 volt and the maximum to 5 volt. The frequency of the pulsed supply voltage was first taken to be 850 kHz as was the result of *fosc* with a fixed supply voltage and duty cycle of 50 %.

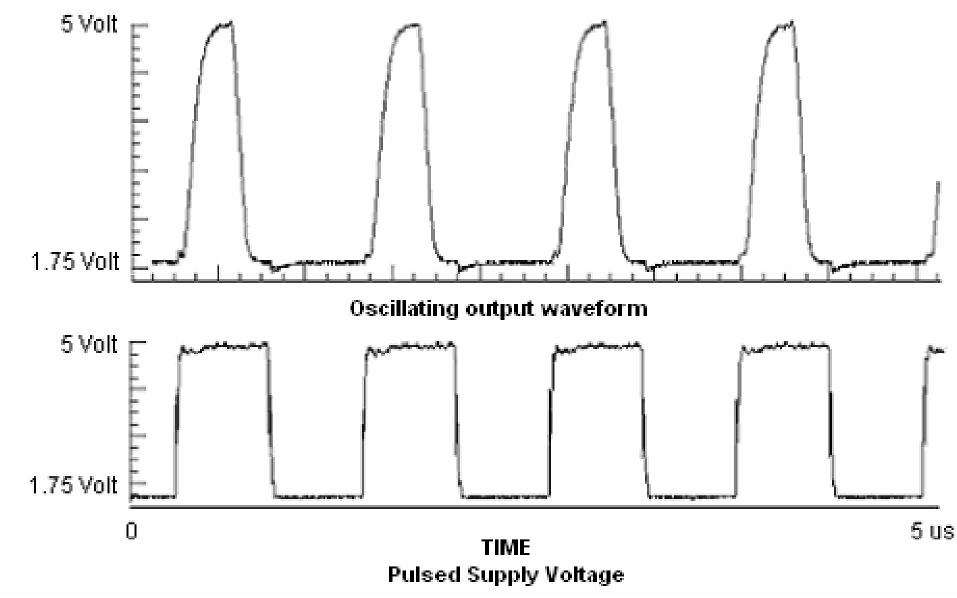


Figure 4.22: *Pulsed supply voltage together with output waveform.*

The first test was carried out using a pulsed supply voltage as calculated before; it is shown in Figure 4.22 together with the output waveform of the circuit. It was noticed that a very clear oscillating output waveform is visible in the result. This is the first indication that the circuit still produces a unique oscillating output waveform if a pulsed supply voltage is applied to the circuit. The same setup has been used with the pulsed supply voltage with a duty cycle of 20 % instead of 50 %. Figure 4.23 shows the results of this test indicating that the circuit is operating as expected (see Section 4.5.3). The option of changing the duty cycle of the pulsed supply voltage produced reliable results.

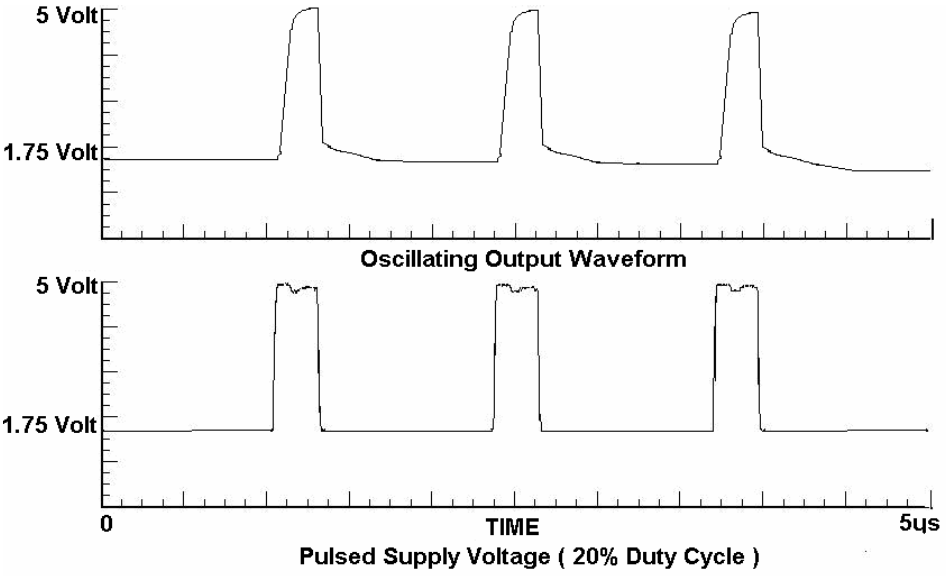


***Figure 4.23**: Pulsed supply voltage with 20 % duty cycle together with the output waveform.*

4.8.1 Measurements with Pulsed Power Supply and Extra Delay

Now, extra delay was added into the critical path by means of inserting two inverters and the exact value of the delay calculated. According to the datasheets the delay should be 30 ns per inverter and therefore the total inserted delay is 60 ns. As previous results already indicated, our components have different values as compared to the given data. A test was carried out to get the precise value of the extra inserted delay.

The first test was performed without the delay to establish the operating conditions. The supply voltage with a maximum value of 5 volt, minimum value of 1.75 volt and a duty cycle of 10 % produced under these circumstances, an oscillating frequency of 917 kHz. When the two inverters were added as extra delay in the critical path, and under the same test conditions as before, the circuit oscillate at 856 kHz. Calculations show that the extra delay is 38 ns. This is in relation to previous findings of the components having a better than "typical" result with regard to delays. It therefore turns out that the propagation delay for each inverter is 19 ns at 5 volt as opposed to the 30 ns at 5 volt as stated in the datasheets.

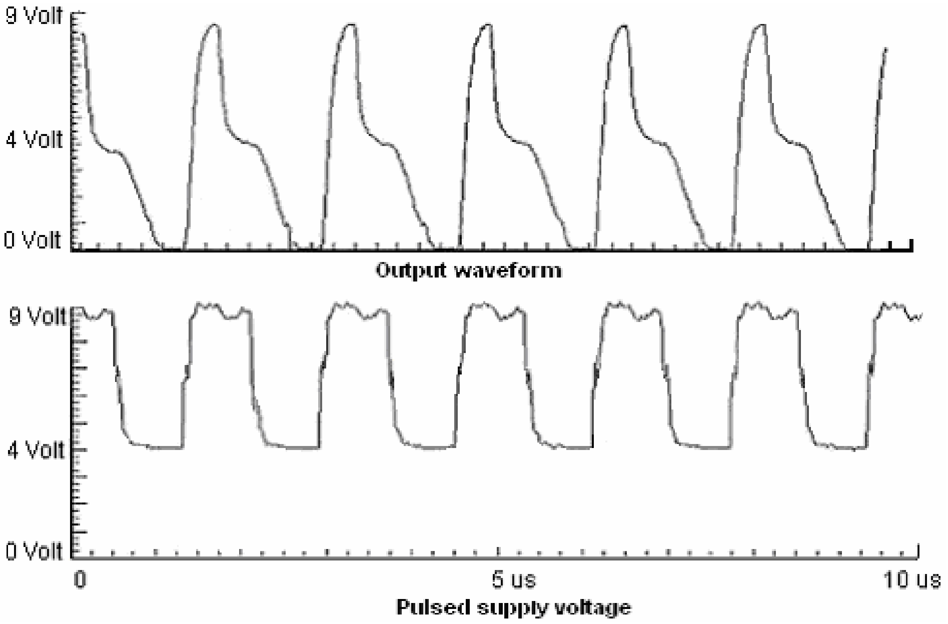


***Figure 4.24**: Output waveform of the test circuit with no extra delay added and a power-supply duty cycle of 50 %.*

According to the datasheet the device can operate from 3 volt to 15 volt. The manufacturer also mentions that 9 volt was the maximum with which the device can operate reliably for long periods of time. This value will later be used as the high value of the pulsed supply voltage (vh). The minimum value was taken as the lowest operating voltage + 1 volt to ensure correct operation of the device with possible power-supply tolerances. This value will later be used as the low value for the pulsed supply voltage (vl).

More measurements showed that if the maximum voltage of the pulsed supply voltage was increased to 9 volt and the minimum value to 4 volt the results of the delay values come closely to the theoretical results which were achieved. For the test involving the extra delay into the critical path, the pulsed supply voltage with maximum value of 9 volt and minimum value of 4 volt was used. The oscillating frequency has an almost linear relationship with the supply voltage and the new oscillating frequency of the circuit is *1160 kHz* (see Figure 4.24). This frequency will be confirmed in the next section. This new value for the pulsed supply voltage was used to carry out the tests with and without the extra inserted delay. The result with no extra delay and the result with the extra inserted delay of 38 ns are shown in Figure 4.24 and Figure 4.25 respectively.

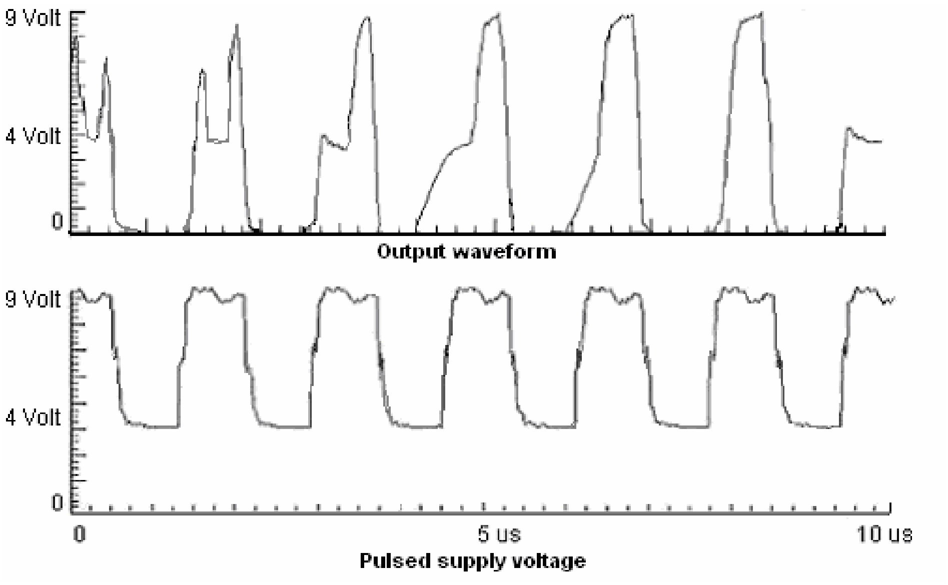


***Figure 4.25**: Output waveform of the test circuit with an extra delay of 38 ns and a supply-voltage duty cycle of 50 %.*

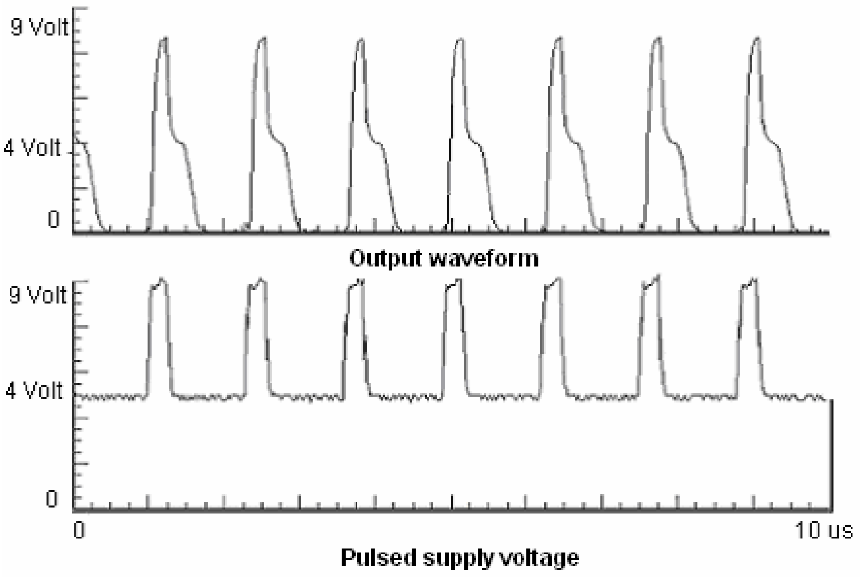


***Figure 4.26**: Output waveform of the test circuit with no extra delay and a power-supply voltage duty cycle of 20 %.*

The results in Figure 4.24 and Figure 4.25 show that, without the extra delay, a stable oscillation output waveform is produced by the circuit. When the extra delay of 38 ns is present the output waveform is no more stable and the oscillation as in Figure 4.24 does not result. The circuit evaluation structure will have to be able to distinguish between these two output waveforms and oscillation frequencies. This evaluation structure will be discussed later.

Subsequently the duty cycle was lowered to confirm previous discussions on the effect of a low duty-cycle value. The first experiment was carried out without any extra delay and next the extra delay of 38 ns was inserted in the critical path that was tested. The results shown in Figure 4.26 and Figure 4.27 respectively, indicate that the extra inserted delay was detected, thereby proving the success of our technique - even if applied to a discrete circuit under difficult test conditions. From these experiments information was gathered which will be discussed in detail in the next section.

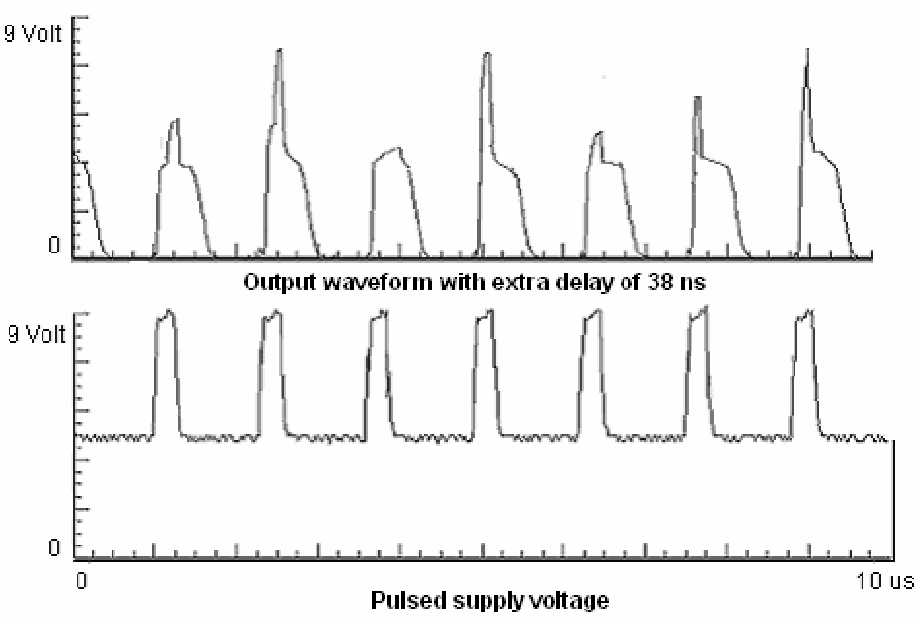


Figure 4.27: *Output waveform of the test circuit with an extra delay of 38 ns in the critical path and supply-voltage duty cycle of 20 %.*

4.8.2 Discussion of Results from the Discrete Test Circuit

It was found that the oscillation frequency varies with environmental factors, which are not surprising as, for example, temperature does influence the delay, as can be seen on datasheets of the various logic gates. Electromagnetic fields generated by human presence could occasionally trigger the circuit out of stability but this will be reduced in a SoC environment.

Because the oscillation frequency of the loop depends on the power-supply voltage, the oscillation frequency changes as the power-supply voltage changes. As a consequence of the varying supply voltage, the normal oscillating frequency of the system under test will alter, however, this change in frequency is easy to calculate [30].

Calculations were carried out to establish a method to calculate the oscillation frequency of the circuit under specific operating and environmental conditions. The following is an explanation of the resulting formula used to calculate the oscillating (fosc) frequency:

$$\text{fosc}_{vs} = \text{fosc}_{vh}*\text{d} + \text{fosc}_{vl} * (1\text{-d}) \qquad \textbf{(4.5)}$$

vh : high value of pulsed supply (see Section 4.8.1)

vl : low value of pulsed supply voltage (see Section 4.8.1)

fosc_{vs} : oscillating frequency under varying power-supply voltage with 50 % duty cycle

fosc_{vh} : oscillating frequency measured for a peak supply voltage (vh)

fosc_{vl} : oscillating frequency measured for low supply voltage (vl)

d : duty cycle

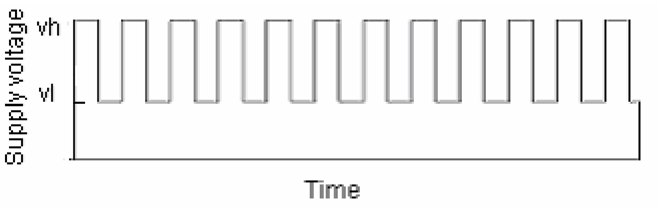


Figure 4.28: *Pulsed supply voltages to be used with Equation (4.5).*

Example:

In the earlier experiment described in Section 4.6 a supply voltage was used with a high value of 9 volt and low value of 4 volt. The oscillating frequency of the circuit at 4 volt was 546 kHz and at 9 volt 1790 kHz.

Applying Equation (4.5) gives the following result:

546 kHz x 0.5 + 1790 kHz x (1-0.5) = 1166 kHz.

This result is within 6 kHz - that is 0.05 % - of the previous measured value (see Section 4.8.1) and confirms the earlier result and the relationship according to Figure 4.21.

It is recommended that, before starting measurements under a varying supply, it is beneficial to measure $fosc_{vh}$ and $fosc_{vl}$ beforehand, in order to make the calculation of the stable oscillation frequency easier. Since in our case the circuit had proven to be sensitive to the environment, (more than 100 kHz difference in oscillating frequency on different days), it was necessary to re-measure these frequencies at the start of each measurement session. Furthermore, it was found that there are more frequencies at which the loop oscillation is stable. For example at double the normal oscillation frequency the circuit is stable and will give good results.

The results from our experiment suggest that the duty cycle must be smaller than 50 % for the optimal detection of small delay-faults. It was also indicate earlier that 10 % is the smallest duty cycle that still gives acceptable results.

4.9 Influences on the Pulsed Supply-Voltage Technique

The disturbances that were experienced with the discrete circuit must be seen in perspective to avoid uncertainty about the ease by which this technique can be applied. First, it must be remembered that the test on the discrete circuit was an effort to establish the feasibility of the technique. It was expected that the results would be influenced by environmental factors like the temperature. Also the use of “long” wires to build the circuit has created its own problems with respect to EMC interference. The electromagnetic fields produced by different equipment used in the test have also added its own part to variations in the test results.

Taking into account all the different factors and interference on the circuit it still produced good results. This is promising because if the method is providing good results under these circumstances it will probably work even better in an integrated circuit.

4.9.1 Process and Application Induced Variations

As mentioned in Chapter 3, virtually every step in the chip fabrication process introduces tolerances; hence a subset is capable of creating substantial changes in the electrical response of circuits. Due to the critical timing necessary when detecting small delay faults, it is crucial to have precise timing in the structure.

The susceptibility of the test method to process and application-induced variations will be similar to those presented in Chapter 3 of the DfDT structure. The variation in Vdd was identified as a possible problem for the test technique [31]. A practical solution will be to eliminate this factor either by having a very stable supply voltage or by adding a regulated voltage-reference circuit. Inaccuracy in the position of the clock timing edges can be due to process variations in the clock generation circuit. The timing inaccuracy can cause undesirable clock-skew in the test technique. A clock-skew correction circuit can therefore be added to solve the clock-skew problem.

4.10 Analysing the Oscillating Signals

The final step in the test technique is deciding if an unwanted and/or an unacceptable delay has occurred in the circuit. The analysis of the oscillating signal is thus the final step to the go/no go decision. The only measurable signal is the output waveform of the CUT. As we are using the OTM it will therefore be either an oscillating or non-oscillating output signal. Let us first look at the non-oscillating output waveform.

The CUT is set up to oscillate as described earlier. If there is a deviation in the oscillating frequency of the CUT then it is a clear indication of a delay-fault. Stuck-at faults that are related to components and/or interconnects that are involved in the oscillator structure manifest themselves in the complete loss of oscillation. Delay-faults in the paths of the oscillation structure can also cause a deviation from the normal oscillating frequency.

The deviation from the normal oscillating frequency is an indication that some delay-fault is present in the specific oscillation ring. It can therefore give an indication of the magnitude of the delay-fault when compared to the normal oscillating frequency.

Contrary to the loss of oscillation, that needs no further analysis, is an investigation of the deviation from the fault-free oscillating frequency. The most obvious and efficient way is to measure the oscillating frequency. The obtained results show that often some of the peaks of the oscillating signal is absent in the case of a delay-fault. It will be necessary to count the peaks to establish the normal oscillating frequency. The threshold value of the peak in the oscillating signal that must trigger the counting mechanism must be set sufficiently high to ensure that malformed peaks will not be able to act as a trigger. The standard measure to take 90 % of the logic high value [32] can be applied for the threshold value for triggering the counter.

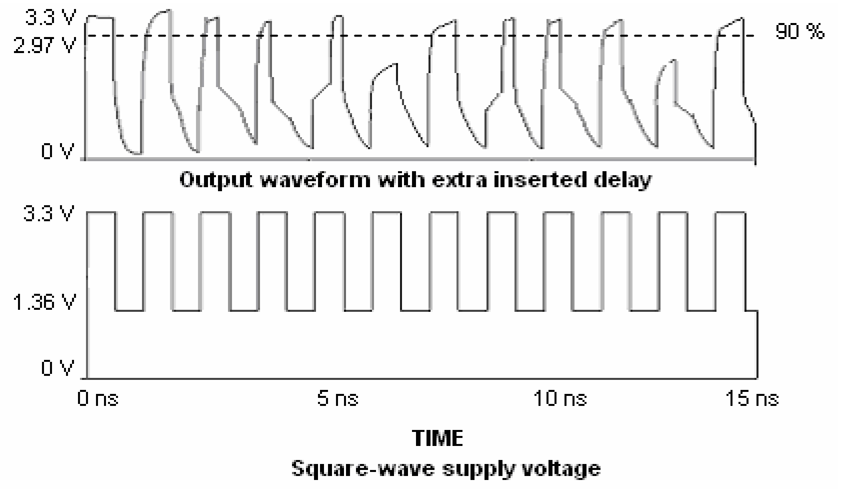


***Figure 4.29**: Figure 4.12 with the added line at 90 % of the logic high value.*

The result that was previously shown has an added line at the 90 % mark. It is now clear from Figure 4.29 that an output waveform that seems to have all the transitions will have two less counts on the counter and confirm our previous results that a delay-fault is present in this CUT. It is again interesting to note that some faults change the amplitude and not the frequency.

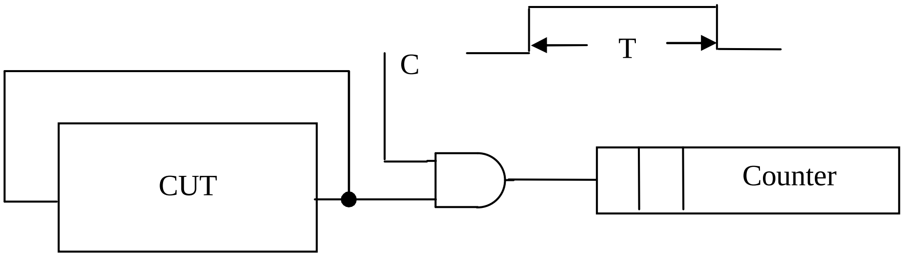


***Figure 4.30**: Configuration to measure oscillating frequency.*

The evaluation hardware must be kept as simple as possible to keep the extra needed silicon to a minimum. The oscillating signal can thus be fed into a counter with threshold value triggering (as shown at 90% in Figure 4.29) and the number of pulses be counted within a specified time slot (T in Figure 4.30). A possible configuration to measure the oscillation frequency to detect if delay-faults have changed the normal oscillating frequency is shown in Figure 4.30. Alternatively an ATE in place of a counter can be used to assess the oscillating frequency. This simple testing circuitry (counter) can form part of the BIST structure of the circuit.

Experimental results show that the number of patterns required to cover all the lines of a circuit is approximately only one tenth of the total number of circuit lines. An added advantage of this test scheme is that it can test the circuit at-speed. It can also be used to measure the maximum speed of a given microprocessor. This is done by measuring the maximum frequency of a set of the critical paths (longest paths) of the microprocessor [33].

4.11 Conclusions

Digital oscillation test methods have already been used in the past for measuring precisely the maximum operating frequency of circuits in a given technology. It has also been used for the detection of faults by observing the output or by counting the output frequency. Low-voltage testing is useful if not all delay faults can be detected at normal operating conditions. Our proposed new technique whereby we combined the principles of both the oscillation test methods and low-voltage testing involves the application of a pulsed supply-voltage to the CUT. The option to make use of a pulsed power-supply is a newly developed technique. The supply voltage has a square waveform varying at the normal operating frequency of the CUT with an offset equal to a low voltage value equal to 2.5 times the CUT threshold voltage, and a high value equal to the normal supply voltage of the CUT. Experimentally it was found that it can be useful to use a supply voltage higher than the normal supply voltage of the circuit.

The effect of a pulsed supply voltage was evaluated in simulations and with experiments with regard to an actual circuit. The results of the measurements made on the discrete circuit produced valuable information about the method. The scheme connects outputs of the CUT to its inputs with odd inversion parity and applies appropriate input patterns to the unconnected inputs to sensitize particular paths of the circuit, converting them effectively into oscillation rings. By observing whether the output oscillates normally, at the target frequency, or not, one can tell whether the circuit incorporates faults.

The test resolution depends on the technique employed to measure and evaluate the oscillating frequency and amplitude. This test technique will have a very small impact on the chip performance because it is only active during the test sequence.

The effect of a change in duty cycle was also evaluated and this effect was used to enhance our test technique to be able to detect even smaller delay-faults. This technique relies on the assumption that all critical paths can be sensitized as are most of the other techniques that detect delay faults.

This testing method requires added hardware, which can be applied externally or built internally in the circuit to be part of a Built-In-Self-Test architecture. The extra silicon area required for this test method will depend on the complexity of the circuitry to change the duty cycle of the power supply voltage. Testing of the circuits can be done at-speed with this technique. The influence of process- and application-induced variations can have a major effect of the testing results. As already mentioned, the test conditions will have to be standardized and normalized to produce the best results. The technique is able to detect stuck-at as well as small delay faults and proved to be efficient in terms of area overhead.

4.12 References

[1] R.C. Aitken, ”Designs are Changing – Can Test Keep up?”, *An Invited address to the International Test Conference*, 2000, Atlantic City, USA, page 13.

[2] H. Speek, H.G. Kerkhoff, M. Shashaani and M. Sachdev, “A Low-speed BIST Framework for High-Performance Circuit Testing”, *Proceedings of VTS*, 2000, Montreal, Canada, pp. 349-355.

[3] V. D. Agrawal and T. J. Chakraborty, "High-Performance Circuit Testing with Slow-Speed Testers", *Proceedings of International Test Conferenc*e *(ITC)*, 1995, Washington, DC, USA, pp. 302-310.

[4] C.J. Lin and S.M. Reddy, “On delay fault testing in logic circuits”, *IEEE Transactions on CAD*, September 1987, pp. 694-703.

[5] M.L. Bushnell and V.D. Agrawal, “Essentials of Electronic Testing”, ISBN 0-7923-7991-8, Kluwer Academic Publishers, 2000.

[6] A. Steininger, “Testing and built-in self-test – A survey”, *Journal of Systems Architecture*, 46 (2000), 2000, pp. 721-747.

[7] G.L. Smith, “Model for Delay Faults Based upon Paths”, *Proceedings of International Test Conference (ITC)*, 1985, Philadelphia, USA, pp. 342 – 349.

[8] K.T. Cheng and H.C. Chen, “Classification and Identification of Non-robust Untestable Path Delay Faults“, *IEEE Transactions on CAD-15*, 1996, pp. 1027-1034.

[9] M. Arabi, H. Ihs, C Dufaza and B. Kaminska, "Digital Oscillation-Test Method for Delay and Stuck-at Fault Testing of Digital Circuits", *IEEE Transactions on Instrumentation and Measurement*, Volume: 48 Issue: 4, August 1999, pp. 798-806.

[10] G. Huertas et al., “Practical Oscillation-Based Test of Integrated Filters”, *IEEE Design and Test of Computers*, Volume: 19 No. 6, December 2002, pp. 64-72.

[11] K.T. Lee and J.A. Abraham,” Critical Path Identification and Delay Tests of Dynamic Circuits”, *Proceedings of International Test Conference (ITC)*, 1999, Atlantic City, USA, pp. 421-430.

[12] http://www.cadence.com/products/digital%5Fic/

[13] H. Hao and E.J. McCluskey, “Very-Low Voltage Testing for Weak CMOS Logic IC’s”, *Proceedings of International Test Conference (ITC)*, 1993, Baltimore, USA, pp. 275-284.

[14] http://www.mosis.org: MOSIS TSMC 0.35µm CMOS process data.

[15] http://vlsi.wpi.edu/technology/AMS/ams.html: AMS 0.8µm CMOS process data.

[16] T.Y.J. Chang, and E.J. McCluskey, “Detecting Delay Flaws by Very-Low-Voltage Testing," *Proceedings of International Test Conference (ITC),* 1996, Washington, DC, USA, pp. 367-376.

[17] U. Sparmann, D. Luxenburger, K. T. Cheng, and S.M. Reddy , “Fast identification of robust dependent path delay faults”, *Proceedings of the 32nd Design Automation Conference (DAC),* 1995, San Francisco, USA, pp. 119-125.

[18] K. Cheng and H. Chen, “Classification and identification of non-robust untestable path delay faults”, *IEEE Transactions on Computer Aided Design of Integrated Circuits & Systems,* vol. 15. No. 8, August 1996, pp. 845-853.

[19] W. Lam, A. Saldanha, R. Brayton, and A. L. Sangiovanni Vincentelli, “Delay fault coverage and performance tradeoffs”, *Proceedings of the 30th Design Automation Conference (DAC),* 1993, Dallas, USA, pp. 446-52.

[20] K. T. Cheng and H. C. Chen, “Delay testing for non-robust untestable circuits”, *Proceedings of International Test Conference (ITC),* 1993, Baltimore, USA, pp. 954- 961.

[21] W. Li, S. M. Reddy, and S. K. Sahni. “On path selection in combinational logic circuits”, *IEEE Transactions on Computer Aided Design of Integrated Circuits & Systems,* vol. 8, no. 1, January 1989, pp. 56-63.

[22] S. Tani. M. Teramoto, I. Fukazawa, and K. Matsuhiro, “Efficient path selection for delay testing based on path clustering”, *Journal of Electronic Testing: Theory and Applications (JETTA),* vol. 15, no. 1, 1999, pp. 75-85.

[23] J. Liou, K. Cheng, and D. Mukherjee, “Path selection for delay testing of deep sub-micron devices using statistical performance sensitivity analysis”, *Proceedings of VLSI Test Symposium.* 2000, Montreal, Canada, pp. 97 - 104.

[24] G.M. Luong and D.M.H Walker, ‘Test generation for global delay faults”, *Proceedings of International Test Conference (ITC),* 1996, Washington, DC, USA, pp. 433-442.

[25] H.J. Vermaak, H.G. Kerkhoff and G.D. Jordaan, " Using a Pulsed Supply Voltage for Delay Faults Testing of Digital Circuits in a Digital Oscillation Environment", *Proceedings of IEEE International Conferences on Electrical, Electronic and IT research activities (Africon),* 2002, George, South Africa, pp. 47-52.

[26] H.J. Vermaak and H.G. Kerkhoff, "Testing for Delay Faults using a Variable Supply Voltage in Combination with the Oscillation Test Method" *IEEE European Test Workshop (ETW)*, 2002, Corfu, Greece, pp. 115-116.

[27] http://www.syntest.com/General.htm

[28] H.J. Vermaak and H.G. Kerkhoff, " Using a Square Wave Supply Voltage to Detect Delay Faults in High Speed Digital circuits", *Proceedings of The Dutch Technology Foundation Conference on Circuits, Systems and Signal Processing (ProRisc)*, 2002, Eindhoven, The Netherlands, pp. 520-525.

[29] http://www.home.agilent.com/USeng/nav/-536898117.0/pc.html

[30] O. Petre, Test Strategies for High-Speed Synchronization Interfaces, *Ph.D. Thesis*, ISBN: 90-365-2022-3, University of Twente, February 2004.

[31] H.J. Vermaak and H.G. Kerkhoff, "Reducing the Susceptibility of Design-for-Delay-Testability Structures to Process- and Application-Induced Variations", *Proceedings of IEEE European Test Workshop (ETW),* 2001, Stockholm, Sweden, pp.35-41.

[32] J.P. Hayes, "*Introduction to Digital Logic Design*", ISBN 0-201-15461-7, Addison-Wesley, USA, 1993.

[33] W. Ching Wu, C. Len Lee and M. Shae Wu, "Oscillation Ring Delay Test for High Performance Microprocessors", *Journal of Electronic Testing: Theory and Application (JETTA)*, vol.16, no. 2, 2000, pp. 147-155.

Chapter 5

Enhanced Wrapper Cells Suitable for Delay-Fault Testing of Embedded Cores

The continuing development of smaller and faster circuits has made Ultra Large Scale Integration (ULSI) feasible. A few years ago, a 1 million transistor, 50 MHz device was considered large and fast - now we are looking at 100 million transistors and 3 GHz clock frequencies in commodity parts. The time-to-market (TTM) requirements and the short product lifetimes dictate that new chips have design cycles measured in months instead of years. Assembling a design team to “hand layout” transistors for these huge designs will never economically achieve the TTM goals - so design reuse of cores, has become the solution to create “massive functionality” in a relatively short period of time.

SoC design methodology, the use of embedded cores, and the compression of the design cycle all individually present test problems. In particular it is creating critical timing problems that are surfacing as delay-faults. The creation of complex designs using embedded cores in a short period of time can be a straight-forward test problem (if the cores are delivered with acceptable testability), or a severe test problem. This can depend on the creation of a chip-level test architecture, the mixing of multiple test strategies as delivered with each core, the management of the physical design, and managing the volume of test data.

To meet the aggressive TTM and quality requirements associated with today’s products, some form of structured testability must be supported within each core and also at chip level. Core-based designs cannot be made with a new ad hoc test-architecture style at the beginning of each design. Cores must be made with reuse considered from the beginning, and a chip-level test architecture assembly process must be developed with consideration of many cores, and their vectors arriving from many sources, and possibly with many different types of individual testability supported.

Design methodologies for SoCs based on reusable predesigned circuits, which are occasionally called intellectual property (IP) circuits and/or cores, are becoming very popular. Verifying that such designs meet their performance objectives is difficult due to the hidden implementation details of the IP circuits. Delay-fault testing of these cores is therefore difficult as access to these cores is not always available.

The testing of embedded cores is receiving a lot of interest nowadays due to the popularity of SoCs. The SoC design community is divided into two groups: the core providers and the core users. The providers supply these cores as verified blocks to the user who is only concerned with the design, manufacturing and testing of his own system. The cores are provided as soft, firm or hard-core, not manufactured and therefore not tested [1]. This leaves the testing of the core to the user after manufacturing of the system. The user would therefore require assistance from the provider for delivering of pre-defined tests with the core.

In this environment, the IEEE 1500 SECT has been developed. It facilitates the testing of embedded cores as separate entities [2]. The IEEE 1500 standard does not cover internal test methods of the cores or SoC test integration and optimisation. The IEEE 1500 is targeted at testing “black-box” third party cores; the implementation details of the cores are hidden from the core user and it is hence mandatory that the core providers deliver tests with these cores to ensure that testing of these cores can be carried out with sufficient fault coverage [3].

5.1 Introduction to Core-Based Design

The core-based design methodology has no universally accepted definition. “System Chips" refer to highly integrated devices, which can also be called systems on silicon, systems-on-a-chip, system-on-chip, system-LSI, system-ASIC, platform ASIC or other similar terms. The definition of what should be included in a SoC varies from analyst to analyst and company to company. Regardless of the nomenclature used or the specific contents, there is no dispute that the trend is rapidly moving toward multi-million-gate devices with more than 80 percent of content determined by pre-designed hardware and software blocks [4].

Much like the way standardized physical components are rapidly mixed and matched today on a printed circuit board, an IP in standardized form needs to be rapidly mixed and matched into system chips. In 1996, a group of major EDA and Semiconductor companies was formed with two goals: firstly, to establish a unifying vision for the chip industry, and secondly, to develop the technical standards required enabling the most critical component of that vision: the mix and match of IP cores from multiple sources. The group that was formed was called the VSI Alliance (VSIA). To facilitate mix and match, VSIA specifies "open" interface standards, which will allow IP cores or blocks to fit quickly into "Virtual Sockets" at both the functional level (e.g., interface protocols) and the physical level (e.g., clock, test, and power structures) [4].

Roadblocks remain for efficient and economical use of IPs. The chief roadblock is the lack of open IP-to-IP interface standards upon which to base IP development and use. Much of the advantage of IP cores is lost if companies must drain resources attempting to recreate data or converting IP formats or interfaces to make them compatible with internal design elements.

Currently, most cores available for integration must be purchased from established vendors, often at very high prices. These costs can be burdensome, especially for small design teams with limited funding. With this problem in mind the OPENCORE project was launched [5]. The main objective of the project is to design and publish core designs which will be freely available, freely usable and reusable open source hardware. The following actions are the means of reaching the objective of the OPENCORE project:

- Develop standards for open source cores and platforms
- Create tools and methods for development of open source cores and platform
- Develop open source cores and platforms
- Provide documentation for these cores and platforms

These tools and methods should allow large, widespread, even international, teams to develop hardware in an open way.

5.2 The Delay-Fault Testing of Core-Based Systems-on-Chip

One of the most fundamental problems faced after designing a core is determining the method of achieving access to the core for test purposes, if it is embedded. This is a system integration engineering issue [6].

Providing test access for delay-fault detection by using the natural functional interface cannot take into account the timing involved with applying vectors to, or observing vectors from, the core interface. This results from the fact that the timing of the complete path is not known at the time of the core development. This implies that the timing vectors (the vectors that can detect delay faults) must be generated for the core late in the design when the overall chip design and timing are known.

If the core will be delivered as hard core, meaning linked to a process at layout level, then the test-access method needs to be defined and developed at the time of the core design, not at the time of integration. This is because the hard core must have the DfT hardware already included at the time of deployment, and the core designers must have placed the DfT within the core during the design phase. If the hard core is delivered with pre-generated vectors, then the test-access method is the method that will allow the vectors to be applied as generated [7]. The possible methods for hard core integration are:

1. **Add Nothing:** Integrate the core and functionally test it - this has a high vector cost and a possible negative fault-coverage impact. Pre-generated vectors cannot be delivered with the core - they must be developed by the integrator by using the VHDL simulation model of the core. This model may be encrypted, so fault grading is not possible. A time-to-volume issue may arise if developing vectors for coverage of the embedded core takes months to achieve an acceptable level.

2. **Direct Access:** Integrate the core by bringing all core signals to the chip primary inputs and outputs - this has a high cost in terms of routing and a possible fault coverage impact. Vectors can be delivered with the core, but not for core interface timing, since the final core interface will not be established until it is embedded within the chip.

3. **Slow Boundary Scan:** The test integration method is to provide functional vectors, but via relatively slow scan elements [8]. This method reduces the number of chip-level routes, but requires converting functional vectors to serial scan vectors, and does not address the timing assessment

requirements. Joint Test Action Group (JTAG) boundary scan is based on serially shifting in the value of the pin interface, and then applying or "updating" that state as one action. Note that a core may support only a boundary scan ring, not a Test Access Port (TAP) controller. In the IEEE 1149.1 standard a compliant chip cannot have two TAP controllers with included bypass registers [8].

4. **Built-In Self-Test (BIST):** This test integration method uses embedded LFSR provided with the core to generate vectors and compress (signature analyze) the response [8]. This has a design and area overhead cost, and may have a possible fault-coverage impact (random vectors are not as efficient as deterministic vectors). Furthermore, there may be a test-time impact, (it takes more random vectors to get an equivalent fault-coverage). BIST may also affect the design time of cores as the LFSRs must be designed and chosen for their ability to attain sufficient coverage, so experiments and fault simulations may be required to assess the suitability of the LFSRs used.

5. **Stored-Pattern Test:** The integration method is to deliver a set of vectors together with the core, and to require them to be placed in an on-chip memory. This method has a high area cost if it requires a dedicated memory (such as a ROM) or a high engineering design schedule cost, meaning time used for data transfer to and from memory, if it requires accessing a system memory (such as an SRAM or EEPROM).

6. **At-Speed Scan:** This integration method adds a number of parallel scan chains to both the cores internal and the signal interface logic. This allows for a reduced test-interface (less global routing: just the scan connections) and also allows for an "in-isolation" timing assessment. This means that the full spectrum of vectors can be generated for the core prior to delivery of the core and the core wrapper (the wrapper will be explained in detail in the next section) will allow these vectors to be applied even if the rest of the chip is inoperable. In this case, vectors that can be reused are generated by ATPG after the core design and delivered together with the core.

These are however not the only option for core integration as in industry the following concept is also used: local structural test-pattern generation is done for their blocks or cores. To generate the actual test vectors at the primary inputs, the concept of test-vector expansion is used which takes e.g. other blocks or wrappers into account.

The test-architecture choices depend on both the business considerations and the size or complexity of the core. If the core is small then it is advisable not to apply any inherent DfT but just to recommend an integration method of multiplexing all the core signals. The test methodology would be to access the core directly from the chip-package pins and to apply an optimized set of functional or structural test vectors. The additional pins will have a high cost. Up to now most approaches to detect delay-faults in embedded cores are software driven. Specialized software are developed and used with the specific core. A wrapper is used to get access to the core [9].

If the business goal is to implement a core that will go into a very low-cost market, then the test costs may be a bigger issue. Also, if the core has a large signal interface, but is designed to go into inexpensive plastic packages (which probably have a small number of pins) then a test-interface reduction may be required. In these cases, a trade-off has to be made to include a more efficient test methodology to reduce the effective test interface. An example is boundary scan that provides a few signals access to the signal interface in a serial manner which allows the reuse of test vectors to be made for the core prior to integration.

5.3 Core Wrappers as DfT Interface

One of the most effective reuse methods that can be applied to a potential core design is to support the design philosophy that is being referred to as “Test Ring”, “Test Wrapper”, “Test Collar”, or “Boundary-Scan Ring”. In the remainder of this thesis we will refer to it as wrappers. These units can be an additional layer of hierarchy known as an “add-on” or “slice wrapper,” or they can be designed-in and be integral to the core design itself. A slice wrapper is generally thought of as a group of logic elements that can be laid across the original core interface, whereas the integral, or built-in, wrapper is made with the core interface logic, and is not separable [10]. A SoC with multiple cores and wrappers is shown in Figure 5.1.

The wrapper can be used for several purposes. If designed correctly, the wrapper can reduce the number of primary inputs and outputs required to fully test the core after it has been embedded. For example, if the core has 130 data signals but these are tied together into a 130-bit scan chain, then the primary inputs and outputs can be reduced to 3 primary pins (a scan input, a scan output, and a scan control-signal). This, however, enables testing of the primary input and output only and, if nothing else is done, would require applying functional vectors converted to serial scanned-in data or the application of a scan-based

ATPG tool to generate core vectors via the serial access [11]. The internal logic is also tested by bringing the core scan-chains to the wrapper boundary as shown in Figure 5.2.

Parallel testing will be possible due to the fact that each core is being surrounded by a wrapper and therefore shielded from any external influence. This option of parallel testing can be investigated in further studies.

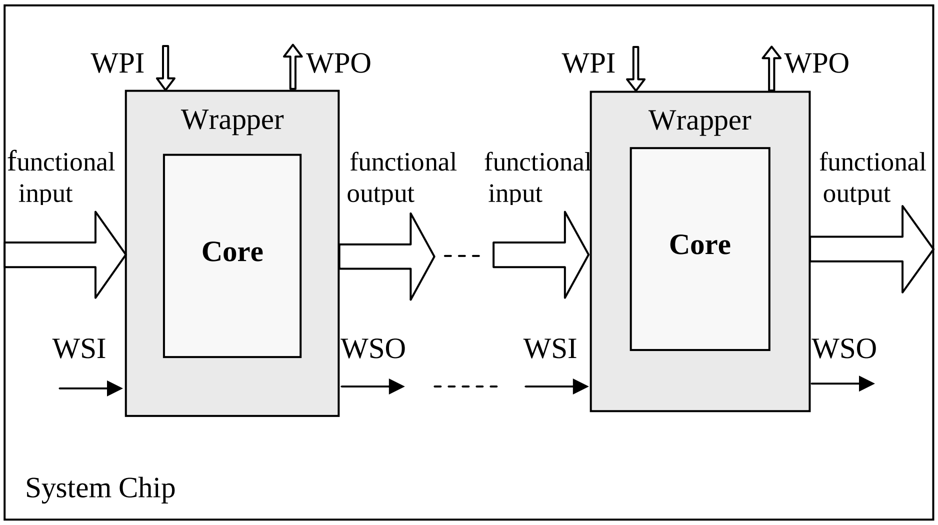


Figure 5.1: *SoC with multiple cores and wrappers joined in chain.*

If the core is designed to operate at some maximum frequency and is to be embedded in a design that will surround it with logic that operates at a different frequency, then the best course of action is to design the test wrapper cells to be also the frequency boundary. This implicates that the cells must be designed for the operating speed of the core and that the cells therefore act as boundary between the core and logic with a different operating frequency. This can be incorporated in several ways:

1) the core and wrapper have a different clock domain from the surrounding chip logic;
2) there is a structure in the wrapper that informs the core when incoming and outgoing data is valid;

3) there is a known cycle ratio for incoming and outgoing data to be valid. (The cycle ratio means the relation of the frequencies to each other of the different domains e.g., the bus-to-core transfer is 2:1, meaning the bus frequency is two times faster than that of the core).

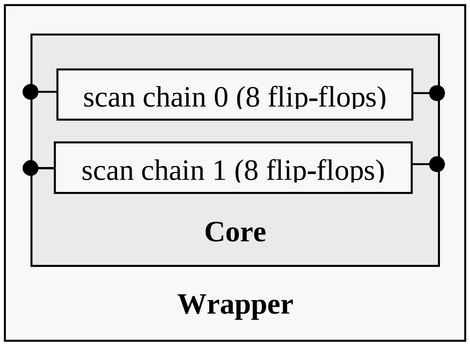


***Figure 5.2**: Core with internal scan-chains brought to wrapper.*

If scan is to be applied to allow an at-speed frequency assessment, then the ability to preserve the timing during the scan cycle is important. This means that the wrapper must provide the control signal or clock signal, used for time-domain separation, as the control signal has to be read-in via a Test Access Mechanism (TAM) in the wrapper-cell shift-register mode, and subsequently applied to the core input.

The question now is how do the existing methods for delay-fault detection adapt to core-based SoCs? This adaption is an open issue, since, in addition to the standard test-quality problems, core-based SoCs present new challenges. These challenges are in particular in terms of development for providers and TAM for integrators. Various solutions, either at the core or system level, have been proposed [11-15]. Regardless of their potential benefits in the long term, unless implemented automatically using a reliable test tool flow, these architecture-specific DfT methodologies do not provide reusability and flexibility.

To address these problems in a well-structured modular way one will have to look at a practical state-of-the-art SoC test approach, such as the IEEE 1500 SECT [11]. Our research will concentrate on the development of a delay-fault test technique within the IEEE 1500 SECT.

5.4 The IEEE 1500 Standard

5.4.1 Basic Principles of the IEEE 1500 Standard for Embedded Core Testing

The IEEE 1500 wrapper, consisting of input and output wrapper cells, acts as a shell around a core that allows that core to be tested as a stand-alone entity by shielding it off from its environment. The wrapper also allows the external circuitry, being other cores and glue logic, to be tested independent from the state of the core. The wrapper must be able to function in three different modes. The first mode is the functional mode, in which the wrapper is transparent and operates as if not existing. Then, one has an *inward-facing* test mode, in which test access is provided to the core itself. Finally, there is an *outward-facing* test mode, in which test access is provided to the external circuitry outside the wrapper. A block level overview including the serial and parallel control and TAMs of the wrapper are shown in Figure 5.3.

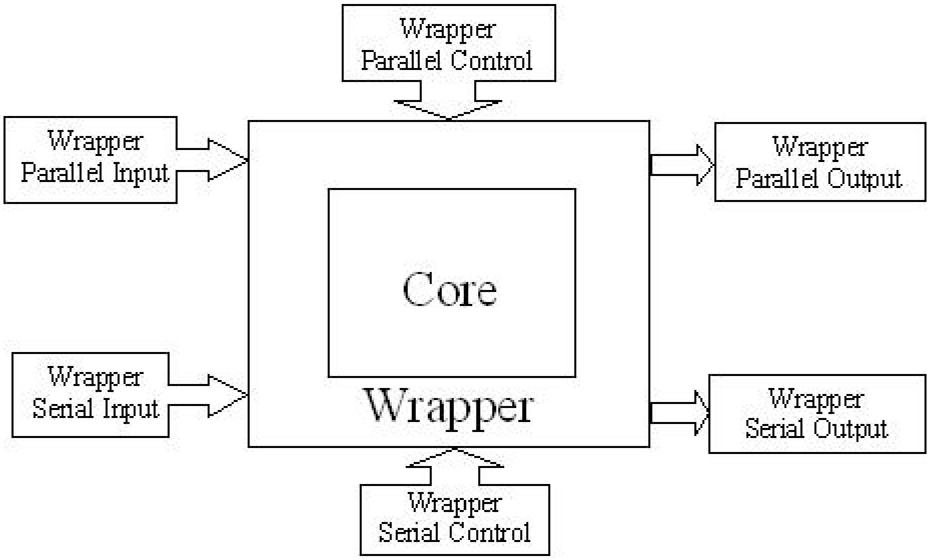


***Figure 5.3**: The IEEE 1500 test-wrapper structure for embedded cores.*

5.4.2 The Wrapper Architecture

Within the IEEE 1500 standard there are optional and mandatory components [11]. Some of the mandatory elements are: a one-bit input/output port pair, WSI (‘Wrapper Serial Input’) and WSO (‘Wrapper Serial Output’), and optionally one or more multi-bit input/output port pairs, named WPI (‘Wrapper Parallel Input’) and WPO (‘Wrapper Parallel Output’). The complete wrapper is controlled via the Wrapper Interface Port (WIP) and an internal Wrapper Instruction Register (WIR). The WIP requires six control signals. The WIP controls the WIR. The WIP allows the WIR to be loaded with an instruction via the WSI.

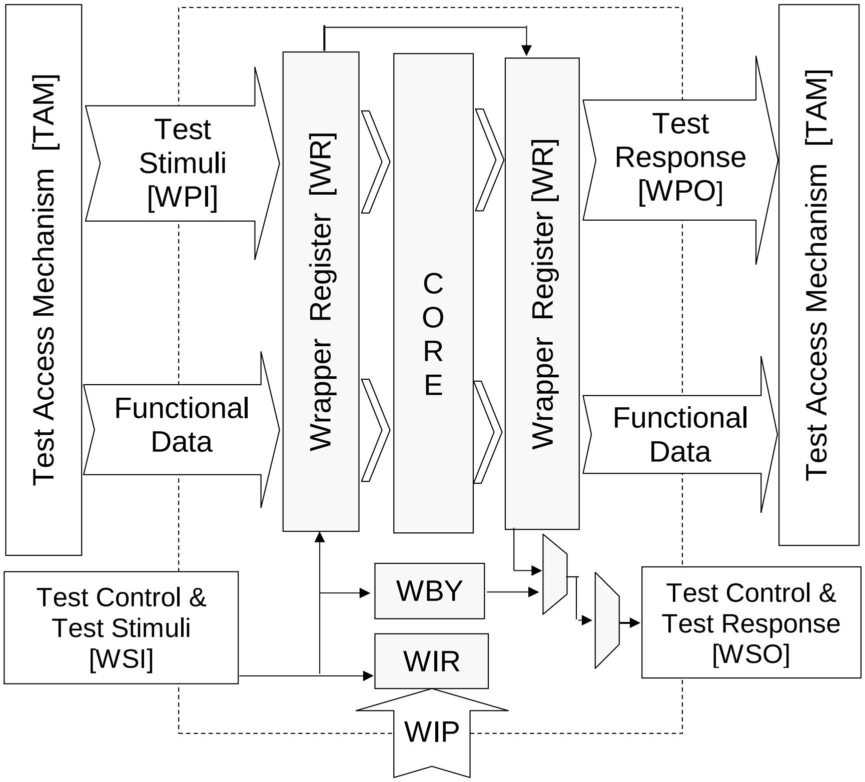


Figure 5.4*: Overview of the IEEE 1500 Wrapper Architecture.*

The operation of the remainder of the wrapper is controlled by both the WIP signals, as well as the instruction loaded into the WIR. The WIP provides control data to the WIR in parallel. Once an instruction is loaded into the WIR, the corresponding test mode becomes active. One more component of the wrapper is the Wrapper Register (WR) which is constructed with Wrapper Cells. There is one Wrapper Cell (WC) per functional digital core terminal.

The IEEE 1500 wrapper also contains a Wrapper Bypass Register (WBY), which serves as a bypass for the serial test data through WSI and WSO. The control signals are used as mentioned earlier to control the overall operation of the wrapper together with the instructions that are loaded into the WIR. Figure 5.4 shows an overview of the IEEE 1500 wrapper architecture.

5.4.3 Wrapper Instruction Register

The WIR provides control over the wrapper cell operation. It determines whether the wrapper is in an inward-facing or outward-facing test mode, whether a serial or parallel access mode is utilized, and which wrapper data register is selected for test access. The WIR may also provide test signals to the core for certain instructions, such as those that enable inward-facing test modes used for internal testing of the core. When loading an instruction, the WIR is exclusively connected in between WS1 and WSO. If WSI and WSO are used for transporting instructions, they cannot be used simultaneously for test data. This mutual exclusion is obtained by the SelectWIR signal of the WIP.

The WIR has a dual register implementation, consisting of a Shift Register and an equally long Update Register. Instructions are scanned into the Shift Register, and become active only when clocked into the Update Register. Parallel inputs to the WIR may optionally be provided in order to capture data into the Shift Register during WIR capture operations. This captured data may be used for test control, testing of the WIR, or the testing of other IEEE 1500 circuitry. Control lines from WIR were omitted for reasons of simplicity in Figure 5.4 but shown in detail in Figure 5.14.

5.4.4 Wrapper Interface Port

The WIP requires six signals [16]. These signals control the operation of the WIR. For example, they determine whether the WIR is uniquely connected between WSI and WSO for loading an instruction, or whether these terminals are free for test data. The overall operation of the wrapper is controlled from the WIP signals and the instruction loaded into the WIR.

• **WRCK** (‘Wrapper Clock’) is the dedicated IEEE 1500 clock signal for WIR, WBY, and WR.

• **WRSTN** (‘Wrapper Reset’) is the dedicated IEEE 1500 asynchronous, active-low reset signal. When asserted, it resets the WIR, putting the wrapper in normal functional mode.

• **SelectWIR** selects whether WSI and WSO are used for instructions or test data. When asserted, the WIR is exclusively connected between WSI and WSO. When SelectWIR is de-asserted, WR, WBY, or any core-internal data register can be selected by means of an instruction in the WIR.

• **CaptureWR, ShiftWR, UpdateWR**. When the corresponding WIP signal is activated, a capture, shift, or update operation will be enabled for the selected wrapper register. In other words, either the WIR (in case SelectWIR is activated) or a wrapper data register identified by the instruction in the WIR (only if SelectWIR is de-activated) is being selected for either a capture, shift or update operation. These operations are synchronous with WRCK.

The WIP timing is such that WIP input signals SelectWIR, CaptureWR, and ShiftWR are set up prior to the rising edge of WRCK and the signal UpdateWR is set up prior to the falling edge of WRCK. In addition, new WSI data will be captured on the rising edge of WRCK, while WSO data changes on the falling edge of WRCK.

For the WIR, multiple operations are not permitted to occur simultaneously. Accordingly, when the SelectWIR signal is active, only one of CaptureWR, ShiftWR or UpdateWR may also be asserted. When the CaptureWR, ShiftWR, and UpdateWR signals are de-activated, the WIR Shift Register and WIR Update Register will hold their current states. The SelectWIR and ShiftWR WIP signals are active prior to the first rising edge of WRCK (see Figure 5.5) and held for example for the next four WRCK cycles. This initiates a WIR shift operation and subsequently shifts four bits of WIR data, shifting data in via WSI on the rising edge of WRCK and shifting data out via WSO with the falling edge of WRCK. In the sixth WRCK cycle, the ShiftWR signal has been de-activated and the UpdateWR signal asserted. Consequently, on the next falling edge of WRCK the data from the WIR Shift Register will be clocked into the WIR Update Register.

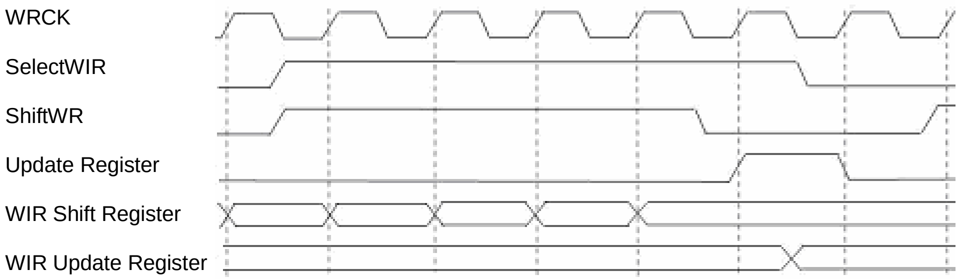


Figure 5.5: *WIP timing diagram.*

This sequence has now updated a new wrapper instruction in the WIR. Following the WIR update, the SelectWIR signal is de-activated. Starting with the next WRCK cycle, a shift of the data register selected by the wrapper instruction will occur. The timing with respect to the WIP, for a WIR shift operation followed by a WIR update operation is illustrated in Figure 5.5.

5.4.5 Wrapper Register

The wrapper input and wrapper output cells provide test access to the core terminals. In a serial test mode, test data is fed through WSI and WSO (see Figure 5.4). For core external testing, the WR forms one serial shift register in between WSI and WSO. For core internal testing, the WR may be combined with core-internal scan registers between WSI and WSO. The bandwidth provided in the serial test modes is dependent on the width of the test access mechanism which is part of the design parameters of the core and wrapper.

In an (optional) parallel test mode, the WR and possibly the core-internal scan chains, are fed from the multi-bit WPI and WPO and hence multiple ‘wrapper chains’ can be constructed in order to reduce the shift lengths involved. An IEEE 1500 wrapper can have zero or more WPI/WPO port pairs, all of user-defined width. The idea is that the width can be adapted to the band-width need of the core and/or the bandwidth availability at SoC level. Therefore to design test architecture for a given set of modules and a given number of test pins, a SoC integrator has to determine (1) the test architecture type, (2) the number of TAMs, (3) the widths of these TAMs, (4) the assignments of the cores to TAMs, and (5) the wrapper design for each core [17].

5.4.6 The Wrapper-Cell Architecture

The WR provides test access to the core. The WR exists of Wrapper Input/Output Cells. In principle there is a single wrapper cell for each functional digital core terminal, although some core terminals e.g. power lines do not have a wrapper cell associated with them [18]. These terminals are associated with special signals such as clock and analogue signals. The wrapper cells must be able to provide the following features:

- Functional pass-through
- No performance degradation
- Controllability from the test data ports using the WSI and WPI
- Observability to the test data ports using the WSO and WPO
- No large silicon overhead

In the various wrapper modes, the wrapper cells are configured into one or multiple test-access chains, through which stimuli can be applied and/or response can be observed. There are multiple types of wrapper cells e.g. depending on the data direction of core terminals, such as input, output and bidirectional. Note that IEEE 1500 only defines the behaviour of such cells and not the implementation thereof. Hence the IEEE 1500 standard allows for *extension* of the functionality of the wrapper cells.

5.4.7 Wrapper Input Cell

The wrapper input cell (WIC) is the part of the wrapper that links the chip and the core. It decides with reference to the control signals which input or test stimuli are to be send to the core via the specific input. The standard wrapper input cell consists of a multiplexer for selecting between the test data or signals for normal operation. The data is clocked in via the D flip-flop (Figure 5.6) and then through the second multiplexer that selects either the signal from the chip or the wrapper test data for testing the core. The standard wrapper input cell is shown in Figure 5.6. The data that is loaded in the flip-flop can also be used to test internal flip-flops or registers inside the core by means of an internal core scan-chain.

The input cell must behave in accordance to the specific wrapper instruction that is applied. This can be the normal mode, where the input wrapper cell output connects to the core input or in isolation mode where the cell output is essentially disconnected.

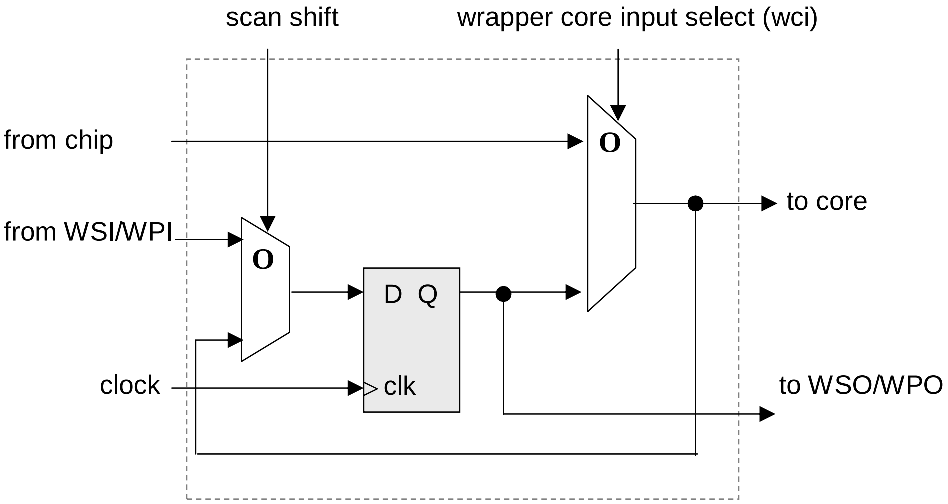


Figure 5.6*: Setup of the standard wrapper input cell.*

It is therefore clear that each wrapper input cell connected to a functional input of a core must have the ability to: (1) control data being propagated into the core and (2) be able to capture data coming from logic external to the core. This simple input cell (Figure 5.6) provides all the functionality that is required from these wrapper input cells. The input cell as discussed is just one version of a WIC and many more sophisticated input cells do exist [3].

5.4.8 Wrapper Output Cell

Similar to the input cell, the wrapper output cell (WOC) links the core and the chip. It also decides with reference to the control signals what input or test stimuli is sent to the chip via the specific output terminal of the core. The standard wrapper output cell consists of a multiplexer for selecting between the test data or signals for normal operation.

The data is clocked in via the D flip-flop and then fed into a second multiplexer that either selects the signal from the core or the data for sending it to the chip. A simple implementation of a wrapper output cell is shown in Figure 5.7. The output cell must behave in accordance to the specific wrapper instruction that is applied via the WIP into the WIR. It can be in the normal mode, being transparent and the core functioning normally or in inward-facing mode that affects the core due to the test that is directed towards the core. The wrapper output cell also has a safe mode that makes sure that the wrapper does not allow unwanted signals to control or influence the core, thereby damaging the core.

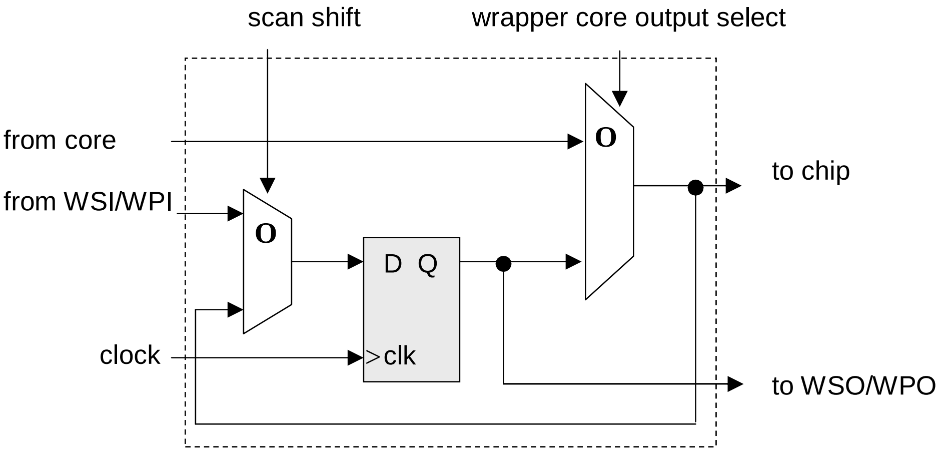


***Figure 5.7**: Set-up of the standard wrapper output cell.*

The cell is part of the wrapper architecture and must behave accordingly and move data through the chosen path, sample the data, and apply data at the moment when the data becomes active and effective, update the output cell and move data from the update register to the chosen path.

It is therefore clear that each wrapper output cell is connected to a functional output of a core and has the ability to: (1) control data being propagated into logic external to the core and (2) be able to capture data coming from the core. The simple cells shown in Figure 5.6 and Figure 5.7 provide all the basic functionality that is required from the wrapper cells. It must again be noted that these cells that have been discussed are just one version of a WC and many more sophisticated output cells do exist.

5.4.9 Wrapper Instructions

Instructions loaded into the WIR (Figure 5.4) determine, together with the WIP signals, the mode of operation of the wrapper. There is a minimum set of instructions and corresponding operations that have to be provided. Optional instructions and their corresponding behaviour are also defined, together with the requirements for an extension of the instruction set. All instructions that establish test modes while utilizing WPI and WPO are optional, as their presence is optional. A minimal implementation of an IEEE 1500 compliant wrapper has no test-data busses. This minimal wrapper provides limited test-access bandwidth via the single-bit WSI and WSO only.

The IEEE 1500 wrapper has various modes of operation. There are modes for functional (non-test) operation, inward-facing test operation, and outward-facing test operation. Different test modes also determine whether the serial test data mechanism (WSI-WSO) or the parallel test data mechanism (WPI-WPO) sometimes referred to as TAMs, if present, is being utilized. The following modes can be distinguished: Normal, Serial InTest, Serial ExTest, Serial Bypass, Parallel InTest, and Parallel ExTest.

The instructions that establish 'serial' test modes will be discussed in detail in order to provide an understanding of the structure and the reasons behind the instructions. A similar reasoning applies to the optional instructions that establish 'parallel' modes. Table 5.1 provides an overview of the mandatory instructions [11].

Table 5.1: *IEEE 1500 Wrapper Mandatory Instructions.*

Instruction	**Mode**	**Type**	**Description**
wBYPASS	Normal + Serial Bypass	Mandatory	Wrapper allows functional mode, WSI-WSO connected through WBY
wEXTESTS	Serial ExTest	Mandatory	Test of core external circuitry through WSI-WSO
wPRELOADS	Other	Conditional Mandatory	Loads data into dedicated shift path of WBR (if existent)

The mandatory wBYPASS instruction establishes two non-conflicting modes, viz. Normal mode and Serial Bypass mode. This instruction configures the wrappers cells such that they allow the normal functional operation of the core. In addition, it connects the WBY in between WSI and WSO, such that the serial test-access mechanism can be utilized to provide test access to other cores (Figure 5.4).

The instruction wEXTESTS accomplishes controllability to and observability of circuitry external to the core. The serial version of this instruction is mandatory to allow for testing of the external logic. The position of the external logic in a SoC is shown in Figure 5.8. It is unknown beforehand how many wrappers from separate cores will be needed to test the logic external to the cores in a SoC. The serial version of this instruction minimizes the risk of not having sufficient primary inputs and outputs to activate all of the wrappers. In fact, all of the wrappers can be concatenated into one chain if needed. It must be remembered that SoC design is not just simply plug and play with cores, but normally requires a lot of external (glue) logic. Some claim that depending on the application close to 50% of the SoC is glue logic.

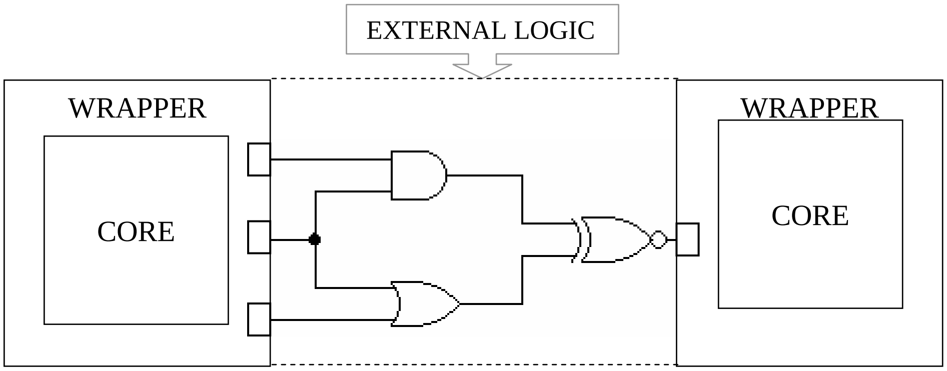


Figure 5.8*: The position of the external logic in a SoC.*

IEEE 1500 does not define core tests; the IEEE 1500 wrapper only focuses on allowing accessibility to execute tests with regards to cores. The main core test instruction, wCORETEST, is meant to be sufficiently flexible to allow any core test to be executed.

A minimal implementation of an IEEE 1500 compliant wrapper has no parallel test-data ports. In this case, the wrapper provides limited test-access bandwidth via WSI and WSO only. However, there are cases where more bandwidth is simply not required. Examples of such cases are cores with built-in stimulus generators and response evaluators [11].

5.4.10 An IEEE 1500 Compliant Wrapped Core

In order to make a core IEEE 1500 compliant, a wrapper is instantiated and added to the core. A program written in the Core Test Language (CTL) has to be created that describes the tests for the cores at the wrapper boundary [19]. A core plus wrapper are depicted in Figure 5.9.

It must again be noted that the IEEE 1500 only defines the behaviour of the wrapper and leaves the implementation free. The wrapper in Figure 5.9 allows the IP core to be connected to the serial Test Access Mechanism (TAM) via WSI and WSO as well as to a parallel (3-bit wide) TAM via WPI and WPO.

The wrapper provides six instructions, supporting normal mode and connections to both serial and parallel TAM. Mode setting through the WIR actually means setting the controls of the multiplexers of the wrapper [20]. The test program that is written in CTL consists of five parts namely the Pattern file, Initialization, Condition, Stimulus and Response [21].

The IEEE 1500 consists not only of a core test wrapper but also includes an information model. The wrapper consists of some additional on-chip hardware around an embedded core, which enables the core to be tested independently from the rest of the chip as was already discussed. The information model standardizes the test-related information that a core provider needs to transfer to the user. The following are some of the guidelines of the information model that is relevant for delay-fault testing:

- All information related to the test wrapper must be supplied in the CTL format if it supports the definition of such information. This information should not rely on user specific extensions or functions.
- All IEEE 1500 wrapper pins should be described in CTL.
- Relevant input and output safe states should be provided for all possible test operating modes.

- Certain signals such as clocks, test-mode signals and Set, Reset and Clear signals are assumed to be at a certain state at the beginning of every test protocol for sequences to be valid.
- Every core should come with at least one definition of the internal test mode of the core in CTL. If the logic model of the core is not available then the test patterns should be provided using CTL.
- All bi-directional and differential core I/Os shall have wrapper cells built in.
- All test patterns must be supplied using the CTL dialect of the STIL [1] test pattern language. This specifically excludes the use of the foreign pattern construct.
- It is recommended that all timing information in the CTL blocks be specified with acceptable margin to allow maximum flexibility for event placement.
- Test patterns shall use test protocols in CTL such that the protocol does not assume that consecutive test patterns are (scan) overlapped.
- Test setup patterns should be separated from the test data patterns.
- There should be a sufficient set of patterns supplied with the core to ensure that every test setup and every protocol is used at least once.
- If the full internal test pattern set is not supplied with the core, the complete pattern set must be made available to the device/chip manufacturer or testing company to use during test.
- Fault grading results for all internal patterns and/or pattern bursts should be provided. This data, in CTL format, includes the total number of faults, the number of faults detected, the number of redundant faults, the number of ATPG untestable faults and the number of possibly tested faults.

Philips already has a full tool-support for this standard and a transfer path has been started with several vendors of cores [22].

5.5 Testing an Embedded Core for Delay Faults

As was stated in the introduction of this chapter, the increased usage of embedded cores necessitates a core-based test strategy in which cores are tested separately or in parallel. The IEEE 1500 standard provides an environment which is able to test embedded cores separately or in parallel [23]. Earlier in this thesis it was explained in detail why delay-fault testing has become increasingly important, especially in high-speed digital circuits. This is the motivation for developing a technique to test embedded cores for delay faults.

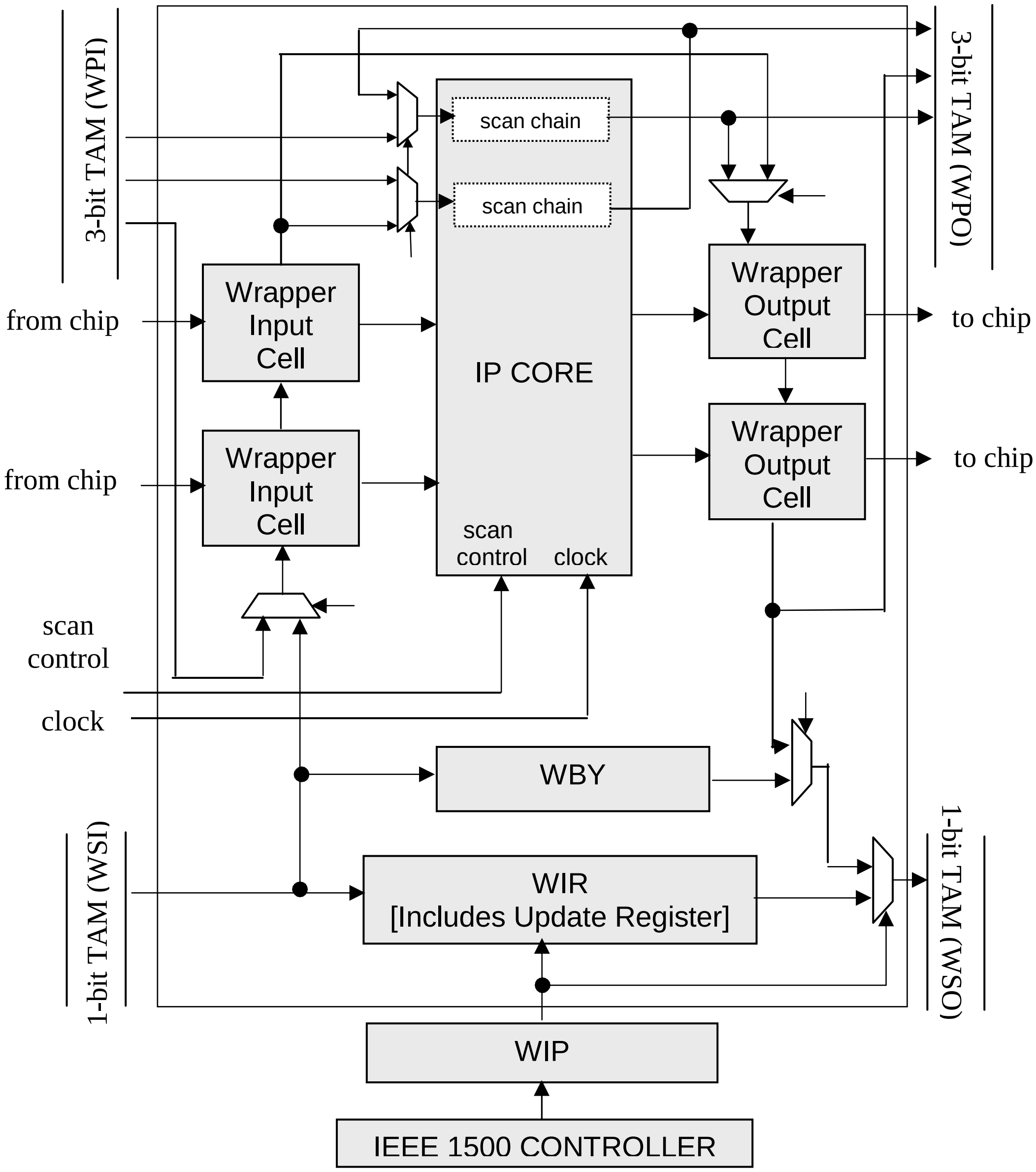


***Figure 5.9**: Set-up of an IEEE 1500 compliant wrapped core.*

As always it is essential to keep the test costs to a minimum and to reduce test costs. An interesting alternative is to use the Oscillation-based Test Methodology (OTM) [24]. It is based on the adaption of the CUT into an oscillating device and the measurement of indirect parameters.

In a combinational circuit, the path that has the longest propagation time from a primary input to a primary output, called the critical path, determines the operating speed of the circuit [4]. Other paths may have a much shorter propagation time and therefore a parametric variation in their delay value may not affect the circuit operating speed unless the changes make their propagation time longer than the critical-path delay.

However, even a very small increase in the critical-path delay will slow down the operating speed of the circuit. Also in a sequential circuit, the system is free of timing failures if every combinatorial path between two memory elements propagates its signal in less time than the interval of the operating system clock. In other words, the input signal of every memory element in the system should have a stable signal before the arrival of the active clock edge.

In our application, the OTM will be used as part of the solution to be able to detect delay faults in high-speed digital cores.

5.6 Enhancements for the Wrapper

To be able to use the oscillation test method to test for delay faults in embedded cores with our proposed method, one requires the IP core providers to supply delay data and partially specified test vectors for a set of representative paths in each core. For each path selected for testing in a core, one requires the following information:

- the input and output of the path
- the delay value of the path
- the test set for all inputs of the core required to sensitize the selected path.

The paths that will be tested for delay faults are a set of paths that are within a certain delay range and a set of the longest paths through the core. As a result of time optimization, nowadays, almost all paths will be critical paths, as a result of delay balancing in circuits. Thus, core specification should define core paths in a way that enables all such paths to be derived. As the IEEE 1500 standard only defines the behaviour of cells and not the implementation, it therefore allows for extension of the functionality of the wrappers cells to assist in the delay-fault testing of the wrapped core.

5.6.1 The Enhanced Wrapper Input Cells

The enhanced wrapper input cell has the added functionality that makes it possible to select the cell for delay-fault testing. Each input cell will be assigned a unique address. This address can be loaded into the added Cell Address Register (CAR) in the cell via the scan-in path using the control signal. The CAR consists of an address register (ADR) and an update register. The update register will be refereed to from now on as the CAR.

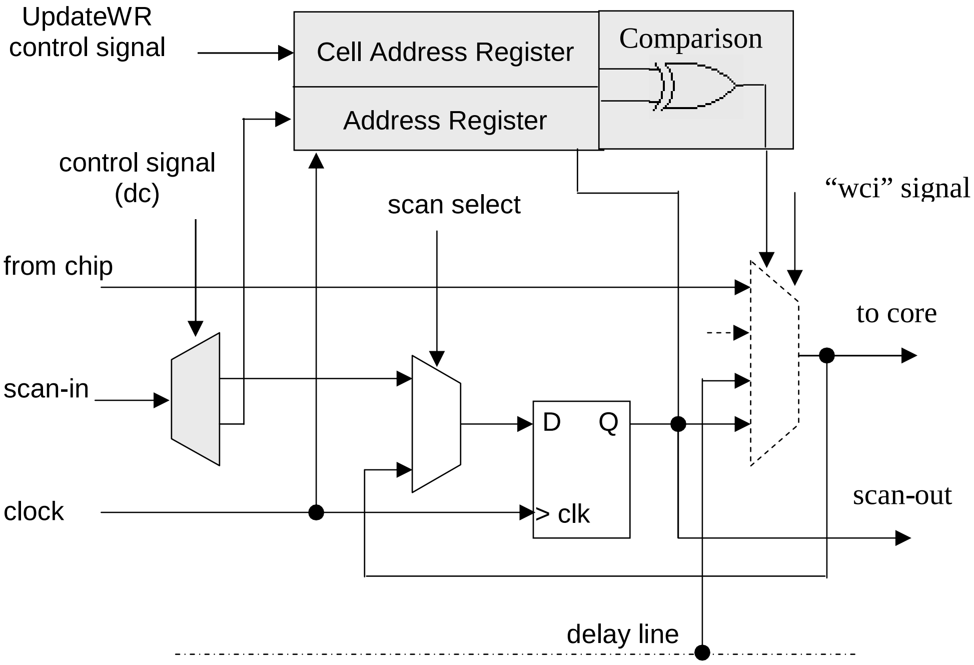


***Figure 5.10**: Set-up of the enhanced input cell.*

The enhanced input cell is shown in Figure 5.10. In the figure, the shaded elements are new to the cell and the elements depicted with a dotted line have been modified. When the unique address is loaded into the ADR it must be updated in the CAR. The UpdateWR signal with a corresponding instruction in the WIR will be used to update the CAR. This loading and updating of the unique cell addresses can be executed while the wrapper is in normal mode and it will therefore not add any time with regard to downtime for testing.

When the cell address that is necessary to be used for delay testing is loaded into all address registers of the cells, the value will be compared to the unique address of the cell and only one cell will have a match between the ADR and CAR and will therefore be selected. This match will produce a signal from the ADR and together with the existing signal "wci" is used to select the multiplexer to allow the output signal from the selected output cell to be connected to the input of the core. The same method as applied to the input cells has been used in the output cells regarding the use and loading of the unique cell address.

5.6.2 Enhanced Wrapper Output Cells

The enhanced wrapper output cell has similar added functionality to that of the enhanced wrapper input cell. Each output cell will also be assigned a unique address. The loading and updating of the unique cell addresses will be done in the same way as with the enhanced wrapper input cell.

The difference with the standard output cell set-up is that the signal provided from the address register, if there is a match between the unique cell address and the address register, is used to enable a tristate buffer to be able to link the output of the core to the specified input of the core via the wrapper input cell. The architecture of the wrapper output cell is depicted in Figure 5.11. In Figure 5.11, the shaded elements are new to the cell.

The delay line shown in the wrapper input cell as well as the wrapper output cell is the link between the input and output cells that is used for the OTM technique. This delay line is the feedback path that is needed to create the oscillation that is used in this technique. The technique requires that the output of the circuit needs to be connected to the input of the circuit and this delay line is fulfilling that requirement. Only the selected wrapper input cell and the selected wrapper output cell are connected to this delay line. The specific sensitized critical path can therefore oscillate due to the delay line connecting the specific output cell to the specific input cell. It is clear from this why it was necessary that each cell can be selected individually. Only the selected wrapper input cell will be connected to the delay line as well as the selected wrapper output cell.

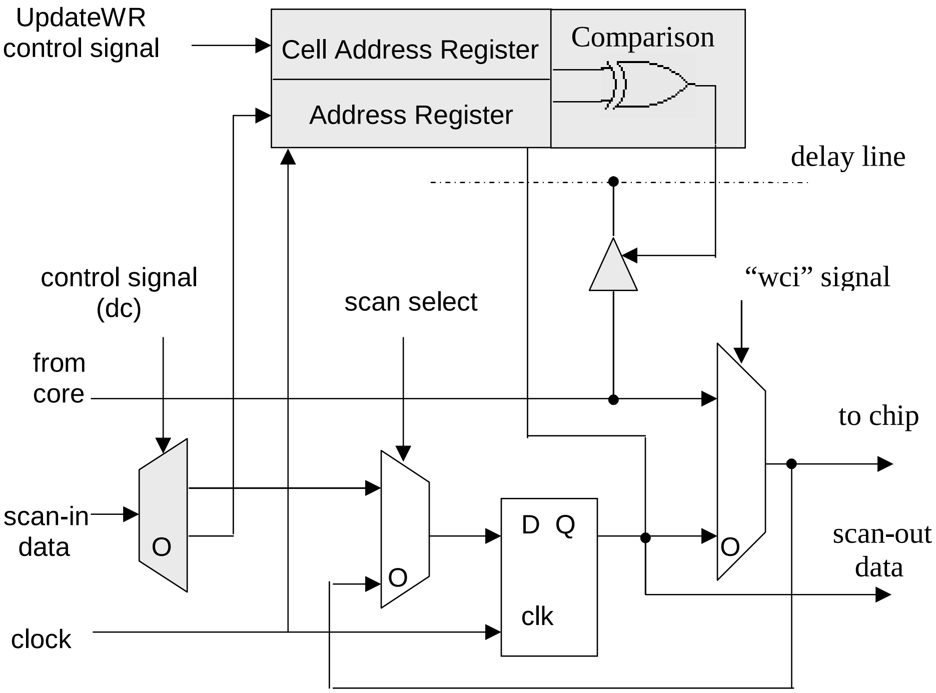


Figure 5.11: *Set-up of the enhanced output cell.*

5.6.3 Enhanced Test Wrapper

The modified wrapper architecture is shown in Figure 5.12. The enhancements to the standard wrapper are limited to the addition of two multiplexers, one demultiplexer and an XOR gate. XOR gate number 4, in Figure 5.12, is located in the "delay line" (Figure 5.12) that is the feedback path from the selected wrapper output cell to the selected wrapper input cell. The purpose of the XOR gate is to establish either an inverting or non-inverting feedback loop. When using the OTM, it is crucial that whenever there is an even

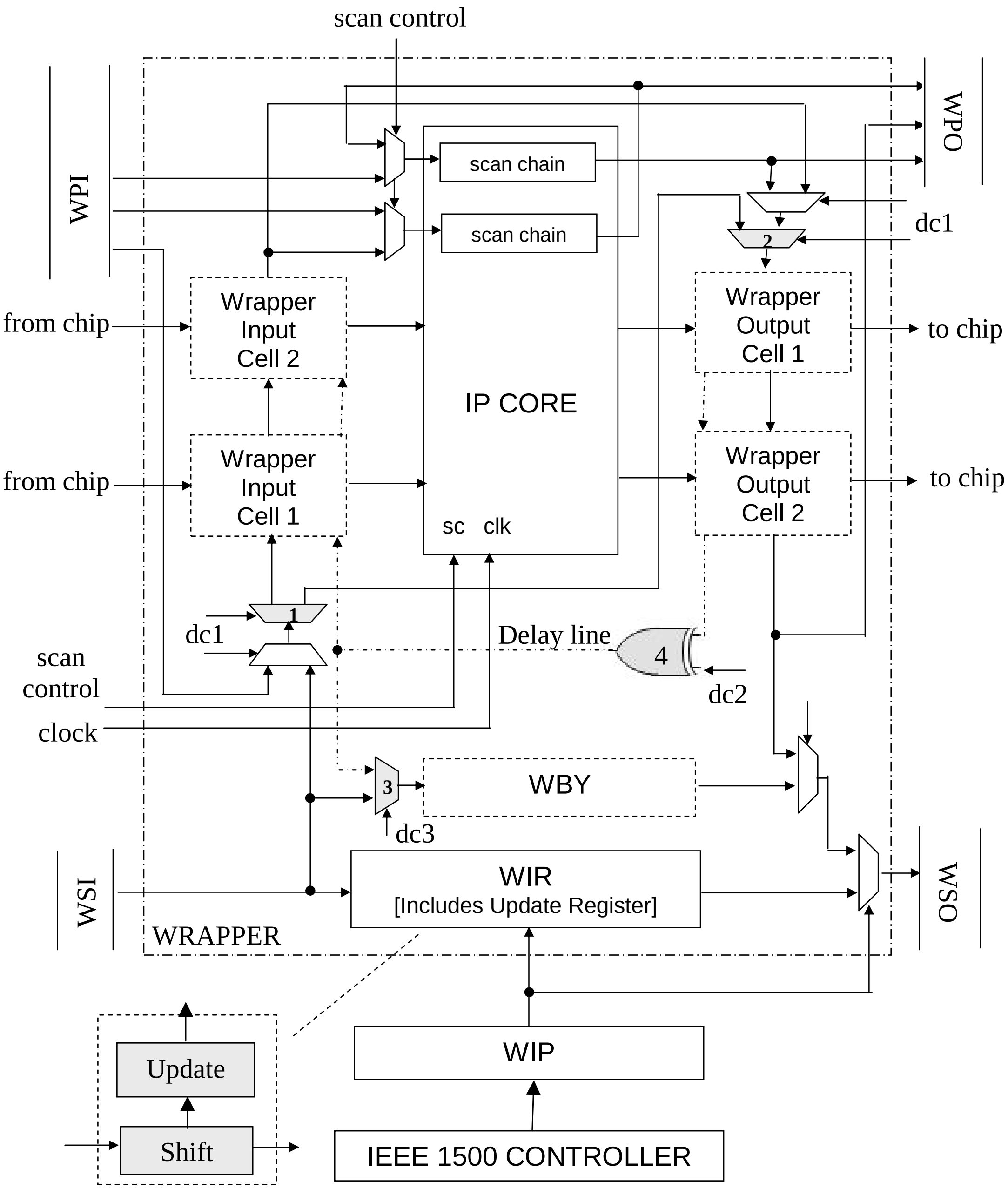


***Figure 5.12**: The enhanced wrapper architecture.*

amount of inverting elements in the sensitized critical path that the feedback path has to incorporate another inverting element to ensure oscillation. The “dc2” control signal will be used to change the XOR gate to either an inverting or non-inverting circuit depending on the number of inverting elements in the sensitized critical path.

Each of the wrapper input and output cells need a unique address for selection purposes. One can give addresses starting from “1” for the first wrapper input cell and the last number to the last wrapper output cell. This will however confuse the user because then the wrapper output cell number two can have an address of “10” depending on the number of wrapper cells. To avoid confusion, the wrapper input cells and output cells will both start at address “1”. To be able to have a different select address for the wrapper input and output cells one requires a separate path for the wrapper output cells to be able to load a different select address than that for the wrapper input cells. This can be accomplished by using a demultiplexer (number 1 in Figure 5.12), a multiplexer (number 2 in Figure 5.12) and the use of control signal “dc1”. Control signal “dc1” is therefore controlling at which moment the addresses are shifted in, when the wrapper input cells are loaded with their addresses and when the wrapper output cells are loaded with their addresses. As in the two previous figures, the shaded elements are new and the elements depicted with a dotted line have been modified in Figure 5.12 [25].

The enhancements to the test wrapper are all enabling testing for delay faults. The wrapper still contains all elements as specified by the IEEE 1500 standard and is therefore compliant to the standard. The ability of the different wrapper cells to be individually selected is an added advantage for testing purposes. This feature of the cells will be very helpful if the logic external to the core (between cores on the chip) has to be tested.

Different paths can be sensitized and the OTM can be used to test critical paths. The different wrapper cells can now be selected to be able to provide different feedback paths to ensure oscillation.

5.6.4 Application of the Wrapper Bypass Register for Test-Response Evaluation

The WBY has been used by us for the evaluation of the oscillation frequency. This is favourable in order to reduce the amount of extra elements required for this proposed testing technique. To be able to use the WBY for the measurement of the oscillation frequency, it is necessary for the WBY to be able to count this frequency. It therefore requires a register and also a counter. As registers are constructed by using flip-flops, it will be possible for the WBY to be configured as a counter.

The input of the bypass register will therefore normally be shifting data or the oscillation frequency will trigger it as a counter in the case of OTM. If the WBY is used in normal mode it will serve as a bypass for the serial TAM. This register can be a one-bit register but also a multi-bit register. The “dc3” control signal will select the input to the register as well as the configuration of the register. This signal will either configure the register as a normal shift register with input from the WSI input or as a counter with the clock as input from the oscillating signal that is taken from the “delay line” (Figure 5.12).

Therefore, if the WBY is enabled for a certain time period while being triggered by the oscillation frequency, a certain counted binary word will be stored in the WBY after this time period. This word will be relative to the oscillation frequency. Any extra delay in the selected critical path will alter the oscillating frequency and therefore also the value of the binary word in the WBY. The WBY is hence actually used to evaluate the test response of a particular critical path.

The value in the WBY can be shifted out serially and subsequently evaluated with a slow-speed tester to form part of a partial BIST system, or be evaluated on chip where the evaluation circuitry form part of a full BIST structure [26]. As a single frequency will not be possible to evaluate on, one need a range of frequencies that will be evaluated. A larger and smaller than circuit, could accomplish the evaluation of this range of acceptable frequencies.

The WBY can hence be used as a bypass register or oscillation counter, depending on the instruction in the WIR together with the WIP signals for bypassing data through the wrapper. The speed requirement of the WBY to detect these oscillation frequencies will be dealt with later.

5.7 Delay-Fault Testing

The essential part of our delay-fault testing method is the evaluation of the oscillation frequency. In the oscillation test method, the oscillating frequency must be observed and compared to the oscillation frequency of a good device to be able to detect any extra delay in the selected critical path. Any extra delay will alter the oscillating frequency and can therefore be detected.

The testing for delay faults in a wrapped embedded core can be carried out by sensitising the critical path [27] that is to be tested and selecting the appropriate wrapper input and output cell.

5.7.1 Test Control Signals

The IEEE 1500 standard clearly states the following requirements and it must therefore be adhered to while creating the wrapper test-control signals: “WIP operations shall be controlled by a single-edge triggered WRCK clock”. The rising edge of WRCK shall be used to sample WIP inputs and the falling edge of WRCK shall be used to drive WIP outputs (WS0 shift and update). The WIR and BYPASS registers shall be controlled by a single edge-triggered WRCK clock at the WIP interface. The rising edge of WRCK shall be used for shift-in and capture operations. The falling edge of WRCK shall be used for time shift-out and update operations [1].

The WIR controls the WBR modes and also selects the serial path between WSI and WSO. This path can be via WBR or WBY. The WIR can also be used to put the core into different modes. The decoding logic of the WIR provides eight (assuming an 8-bit register) registered outputs to control the test mechanism which can include a mixture of wrapper controls and OTM test-control signals.

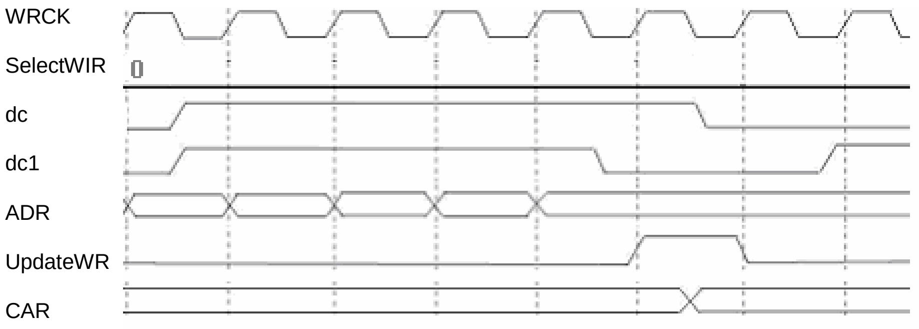


Figure 5.13: The timing diagram for loading the unique addresses into a cell.

Our delay-testing procedure makes use of four control signals: *dc*, *dc1*, *dc2* and *dc3* (see Figure 5.12 and Figure 5.13). The specific use of these signals is as follows:

- *dc* - Allows data to be shifted into the ADR
- *dc1* - Loads address data into the input and output cells
- *dc2* - Used to establish either an inverting or non-inverting feedback loop and enabling WBY for counting
- *dc3* - Selects input signal and configuration for the WBY

The SelectWIR signal must not be active to allow test data to be entered via WSI and exit via WSO. The UpdateWR is used to update the data shifted into the ADR register into the CAR register of the cells. The timing diagram for loading the unique address of the cells into their CAR is shown in Figure 5.13.

When looking at the different signals from the WIP that control the WIR, the following implementation is possible to generate these required control signals in order to control the delay-fault testing of the core. With the WRCK signal, the UpdateWR can be used to provide the “dc” signal on one of the eight registered outputs of the WIR to be able to update the unique addresses that were loaded during normal operation of the core into the cell address registers.

The “dc1” signal can be simultaneously generated on a separate registered output of the WIR that will allow the loading of the address data into the wrapper output cells. The “dc2” and “dc3” signals can be generated in a similar way to configure the delay line feedback and the WBY for response evaluation of the oscillating frequency when needed. The configurations of these control signals from the WIR are shown in Figure 5.14.

The two new instructions to facilitate delay-fault testing that will be loaded into the WIR can be defined as follows:

- LOAD_UNIQ: This instruction enables the loading of the unique addresses of all wrapper input and output cells from the WSI into the CARs. This instruction uses the “dc” and “dc1” signals.

- OSC_EN: This instruction configures the delay-line feedback characteristics as well as the WBY. This is accomplished via the generated “dc, dc2 and dc3” control signals.

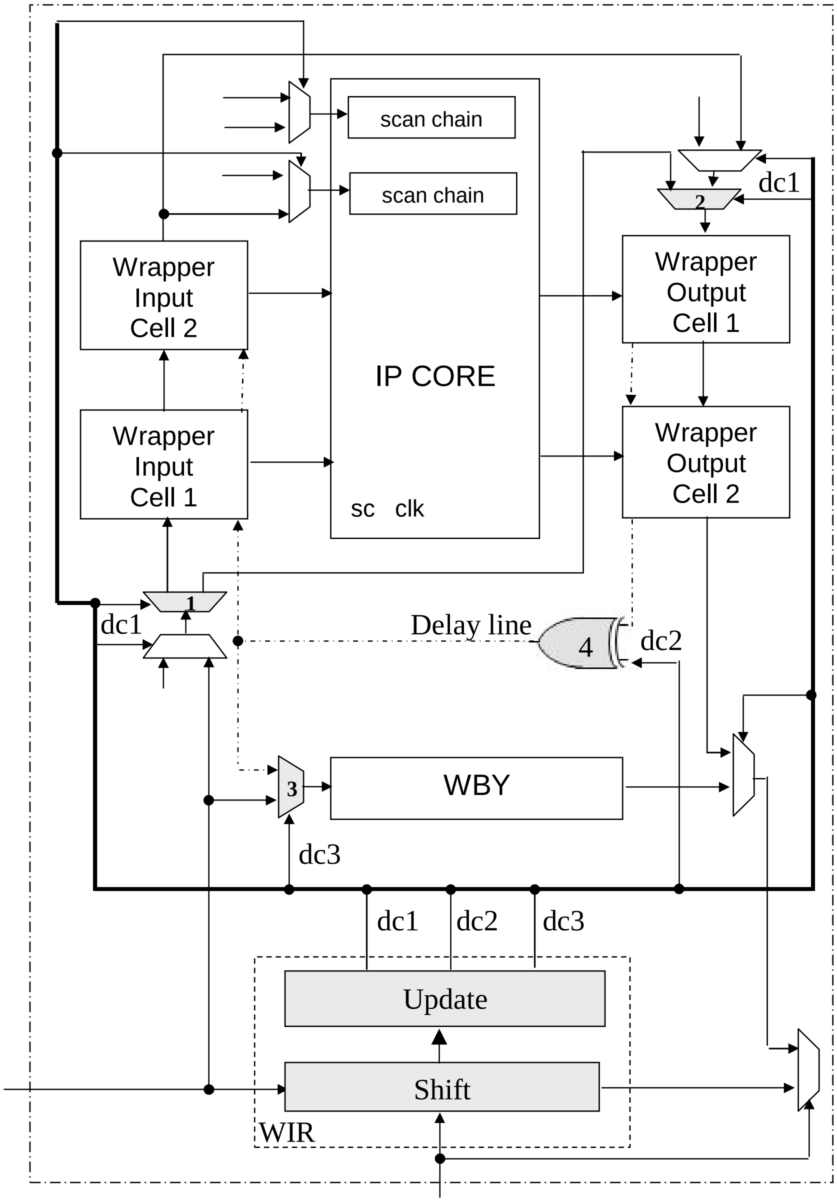


Figure 5.14: *The configuration of the flow of control signals from WIR. (Non-WIR related connections are omitted from the figure and the dark lines in the figure are indicating that those lines consist of multiple lines).*

5.7.2 Wrapper Cell Selection

The unique cell addresses can be loaded into the cell address registers by using the WPI and using the control signal "UpdateWR" to update the loaded unique address into the CAR. The wrapper is in the normal mode and the operation has no effect on the inputs and outputs to and from the chip to the core.

The first step in the delay-fault testing procedure is to load the selection addresses into the address registers of the cells. There will be one address for the input cells and a different address that will be loaded into the output cells. It is critical that only one input and one output cell are selected for the test. The addresses of the cells that have to be used must therefore be selected and will be called the selection addresses. When the selection addresses are loaded into the address registers of the cells, these addresses will be compared with the unique addresses of the cells once the "dc2" signal is active. The setup of the cells that is concerned with the selection process is shown in Figure 5.15.

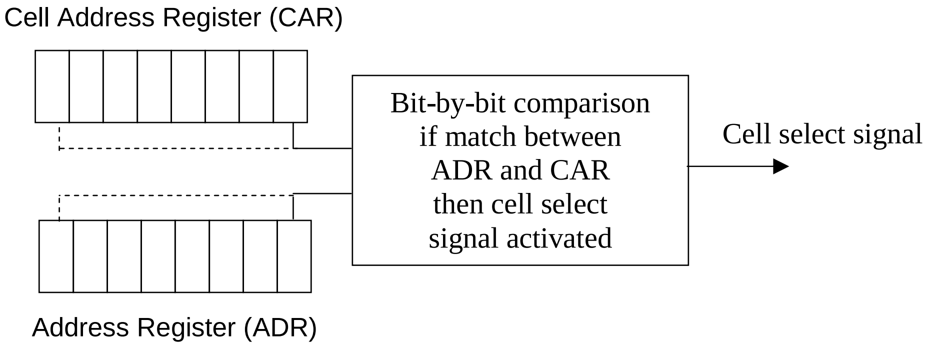

Cell Address Register (CAR)
Bit-by-bit comparison if match between ADR and CAR then cell select signal activated
Cell select signal
Address Register (ADR)

Figure 5.15*: The setup for the wrapper cell selection.*

The number of bits required in the CAR and ADR depends on the number of input and/or output cells. This is because each cell must have an unique address and therefore sufficient bits to provide unique addresses for all cells are needed. This will result in an activation of the select lines of only one input and one output cell (Figure 5.16). In our example, input cell number 2 (Figure 5.14) acts as the beginning and output cell number 1 as the end of our selected critical path in the core. Once the cells are selected and they are linked from the output cell to the input cell, one is able to perform the oscillation test to detect delay faults. The timing waveforms showing the different cells being selected are depicted in Figure 5.16, input cell 1 and output cell 2 are not selected.

The wrapper input cell is primary required for the transport of data from the chip to the core. These cells also provide access for the load-in or load-through of data. As explained earlier, the enhanced cell has also the ability that a unique address can be stored in the CAR and then be compared with the selection address in the ADR for delay-fault testing. These signals come from simulations of the example for the wrapper shown in Figure 5.14. In Figure 5.17, the timing and signal set-up in the case of the initial loading of the unique addresses are shown. These simulations were carried out in Modelsim. These same timing signals were used in our implementation of the wrapper to a microcontroller core that will be discussed in the next section.

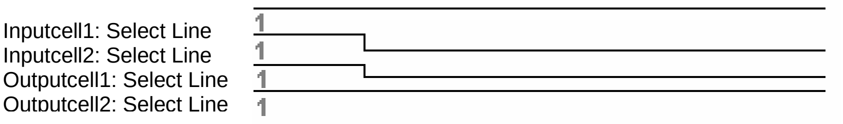


***Figure 5.16**: Wrapper input and output cell select lines.*

The results show that the unique address only gets updated into the CAR from the ADR once the UpdateWR:CAR signal becomes active (see Figure 5.17 lines b and c). The two wrapper input cells (1&2) now have two unique addresses. The same sequence is shown for the two wrapper output cells (1&2), and signal “dc1” is used to shift data into the wrapper output cells in order to load the unique address faster. Note that an instruction must first be loaded into the WIR and be updated in the WIR to allow the loading of the unique address for the input and output cells. This updating of the WIR is initiated by the UpdateWR: WIR signal as can be seen in Figure 5.17 at line number 1.

As mentioned earlier, all these operations can be done while the wrapper is in normal mode. Once the unique addresses of the input and output cells have been loaded into the CAR of each cell, the cells are ready to be selected for the delay-fault test. Only one wrapper input and one wrapper output cell must be selected. The output of the wrapper output cell will be connected to the input of the “delay line” (Figure 5.12) and the output of the delay line to the input of the wrapper input cell. This then provides the feedback loop that is required for the oscillation test method. If an inverting feedback is required then signal “dc2” will provide it via the added XOR gate in the delay line.

The only element in the feedback path that adds to the delay time of the critical path is the XOR gate, the multiplexer in the wrapper input cell and the buffer in the wrapper output cell. As this is a constant delay value, it can be taken as part of the critical path and will therefore not influence the delay-fault detection. As the size of these added delays are small in relation to the oscillating frequencies, it will not have a large influence on the oscillating frequency and therefore not a major effect on the delay-fault detection.

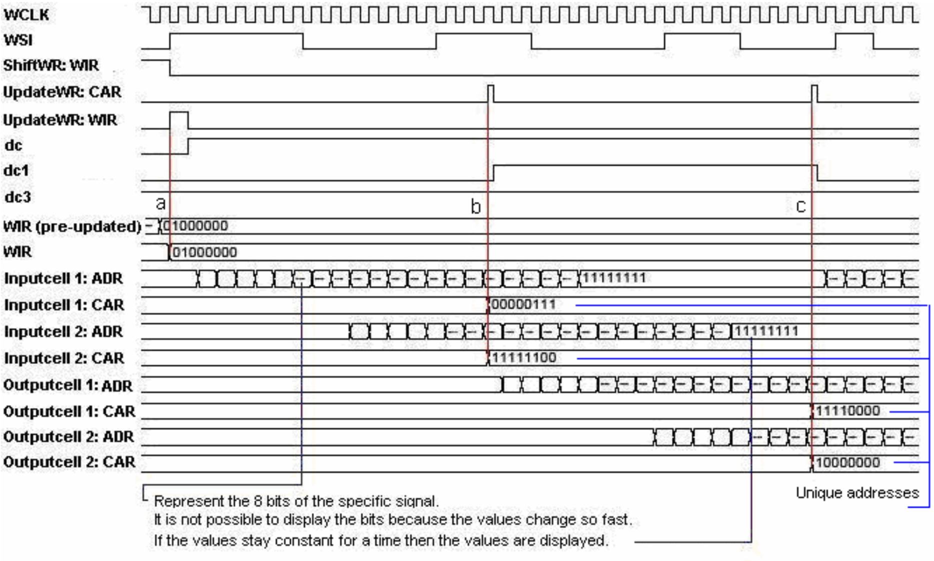


***Figure 5.17**: Wrapper input and output cell selection timing diagram.*

5.8 Simulation Results of the Enhanced Wrapper in Modelsim

The main inputs to a simulation program are:

1) a circuit description;
2) stimuli and;
3) user commands.

The enhanced wrapper circuit is being described using a hardware description language, in our case VHDL.

Using VHDL offers a number of advantages. Because it is supported by most companies that offer digital hardware technology, VHDL provides design *portability*. A circuit specified in VHDL can be implemented in different technologies, without having to change the VHDL specification. Design portability is an important advantage because digital circuit technology changes rapidly. By using a standard language, the designer can focus on the required functionality of the desired circuit without being overly concerned about the details of the technology that will eventually be used for implementation.

The functional behaviour of the different components of the wrapper have been simulated to verify their correct operation. The components were subsequently combined to form the complete wrapper and were again simulated to verify the correct operation of the entire wrapper.

The Modelsim simulator is a functional-behaviour simulator. The architecture (interconnects of functional modules such as adders, multiplexers, decoders, multipliers, registers, etc.) is considered. The functions of the modules are expressed in programming-language constructs. No implementation details are assumed to be known. Although no processing or propagation delays of signals are considered, signal changes are normally assumed to be synchronized with respect to some clock signal. This makes the simulation of a module similar to an input to output data transfer, even though there may not be explicit registers at the ports of the module. Therefore, this level is sometimes referred to as the register-transfer level (RTL.) A significant advantage of this level is that stimuli generation and response checking functions can also be blended with the circuit model as additional VHDL code. The code that generates stimuli is normally referred to as the test bench.

5.9 Implementing the Enhanced Wrapper

5.9.1 The Eight-Bit Microcontroller Core

A VHDL description of the enhanced wrapper architecture has been developed in a Modelsim environment. The different parts of the wrapper were developed separately as VHDL components for reuse in follow-up projects.

The proposed enhancements were implemented with regard to an 8-bit microcontroller core [28]. The core (in VHDL code) was synthesized making use of Cadence [29] as the static-delay calculator and using the “umcl18u250t2” technology library which is 180nm technology [30]. The longest path in the core was synthesized to be 7.17 ns. This will therefore be the most critical path in the µC as it will decide the operating frequency being the slowest path in the core. This delay originates from input data line number 7 via the core to the address output register-line number 15.

The enhanced wrapper including the µC core was implemented in VHDL and the correct operation verified. One problem that had to be overcome was that the microprocessor output lines only get updated on the rising edge of the clock pulse. Therefore a test routine was developed to ensure that the output lines were updated constantly while the clock is active high to ensure that the oscillation test method could be used.

The original code was changed by allowing the microprocessor output logic to be updated constantly while the “dc1” signal and the clock are both active (core in test mode). Whenever the “dc1” signal is not active (normal operation of core) then the output logic is only updated on the rising edge of the clock signal.

5.9.2 Simulation Results

The simulation results of the OTM approach are shown in Figure 5.19. The results show how the WBY acts as a counter to measure the oscillating frequency. The selection of the wrapper input and output cells are carried out in a similar way to that previously explained for the enhanced wrapper.

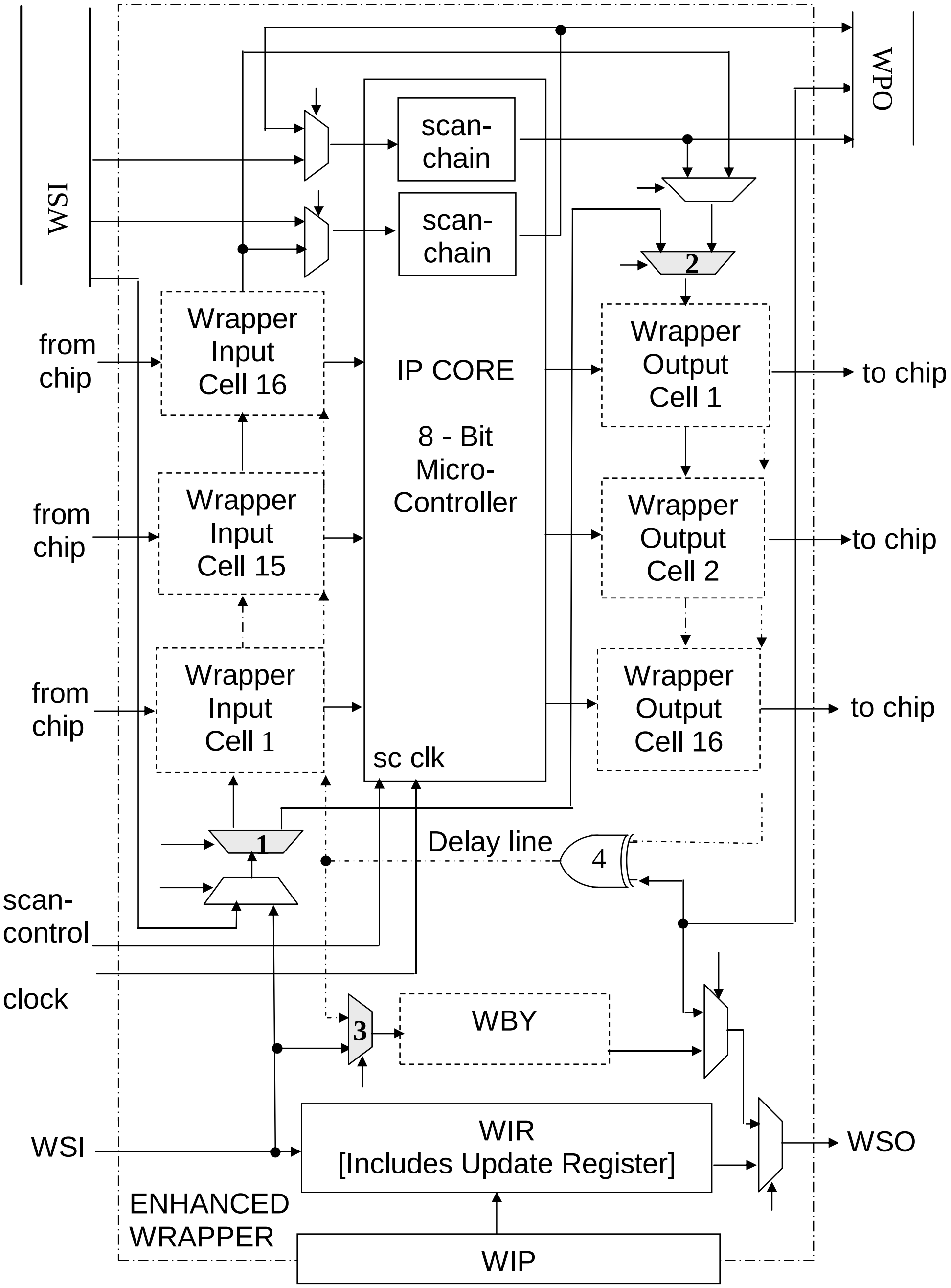


***Figure 5.18**: Enhanced wrapper for delay-fault testing of embedded cores.*

In Figure 5.19, input cell number 2 and output cell number 1 are selected and the figure clearly shows that oscillation occurs between these two cells and through the core. Before the oscillation test starts, the WBY is being reset. Signal "dc" together with "dc3" enables the WBY for a certain time period and it counts the number of pulses from the oscillating frequency in that time frame. It must be remembered that all the control signals for the delay-fault testing will be generated via the WIR. The OSC_EN instruction will be used to configure the WIR for the testing process. The final value in the WBY can be shifted out serially and evaluated on-chip or with a slow-speed tester. The WBY can then be used as a bypass register or oscillation counter depending on the instruction in the WIR together with the WIP signals for bypassing data through the wrapper [31].

An extra delay of 400 ps was added to the path through the core by adding a delay module on the path from the input logic line number 7 to the address output register line number 15. This delay module is a VHDL module as the complete core is in VHDL code. The added delay resulted in the value of the digital number in the WBY to change from 35 to 31 (decimal representation). This indicates that the added delay-fault of 400 ps has been detected. This therefore proves that delay faults, that are a combination of path and gate delays, can be detected with this technique. The 0.4 ns extra delay, on an initial critical path delay of 7.17 ns represents a detectable delay of 5 %.

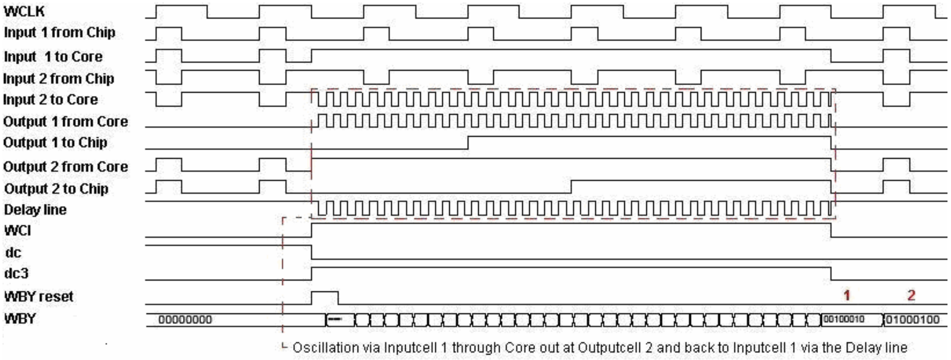


***Figure 5.19**: The delay-fault testing timing diagram.*

The simulation results show that the core is in normal operating mode until the control signals dc and dc3 are activated. As previously showed and explained, the specific cells for the test are pre-selected while the core was in the normal operation mode and the µC outputs are updated constantly to ensure that the oscillation test method can be used.

The oscillation through the core is visible as well as the operation of the bypass register which is triggered by the oscillating signal. The oscillation occurs via the selected input cell 1, then through the µC core, out at the selected output cell 2 and then back to input cell 1 via the delay line as shown in Figure 5.18. When the control signals dc1 and dc2 get deactivated, the oscillation stops and the counting result in the WBY is visible in Figure 5.19 at number 1. In Figure 5.19, at number 2 the result in the WBY gets shifted out via the WSI. The result also shows that the core is put into the normal operating mode after the test.

5.10 The Effect of Elements in the Feedback Path on the Test Technique

The question can arise as to what extend the oscillating frequency is related to the actual operating speed of the circuit. The maximum operating speed of the circuit is dependent on the critical path having the longest delay time. In other words the slowest path through the core determines the maximum operating speed of the core.

The enhancements to the wrapper do add some elements into this critical path for either providing an inverting feedback line or multiplexers for selection of specific signals and data. The critical path that is measured is, hence including extra elements and the oscillating frequency is of course dependent on the complete path. The operating speed of the core is however still related to the oscillating frequency but it is now also related to a factor that is introduced by the extra elements. The added elements are the same and therefore the relationship between the operating speed of the core and the oscillating frequency will also remain constant for a specific core and wrapper.

The wrapper is a test access wrapper and therefore it is used for testing purposes, like (in our case) for the detection of delay-faults. One has to remember that these extra elements are only present in the feedback path in the test mode and the core can still operate at its normal operating speed in the normal mode. The enhancements to the wrapper have minimal effect on the performance of the core in normal mode. This is possible because the data only pass through two extra multiplexers which have a minimal degradation of the data integrity.

The additional circuitry that was added to enhance the wrapper must be tested to ensure that it does not provide false information regarding delay-faults. The added multiplexers (numbers 1 to 2 in Figure 5.18) can be tested by initialising a cycle for loading the unique addresses into the cells and enter the data via the WSI and evaluate the data exiting at WSO. The multiplexer (number 3 in Figure 5.18) at the WBY can be tested by putting the wrapper in the bypass mode and evaluate the data exiting at WSO. The only remaining element to be tested is the XOR logic gate (number 4) used in the "delay line" in Figure 5.18.

For the testing of the XOR logic gate the same setup as for testing the first two multiplexers is needed. If an input and output cell is selected then data can be send via the cells to the XOR in the "delay line". The output of the XOR logic gate can be evaluated at WSO by sending it via WBY. The "dc2" signal can then be used to invert the output of the XOR logic gate. This change of state of the XOR logic gate output can be monitored at WSO.

5.11 Implementing the Test Technique to Sequential Logic

In Chapter 4, the use of the OTM was explained with combinational circuits. An 8-bit digital adder was used to demonstrate the practicality of the OTM method.

In a sequential circuit (like the micro-controller core that was used earlier), the system is free of timing failures if every combinational path between two memory elements propagates its signal in less time than the interval of the operating system clock. In other words, the input signal of every memory element in the system should have a stable signal before the arrival of the active clock edge.

A simplified example of a sequential circuit is shown in Figure 5.20. To make sure that the system is fault-free in terms of delay faults, the clock period should be larger than the sum of the propagation delay of the input flip-flop FFi

(t_{PDFFi}), the propagation delay of the combinational circuit (t_{PDCC}), and the setup time of the output flip-flop FFo (t_{SUFFo}) [24].

$$T_{CK} \geq (t_{PDFFi} + t_{PDCC} + t_{SUFFo}) \qquad \textbf{(5.1)}$$

The above relationship can be rewritten as follows.

$$t_{PDCC} \leq (T_{CK} - t_{PDFFi} - t_{SUFFo}) \qquad \textbf{(5.2)}$$

Therefore, a given delay increase in a path may result in malfunctioning of the circuit but the same delay increase in another path may not affect the circuit functionality and performance.

Some sequential circuits include some feedback in their complex configurations. State changes therefore occur in direct response to signal changes on the primary input lines, and different memory elements can change state at different times.

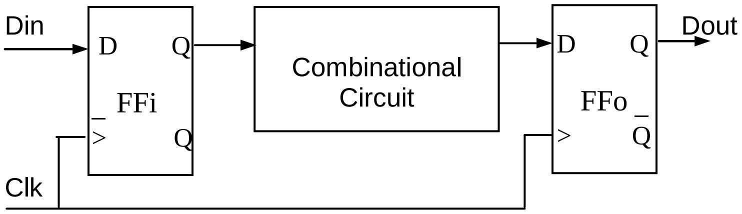


Figure 5.20: *A simple sequential circuit.*

A delay test checks for excess propagation delay through a combinational path in a sequential circuit. A rising or falling signal transition is propagated through the combinational path, which starts at a primary input or a flip-flop and terminates at a flip-flop or primary output. If the combinational path terminates at a flip-flop, the captured logic state, i.e., the fault effect, is propagated to a primary output for observation. If the combinational path terminates at a primary output then no fault effect propagation is necessary [32].

In Chapter 3 of this thesis a scan flip-flop was discussed and used for delay-fault detection. This scan flip-flop can be used in sequential circuits where necessary to interrupt feedback cycles and hence improves the delay test coverage for the sequential circuit.

5.12 Conditions for Testing of Delay Faults

The conditions for testing of a delay-fault in sequential circuits can be divided into three parts:

- combinational path activation,
- sequential path activation,
- fault effect observation.

These three parts will now be discussed.

1. **Combinational Path Activation:** This refers to the existence of a pair of vectors that, when applied at the inputs of the combinational logic, can propagate a signal transition through the target path. It can be shown that the vector-pair must 1) produce a transition at the origin of the path and 2) sensitize the entire path during the second vector. When no such vector-pair is possible, the path is called a *combinationally false path* [33]. Such paths are also called non-robustly untestable or singly-untestable paths [34]. For reliable testing, however, a vector pair must also satisfy additional conditions such that the transition propagation will be guaranteed in spite of interference from other off-path signals. These tests are referred to as *combinationally robust* [33]. Notice that the absence of a combinationally robust test does not make the path false. A prerequisite condition for any kind of sequential path delay test is that the path must not be combinationally false.

   In the present work we assume using the OTM for detection of the delay-fault in a combinational path, instead of using a vector-pair as earlier explained in Chapter 4 of this thesis.

2. **Sequential Path Activation:** Since some inputs of combinational logic are controlled by flip-flops, not every vector pair may be possible. The success of *sequential-path activation* requires the existence of a primary input sequence of vectors that, when applied with a given (or unknown) starting state, will produce two vectors at the input of the combinational logic to propagate a transition through the target path. In a specific

delay test methodology, called the *variable-clock method* [33], the clock is slowed down during this sequence, with the exception of the last vector, which is driven with the rated clock period. The slow clock makes the sequence robust with respect to delay variations in the circuit. The sequential path activation condition cannot be satisfied for a combinationally false path.

In this thesis, the Oscillation Test Method is used to establish the presence of a delay-fault. The target path has to be sensitized. The oscillation frequency of the sensitized path will be monitored to detect the delay-fault (if present) as discussed earlier in this work.

3. **Fault-Effect Observation:** Once the path is sequentially activated, the test will be complete if the path destination is a primary output. When the destination is a flip-flop that is not directly observable, the content of the flip-flop should be propagated to a primary output. The observation of the oscillation frequency will be done to be able to detect any delay faults present in the combinational or sequential path.

The main reason for low coverage of delay faults in a sequential circuit is the controllability of state variables as was discussed earlier. This is similar to the cause of low coverage of stuck-at faults in sequential circuits. To ensure better fault coverage in a sequential circuit the combinational logic has to be synthesized to eliminate combinational false paths in the sequential circuit [32].

To deal with this issue when testing the behaviour of sequential circuits, certain assumptions are made. The model for these assumptions is known as the Huffman model [35]. The Huffman model greatly reduces the complexity of the sequential behaviour of a circuit by restricting all propagation delays to the feedback lines. In this case the number of the feedback lines is assumed much less than the number of gates.

Taking the previous results from the simulations with the microprocessor core into account, applying the oscillation test technique to a sequential circuit is viable.

5.13 Conclusions

The quest for the detection of delay faults is driven by the increased performance requirements of SoC circuits. Our proposed architecture is primarily aimed at the detection of small delay-faults in embedded cores. VHDL simulation results proof that our proposed enhanced wrapper will be able to test embedded cores for delay faults. This advantage of being able to do delay-fault testing of embedded cores is justifying the extra chip area needed to implement it.

The area overhead introduced by the standard IEEE 1500 wrapper amounts to less than 3 % of the core area, according to case studies done, and less than 1 % for the wrapper controller in complex circuits with more than ten million gates. The performance reduction in terms of frequency lost is reported to be less than 1 % due to the introduction of the standard IEEE 1500 wrapper. The increase in power consumption depends on the amount of cores in the SoC and is therefore difficult to determine and therefore the extra power consumption due to the introduction of a wrapper to the core is reported as less than 5 % [36-38].

The input cells and output cells were enhanced to allow the cells to be individually selectable and this is an advantage to the cells and wrapper. The advantages of individual cell selection can be utilized when delay-fault test techniques are developed for the external logic between cores. The different cells of the cores can then be used to develop these test techniques because they can then be used individually when selected.

Another advantage is the re-use of the WBY for measuring the oscillation frequency. The proposed enhancements to the IEEE 1500 wrapper will make it possible that delay-fault testing of various cores can be done in parallel in order to reduce test time.

The complete enhanced wrapper that includes the input and output cells, were verified by VHDL simulations to ensure correct operation within the IEEE 1500 guidelines. An extra delay was added in the example core in the critical path and the results show that our enhanced wrapper was able to detect this added delay-fault.

It was also shown that the oscillation test method can be applied to sequential circuits as well. Therefore, if the critical path in the core contains sequential logic it can still be tested with the oscillation test method.

5.14 References

[1] E.J Marinissen, R.Kapur, M. Lousberg, T. McLaurin, M. Ricchetti, and Y. Zorian, ”On IEEE P1500’s Standard for Embedded Core Test”, *Journal of Electronic Testing: Theory and Applications (JETTA)*, vol. 18, no. 4-5, August 2002, pp. 365–383.

[2] Y. Zorian, “Test Requirements for Embedded Core-Based Systems and IEEE P1500,” *Proceedings of IEEE International Test Conference (ITC)*, 1997, Washington, DC, USA, pp. 191–199.

[3] T. McLaurin and S. Ghosh,” ETM10 Incorporates Hardware Segment of IEEE P1500”, *IEEE Design & Test of Computers*, May-June 2002, pp. 8-13.

[4] www.vsi.org

[5] http://www.opencores.org

[6] E.J. Marinissen, S.K. Goel, and M. Lousberg, “Wrapper Design for Embedded Core Test ”, *Proceedings of IEEE International Test Conference (ITC*), 2000, Atlantic City, USA, pp. 911–920.

[7] Y. Zorian, E.J. Marinissen and S. Dey, “Testing Embedded-Core Based System Chips”, *Proceedings of IEEE International Test Conference (ITC*), 1998, Washington, DC, USA, pp. 130–143.

[8] A. L. Crouch, “Design-for-Test for Digital IC,s and Embedded Core Systems”, ISBN 0-13-084827-1, Prentice Hall, 2000.

[9] A. Krstic et al,” Embedded Software-Based Self-Test for Programmable Core-Based Design”, *IEEE Design & Test of Computers*, July-August 2002, pp. 18-27.

[10] E. Larsson and Z. Peng, “An Integrated System-on-Chip Test Framework”, *Proceedings of Design, Automation, and Test in Europe Conference (DATE*), 2001, Munich, Germany, pp. 138–144.

[11] E.J Marinissen, Y. Zorian, R. Kapur, T. Taylor, and L. Whetsel, ”Towards a Standard for Embedded Core Test: An Example”, *Proceedings of IEEE International Test Conference (ITC),* 1999, Atlantic City, USA, pp. 616-627.

[12] L. Chen and S. Dey, “DEFUSE: A deterministic functional self-test methodology for processors”, *Proceedings of IEEE VLSI Test Symposium*, 2000, Montreal, Canada, pages 255-262.

[13] I. Ghosh, S. Dey, and N. K. Jha, “A fast and low cost testing technique for core-based system-chips”, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* 2000, Vol. 19(8), pp. 863 -876.

[14] W. C. Lai and K.-T. Cheng, “Instruction-level DFT for testing processor and IP cores in system-on-a-chip”, *Proceedings of IEEE/ACM Design Automation Conference (DAC),* 2001, Las Vegas, USA, pp. 59 -64.

[15] S. Ravi, G. Lakshminarayana, and N. K. Jha, “Testing of core-based systems-on-a-chip”, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2001, Vol. 20(3), pp. 426 -439.

[16] S. Adham et al., “Preliminary Outline of IEEE P1500 Scalable Architecture for Testing Embedded Cores ”, *Proceedings of IEEE VLSI Test Symposium (VTS*), 1999, Dana Point, USA, pp. 483–488.

[17] S. Goel and E.J. Marinissen, “Effective and Efficient Test Architecture Design for SoCs ”, *Proceedings of IEEE International Test Conference (ITC*), 2002, Baltimore, USA, pp. 529–538.

[18] V. Iyengar, K. Chakrabarty, and E.J. Marinissen, “Co-Optimization of Test Wrapper and Test Access Architecture for Embedded Cores,” *Journal of Electronic Testing: Theory and Applications (JETTA),* vol. 18, no. 2, April 2002, pp. 213–230.

[19] R. Kapur et al., “P1500-CTL: Towards a Standard Core Test Language”, *Proceedings of IEEE VLSI Test Symposium (VTS*), 1999, Dana Point, USA, pp. 489–490.

[20] R. Kapur et al., “CTL—The Language for Describing Core-Based Test,” in *Proceedings of IEEE International Test Conference (ITC*), 2001, Baltimore, USA, pp. 131–139.

[21] M. Keating and P. Bricaud, “*Reuse Methodology Manual for system-on-a-Chip Design*s”, Norwell, Massachusetts: Kluwer Academic Publishers, June 1999.

[22] http://www.research.philips.com/password/archive/23/pw23_technews.html

[23] E.J. Marinissen, “The Role of Test Protocols in Automated Test Generation for Embedded-Core-Based System ICs,” *Journal of Electronic Testing: Theory and Applications (JETTA)*, vol. 18, no. 4, August 2002, pp. 435–454.

[24] M. Arabi, H. Ihs, C Dufaza and B. Kaminska, "Digital Oscillation-Test Method for Delay and Stuck-at Fault Testing of Digital Circuits", *IEEE Transactions on Instrumentation and Measurement*, Volume: 48 Issue: 4, August 1999, pp. 798-806.

[25] H.J. Vermaak and H.G. Kerkhoff, "Delay-fault Testing of Embedded Cores using an Enhanced P1500 Compliant Wrapper”, *Proceedings of the Dutch Technology Foundation Conference on Circuits, Systems and Signal Processing (ProRisc*), 2003, Eindhoven, The Netherlands, pp 158-164.

[26] H.J. Vermaak and H.G. Kerkhoff, “Delay-fault Testing of Embedded Cores”, *Proceedings of Conference on Embedded Systems & Software (Progress*), 2003, Nieuwegein, The Netherlands, pp. 250-256.

[27] K.T. Lee and J.A. Abraham,” Critical Path Identification and Delay Tests of Dynamic Circuits”, *Proceedings of IEEE International Test Conference (ITC)*, 1999, Atlantic City, USA, pp. 421-430.

[28] http://www.free-ip.com

[29] http://www.cadence.com

[30] http://www.umc.com/english/process/d.asp

[31] H.J. Vermaak and H.G. Kerkhoff, “Enhanced P1500 Compliant Wrapper suitable for Delay-fault Testing of Embedded Cores”, *Proceedings of IEEE European Test Workshop (ETW)*, 2003, Maastricht, The Netherlands, pp. 121-126.

[32] T.J.Chakraborty, V.D.Agrawal and M.L.Bushnell, “Improving Path Delay Testability of Sequential Circuits”, *IEEE Transactions on VLSI Systems*, Vol. 8, No. 6, December 2000, pp. 736-741.

[33] T. J. Chakraborty, V. D. Agrawal, and M. L. Bushnell, “On variable-clock methods for path delay testing of sequential circuits,” *IEEE Transactions on Computer-Aided Design,* vol. 16, November, 1997, pp. 1237-1249.

[34] A. Krstic and K.-T. Cheng, “*Delay Fault Testing for VLSI Circuits.”*, *Norwell*, USA, Kluwer, 1998.

[35] J.P. Hayes, “*Introduction to Digital Logic Design*”, ISBN 0-201-15461-7, Addison-Wesley, USA, 1993.

[36] P. Bernardi et al, "Testing logic cores using a BIST P1500 compliant approach: a case of study," *Proceedings of Design, Automation and Test in Europe Conference (DATE)*, 2005, Munich, Germany, pp. 228 - 233.

[37] A. Manzone et al, “Integrating BIST techniques for on-line SoC testing”, *Proceedings of IEEE International On-line Testing Symposium (IOLTS)*, 2005, Saint Raphael, France, pp. 235-240.

[38] D. Appello et al, “A P1500 compliant BIST-based approach to embedded RAM Diagnosis”, *Proceedings of IEEE Asian Test Symposium (ATS)*, 2001, Kyoto, Japan, pp. 97 - 102.

Chapter 6

Conclusions

This thesis is focussed on delay-fault testing. The susceptibility of circuits that were developed to be used for delay-fault testing was investigated. The oscillation test methodology was taken as basis for developing a new test technique to test for delay-faults. The testing of embedded cores for delay-faults was also part of the work of this thesis. In the following sections the results of the developed test techniques as well as the results of the susceptibility analyses, will be summarized and reviewed, followed by some options for further research to improve current results.

6.1 Summary

The research work in this thesis has introduced different Design-for-Delay-Testability (DfDT) methodologies. The work was mainly concerned with the development of new test techniques to detect delay faults in digital circuits with different architectures. The architectures vary from simple combinational, to sequential, to complex SoC architectures. Different solutions were proposed as well as evaluated.

Chapter 1 and 2 have introduced the testing methodologies and trends in high-speed digital integrated circuit technology. It covers the basics and state-of-the-art delay-fault testing. It also answered the question why delay-fault testing has become increasingly important.

In Chapter 3, a new type of DfDT structure and associated BIST architecture for detecting delay faults in digital high-performance circuits has been proposed. It circumvents the requirement of an expensive high-speed tester. Furthermore, the susceptibility of the proposed structure to process- and application-induced variations has been investigated. Due to the critical timing necessary when detecting small delay faults, it is crucial to know what to expect from these variations and subsequently reduce their influence.

Chapter 4 discussed high-performance digital circuits with aggressive timing constraints which are usually very susceptible to delay faults. Research on delay-fault detection indicates a need for a rather complicated test setup together with precise test clock requirements. In this chapter, we propose a test technique based on the digital oscillation test method. The technique consists of sensitizing a critical path in the digital circuit under test and incorporating the path into an oscillation ring. The supply voltage to the oscillation ring was then varied to be able to detect delay and stuck-at faults in the path.

Chapter 5 is concerned with the continuous advances in the manufacturing processes of integrated circuits which provide designers with the ability to create more complex and denser architectures and increased functionality on a single chip. The increased usage of embedded cores necessitates a core-based test strategy in which cores are also tested separately. The IEEE 1500 is an IEEE standard which aims to improve the testing of core-based system chips. This chapter deals with the enhancement of the Test Wrapper and Wrapper Cells to provide a structure facilitating testing of embedded cores for delay faults. This approach allows delay-fault testing of cores by using the digital oscillation test method and the help of the enhanced elements while staying compliant to the IEEE 1500 standard.

6.2 Conclusions

Technology scaling challenges designers by increasing the delay uncertainty in many areas of devices. These areas include both interconnects and gate delays due to the process and environmental variations, and crosstalk. DSM designs experience a massive increase of subtle defect types such as resistive bridges, resistive contact/vias, and gate oxide shorts. These defects manifest itself as delay faults due to its influence on the circuit’s timing characteristics. As these are all given and proven facts that will not go away solutions have to be found to ease the burden of testing these new high-speed digital circuits.

Much research has already been carried out in delay-fault testing. This research includes the development of fault models like: the gate-delay fault model, path-delay faults, robust delay faults, non-robust delay faults and hazard-free delay faults. For all these approaches, the testing setup is rather complicated. For example it needs two sets of latches with precisely controllable test clocks and test-pattern pairs. These approaches are therefore time consuming and occupy a lot of silicon area. It was therefore important to look innovatively at delay-fault testing and especially the development of new techniques and architectures to detect delay-faults.

In the first part of the thesis we have looked at structures to make it possible to test high-speed circuits with low-speed testers and still ensuring good quality chips. A new type of DfT structure was developed which make use of special controlled-delay flip-flops. The complete structure including the programmable delay-line (PDL) and the programmable duty-cycle controller (DCC), the two key elements of the structure, have been described in detail and simulated in VHDL at system level and using HSPICE data for different elements. HSPICE simulation results of the structure in the fault-free case with the critical path present in the combinational logic part had a fault-free delay of 4.8 ns. An extra delay of 620ps was inserted in the critical path to emulate a possible delay-fault with the critical path now having an overall delay of 5.4 ns. The final result did show that the induced delay-fault of 620ps was successfully detected by our DfDT structure.

The DfDT structure in Chapter 4 has been designed to detect small delay faults, of which the minimum delay size is determined by design and processing parameters. In general, the effect of process variations manifests itself first in the most critical paths in the design, those with maximum and minimum delays. In order to investigate the robustness of the delay-detection architecture to these variations, it was decided to look at the variation in gate oxide thickness, threshold voltage and the relation between channel length and width (W/L). This was in order to establish the circuits' susceptibility to these process variations. The simulations show that a variation in Vth results in a 2,1 % change in delay time, the variation in tox gives a 0,9 % change and the W/L variation is resulting in a 1,4 % change in delay time.

Application-induced factors are defined as factors contributing to variations in delays that are not part of the manufacturing process, but depend on the environmental conditions. A specific design will introduce variation upon itself due to a particular circuit style chosen, layout configurations for transistors and interconnects and signal flow. When looking at application-induced variations, there are two major contributors being the supply voltage (Vdd) and temperature (T). The results showed that the major factor in delay-time variations due to application induced variations was Vdd. A regulated voltage-reference circuit was developed to correct this influence on the DfDT structure. The developed voltage reference circuit occupies 45µm^2 in our 0.35 µm CMOS process. The circuit was adapted into the CMOS process and the simulation results shown its ability in present technologies. The external supply voltage (Vsup) was taken as 5V with a 10% variation. The Δ delay has now been reduced from 7.2% to an exceptable 1.9 %.

Tolerance to process-induced skew remains one of the major concerns in the design of large-area and high-speed clock distribution networks. Having a high clock-skew, there will be a large uncertainty about the time the detection is carried out. Hence, time delay cannot be measured accurately. Both PLLs and DLLs check the phase for skew and upon detection, a correction procedure is started which uses feedback and normally takes a few clock cycles to be accomplished. We have implemented a skew-reduction circuit to provide a constant skew-corrected signal for use in our structure. This skew correction circuit does not use feedback and will only be activated during the test cycle. The method makes use of a PDL (the same design as within our DfDT architecture), which provides a number of delayed replicas of the skewed clock signal. The replicas are then evaluated to select the one that is closest to the reference clock signal. Simulations were carried out in HSPICE with this skew correction included in our DfDT architecture and the results showed that a constant clock-skew is 4.2% of the clock period and a clock-skew up to 32% of the clock period could be handled successfully.

The extension of the DfDT structure with the clock- skew reduction and voltage-reference circuit requires additional chip area. The extra components to implement our complete structure, including the skew-reduction and voltage-reference circuits consist of 3846 MOS transistors for which a chip area is needed of 0.008mm2. The increase in chip area is acceptable considering the complexities envisioned. The DfDT structure will have a very small impact on the chip performance due to it being only active during the test sequence.

Next we developed a test technique based on the digital oscillation test method to test for delay-faults. Our technique introduced a varying supply voltage in combination with the oscillation test method to detect delay faults as well as stuck-at faults in the particular sensitized critical path in the digital circuit under test. The technique connects outputs of the CUT to its inputs with odd inversion parity and applies appropriate input patterns to the unconnected inputs to sensitize paths of the circuit, converting them to oscillation rings. Next, a varying supply voltage was applied to the CUT to be able to detect delay-faults. The next part was focussing on establishing the characteristics of the circuit due to the varying supply voltage. The last part of the test was to observe the output waveform of the CUT whilst varying the supply voltage. The characteristics of the output waveform showed if any delay or stuck-at faults are present.

One has to remember that a stuck-at fault can be detected as a delay-fault with infinite delay. The effect of a change in duty cycle of the varying supply voltage was also established. By observing whether the output oscillates normally at the target frequency, one can tell whether the circuit was working properly or not.

This testing method requires simple added hardware, which can be applied externally or built internally in the circuit to be part of a BIST architecture. The area overhead for this test method is about 12% which includes circuitry to change the duty cycle.

Continual advances in the manufacturing processes of integrated circuits provide designers with the ability to create more complex and denser architectures and increased functionality on a single chip. The increased usage of embedded cores necessitates a core-based test strategy in which cores are also tested separately. The IEEE 1500 standard for Embedded Core Test is a standard of which the aim is to improve the testing of core-based system chips. The last part of the thesis deals with the enhancement of the Test Wrapper and Wrapper Cells to provide a structure to be able to test embedded cores for delay faults. Our approach allows for delay-fault testing of cores by using the digital oscillation test method and the help of the enhanced elements while staying compliant to the IEEE 1500 standard.

The enhancements were made to allow for delay-fault testing of embedded cores without needing much extra chip area. The input cells and output cells were enhanced to allow the cells to be individually selectable and is a major advantage for the cells and wrapper. The enhancements were simulated in ModelSim using an 8-bit microcontroller core. The core was synthesized and the worst-case delay through the core was simulated to be 7.17ns. An extra delay of 400ps was added in the critical path and the results show that our enhanced wrapper was able to detect this added delay-fault. One problem that had to be overcome was that the microprocessor output lines only get updated on the rising edge of the clock pulse. Therefore a test routine was developed to ensure that the output lines were updated constantly to ensure that the oscillation test method could be used.

The complete enhanced wrapper that includes the input cells and output cells were verified by simulations to ensure correct operation within the IEEE 1500 guidelines. The enhancements to the wrapper will make it possible that delay-fault testing of various cores can be done in parallel in order to reduce test time.

This thesis not only highlighted the need to detect delay-faults but provided new techniques to test digital circuits as well as embedded cores for delay faults. This thesis furthermore did supply an analysis of the susceptibility of DfDT structures to process and application induced variations as well as to clock-skew. Lastly suggestions have been provided on how to get the influence of these variations to exceptable levels for effective testing.

6.3 Recommendations for Future Research

The limits of current technology do not necessarily mean an end to progress. With a sustained and coordinated commitment to basic research by the semiconductor industry and academia the obstacles to continued advances in circuit technology can be overcome. The Nanotechnology Era will require new materials, new device structures, new manufacturing methods and hence testing approaches. Device test needs must be managed through DfT to enable low-cost manufacturing test solutions including reduced pin-count test, equipment reuse, and reduced test time. Together with the development of new DfT techniques comes the quest for fundamental research in existing and novel fault models to address emerging new types of defects.

In the first part of the thesis DfDT structures were developed and their susceptibility to variations were analysed. The structures were then modified/enhanced to compensate for variations and the results were positive towards detection of delay-faults. An actual circuit implementation in an advanced 90 nm CMOS process and the use of our DfDT structure in a BIST environment can be part of future research. This will help to establish the possibility of implementing these structures in high-speed digital circuits.

There is still much work to be done in the area of testing core-based design. The amount of circuitry within a SoC that is not part of the embedded cores is increasing fast. The possibility to test this external circuitry within SoC for delay faults, with the use of the technique and structures that were enhanced as part of this thesis, can be investigated. The added feature to the wrapper cells enabling their selection per cell can be utilized to enable the testing of this interfacing circuitry between the different embedded cores.

Another principle that can be investigated in future research is determining how to test all critical paths through the core simultaneously by joining them during the test phase within an oscillation test environment. The oscillating frequencies of the complete core will then be evaluated to check for delay-faults in the core. This will save time and give a rough idea of any gross delay-faults in the core. The methodology to test all cores within a SoC in parallel can also be investigated.

The IEEE 1500 standard was developed to standardize the testing of embedded cores. This implementation and adherence to this standard must be a high priority within the testing community. This standard can then be used to develop Delay-fault test techniques that can be adapted to embedded cores in the fast growing SoC technology.

6.4 Original Contributions of this Thesis

The accomplishments in this thesis can be summarized as:

- Implementing and evaluating a complete DfDT structure to detect small delay faults using the Controllable Delay Scan Flip-Flop's. This structure includes the following two key elements:

    - Programmable Delay-Line
    - Programmable Duty-Cycle Controller

- The susceptibility of the DfDT structures to process- and application-induced variations were established.

- Methods to reduce the susceptibility of the DfDT structures were developed and applied to the structures and subsequently evaluated:

    - A clock-skew compensation structure was developed to accommodate up to 32 % clock-skew within the DfDT structure.
    - A voltage reference circuit was developed to reduce the effect of variations in the supply voltage to the DfDT structure.

- A new test technique to test for small delay faults by using a pulsed supply voltage within an oscillation test methodology environment was developed.

- The application of the pulsed supply voltage technique to a discrete circuit for analysis of the technique.

- The development of a test strategy to test for delay faults in embedded cores in a SoC.

- Design and evaluation of enhanced wrapper and wrapper cells to accommodate the application of the developed oscillation test to embedded cores.

Summary

The importance of delay faults is enhanced by the ever increasing clock rates and decreasing geometry sizes of nowadays' circuits. This thesis focuses on the development of Design-for-Delay-Testability (DfDT) techniques for high-speed circuits and embedded cores. The rising costs of IC testing and in particular the costs of Automatic Test Equipment are major concerns for the semiconductor industry. To reverse the trend of rising testing costs, DfDT is getting more and more important.

In order to fully exploit the capability of nowadays' advanced process technology and at the same time meet the shrinking product development schedules, the electronic industry is moving toward a design flow that integrates pre-designed and pre-verified cores into system-on-a-chip platforms, based on the ”reuse” philosophy. Cores must be made with reuse considered from the beginning, and a chip-level test architecture assembly process must be developed with consideration of many cores, and their test vectors arriving from many sources, and possibly with many different types of individual testability supported. One of the most effective reuse methods that can be applied to a potential core design is to support the design philosophy that is being referred to as “Test Wrappers”.

The quest therefore is for providing a DfDT structure to decrease testing costs by testing the high-performance digital circuit at a significantly lower speed than its operating speed while still guaranteeing its high-speed behaviour. A Controlled Delay Scan Flip-Flop (CDSFF) was developed to detect small delay faults in digital high-speed circuits. The manipulation of delay between and duty cycles of the original clock and an additional test clock, within the CDSFF, are crucial in this method. The complete DfDT structure was developed as part of a BIST structure. This method, therefore, avoids the requirement of an expensive high-speed tester. Depending on the application, a full BIST or a partial BIST approach employing a low-speed tester can be used.

The susceptibility of the complete DfDT architecture to process and application-induced variations was shown by means of Monte Carlo analyses. The variation in the supply voltage was identified as a possible problem area for our structure and a practical solution was developed to soften the factor influence of this factor on our structure. Inaccuracy in the position of the clock timing edges can be due to process variations in the clock-generation circuit. The timing inaccuracy causes undesirable clock-skews in the DfDT structure. A clock-skew correction architecture was added to the DfDT structure to be able to still provide reliable results within a clock-skew environment.

For many of the delay-fault detection approaches, the test setup is rather complicated and requires the generation of test-pattern pairs. These approaches are therefore time consuming and occupy a lot of silicon area. An interesting alternative is the Oscillation-based Test Methodology. It is based on the reconfiguration of the CUT into an oscillating device and on the measurement of indirect parameters of the oscillating frequency and/or amplitude. The technique combines the principles of both the oscillation test methods and low-voltage testing; it involves the application of a pulsed supply-voltage to the CUT. By observing the oscillation frequency one can tell whether the circuit contains faults. The effect of a change in duty cycle was used to enhance the test technique to be able to detect small delay-faults.

Existing approaches for core-based SoC testing do not provide any explicit mechanism for applying two-pattern tests, which is necessary to achieve a reliable coverage of delay and stuck-open faults. Our enhanced wrapper architecture is primarily aimed at the detection of small delay-faults in embedded cores. The complete enhanced wrapper that includes the input and output cells were to ensure correct operation within the IEEE 1500 standard guidelines. The advantages of the individual cell selection, that is part of the enhanced wrapper cells, can be utilized when delay-fault test techniques are developed for the external logic between cores. To ensure a high quality of defect screening, the test strategies use DfT hardware for controlling and observing the cores’ terminals.

Abbreviations

Abbreviation	Description
ASIC	Application Specific Integrated Circuit
ATE	Automatic Test Equipment
ATPG	Automatic Test Pattern Generation
BIST	Built-In Self-Test
CDSFF	Controllable Delay Scan Flip-Flops
CMOS	Complimentary Metal-Oxide Semiconductor
CPU	Central Processing Unit
CTL	Core Test Language
CUT	Circuit Under Test
DCC	Duty-Cycle Control
DfDT	Design-for–Delay-Testability
DfT	Design-for-Testability
DLL	Delay-Locked Loop
DSM	Deep-Submicron
EDA	Electronic Design Automation
EEPROM	Electrically Erasable Read Only Memory
EMC	Electro Magnetic Compatibility
IC	Integrated Circuit
IEEE	Institute of Electronic and Electrical Engineers
IP	Intellectual Property
ITRS	International Technology Roadmap for Semiconductors
JTAG	Joint Test Action Group
LFSR	Linear-Feedback Shift-Register
LVT	Low-Voltage Testing
MISR	Multiple Input Signature Registers
MOS	Metal-Oxide Semiconductor
OTA	Overall Tester Accuracy

OTM	Oscillation-Based Test Methodology
PCM	Process Control Modules
PDL	Programmable Delay Line
PLL	Phase-Locked Loop
ROM	Read Only Memory
RTL	Register-Transfer Level
SECT	Standard for Embedded Core Test
SIA	Semiconductor Industry Association
SiP	System-in-Package
SoC	System-on-Chip
SRAM	Static Random Access Memory
TAM	Test Access Mechanism
TAP	Test Access Port
TPG	Test Pattern Generation
TRP	Test-Resource Partitioning
TTM	Time-to-Market
ULSI	Ultra Large Scale Integration
VHDL	Very High Speed Integrated Circuit Hardware Description Language
VLSI	Very Large Scale Integration

List of Scientific Publications

- H.J. Vermaak and H.G. Kerkhoff, “Reducing the Susceptibility of Design-for-Delay-Testability Structures to Process- and Application-Induced Variations”, Proceedings of IEEE European Test Workshop (ETW), Saltsjöbaden, Stockholm, Sweden, May 2001, ISBN 0-7695-1017-5, pp. 35-41.

- H.J. Vermaak and H.G. Kerkhoff, “The Ability of Design-for-Delay-Testability Structures to function in a environment with Process- and Application-Induced Variations”, Proceedings of Conference on Circuits, Systems and Signal Processing (ProRisc), Veldhoven, The Netherlands, November 2001, ISBN 90-73461-24-3, pp. 694-699.

- H.J. Vermaak and H.G. Kerkhoff, “Testing for Delay Faults using a Variable Supply Voltage in Combination with the Oscillation Test Method ”, Proceedings of IEEE European Test Workshop (ETW), Corfu Island, Greece, May 2002, pp. 115-116.

- H.J. Vermaak, H.G. Kerkhoff and G.D. Jordaan, “Using a Pulsed Supply Voltage for Delay Faults Testing of Digital Circuits in a Digital Oscillation Environment”, Proceedings of IEEE Conferences on Electrical, Electronic and IT Research (AFRICON), George, South Africa, October 2002, ISBN: 0-7803-7570-X, pp. 47-52.

- H.J. Vermaak and H.G. Kerkhoff, “Using a Square Wave Supply Voltage to Detect Delay Faults in High Speed Digital circuits”, Proceedings of Conference on Circuits, Systems and Signal Processing (ProRisc), Veldhoven, The Netherlands, November 2002, ISBN: 90-73461-33-2, pp. 520-525.

- H.J. Vermaak and H.G. Kerkhoff, “Enhanced P1500 Compliant Wrapper suitable for Delay-fault Testing of Embedded Cores ”, Proceedings of IEEE European Test Workshop (ETW), Maastricht, The Netherlands, May 2003, ISSN 1530-1877, pp. 121-126.

- H.J. Vermaak and H.G. Kerkhoff, “Delay-fault Testing of Embedded Cores”, Proceedings of Conference on Embedded Systems & Software (PROGRESS), Nieuwegein, The Netherlands, October 2003, ISBN 90-73461-37-5, pp. 250-256.

- H.J. Vermaak and H.G. Kerkhoff, " Delay-fault Testing of Embedded Cores using an Enhanced P1500 Compliant Wrapper”, Proceedings of Conference on Circuits, Systems and Signal Processing (ProRisc), Veldhoven, The Netherlands, November 2003, ISBN 90-73461-39-1, pp. 158-164.

- H.J. Vermaak and H.G. Kerkhoff, "Using the Oscillation Test Method to test for Delay Faults in Embedded Cores", IEEE Conferences on Electrical, Electronic and IT Research (AFRICON), Gabarone, Botswana, September 2004, ISBN: 0-7803-8605-1, pp. 1105-1110.

Acknowledgements

This research would not have been possible without The Lord giving me the talent and for His grace helping me to complete this project. Without His strength and courage for me I would not have been able to do this.

This study would have been impossible without the effort and willingness of a number of people. I owe gratitude to mentors, colleagues, friends and family who have a part in making the last five years an unforgettable experience. I would like to thank:

- Above all, my assistant-promoter Hans Kerkhoff, who was willing to risk it to take me aboard the group. He has introduced me to delay-fault testing, laying the foundation for my research. His continuous confidence, motivation and trust in me ensured that I was able to finish the research. His effort, help and comments during the writing of the thesis were of vital importance to complete the thesis. Moreover was he keen and willing to help and support me throughout my stay at Twente as well as the time back home in South Africa.

- Jorrie Jordaan and Prof. Chris van Rensburg for believing in me and playing a major, sometimes painful part, in ensuring the completion of my research at Twente. They were never willing to surrender even if things were looking very dark. Jorrie’s door was always open to me and his motivation and support never faded.

- Thijs Krol for being my promoter and always willing to help when needed.

- The Central University of Technology, Free State for the opportunity they have given me to get engaged in the research at Twente. Thank you for the opportunity to be exposed to a research environment that was something to experience and to gain tremendously from.

- Arun Joseph for being my roommate and all the other TDT-friends: Octavian, Vladimir, Frank, Liquan, Milan, Rob, Egbert and Marcel.

- All the support from the secretariat, system administration and technical support.

Finally and the most important of all I would like to thank my parents who gave me the opportunity to study. Many thanks to my daughters, Lin-Mari and Mari-Louise, who went through "everything" with me. Now my time is yours again.

I want to thank my wife, Linda, for her understanding and encouragement throughout my studies. Thank you for supporting me and be with me always. Without you nothing would have been the same!

Herman Vermaak
August 2005

Biography

Herman Vermaak was born on the 20th of February 1963 in Uitenhage, South Africa. In 1981 he entered the Faculty of Electronic Engineering at the University of Stellenbosch until 1983. After that, he did his two-year military service from 1984 to 1986.

After starting his career in 1987 at Goodyear Tyre Company in Uitenhage as Electronic Technician he obtained his National Diploma in Electrical Engineering from the Port Elizabeth Technikon in 1988 and his National Higher Diploma in Electrical Engineering in 1989. He then moved to Delta Motor Corporation in 1989 as Senior Electronic Technician and in 1992 became Project Engineer for Automation. He obtained his Masters Diploma in Technology, Electrical Engineering, in 1995 with the title of his dissertation “Condition Monitoring of Heavy Metal Presses”. This project was implemented in the Delta Motor Corporation plant in Port Elizabeth. In 1995 he joined the Technikon Free State as Lecturer in Electrical and Computer Systems Engineering.

In August 2000, Herman started his Ph D studies at the University of Twente in the Testable Design and Testing group in the field of delay-fault detection under the supervision of Prof. ir. T. Krol and Dr. ir. H.G. Kerkhoff. This work led to a number of publications at conferences and workshops and finally to this thesis.

Herman is currently Head of Computer Systems Engineering at the Central University of Technology, Free State (Former Technikon Free State).