

EMBEDDED TEST ENGINE FOR EFFICIENT AT-SPEED SCAN TESTING AND PERFORMANCE BINNING OF MICROPROCESSORS

Eddie Lawlor

Department of Electronic Engineering,
National University of Ireland, Maynooth,
Maynooth, Co. Kildare, IRELAND
Phone: +353-1-7086197, email: elawlor@eeng.may.ie

Ronan Farrell

Department of Electronic Engineering,
National University of Ireland, Maynooth,
Maynooth, Co. Kildare, IRELAND
Phone: +353-1-7086197, email: rfarrell@eeng.may.ie

***Abstract** – In this paper a modified architecture for at-speed scan testing is presented. This new architecture addresses the trend in the semiconductor industry for increased at-speed structural testing. The proposed architecture offers reduced time for standard at-speed testing, and, in particular, substantial savings for the repeated at-speed testing required for microprocessor speed and performance binning. The architecture has been demonstrated on UMC 0.18 μ m and has achieved with little die overhead.*

***Keywords** – at-speed scan testing, performance speed binning, design-for-test, embedded test.*

I. INTRODUCTION

Tremendous changes have occurred within the semiconductor industry in recent years. This has been driven by industry requirements and consumer expectations of smaller, faster, more reliable, and less expensive integrated circuits. However as geometries decrease, many of the assumptions commonly used in the design of microelectronics become less valid. In particular, as geometries decrease, there is increased variability between nearby individual transistors due to random variations of doping concentrations in ever smaller active areas. These random process variations result in performance variations which are particularly noticeable at the high speeds of modern microprocessors. Specifically, these random variations can give rise to circuits with very slow-to-rise or very slow-to-fall switching transitions [1]. At low frequencies, these show transition effects are not detectable, and can only be detecting by testing the device at high speeds. These defects can cause an increase in the signal delay through digital logic paths in the device, and thus are called delay defects. Another increasingly important source of delays is due to the interconnect or wiring between

transistors [2]. These delays are due to the transmission line nature of on-chip wires carrying high frequency signals. These delays become larger as the length to width ratio of the wire increases. Random boundary variations will vary the effective width of the wire, introducing delay variations and thus delay faults. A microprocessor will fail to operate at its rated speed if it contains a delay defect that creates a path delay higher than the maximum allowed path delay. The purpose of a delay test is to verify that a microprocessor operates correctly at the specified clock speed.

To date, delay testing has required carefully designed functional test patterns that exercise the most delay-sensitive paths in the device. However modern high-speed digital design methodologies leads to a large number of short paths which may be delay sensitive. This, in combination with increased random variation due to geometry reductions, means that it is difficult to ensure sufficient delay-fault coverage using functional test approaches.

In recent times, there have been developments in the area of at-speed scan testing. This is a structural test approach and allows for increased fault coverage. Commercial tools from Mentor Graphics and Synopsys [3, 4], among others, now exist to assist digital system designers. However while at-speed scan promises increased coverage, significantly increased production test time, and thus costs, have held back deployment.

In this paper, an architecture for a modified IEEE 1149.1 SCAN Controller is presented which will allow for reduced test time costs for at-speed scan and the related task of performance identification. The modified architecture remains fully compatible with the IEEE 1149.1 specification [5]. It will be shown that the on-chip area overhead is small but improvements to the specifications of the on-chip clock generation systems may be required.

II. IEEE 1149.1 SCAN BASED TESTING

IEEE 1149.1 Scan Testing is probably the best-known structural test technique, as it can provide high level of fault coverage and is easily automated. Commercial tools that can design scan test patterns have vastly eased the digital logic test challenge.

The scan process groups flip-flops into chains. Using these it is possible to feed in a known data state (these are called test patterns or vectors), pass this through a set of combinational logic and read out from another chain of flip-flops the result. This result is compared with the expected result, and if it differs, a fault exists. The process of loading the test pattern and reading out the resulting data is called scanning in and scanning out, hence the name.

To achieve this capability, each flip-flop needs to be replaced with a scan-enabled flip-flop, which is typically a standard flip-flop with an additional multiplexor. This will allow the functional path to be altered to connect to another flip-flop when in test mode, thus forming the chain. Many chains can be formed allowing multiple parallel tests. The number of chains and the length of the chains are important parameters as the number of chains corresponds to the number of access points on the device (generally restricted by the number of pins on the device), and the length of the chains is limited by the time it takes to load and unload the data, and the memory requirements of the external tester. Ideally a combination of a large number of parallel chains with short chain lengths is optimal.

In addition to the scan-enabled flip-flops, there needs to be some control logic for interfacing with the external tester. This is often called a test-access port (TAP) controller. Each chain will require a TAP controller. In the IEEE 1149.1 specification [5], the TAP controller is a finite state machine with a number of standard operations, but with the possibility of additional user-defined operations. In

our proposed solution, this additional functionality will be utilized.

The process of arranging the scan chains, once the maximum length and number has been chosen, has been fully automated. A wide range of commercial tools is available [3, 4]. The generation of the patterns to test for faults now needs only minimal human intervention. The tools that generate the scan chains and patterns are called automated test pattern generation (ATPG) tools. The goal of any ATPG program is to achieve the highest possible fault coverage given the restrictions of the external tester and available time.

IEEE 1149.1 Scan testing was designed to detect static faults, faults that prevent a logic gate from changing state. These faults are frequency independent and thus do not require the tests to be run at the full operational frequency. However with ever-decreasing geometries, delay faults are becoming more prominent. In simple terms, these are faults result in logic gates not being able to operate at the full operational speed but correctly function at lower frequencies. Thus they avoid detection using traditional scan test techniques.

One of the main performance limitations of scan-test is the time required to upload and download the data from the device. Given that external testers typically interface with the devices under test at 100 MHz, this can be significant when using long chains. Typical microprocessor scan chain length can vary anywhere from 500 to 2000 bits [6]. The load time associated with loading a 2000-bit scan chain is 20 μ s using a 100 MHz external tester..

This load time can be reduced by using test-pattern compression techniques [7] but this requires additional on-chip logic to encode and decode the data for transmission. Test pattern compression is transparent to the test process.

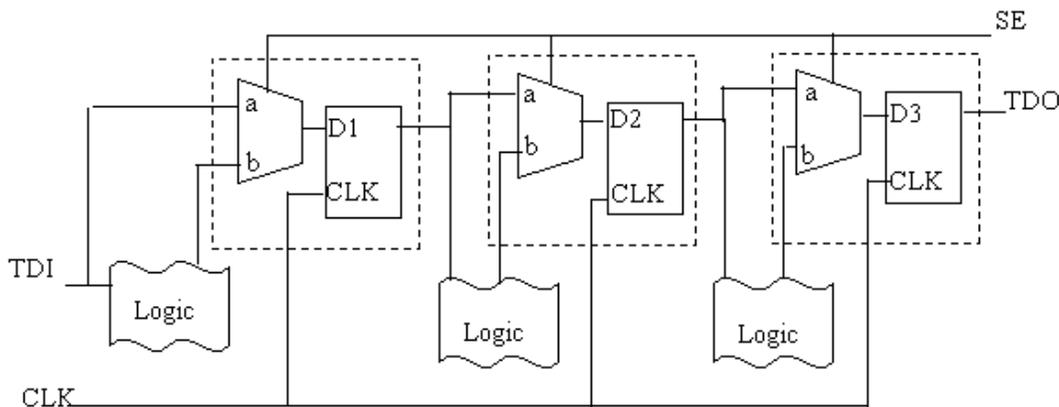


Figure 1: Organisation of Scan Chain within a Device.

III. AT-SPEED TESTING

At speed testing is used to detect delay faults, faults that only appear at full functional speed. Typically at-speed testing for delay faults use carefully designed functional test patterns that exercise the most delay sensitive paths in the device. However with the increasing number of delay sensitive paths it is difficult to predict a representative small subset of these paths that can provide sufficient coverage. A trend in the industry is to apply structural scan testing approaches at high frequencies. The principle is that once the input scan chain has been prepared, the data is clocked through the combinational logic, and stored in the scan-out register chain. If the propagation delay between the two chains of registers is too large, the correct values will not propagate through in time to be captured, thus detecting large delays.

Delay testing in microprocessors is often combined with “speed-binning”, the process of grading microprocessors according to their maximum operating speed. Typically a microprocessor may be sold at four or five different operating frequencies. Due to manufacturing variations, a device may not be capable of operating at the maximum frequency but works perfectly at a lower frequency. One the maximum correct operating frequency has been identified; the device can be fused to operate at just this speed, and thus sold at this speed rating. Paths with delay faults generally incur excessive delays, thus preventing high speed

performance, but some paths may have too little delay. This is rarer but requires testing. Thus for speed-binning, the device needs to be tested at all frequencies at which it may be used. The same tests are thus repeated at different frequencies.

Speed-binning and at-speed scan testing are time-expensive processes. Typically the device is set to one frequency, once the chains are loaded, the test is performed and the data read off. This is repeated with different patterns to achieve coverage. When complete, a new frequency is set, and the process is repeated. Due to the low interface speed with the external tester (typically 100 MHz) this repeated loading and unloading takes a large amount of time. For example if a microprocessor has scan chain 2000 bits long and has six possible operating speeds. This means that each test pattern needs to be loaded six times which has an equivalent load time of 120 μ s. If there were 4000 test patterns, the average test time associated with testing the part at speed is 0.48 seconds.

IV. PROPOSED SOLUTION

Due to recent advances in phase lock loops (PLL) for on-chip clock generation [8] it is now possible to explore a different approach in reducing at-speed structural test time. We propose to load the test pattern once and use the PLL to switch between the different speeds of operation. The tests can then be repeated using the patterns already loaded, thus

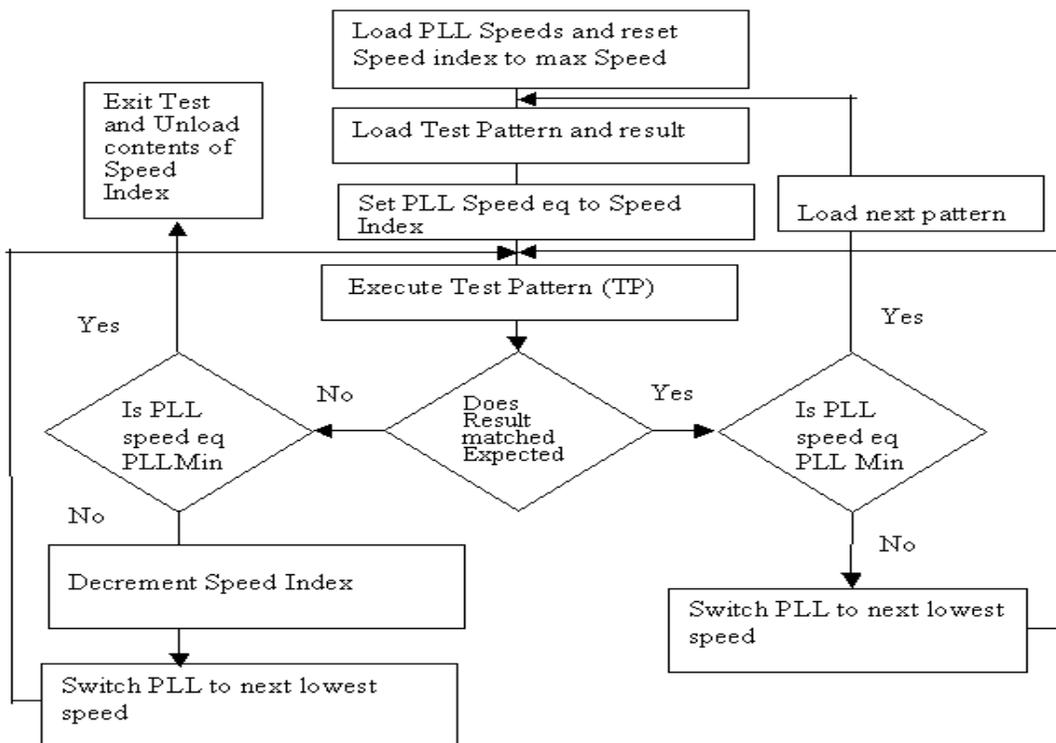


Figure 2: Proposed Algorithm for Efficient Microprocessor Performance Binning.

reducing the time impact of communicating with an external tester. The time saving achieved depends on size of the patterns, the number of speeds to be tested, and the performance of the phase-lock-loop. The full algorithm is shown in Figure 2.

Whenever switching PLL clock frequencies, a certain amount of time must pass for the clock frequency to have sufficiently settled before it can be used. This is called the lock time. This needs to be factored in when estimating the potential time saving from this new approach.

As a mathematical guide, the time required for existing tests is given by

$$T_{OLD} = \left[\frac{2 * (\# patterns) * (length)}{ATE Speed} + PLL_Lock \right] * (\# speeds)$$

$$T_{NEW} = \left[\frac{2 * (length)}{ATE Speed} + (\# speeds) * (PLL_Lock) \right] * (\# patterns)$$

where

#patterns:	Number of patterns
#speeds	Number of frequencies to test
Length	Length of patterns
ATE Speed	Speed of Tester Interface
PLL Lock	PLL Lock Time

As can be seen from these equations, any potential saving in test time will occur if the phase-lock-loop lock time is shorter than the time required to load a test pattern onto the device. Thus the maximum allowable PLL lock time is dependent on the pattern length and the speed of the interface to the tester.

In communication applications, lock times are in the order of microseconds or parts thereof. However typically in microprocessors where the frequency stays the same, lock times have not been important and can be quite large. As can be seen in the Figure 3, performance improvements are possible as

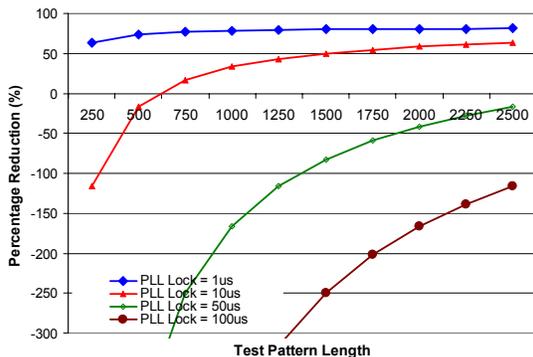


Figure 3: Time Saving Through Proposed Method

the test pattern length increases, but only if the PLL lock time is small, estimated to be 10μs for chains of 1000 bits.

To achieve this functionality, it will be necessary to provide on-chip some way of storing the expected result from test. This can then be compared on the device, at full speed, to the test results. A result register can be updated to store the value of the test result at each of the speeds. Upon completion of the test at the different frequencies, the device returns to the external tester the result register (no more than 8 bits long) that will indicate the success of the tests at the different frequencies. This will significantly simplify the capabilities required on the external tester, allowing for both recurrent savings due to reduced test time and capital savings on simpler testers.

V. APPLICATIONS AND DESIGN ISSUES

Figure 3 shows the architecture of the proposed solution. The majority of the function blocks shown are defined by the IEEE 1149.1 standard. The additional blocks are enhancements that avail of the additional user-defined TAP codes that are allowed in the standard. The additional components are the

- Input and output registers (IPR/OPR)
- A compare function (CMP)
- Speed register and control (SRC)

and a modified TAP controller. In the following section, these components will be discussed briefly.

The Tap Controller

The TAP controller is defined in the IEEE standard as a 16-state finite state machine that proceeds from state to state based on the test-clock (TCK) and test-mode-select (TMS) signals. It provides the signals that control the test data registers, and the instruction registers. This includes signals to the instruction and data registers, such as clock, shift and update. Additional user-defined states are available in the state machine if necessary. These will be used in this proposal to load the data into the input and output registers, and trigger the speed-control logic to perform the scan-tests at the different speeds. The speed control logic is responsible for implementation of the algorithm shown in Figure 2.

Input and Output Registers

The output register is used to store the expected outcomes for the at-speed scan-test. This will require a new operation from the TAP controller to enable the data to be read into this register. The test results data can be streamed serially to the compare function where it can be compared XOR bit-wise

with the output register contents. In this way it is not necessary to have a complex compare function or store the results from the scan-test.

The input register is used to hold the test pattern between tests. This is then scanned into the scan-chain, avoiding the need to communicate with the external tester.

Speed Register and Control Logic

The speed register is used to store the results from the various speed tests. Normally this corresponds to a bus ratio that can be used to drive the actually frequency of the phase-lock-loop. Typical microprocessor phase-lock-loops produce clock signals that are integer multiples of a reference crystal frequency. The speed register will map from a bus speed (say 0 to 6) to an integer multiple that will be used to control the phase-lock-loop.

The speed control logic needs to be able control the entire sequence required to load a test pattern onto the scan-chains, execute it, retrieve the results, compare with the expected outcomes and record the success or failure of the test. This then needs to be repeated at all potential speeds of operation.

In summary, the speed control logic is the state machine that sequences all the stages and repetition required for the speed binning process. The underlying algorithm is shown in Figure 2.

Phase Lock Loop

Most microprocessor now use a phase-lock-loop for internal clock generation [8]. Phase-lock-loops can generate a low noise (jitter) clock signal at a user-defined multiple of some reference clock. The reference clock is generally a high precision, low noise quartz resonator. Using such a reference, a highly precise, low noise, output clock frequency can be obtained. On-chip registers control the ratio between the reference clock and the output frequency. These can be programmed for difference frequencies during configuration. Low jitter noise and high frequency accuracy are the primary requirements for microprocessor clock generators. The time it takes to acquire that frequency is of low importance as the frequency is never changed during normal use, and during the standard technique for speed-binning it needs only change a limited number of times. In our proposal, the PLL will need to change frequency thousands of times, the number of individual patterns times the number of operating frequencies. To achieve time saving, the lock time for the PLL must be less than the time required to load a test pattern onto the device. For a 2000 bit pattern, this corresponds to 20µs.

There has been several International Test Conference papers in recent years that present microprocessors deploying clock-generators with lock times below 10 µs [8, 9]. With increasing operating frequencies, lock times should tend to decrease. However if this performance is not normally available on a device, the PLL will need to

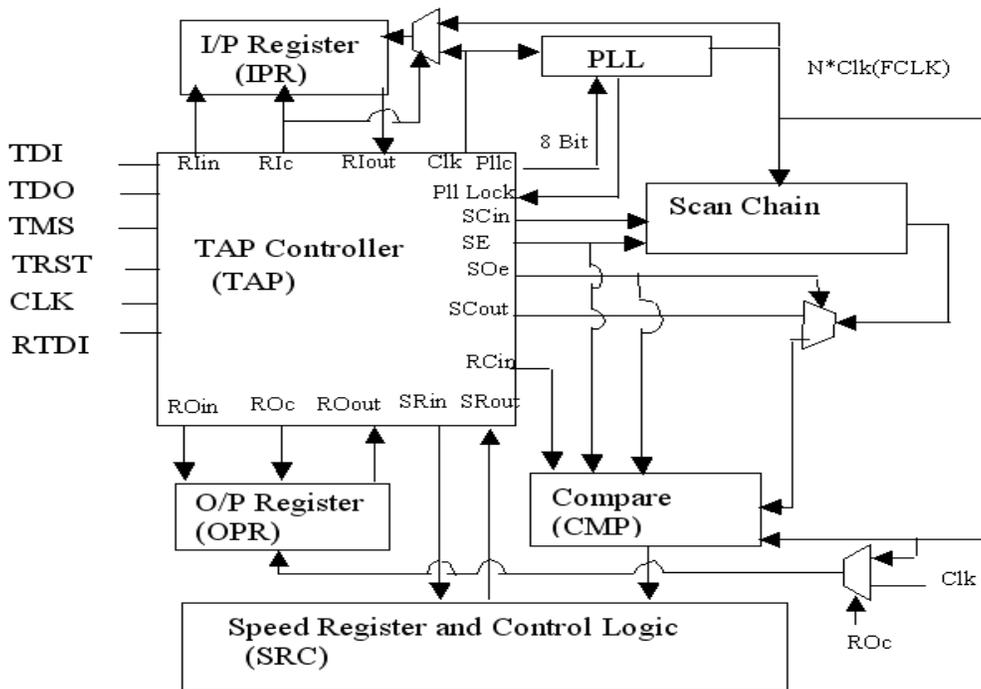


Figure 4: Functional Components for the Embedded At-Speed Test Engine.

be improved. Improving settling time performance is normally achieved at the expense of reduced noise performance or increased on-chip area. This will require balancing test costs with design constraints or manufacturing costs.

VI. IMPLEMENTATION AND RESULTS

This modified IEEE 1149.1 test architecture was implemented using UMC 0.18 μ m digital logic standard cells with the Synopsys Design Flow. Using Verilog as the RTL language, the design was successfully synthesised and is functional at the design speed of 400 MHz.

The synthesised design prior to clock tree insertion came to an estimated size equivalent to 5576 INVD1 inverters (10,000 transistors). This size does not include the input and output registers which will require one flip-flop per bit (unless existing on-board memory is used). Given that modern Pentium IV processors contain 42 million transistors, this is a small fraction of the total size of the device for up to a 60% saving on the speed-binning test costs for a microprocessor.

The next step of the development process will be to undertake an economic analysis to determine the suitable conditions for deployment in a commercial product.

VII. CONCLUSIONS

In the coming years, the need for at-speed structural testing will increase. In addition, increased performance variability and economic demands will demand more rigorous performance binning. In this paper we have presented a modification to the standard IEEE 1149.1 Scan test architecture that reduces the time for at-speed testing and performance binning of microprocessors. The functionality of our proposal has been validated through implementation and subsequent simulation on a commercial process. It was demonstrated that the additional on-chip functionality required can be achieved with only a small increase in die area.

VIII. ACKNOWLEDGEMENTS

This work has been supported through the assistance of Intel Ireland.

REFERENCES

- [1] M. L. Bushnell and Vishwani D. Agrawal, “*Essentials of Electronic Testing for Digital, Memory and Mixed-signal VLSI Circuits* (Frontiers in Electronic Testing)”, Kluwer Academic Publishers. 2000.
- [2] B. Chatterjee, M. Sachdev, A. Keshavarzi, “*A DFT technique for low frequency delay fault testing in high performance digital circuits*”, IEEE International Test Conference, pp. 1130-1139, October 2002.
- [3] Mentor Graphics, “<http://www.mentor.com>”
- [4] Synopsys, “<http://www.synopsys.com>”
- [5] *IEEE standard Test access Port and Boundary-Scan Architecture 1149.1 specification*, “<http://www.ieee.org>”
- [6] B. Bailey, A. Metayer, B. Svrcek, N. Tendolkar, E. Wolf, E. Fiene, M. Alexander, R. Woltenberg, R. Raina, “*Test methodology for Motorola high performance e500 core based on Power PC instruction set*”, IEEE International Test Conference, pp. 574-583, October 2002.
- [7] M. Mayberry, J. Johnson, N. Shahriari, M. Tripp, “*Realizing the benefits if the structural test for Intel microprocessor*”, IEEE International Test Conference, pp. 456-463, October 2002.
- [8] Jian Zhou, Huiting Chen, “*A 1 GHz 1.8v Monolithic CMOS PLL with Improved Locking*”, IEEE 2001 Midwest Symposium on Circuits and Systems (MWSCAS), 2001. August 2001.
- [9] Y. Xiaomin, T. Wu, J. McMacken, “*A 5 Ghz Fast-switching CMOS Frequency Synthesizer*”, IEEE Radio Frequency Integrated Circuits (RFIC) Symposium, pp. 479-482, June 2002.