

MICROPROCESSOR TESTING

Arun B. Kannan

1. ABSTRACT:

Today's Microprocessors consists of millions of transistors operating at extraordinarily high speeds. Verification and Test of these high performance devices continuously challenges engineers in the microprocessor design cycle. This paper gives a broad perspective of the field of microprocessor testing. Understanding of the design flow of a microprocessor is important so that difference between the various verification and validation efforts from testing is realized. The System and Physical Faults in the system are modeled either by Structural or Functional modeling. And based on the type of modeling; test strategies are developed. The corresponding test methodologies are; Structural and Functional testing. Functional Testing was traditionally adopted for LSI because the classical structural tests became computationally expensive. But Design For Testability approaches have eased structural testing. Both functional and structural testing has pros and cons. A balancing act is desired keeping in mind the various microprocessor 'design and market' goals especially cost-performance. The techniques implemented in Power PC 603, a typical modern microprocessor is analyzed in order to illustrate the key ideas.

2. INTRODUCTION:

Current high-performance microprocessors are a result of advancements in Integrated Circuit (IC) technology and Computer Architectural Innovations. IC technology will be able to integrate a billion transistors on a single chip by the end of this decade. The Design engineers enable to produce the performance that will be impossible to achieve only with technology scaling, by implementing various techniques like speculative execution, out-of-order execution etcetera.¹

There could be a lot of things that could go wrong during the design of a microprocessor. Therefore it is important to know the design flow of a modern microprocessor.² Generally, the design phases are as follows: 1) *Product definition*; defining the processor's feature set 2) *Building architecture blocks*; creating a block level structure of the performance model 3) *Foundry Specifications*; develop a 'Technology File' 4) *Feasibility studies*; analyzing if the architectural ideas explored can be implemented using the available technology 5) *Tools development*; developing various CAD tools used at various stages 6) *RTL modeling*; Hardware Description Language modeling of the block level model 7) *Functional verification/validation*; verification/validation of design by computer simulation or by other methods 8) *Schematic design*; automated cum manual translation of RTL model into a gate level model 9) *Logic Verification*; verification of faithful translation from RTL to Gate level assuming an ideal transistor model (transistor as 0/1 switch) 10) *Layout*; automated cum manual translation from gate model to transistor level model 11) *Circuit Verification*; checks like racing, cross-talk, electromigration using a more precise transistor model (leaky, noisy and modeling other delays) 12) *Fabrication*; releasing design to the fabrication facility 13) *Testing*; procedure to detect faulty chips from the good ones and 14) *Debug*; fixes are made in required phases into the second or third pass of silicon.

We can observe from the design methodology that; the process has a distributed form of testing. Depending on the design phase various names are given. Requirements and specifications are audited, design and tests are verified / validated, and fabricated parts are tested.³

In section 3, the fault models and testing strategies are discussed and in section 4, a case study of Power PC 603 is presented.

3. FAULT MODELS AND TESTING STRATEGIES:

Modeling plays a central role in the design, fabrication, and testing of a digital system. The way we model a system and faults in the system, has important consequences for the way we simulate it to verify its correctness and the way we generate tests for it.⁴

There are two basic types of *system/fault models*:

3.1 Structural model:

A typical structural model of a system is a connectivity language that specifies the I/O lines of the system, its components, and the I/O signals of each component. For example:

```
CIRCUIT AND
INPUT A
INPUT B
OUTPUT C
C = AND (A, B)
```

Faults defined in conjunction with a structural model are referred to as structural faults; their effect is to modify the interconnections among components. The main types of structural faults are shorts and opens and they are mapped into *stuck-at* and *bridging faults*.

3.2 Functional model at Register Transfer Level (RTL):

RTL's provide models for systems at the register and instruction set levels. For example:

register IR [0? 7]; defines an 8-bit register IR

memory ABC [0? 255; 0? 15]; defines a 256-word memory ABC with 16-bit word.

C=A+B, delay = 100; if A, B and C are registers then it denotes the addition of values of A and B, followed by the transfer of result into C after 100 time units since the initiation of transfer.

Functional faults attempt to represent the effect of physical faults on the operation of a functionally modeled system. *Addressing fault* affecting register-decoding function is an example of a functional fault.

3.3 Structural and Functional testing:

If the test generation methods are based on the structural model of a system under test and their objective is to produce tests for structural faults then the method is structural testing. PODEM, CONTEST etcetera are (ATPG) algorithms that target the modeled stuck-at faults.

Functional Testing methods are based on the functional model of the system and produce tests for the functional faults modeled. Thatte and Abraham⁵ derived functional test sequences for an 8-bit microprocessor and fault coverage of 96% was obtained. Functional testing has not yet achieved the level of maturity and the success of structural testing methods. It is still not automated and generally hand-written.

3.4 Issues in Structural testing:⁶

Structural test generation methods will not be able to cope up with complex VLSI devices because the latter contain a large number of gates, flip-flops, and interconnections and therefore require an enormous amount of computation to generate complete test sets.

In order to ease test generation efforts Design for Testability (DFT) techniques are employed. But DFT hardware increases Die Area. Increase of die area may affect yield and may require construction of a new multibillion dollar fab. Power scales linearly with die area and wattage is a limiting factor in desktop CPU design. All available transistors need to be used to implement all the latest micro architectural advances to enhance performance and should not be sacrificed for DFT. At the same time, zero DFT will give unacceptable quality. But structural testing will be less expensive (less capable at the same time) and test generation could be automated.

3.5 Issues in Functional testing:⁷

Historically INTEL has tested its microprocessors with manually generated functional tests that are applied at speed using functional testers. The advantages of functional testing are: 1) high correlation to customer experience 2) high efficiency in test time and 3) accurate measurement of test speed.

The disadvantages are: 1) it is expensive (cost wise, P2 ATE was 3.6X times the INTEL 386 ATE) and 2) Writing functional tests requires both architectural design and testing expertise and 3) It cannot be readily automated.

3.6 Balancing act:

A 'Distributed Test' strategy developed by INTEL is outlined below (figure 1) that employs both structural and functional testing. A test designed to measure the full chip speed as driven by I/O's to detect a timing glitch or a speed path in the bus logic. This type of test requires the full bus capabilities and is typically provided by a functional tester. Structural tests are employed for scan based logic tests and Direct Access Testing (DAT) based cache tests. A BIST test could be run on the burn-in platform.

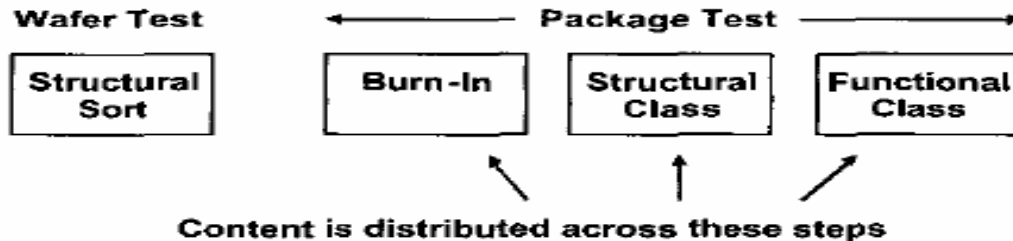


figure 1⁷

4. CASE STUDY OF POWER PC 603:

The 1.6 million transistor 603 processor was designed at the Somerset design Center by engineers from Apple, IBM and Motorola. It is also a very good prototype of a modern microprocessor. The following are the *set of test objectives* agreed by both Motorola and IBM;

- 1) Maintain original die size goal
- 2) Maintain original performance objectives
- 3) Achieve greater than 95% DC stuck fault coverage
- 4) Achieve greater than 90% AC transition fault coverage
- 5) Generate functional test vectors to supplement DC untested logic areas
- 6) Provide an IEEE 1149.1 test interface
- 7) Ensure existence of full debug, diagnostic and verification capabilities

4.1 IBM techniques:

- | | |
|---|-------------------------------|
| 1) Full Scan: Level sensitive scan design (LSSD): | DC stuck fault coverage > 99% |
| | AC transition testing > 95% |
| 2) Array BIST and Scan based array test patterns: | Embedded arrays. |
| 3) Weighted random pattern Testing: | To reduce test data volume |

4.2 Motorola techniques:

- 1) Range of DFT techniques; ranging from full scan and partial scan to ad-hoc on-chip test circuitry including special instructions, PLA and array control and BIST.
- 2) Functional test vectors are used extensively to speed sort.

4.3 Somerset solution:

The best of both worlds was integrated and a new test methodology was obtained.

- 1) **Full scan LSSD:** It was employed to test *all logic circuits*. APG was used to generate both DC stuck at and AC transition fault patterns.
- 2) **ABIST:** *All large memory arrays* (Instruction and Data caches)
- 3) **Functional Testing:** The process of creating functional vectors for component testing consists of writing assembly language programs that exercise various areas of the chip. Functional patterns are used to test *all small arrays* (6 arrays of MMU, General purpose register, Floating point register and non-scan data queues) and other areas of the chip that could not be fully tested by the scan based patterns. Apart from small arrays, functional patterns were used to *enhance AC fault detection*. The functional vectors had the advantage of operating at speed, which in turn would enhance the test coverage by *detecting delay or speed related defects*.

4) IEEE boundary Scan: To support the end customer's board and system test needs, an IEEE 1149.1 serial port was included. (The latest microprocessors will have Analog Test Bus; IEEE 1149.4 standard incorporated too)

5) PLL testing: As an analog circuit special consideration is necessary, not only to test functionality, but also to ensure operation over voltage, temperature and frequency ranges. An independent functional measurement is performed to verify the frequency operation of PLL.

6) IDDQ: Manufacturing defect analysis showed that some defects in circuits could only be tested efficiently using IDDQ measurement. Due to the very long tester times involved with IDDQ measurement, only limited use of this technique is used in the manufacturing environment.

7) Diagnostics: Physical Failure Analysis tools link logical faults to physical faults. These are repeatedly used to identify and isolate actual physical defects on the chip. No tool has yet been developed to help diagnose functional pattern fails.

By employing such an approach; all the aforementioned objectives were satisfactorily achieved.⁸

5. CONCLUSION:

After manufacturing a microprocessor and the initial wafer sort; a suite of manufacturing tests are run. These tests reject any chip that does not meet operational, electrical, timing or environmental specifications. Speed grading of the chips are also done subsequently. This is done to determine the maximum clock frequency for reliable operation, so that chips can be sorted and sold accordingly. Either functional or structural tests can be derived for the various parts of the microprocessor namely; random logic, memory arrays, analog circuitry, test controller, boundary scan etcetera. The proportion of the functional tests or structural tests in the total set of tests is based on various factors that were discussed in the paper.

After running the manufacturing test suite, the microprocessor is subjected to Burn-In/Stress test. In this, parts are subjected to high environmental stress while a test program is run to stimulate the part and detect defects as they appear.

Locating the causes of common defects and reliability problems so that design and process changes can be made to improve product quality is called Failure Mode Analysis (FMA) and is the final step before good parts are shipped and the first step for the next pass of silicon.⁹

6. REFERENCES:

- 1) Microarchitectural Innovations: Boosting Microprocessor Performance beyond Semiconductor Technology Scaling; Andreas Moshovos and Gurindar S. Sohi; Proceedings of the IEEE, Vol. 89, No. 11, November 2001
- 2) Designing an ALPHA microprocessor; Matt Reilly, Compaq Comp. Corp.; July 1999, IEEE Computer
- 3) Essentials of Electronic Testing for digital, memory and mixed-signal VLSI circuits; Michael L. Bushnell, Vishwani D. Agarwal; Kluwer Academic Publishers
- 4) Digital systems testing and testable design; Miron Abromovici et al; IEEE Press
- 5) Test generation for microprocessors; Satish M. Thatte, Jacob A. Abraham; IEEE Transactions on Computers, Vol. C-29, No.6, June 1980
- 6) Pentium pro processor Design for test and debug; Adrian carbine, Derek Feltham INTEL; July-September 1998 IEEE Design and Test of Computers
- 7) Realizing the benefits of structural test for INTEL microprocessors; Mike Mayberry et al INTEL; International Test Conference 2002
- 8) Balancing structured and ad-hoc design for test: Testing of Power PC 603™ microprocessor; Craig hunter et al; International Test Conference 1994
- 9) Testability features of the MC68060 microprocessor; Alfred. L crouch et al, Motorola Inc; International Test Conference 1994.