A Window into Design: MSP430 Design-for-Test
Michael Zwerg

Agenda

• Introduction And Definitions
• MSP430 Production Test
• Silicon Defect and Fault Modeling
• Basics On Design for Testability
• MSP430 DFT Strategy for 5xx Family
• Conclusion
Introduction

• TI is serious about quality. The process of learning how to achieve ever-higher levels of reliability, even in the presence of increasing technological challenges, is ongoing.
• The intent of this course is to inform the attendee of the various problems encountered in silicon circuits today, and how TI tackles them. Various silicon fault models, appropriate design-for-test (DFT) countermeasures, and how they enable delivery of a quality, cost effective end product are discussed.

MSP430 Realization Process
**Verification vs. Testing**

- **Design Synthesis**
  - Given a specific function, develop a procedure to manufacture a device using known materials and processes.

- **Verification**
  - Predictive analysis to ensure that the synthesized design, when manufactured, will perform the given specific function.

- **Test**
  - A manufacturing step that ensures that the physical device, manufactured from the synthesized design, has no manufacturing defect.

---

**Verification vs. Testing**

<table>
<thead>
<tr>
<th>Verification</th>
<th>Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verifies correctness of the design.</td>
<td>Verifies correctness of manufactured device.</td>
</tr>
</tbody>
</table>
| Performed by simulation, HW emulation or formal methods. | Two part process:  
1. Test generation: SW process executed once during design.  
2. Test application: electrical test applied to device. |
| Performed once prior to manufacturing | Test applied to every manufactured device          |
| Responsible for quality of design.   | Responsible for quality of design                  |
Ideal Tests

- Ideal tests detect all defects produced in the manufacturing process.
- Ideal tests pass all functionally good devices.
- Very large numbers and varieties of possible defects need to be tested.
- Difficult to generate tests for some real defects. Defect-oriented testing is an open problem.

Cost of Testing

- Design for testability (DFT)
  - chip area overhead and yield reduction
  - performance overhead
- Software processes of test
  - test generation and fault simulation
  - test programming and debugging
- Manufacturing test
  - Automatic Test Equipment (ATE) capital cost
  - test center operational cost
**ATE Testing Principle**

- Automatic Test Equipment (ATE) is used for manufacturing test

![Diagram showing ATE testing principle](image)

**Manufacturing Test**

(Also called production test)
- Determines if manufactured chip meets specs
- Must cover high % of modeled faults
- Must minimize test time (to control cost)
- No fault diagnosis
- Go/no-go decision is made
- Target is: Tests every device on chip

**ATE 2008**

MSP430 Advanced Technical Conference
Agenda

- Introduction And Definitions
- MSP430 Production Test
- Silicon Defect and Fault Modeling
- Basics On Design for Testability
- MSP430 DFT Strategy for 5xx Family
- Conclusion

MSP430 Test Flow

MSP430 Full Performance Probe Flow
(includes Embedded Flash and Mixed Signal Test)

From Wafer Fab

<table>
<thead>
<tr>
<th>WaferSort1</th>
<th>(Flash) Data Retention Bake</th>
<th>WaferSort2</th>
<th>Data and silicon post processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>x - Site</td>
<td></td>
<td>x - Site</td>
<td>To Assembly / Test - site</td>
</tr>
<tr>
<td>Room temp</td>
<td></td>
<td>High temp</td>
<td></td>
</tr>
<tr>
<td>Probe</td>
<td></td>
<td>Probe</td>
<td></td>
</tr>
</tbody>
</table>

MSP430 Probe Test Platform
**MSP430 Test Flow**

**MSP430 Assembly and Final Test Flow**

- From Probe Site
- Assembly eWafermap
- Final Test x-Site Room temp
- Tape&Reel Bake DryPack
- To Warehouse

**ATC 2008**

**MSP430 Final Test Platform**

---

**Full Performance Probe**

**First probe insertion - WS1**
- at room temperature
- highly parallel test and therefore connects only minimum of bondpads
- consists of
  - flash memory test
  - time consuming tests
  - tests with reduced pin count (SCAN)
  - tests with high fail detection

**Second probe insertion – WS2**
- at high temperature
- maximum sites doable with all bondpads
- consists of
  - flash memory test after bake
  - pin connectivity and leakage
  - thorough functional and parametric test
Final Test

Final test - FT
- after packaging and symbolization
- at room temperature
- focuses on assembly induced failures
- maximum sites parallel test depending on handling system
- consists of
  - flash memory test
  - flash erase
  - bond connectivity test
  - device and module self tests
- Flash Factory Programming of customer code

Agenda

- Introduction And Definitions
- MSP430 Production Test
- Silicon Defect and Fault Modeling
- Basics On Design for Testability
- MSP430 DFT Strategy for 5xx Family
- Conclusion
Motivation

• Models are often easier to work with
• Models are portable
• Models can be used for simulation, thus avoiding expensive hardware/actual circuit implementation
• Nearly all engineering systems are studied using models
• All the above apply for logic as well as for fault modeling

Defect, Fault and Error

• **Defect** (imperfection in hardware):
  – A defect in an electronic system is the unintended difference between the manufactured hardware and its original design.

• **Error**:  
  – A wrong output signal produced by a defective system is called an error. An error is an “effect” whose cause is some “defect”.

• **Fault** (imperfection in function):  
  – A representation of a “defect” at the abstracted function level is called a fault.
Why Model Faults?

- I/O function tests inadequate for manufacturing (functionality versus component and interconnect testing)
- Real defects too numerous and often not analyzable
- A fault model identifies targets for testing
- A fault model makes analysis possible
- Effectiveness measurable by experiments

Some Real Defects in Chips

- Processing defects
  - Missing contact windows
  - Mask misalignment
  - Oxide breakdown
  -...

- Material defects
  - Bulk defects (cracks, crystal imperfections)
  - Surface impurities (ion migration)
  -...

- Time-dependent failures (Age defects)
  - Dielectric breakdown
  - Electromigration
  - Vth shift
  -...

- Packaging failures
  - Contact degradation
  - Seal leaks
  -...
Silicon Fault Modeling

- There are two fault models, which are commonly used to describe silicon defects.

1. **Stuck-at fault model**
   - catastrophic fail
   - model for mechanical defect (short, several transistors)

2. **Delay fault model** (transition, path delay)
   - partial fail
   - model for process variation, or single transistor defect

---

Stuck-At Fault Model

- Assumption:
  Silicon defects will have an impact on the logical function of a circuit.

- On the next few foils, an **AND** gate is used as an example to explain fault models.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
**Stuck-At Fault Model**

- A short to VDD on output will change the logic (table) of this gate
- This fault is called a Stuck-At-One (SA1) fault
- Since the output is never changing to zero, this fault is easy to detect

### Stuck At Fault Model
- simple VCC/VSS short
- functional defect

### Stuck-At Fault Model (Continued)

- A short to VSS on one of the inputs will also change the logic (table)
- This fault is called a Stuck-At-Zero (SA0) fault
- This fault is harder to detect, because it will only surface at one of the four logic states
Delay Fault Model

• Not all silicon defects show such a drastic fault effect as discussed in previous slides

• For a more detailed fault model, the transistor representation of the logic has to be considered

Delay Fault Model

• Delay faults are much more difficult to detect and there are two fault models to differentiate:

  • **Transition delay fault**
  – single, local defect
  – error on one certain transition

  • **Path delay fault**
  – process variation with weak (slow) corner
  – error on critical path
Transition Delay Fault Model

- Transistor level schematic of the AND gate

<table>
<thead>
<tr>
<th>Input A</th>
<th>Input B</th>
<th>Output Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Transition Delay Fault Model

- Zero on input A and B, will lead to zero on output Y

<table>
<thead>
<tr>
<th>Input A</th>
<th>Input B</th>
<th>Output Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Transition Delay Fault Model

- One on input $A$ and $B$, will lead to one on output $Y$

<table>
<thead>
<tr>
<th></th>
<th>$A$</th>
<th>$B$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Continued

Transition Delay Fault Model

- Opposite level on input $A$ and $B$, will lead to zero on output $Y$

<table>
<thead>
<tr>
<th></th>
<th>$A$</th>
<th>$B$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Continued
Transition Delay Fault Model

• One transistor is week (slow), but the logic looks still functional

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Transition Delay Fault Model

• Transistor defect leads to floating node, which makes the result transition dependent

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>?</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Transition Delay Fault Model

• transistor defect
• non static faults
Fault Model Conclusion

• Silicon faults can be modeled with different levels of accuracy
• Some fault models are easier to detect than others
• For stuck-at faults a simple functional check is sufficient
• To detect a transition fault, critical or at-speed timing has to be applied

Agenda

• Introduction And Definitions
• MSP430 Production Test
• Silicon Defect and Fault Modeling
• Basics On Design for Testability
• MSP430 DFT Strategy for 5xx Family
• Conclusion
Motivation

• As shown in the previous sections, it is difficult to generate a test that achieves good test coverage under several fault model considerations.

• Design for testability (DFT) refers to those design techniques that make test generation and test application cost-effective.

Functional Test

Test method, where the circuit under test is exercised by using functional vectors.
Functional Test

**CPU Register Read/Write**
- easy, because all registers are accessible by CPU (SW)

**State Machine Logic**
- difficult, because the registers are not accessible

---

**Partial Logic of a FSM**
Functional vs. Structural Test

• Functional test
  – technically all combinations of a logic have to be tested
  – difficult to achieve good coverage in reasonable time

• Structural test
  – automatic pattern generation for specific circuit
  – Build-In Self Test (BIST)
  – SCAN design (add test hardware to all flip-flops to make them a giant shift register in test mode)

Software BIST

• MSP430 is a full System on Chip (SoC), which gives us the benefit of doing a (build in) self test via the CPU.

• One good example is the RAM. By executing a program with the CPU, extensive read and write operations with different pattern (all one, all zero, checkerboard, inverse checkerboard) can be applied to the memory.
SCAN Design

- For future devices the peripherals are getting more features and therefore more complex, which means it is much more difficult to generate a self test with a good coverage.

- SCAN design is a DFT method, where you are adding HW and an external access to the device. This will allow a detailed control of all Flip-Flops in your design.

- This way a computer can generate a directed test pattern with a known test coverage.

Multiplexed SCAN

- Every Flip Flop in the design is replaced with SCAN equivalent
- Add multiplexer for additional data input
- Add external test access to the device
Multiplexed SCAN

During the **shift cycle** all flops in the design are concatenated to a giant shift register

SCAN_EN = 1, SHIFT CYCLE

During the **capture cycle** all flops in the design are connected as functional path

SCAN_EN = 0, CAPTURE CYCLE
Reset and Clock Control

- All resets and clocks have to be controllable by external pins
- Add multiplexer for reset and clock control

SCAN Pattern

- Stuck At Pattern
- Transition Delay Fault Pattern
Conclusion

• SCAN is an efficient method of structural test, to get a high test coverage on a given circuit

• But SCAN does add test logic, which is a substantial overhead in functional (normal) mode
  – more area, means higher cost
  – more logic, means higher power consumption
Agenda

• Introduction And Definitions
• MSP430 Production Test
• Silicon Defect and Fault Modeling
• Basics On Design for Testability
• MSP430 DFT Strategy for 5xx Family
• Conclusion

Modular SCAN Methodology

- Insert SCAN on module hierarchy
- Limit number of SCAN chains to a minimum
  - IFCLK group
  - SCAN clock group
- Add test observation points for not SCAN-able logic
Modular SCAN Methodology

- Use JTAG to select Module Under Test
- Only 6 additional pins are needed for SCAN test (support low pin count devices & massive parallel wafer sort)
- Easy pattern debug, because single module is visible to outside (no side effect by other modules)

Functional Test Modes

- WDT_A is extended to a 32bit counter
- Very power sensitive, therefore no SCAN
- 4 Giga cycles for one full test
- Test mode splits the counter to 4x 8bit counter (only 256 cycle)
5xx Production Test Strategy

- SCAN versus functional test
- Based on configuration of MSP430F543x

Agenda

- Introduction And Definitions
- MSP430 Production Test
- Silicon Defect and Fault Modeling
- Basics On Design for Testability
- MSP430 DFT Strategy for 5xx Family
- Conclusion
Conclusion

- It has been shown, that special Ultra Low Power design techniques require a careful selection of test features for production test.

- For the 5xx family SCAN will be an important measure to guarantee a good DPPM rate at production test.

- In addition special test modes on functional level will help identifying defective parts on circuits where SCAN can not be applied.