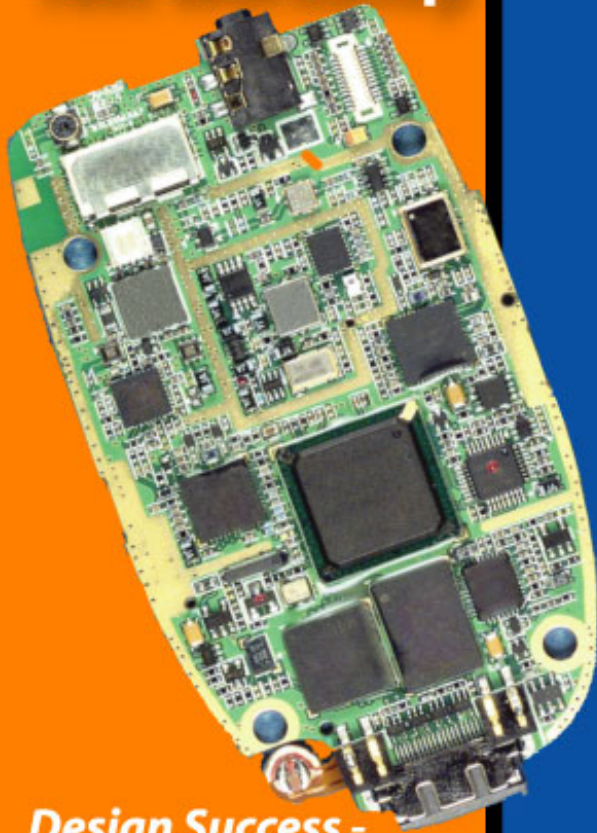


12th Annual

KGD

**Packaging &
Test Workshop**



*Design Success -
KGD Starts
at the Beginning*

Sept. 11-14, 2005 , Napa, CA

Complete Path Delay Solution for High Reliability Test Screening

Glenn Bedal

Principle Design Automation
Engineer

Medtronic, Inc.

12th Annual

KGD

**Packaging &
Test Workshop**

Agenda

- Industry Test Overview
- Fault Modeling and Detection Methods
- Path Delay Test Methodology
- Test Development and Characterization Process Flow
- Debug, Analysis, and Correlation

Sept. 12-15, 2005

Napa, California

© 2005 DPC

Industry Test Overview

- Most of the industry is Yield Focused
 - Only critical true timing paths are considered for path delay testing
- DSM reliability reduced to 2 years from 10+ years on older processes
- High reliability markets (8-11 years)
 - Automotive
 - Medical
- Common products drive industry test development are uProcessors and commodity uControllers (Cell Phones).

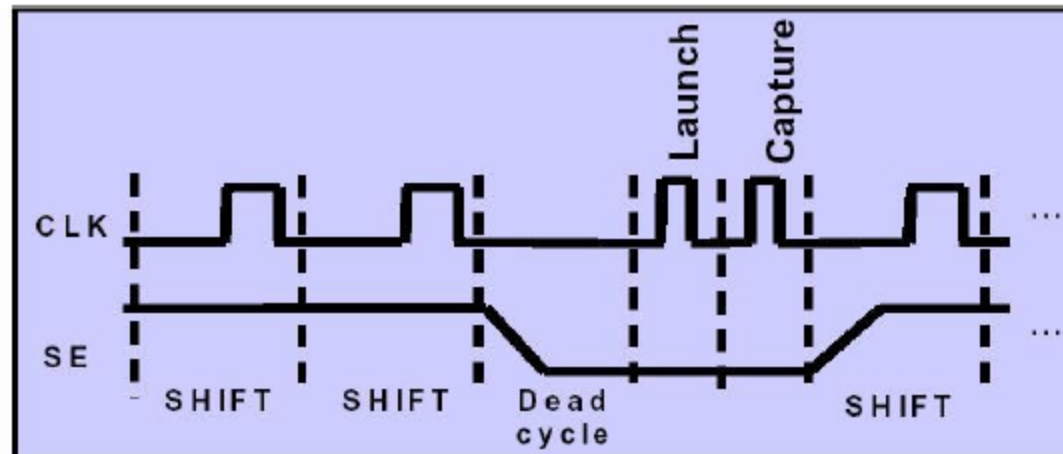
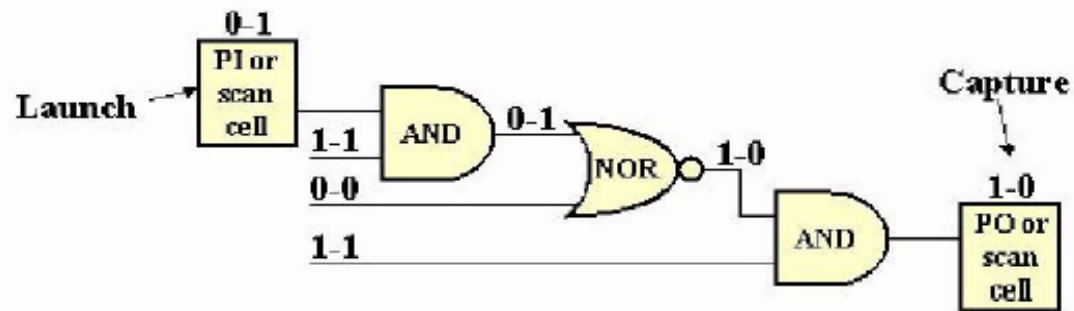
Fault Modeling and Detection Methods

- **Single Stuck-at**
 - Defect Types (shorts)
 - Requires direct controllability and observability
- **IDDQ**
 - Defect Types (shorts, some resistive, and some opens)
 - Requires low quiescent currents (<100nA-500nA)
- **Bridging**
 - Defect Types (shorts and resistive)
 - Requires layout extraction database for bridging pairs
- **Transition Delay**
 - Defect Types (resistive slow-to-rise and slow-to-fall)
 - Requires direct controllability and observability
- **Path Delay**
 - Defect Types (resistive, capacitive, vias, process variations)
 - Requires controllability, observability, and static timing information)

Path Delay Test Methodology

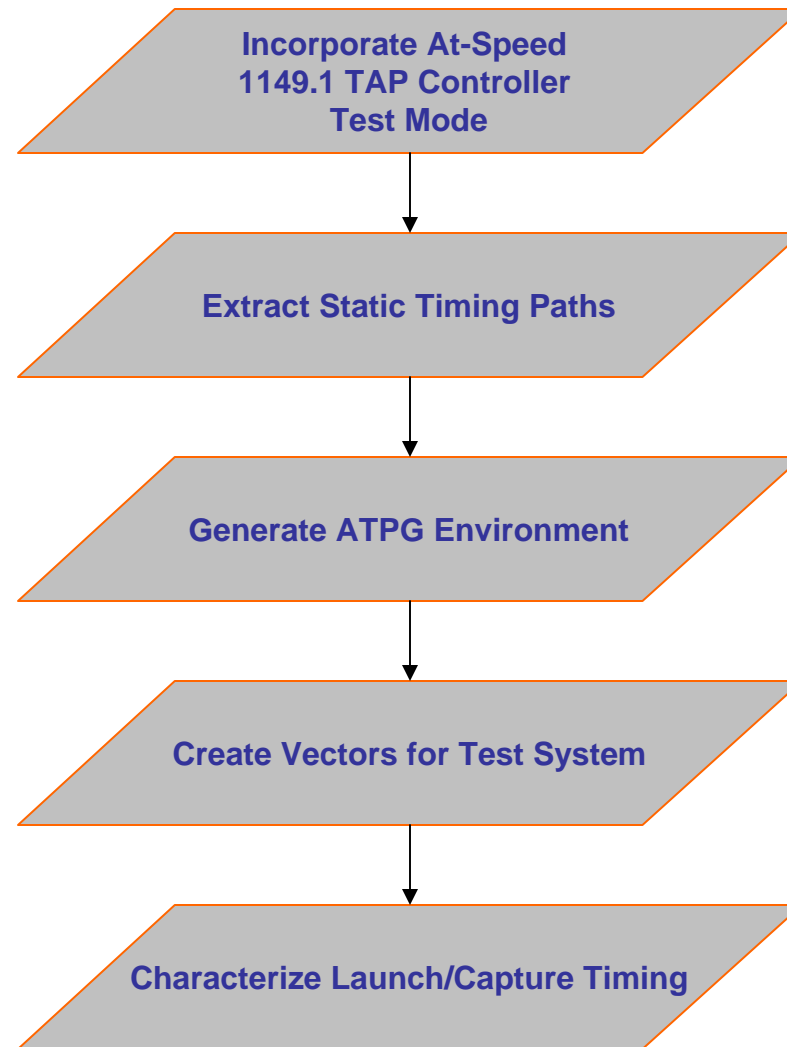
- Path delay fault model and scan ATPG detection techniques
 - Less absolute for diagnostics than transition fault with multiple vector sets
 - Expected timing can create more absolute pass/fail reliability screen metrics through static timing analysis paths

Path Delay Test Methodology



- Launch on one scan Flip-flop and Capture on another Flip-Flop through combinational logic that has been setup for logical non-blocking observability to capture Flip-Flop

Test Development and Characterization Process Flow



Test Development and Characterization Process Flow



Incorporate At-Speed
1149.1 TAP Controller
Test Mode

- IEEE 1149.1 TAP Controller
 - At-speed instruction loaded in the IR register
 - Mode is executed in the Run_Test_Idle state per standards requirement
 - Scan enable is controlled by external instead of TAP
 - Boundary pads capture internal logic

Test Development and Characterization Process Flow



Extract Static Timing Paths

- Use Synopsys DesignCompiler™ to Extract Timing Paths (Slow corner and max delay)
 - Functional combination paths are extracted from Q/NQ to D of flip-flips
 - Command Syntax:
 - `report_timing -path full -input_pins -nets -transition_time -significant_digits 3 -delay max -nworst 100 -lesser_path 100 -greater_path 1 -max_paths 100 -from find(pin -hier "*Q") -to find(pin -hier "*/D")`
 - `true_delay_prove_true_backtrack_limit = 1000000`
 - `true_delay_prove_false_backtrack_limit = 1000000`

Test Development and Characterization Process Flow



Extract Static Timing Paths

- Group paths into categories based on lesser_path and greater_path option value ranges
- Paths are absolute timing paths through combinational logic that need no attention to different clock domains
- Structure of paths are library dependent and may require path modification depending on tools

Test Development and Characterization Process Flow



- Path format:

Startpoint: core_i_my_digital_AC_top1_rising_edge_reg (rising edge-triggered flip-flop)
 Endpoint: core_i_my_digital_AC_top1_ACcount_reg_2_ (rising edge-triggered flip-flop)
 Path Group: (none)
 Path Type: max

Point	Fanout	Trans	Incr	Path
core_i_my_digital_AC_top1_rising_edge_reg/Q (sdfrc28a)	<-	1.900	19.609 *	19.609 f
core_i_my_digital_n97 (net)		3	0.000	19.609 f
core_i_my_digital_AC_top1_U40/I2 (nor2c28a)			1.900	0.002 * 19.611 f
core_i_my_digital_AC_top1_U40/O1 (nor2c28a)			4.339	3.822 * 23.433 r
core_i_my_digital_AC_top1_n45 (net)		5	0.000	23.433 r
core_i_my_digital_AC_top1_U54/S1 (muxi2c28a)			4.339	0.000 * 23.433 r
core_i_my_digital_AC_top1_U54/O1 (muxi2c28a)			0.591	3.360 * 26.793 f
core_i_my_digital_AC_top1_n51 (net)		1	0.000	26.793 f
core_i_my_digital_AC_top1_U47/I1 (aoi21c28a)			0.591	0.000 * 26.793 f
core_i_my_digital_AC_top1_U47/O1 (aoi21c28a)			1.698	1.072 * 27.865 r
core_i_my_digital_AC_top1_n35 (net)		1	0.000	27.865 r
core_i_my_digital_AC_top1_ACcount_reg_2_/D (sdfrc28a)			1.698	0.000 * 27.865 r
data arrival time				27.865

Test Development and Characterization Process Flow



Generate ATPG Environment

- Mentor Graphics Fastscan[™] environment
 - Convert paths into the following format:

```
//startReg: core_i_my_digital_AC_top1_rising_edge_reg  
//endReg:core_i_my_digital_AC_top1_ACcount_reg_2_  
path "path1" =  
pin core_i_my_digital_AC_top1_rising_edge_reg/Q ;  
pin core_i_my_digital_AC_top1_U40/I2 ;  
pin core_i_my_digital_AC_top1_U40/O1 ;  
pin core_i_my_digital_AC_top1_U54/S1 ;  
pin core_i_my_digital_AC_top1_U54/O1 ;  
pin core_i_my_digital_AC_top1_U47/I1 ;  
pin core_i_my_digital_AC_top1_U47/O1 ;  
pin core_i_my_digital_AC_top1_ACcount_reg_2_/D ;  
end;
```

Test Development and Characterization Process Flow



Generate ATPG Environment

- Mentor Graphics Fastscantm environment
 - Generate dofile and testproc config file

Dofile Commands

- add pin constraint XATSPEED_SE c0
- set fault type path_delay -Mask_nonobservation_points
- set system mode atpg
- load path D271_B02_slow_mentor_paths_1_only

Test Development and Characterization Process Flow

Generate ATPG Environment

- Mentor Graphics Fastscantm environment
 - Generate dofile and testproc config files

Testproc Commands

```
timeplate tp_shift =  
  force_pi 0;  
  measure_po 100;  
  pulse XTCK 800  
200;  
  period 1000;  
end;  
timeplate tp_launch  
=  
  force_pi 0;  
  measure_po 100;  
  pulse XTCK 800  
200;  
  period 1000;  
end;
```

```
timeplate  
tp_capture =  
  force_pi 0;  
  measure_po  
100;  
  pulse XTCK  
200 200;  
  period 1000;  
end;
```

```
procedure capture  
launch_capture =  
  cycle =  
    timeplate  
tp_launch;  
    force_pi;  
    pulse XTCK;  
  end;  
  cycle =  
    timeplate  
tp_capture;  
    pulse XTCK;  
  end;
```

Test Development and Characterization Process Flow



Create Vectors for Test System

- Mentor Graphics Fastscan™ may not be able to create pattern for some faults that may be considered false paths
 - Reported as “path/edge ambiguity in path”
- Write out STIL IEEE-1450 format from Mentor Graphics Fastscan™
- Current ATPG tools do not account for or understand path timing

Test Development and Characterization Process Flow

Characterize Launch/Capture Timing

- Shmoo launch and capture clock edges on path(s) on Teseda V520 test system
- Pushing launch edge to end of cycle and capture clock to beginning of cycle
- Be careful to note that there may be some conditions where the shift could fail as the launch and the shift clocks are the same timing (i.e. a Teseda V520 tester restriction)



Test Development and Characterization Process Flow



Characterize Launch/Capture Timing

- We were able to analyze and characterize an induced defect on a 0.25um product targeted path on the Teseda V520 test system.
- Probe pad FIB'd onto IC on delay path to induce a loading to model a defect.
- The Teseda V520 was placed under a Microscope and we placed a capacitance on a FIB'd pad to induce defect capacitive loading.

Test Development and Characterization Process Flow



Characterize Launch/Capture Timing

Load	Passing Launch to Capture Time
Probe+100pF capacitance	always failing out to 800ns
Probe+50pF capacitance	above 800ns
Probe+20pF capacitance	above 495ns
Probe+10pF capacitance	above 398ns
Probe+5pF capacitance	above 345ns
Probe capacitance is not known.	

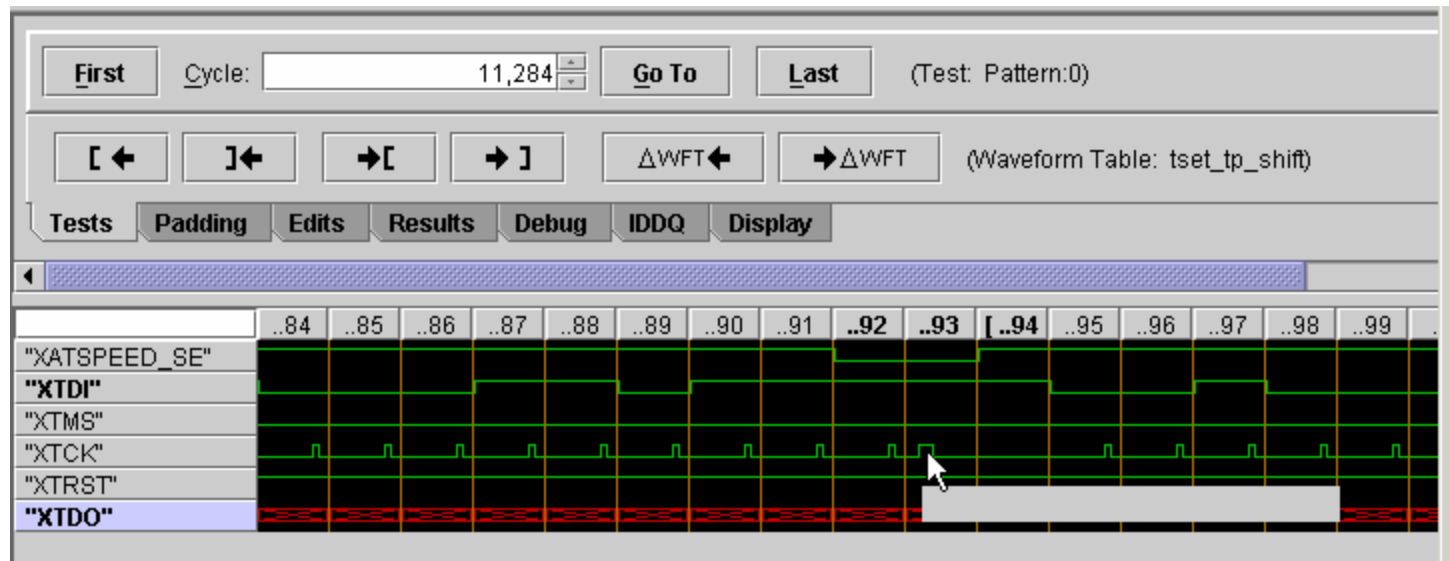
Sept. 12-15, 2005

Napa, California

© 2005 DPC

Path Delay Test Methodology

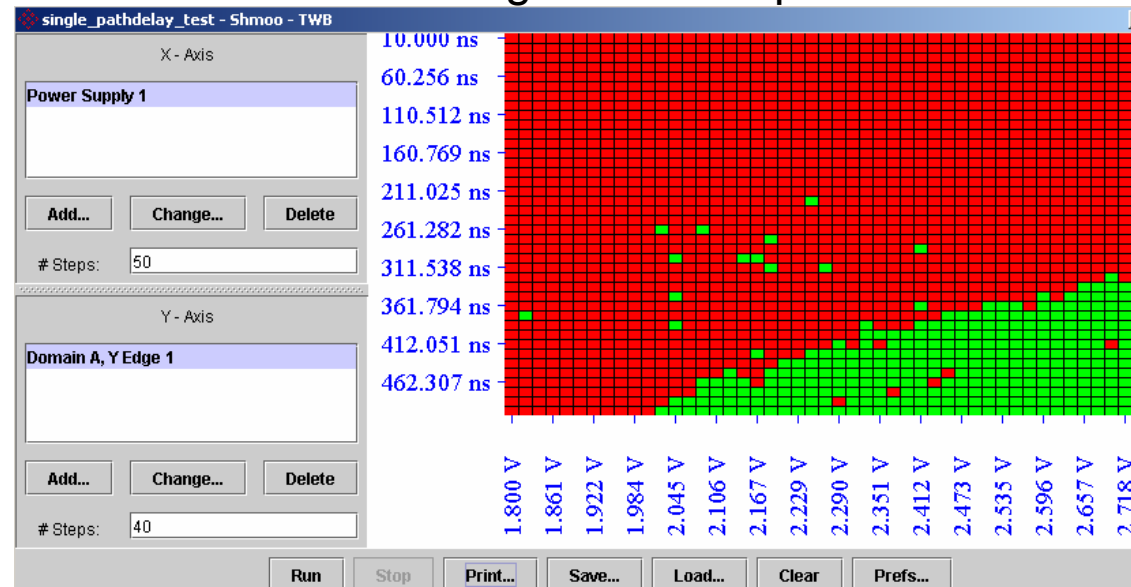
Characterize Launch/Capture Timing



Test Development and Characterization Process Flow

Characterize Launch/Capture Timing

- It is effective to detect defects that exhibit themselves down to 0.01pF effective load.
- This also shows that on this path that current Scan stuck-at and functional fast speed testing can only catch very gross defects that are much larger than 140pF effective loading.



Debug, Analysis and Correlation

- ATPG tools do not support debug diagnostics for path delay fault pattern failures
- Correct characterization to add process variation time guardbanding is important when looking for reliability defects instead of just yield critical timing related defects
- More data needs to be collected to define which defects can be screened as reliability issues and the path timing guardbanding

Sept. 12-15, 2005

Napa, California

© 2005 DPC

Conclusion

- ATPG tools do not have a good accurate or efficient method for generating path delay model-based patterns
- The Teseda V520 test system is efficient for Characterization/Debug of path delay patterns
- Correctly implemented path delay patterns can detect defective vias of hundreds of ohms
 - Good via is about 1-3 ohms
 - Bad vias can be 1Kohms to 5Kohms
- This data shows that accurate path delay testing can be done and results are repeatable (at least on the Teseda tester)