



Switching Voltage Regulator Noise Coupling Analysis for Printed Circuit Board Systems

Amy Luoh

Gene Garrison

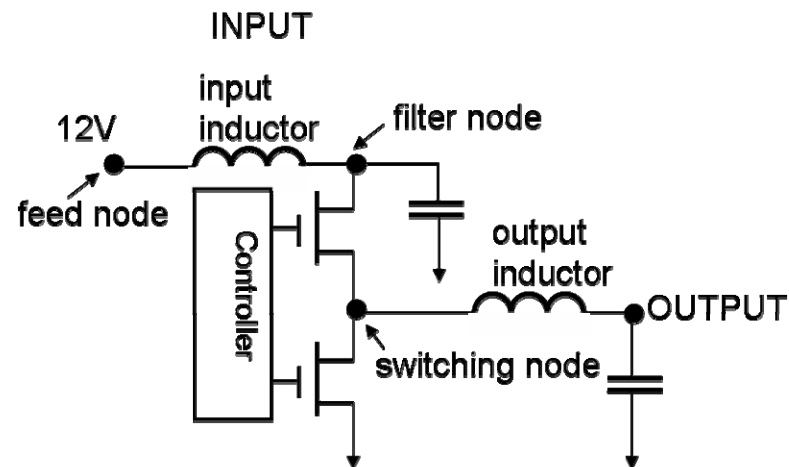
Jon Powell

Introduction

- I. We found we had VR noise coupling problems.
- II. We needed to find a method to predict and prevent these problems.
- III. We developed a methodology to use a commercial tool to simulate and quantify our problems.
- IV. We show correlation.

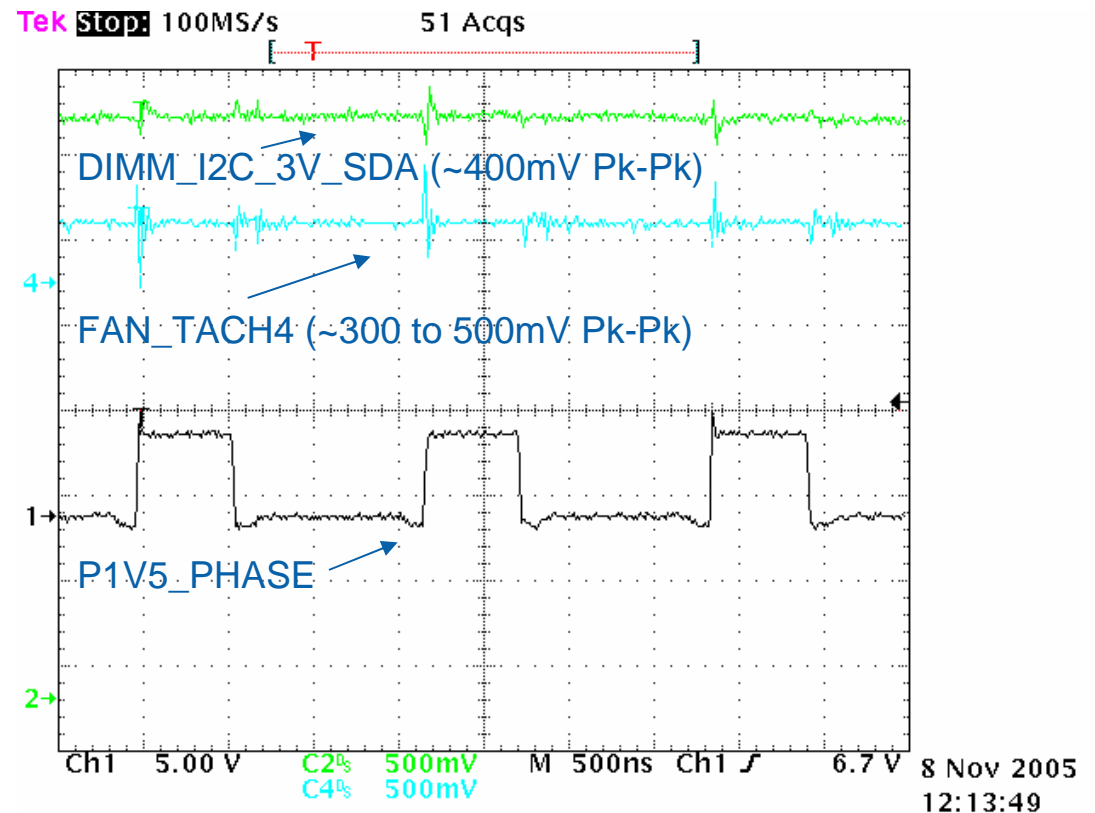
Define a Switching VR

- Buck regulator uses 2FETs to generate a lower voltage
- FETs are controlled by non-overlapping square waves.
- Input and output filters isolate the switching noise
- Primary source of noise is switching node, in this case switching between 0V and 12V and 0A to 25A.



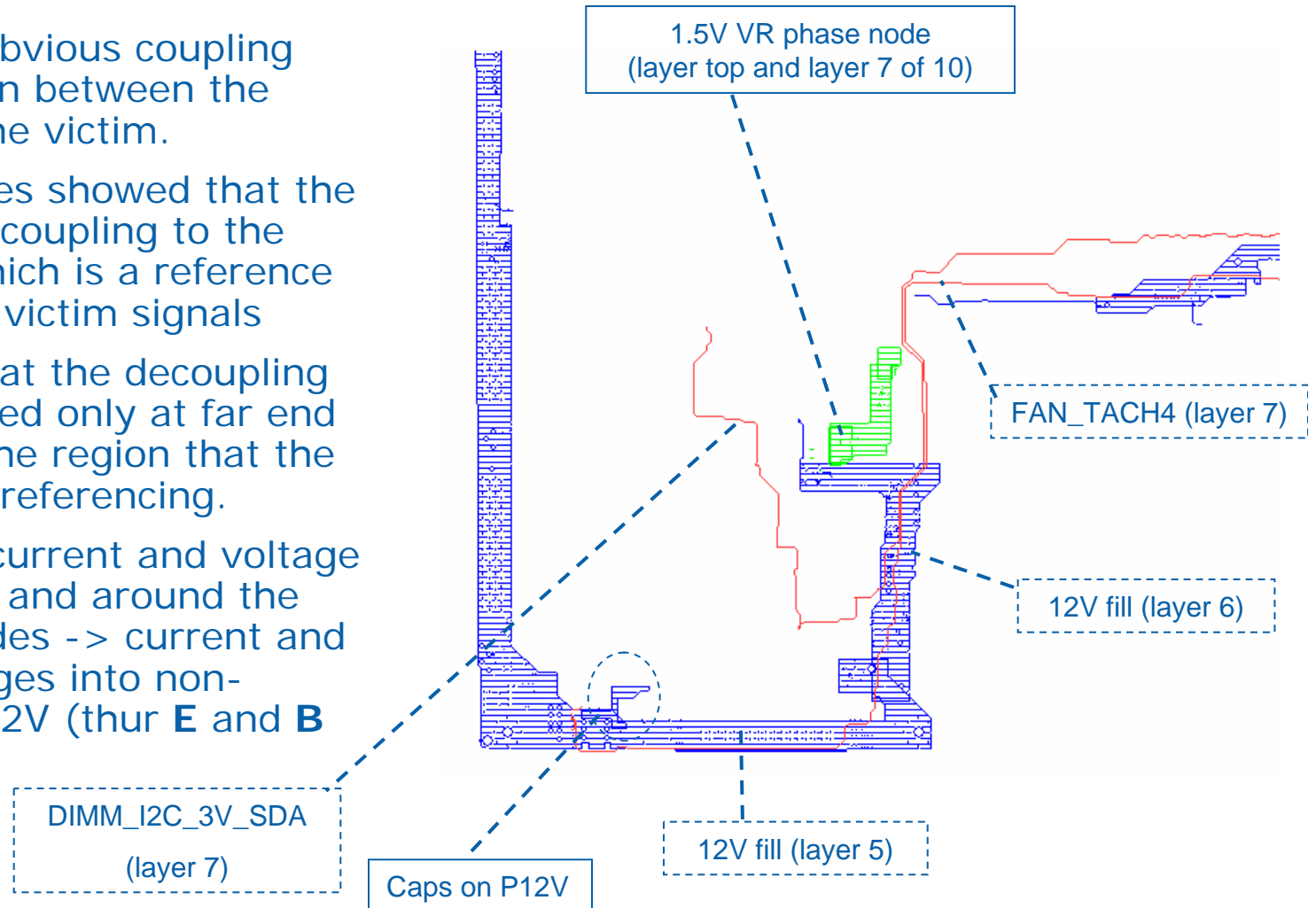
Initial VR problems

- System hang – 3 strikes counter failure.
- In the course of server systems debug efforts we discovered unexpected noise spikes on victim signals (I2C...).
- The frequency and magnitude of the spikes lead us to suspect VR noise.
- Simultaneous probing of the noise on I2C and VR nodes identified the source as a VR switching node.



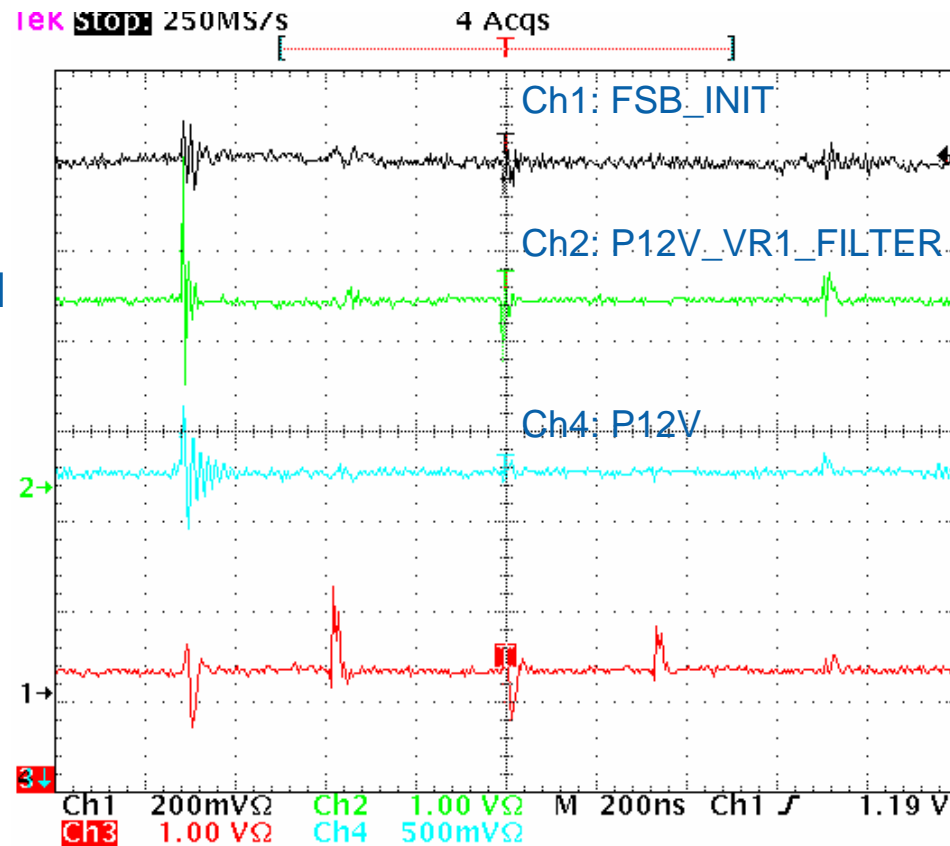
Further Investigation

- Initially, no obvious coupling path was seen between the source and the victim.
- Further studies showed that the 1.5V VR was coupling to the P12V fill – which is a reference plane for the victim signals
- Also found that the decoupling caps are placed only at far end but none in the region that the victims were referencing.
- Root cause: current and voltage transitions in and around the switching nodes -> current and voltage changes into non-decoupled P12V (thru **E** and **B** coupling).



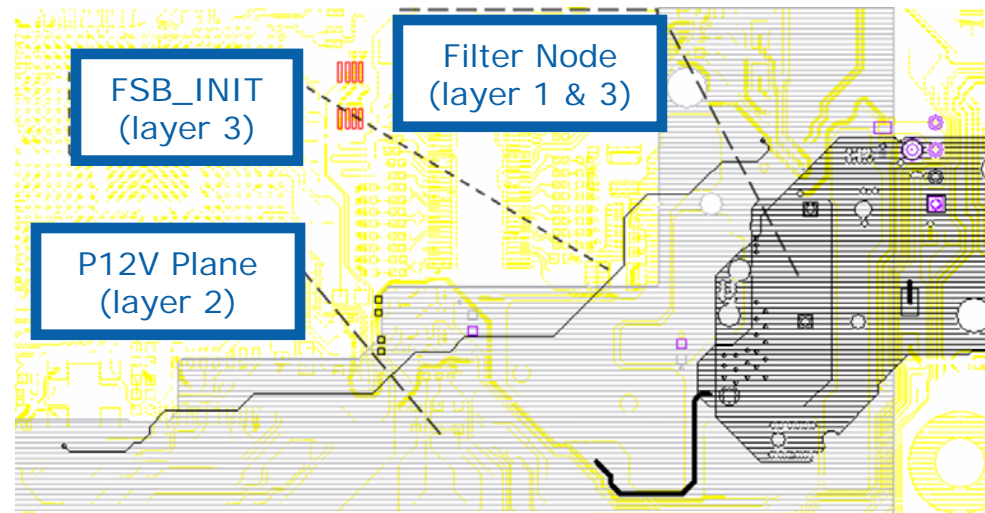
Another example

- Blue screen or system lockup at low temperature on a late fab design.
- Again, the frequency and magnitude of the spikes on FSB_INIT lead us to suspect VR switching noise.
- Simultaneous probing of noise on FSB_INIT and VR nodes identified the source as a VR filter node.



Further Investigation

- FSB_INIT is referenced to 12V fill.
- 12V fill was running underneath the VR and picking up noise from the filter nodes.



Why is VR noise coupling such an issue?

Most key:

- “Green” and cost considerations are driving more efficient VR’s
 - Which implies faster switching edges
 - And lower resistivity paths
 - We expect this trend to continue

Other factors:

- Signal margins are getting smaller
 - In both timing and amplitude
- The number of VR’s per board is growing
 - Driven by silicon process geometry shrinkage
 - Integrate more functionalities into the chip
- Routing density is increasing
 - Isn’t this always true?

Decision:

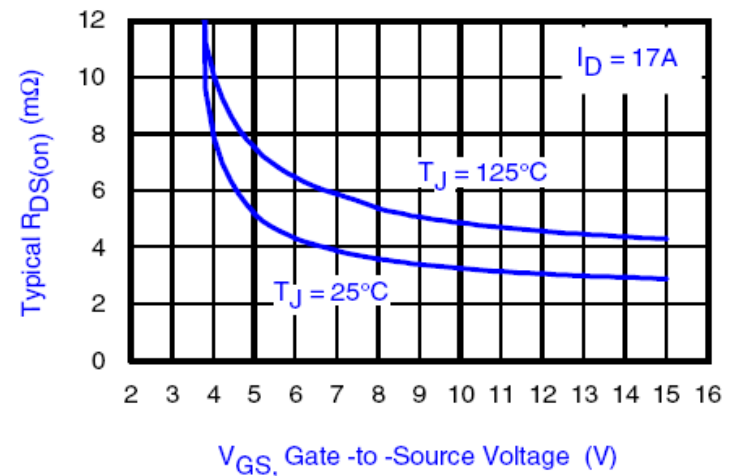
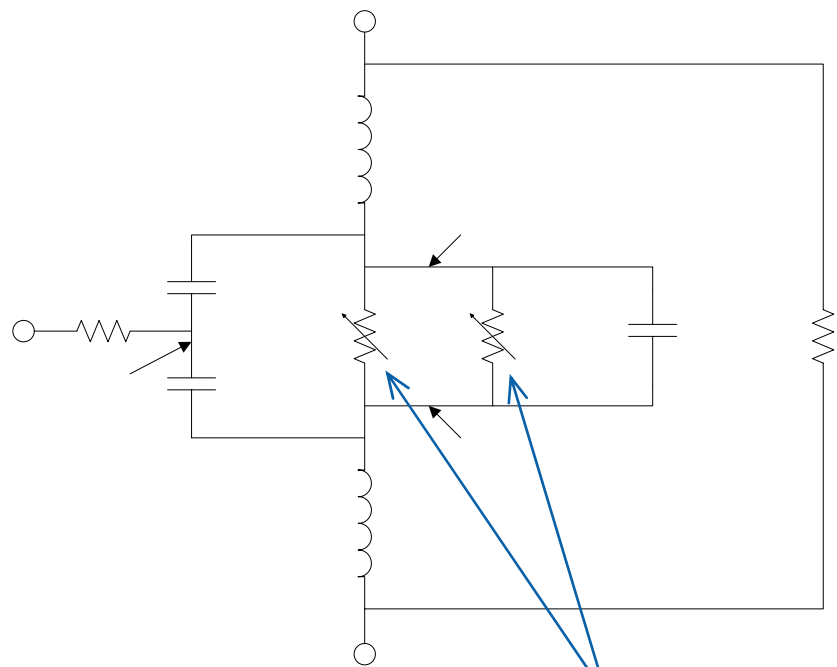
- We badly needed a simulation methodology that would enable us to detect and correct these problems before building boards

Development of Simulation Methodology

- These VR noise issues were proving to be very hard to predict or debug.
- We launched an effort to find a simulation technology.
- We needed a simulation tool that:
 - Would simulate coupling between planes, shapes and transmission lines
 - Would simulate coupling between vias
 - Would simulate lossy transmission line effects
 - Would be capable of full-board simulation with minimal translation effort
 - Would be reasonably fast
- We selected Sigriety's Speed2000 as best meeting these requirements

I. Models for the VR FETs

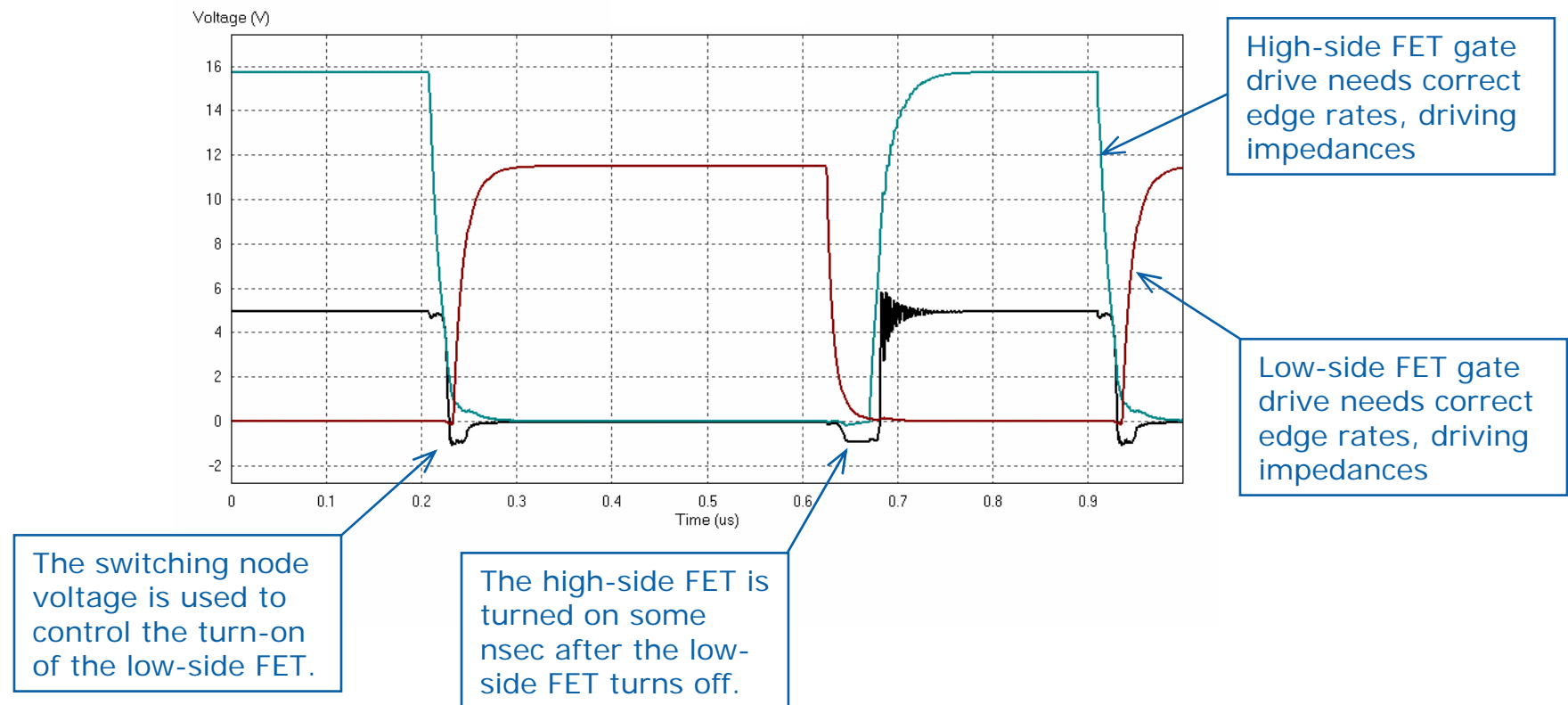
- Behavioral models gave us simulation stability and speed
- All element values can be derived from the device data sheet
- Correlation to BSIM models was excellent



Build as VCR model from FET data sheet or from BSIM model

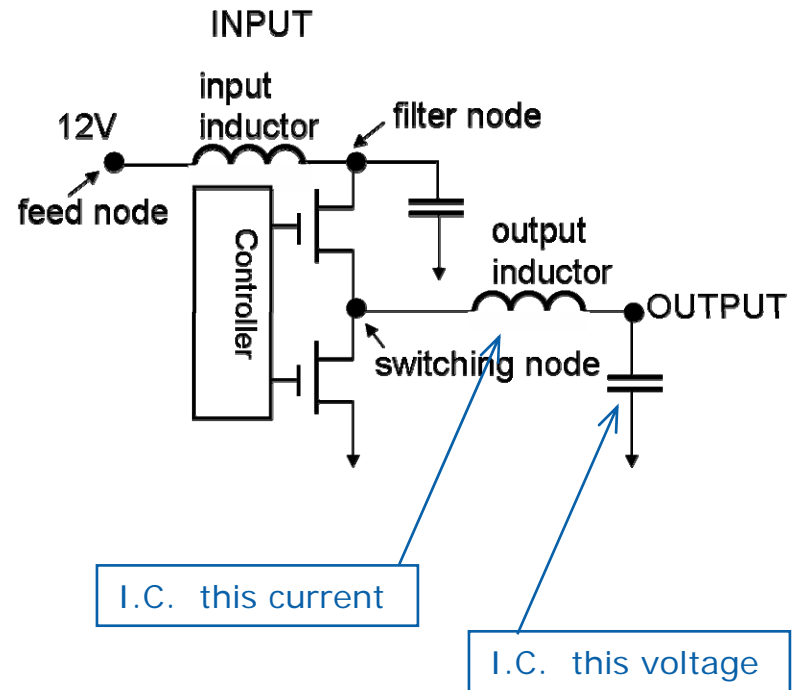
II. Models for the VR controller

- The VR controller needs to be carefully modeled to provide the correct stimulus to the FETs.
- Implemented as a complex combination of VCVS/VCR/CCVS etc.



III. Initial conditions are key

- It takes hundreds of milliseconds for an VR to initialize and settle into its steady state
 - This would translate into months of simulation time
 - We want the simulation to start with the VR already “settled”
 - pre-calculate output inductor current and output capacitor voltage and apply those as initial conditions in Speed2000 simulation
 - Caution: Initial conditions behaviors in Hspice and in Speed2000 are different



IV. Automated the process for simulation setup

- A lot of effort to understand how best to model victim signals, with best tradeoff between accuracy, stability, setup time, and runtime
 - IBIS model is more accurate but increase simulation time and unstable
 - simple RC models are much more efficient and provide adequate accuracy
- Developed a setup script to find all drivers, receivers, resistive loads, capacitive loads
 - Attach models to discrete resistors and capacitors
 - Model driver and receiver pins as an RC to the closest GND
- VR output loading
 - A simple resistor in the approximate center of the output voltage plane is sufficiently accurate
 - Placed manually for the VR being simulated

V. Post processing script

- We developed a post processing script that takes the large amounts of output data and reports the worst case pk-pk noise for each receiver
- Worst case noise is reported for each VR phase

	A	B	C	D	E	F	G	H	I	J
	NET	PHASE	PART	PIN	G_PIN	T_MIN	V_MIN	T_MAX	V_MAX	V_DELTA
260	P5V		C5E6	1		4.30E-07	-1.23E-01	4.25E-07	1.07E-01	0.2305051
261	P5V	WC=1	C5E6	1		4.30E-07	-1.23E-01	4.25E-07	1.07E-01	0.2305051
262	P5V		C5E7	1		4.67E-07	-1.24E-01	4.15E-07	1.06E-01	0.2302194
263	P5V	WC=1	C5E7	1		4.67E-07	-1.24E-01	4.15E-07	1.06E-01	0.2302194
264	VR_CPU1_PWM2_E		U5E3	2		9.12E-07	-1.74E-01	9.09E-07	5.53E-02	0.2291302
265	VR_CPU1_PWM2_E		U5E3	2		1.91E-06	-1.60E-01	1.88E-06	6.85E-02	0.22867893
266	VR_CPU1_SW2		U5E3	8		1.91E-06	-1.68E-01	1.91E-06	5.76E-02	0.2259483
267	P12V_CPU1		C5P26	1		1.42E-06	1.19E+01	1.42E-06	1.21E+01	0.22562
268	P12V_CPU1		C5B16	1		1.48E-06	1.19E+01	1.41E-06	1.21E+01	0.22552
269	V_IO_VSYN_R		U2D4	6		4.99E-07	-1.12E-01	4.94E-07	1.13E-01	0.2242965
270	V_IO_VSYN_R	WC=1	U2D4	6		4.99E-07	-1.12E-01	4.94E-07	1.13E-01	0.2242965
271	P5V		C5A14	1		4.50E-07	-1.10E-01	4.56E-07	1.14E-01	0.2241876
272	P5V	WC=1	C5A14	1		4.50E-07	-1.10E-01	4.56E-07	1.14E-01	0.2241876
273	P12V_CPU1		C5B14	1		1.40E-06	1.19E+01	1.41E-06	1.21E+01	0.22406
274	P12V_CPU1		C5N13	1		1.89E-06	1.19E+01	1.90E-06	1.21E+01	0.22208
275	P3V3		C4F14	1		4.53E-07	-1.03E-01	4.48E-07	1.15E-01	0.2183612
276	P3V3	WC=1	C4F14	1		4.53E-07	-1.03E-01	4.48E-07	1.15E-01	0.2183612
277	P12V_CPU1		C9A14	1		1.34E-06	1.19E+01	1.44E-06	1.22E+01	0.21756
278	VR_CPU1_PWM1_U		U5D1	1		3.87E-07	-1.04E-01	4.00E-07	1.12E-01	0.215604
279	JTAG_CPU0_TDI		U4M1	1		4.05E-07	-9.42E-02	3.89E-07	1.20E-01	0.21383083
280	JTAG_CPU0_TDI	WC=1	U4M1	1		4.05E-07	-9.42E-02	3.89E-07	1.20E-01	0.21383083
281	P12V_CPU1		C9A13	1		1.34E-06	1.19E+01	1.44E-06	1.21E+01	0.21263
282	VR_CPU1_PWM1_E		U5D1	2		4.13E-07	-1.68E-01	4.10E-07	3.97E-02	0.20751651
283	P12V_CPU1		C1L3	1		1.34E-06	1.19E+01	1.44E-06	1.21E+01	0.20679
284	P5V		C5B15	1		4.50E-07	-9.99E-02	4.55E-07	1.03E-01	0.20334316
285	P5V	WC=1	C5B15	1		4.50E-07	-9.99E-02	4.55E-07	1.03E-01	0.20334316
286	VR_CPU1_SW1		U5D1	10		4.13E-07	-1.65E-01	4.10E-07	3.77E-02	0.20293039
287	VR_CPU1_SW4		Q5D3	5		9.12E-07	-1.35E-01	9.09E-07	6.68E-02	0.20194169
288	VR_CPU1_SW4		Q5D3	7		9.12E-07	-1.35E-01	9.09E-07	6.68E-02	0.20194169
289	CSL_CPU0_SKTOCC		U4M1	6		4.24E-07	-9.48E-02	4.18E-07	1.06E-01	0.20030664
290	CSL_CPU0_SKTOCC	WC=1	U4M1	6		4.24E-07	-9.48E-02	4.18E-07	1.06E-01	0.20030664
291	P3V3		C4F14	1		4.43E-07	-0.94E-01	4.38E-07	1.04E-01	0.19999165

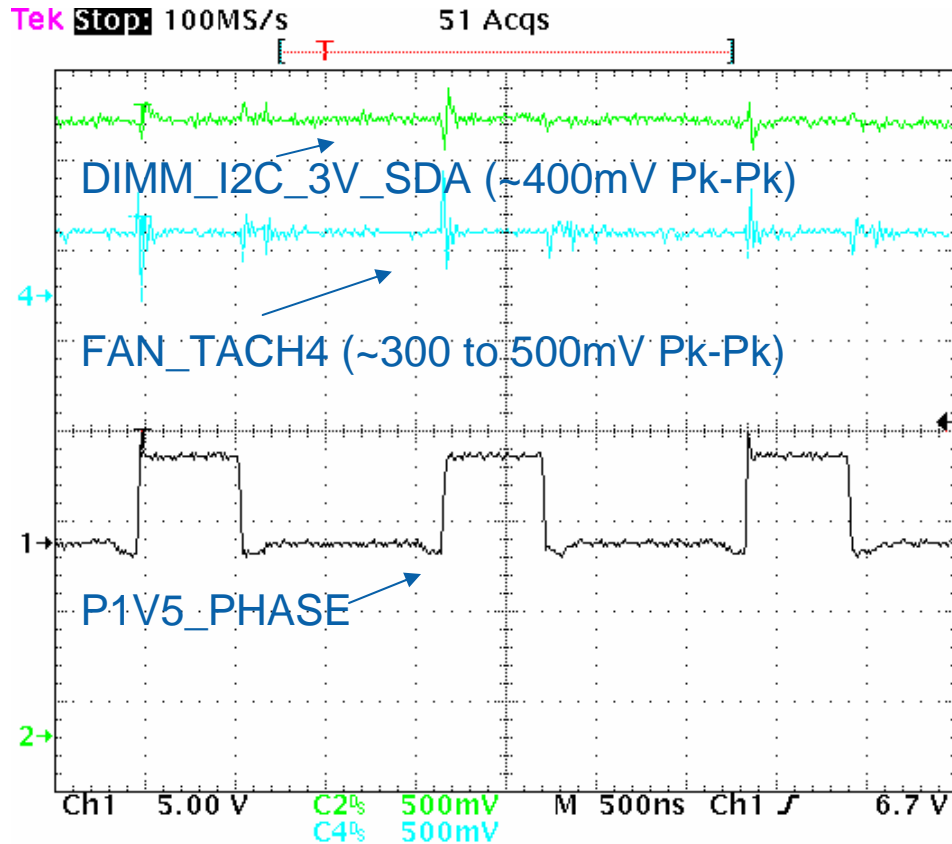


Simulation Process

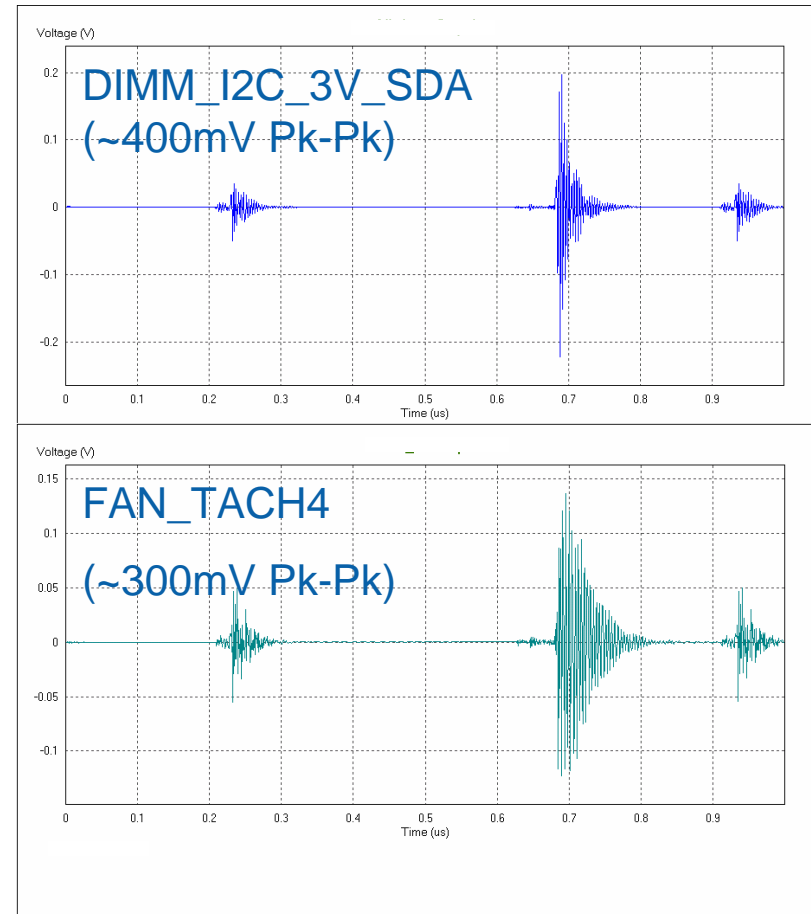
1. Make VR models, including behavioral FETs and controller models, verify with SPICE
2. Generate input database with automated script, import into Speed2000
3. Run simulation on a given VR
4. Post process to extract worst case noise peaks
5. Have SI engineers review the report and identify any problem victim signals
6. Identify the coupling mechanism to the problem nets, using the simulation results to verify
7. Devise a fix and test the fix in simulation
8. Implement the fix in the board

Initial Problem, Correlation

Measured results:



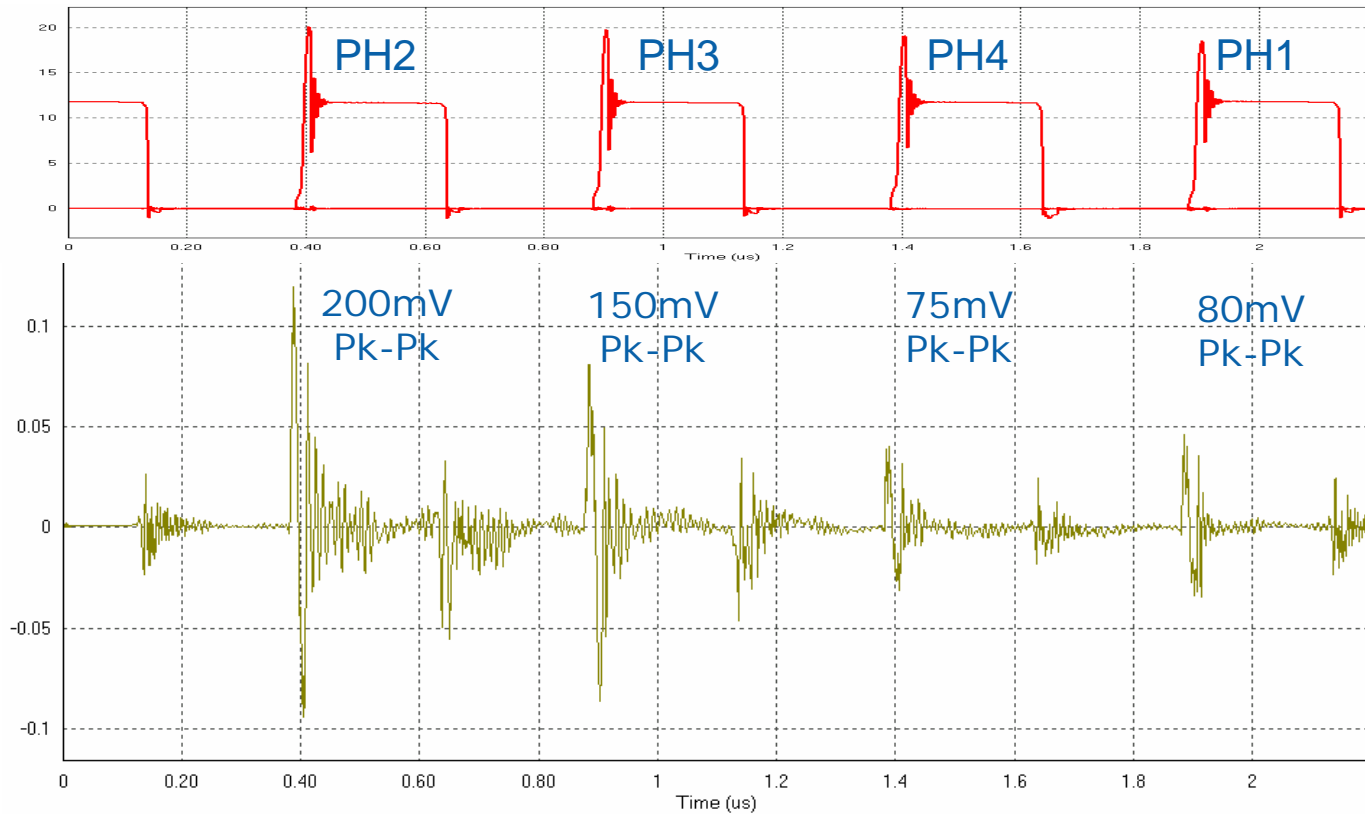
Simulated results:



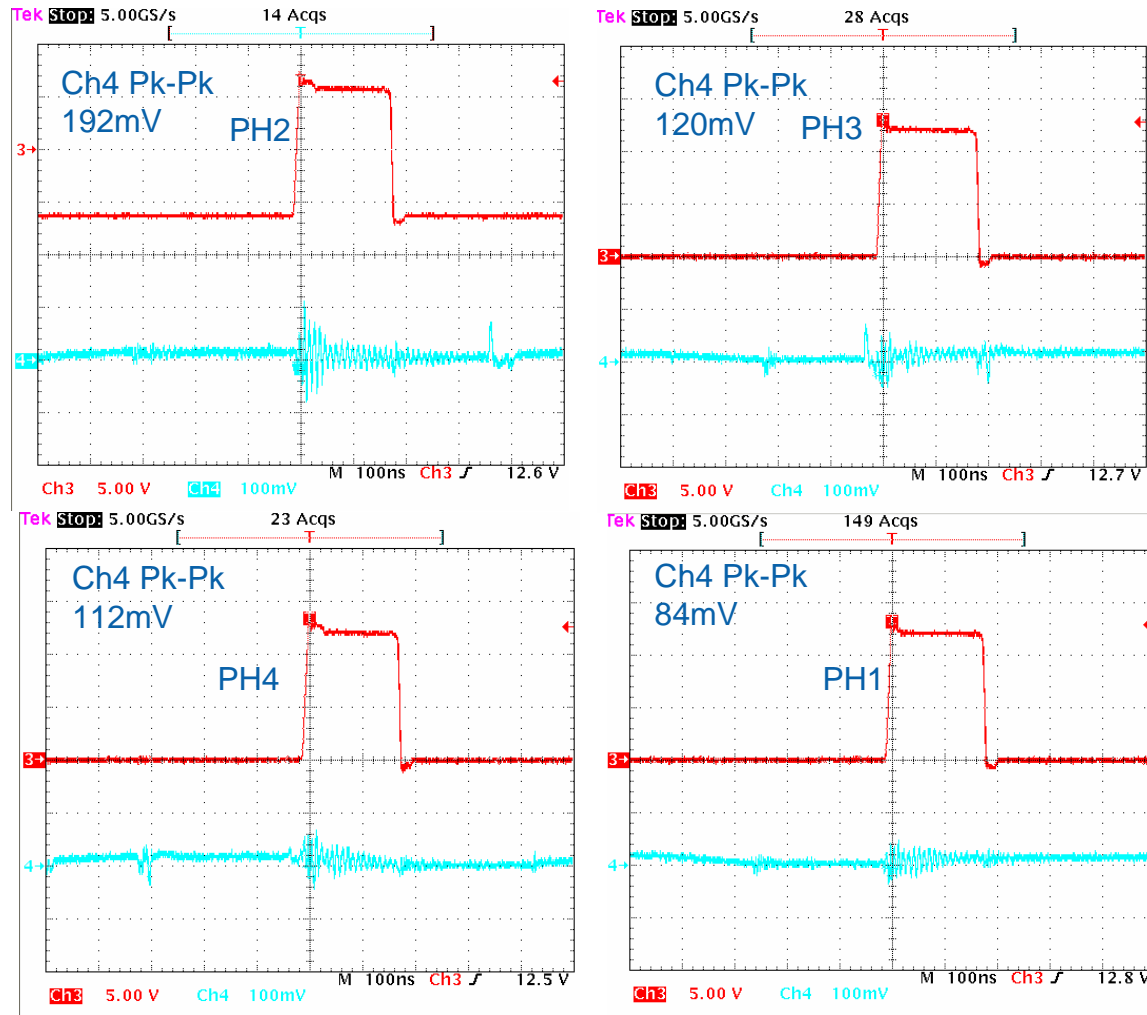
A New Design – The Process

Simulations show excessive noise on JTAG signals

- This signal is a 1.1V level signal
- This signal is associated with CPU debug functionality and may make the CPU hang

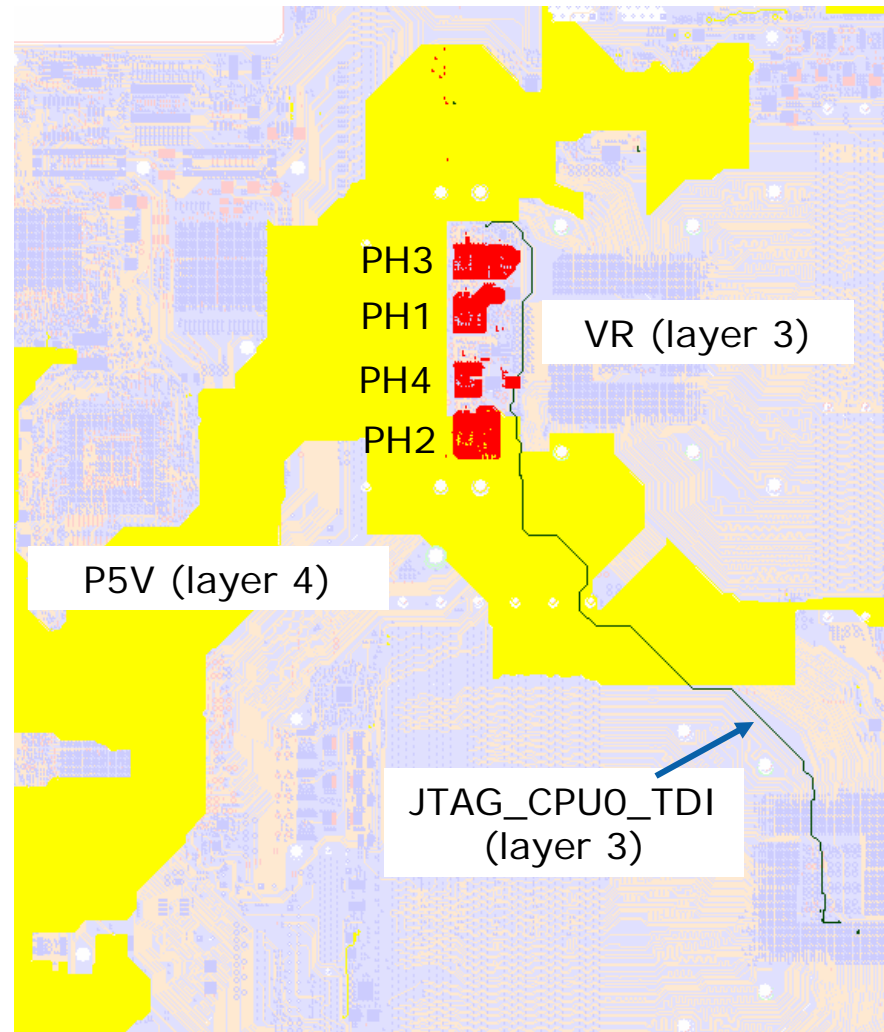


JTAG Measurement Results



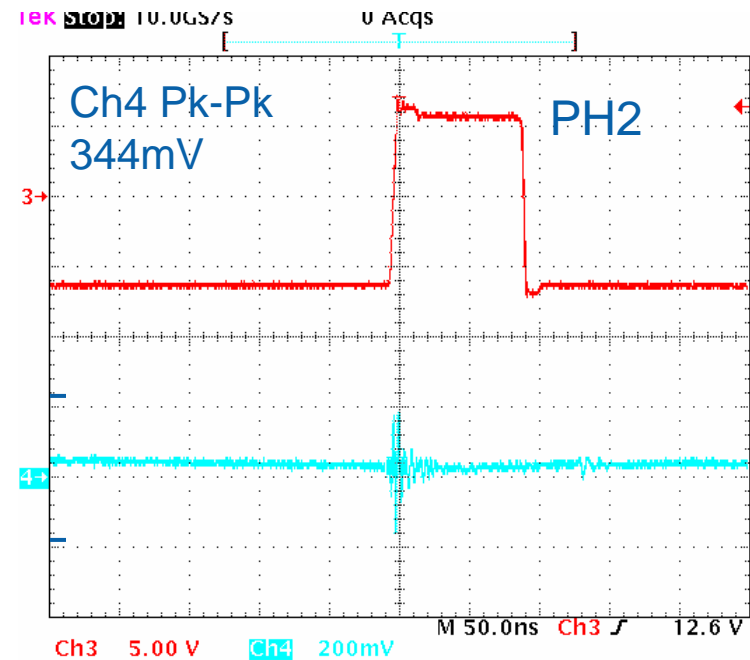
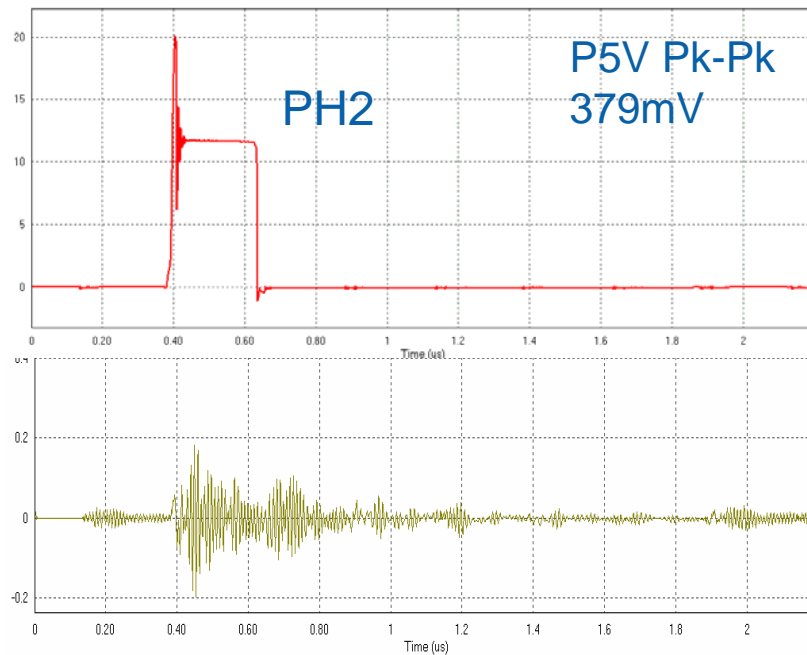
JTAG Layout

Where's the coupling?

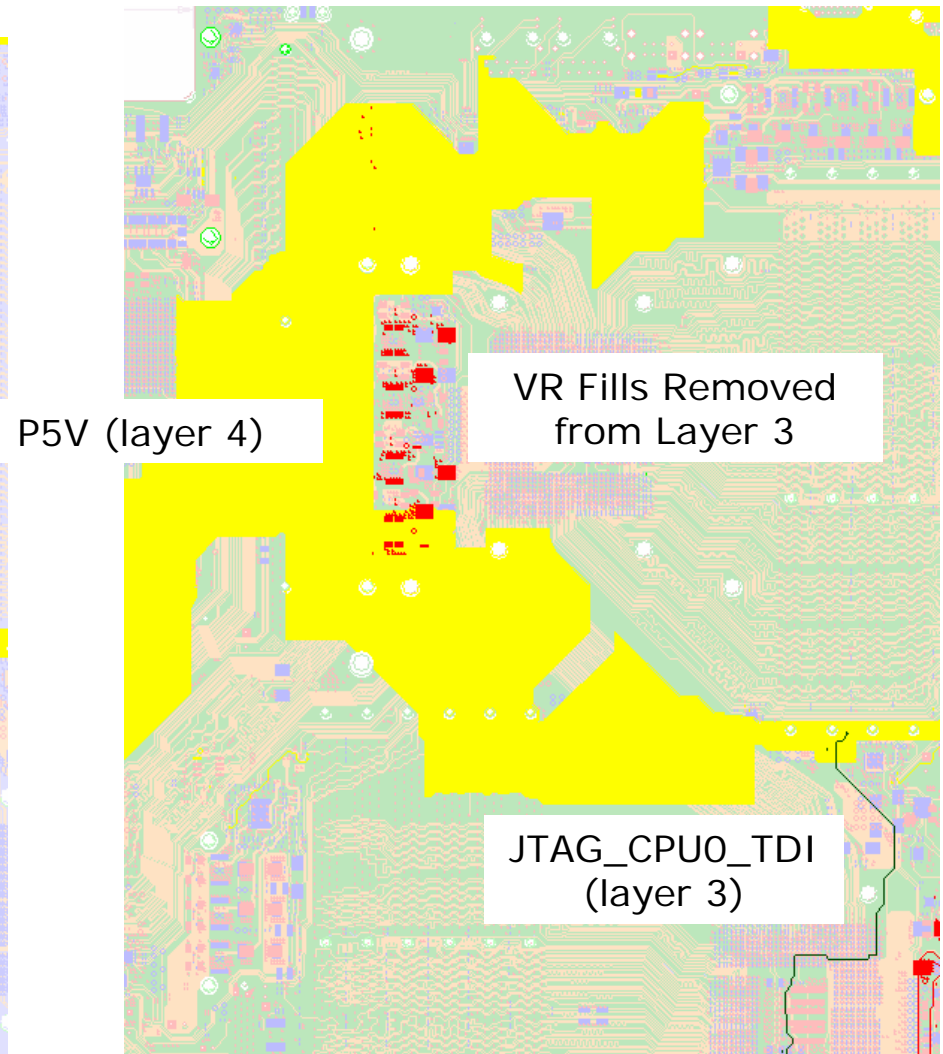
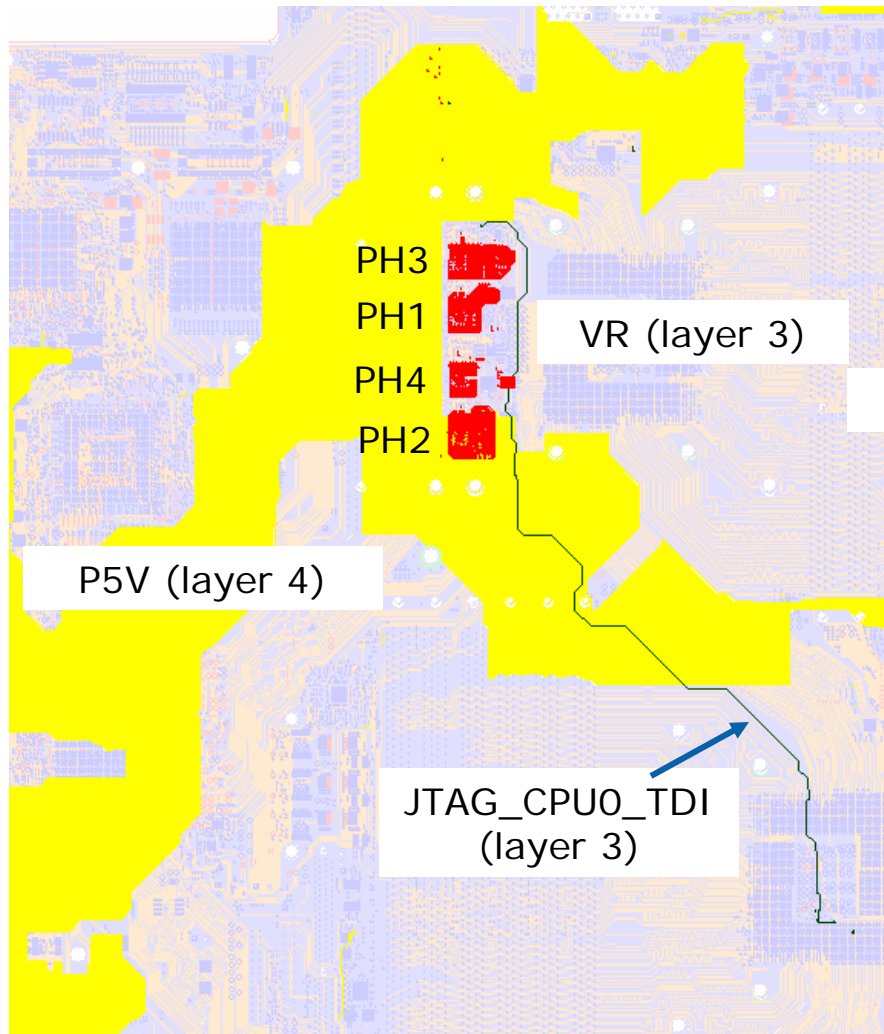


5V Power Fill

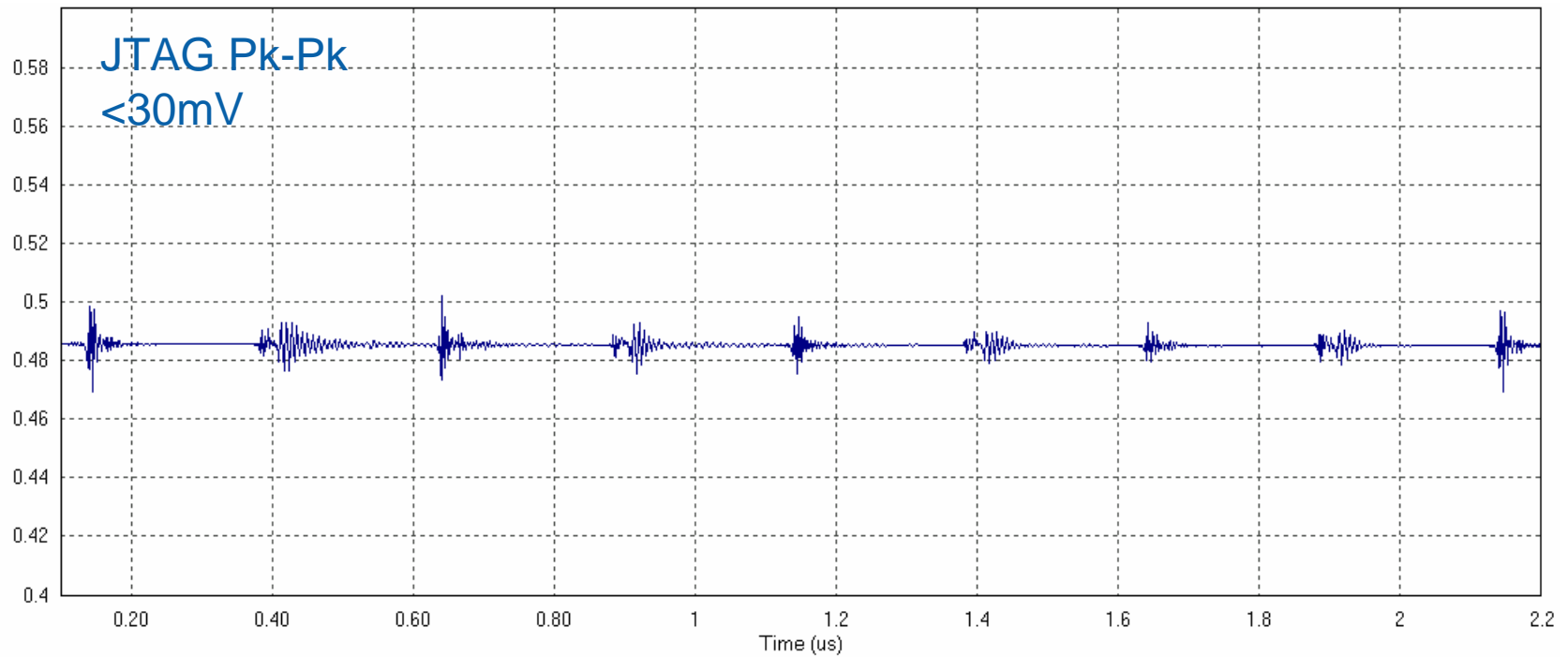
Simulations show noise on 5V fill, verified in measurement.



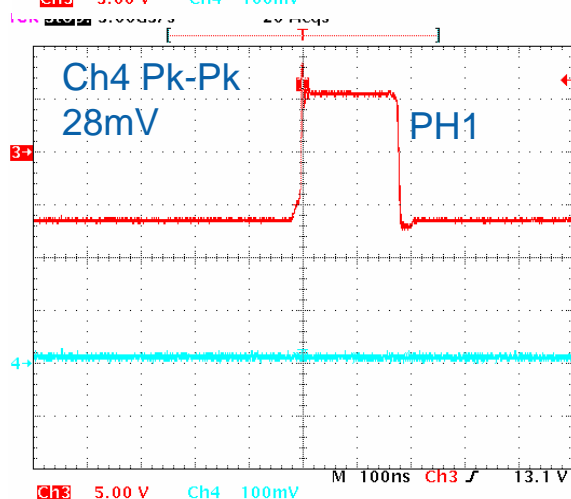
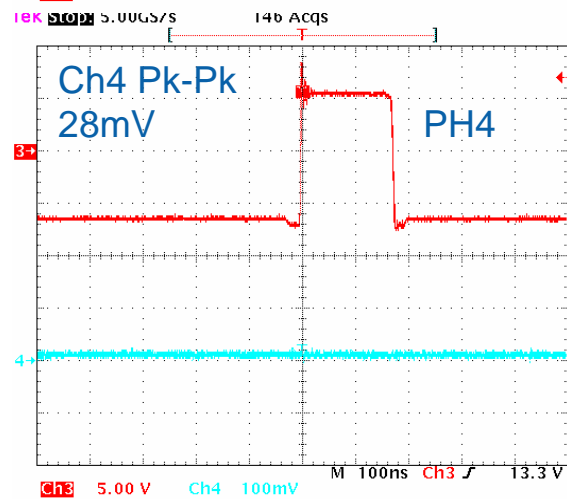
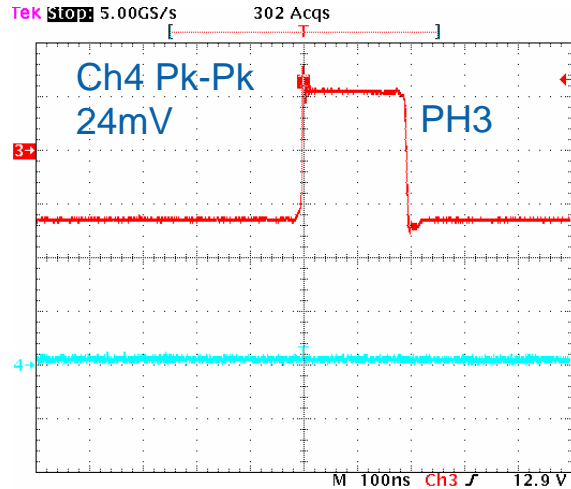
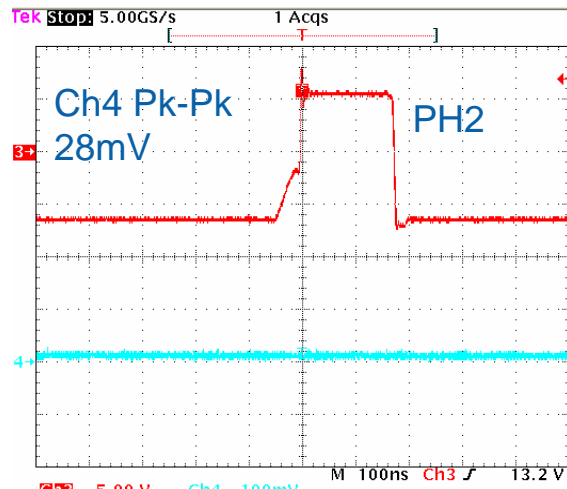
JTAG repaired layout



Repaired JTAG Simulation



Repaired JTAG Measurements



What is the cost?

- We have described how we made every effort to simplify and to quicken the simulations.
- Even so, to simulate a 2 CPU socket, 8 layer server board we need:
 - A 64 bit OS
 - 24 GB of RAM
 - In a dual socket, 4 core system
 - And about 3 days of simulation time per VR phase
- We have two of these machines available
 - Plus the 4 Speed2000 licenses
- Then there was the 2 years to development this methodology
- And for each project
 - FET model development
 - VR Controller model development
 - We maintain an IBIS library of parts which we leverage

Summary

- The Methodology has been used to identify coupling paths in problems that were found in the lab, and to develop and test fixes in simulation.
 - This has demonstrated the sufficient accuracy of The Methodology.
- The Methodology is being used to identify potential issues and to drive and validate corrective action before building boards.
- Through the use of The Methodology, we have learned to identify high risk PCB constructs that we could then avoid with specific design rules.
 - This leads to fewer problems.
 - Implementation as Design rules is relatively cheap and easy.
- The major shortcoming: excessive computational simulation time.
- The tool is expensive in terms of the software, the hardware, and the simulation time involved, but we believe the cost is necessary.

Discussion



Speed2000 - SpeedGen

Workflow: SPEED GENERATOR

SSO - Auto IBIS Simulation

Power Ground Noise Simulation

Layout Setup

- Load Layout File
- Check Stackup
- Prepare Power Ground Nets

Simulation Setup

- Add Excitation
- Setup VRM & Decaps
- Specify Simulation Time
- Specify Observations (Voltage)
- Specify Observation (Current)
- Auto Mesh Setup
- Error Check and Warning

Simulation

- Start Simulation

What If Analysis

- Change Decaps
- Stackup

Transistor Simulation

EMC/EMI Radiation Simulation

Customize Workflow

Circuit/Linkage Manager

A	Ckt Name	Model Name
✓	Q6A1	FET_NCH4D1...
✓	Q5E3	NPN_SMLF-M...
✓	Q5E2	NPN_DUAL_S...
✓	Q5E1	FET_NCH4D2...
✓	Q5D4	FET_NCH4D1...
✓	Q5D3	FET_NCH4D2...
✓	Q5D2	FET_NCH4D1...
✓	Q5D1	FET_NCH4D2...
✓	Q5C3	FET_NCH4D1...
✓	Q5C2	FET_NCH4D1...

Ckt Node	Pkg Node	Layer Name
6	Node3914!16::P1...	Signal\$TOP
5	Node3913!15::P1...	Signal\$TOP
4	Node3912!14::P1...	Signal\$TOP
3	Node3911!13::P1...	Signal\$TOP
2	Node3910!12::VR...	Signal\$TOP
1	Node3909!11::VR...	Signal\$TOP

Link Unlink

```

cgd dd ing 210e-12
rg gg ing 1.7
cgs ing ss 1360e-12
cds dd ss 280e-12

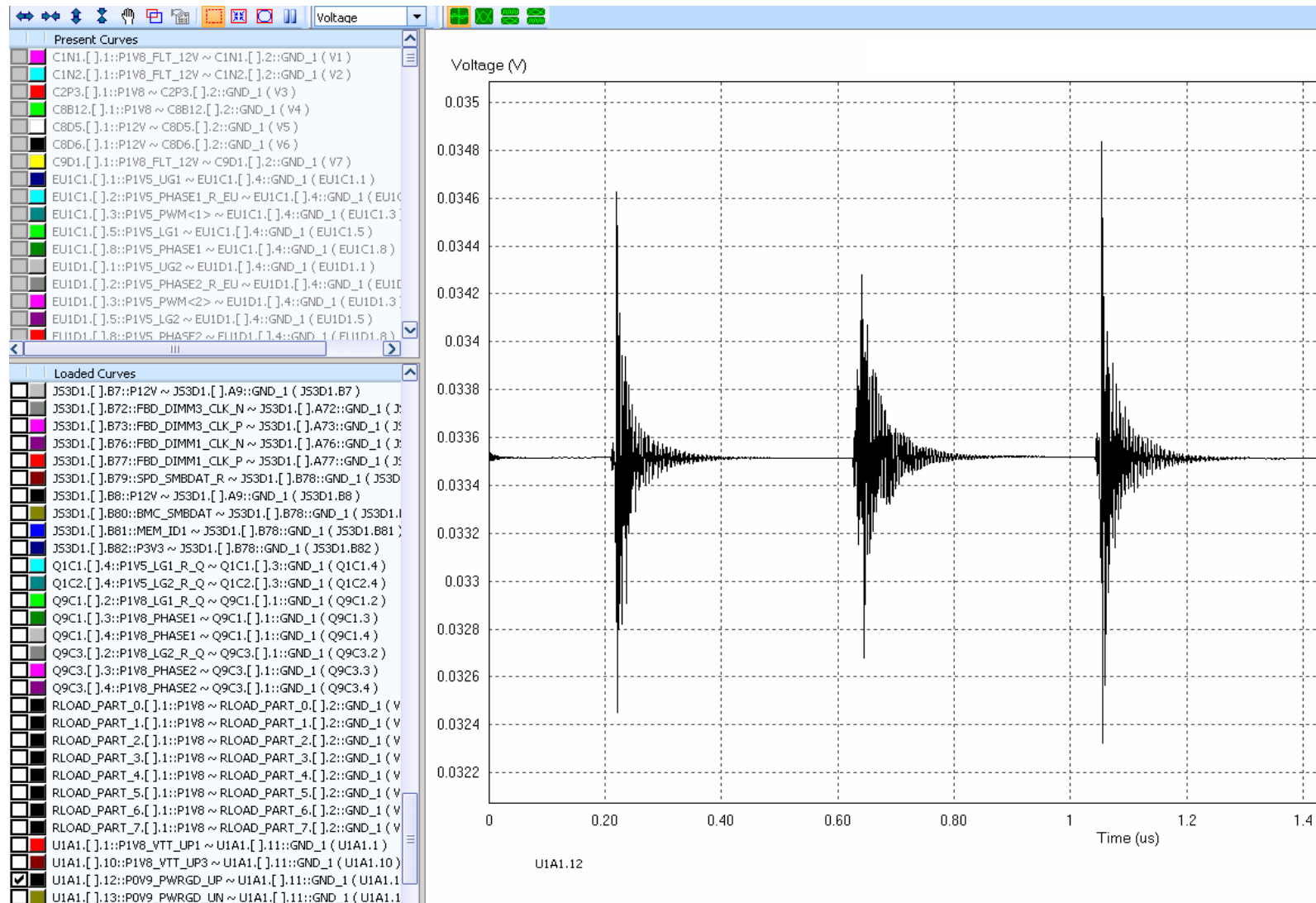
Gds dd ss VCR PWL(1) ing ss
+ -2.00E+01,4.99E+07 2.64E+00,4.99E+07 2.65E
+ 2.67E+00,2.42E+02 2.68E+00,1.32E+02 2.69E
+ 2.71E+00,4.15E+01 2.72E+00,3.16E+01 2.73E
+ 2.75E+00,1.66E+01 2.76E+00,1.40E+01 2.77E
+ 2.79E+00,8.95E+00 2.80E+00,7.87E+00 2.81E
+ 2.83E+00,5.60E+00 2.84E+00,5.06E+00 2.85E
+ 2.88E+00,3.54E+00 2.90E+00,3.03E+00 2.92E
    
```

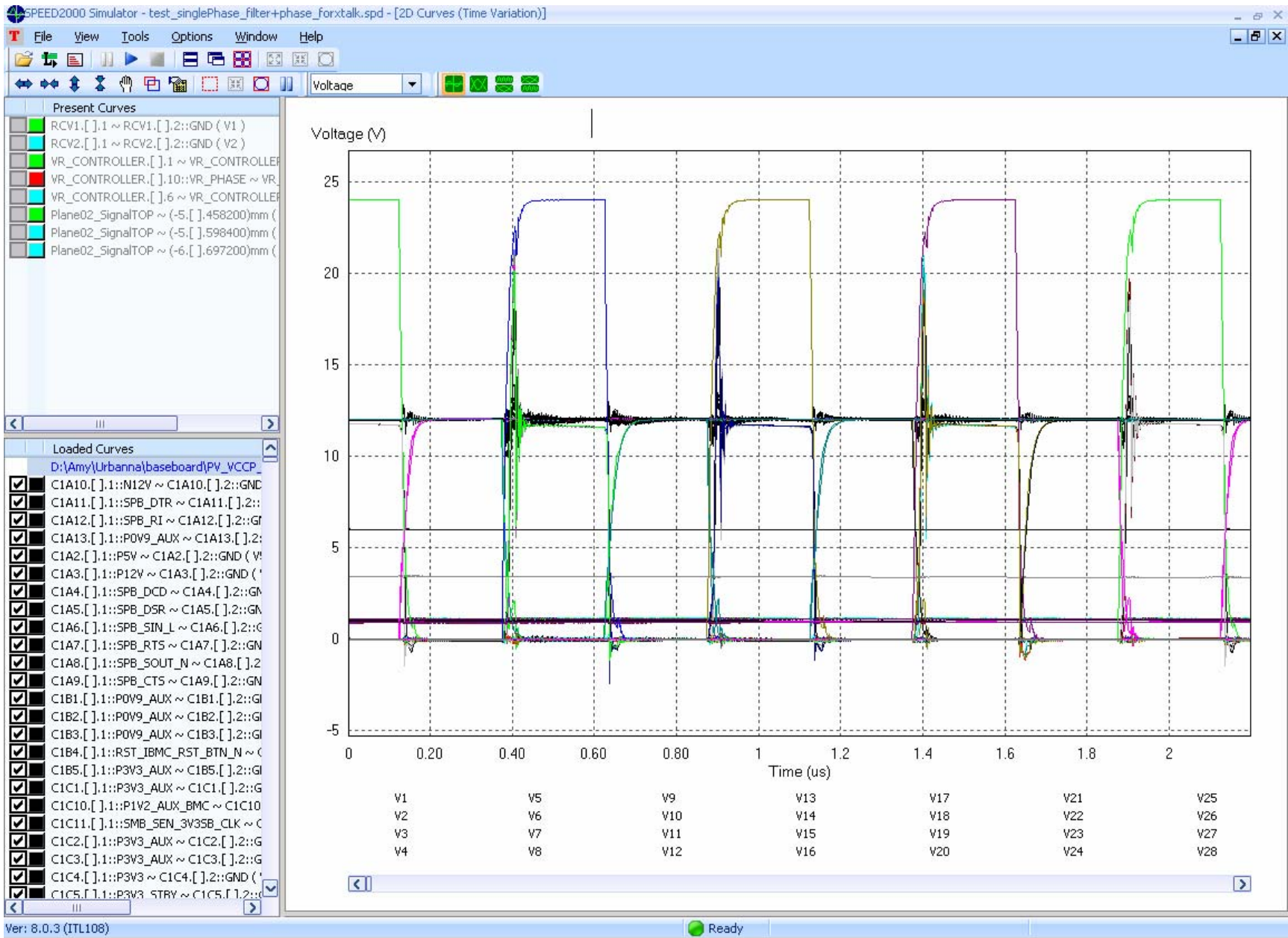
Output

Miscellaneous Mesh - Errors VariablesCheck

Ver: 8.0.0 (ITL53) Mouse(mil): X: 6442.54, Y: 534.33 Ready Finishing project loading

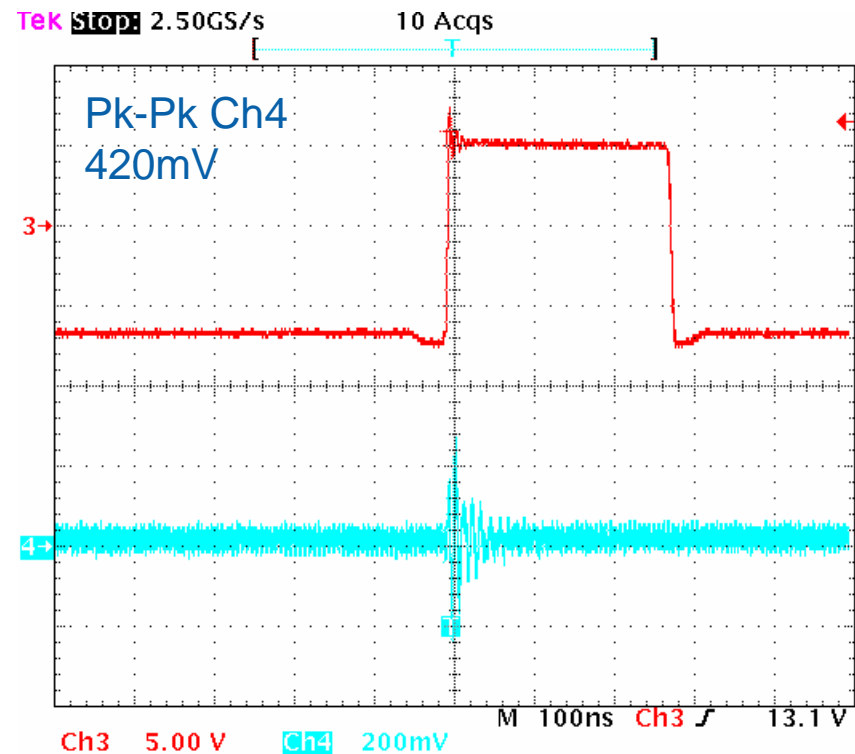
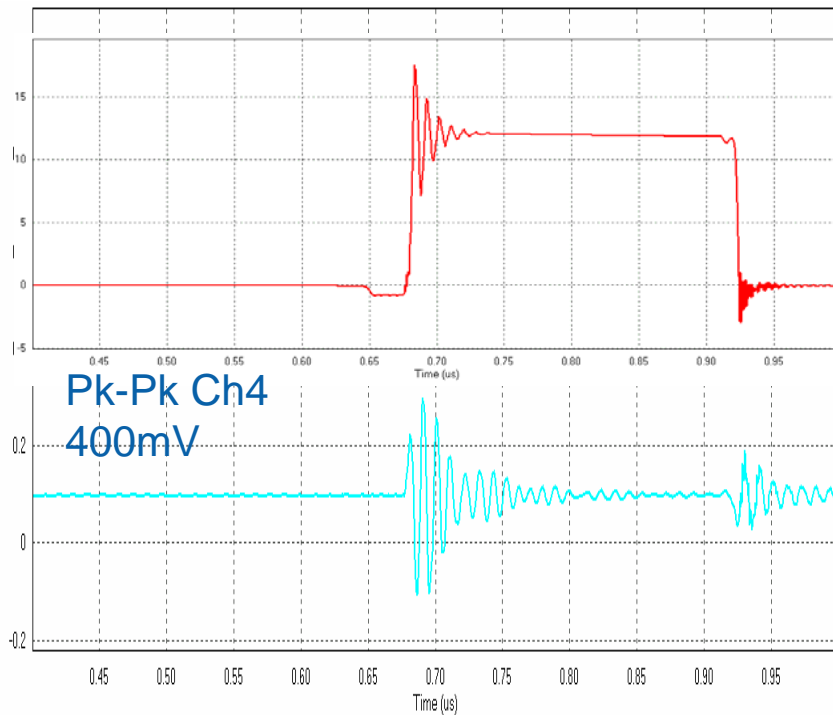
Speed2000 - SpeedSim



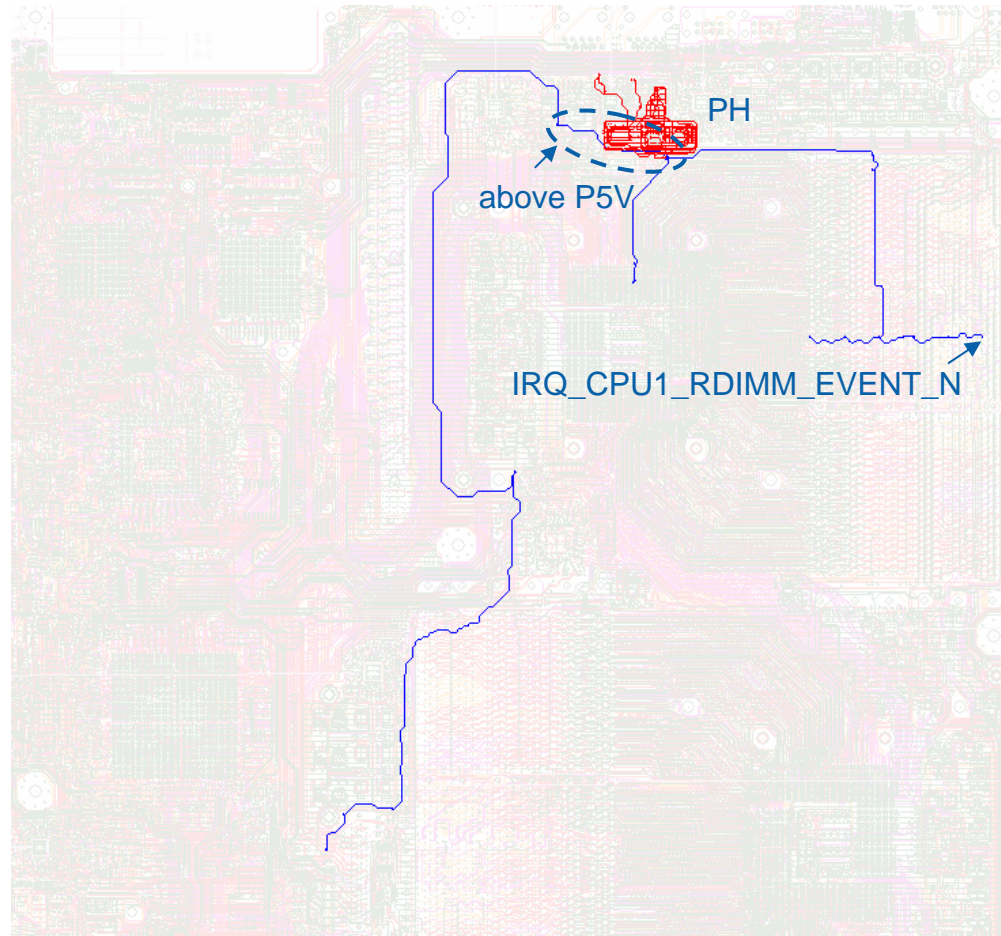


On the same design, IRQ - Measurements and Simulation

- Another interface on this board where simulations showed a coupling issue, confirmed by measurement.



IRQ Layout (Fab 1)



IRQ on Fab2 – After the fix

- Because these simulations take so long to complete, we were not able to re-simulate a fix before the next tape-out. So we implemented our best guess of a fix.
- Measurements and simulations of the fix agree – we should have simulated before implementing.

