AN ABSTRACT OF THE THESIS OF

Martin A. Held for the degree of Master of Science in

Electrical and Computer Engineering presented on June 17, 2005.

Title: A Methodology for Efficient Substrate Noise Estimation from Large Scale Digital Circuits in Mixed Signal SoC's

Abstract approved:

Redacted for Privacy

Terri S. Fiez

Redacted for Privacy

Kartikeya Mayaram

A methodology for rapid estimation of substrate noise generated by digital circuitry in mixed-signal circuits is presented. This methodology is incorporated into the Silencer! framework, and also provides for future improvements including pre-layout noise estimation.

Measurements of a test chip fabricated in the TSMC 0.25μm heavily doped logic process validate the methodology and simulation results. Substrate noise coupling from a large-scale digital processor to simple analog circuits for two different designs with the same functionality has been analyzed. Compared to existing methodologies there is a speedup of over 80 times with one-tenth the memory use is obtained with only a 5 percent loss in accuracy. This allows for the rapid identification of major noise injection mechanisms and solutions for their reduction.

A Methodology for Efficient Substrate Noise Estimation from Large Scale Digital Circuits in Mixed Signal SoC's

by

Martin A. Held

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented June 17, 2005
Commencement June 2006

Master of Science thesis of Martin A. Held presented on June 17, 2005

APPROVED:

Redacted for Privacy

Co-Major Professor, representing Electrical and Computer Engineering

Redacted for Privacy

Co-Major Professor, representing Electrical and Computer Engineering

Redacted for Privacy

Director of the School of Electrical Engineering and Computer Science

Redacted for Privacy

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Redacted for Privacy

Martin A. Held, Author

ACKNOWLEDGMENT

During my 7 years in the university, I've been able learn from a wide variety of subjects. I know for a fact that this would all have been impossible without the dedicated people who work for the university, all the people who have ever helped me out deserve acknowledgement.

Primarily, however, I would like to thank Dr. Terri Fiez and Dr. Kartikeya Mayaram, who have helped me throughout this project, and also found and provided the means for me to do so. I would also like to thank the Semiconductor Research Corporation (SRC), the Defense Advanced Research Projects Agency (DARPA), and Boeing for providing funding, as well as Artisan Components, Inc and Theseus Logic, Inc. for providing material support.

A person often sought out by graduate students is Robert Batten. His knowledge of lab equipment, electrical engineering concepts and ability to help in finding solutions is invaluable. He is also simply a good coworker to talk to on a long day with a seemingly endless simulation running.

The other graduate students, both in my group and others have also had a tremendous impact on my research, and their work deserves acknowledgement. I would like to thank James Ayers, Husni Habal, Sirisha Adluri, Brian Owens, Chengang Xu, Chee Boon (Steven) Law, Patrick Birrer, Matt MacClary, SasiKumar Arunachalam and Robert Shreeve, who's work my research directly depended upon. Also thanks to Zhimin Li, Triet Le, Chris Hanken, Jim Le, and everyone else who's work in substrate coupling I used.

In addition to other graduate students are the office staff that make all the research possible. I would in particular like to thank Clara Knutson for handling

all the purchase orders, Morgan Garrison for parts purchasing, and Brian Lindsey for shipping and other various needs.

I would also like to thank all my friends and family who have been supportive and understanding of my endeavors, especially those in the SCUBA community such as Don Heer and Ron Vearrier who have expected my presence only to find I had a deadline to meet.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

Figure Page

LIST OF TABLES

LIST OF APPENDIX FIGURES

Figure Page

LIST OF APPENDIX FIGURES (Continued)

Figure Page

A Methodology for Efficient Substrate Noise Estimation from Large Scale Digital Circuits in Mixed Signal SoC's

1. INTRODUCTION

1.1. Motivation

With the increasing demand for low-power and highly integrated circuits, designs are becoming increasingly more complex. From cell phones, wireless accessories, MP3 players and laptops in the consumer market to high-end military products such as missile guidance systems all require high levels of integration and miniaturization. System on chip (SoC) technology allows for continued improvements in power consumption and size, as all the components are included onto a single die or substrate. This reduces the total number of chip packages, which constitute a large fraction of PCB area. Examples of functions that may be integrated onto a single die include radio frequency transievers for wireless communications, analog audio amplification or data conversion, and digital control and processing operations.

As multiple designs are included on a single substrate, or even with one single large design, there are many challenges in verifying that the final product will behave within the performance specifications. In addition to performance checks such as timing, designs must pass a noise verification test to confirm that unwanted noise will not adversely interfere with system performance. As digital logic circuitry dominates in present day SOC designs, the amount of noise that the digital circuitry injects into the substrate is an important consideration. The

injected noise can degrade the performance of analog or RF blocks. Therefore, a designer must be able to accurately and quickly predict the amount of noise coupling in large scale designs [1].

1.2. Thesis Description and Outline

The simulation requirements for substrate coupled noise from large digital designs has consistently required a large amount of simulation time. The objective is to create a methodology that breaks a large simulation into more manageable parts which is then replaced with faster higher-level simulations. Although a loss of accuracy may be incurred because of the approximations made to accommodate multiple simulation levels, this can be offset by the large potential for a reduction of the simulation time and computational requirements.

This thesis discusses the development, implementation and integration of a transition-based simulation methodology into a comprehensive framework for substrate coupling analysis [2, 3]. Also discussed are complex designs used to verify the flow and the improvements in simulation time. Chapter 2 introduces the current methodologies, and the significant steps necessary to allow simulation of large scale digital circuits at a high level. Chapter 3 presents designs used with the methodology and the measurement setup for verification. Chapter 4 presents the simulation methodology and specific models that are used. Chapter 5 presents the simulation and measurement results, including additional simulations and measurements where necessary. Lastly, final conclusions and other areas of interest for future research are discussed in Chapter 6.

2. SUBSTRATE SIMULATION METHODOLOGY DEVELOPMENT

2.1. Substrate Coupling Network

For the purposes of this work, substrate coupling refers to any currents in the substrate that are coupled from one device to another. The generation of these currents is assumed to be included in low-level transistor models. These substrate currents may originate from impact ionization, capacitively coupled devices, or circuit noise. Impact ionization [4] noise becomes important as devices are scaled down and electric field intensities in the bulk of a transistor become high. Capacitively coupled noise is generated by parasitic capacitances to the substrate. Linear capacitances from components such as interconnects and poly resistors, and non-linear capacitances are due to reverse-biased p-n junctions occurring in MOS transistors and N-well resistors. Directly coupled [5] substrate noise occurs when a node is directly connected to the substrate, such as in guard rings, die-perimeter rings, and substrate taps used for bulk biasing.

For each noise source, a substrate port must be defined through which current is injected. This area is typically taken to be the area directly below an interconnect or the outline of a diffusion region. Assuming the frequencies of interest are below a few gigahertz, the substrate can then be modeled as a resistive network between each of the ports. For example, Figure 2.1 shows how a resistive substrate network is represented for two directly coupled substrate contacts as a pi-network.

This model can be easily extended for capacitively coupled contacts. Assuming a contact of the same size and location, the resistive network would be

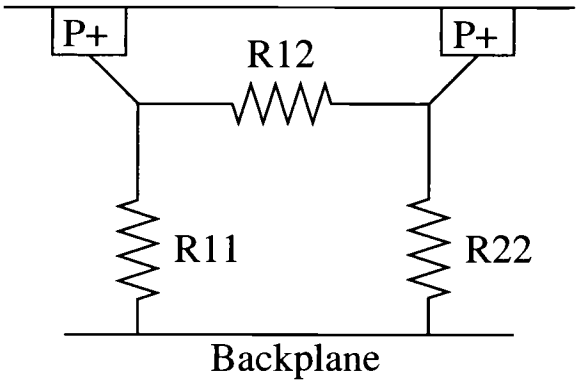


FIGURE 2.1. P+ contact to P+ contact resistive pi network.

equivalent. As shown in Figure 2.2, an added capacitor between the node and the substrate network represents a junction capacitance from a diffusion region.

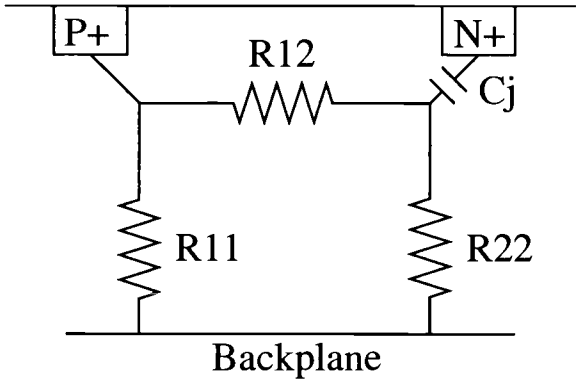


FIGURE 2.2. P+ contact to N+ contact resistive pi network.

To build the final substrate coupling network, all substrate contacts are analyzed by an extraction tool using either a boundary-element solver such as EPIC [6], or a macromodel [7]. Figure 2.3 shows how a three-port substrate would be modeled as a resistor network.

This extraction can be done for most kinds of substrates. The two commonly used substrates are termed heavily doped substrates and lightly doped substrates. Heavily doped substrates typically have low bulk resistance and are

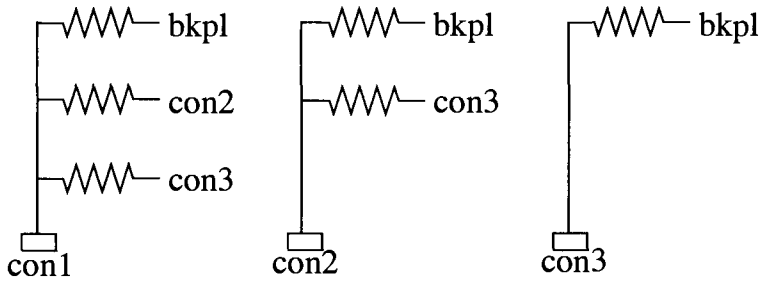


FIGURE 2.3. Network of multiple contacts.

used in large digital circuits to avoid latch up problems. Lightly doped substrates are commonly used in RF designs to reduce cost and increase the isolation ability where latch up is not as large a concern. However, for heavily doped epitaxial processes with low-ohmic substrates, it has been shown that the cross-resistances of the substrate are significantly larger than the vertical resistances [8–10]. This can greatly simplify the creation of the substrate network.

For digital circuits, the definition of substrate ports is difficult since multiple transistors share a single substrate region. If a transistor contains its own substrate diffusion region, the substrate port is the same as the diffusion region outline, as shown in Figure 2.4. For larger digital designs, the substrate diffusion regions of multiple transistors are combined to save area. By assuming that the substrate bulks within a cell are ideally connected, an approximate substrate port can be created. This is shown in Figure 2.5 where region 1 has 6 transistors in one diffusion region, region 2 has a single transistor, and region 3 has 5. Also included are the equivalent bulk and N-well ports.

This approximation is made using a far-field assumptions that a distant contact cannot resolve the differences between three small contacts and one of equal size. Figure 2.6 shows a representation of how three contacts and a bulk connection are combined into two contacts. For a realistic example of three 2.1

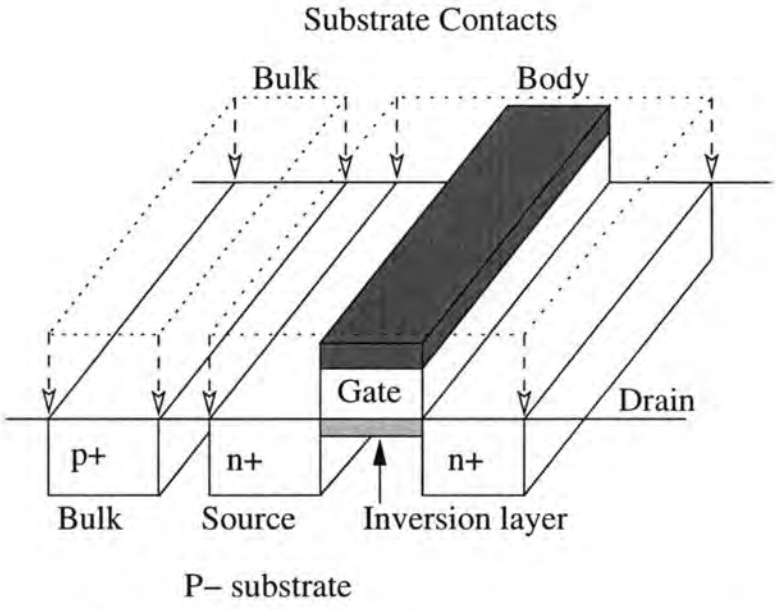


FIGURE 2.4. Layout of a single transistor and substrate ports.

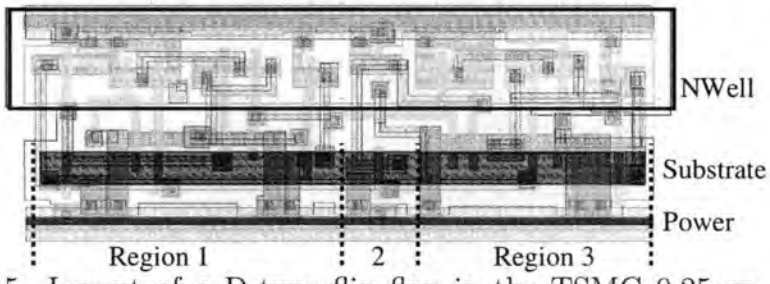


FIGURE 2.5. Layout of a D-type flip flop in the TSMC 0.25μm process with substrate ports indicated.

x 0.9 μm contacts separated by 0.3μm combined into one 6.3 x 0.9μm contact, the error in the resistive network is at most 15 percent, with critical values in the 5 percent range. Also because a heavily doped substrate is assumed, the cross resistances are typically larger than the resistances to the backplane.

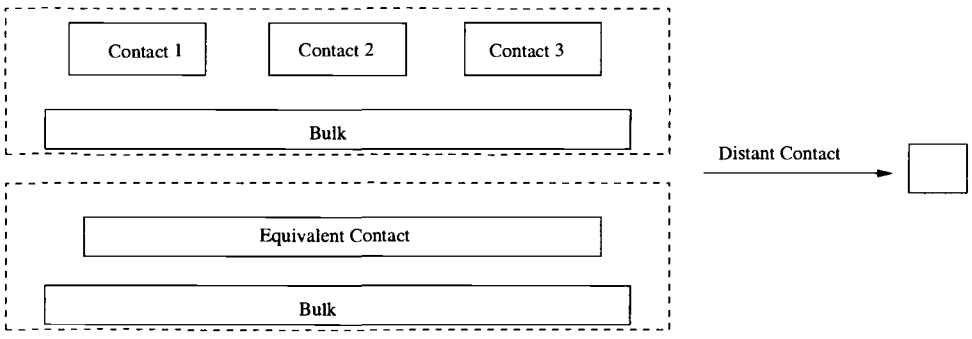


FIGURE 2.6. Far-field approximation for three substrate contacts.

2.2. Simple Substrate Simulation Model

One of the issues with the accurate estimation of noise injection from large scale digital circuits is the time required to simulate a design at the transistor level along with the substrate network and any noise sensitive blocks. When substrate coupling was first identified as an important consideration in SoC design, the entire design was extracted with all parasitics and then simulated. Figure 2.7 shows the flow for simulating a mixed signal design in this manner.

While accurate, this methodology cannot be realistically used for very large designs, due to the computational requirements. This is particularly true for digital circuits with more than a few thousand transistors. Thus, a segmented approach needs to be developed that reduces the total simulation time. The first approach is to simply remove the substrate coupling network from the simulation and save the noise injected at each substrate port. These ports typically only include transistor diffusion regions, although interconnects can be modeled as well. A second simulation is then performed with the substrate network and the noise sensitive blocks. Figure 2.8 shows how the simulation flow is partitioned.

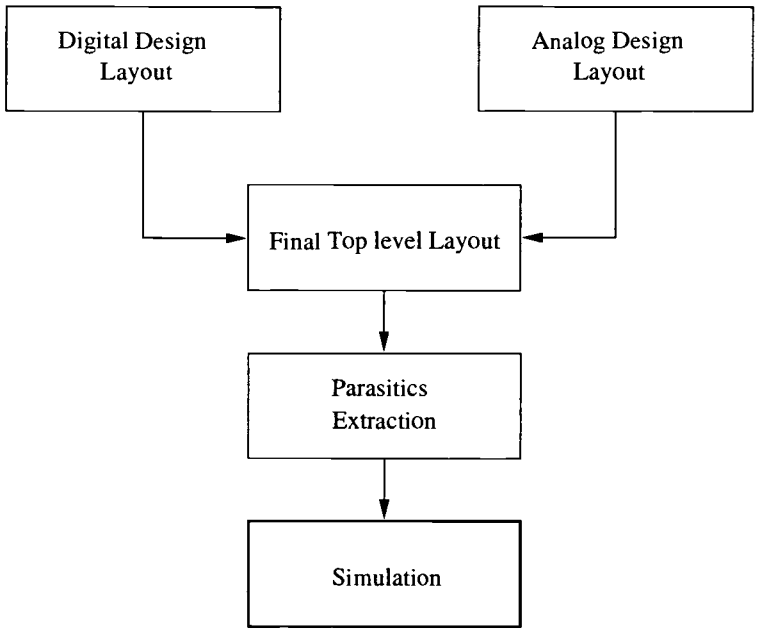


FIGURE 2.7. Single step simulation flow.

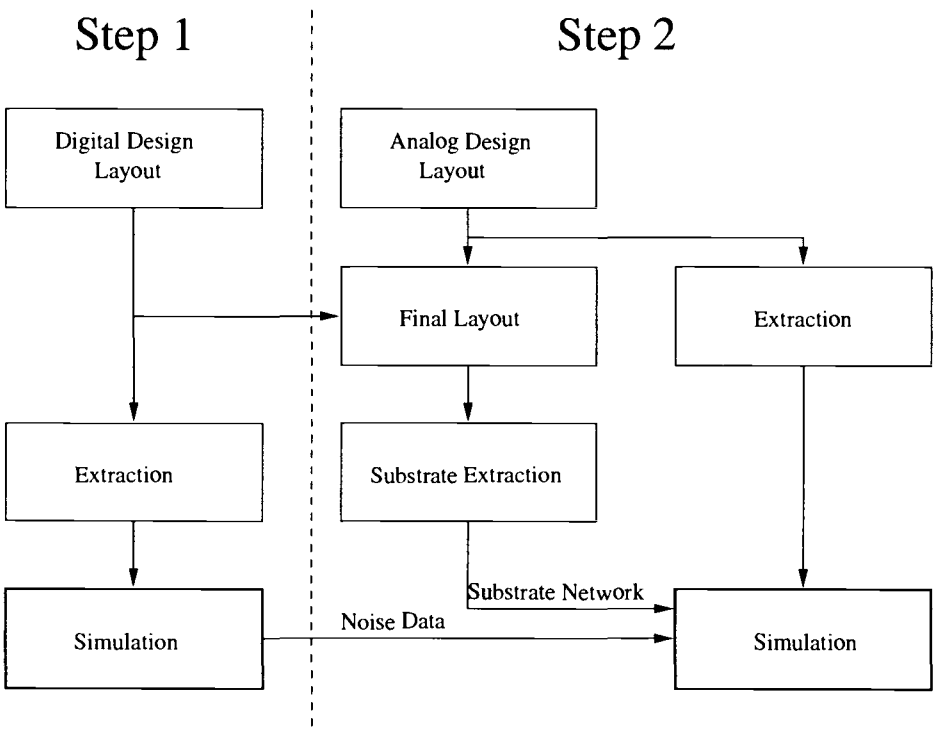
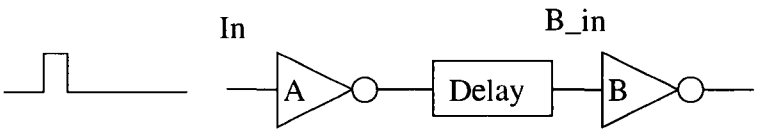


FIGURE 2.8. Two step substrate simulation flow.

However, the total computational resources required are still roughly equivalent, as every part of the entire design needs to be simulated at the transistor level. The only advantage of a two step methodology is that the noise injection from the large digital block is saved and does not need to be re-simulated if the analog block or substrate network changes. In order to reduce the total simulation time, the noise injection from the digital circuit needs to be computed more efficiently.

2.3. Improved Simulation Methodology

The first change to the two-step simulation methodology is to use a high-level, behavioral simulation that models the functionality of the entire design at a block level and therefore runs quickly. From this simulation, transition stimuli for each block are recorded for later use. Figure 2.9 shows how this is done with a pair of standard inverter blocks, labeled A and B. For each block, a pair of transition vectors are created, one for each possible transition. A VHDL simulation is used, although any other high-level simulation language could be used.



In	00000011111000000000000000000000
B_in	11111111100000111111111111111111

	Transition Vectors
A{0→1}	00000010000000000000000000000000
A{1→0}	00000000000100000000000000000000
B{0→1}	00000000000000100000000000000000
B{1→0}	00000000010000000000000000000000

FIGURE 2.9. Creation of transition vectors for two inverters.

Assuming that noise injection is dependent on how each block's inputs transition, a low-level transistor simulation of each block is performed where neither the substrate network nor the package parasitics are included. The currents injected into the substrate and power rails are simulated as shown in Figure 2.10 and saved for later use. For every possible input stimulus, the bulk currents are saved into a library of noise signatures for the cell. Because the cells are standardized and are often used in multiple designs, the noise signatures need only be generated once, assuming they do not include the effects of any package parasitics. The inclusion of package parasitics is taken into account only in the final simulation, thus current variations due to power rail fluctuations are not accounted for in earlier stages.

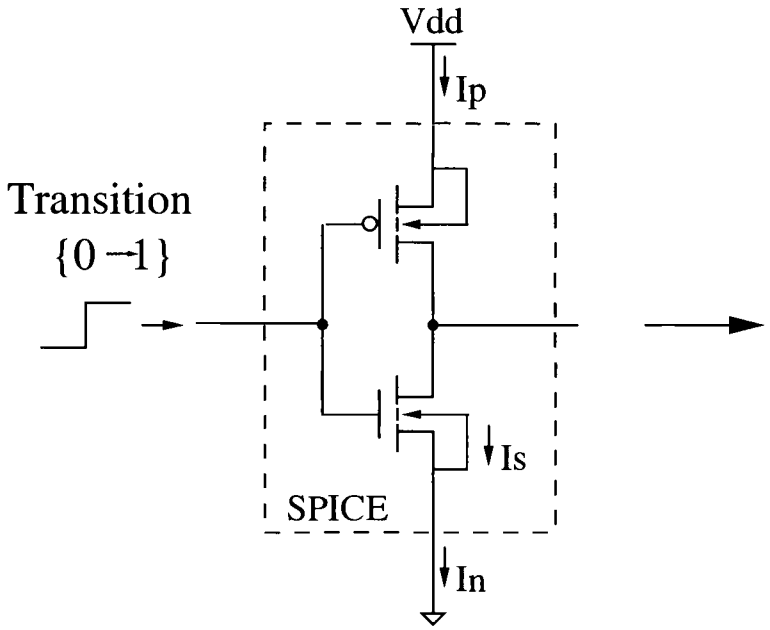


FIGURE 2.10. Substrate and power rail current simulation in SPICE for an inverter.

The noise injection waveforms are constructed by convolving the transition vectors with the proper noise signature in the library for each cell. Figure 2.11

shows the complete approach for constructing the noise current waveforms from the previous transition vectors for cells A and B.

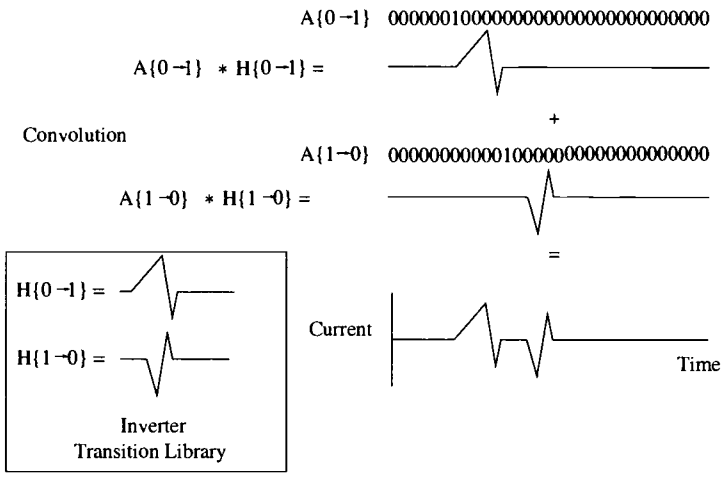


FIGURE 2.11. Convolution and addition of transition vectors.

From the layout information, a resistive substrate network is extracted and then added to the final circuit simulation as shown in Figure 2.12. The circuit netlist can include any other sensitive analog components that require SPICE-level simulation. The computational resources that the final netlist requires is largely dependent upon the complexity of the bondwire models and analog circuits, since the digital circuitry and substrate coupling network are modeled as current sources and resistors.

It should be noted here, however, that this approach only models feed-forward noise estimation. Because of the complete separation of the digital design from other circuit blocks, it may be difficult to ascertain any feedback that may occur from the noise adversely affecting an analog block that changes the input to the digital system. For example, while noise injection into a PLL can be estimated, the effect this jitter causes in the digital circuit will not be estimated as there is no feedback path from the analog domain to the digital. This is usually not an issue, however, as the digital circuitry is designed to tolerate some amount of noise.

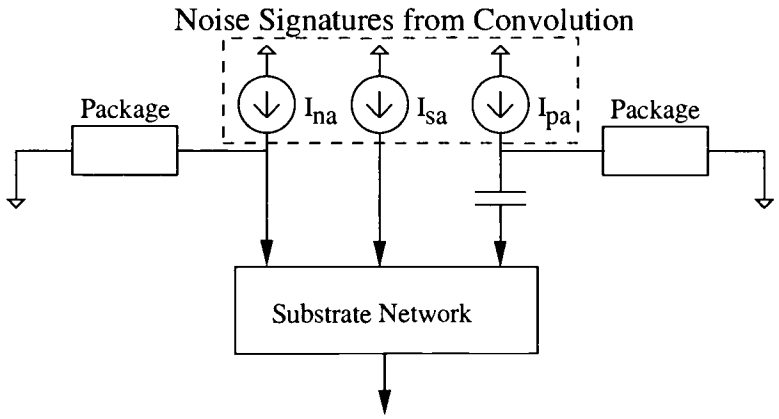


FIGURE 2.12. Top level SPICE simulation setup.

2.4. Summary

The original simulation methodologies for predicting substrate digital noise injection had one or two distinct simulation steps. By essentially splitting the first step in the two-step method into two easily executed stages, a three-step simulation methodology has been created that can be integrated into most standard digital design flows. Figure 2.13 shows a basic digital flow for which this might be used. The core of the flow is the synthesis step, which is the translation of behavioral, register transfer level (RTL) code that the designer writes into a structured gate-level netlist. An analogy to this would be compiling high level C++ code into assembly code. This step also provides estimates as to how fast the design may run. Subsequently, the netlist is implemented into a layout that can be fabricated, and more accurate timing analysis and verification steps are completed. This is analogous to an assembler producing binary machine code that can be executed. By using data from these intermediate design stages, a comprehensive flow for substrate noise coupling can be generated, as shown in Figure 2.14.

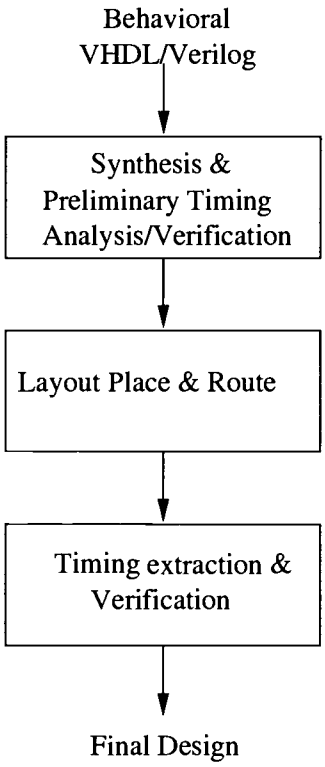


FIGURE 2.13. Standard digital design flow.

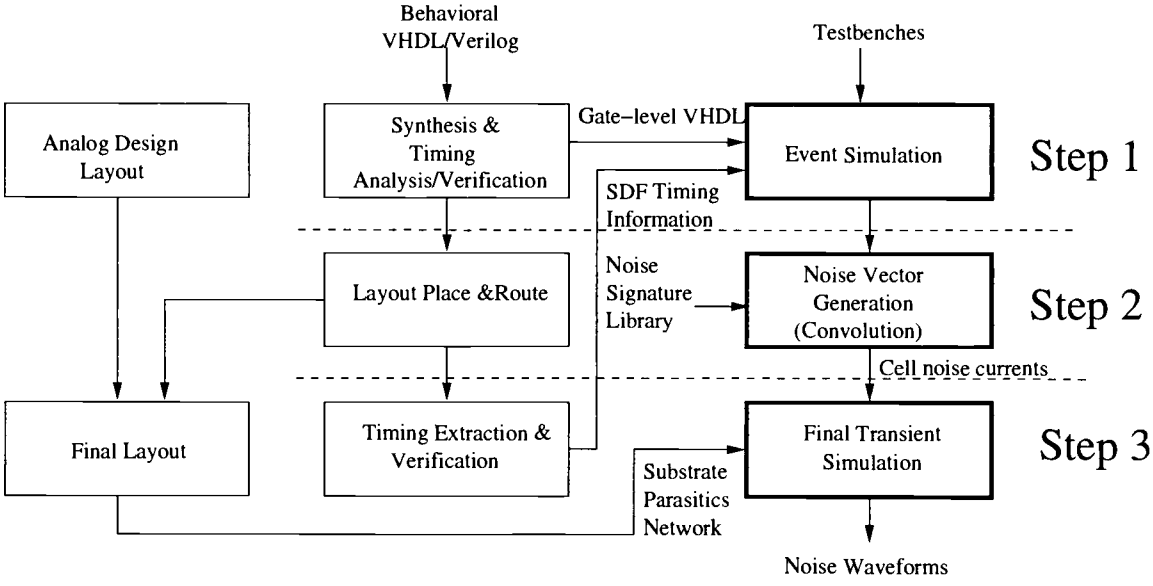


FIGURE 2.14. Three step substrate noise coupling simulation flow.

Here, the gate-level VHDL description is combined with accurate timing information to create the event simulation. A program combines the transition data with the noise signature library by convolving and adding them together to generate the cell noise currents. These are used in the final simulation to determine the effects of substrate coupled noise on an analog circuit.

3. DSP CORE DESIGN AND LAYOUT

In order to fully verify the simulation methodology, a test chip in the TSMC 0.25μm logic process was fabricated. This test chip (Eris) has two 16-bit digital signal processors (DSPs) and substrate noise sensing blocks for verification.

3.1. DSP Core Description and Uses

For the purposes of validation, a generic 16-bit DSP processor is used. The DSP is intended to emulate a large digital design that might be on the same die as some analog circuitry. A basic 16-bit 5-stage pipelined processor was used [11] with a 16-bit multiply-accumulate (MAC) block and a 64-bit accumulator. Figure 3.1 shows the basic architecture of the processor excluding the control logic.

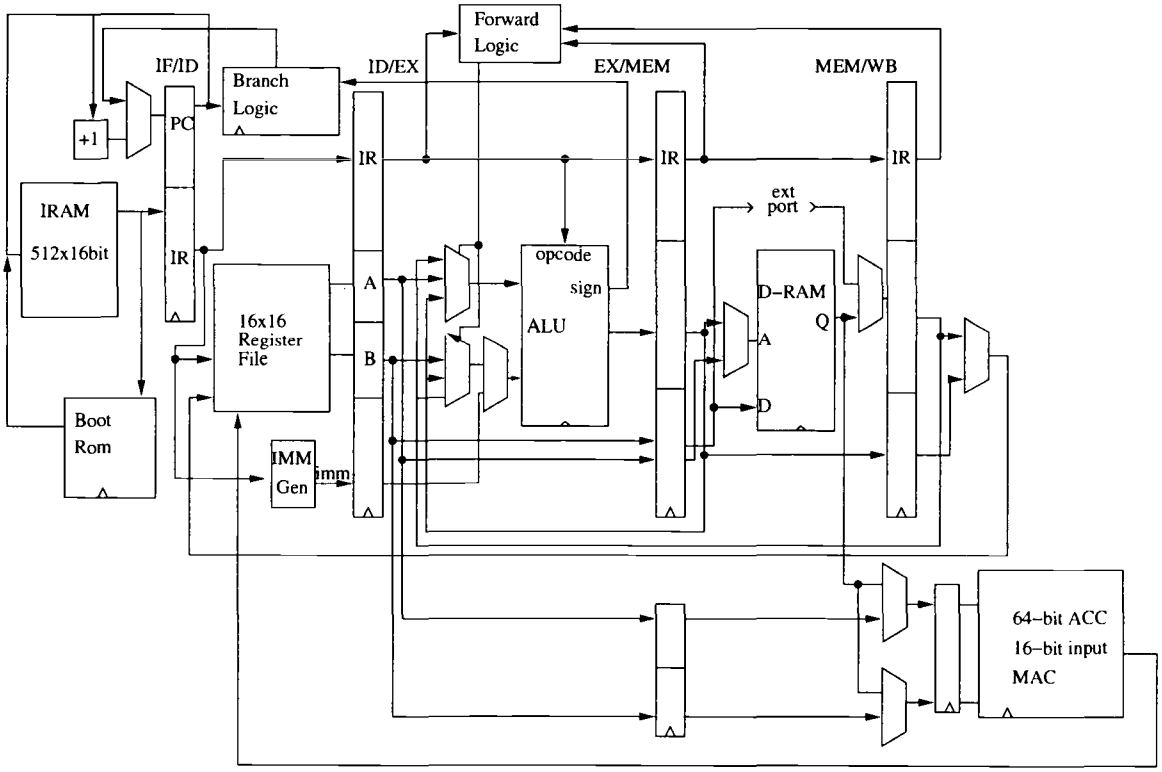


FIGURE 3.1. Top level DSP core architecture.

The DSP core has been designed to implement a FIR filter. Programs demonstrating up to 384 taps have been tested that complete nearly 16 MAC instructions for every 19 clock cycles using counter loops. It would be possible to implement 512 taps at the cost of efficiency. The DSP can also be used as a generic processor for processing data, although it only has one external 16-bit data port which limits its ability to access external memory. For testing purposes, a variety of instructions are used, including load/store, ALU and MAC functions.

The first DSP processor was compiled and synthesized using the flow described in Chapter 2 and the Artisan SAGE-X standard cell library for the TSMC 0.25 μm logic process (CL025E). While this process has 5 metal layers, the top metal layer was not used and made available for later routing. The layout, not including the dual-port SRAMs, was 0.5mm x 1.0mm, of which roughly half the area was from the MAC functional block. A total of 9100 standard-cell instances are used, including 1400 generic D-type flip-flops. After the design was imported into Cadence Design Framework II, the SRAMs were added along with 25-μm wide power rings for each section of the core. The power rings were made on metals 3 and 4 on top of each other in order to maximize the capacitance between the rails for decoupling. Metal 2 was reserved for input and output lines crossing the power rails, and metal 1 was used for a guard ring.

The second DSP core was designed to have a different noise signature. In order to do this, the most active cells in the design were identified by the RMS noise current they produced. By examining the pre-noise simulation current vectors, it was found that 70 percent of the RMS noise was generated by the D-type flip flops.

To change the noise signature of the DSPs, the 1400 instances of the flip-flops were changed to a modified version with the same functionality. Figure 3.2

shows an example of the type of flip-flop implemented in the Artisan standard cell library using transmission gates. This design allows for a compact implementation with good speed by using buffering on the gates. The modified design used another traditional approach, shown in Figure 3.3.

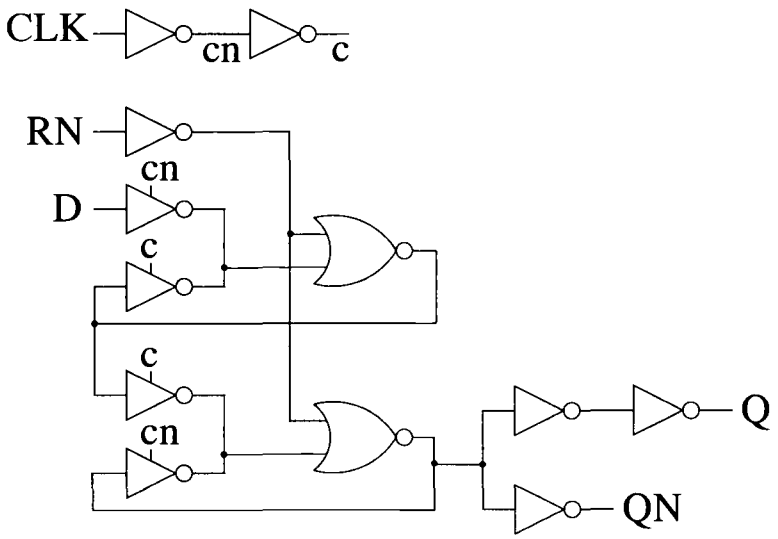


FIGURE 3.2. Transmission gate flip flop.

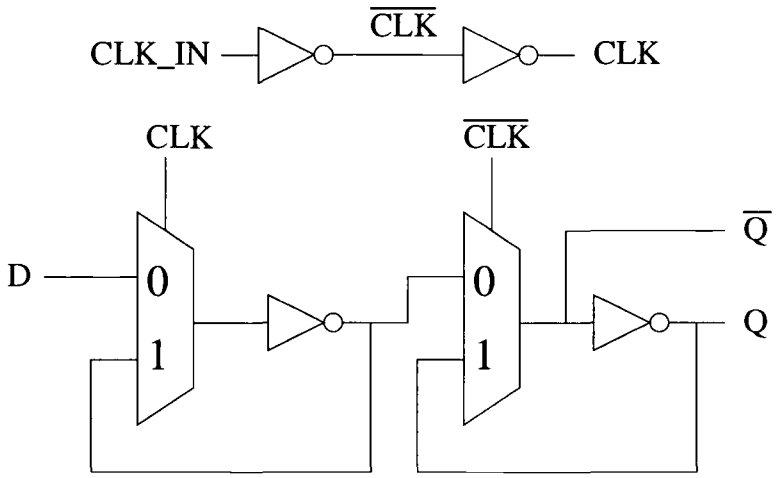


FIGURE 3.3. CMOS D flip flop.

After redoing RMS noise calculations, it was initially found that the total RMS noise of the second design was approximately 40 percent higher. This was

incorrect, however, as was discovered later. After fabrication it was found that the RMS noise had not taken into account the full data set. Because the noise waveform generation program reduces the size of the current waveforms, it saves the data in a sparse format and excludes all data points that are zero valued. Subsequently, when the RMS noise value was calculated, only the valued data points were used for averaging, whereas a true average would include the zero values. Taking sparsity into account, the RMS noise is only different by about 0.5 percent. If the clock rate is increased, however, the RMS noise difference may be increased because of the reduction in the sparsity of the waveform. These preliminary calculations also did not consider peak noise in any way.

Both DSP cores share an IO interface through a set of OR gates. Because the IO buffers can generate their own noise signatures, the DSP cores can operate without the buffers turned on after a program has been loaded. This allows for measurement of noise generated by only the dsp processor. Figure 3.4 shows the schematic for the IO buffers. The IO buffers used are the TSMC-based EZ-IO tpz873gez provided by Artisan Components.

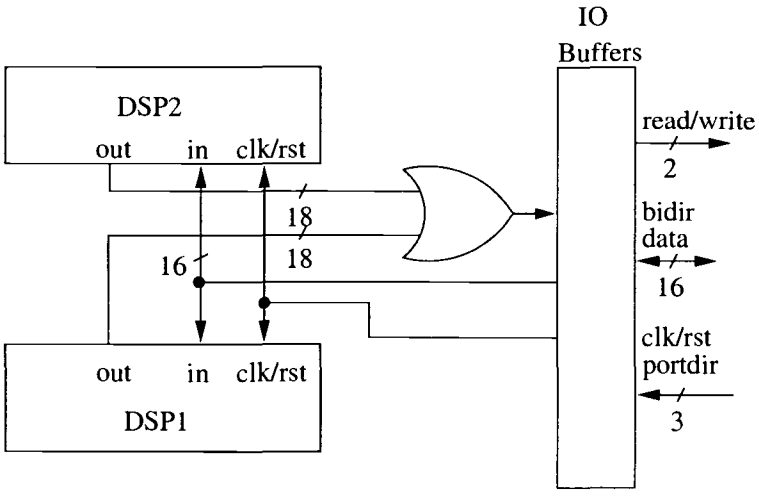


FIGURE 3.4. IO block integrating two DSPs into one 16-bit interface.

3.2. Die Photo

The die photo for the Eris test chip is shown in Figure 3.5. This chip is packaged in a 132-pin Pin Grid Array package (PGA132M). In addition to the sense circuitry, a number of other designs were also included on this test chip. A 50μm wide die perimeter ring (DPR) is used to connect to the substrate backplane. The processors are in the lower left of the chip with a sense amplifier located symmetrically in between them. Figure 3.6 shows a closeup of the first DSP.

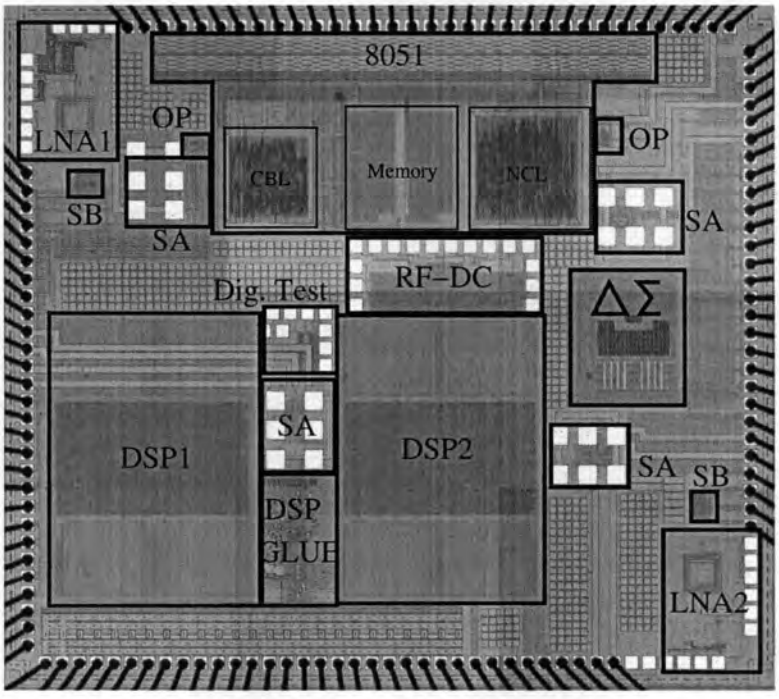


FIGURE 3.5. Die photo the ERIS test chip.

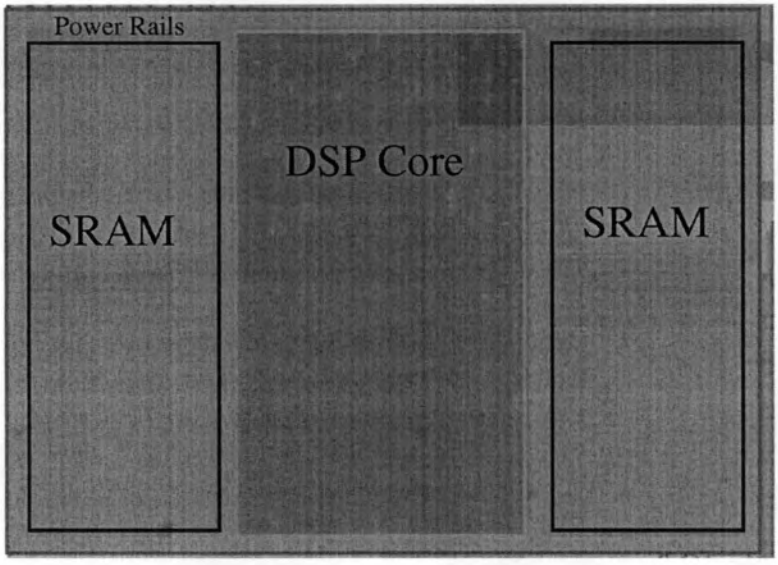


FIGURE 3.6. Die photo of the first DSP.

4. SIMULATION METHODOLOGY

The simulation setup used is shown in Figure 4.1. For each cell, the substrate noise waveforms are represented with current sources. The substrate is modeled with a resistive pi network that is calculated for each cell. A single resistor and capacitor combination represents the n-well connection to the backplane. Package parasitics are included using the package model for the PGA132M package.

The power rails for the digital circuits and the analog circuits are assumed to be entirely decoupled, which they are not in reality. The entire model used for simulation purposes with the Eris test chip is shown in Figure 4.2. The individual bondwire values for Cp1, Cp2, Lp1, Lp2, Rp1 and Rp2 are pin-dependent and are listed in the package datasheet. The major values, (Lp1, Rp1 and Cp1) are typically around 7nH, .1mΩ, and 6pF, respectively. The Eris test chip uses multiple bondwires for the power and ground networks, and these values are combined in the simulation.

By combining the given bond wire, package, and PCB models with the current source vectors generated with the Silencer! tool, the full simulation netlist is created. For the substrate parasitic network, assuming a heavily-doped substrate, the inter-cell cross-resistances are ignored. The N-well capacitance and contact resistance is calculated from the total N-well area, and then adjusted to account for the large quantity of capacitance caused by the sidewall area of a diffusion region. Because a standard-cell based design has large alternating N-well rows, there is significantly more sidewall area than an equivalent rectangle with the same area. Figure 4.3 shows the N-well instances for the processor, and how it has a large perimeter to area ratio. However, the TSMC process parameters do

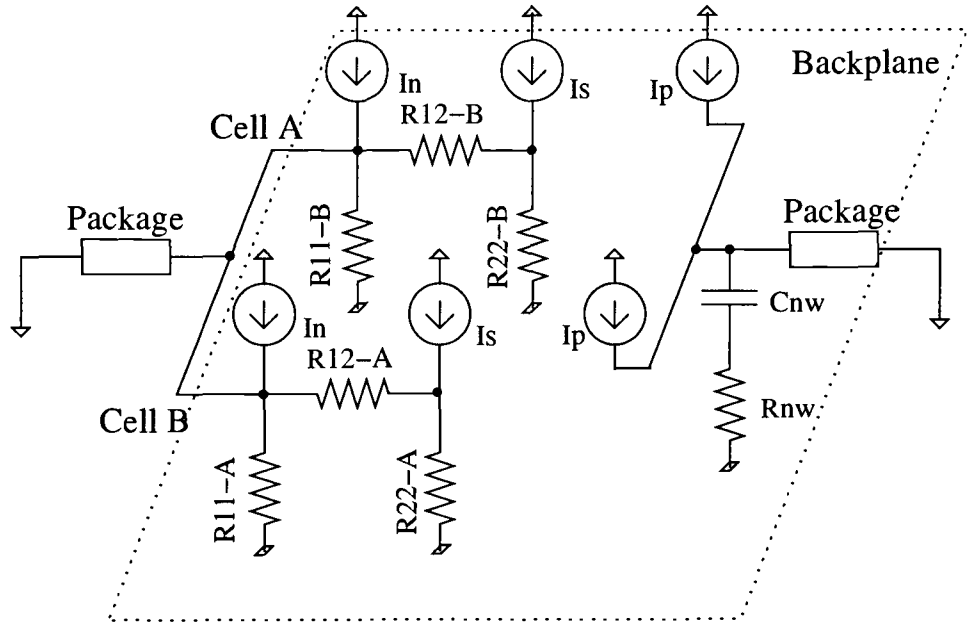


FIGURE 4.1. Noise simulation setup.

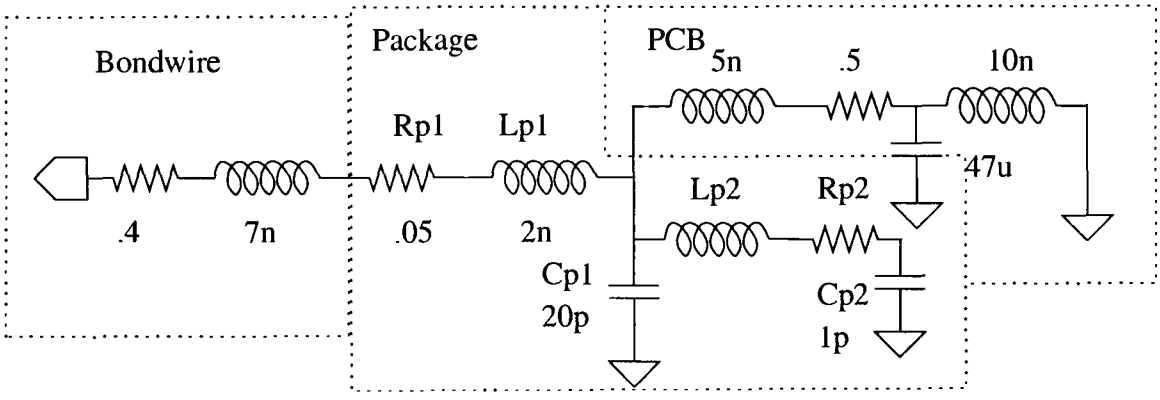


FIGURE 4.2. Full parasitics model accounting for multiple bondwires.

not provide a perimeter capacitance value. The true capacitance value will be considerably higher than the extracted value.

The die-perimeter ring and guard rings are also added in a similar manner. Figure 4.4 shows the die perimeter ring connection to the package pin model and the substrate network using an extracted resistance. The package model is shown as part of Figure 4.2 with values for multiple bondwires adjusted accordingly.

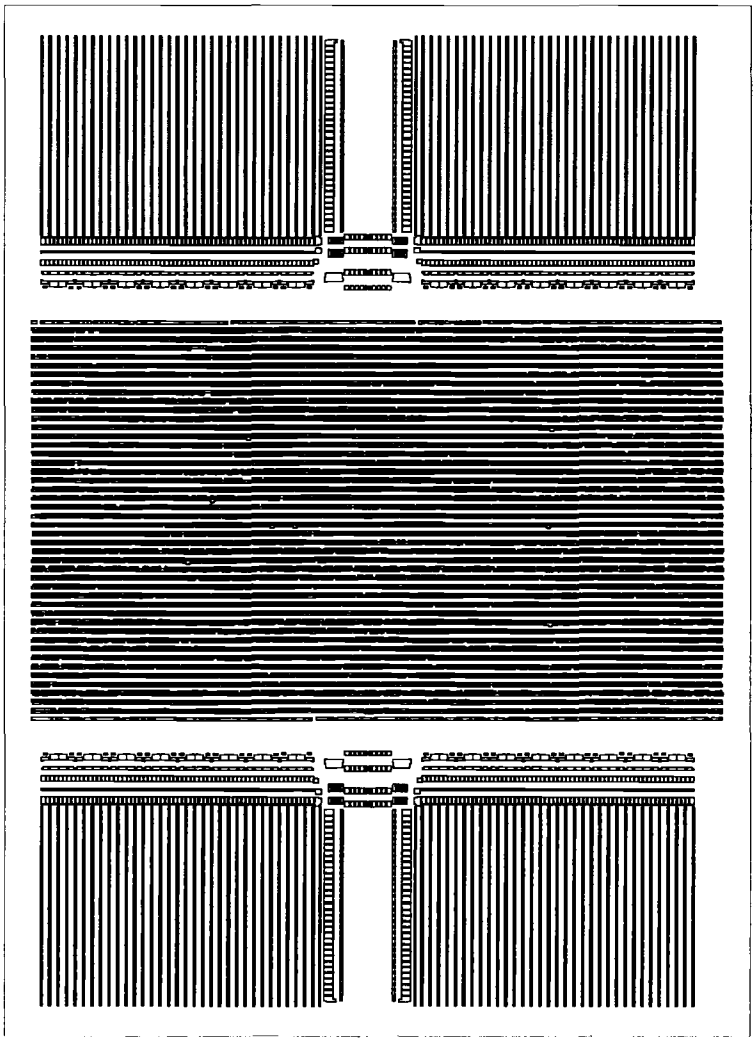


FIGURE 4.3. Top level DSP core layout showing only N-well instances.

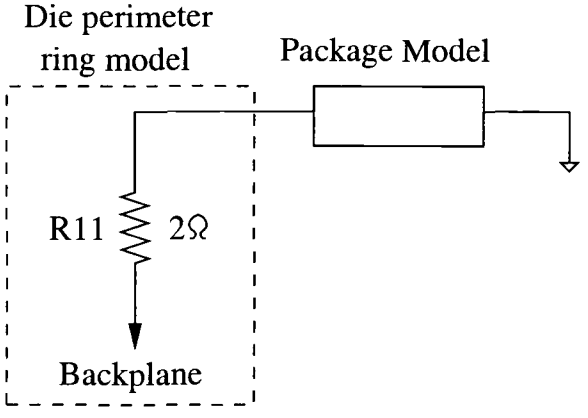


FIGURE 4.4. Die perimeter ring connection to ground.

5. SIMULATION AND MEASUREMENT RESULTS

5.1. Simulation Execution Time

Table 5.1 shows the execution times for a full SPICE simulation and the methodology described in this work. The times are for simulating 17 clock cycles of the same DSP design. The last three rows in the table present the execution time for each of the three main subcomponents of the improved methodology. As can be seen, there are significant improvements. Because the DSP core is simulated without any package information in this approach, it needs only be simulated once. Each modification to the package parasitics (grounding a die-perimeter ring, for example), requires only a short re-simulation of the simplified top-level SPICE netlist. In addition, only one or two clock cycles may be required for simulation, assuming the noise from a previous clock cycle does not interfere with the next noise spike. This methodology also provides a significant savings in memory, as shown in Table 5.1. Spectre was used as the SPICE-equivalent simulator, and the times do not include the extraction of the netlist from the layout in the Cadence Design Framework II.

TABLE 5.1. Execution time for presented methodologies.

	Execution Time
One Step Methodology	204 hours
Three Step Methodology	2.5 hours
Gate level simulation	2 hours
Vector generation	5 minutes
Top level Spice	25 minutes

TABLE 5.2. Memory requirements

	Memory
One Step Methodology	430 Mb
Three Step Methodology	46 Mb maximum

A comparison of the output of the two methodologies is shown in Figure 5.1 for a single flip flop design with the same package parasitics. Here, the peak to peak voltages are within 5 percent, and both demonstrate similar noise responses.

5.2. Simulation and Measurement Results

All measurements were taken using a Tektronix TDS7404 4GHz 20GS/s Oscilloscope with a pair of Cascade Microtech SG150 microprobes. A Tektronix TLA720 logic analyzer/pattern generator was used to control the DSP cores. Figures 5.2 (a) and (b) present the simulation and measurement results from the first DSP core with the DPR floating for one rising edge and one falling edge of the clock.

The presence of a 1MHz square wave is seen in the measurements that is not in the simulation results. Upon further investigation, this was found to be related to differences in the ground plane voltages, caused by current being drawn by the DSP cores. Because the methodology is transition-based and does not include static power consumption, this shift cannot be directly modeled in the noise simulation. Instead, a voltage source models the shift in the power rail based on measured results.

Figure 5.3 shows a measurement with a differential probe of different points on the ground plane where the sense amplifier and DSP cores are connected. This

measurement has been filtered to exclude high frequency noise. As can be seen, a small shift in the ground plane is causing a difference between the sense amp ground and the DSP grounds. This is added back into the simulation model as shown in Figure 5.4.

Here, we can now see that the simulation results match the measurements quite closely. The peak-to-peak magnitudes for both the simulation and measurement are given in Table 5.2. The simulation results are shown to be within 10 percent of the measurement results. In addition, the ringing behavior of the noise is also roughly equivalent in both waveforms. The frequency spectra are shown in Figure 5.5. Here the two low frequency peaks are similar, however it can be seen that some high frequency tones in the simulation are still missing, and may be lost due to the package model.

TABLE 5.3. Peak-to-peak voltages for simulation and measurement.

	Rising Edge	Falling Edge
Simulation	29.5 mV	43 mV
Measurement	32 mV	42 mV

The simulation and measurement results presented also have 4 significant topics of interest. The first is the low frequency component found in all the waveforms. This was found to be caused by a combination of the large N-well capacitance in the VDD rail of the cores with the bondwire and bondfinger inductances. These two combined create a frequency response at around 60 MHz. The backplane resistance largely determines the damping rate of this frequency, as it is the connection between the package parasitics and the N-well capacitance. This is shown with data from the first DSP with the DPR floating in Figure 5.6.

The second interesting noise frequency is at around 210, 300 and 400 MHz, as shown in Figure 5.7. These frequencies are found to be due to the interaction between the PCB trace parasitics, the decoupling capacitor and the Cp1 capacitance to ground.

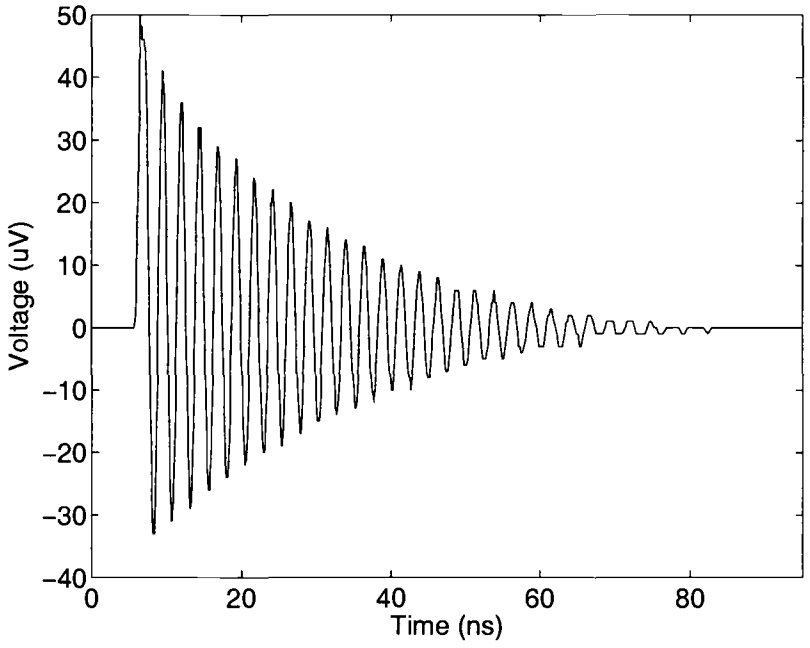
Third, the package parasitics are a significant contributor to the noise generation. To show this further, the DSPs were powered through 6 bondwires, 3 per power rail. By removing one or two of the bondwires in each, the effect of the packaging on substrate noise can be demonstrated. Figures 5.8 through 5.10 show the measurement results as these bondwires are removed. Figures 5.11 through 5.13 show the corresponding simulation data. These result assume a floating die perimeter ring.

Lastly, the addition of a die-perimeter ring can reduce noise for certain cases, but it can also be detrimental, as the additional resonant frequencies of the package parasitics can have adverse effects. This has been reported by other researchers previously in [12]. For these measurement results, we see a significant reduction in noise as shown in Figure 5.14.

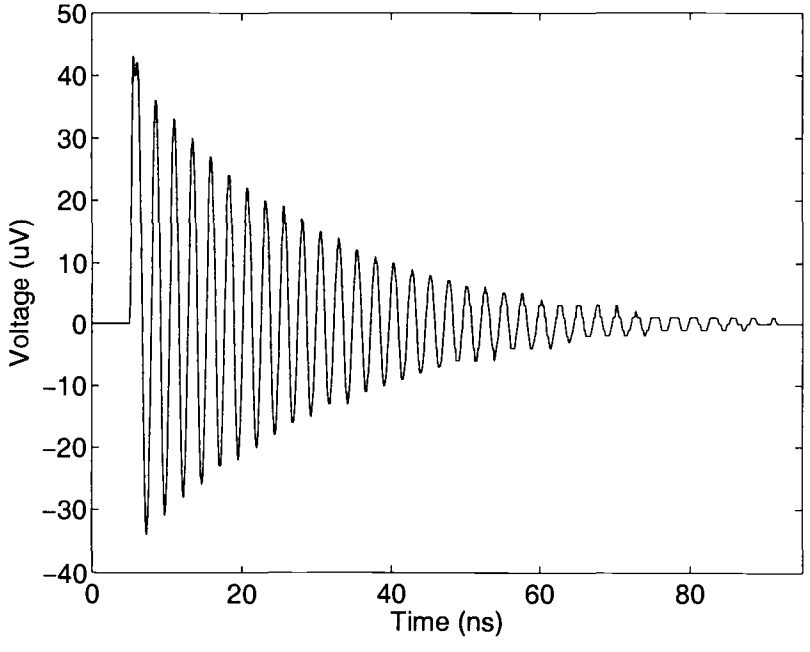
5.3. Summary

While all the simulation and measurement results do not show an exact match, they capture the essence of what is occurring, the mechanisms through which the noise arises, and the peak-to-peak voltages. In addition, we can also see the need for a good package, as all the major noise effects here are being caused by the bondwire and package parasitics. The approach is also shown to be a fast and effective aid that can be used to estimate noise early in the design

phase. Additional simulation and measurement results for various configurations can be found in Appendix B.

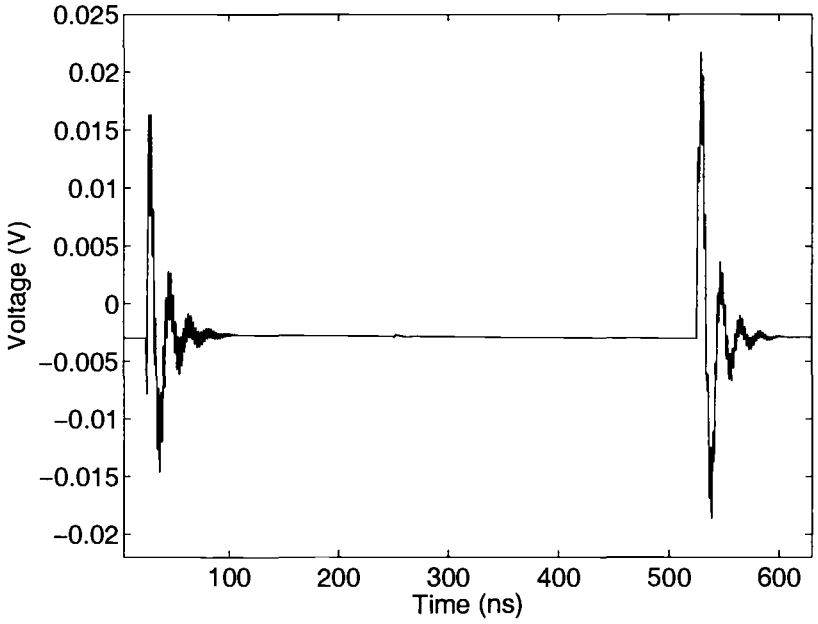


(a)

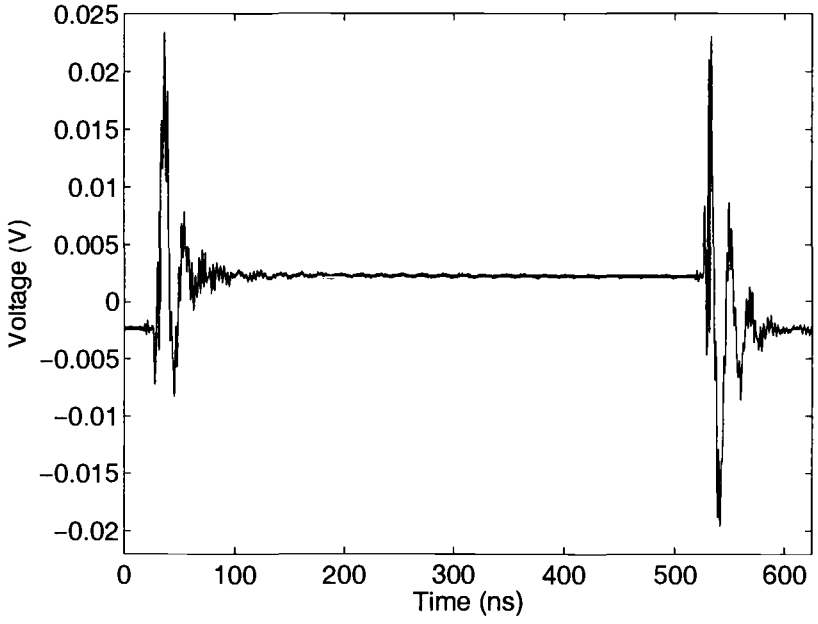


(b)

FIGURE 5.1. Simulation results using (a) the one step methodology and (b) the three step methodology for a single cell.



(a)



(b)

FIGURE 5.2. (a) Simulation and (b) measurement results for the first DSP core with the DPR floating

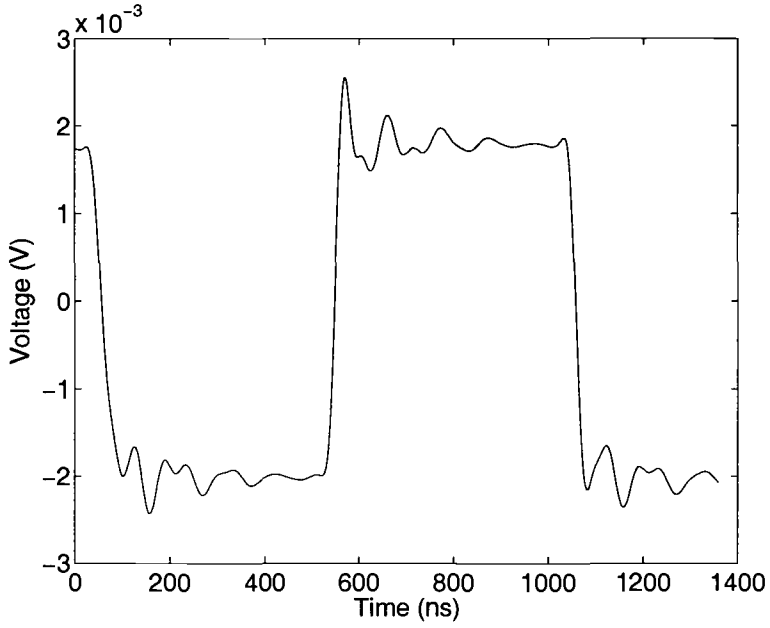
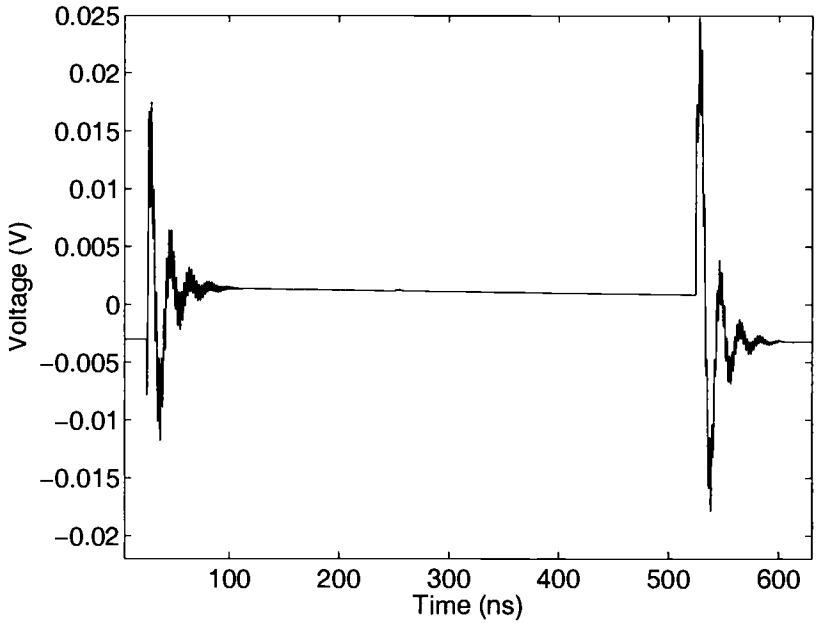
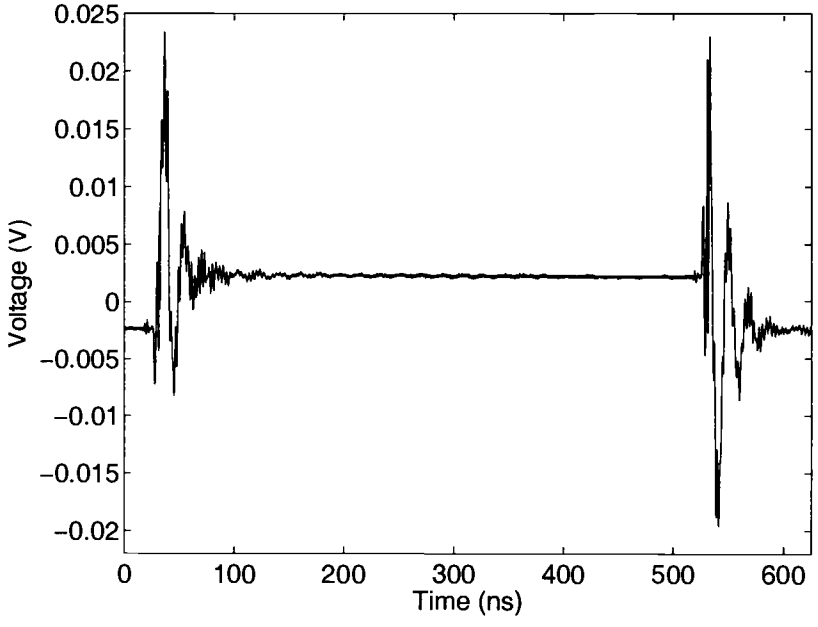


FIGURE 5.3. Filtered measurement of ground reference differences.

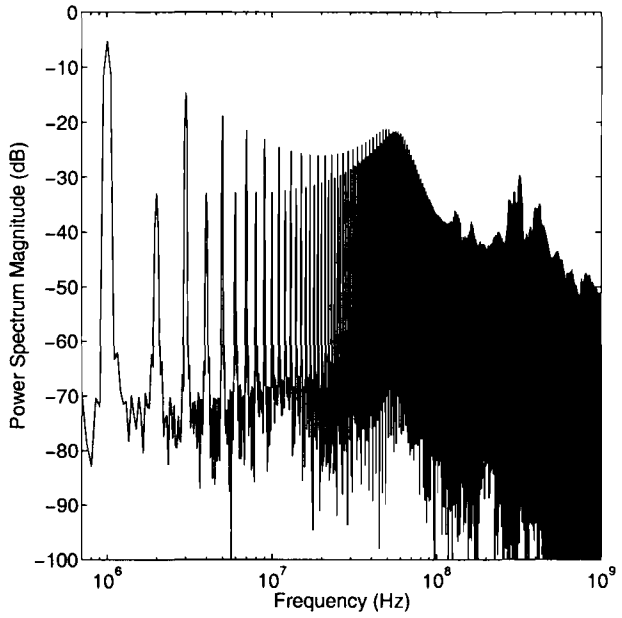


(a)

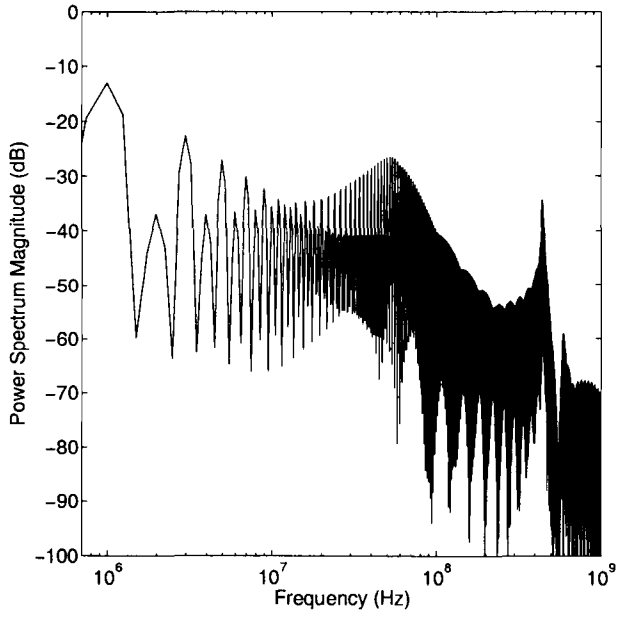


(b)

FIGURE 5.4. (a) Simulation with a 1MHz square wave added to model the ground reference differences and (b) measurement results for the first DSP core with DPR floating

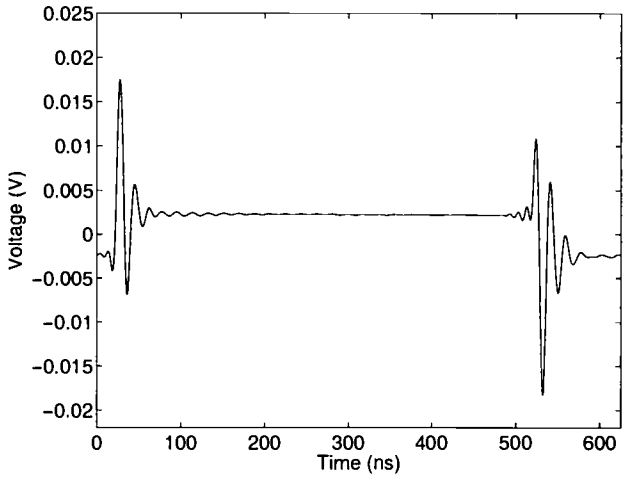


(a)

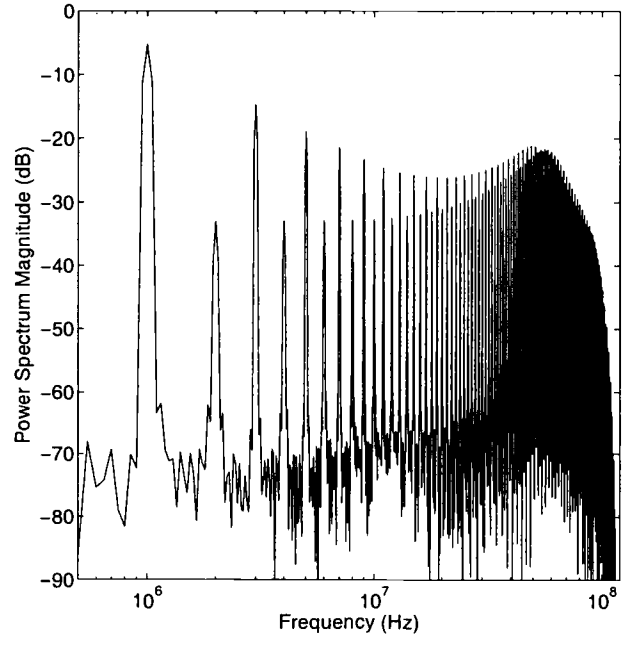


(b)

FIGURE 5.5. (a) Measured and (b) simulated power spectrum densities.

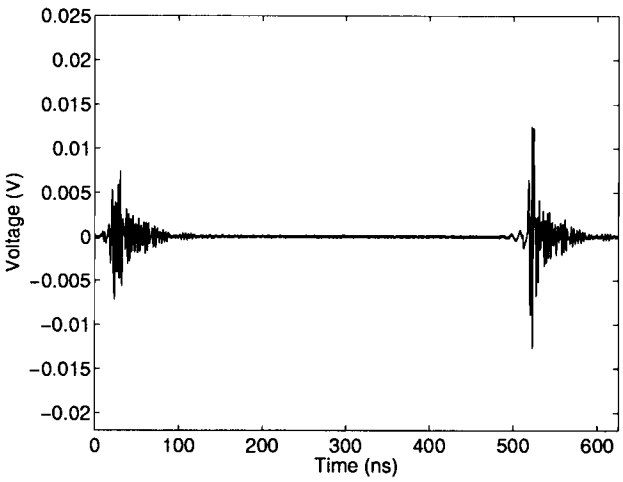


(a)

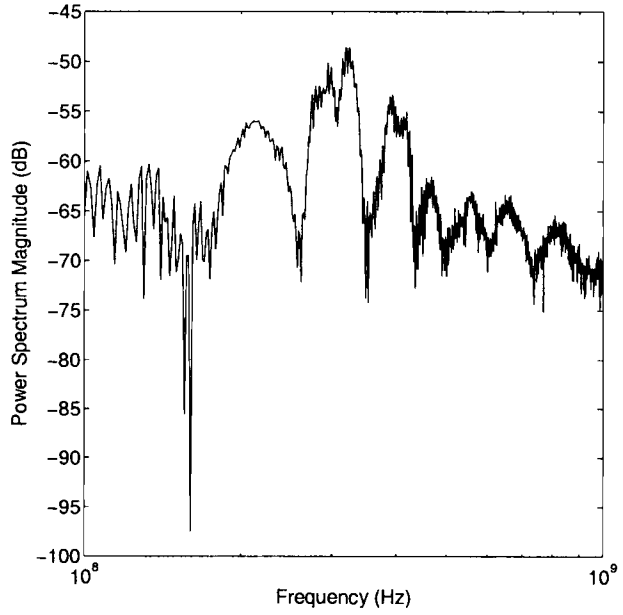


(b)

FIGURE 5.6. Filtered data (a) frequency < 100MHz and (b) its power spectrum density.



(a)



(b)

FIGURE 5.7. Filtered (a) frequency data > 100MHz and (b) its power spectrum density.

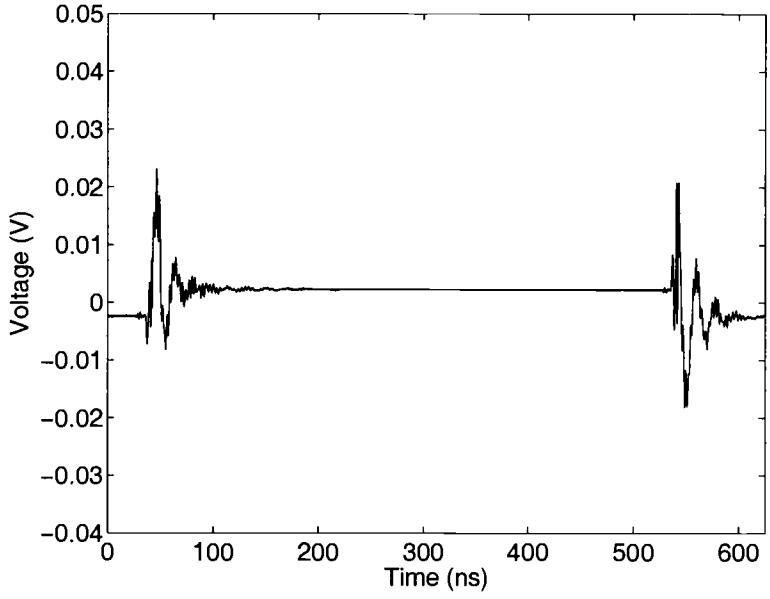


FIGURE 5.8. Measured DSP noise with 3 bondwires per power rail.

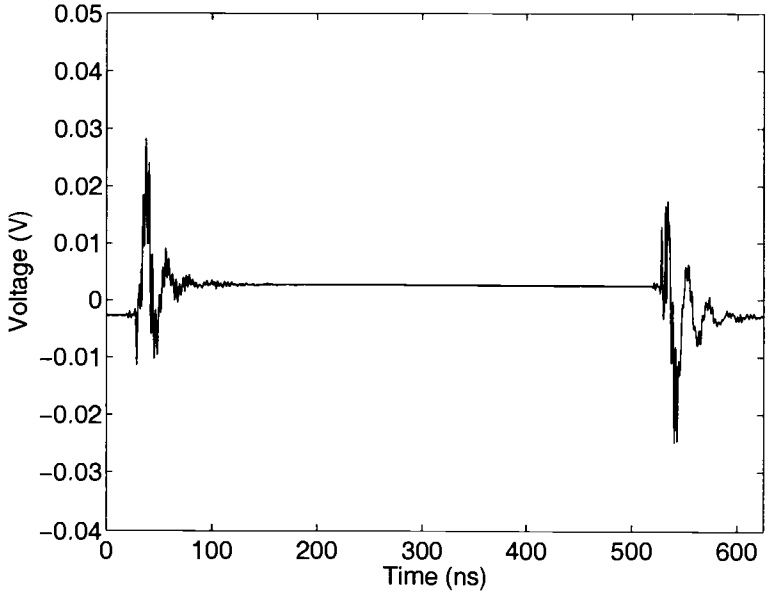


FIGURE 5.9. Measured DSP noise with 2 bondwires per power rail.

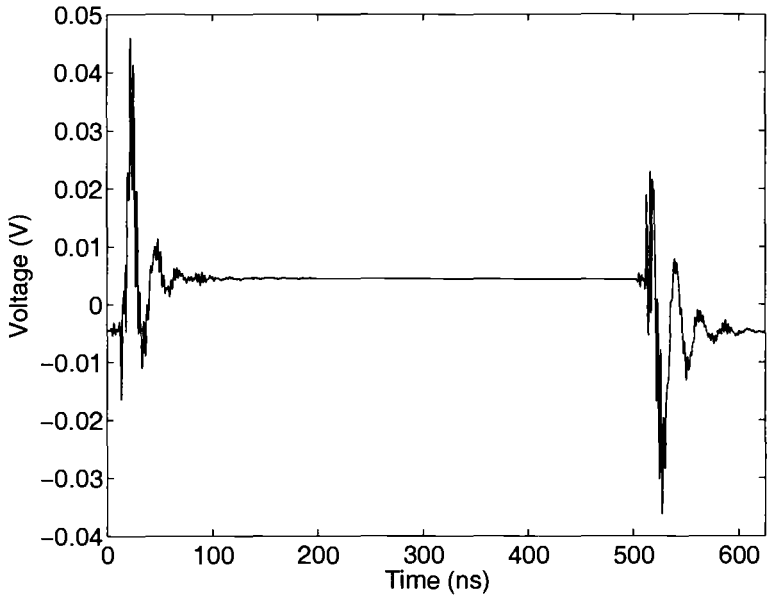


FIGURE 5.10. Measured DSP noise with 1 bondwire per power rail.

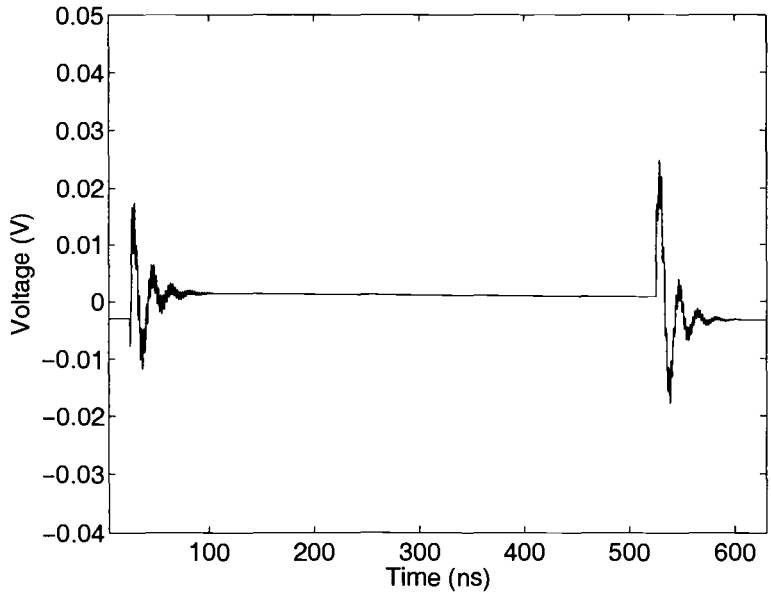


FIGURE 5.11. Simulated DSP noise with 3 bondwires per power rail.

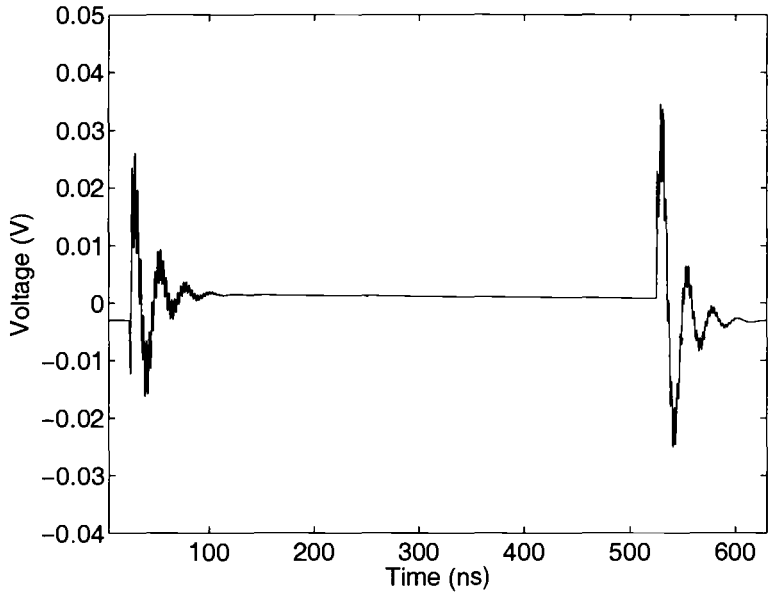


FIGURE 5.12. Simulated DSP noise with 2 bondwires per power rail.

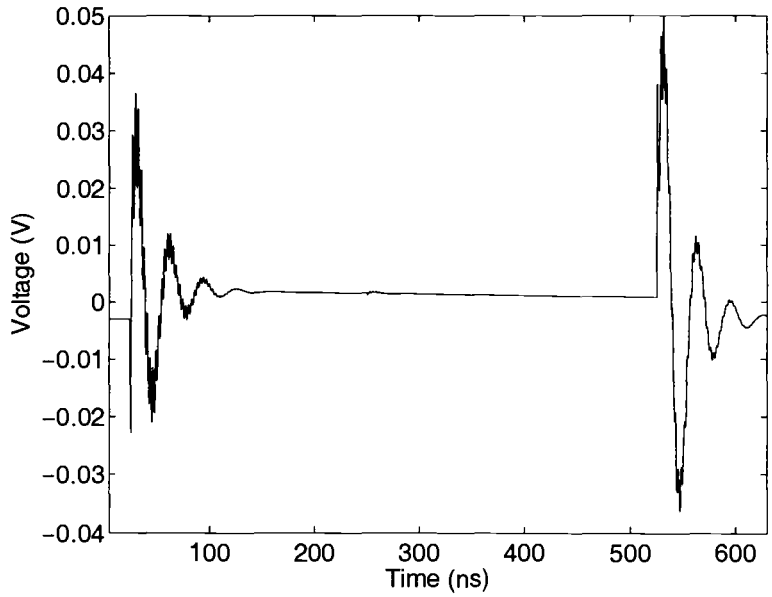
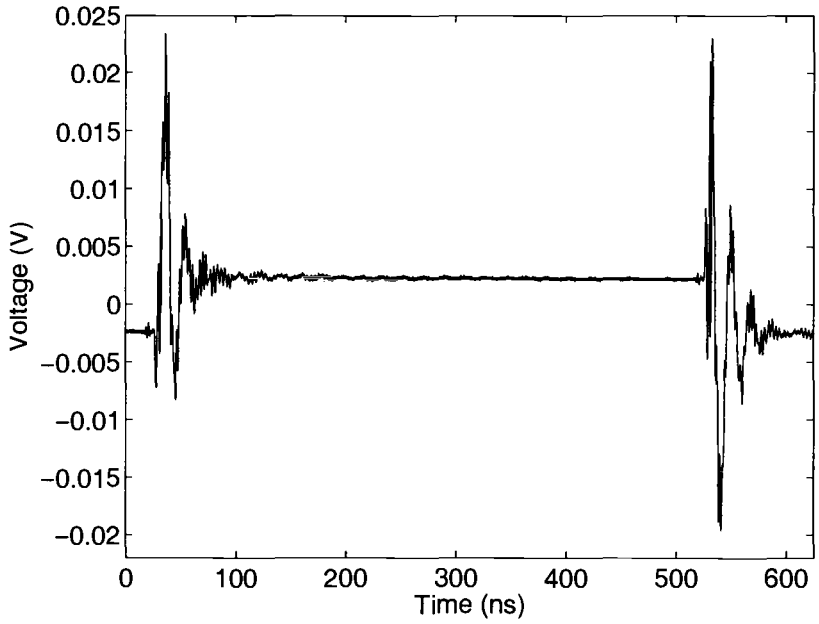
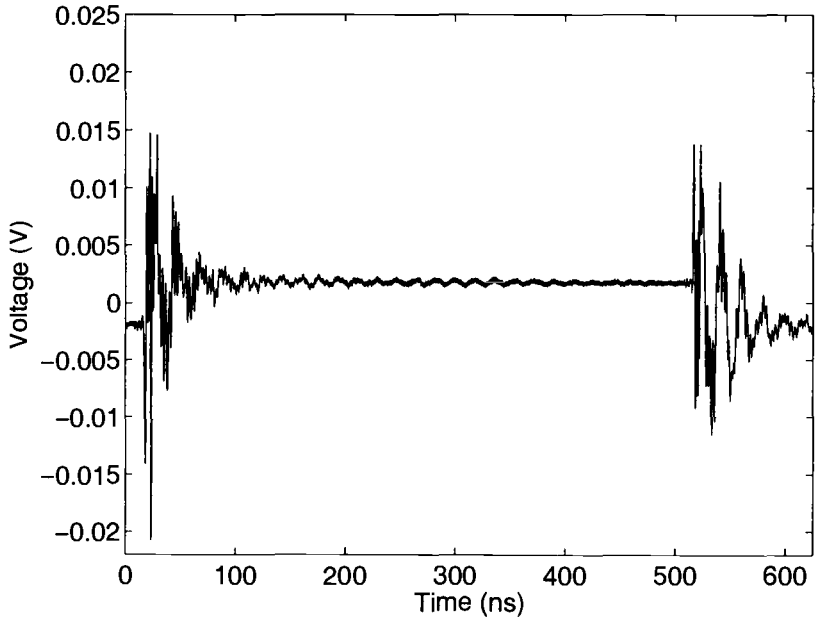


FIGURE 5.13. Simulated DSP noise with 1 bondwire per power rail.



(a)



(b)

FIGURE 5.14. Measurement results for the first DSP core with (a) the DPR floating and (b) the DPR grounded.

6. CONCLUSIONS AND FUTURE WORK

6.1. Conclusions

This thesis has presented a simulation methodology that can quickly and efficiently estimate the amount of noise generated by a large digital design in a heavily doped process. This implementation is over 80 times faster with one-tenth the memory usage compared to previous simulation approaches. While the accuracy of the three-step methodology may not be as high as that of a one-step, the needs of a designer are met so that the major noise injection paths can be quickly identified. This allows a designer to make early design decisions to reduce the effects of noise coupling.

Additional simulations and measurements have shown how large package parasitics can adversely effect noise performance, and how critical it is that accurate package models be considered during early stages of a design. Use of overly-simplistic package models can yield incorrect and potentially misleading results.

6.2. Future Work

While the simulation and measurement results show the major noise injection methods, there are still several areas where this work can progress to possibly produce more accurate results.

As mentioned in Chapter 2, the noise current library does not take into account any modification of the noise signatures by the power rail parasitics. The library model assumes no package parasitics, whereby the current waveforms may contain very high frequency spikes that are reduced when parasitics are introduced

and the power rail moves significantly. Future research may include this effect by defining a transfer function from the power rail voltage to the substrate noise current waveforms, and modify them accordingly for the final simulation stage. This could also possibly be modeled to a first order with simple inductors and capacitors added in the final top level SPICE/Spectre netlist.

Other possible future work could involve the reduction of bondwire parasitics in order to decrease the dominance of package parasitics in noise injection. This could be done using flip-chip packaging, however, the access to the die surface for measuring the noise can be an issue. This might be solved by mounting the chip onto a thin flex-board or other similar thin substrate with a hole cut out for noise probing. This is then mounted to standard PCB, as shown in Figure 6.1. Epoxy at the back side of the die keeps it firmly in place. With proper design, these traces could also be impedance-matched for RF designs.

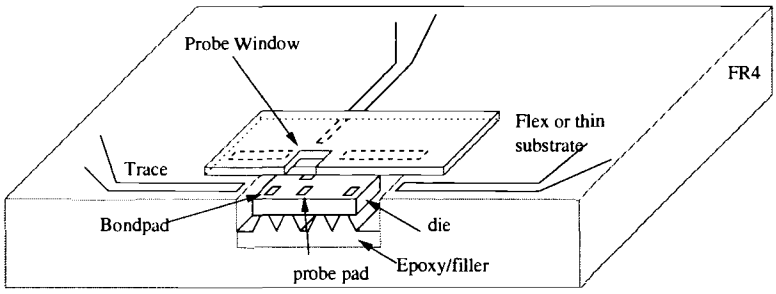


FIGURE 6.1. Low inductance package for probing.

A similar, but more difficult method would be to place the die in epoxy into a recession in the PCB as before, but instead use excessive epoxy, planarize the die to the PCB and then deposit metal. A chrome-glass mask would need to be made to then etch the traces. Another possibility is to simply package only one design at a time. The Eris chip suffers from the requirement that all designs be bondwired in one package. Flip chip and other low-inductance methods

will require that the number of pins be limited and spread out around the die. However, it has been noted that there is a justifiable need for this [13].

This work also relies on a heavily-doped substrate where the assumption can be made that that cross resistances (R12) are only necessary within the cell. This inherently ignores any top-level layout of the full design. However the current limitations of macro-model implementations require a large computation time for generating the substrate network, and work is being done to allow problems containing many contacts to be solved. Only when this problem has been solved can a lightly-doped design be tested, as cross-resistances become non-negligible with the high-resistivity of the substrate.

As mentioned previously, there are actually multiple contacts in the majority of standard cell designs, and that an approximate single contact has been used to lump their currents together. It can be shown that for this design, the substrate parasitics extraction requires significant computation time. If this problem can be reduced by extending the approximation from standard cells to groups of standard cells, a significant reduction in the parasitic network extraction time could result. It is believed, however, that this model will not work as well in lightly doped processes as it would in a heavily doped process due to the dependence on cross coupling resistances.

The Eris test chip also contains a small number of fall-back digital test structures that require their own wire-bonding. Given sufficiently low parasitics, these may be able to further verify the model. Their proximity to noise sensing structures is also beneficial.

Lastly, the methodology for measuring the noise transition vectors in the flow is highly inefficient, as it relies on functions inserted into the gate-level VHDL code. Although this allows Silencer! to work with any VHDL simulator, if the

measurement code were incorporated into the simulator, it would not have to parse and execute the VHDL. These could be more efficiently implemented directly into the high level simulator with C or C++, and output in binary format instead of ASCII text. This, however, requires access to the source code for an existing simulator, or the development of a entirely new source tree.

BIBLIOGRAPHY

[1] N.K. Verghese, D.J. Allstot and S. Masui, "Rapid Simulation of Substrate Coupling Effects in Mixed-Mode ICs" *Proc. of the IEEE Custom Integrated Circuits Conf.*, pp 18.3.1-18.3.4, May 1993

[2] P. Birrer,"Silencer! A Tool for Substrare Noise Coupling Analysis," *M.S. thesis*, Oregon State University, Corvallis, OR, 2004.

[3] H. Habal,"Accurate and Efficient Simulation of Synchronous Digital Switching Noise in Systems on a Chip," *M.S. thesis*, Oregon State University, Corvallis, OR, 2004.

[4] P. Miliozzi, L. Carloni, E. Charbon, and A Sangiovanni-Vincentelli, "Sub-Wave: a Methodology for Modeling Digital Substrate Noise Injection in Mixed Signal ICs," *Proc. IEEE CICC*, pp. 385-388, May 1996.

[5] R. Singh, "A Review of Substrate Coupling Issues and Modeling Strategies," *Proc. IEEE CICC*, pp. 491-498, May 1999.

[6] C. Xu, T. S. Fiez and K. Mayaram, "EPIC: A Program for the Extraction of Parasitics in IC's with a Green's Function Method," School of EECS, Oregon State University, 2003.

[7] A. Sharma,"Predictive Methodologies for Substrate Parasitic Extraction and Modeling in Heavily Doped CMOS Substrates," *M.S. thesis*, Oregon State University, Corvallis, OR, 2003.

[8] J. Ayers, "A comparison of Substrate Noise Coupling in Heavily Doped and Lightly Doped Substrates for Mixed-Signal Circuits" *M.S. thesis*, Oregon State University, Corvallis, OR, 2004.

[9] B. Owens, "Simulation, Measurement, and Suppression of Digital Noise in Mixed-Signal Integrated Circuits," *M.S. thesis*, Oregon State University, Corvallis, OR, 2003.

[10] M. Heijningen, M. Badaroglu, S. Donnay, G. G. E. Gielen, H. J. Man, "Substrate Noise Generation in Complex Digital Systems: Efficient Modeling and Simulation Methodology and Experimental Verification." *IEEE Journal of Solid-State Circuits*, vol. 37, pp. 1065-1872, August 2002.

[11] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitiative Approach*, Morgan Kaufmann Publishers, San Fransisco, 2003. Hennessy and Patterson, Computer Architecture and Design

[12] L. Forbes, W.T. Lim and K. T. Yan, "Guard Ring Diodes for Suppression of Substrate Noise and Improved Reliability in Mixed-Mode CMOS Cicruits," *Proc. of the Intl. Symp on Physical and Failure Analysis of Integrated Circuits*, pp. 145-148, Dec 1995.

[13] M. Felder and J. Granger, "Analysis of Ground-Bounce Induced Substrate Noise Coupling in a Low Resistive Bulk Epitaxial Process: Design Strategies to Minimize Noise Effects on a Mixed-Signal Chip," *IEEE Transactions on Circuits and Systems-II*, vol. 46, pp. 1427-1436, Nov. 1999.

APPENDICES

APPENDIX A. Simulation and Measurement Results

This section contains the simulation and measurement results for both DSP cores in each configuration, as well as additional measurements on other test circuits.

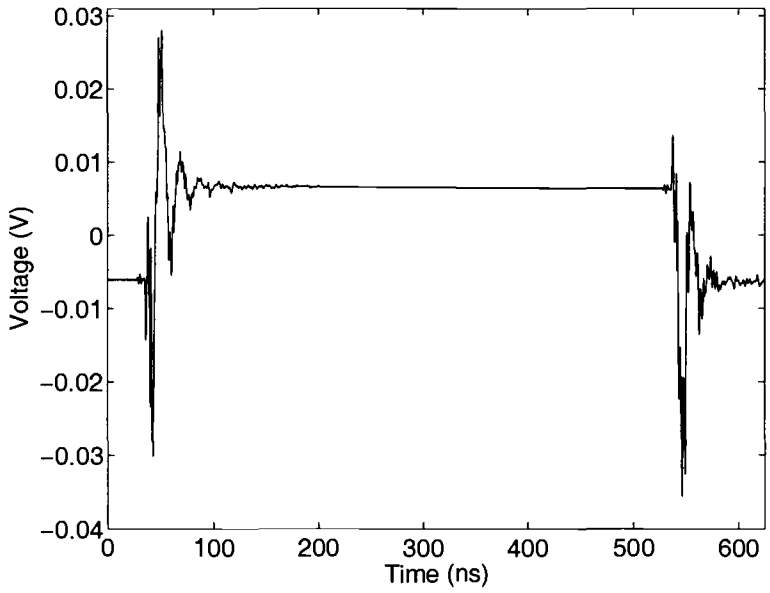


FIGURE A-1. Measurement results for both DSP cores running simultaneously with the DPR floating.

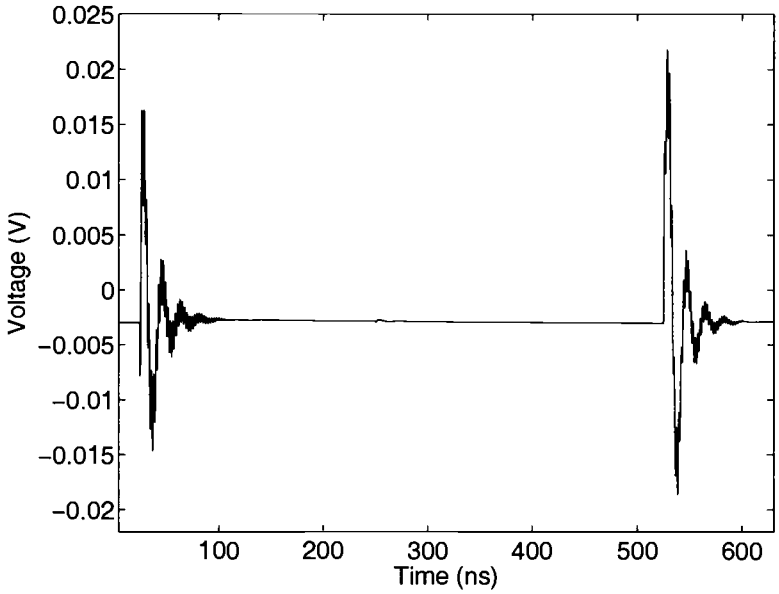


FIGURE A-2. Initial simulation results for the first DSP core with the DPR floating.

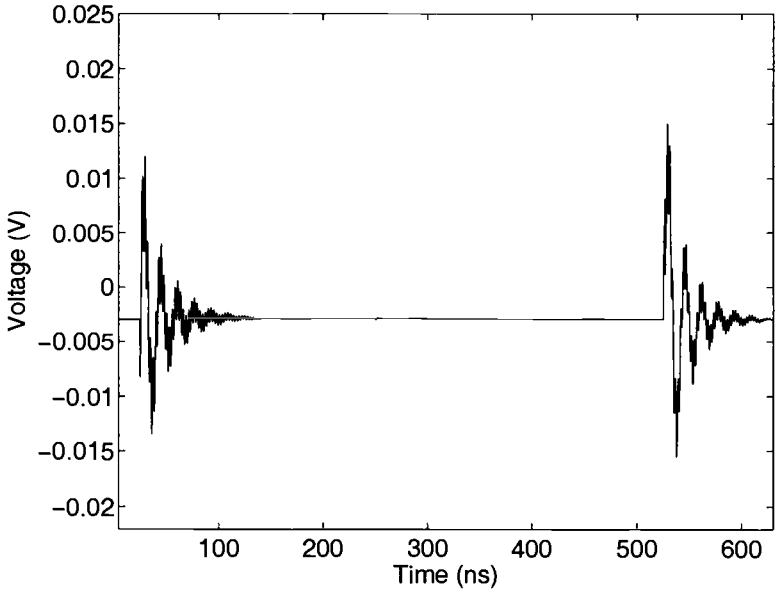


FIGURE A-3. Initial simulation results for the first DSP core with the DPR grounded.

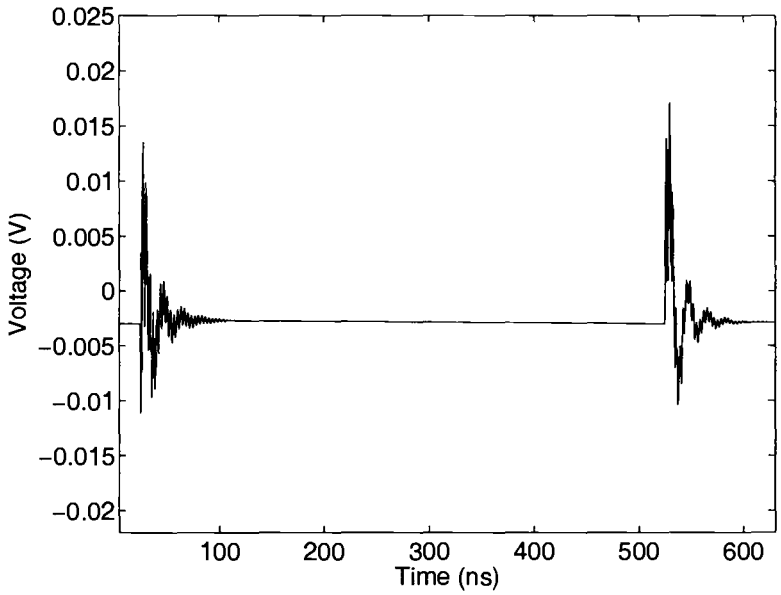


FIGURE A-4. Initial simulation results for the second DSP core with the DPR floating.

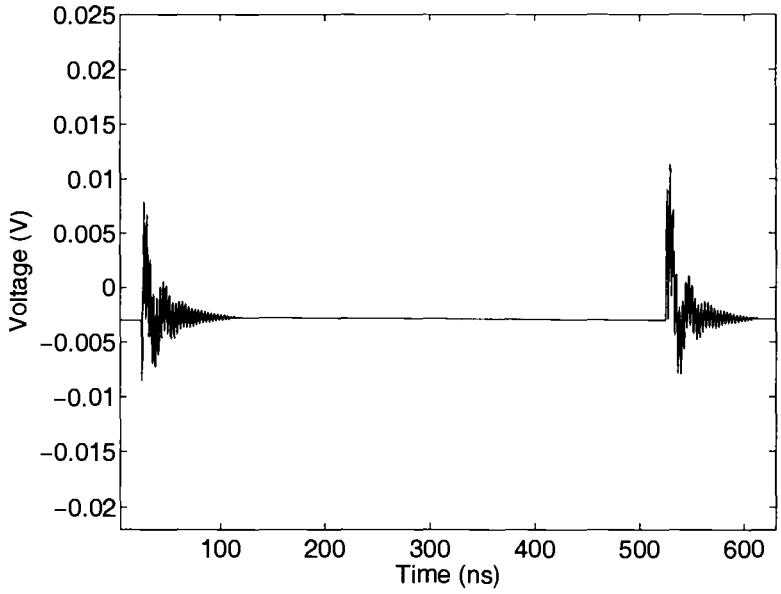


FIGURE A-5. Initial simulation results for the second DSP core with the DPR grounded.

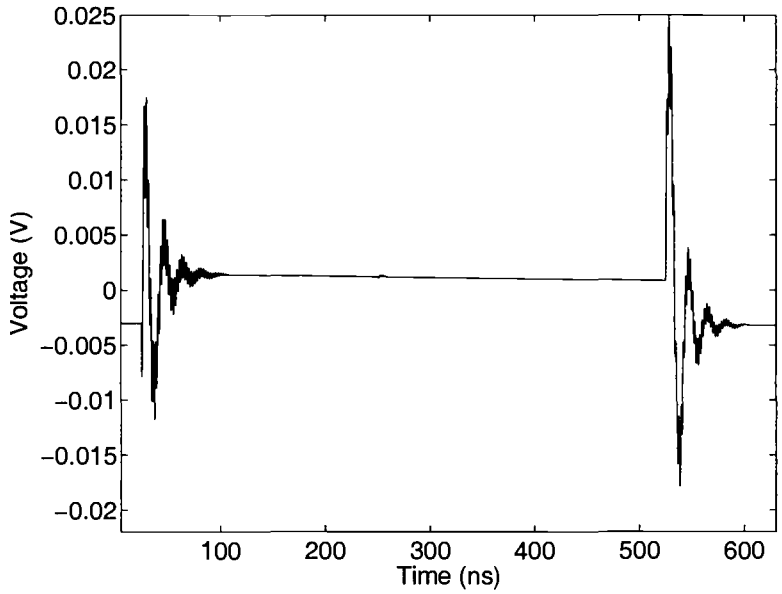


FIGURE A-6. Final simulation results for the first DSP core with the DPR floating.

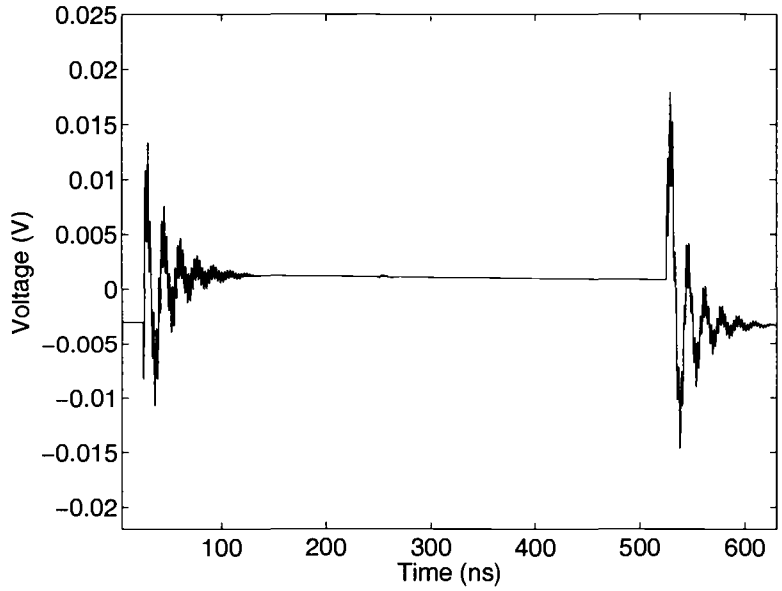


FIGURE A-7. Final simulation results for the first DSP core with the DPR grounded.

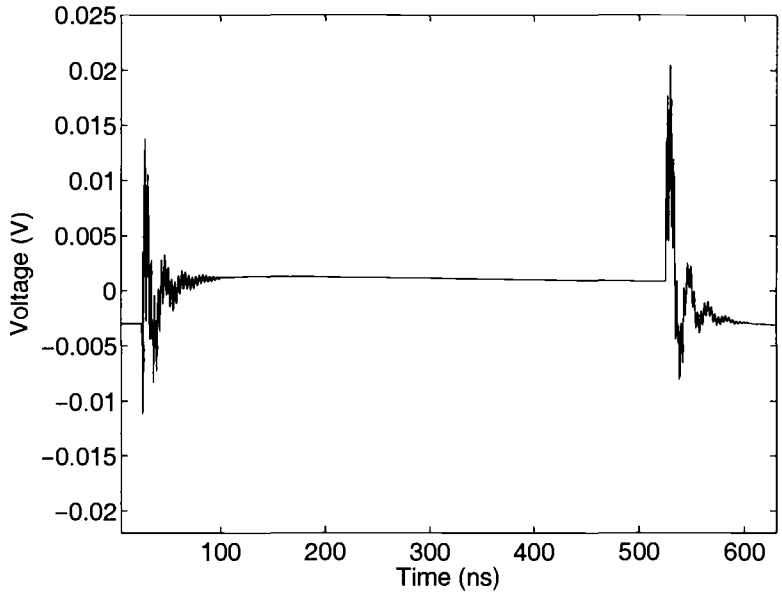


FIGURE A-8. Final simulation results for the second DSP core with the DPR floating.

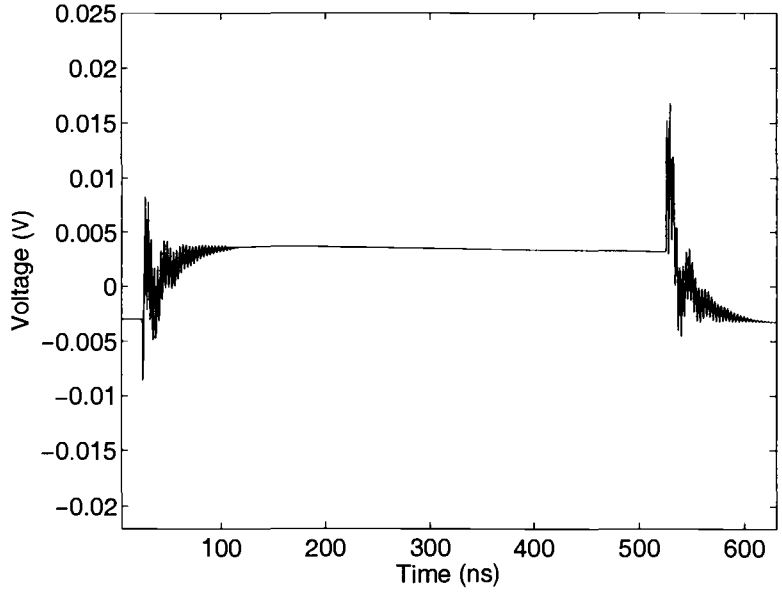


FIGURE A-9. Final simulation results for the second DSP core with the DPR grounded.

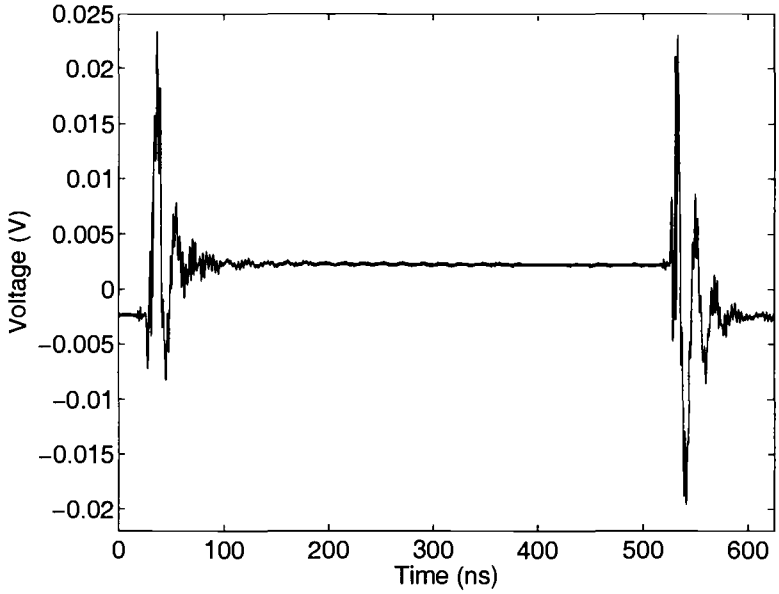


FIGURE A-10. Measurement results for the first DSP core with the DPR floating.

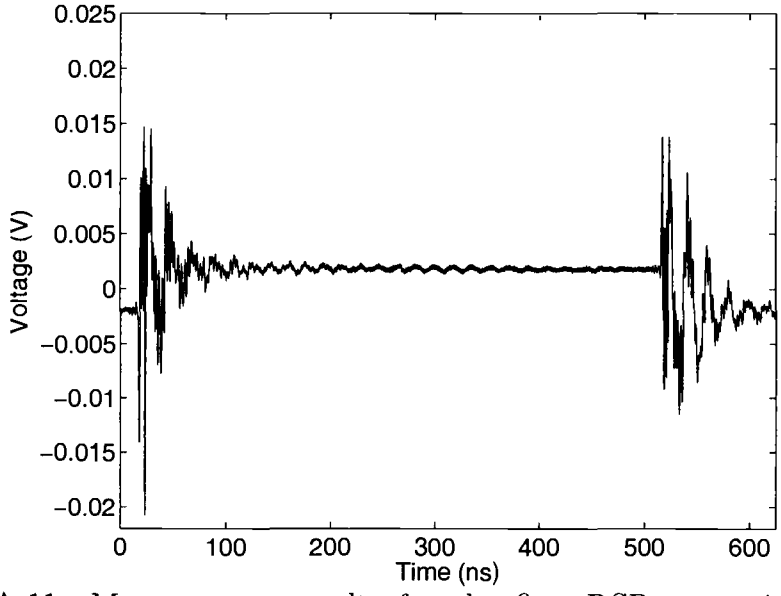


FIGURE A-11. Measurement results for the first DSP core with the DPR grounded.

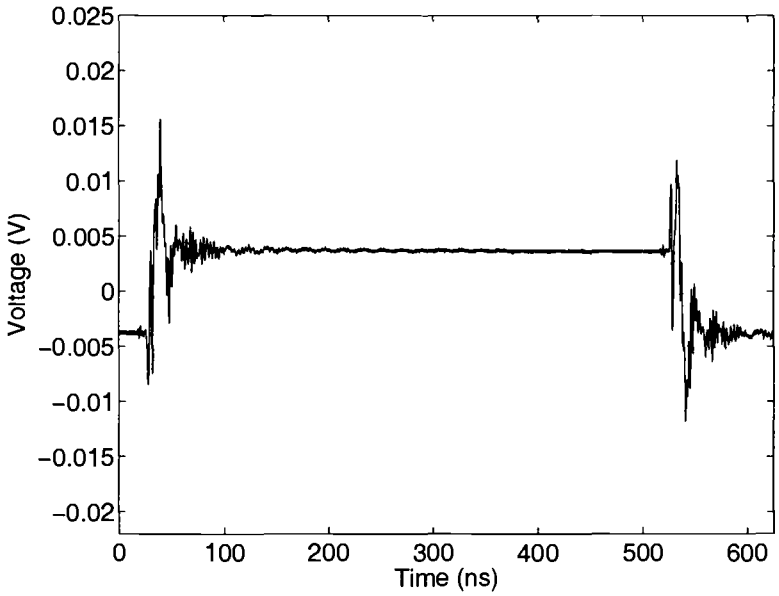


FIGURE A-12. Measurement results for the second DSP core with the DPR floating.

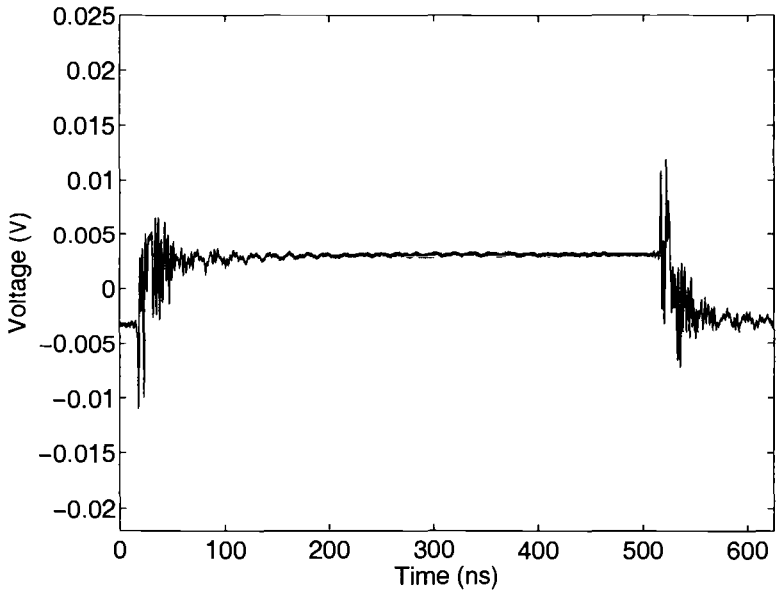


FIGURE A-13. Measurement results for the second DSP core with the DPR grounded.

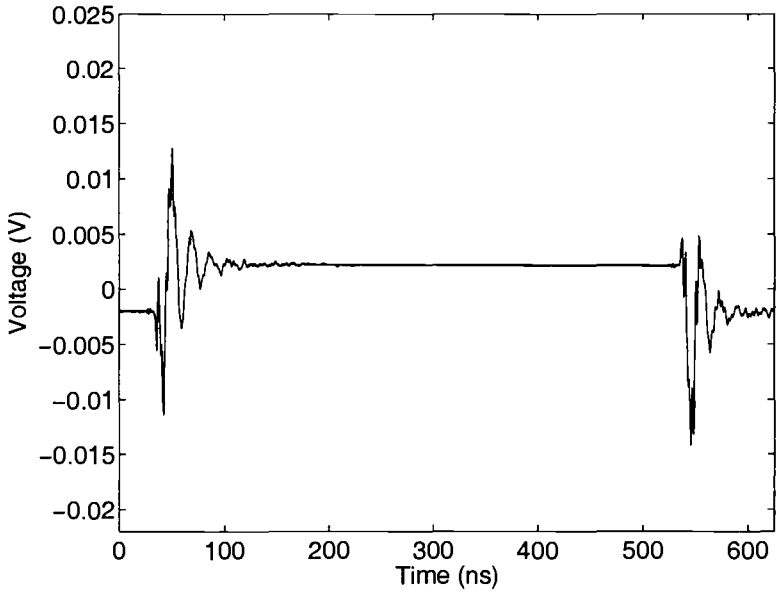


FIGURE A-14. Measurement results for the first DSP core to the second sense amplifier and the DPR floating.

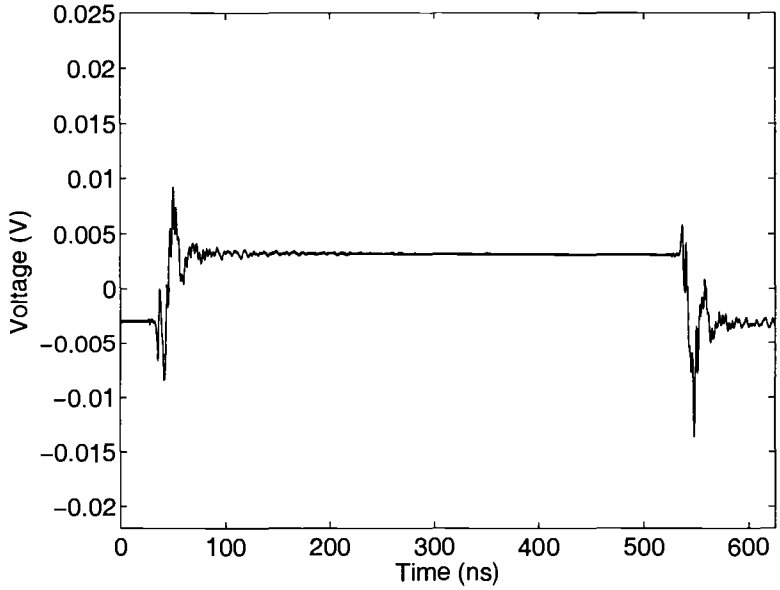


FIGURE A-15. Measurement results for the second DSP core to the second sense amplifer and the DPR floating.

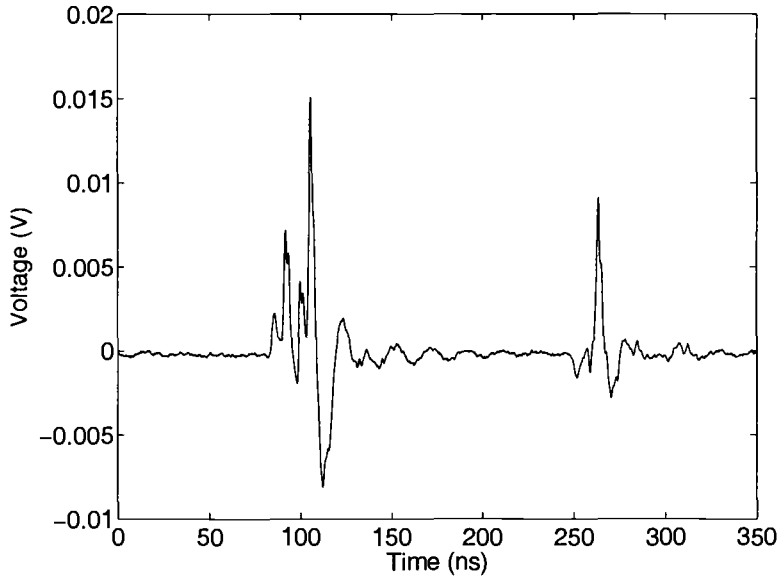


FIGURE A-16. Measurement results for the synchronous 8051, 3MHz clock, and the DPR floating.

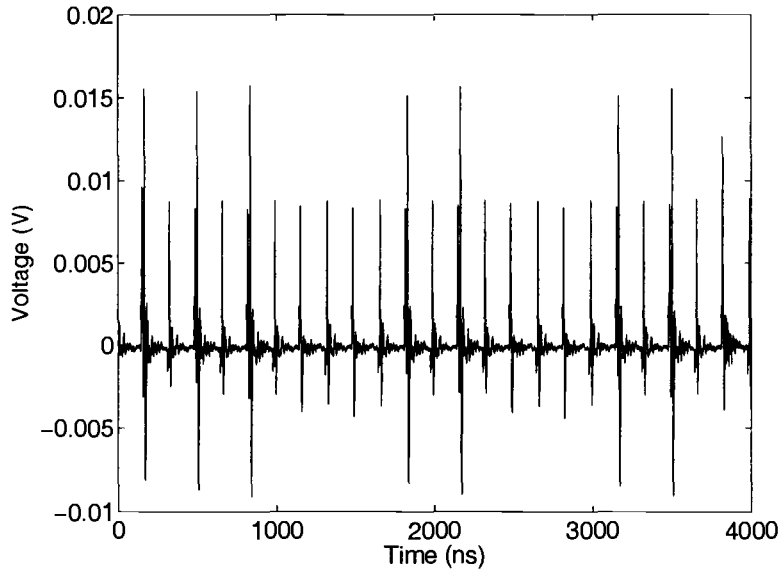
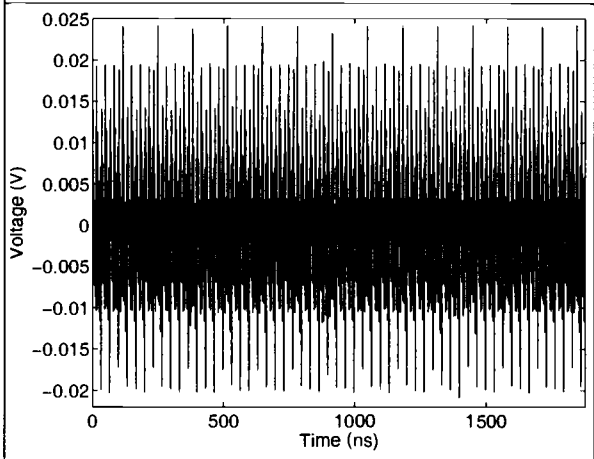
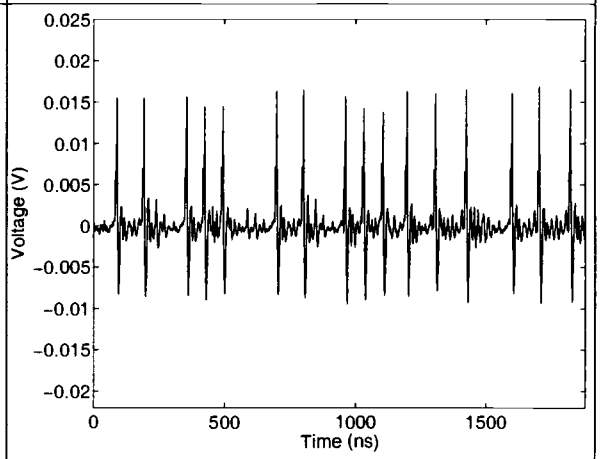
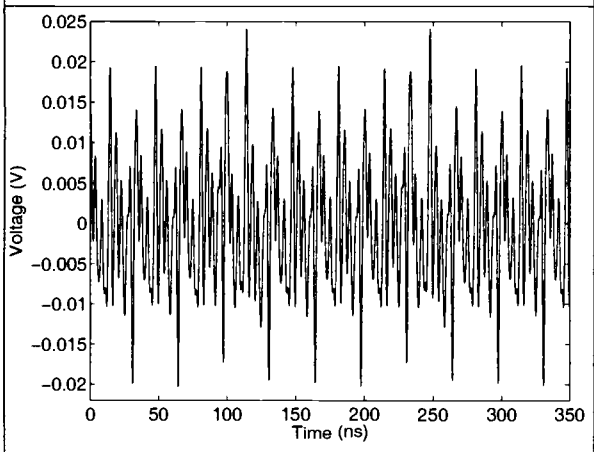
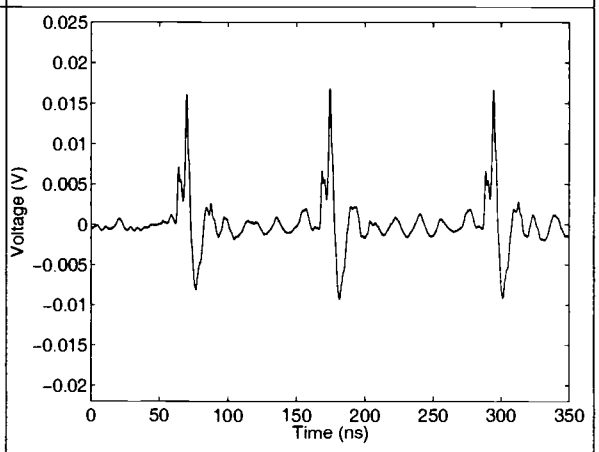


FIGURE A-17. Measurement results for the synchronous 8051, 3MHz clock, and the DPR floating.

TABLE 6.1. Comparison of syncronous and asyncronous measurements with the DPR floating.

Syncronous	Asyncronous
	
	

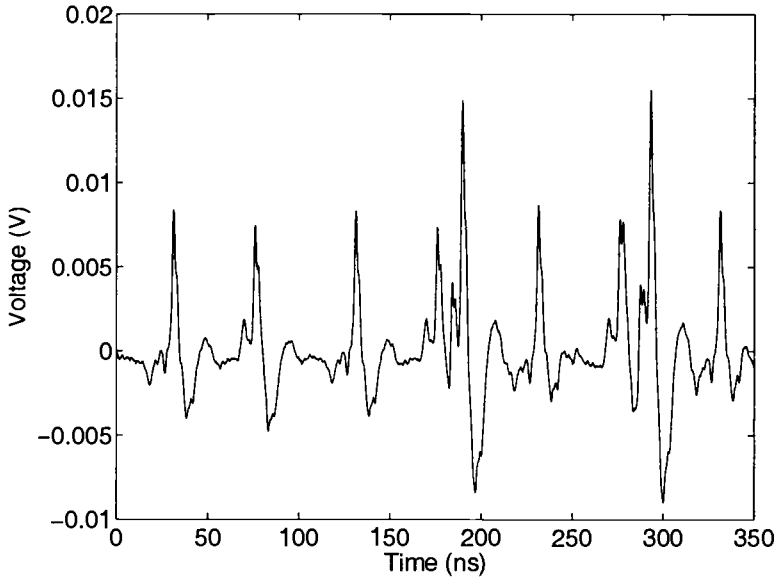


FIGURE A-18. Measurement results for the synchronous 8051, 10MHz clock, and the DPR floating.

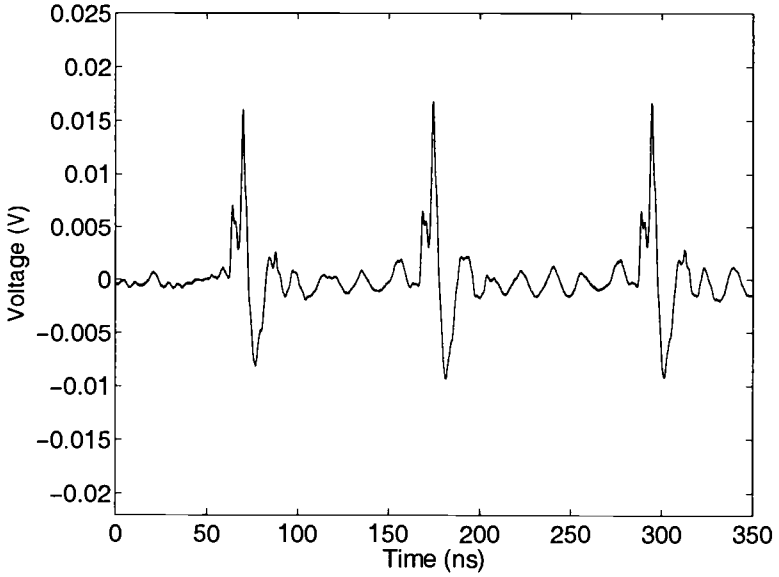


FIGURE A-19. Measurement results for the asynchronous 8051, and the DPR floating.

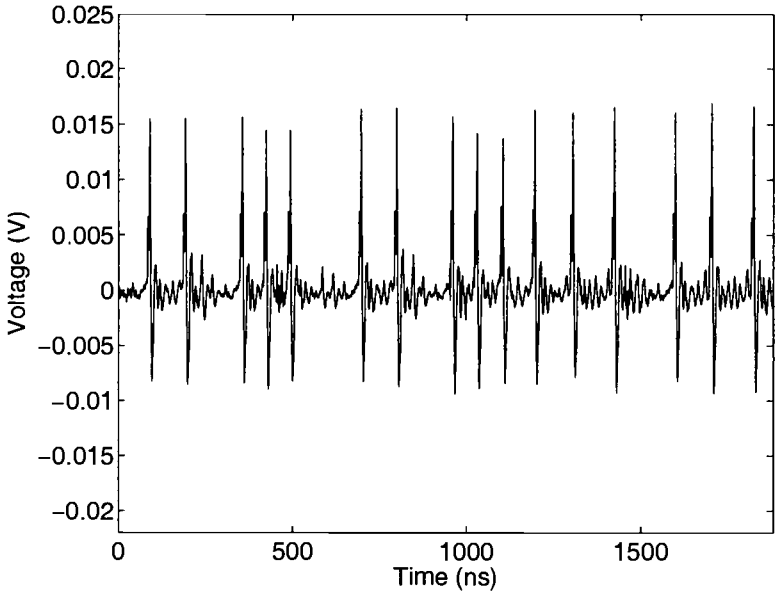


FIGURE A-20. Measurement results for the asynchronous 8051, and the DPR floating.

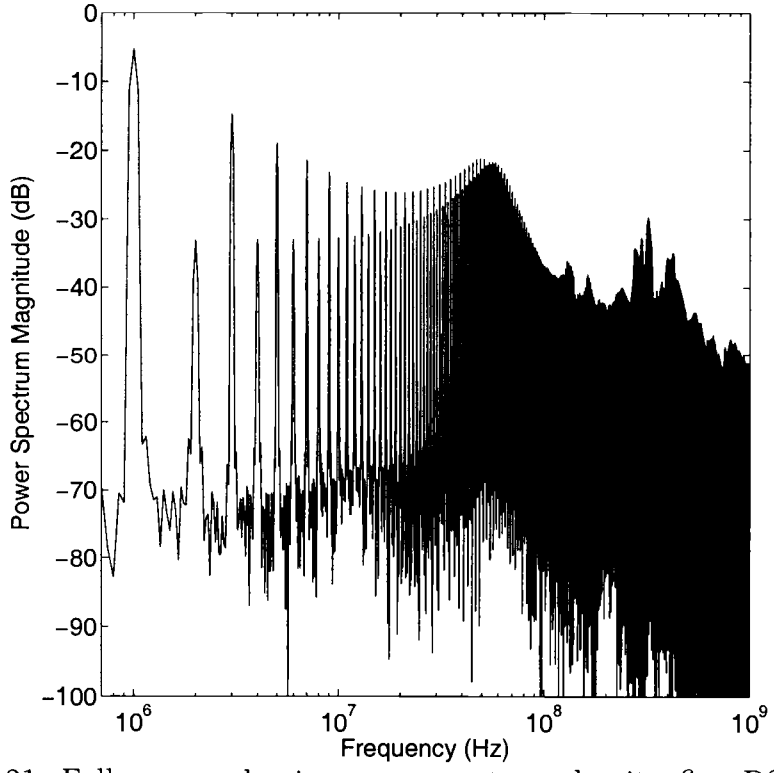


FIGURE A-21. Full measured noise power spectrum density, first DSP core with the DPR floating.

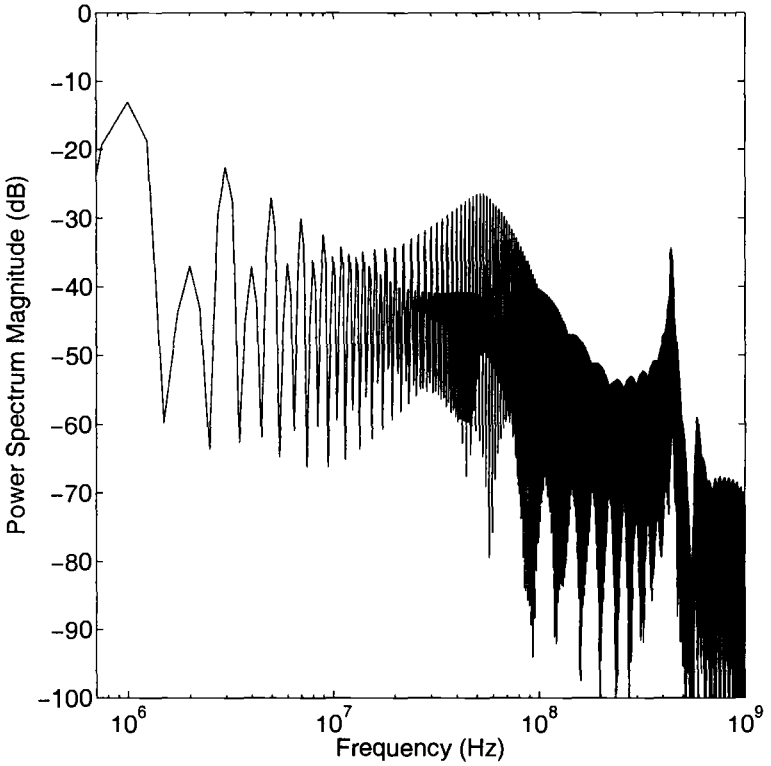


FIGURE A-22. Full simulated noise power spectrum density, first DSP core with the DPR floating.

APPENDIX B. DSP Core Description

The 16-bit DSP core is a 5-stage pipelined processor modified with forwarding logic to avoid stalls on data hazards, a two-instruction branch (test and branch) with delay slot to avoid branch stalls, and a boot ROM for initial loading of memory contents. A 16-bit Wallace-tree carry-save MAC block with 64-bit accumulator is added along with pathways to dual-port memories allowing for one multiply-accumulate instruction to be executed per clock cycle for up to 256 consecutive clock cycles. Data from the MAC block is read back directly into the register file, for which there are 4 dedicated registers.

The register file contains 16 total registers, half of which are special-purpose. Registers 0 and 1 are constant value registers and always return the value 0 or 1. Registers 2 and 3 contain incrementable address counters used for retrieving MAC data from RAM, and registers 4 through 7 contain the 64-bit resultant after a MAC load instruction is executed. Registers 8-15 are left as general-purpose use. Additionally, because of how the core's forwarding logic is designed, data written to registers 0,1, and 4-7 will be ignored, but can be accessed in the two clock cycles following an attempted write by causing an intentional read after write (RAW) data hazard, and using system flip-flops and forwarding logic as virtual registers.

Figure B-23 shows the basic architecture of the processor without the control logic

B.1. Primary uses

This DSP core is designed to primarily implement an efficient FIR filter. Programs demonstrating up to 384 taps have been tested that complete 16 MAC

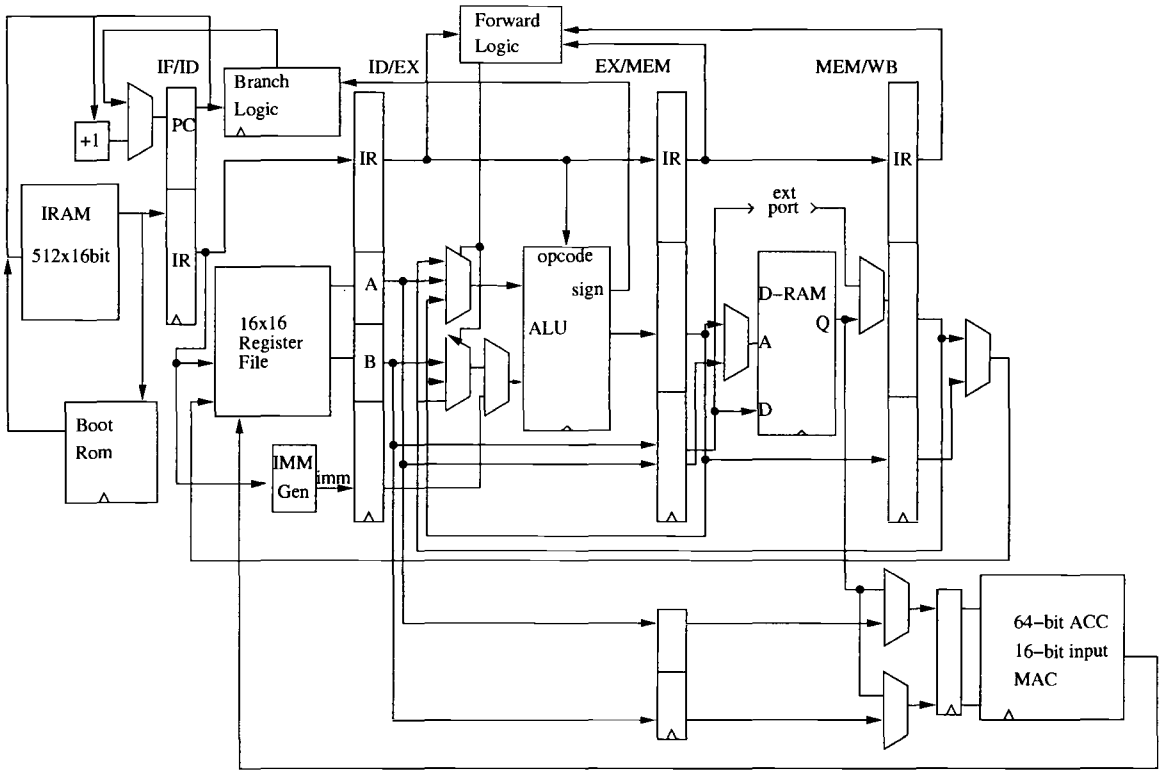


FIGURE B-23. Top level DSP core architecture.

instructions for every 19 clock cycles using counter loops, although possibly up to 512 taps could be implemented at the cost of efficiency. It can also be used as a generic data processor, although it only has one external 16-bit memory port which limits its ability to access external memories. For noise testing, a variety of instructions are used, including MAC, load, store, branch and various ALU functions.

B.2. Instruction Set Architecture

The DSP has 14 specific instructions as listed in Table B-1. All instructions are intended to be split into 4 fields of 4 bits, with the first containing the

instruction code, and the rest of the fields the opcode data. The first 8 instructions are classified as ALU instructions, and follow the format of having the second and third fields specifying the data in the register file to be operated on, and the last field being the destination register. The only exception is the shift operand, which only requires one input, so that the third field is used for parameters on how many bits the value is to be shifted, with or without carry, and in what direction.

The ALU has basic 16-bit add, subtract, and shift instructions, each with carry capability, as well as bitwise AND, OR, and XOR functions.

The load and store functions follow a similar field convention, where the second field always contains the register with the base memory location at which the value is either already stored in, or to be stored in. The sign extended immediate value (field 3 or 4) is also added to the base memory location to obtain the final address. This method allows for simple external data I/O as the address for the external memory port is 0xFFFF, which can be expressed as a base location of 0x00 (register 0) plus an immediate value of 0xF (-1). The other field (4 or 3) contains the register from which data will be retrieved or to which it will be stored. When external memory is accessed, either the data read or data write lines will pulse high for one clock cycle, indicating either that data on the bus has been read, or that that the current data is valid.

The test instruction specifies two registers to compare, how to compare them, and which branch bit register to store the result in. Possible branch types are branch on not equal (BNE), branch on equal to (BEQ), branch on less than (BLT), branch on more than (BMT), branch on greater or equal to (BGET) or branch on less than or equal to (BLET). In order to actually branch on the result of the test, the branch instruction is executed. By requiring two instructions for conditions and adding a delay slot, no computation cycles are wasted. A test

instruction can be executed within the delay slot, however its results will not be accessible until the next branch instruction executes. The branch instruction will use the result stored in the branch bit registers unless the condition bit is unset, where it will always branch.

The last instruction is the NOP or Stall instruction. This can be any of the three left over instruction slots. While 0xC and 0xD can be used as instructions in a future modified version of the core, 0xF should always be reserved as a null operand.

In order to sense noise on the substrate, two separate designs can be used. The primary method is the sense amplifier first designed for the TSMC 0.35μm process and later re-adapted for the IBM 7HP 0.18μm, the MIT LL 0.18μm SOI, and the TSMC 0.25μm processes. This amplifier is a low-gain differential amplifier with source-follower outputs designed to drive a pair of 50-Ω probes. The bandwidth of the amplifier is from 10kHz to 1.5GHz with a gain of 6dB. A schematic of the sense amplifier is shown in Figure B-24.

The other design that can be used for noise sensing is an opamp that is tied in a unity gain configuration. This design was originally used in a folding A-D converter, and its schematic is shown in Figure B-25. This design was included, however, in case the sense amplifier did not function for unexpected reasons.

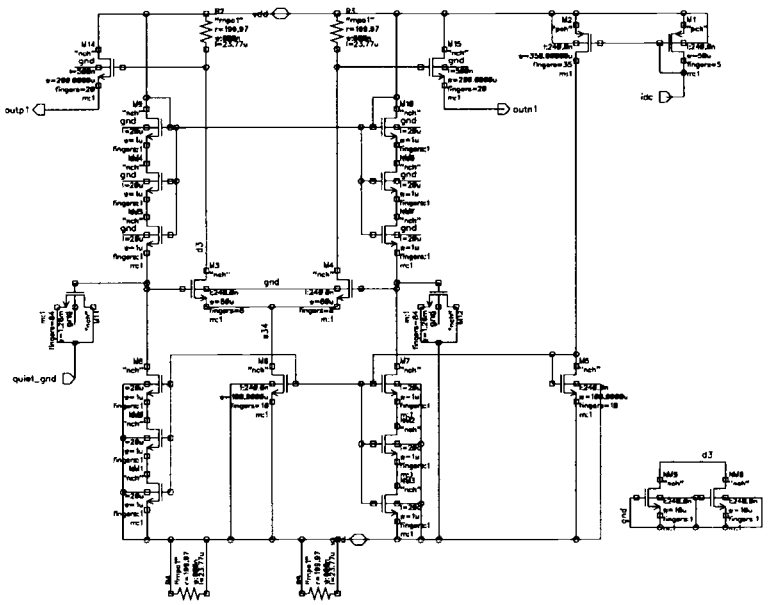


FIGURE B-24. Schematic for the substrate noise sense amplifier.

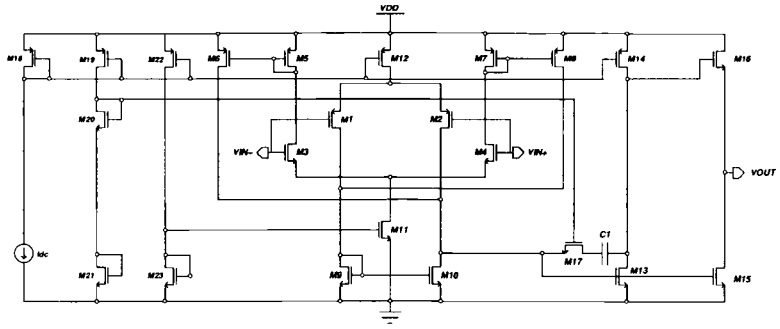


FIGURE B-25. Two stage opamp schematic.

TABLE B-1. DSP Processor ISA listing

Instruction	Field 1	Field 2	Field 3	Field 4	Function
Add	0x0	rs	rt	rd	rd = rs + rt
Sub	0x1	rs	rt	rd	rd = rs - rt
AND	0x2	rs	rt	rd	rd = rs a rd
OR	0x3	rs	rt	rd	rd = rs —— rd
Add w/Carry	0x4	rs	rt	rd	rd = rs + rt + c
Sub w/Carry	0x5	rs	rt	rd	rd = rs + rt + c
XOR	0x6	rs	0x	rd	rd = rs $\oplus$ rt
Shift	0x7	rs	D C (AMT)	rd	rd = rs shifted
Load	0x8	rs	imm	rd	rd = mem(rs+imm)
Store	0x9	rs	rt	imm	mem(rs+imm) = rt
Test	0xa	rs	rt	bb (MODE)	-
Branch	0xb	rel bit (IMM)	(IMM)	(IMM)	-
NOP	0xc	NA	NA	NA	-
NOP	0xd	NA	NA	NA	-
MAC	0xe	rs	rt	inc1 inc2 (MODE)	-
NOP	0xf	NA	NA	NA	-