AN ABSTRACT OF THE THESIS OF

Sirisha Adluri for the degree of Master of Science in Electrical and Computer Engineering presented on November 14, 2003.

Title: Contributions to Substrate Noise due to Supply Coupling and Pin Parasitics

Abstract approved:

Redacted for Privacy

Kartikeya Mayaram

Redacted for Privacy

Terri Fiez

This thesis presents the contributions to substrate noise due to supply coupling and the effect of pin parasitics on the substrate noise generated by digital circuits. Various sources of substrate noise and their effect on analog circuits sharing the same substrate are discussed. A simulation approach to isolate the various components of substrate noise is elaborated. The importance of including package parasitic models in digital noise simulations is discussed and the dominance of supply inductance over the other package parasitic elements is established. The effects of transistor sizing and supply inductance on the noise picked up by a sensor placed close to a digital circuit are described. Some noise suppression techniques and their validation from measurements on a TSMC 0.35μm process mixed signal chip are also presented.

Contributions to Substrate Noise due to Supply Coupling and Pin Parasitics

by

Sirisha Adluri

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented November 14, 2003
Commencement June 2004

Master of Science thesis of Sirisha Adluri presented on November 14, 2003

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

Sirisha Adluri, Author

ACKNOWLEDGEMENTS

First, I would like to thank my advisors Dr. Karti Mayaram and Dr. Terri Fiez for giving me an opportunity to work in the substrate coupling group at OSU. They have been a constant source of guidance and support during every stage of my research work. I took several of their classes which helped me hone my skills in the field of Analog Circuit Design.

I express my gratitude to Dr. Un-Ku Moon and Dr. Gabor Temes for the courses they taught and for instilling the much required confidence to continue in the field of Analog Circuits. I would like to thank Dr. Chengang Xu and Dr. Dean Jensen for agreeing to serve on the committee for my Masters Defense.

I would like to take this opportunity to thank Ferne Simendinger, Sarah O'Leary, Brian Lindsley and all the office staff for all the help provided during my academic stay at OSU.

My sincere thanks to all the group members of the substrate coupling group, Brian Owens, Ajith Sharma, Shuching Hsu, Husni Habal for their help and co-operation over the two years. I thank Patrick Birrer for helping me in proof reading my thesis report and his useful suggestions in my work. I am also grateful to Taras Dudar for the help and guidance he gave during the initial phases of my work. Thanks also to Vova for the help with simulations in HSPICE and Cadence.

I am also grateful to Kannan Soundarapandian for the useful suggestions in my research area, I would also like to thank Jose Silva and Robert Batten for their

help during measurements. Thanks are also due to Dr. Chengang Xu for his help in using EPIC.

I would like to acknowledge all the people in the lab, Manas, Prachee, Raghuram, Manu, Triet, Prasad, Shubha, Vivek, Sasi, Sudhakar, Yuhua, James, Paul, Scott and Hui En for making my days at work memorable.

I am deeply indebted to my husband Gowtham, for guiding me and helping me develop an interest in the field of Analog Design and for being there all the time. Without him, I wouldn't be at this stage today. I am also grateful to my parents, for providing an opportunity to succeed in my life, for showing me the right direction, always. I thank God for giving me the strength to help me move on, during good times and bad.

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF TABLES

CONTRIBUTIONS TO SUBSTRATE NOISE DUE TO SUPPLY COUPLING AND PIN PARASITICS

1. INTRODUCTION

The integration of both analog and digital circuits to create systems on a chip has become a reality in present day IC designs. With this high level of integration, substrate noise coupling from the switching digital circuits to the sensitive analog circuits is a significant problem. This noise coupling occurs through the shared silicon substrate and the power lines of the analog and digital circuits [1]. While device scaling has enhanced the raw speed of transistors, unwanted interactions between sections of the integrated circuits have also increased. Hence, it is becoming increasingly important to be able to predict the amount of substrate noise generated by digital circuits in the design phase. Furthermore, the operation of analog circuits on the same substrate needs to be analyzed properly.

To predict substrate noise accurately, issues such as inclusion of a model for the substrate, package parasitics, backplane connection etc. need to be addressed. Various approaches to modeling the substrate are discussed in [2-10]. This thesis deals with the issue of proper inclusion of package parasitics in digital noise simulations and their effect on substrate noise generation. The thesis also discusses

the need for separating supply noise components from the switching noise components of substrate noise in simulations and elaborates on a way to do it.

Chapter 2 discusses the different sources of substrate noise in detail. It also explains the different ways in which substrate noise can affect the analog circuitry which share the same substrate with digital blocks [13], [14]. Chapter 3 describes the use of a substrate macro model in digital block simulations and briefly describes various substrate models [2-10]. A simulation approach to isolate supply and switching noise components of the substrate noise is explained. In Chapter 4, the effect of pin parasitics are described [14].Various types of packages that are available and the range of their parasitics are presented [15-18]. Some of the package models described are used in simulations performed on a digital stepped buffer. The stepped buffer and its design are also described. Simulations illustrate how substrate noise will be underestimated if package parasitics are not included. It is shown that inductance dominates the parasitic effect compared to the resistors and capacitors. Various other factors affecting the substrate noise, like size of transistors, rise and fall time of input signals are discussed. Chapter 5 discusses some noise suppression techniques [8], some of which are validated using measurements on a mixed-signal test chip fabricated in a 0.35μm CMOS TSMC process. Measurements also show that substrate noise can be accurately predicted if appropriate pin parasitics are added in simulations. Conclusions and future directions are given in Chapter 6.

2. SOURCES OF SUBSTRATE NOISE

A digital circuit is a block that deals with signals which are at one or the other of two logic levels. The signals are either at the highest level (logic 1) or the lowest level (logic 0). For example, an inverter, shown in Figure 2.1 is a digital circuit which inverts the logic level of the input and presents it at the output. Switching from one logic level to another at high speeds tends to generate noise due to parasitic elements such as capacitors and inductors. This chapter discusses the various sources of noise generated by digital circuits. These sources of substrate noise are described using a simple inverter example.

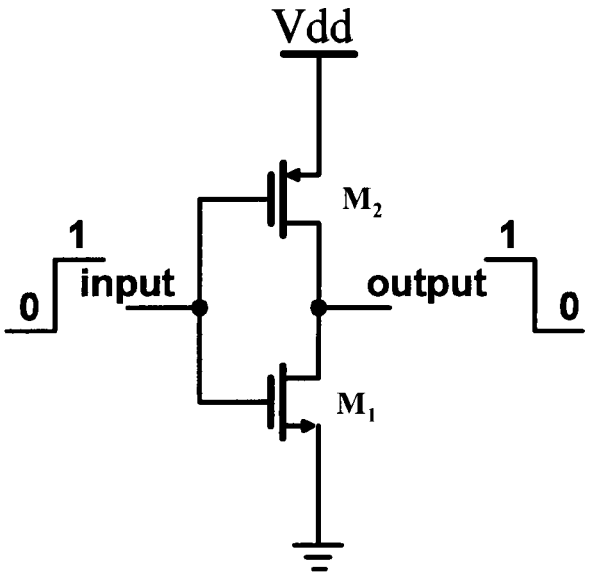


Figure 2.1 Schematic of a simple inverter.

Some terminology which will be used in the following sections is described below.

Bulk node: The bulk connection in the schematic of a transistor (node BN in Figure 2.3).

Bulk/Substrate contact: The bulk/n-well contact in the cross-section of a transistor (Figure 2.2). This is used to connect the substrate/n-well to ground/vdd.

Backplane: The back side of the p- substrate, shown in Figure 2.2.

2.1 Noise due to switching transistors

In an inverter, when the input signal makes a transition from high to low (falling edge) or from low to high (rising edge), the output makes a transition in the opposite direction. There is a junction capacitance C_{db} from the output to the bulk node for the NMOSFET, as shown in Figure 2.2. For simplicity, the PMOS transistors are not considered. Ideally, the substrate should be connected to ground. In this case, any kind of transition at the output will not affect the substrate. But in reality, the substrate is connected to ground through a substrate contact, shown as the p+ substrate contact in Figure 2.2. As can be seen from the cross-section of the layout of the inverter, the substrate is not connected to an ideal ground, due to the resistivity of the substrate. There will be some impedance in the path from the substrate to the ground, shown as R, from the bulk node to the substrate contact in this figure. Any transition in the output is coupled to the substrate through the junction capacitance. The resulting substrate voltage shows the same characteristic as the switching node (output node), i.e., if the output transitions from high to low, the substrate voltage

shows a negative peak and if the output transition is from low to high, the substrate voltage has a positive peak.

Substrate noise due to the switching transistors is illustrated using an inverter, with size $(W/L)_n$ = 100μm/0.6μm, $(W/L)_p$ = 200μm/0.6μm (Figure 2.3). A 10Ω resistor is added in the ground path to the substrate. The noise that is present at the NMOSFET bulk node (BN in Figure 2.3) is shown in Figure 2.4. There is no impedance added in the ground path from the source of the NMOS transistor. For simplicity, the bulk connection of the PMOS transistor is considered ideal.

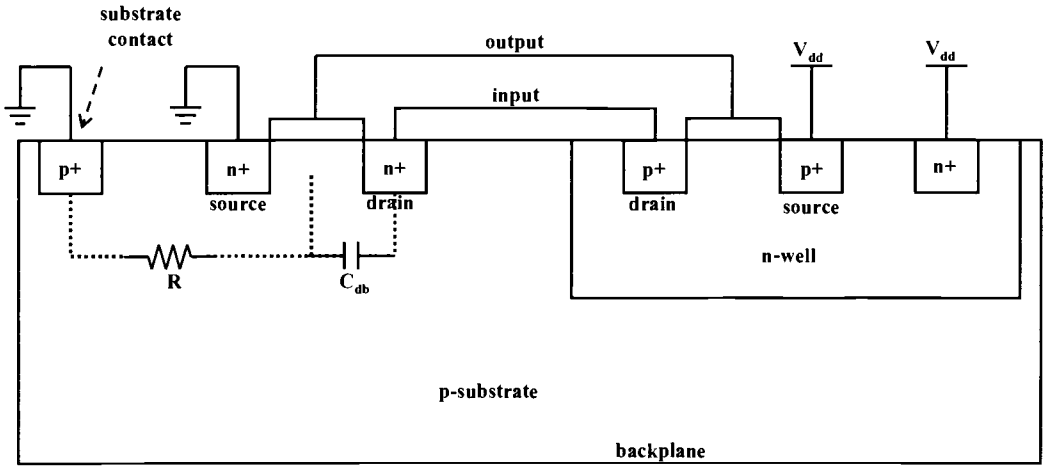


Figure 2.2 Cross-section of a simple inverter.

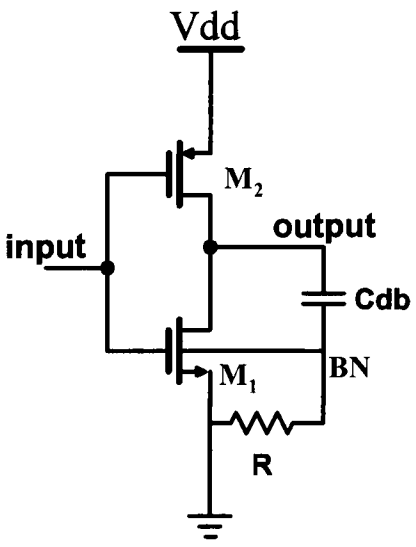


Figure 2.3 Simulation setup to illustrate substrate noise due to switching transistors. C_{db} is included in the MOSFET SPICE model but shown here to illustrate the coupling of switching noise to the substrate.

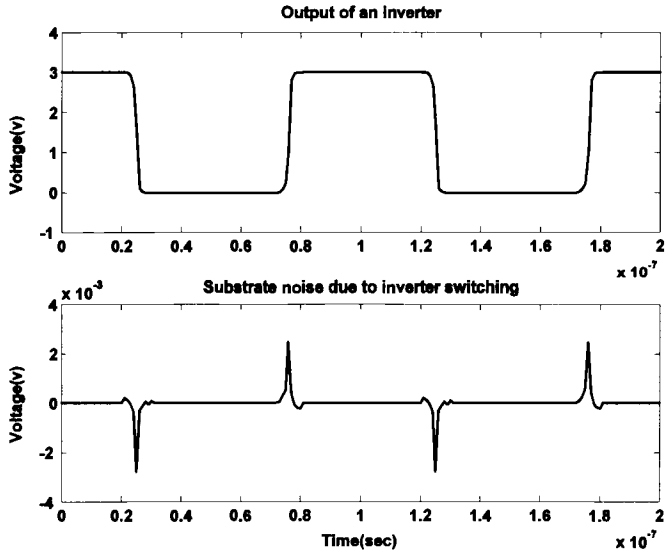


Figure 2.4 Switching noise measured at the bulk node of the NMOSFET in an inverter.

2.2 Noise due to power supply coupling

So far, we have assumed that the power supplies for the inverter are ideal. In reality, there is a finite impedance in the supply lines due to a number of factors. They include package inductances, resistance of the metal traces that connect the supplies to bond pads, and bond wire inductance. These impedances affect the substrate noise in the following way. When an input signal makes a transition from high to low or from low to high, it does so very quickly. There will be a short time during which the input will have a value such that both the PMOS and NMOS transistors are "on" at the same time. Due to this, there will be a direct path from the positive supply to the negative supply and there will be a finite amount of current flowing through the transistors, which is called the 'shoot through' current [22]. Due to this current and the impedance in the supply lines, the actual supply voltages for the inverter will vary. These variations in the supply voltage get coupled to the substrate, either directly, if the bulk node is tied to the source node of the transistor or through the junction capacitors from the source node to the bulk node, if the source and bulk nodes are not tied together.

The supply noise present at the NMOSFET bulk node when an inverter of size $(W/L)_n$ = 100μm/0.6μm is simulated, with a 10Ω resistor in the ground path of the NMOSFET source of the inverter (Figure 2.5) is shown in Figure 2.6. The supply to the PMOSFET source is considered ideal, for simplicity. When the output makes a transition from high to low, any capacitance present at the output node discharges and the discharge current flows into ground. Hence, the ground bounce is positive.

Similarly, a vdd bounce is negative, since the current required to charge the output capacitance when the output node goes from low to high is taken from the positive power supply. If any of the impedances in the supply lines are inductive, the current flowing through them results in an inductive effect or *Ldi/dt* voltage bounce. This will be described in detail in Chapter 3. The inductance combined with the circuit capacitances between the supply and ground causes ringing of the power supplies, which gets coupled into the substrate.

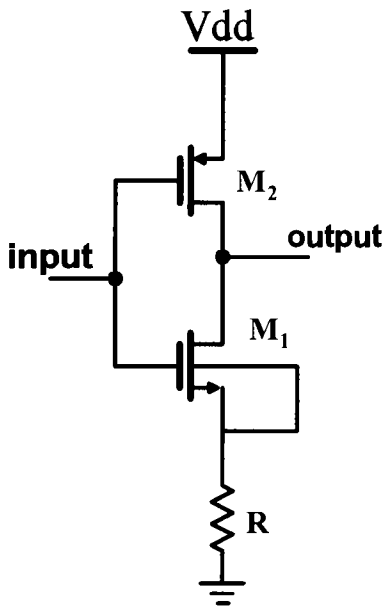


Figure 2.5 Simulation setup to illustrate substrate noise due to supply coupling.

2.3 Impact ionization

Impact ionization is a highly localized phenomenon due to high electric fields experienced in the drain-channel-substrate interface. When the electric field in the depleted drain end of a MOS transistor becomes large, the electrons achieve high

velocity and collide with the silicon atoms in the substrate, to create electron hole pairs. This gives rise to a substrate current. Impact ionization is important when low frequency input signals are used in digital circuits [11].

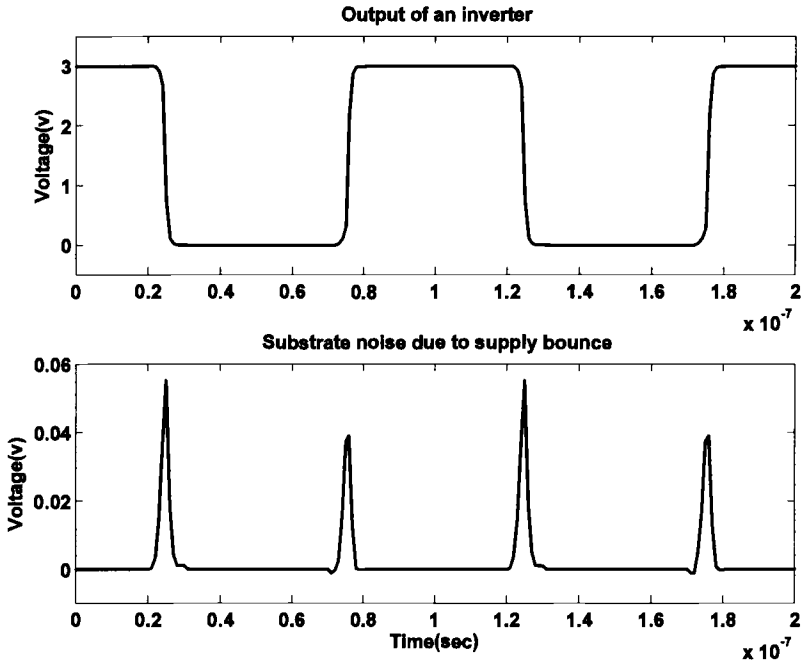


Figure 2.6 Noise at the NMOS bulk node of an inverter due to supply bounce.

2.4 Propagation of substrate noise and the effect on analog circuit performance

A mixed-signal circuit has both analog and digital circuits on the same substrate. The relentless switching of the digital circuits constitutes a significant source of noise, as was seen in the previous section. This noise propagates to the analog circuits in many ways: through the power supply lines, through the common

substrate and by coupling through the signal lines. These noise sources may adversely affect the performance of the analog circuits in many ways.

If the analog and digital circuits share supply lines and if the power supply rejection ratio (PSRR) of the analog circuit is low, then the performance of the analog circuit could be affected due to the noise present in the supply lines. The coupling of noise through the supply lines can be alleviated by using different supplies for the analog and digital circuits. Also, by increasing the PSRR of the analog circuits using techniques such as differential operation, the effects of supply noise on analog circuit functionality can be minimized. Supply noise minimization techniques will be discussed later.

If any of the substrate noise sources discussed in the previous sections change the common substrate potential, it results in a change in the threshold voltage of analog transistors [13]. Since the dependence of the threshold voltage on the substrate potential is nonlinear, substrate noise increases the nonlinearity of the analog circuits.

The amount of coupling through the substrate depends on the doping of the substrate material. Two types of substrates are typically used for CMOS ICs: heavily doped and lightly doped substrates.

2.4.1 Heavily doped substrates

A heavily doped substrate [2] is used in mixed-signal ICs to avoid latch-up [14] problems resulting from parasitic bipolar transistors. The bulk is heavily doped for a depth of 4μm to 300μm and hence the resistivity is on the order of 1 mohm-cm. Figure 2.7 shows the cross section of a typical heavily doped substrate. These

substrates use a thin epitaxial layer with low doping (high resistivity). Noise coupling between circuits takes place through the high resistivity epi-layer for small separations. When the separation of the components is more than 100 μm, most of the coupling between the circuits takes place through the low resistance bulk, due to the heavy doping. Therefore, the coupling is nearly independent of distance beyond a certain separation, which means placing the analog and digital circuits farther apart will not reduce the amount of substrate noise coupled to the analog circuit. Since the coupling between circuits takes place through the low resistive bulk for separations greater than 100μm, guard rings are not very effective in reducing the coupling unless they are placed very close to the digital circuits.

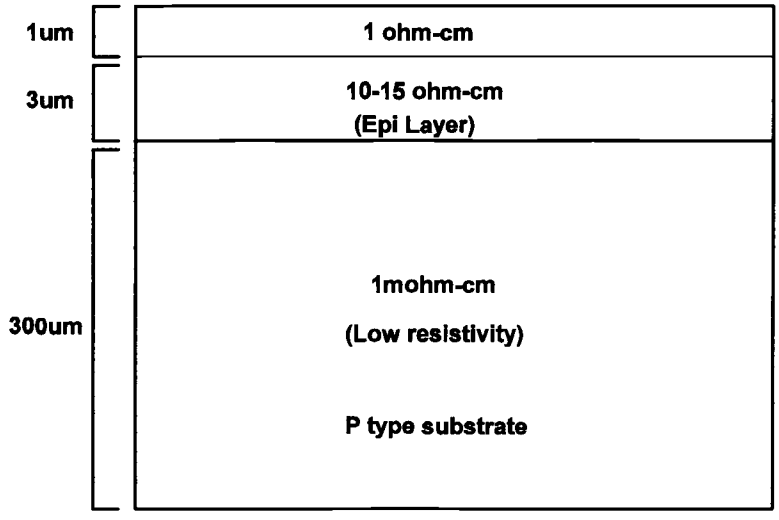


Figure 2.7 Cross-section of a heavily doped substrate.

2.4.2 Lightly doped substrates

This type of substrate has a bulk which is lightly doped and hence has a higher resistivity (20-50 ohm-cm). Figure 2.8 shows the cross section of a typical lightly doped substrate. In this case, the coupling is expected to be a strong function of the separation between circuits. Therefore, use of guard rings and increasing spacing between analog and digital circuits are effective techniques for reducing coupling in lightly doped substrates.

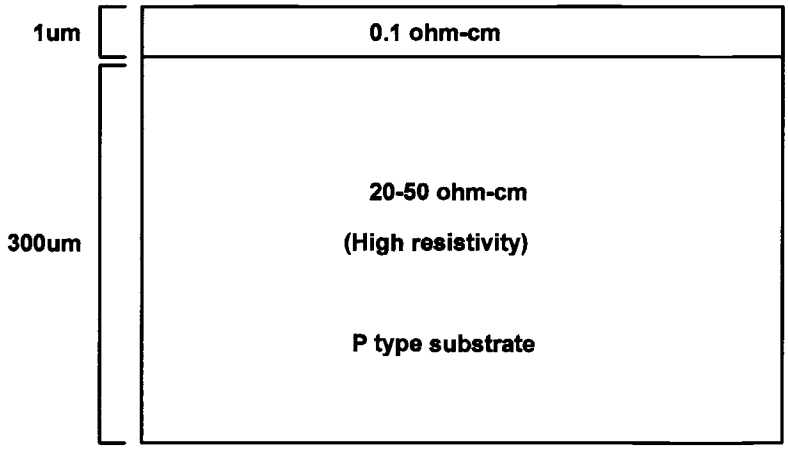


Figure 2.8 Cross-section of a lightly doped substrate.

3. SIMULATION APPROACH TO DISTINGUISH SUPPLY AND SUBSTRATE NOISE

3.1 Need for a substrate model

To accurately simulate the amount of substrate noise generated, the various sources of substrate noise discussed in the previous chapter must be included in simulations. Switching noise generation at the bulk nodes due to the capacitive coupling from the source/drain junctions is included in the BSIM3v3 transistor model through the junction capacitances. A SPICE model of an inverter with a substrate model included is shown in Figure 3.1. Resistors between the bulk nodes and the substrate contacts of transistors have to be included in simulations, to emulate the non-ideal ground provided by the substrate contacts at the bulk nodes (resistors R_{01} and R_{11} in Figure 3.1). The propagation of the noise from the bulk nodes to the substrate must be included in the simulations, by adding resistors from the bulk nodes to the substrate (resistors R_{12} and R_{02} in Figure 3.1). Impact ionization and threshold voltage variations are also included in the model of the transistor.

A model is needed to include the noise due to the supply coupling. The coupling of the supply noise from the substrate contact to the substrate has to be included by adding resistors from the substrate contacts to the substrate (resistors R13 and R03 in Figure 3.1). The coupling from the positive power supply to the substrate is included by adding capacitors and resistors from the n-well contacts to the substrate.

There are many substrate models available. The models differ in the way the substrate is treated and in their complexity. Some of the models for the substrate network are briefly described below.

3.1.1 Finite difference method

In this method [5], the Laplace equation is discretized by using a mesh over the whole volume of the substrate. This method is accurate and versatile, but computationally expensive, especially when fine meshes are used.

3.1.2 Boundary element method

Boundary element methods are Green's function based, in which, the substrate surface voltage is expressed as an integration of Green's function over the substrate boundaries [4]. Numerical integration of the Green's function requires the contacts in the substrate to be discretized. However, since this discretization is over the contact surface, and not over the whole substrate volume, this method is computationally more efficient, and thus widely used. One example is Substrate Coupling Analysis (SCA) tool by Cadence. In some of the simulations shown in the later chapters, SCA [19] is used as the tool to extract the resistive network, due to its convenience.

3.1.3 SPICE substrate model

Referring to Figure 3.1, the resistances R_{02}, R_{12} (resistances between the bulk node and substrate, R_{bulk}), R_{03}, R_{13} (resistances between substrate contact and the substrate, R_{bulk}) can be calculated using the formula [10].

$$R_{bulk} = R_{area} \parallel R_{perimeter} \quad (3.1)$$

$$R_{area} = \rho \times t / Area \quad (3.2)$$

$$R_{perimeter} = \rho / Perimeter \quad (3.3)$$

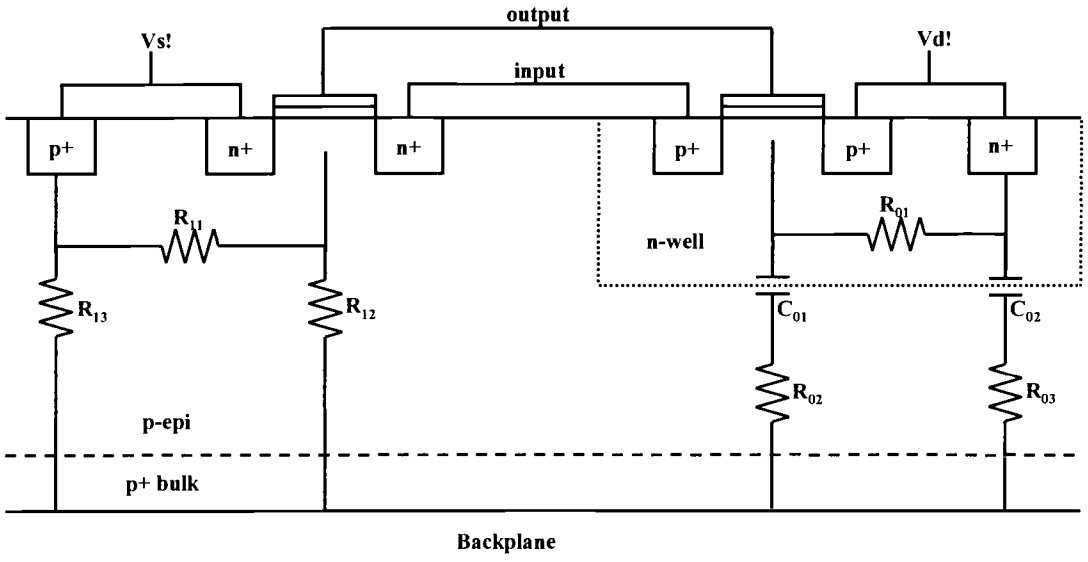


Figure 3.1 SPICE substrate elements for a CMOS inverter.

where

ρ = resistivity of the epi-layer.

t = thickness of the epi-layer.

The area and perimeter of the contacts can be calculated from the geometry of the substrate contact or the gate geometry. The lateral resistance R_{01}, R_{11}, between the substrate contact and the bulk node can be calculated using the formula

$$R = R_{sheet} * L / W \quad (3.4)$$

where

L= the length between the two contacts (Figure 3.2).

W= the mean width of the contacts.

R_{sheet} = resistance of a square piece of epi-layer.

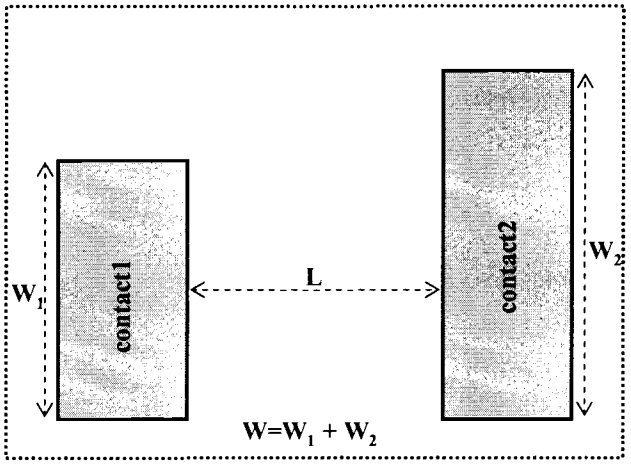


Figure 3.2 Top view showing the length and mean width of contacts used to calculate the lateral resistance in Eq. (3.4).

The capacitors C_{01} and C_{02} are the junction capacitances between the n-well and the p-substrate. One way of calculating the values of the capacitors C_{01} and C_{02} is by dividing the total n-well capacitance in a weighted manner (depending on the size of the contacts) among all the substrate contacts and the area under the gate in the n-well.

The SPICE substrate model can be used only for heavily doped substrates as the resistances that are calculated depend on the epi-layer characteristics.

3.1.4 Scalable macro model

This model, as described in [2] and [9] is scalable with contact dimensions and their spacing from one contact to another. First, a z-parameter matrix of the substrate network, depending on the contact dimensions, distance between contacts and the process parameters is obtained. The parameters are calculated using

$$Z_{11} = \frac{1}{K_1 Area + K_2 Perimeter + K_3} \tag{3.5}$$

$$Z_{12} = \alpha e^{-\beta x} \tag{3.6}$$

where K_1, K_2, K_3, α and β are process dependent parameters and x is the distance between the contacts. From the z-parameter matrix, lateral resistances and the resistances to the back plane can be obtained.

3.2 Reasons to separate supply noise from switching noise

As was described in Chapter 2, switching noise in the substrate appears due to the junction capacitors of the digital circuit, which in turn depend on the size of the transistors. Therefore, the amount of switching noise is dependent on the sizes of transistors in the digital circuit and also due to imperfections in the ground connection to the substrate. On the other hand, the amount of supply noise depends on the impedance in the supply lines and the impedance between the substrate contact and the substrate itself. This depends on the particular layout of the digital circuit, how the inputs and outputs are routed to the pins and the package. In order to optimize the digital circuit design for minimum substrate noise, it is desirable to isolate the supply noise and switching noise in simulations and then use noise suppression techniques

specific for switching and supply noise to suppress the total noise in the substrate. Some of the noise suppression techniques will be discussed in Chapter 5.

3.3 Simulation approach to isolate supply and switching noise

To simulate the total noise in the substrate, the set up shown in Figure 3.3 is used. The junction capacitors from the source and drain to the bulk node (C_{j1} through C_{j4}), which are a part of the MOSFET model, are shown explicitly in this figure. Resistors R_{01} through R_{13} and capacitors C_{01}, C_{02} constitute the substrate coupling network. In the setup shown, both the switching and supply noise sources get coupled to the substrate. The switching noise reaches the backplane through the junction capacitances and then through the substrate coupling network. The supply noise at the source terminals of the transistors reaches the backplane through the substrate coupling network. Thus, the total noise observed at the backplane will be the sum of the switching noise and the supply noise. Modifications to this set up will help suppress either the switching or the supply noise at the backplane.

The setup to simulate the supply noise is shown in Figure 3.4. Since the switching noise is due to an imperfect ground (vdd) at the bulk nodes (points A and B in this figure), it can be suppressed by providing an ideal ground (vdd) at the bulk nodes. If points A and B in Figure 3.4 are connected to ideal power supplies, then the noise due to the switching output node cannot couple to the substrate. The supply noise still reaches the backplane from the source nodes via the substrate coupling network.

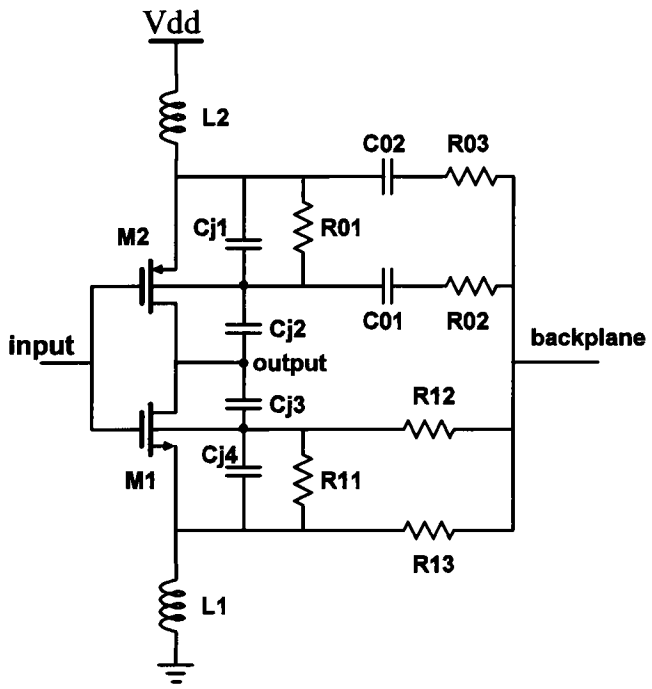


Figure 3.3 Setup to measure the total noise at the substrate.

Figure 3.3 can be modified to block the supply noise from reaching the substrate. The setup to simulate switching noise only at the substrate is shown in Figure 3.5. If the path through which the supply noise reaches the substrate is disconnected from the supply noise source, then the only noise that reaches the substrate is the switching noise. This is done by isolating the substrate contacts (points C and D in Figure 3.5) from the source supplies and connecting them to ideal supplies. There will be some coupling from the source nodes to the substrate through the junction capacitors C_{j1} and C_{j4}, but this coupling will be negligible for low supply inductance values (< 0.5nH)*.

* Further discussion about this is included in Section 4.3.3.

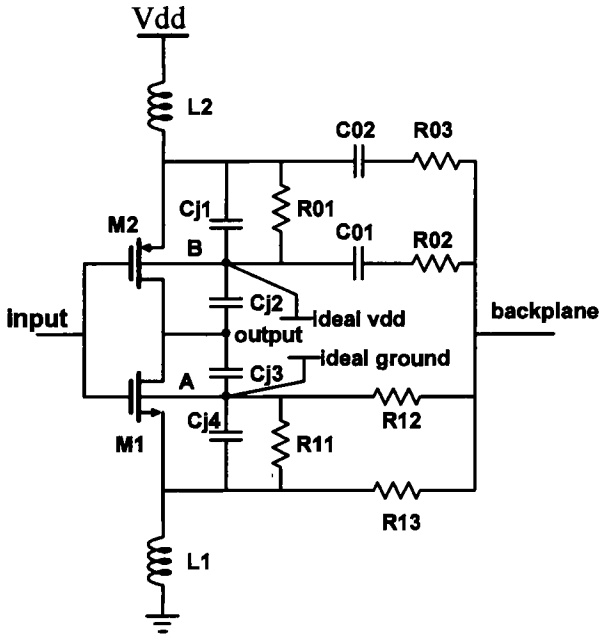


Figure 3.4 Setup used to measure supply-only noise at the substrate.

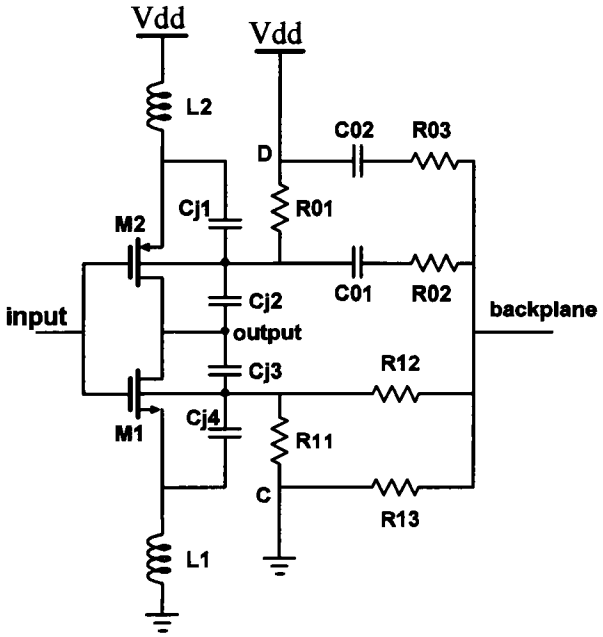


Figure 3.5 Setup used to measure switching-only noise at the substrate.

Using these three different setups, for any given digital circuit, supply noise and switching noise can be simulated and analyzed separately. The simulation approach discussed in this section is applied to a stepped buffer. The stepped buffer consists of seven stages of inverters, with each successive stage being a power of e[†] larger than the previous. Every stage but the last is loaded with another inverter, as shown in Figure 3.6. The sizes of the first inverter transistors are $(W/L)_n$ = 5μm/0.6μm and $(W/L)_p$ = 10μm/0.6μm.

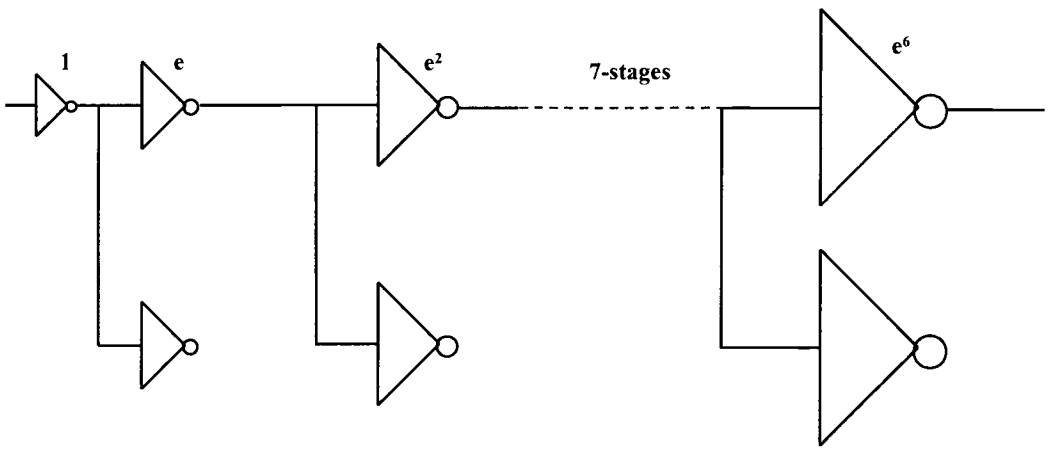


Figure 3.6 Block diagram of a stepped buffer.

The resistive network for the stepped buffer is extracted using SCA. A supply inductance of 5nH is added in the power and ground lines. The supply and switching noise generated by the stepped buffer are shown in Figure 3.7. The peak-to-peak value of the supply noise seen in the figure is three times larger than the peak-to-peak

[†] e=2.718

value of the switching noise. The dominance of the supply noise over the switching noise is due to the presence of the large supply inductance. The effect of supply inductance on substrate noise will be described in detail in Chapter 4. From Figure 3.7, it can be seen that there is less ringing in supply noise when compared to the ringing in switching noise. This is because; the junction capacitors C_{j1} through C_{j4} are connected to an ideal ground in the setup for the supply noise (Figure 3.4), when compared to a non-ideal ground in the setup for the switching noise (Figure 3.5).

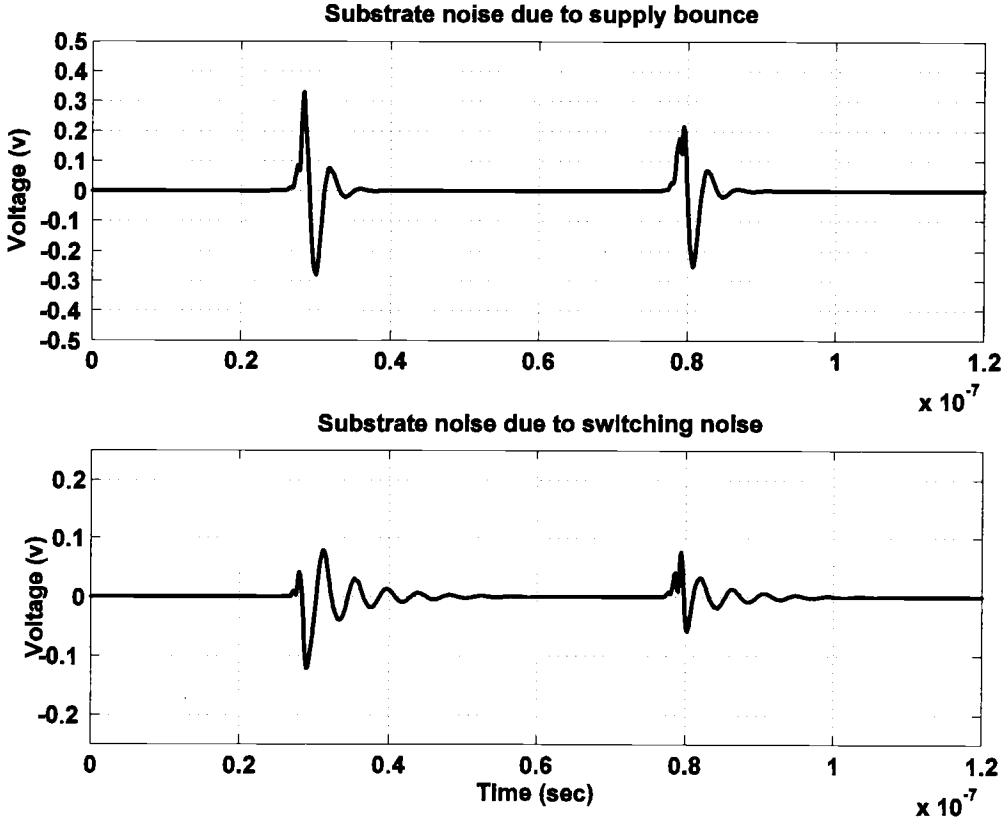


Figure 3.7 Plots of supply noise (top) and switching noise (bottom) isolated for a stepped buffer.

The simulation approach to isolate the switching and supply noise components of substrate noise is used in Chapter 4 to simulate the effect of supply inductance on the substrate noise components. Based on the simulation results, appropriate noise suppression techniques can be used to suppress the dominant source of substrate noise, during the digital circuit design. Various noise suppression techniques are discussed in Chapter 5.

4. EFFECT OF PIN PARASITICS ON SUBSTRATE NOISE

Even though the speed of the integrated circuits has steadily increased owing to device scaling and circuit innovations, the performance of packages has not improved significantly. As a result, packaging continues to limit the achievable performance of today's high performance ICs. Hence, package parasitics should be taken into account in the design of integrated circuits. Simulations should include a reasonable circuit model of a package, without which substrate noise generated by digital circuits will be greatly underestimated. This will be shown in the following sections.

4.1 General effects of package parasitics

A package parasitic model consists of self inductance, mutual inductance, self capacitance, mutual capacitance and resistance. The effect of each of these parasitics is discussed in this section:

4.1.1 Self inductance

The package trace of a bond wire exhibits a finite self inductance. Depending on the length of the trace and the type of the package, the value of the inductor can range from 1nH to 20nH. The effect of supply inductance on substrate noise is described with the example of an inverter, shown in Figure 4.1. When the output is transitioning from one logic level to another with a transition time equal to dt, the current flowing through the load capacitance will be $C(dV_l/dt)$, where dV_l is the

voltage difference of the transition, shown in the figure. This current will be drawn from either the positive power supply or sunk into ground, depending on the transition (high to low or low to high). The current in the power supply flows through the inductor, causing a change in the voltage across the inductor, called supply bounce. The whole effect is called the *L(di/dt)* effect, as the change in voltage across the inductor is equal to *L(di/dt)*.

The inductance of the pin parasitics also affects the input signals. The inductor along with any parasitic capacitors present at the input of any circuit forms a low pass filter and attenuates high frequency components and causes ringing in transient waveforms. Inductance in the input and output signal paths could also cause a phase change of the signals, which might be critical in certain analog circuits [14].

4.1.2 Mutual inductance

Mutual inductance between two pins in a package couples noise present on one pin to an adjacent sensitive pin. Even if analog power supplies are separated from digital supplies, there will still be some noise coupled to the analog power lines due to the mutual inductance if the analog and digital supplies are connected to adjacent pins. This coupling can be reduced through proper design of the pad frame and good positioning of the pads. Mutual inductance also manifests itself in parallel bond wires used to lower the overall self inductance of a connection. If two wires are connected in parallel, the total inductance will be $(L_s+M)/2$, instead of $L_s/2$, where M is the mutual inductance between the two wires and L_s is the self inductance.

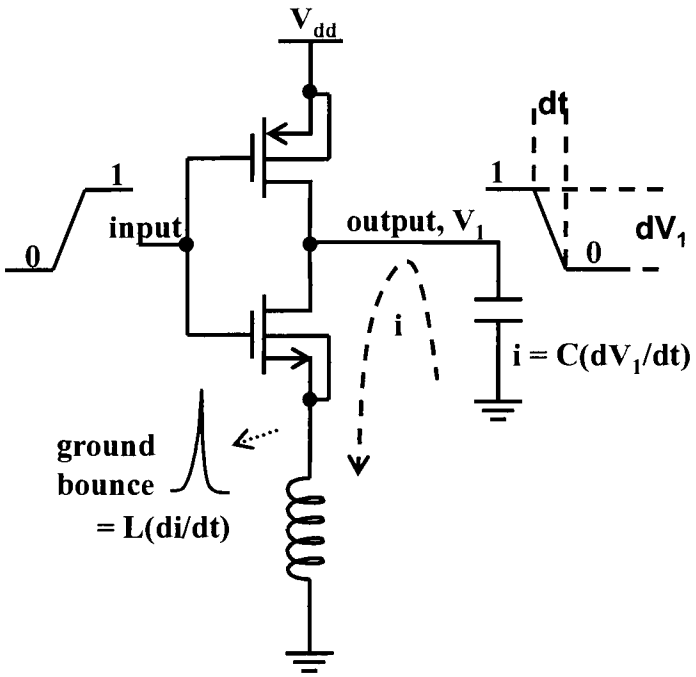


Figure 4.1 Schematic showing the *Ldi/dt* effect of an inductor in supply lines.

4.1.3 Self and mutual capacitance

The capacitance of a trace, along with its inductance gives rise to a finite resonance frequency, which can be stimulated by various transient currents drawn by a circuit. Moreover, a capacitance present in the supply lines, along with any supply inductance causes ringing in the power supply, which gets coupled to the substrate.

4.2 Types of packages

There are a wide variety of packages available in the present day market. They include high electrical performance packages such as flip-chip packages, which are

expensive, and also there are comparatively low performance packages like DIPs, PGAs, BGAs, and QFPs. Packages can be broadly classified in two ways:

(i) depending on their mounting methods – through hole or surface mount.

(ii) depending on the type of packaging - ceramic or plastic.

Conventionally, there are two types of mounting methods [15] as described in the following subsections.

4.2.1 Through hole mounting

This method requires packages which have leads that rely on holes in a multilayer PCB for the purpose of mounting, shown in Figure 4.2. The extended leads are then soldered for permanent mounting. Some of the packaging technologies which use this mounting method are: Dual In Line (DIP) and Pin Grid Arrays PGA).

4.2.2 Surface mounting

This type of mounting addresses the need for increased IC pin count and the requirement for smaller weight and size of the packaged circuits. This packaging family, called surface-mounted technology (SMT), consists of soldering the leads on the surface of the PCB instead of utilizing through-hole mounting, and is shown in Figure 4.2. This type of mounting also provides better electrical performance over through hole mounting. Some examples are SOICs, Chip Carrier Packages (LCCs etc), BGAs etc.

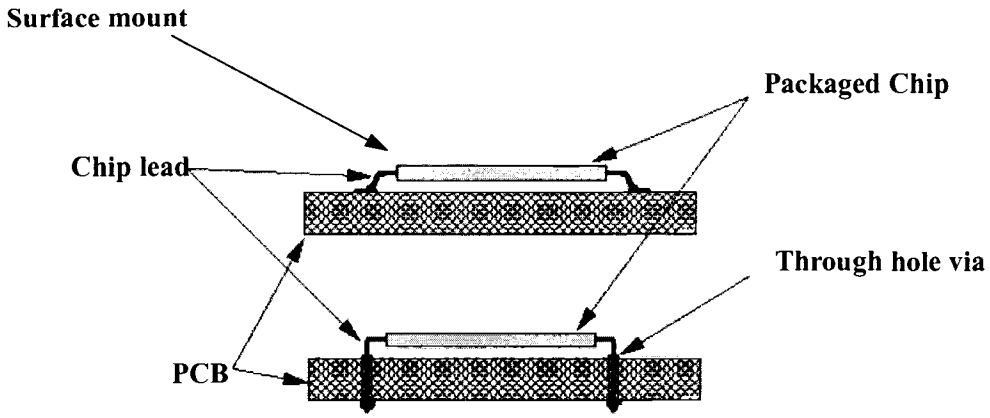


Figure 4.2 A general schematic showing the difference between surface mounting (top) and through hole (bottom) technology [15].

There can be ceramic or plastic packages in both types of mounting methods. The range of pin parasitics for each of the packages described is tabulated in Table 4.1.

Table 4.1 Range of pin parasitics for different packages [18], [20].

Type	No of pins	Self Inductance (nH)	Mutual Inductance (nH)	Capactiance (pF)
DIP	40	4 to 9	2 to 10	0.6 to 5
PGA	108	3 to 7	-	1.5 to 3
SOIC	44	4 to 12	2 to 5	0.1 to 0.2
BGA	225	7.5 to 14.5	4 to 7	0.1 to 0.2
LPCC	32	1.1 to 1.47	0.17 to0.23	.08 to 0.11
Flip chip	64	0.26 to 1.5	-	0.18 to 0.38

Some of the packages and their electrical characteristics used in simulations are described below. The mutual inductance between two pins is not considered for the electrical model used in the simulations, because the effect of mutual inductance on substrate noise can be reduced by connecting pins adjacent to a noisy supply line to ground.

PGA package:

The electrical characteristics of the PGA package can be seen in Figure 4.3 [20]. PGAs are usually used where a high pin count in the range of 100 to 400 is required. The range of values listed in Table 4.1 for the PGA are for R_1, L_1 and C_1. The range of values for R_2, L_2 and C_2 for a 108 pin PGA package are 0.04Ω to 0.29Ω, 0.69nH to 4.65nH and 0.19pF to 1.3pF, respectively.

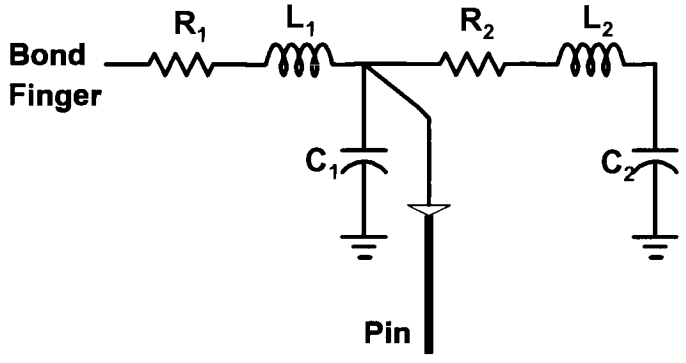


Figure 4.3 Electrical characteristics for a PGA package.

SOIC Package:

A typical SOIC package trace model is shown in Figure 4.4 [17], [18]. The range of values for the L, C and R depend on the number of pins in the package. SOIC s are used for pin counts ranging from 10 to 50.

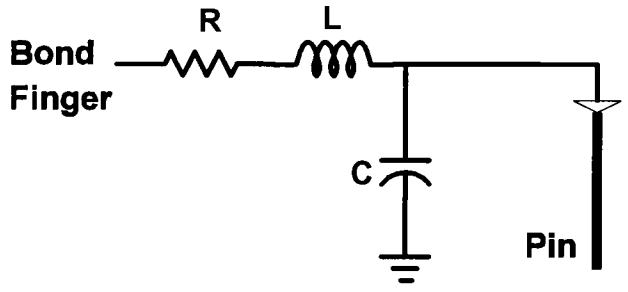


Figure 4.4 Electrical characteristics for an SOIC package.

LPCC Package:

Typical values of the parasitics L, C and R_1 are listed in Table 4.1. Values of R_2 range in the order of mega ohms (6MΩ to 10MΩ) [23].

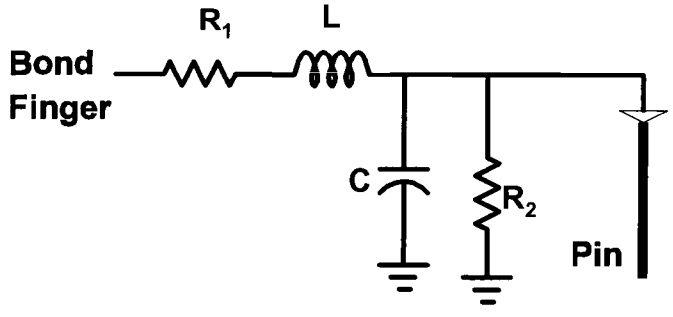


Figure 4.5 Electrical characteristics of an LPCC package.

4.2.3 Flip-chip packages

Flip-chip (FC) [16] is not a specific package (like SOIC), or a package type (like BGA). Flip-chip describes the method of electrically connecting the die to the package carrier. The package carrier then provides the connection from the die to the exterior of the package. In standard packaging, the interconnection between the die and the carrier is made using bond wires. In contrast, the interconnection between the die and carrier in flip-chip packaging is made through a conductive "bump" that is placed directly on the die surface. The bumped die is then "flipped over" and placed face down, with the bumps connecting to the carrier directly. This type of packaging reduces the inductance significantly because the length of the interconnect is greatly reduced, thus giving better electrical performance.

4.3 Simulations of substrate noise generated by stepped buffer

The effect of pin parasitics on the substrate noise will be presented by some simple simulations. First, it will be shown how the substrate noise will be underestimated if the pin parasitics are not included in substrate noise simulations. Then, it will be shown that of all the parasitics, inductance has the most significant effect on substrate noise. For the simulations, a digital stepped buffer is taken as a representative circuit for a digital block.

4.3.1 Digital circuit

Stepped buffers are used in many digital designs to provide buffering for signals. The stepped buffer is used as a digital circuit in simulations and

measurements so that it can generate adequate amount of measurable noise in the substrate. The buffer has seven stages of inverters with each successive stage being a factor of *e* larger than the previous, for delay optimization [24] (Figure 4.6). The sizes of the first inverter transistors are $(W/L)_n = 5\mu m/0.6\mu m$ and $(W/L)_p = 10\mu m/0.6\mu m$. Additionally, every stage but the last is loaded with another inverter to provide more substrate noise. The schematic of the stepped buffer is shown in Figure 4.7 and the layout is shown in Appendix C. The supplies to the sources and bulks of the transistors are separated and connected to different pins so that supply noise in the substrate can be reduced. The stepped buffer can be used for input frequencies up to 15 MHz.

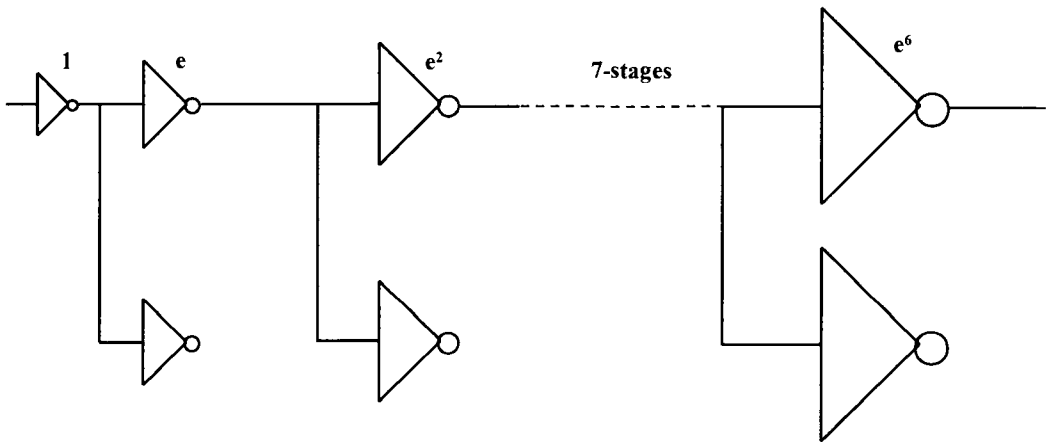


Figure 4.6 Block diagram of a stepped buffer.

The stepped buffer is simulated for substrate noise, with no pin parasitics. Ideal power supplies are connected to the sources and substrate contacts of the transistors. The input is at a frequency of 780 kHz. The resistive network for NMOS

and PMOS transistors is obtained from SCA. The transient waveform of the substrate noise picked up by a sensor (a p+-contact shown in the layout of the stepped buffer in Appendix C) placed 5μm away from the stepped buffer is shown in Figure 4.8 (top).

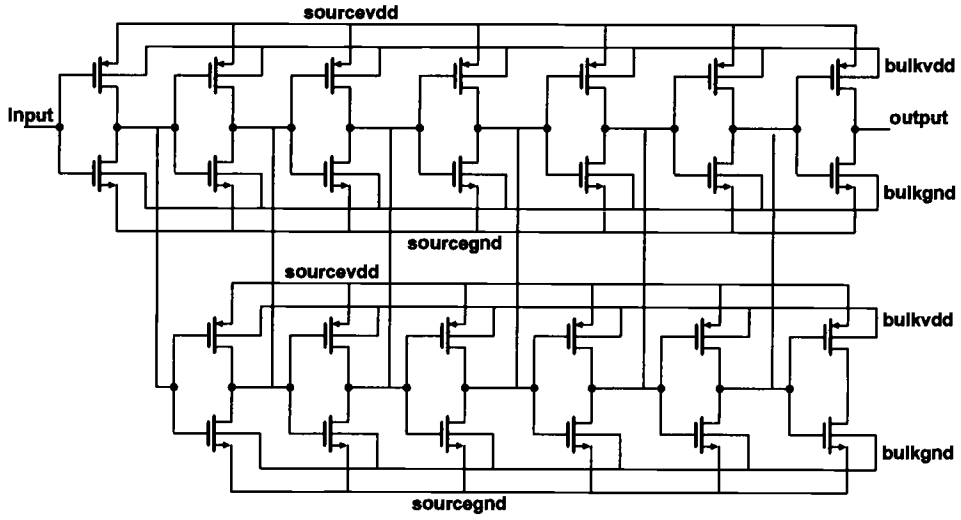


Figure 4.7 Schematic of the stepped buffer.

The stepped buffer is also simulated with pin parasitics. The schematic of a single stage of the stepped buffer with all the pin parasitics from a PGA108 package included in the supply lines and the substrate network added is shown in Figure 4.10. The values of the parasitics are chosen from Table 4.1, by taking the average from the given range of values. It can be seen in the figure that the source and bulk nodes of the stepped buffer are isolated. The transient waveform of the noise picked up by the sensor when pin parasitics from a PGA108 package are included is shown in Figure 4.8 (bottom). If the peak-to-peak values of the substrate noise in both the cases are

compared (Figure 4.9 (a)), the noise picked up by the sensor is more by a factor of 3 when the pin parasitics are included.

The noise picked up by a sensor placed at 5μm away from the stepped buffer when package parasitics from flip-chip, LPCC, SOIC, BGA and DIP packages are included is shown in Figure 4.9. The values of the parasitics for these packages are chosen from Table 4.1, by taking the average from the given range of values. When the rms[‡] value of noise picked up by the sensor in all the cases is compared, it can be seen that the noise varies from 3mV, when no package model is included, to 24mV when the package model from BGA is included, which is an increase by a factor of 8. When package models from flip-chip and LPCC packages are included, the noise picked up by the sensor is a factor of 3 lower than the noise generated from a BGA package model. This is due to the low value of supply inductances for the flip-chip and LPCC packages (0.88nH to 1.285nH), compared to the BGA package (11nH). As packages with high values of supply inductance, like PGA, DIP and SOIC are included, the noise picked up by the sensor also increases, as shown in Figure 4.9.

[‡] rms values are calculated using built-in Cadence functions.

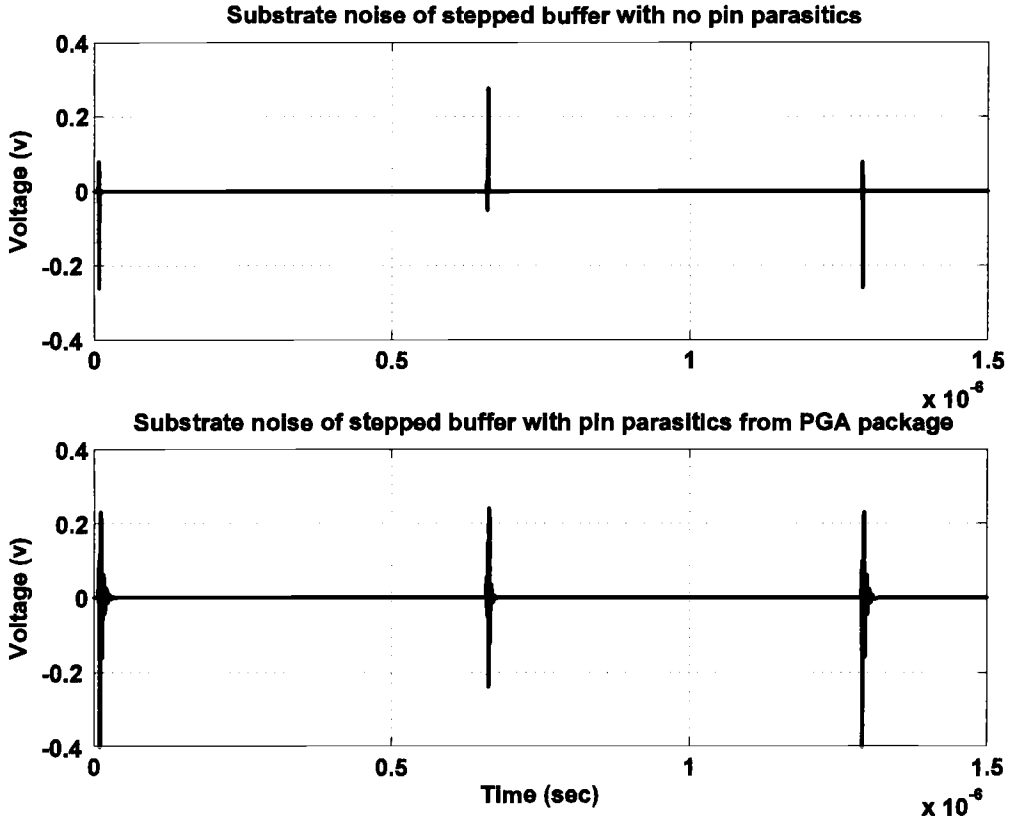
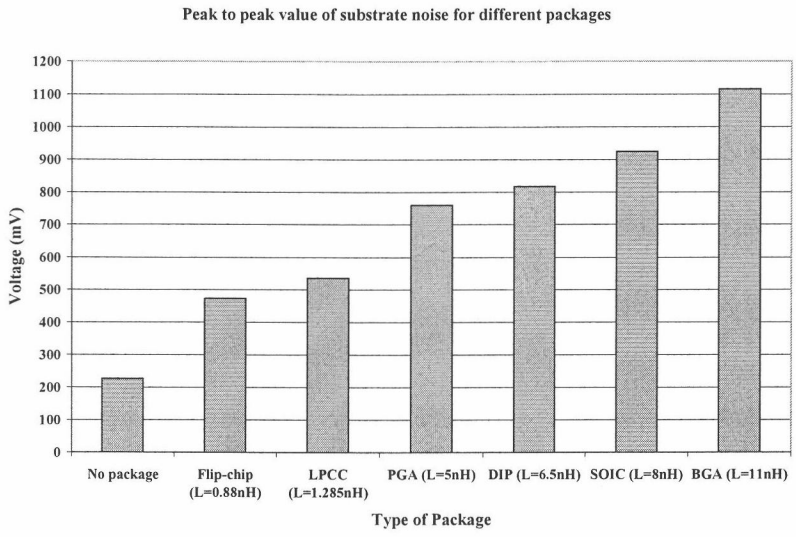
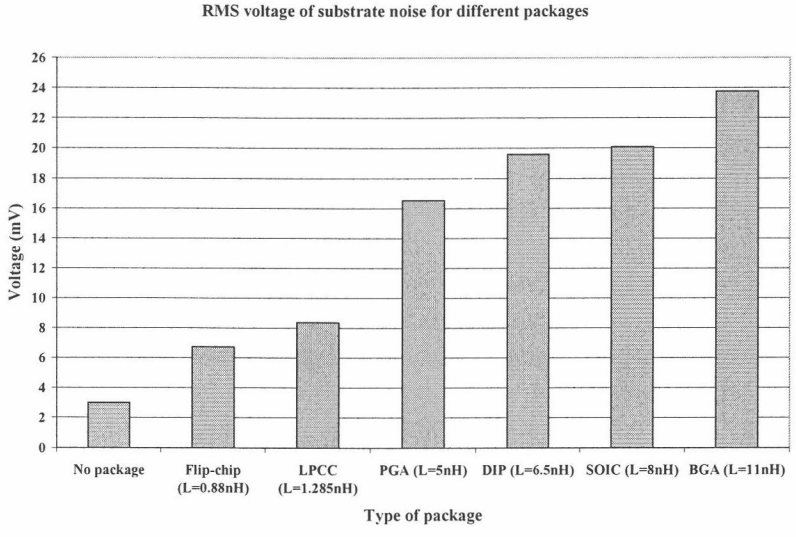


Figure 4.8 Simulation showing substrate noise picked up by a sensor placed 5μm away from the stepped buffer without (top) and with (bottom) pin parasitics included in the simulations. The input frequency of the stepped buffer is 780 kHz and the source and bulk nodes of the transistors in the buffer are separated.



(a)



(b)

Figure 4.9 Graphs showing (a) peak to peak and (b) rms values of noise picked up by a sensor placed 5μm away from the stepped buffer for different package parasitics. The input frequency of the stepped buffer is 780 kHz and the source and bulk nodes of the transistors in the stepped buffer are separated.

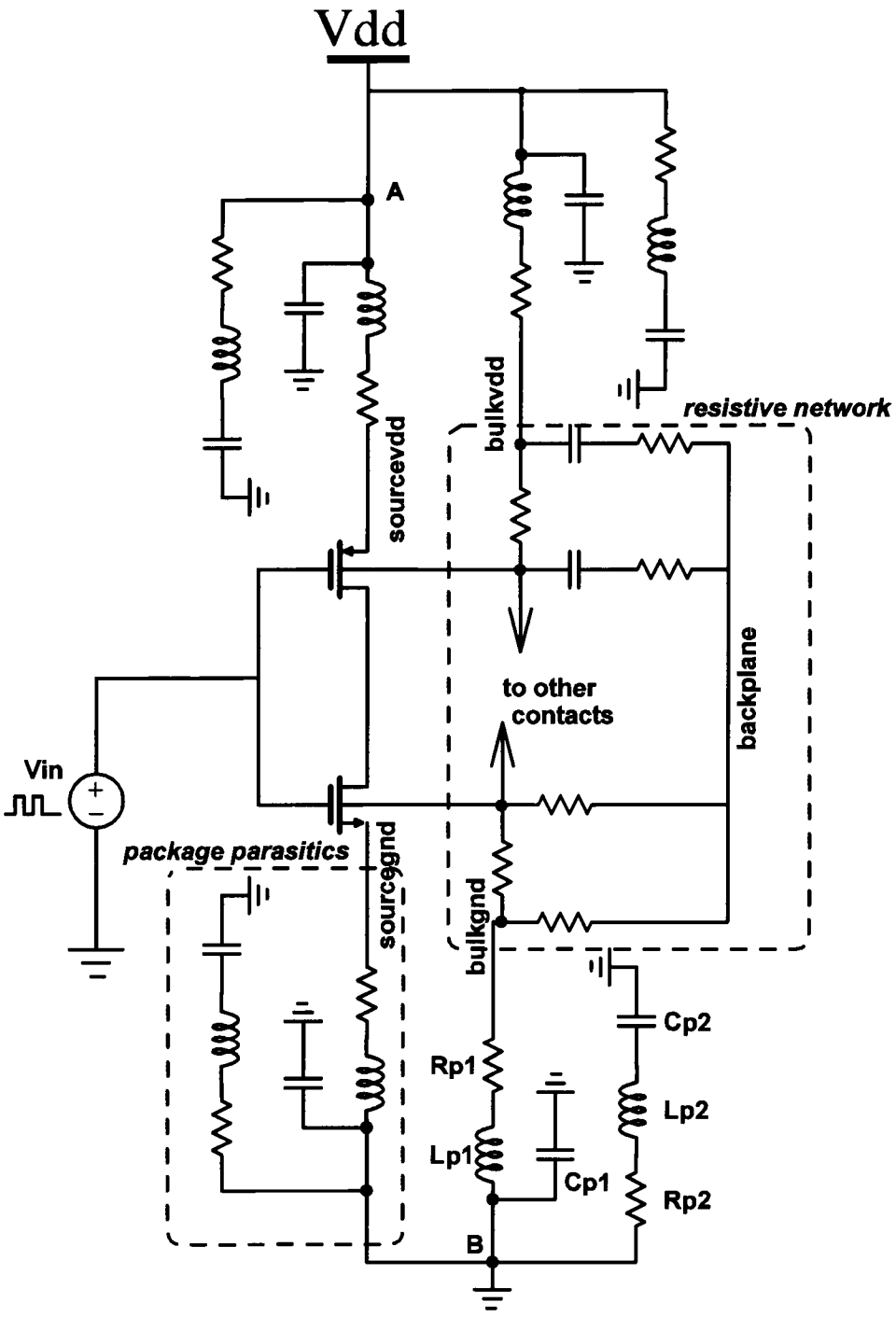


Figure 4.10 One stage of the stepped buffer showing the substrate network and the pin parasitics of a PGA package.

4.3.2 Dominance of inductance in pin parasitics

Even though pin parasitics include capacitors and resistors, supply inductors dominate the total parasitic effect in most cases. This is because of the *Ldi/dt* effect described in Section 4.1.1. The dominance of inductors is shown by the following analysis. Taking typical values of inductance (L_{p1}) and capacitance (C_{p1}) of the pin parasitics for a PGA121 package, for the schematic shown in Figure 4.10 and then varying the value of the resistor (R_{p1}) in the available range, the peak to peak value of the substrate noise picked up by a sensor placed 5μm away from the stepped buffer is observed. L_{p2}, R_{p2} and C_{p2} are kept constant. Their effect on substrate noise can be neglected since they are connected between ideal voltage sources and ground. Inductor L_{p1} is kept constant at 2nH, capacitor C_{p1} at 3pF and the value of the resistor is varied from 100mΩ to 400mΩ. The peak to peak and rms values of the noise picked up by the sensor are tabulated in Table 4.2 for an input signal frequency for the buffer of 780 kHz. From the table, it can be seen that as the resistor value is varied, there is at most a 7% variation in the noise picked up by the sensor.

In a similar way, keeping R_{p1} and L_{p1} constant at 200mΩ and 2nH respectively, C_{p1} is varied from 1pF to 5pF. From Figure 4.10, it can be seen that one end of the capacitor is connected to the voltage sources providing vdd/gnd (points A and B in Figure 4.10) and the other end is connected to ground. So, ideally, the capacitor does not have any affect on the substrate noise. But in reality, there will be some finite impedance between these voltage sources and the points they are connected to the pins (nodes A and B in Figure 4.10). This impedance depends on the

measurement setup. For a wire inductance of 0.01nH, from Table 4.3, it can be seen that the noise picked up by the sensor varies by only about 3% as C_{pl} is varied from 1pF to 5pF.

Now, setting R_{pl} to 200mΩ, and C_{pl} to 3pF, the value of L_{pl} is varied from 1nH to 10nH and the variation in the noise picked up by the sensor is observed. Table 4.4 shows that as the inductor is varied, there is a 41% variation in the substrate noise. Moreover, as the value of inductance increases, ringing in the substrate noise also increases. If only the peak-to-peak values of the substrate noise are considered, this effect is not observed. The effect of ringing can be observed by considering the rms value of the noise picked up by the sensor. From Table 4.4, it can be shown that the rms value of noise picked up by the sensor varies by more than 100% as the inductor value is varied in the given range. Hence, based on the tabulated data, it can be shown that the effect of inductance dominates over the resistance and capacitance in the package parasitics.

Table 4.2 Substrate noise picked up by a sensor placed 5μm away from the stepped buffer as pin resistance is varied, for L=2nH, C=3pF.

R (mΩ)	100	200	300	400
peak-to-peak voltage (mV)	754.3	735.4	716.5	704.3
rms voltage (mV)	12.79	12.53	12.28	12.04

Table 4.3 Substrate noise picked up by a sensor placed 5μm away from the stepped buffer as pin capacitance is varied, for R=200mΩ, L=2nH.

C (pF)	1	2	4	5
peak-to-peak voltage (mV)	745.4	738.9	731.1	727.8
rms voltage (mV)	12.59	12.57	12.48	12.43

Table 4.4 Substrate noise picked up by a sensor placed 5μm away from the stepped buffer pin inductance is varied, for R=200mΩ, C=3pF.

L (nH)	1	3	6	10
peak-to-peak voltage (mV)	678.5	801.3	890.4	958
rms voltage (mV)	9.712	14.49	18.34	22.5

4.3.3 Effect of inductance on supply noise

In the previous section, it was shown that inductor dominates over the other pin parasitics when considering substrate noise. In this section, it will be shown that the inductor affects the supply noise to a greater extent than the switching noise.

Switching noise, as seen in Section 2.1, is due to the junction capacitances of the transistors and due to an imperfect ground at the bulk nodes of the transistors. Since the non-ideal ground at the bulk nodes of the transistors is not influenced by the inductor in the power supply to a large extent, switching noise in the substrate is not affected by the pin inductors in the power supply. A change in the value of the inductor has a significant effect on the supply noise. As the inductance increases, *L(di/dt)* increases, increasing the bounce on the supply lines and also increasing the ringing in the supply noise. Figure 4.11 shows the effect of supply inductance on the

switching noise and supply noise generated by the stepped buffer. This figure is obtained by simulating a stepped buffer at a frequency of 10 MHz, with package parasitics added in the supply lines and using the simulation approach described in Section 3.3. The variation in the noise picked up by a sensor placed 5μm away from stepped buffer is observed as the supply inductance is varied. The layout of the stepped buffer and the sensor (a p+-contact) are shown in Appendix C. The simulation flow is shown in Appendix A.

When the supply inductance is small, supply noise is less than the switching noise, because the *L(di/dt)* effect is small. It can be seen from Figure 4.11 that for small values of supply inductance (< 10pH), the supply noise does not decrease as the inductance is decreased, but it remains constant. For small values of inductance, supply noise due to the resistance present in the supply lines dominates over the L(di/dt) effect. Hence supply noise remains constant. As the inductor value is increased, supply noise also increases and after a certain value of the inductance, the supply noise becomes more than the switching noise, thus becoming the major contributor to substrate noise. Switching noise remains almost the same as the inductance is varied. At high inductance values, there is a small increase in the switching noise, which is due to the increased coupling through the junction capacitors C_{j1} and C_{j4} shown in Figure 3.5.

The value of the supply inductor for which the supply noise becomes dominant over switching noise for a stepped buffer is 0.15nH. For the stepped buffer, it is observed that the input frequency does not affect the inductor value at which the

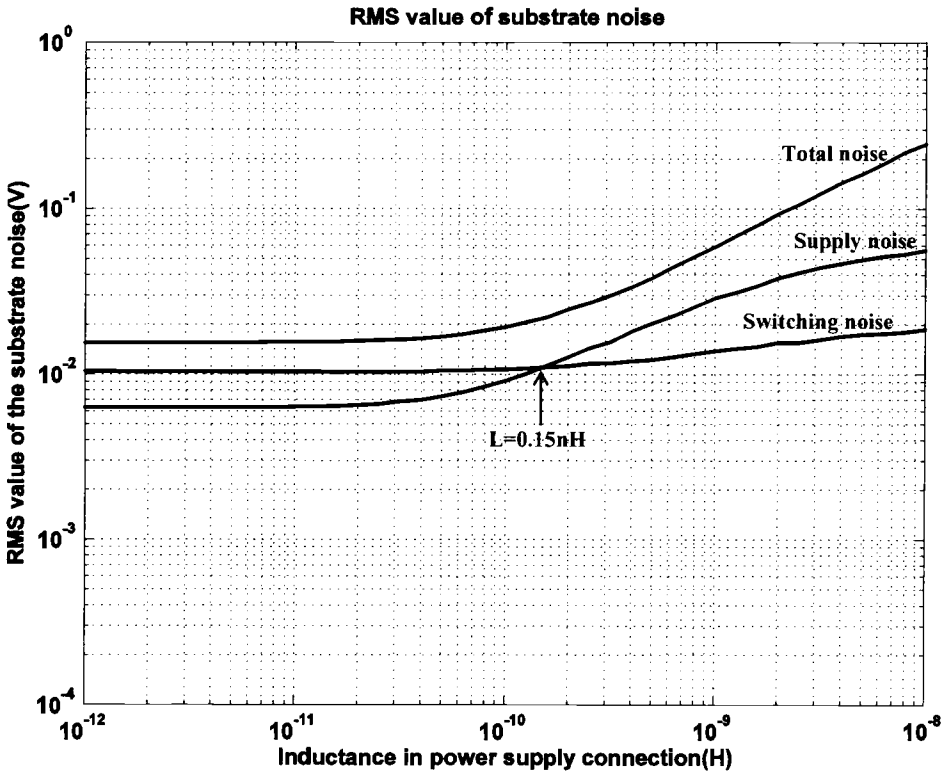


Figure 4.11 The effect of supply inductance on switching and supply noise generated by the stepped buffer, which is picked up by a sensor placed 5μm away from the stepped buffer. The input frequency of the stepped buffer is 10 MHz. For small inductance values switching noise is dominant and for large inductance values supply noise dominates.

supply noise dominates the switching noise (cross-over inductance). Substrate noise is generated due to the transitions at the outputs of the stepped buffer. For low supply inductor values, there will only be a small amount of ringing in the substrate noise and it settles before the next transition of the output occurs. When the substrate noise settles quickly, a change in the frequency of the input signal does not affect the amount of substrate noise generated in one cycle. This is shown in Figure 4.12 (a) and

(b). The stepped buffer is simulated for input frequencies of 100 kHz and 10 MHz, with a supply inductance of 0.15nH. It can be seen from the figures that since there is no ringing in the substrate noise, the noise remains the same with a change in input frequency.

As the supply inductance increases, the amount of ringing in the substrate noise increases. For high inductor values, the substrate noise may not settle before the next transition at the output occurs. This is shown in Figure 4.12 (c) and (d), when the stepped buffer is simulated for input frequencies of 100 kHz and 10 MHz, with a supply inductance of 10nH. In this case, a change in the input frequency changes the amount of substrate noise generated per cycle. For the stepped buffer, the cross-over inductance is 0.15nH. As seen in Figure 4.12 (a) and (b), this value of inductance does not cause ringing in the substrate noise, which implies that, when the input frequency changes, for supply inductor values less than 0.15nH, the substrate noise characteristics don't change. From the above discussion, it can be concluded that the cross-over inductance is not affected by the input frequency of the stepped buffer.

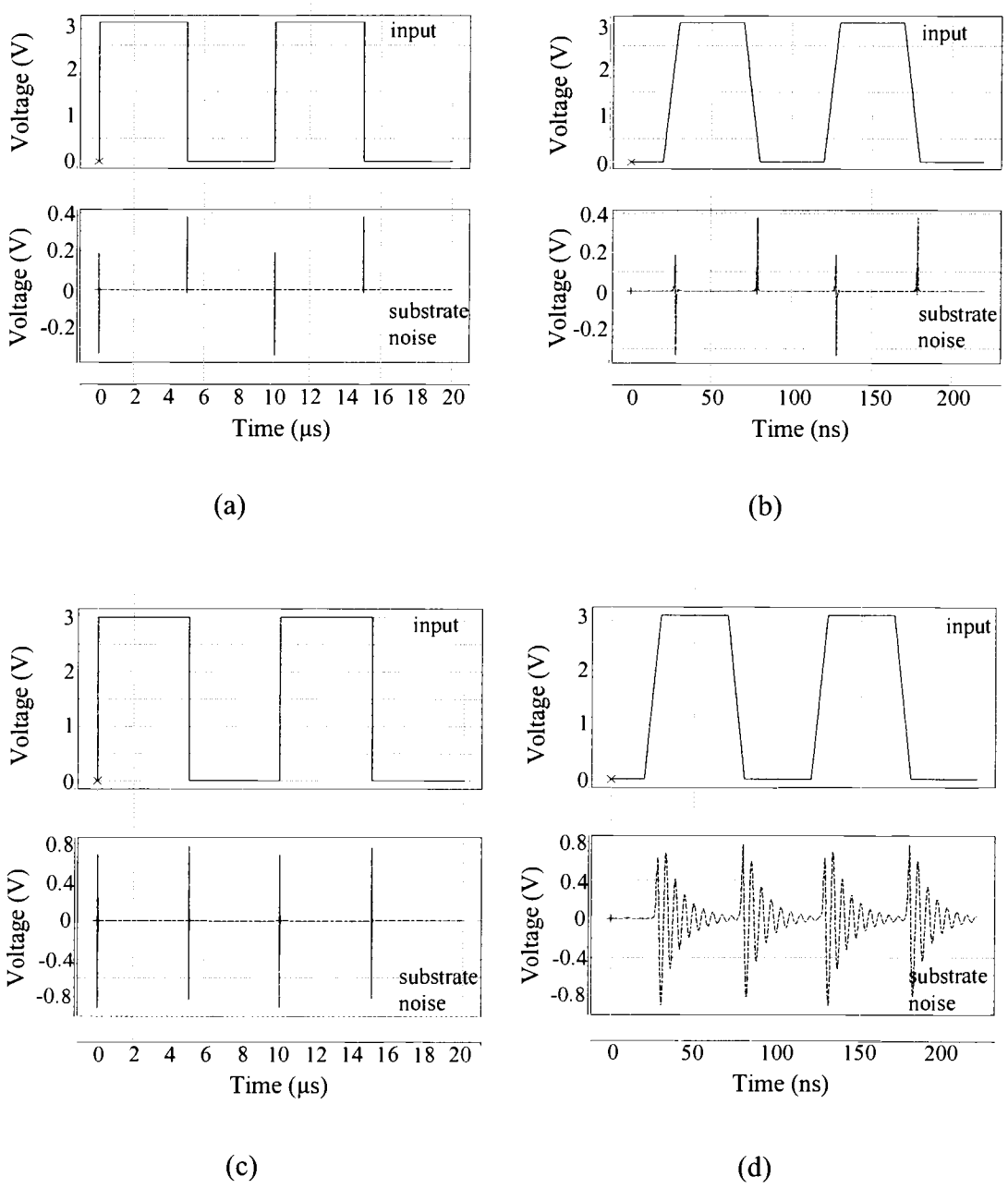


Figure 4.12 Transient waveforms of the input and total substrate noise generated by the stepped buffer with different supply inductances and input frequencies. (a) Supply inductance = 0.15nH, input frequency = 100 kHz. (b) Supply inductance = 0.15nH, input frequency = 10 MHz. (c) Supply inductance = 10nH, input frequency = 100 kHz. (d) Supply inductance = 10nH, input frequency = 10 MHz.

This result is shown in Figure 4.13[§]. The supply and switching noise generated by the stepped buffer at input frequencies of 100 kHz and 10 MHz and picked up by a sensor placed 5μm away from the stepped buffer are plotted and the cross-over inductance for both these frequencies is marked. It can be seen from the figure that the cross-over inductance remains the same with a change in input frequency.

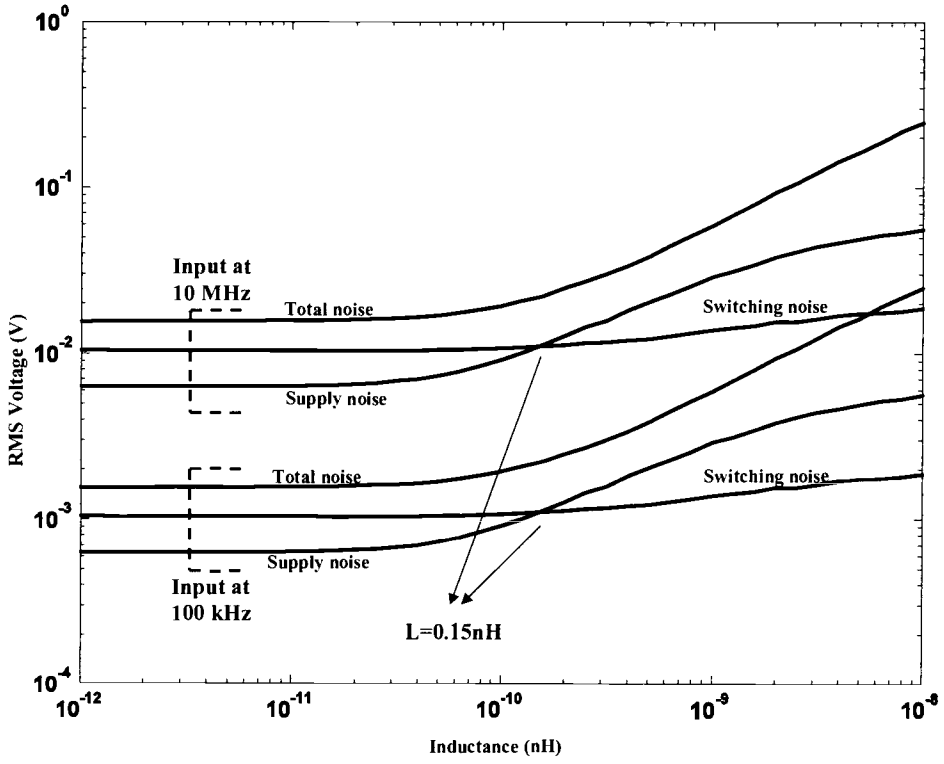


Figure 4.13 The effect of input frequency on the cross-over inductance for the stepped buffer. There is no change in the cross-over inductance as the input frequency is changed from 100 kHz to 10 MHz.

[§] Note that in Figure 4.13, the rms values of the substrate noise are different for different frequencies, even though the amount of substrate noise generated is the same. Since the time periods of different frequencies are different, and the rms value is calculated over one time period, the same amount of substrate noise is averaged over a greater interval for low frequencies. Hence the rms value of substrate noise is different for different input frequencies.

4.3.4 Effect of sizing and simultaneous switching on substrate noise

There can be many other factors that affect the amount of substrate noise generated, including the number of transistors in a digital circuit switching at the same time. As the number of transistors switching at the same time increases, the peak-to-peak substrate noise increases. This can be explained as a simultaneous excitation of the substrate due to various switching transients. The effect of large digital circuits can also be analyzed in the same way. Large digital circuits can be considered as many smaller circuits in parallel and switching simultaneously. So, substrate noise is affected in the same way as for transistors switching simultaneously.

Two different sizes of the stepped buffer are simulated and the noise picked up by a sensor placed 5μm away from the stepped buffer is plotted as a function of supply inductance in Figure 4.14. One stepped buffer is of size $(W/L)_{n1}$ = 1μm/0.6μm, $(W/L)_{p1}$ = 2μm/0.6μm (size of the first stage of inverter in the stepped buffer) (size 1x). The size of the second stepped buffer is increased by a factor of 5 (size 5x), compared to the smallest stepped buffer. As the size of the stepped buffer is increased from 1x to 5x, noise picked up by the sensor at low inductor values also increases by a factor of 5 from 3.3mV to 15.5mV. But the noise picked up by the sensor at high inductor values increases only by a factor of 2.2, from 0.11V to 0.24V. This is because, at high supply inductor values, supply noise dominates over switching noise, and supply noise is affected to a smaller extent by transistor sizing, compared to switching noise.

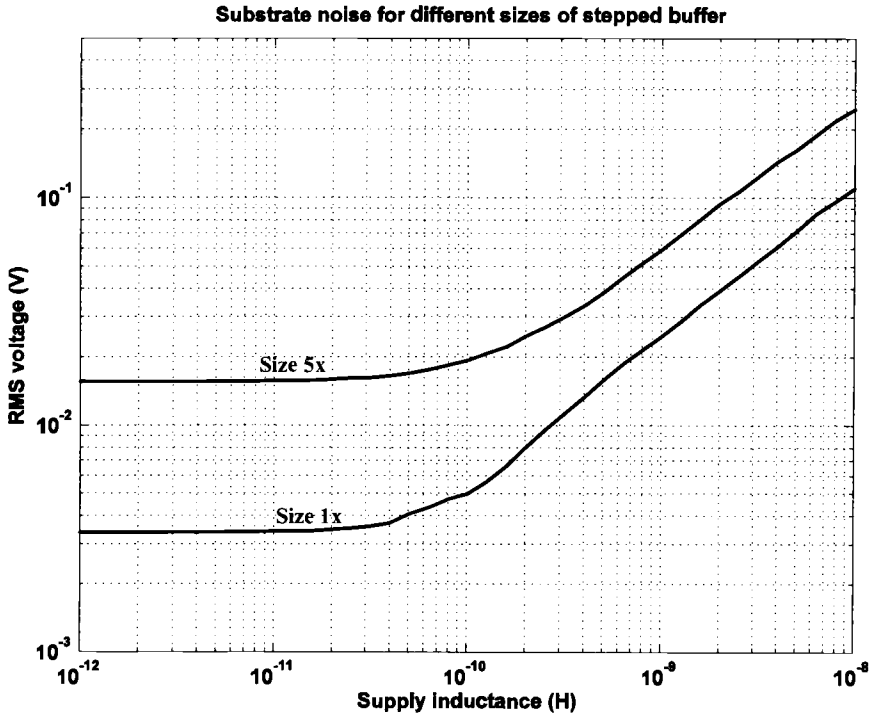


Figure 4.14 The effect of increasing the size of the stepped buffer on the substrate noise picked up by a sensor placed 5μm away from the buffer as a function of supply inductance. The substrate noise increases as the size of the buffer is increased. The input frequency of the stepped buffer is 10 MHz. 1x: $(W/L)_{nl}$ = 1μm/0.6μm, $(W/L)_{pl}$ = 2μm/0.6μm, 5x: $(W/L)_{nl}$ = 5μm/0.6μm, $(W/L)_{pl}$ = 10μm/0.6μm.

Since the switching and supply noises are affected to different extents by an increase in transistor sizing, the cross-over inductance is expected to change as transistor sizes are changed. Two stepped buffers of size 1x and 5x, respectively are simulated for the cross-over inductance at an input frequency of 10 MHz and the results are plotted in Figure 4.15. It can be seen from the figure that as the size of the

stepped buffer is increased, the cross-over inductance changes from 0.07nH to 0.15nH.

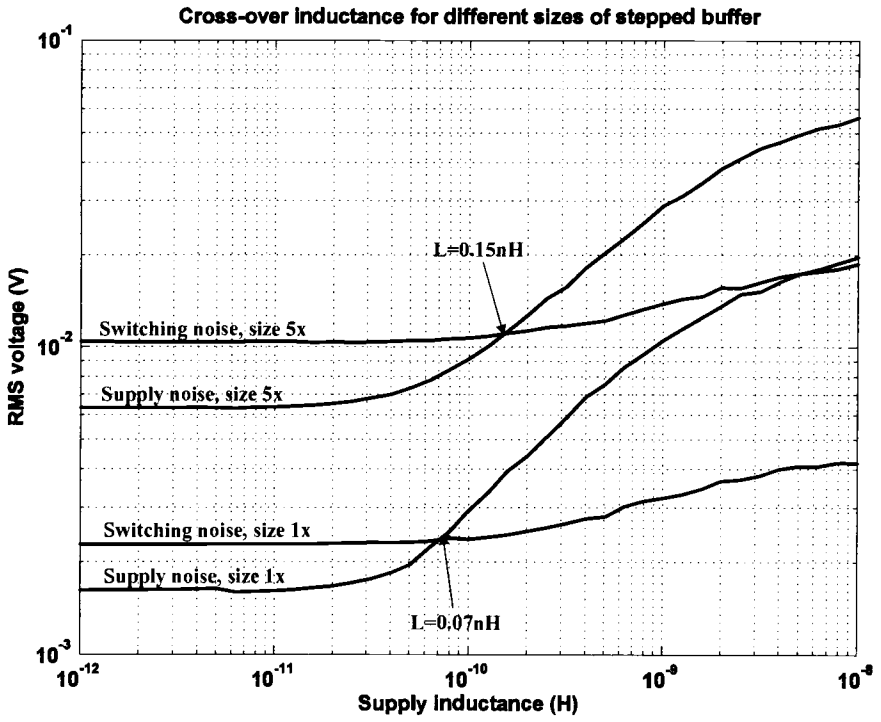


Figure 4.15 Effect of transistor sizing on the cross-over inductance. As the size of the stepped buffer is increased, the value of the cross-over inductance increases. The input frequency of the buffer is 10 MHz. 1x: $(W/L)_{n1}$ = 1μm/0.6μm, $(W/L)_{p1}$ = 2μm/0.6μm, 5x: $(W/L)_{n1}$ = 5μm/0.6μm, $(W/L)_{p1}$ = 10μm/0.6μm.

4.3.5 Effect of rise and fall time of input signal on substrate noise

As the rise and fall time of the input changes, its effect on the substrate noise generated by the stepped buffer is not significant. The rise and fall times of the output of each stage of the stepped buffer are plotted in Figure 4.16. From the figure, it can be seen that as the transition time of the input changes, the transition times of only the

first two stages of the stepped buffer change considerably. The change in transition time of the third stage is much less compared to the first two stages. The transition times of the other four stages remain the same**. Since the first two stages of the stepped buffer are relatively small compared to the other five stages, their contribution towards the substrate noise generated by the stepped buffer will be small. Hence, any change in the substrate noise due to a change in the rise and fall times of the outputs of the first two stages of the stepped buffer is also small.

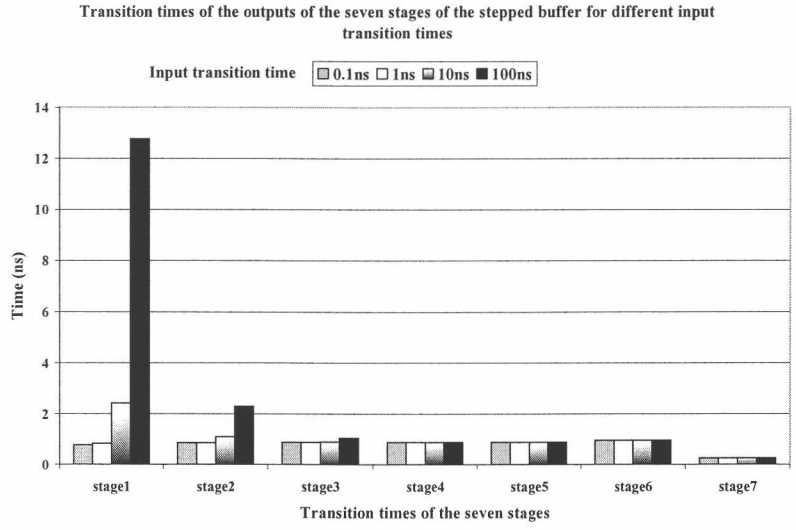


Figure 4.16 Graph showing the variation of the transition times of the outputs of the seven stages of the stepped buffer as the input transition time is varied from 0.1ns to 100ns. The transition times of the first two stages change considerably.

** The transition time of the last stage of the stepped buffer (output) is less compared to the remaining stages because the output is not loaded.

5. NOISE SUPPRESSION TECHNIQUES AND MEASUREMENT RESULTS

5.1 Noise suppression techniques

For mixed-signal chips in heavily doped substrates, if the digital noise generators and the sensitive analog circuits are placed at distances greater than 100μm, the current flow between them is through the low resistivity bulk, as discussed in Section 2.4.1. Hence, noise suppression techniques such as the use of guard rings, increased spacing between components etc. are not effective in reducing the noise coupling from the digital to the analog blocks. Some techniques which reduce the amount of noise coupled to the substrate are discussed below.

5.1.1 Separate source and bulk connections

In digital circuits, under normal circumstances, the source and bulk nodes of a transistor are tied together and then routed to a pin. In this scenario, all the supply noise present on the source nodes of the transistors appears on the substrate directly. To reduce the amount of supply noise coupled to the substrate, the source and bulk nodes of the transistors can be separated and routed to different pins. In this case, the supply noise present on the source nodes will not couple to the bulk node directly and thus, the amount of noise in the substrate decreases. This technique can be used to reduce supply noise coupling to the substrate.

5.1.2 Use of a grounded die perimeter ring

As was discussed in Chapter 2, switching noise is generated due to an imperfect ground at the bulk nodes of the transistors. If the ground connection to the bulk nodes can be improved, then the amount of switching noise coupled to the substrate can be reduced. This can be achieved by using a die perimeter ring [8]. A die perimeter ring is a ring of p+ diffusion, which is placed around the entire perimeter of the chip, with many contacts to the substrate. When this ring is grounded, it provides a low impedance path from the substrate to the ground, due to the many contacts that are present in the ring and helps in reducing switching noise coupled to the substrate. Care must be taken to ground the die perimeter ring with minimum parasitics in its ground path. This can be done by down bonding [14]. If the die perimeter ring is not grounded with minimum parasitics in its path, the reduction in noise coupling to the substrate may be small, as shown in Figure 5.1. The figure shows that as the inductor value in the backplane increases, the substrate noise picked up by a sensor placed at a distance of 5μm from the stepped buffer also increases. For a small inductance (<1nH) in the backplane, there is a small increase in the substrate noise (from 6 mV_{rms} to 7 mV_{rms} for supply inductance values less than 10pH with a 1nH inductance in the backplane, (Figure 5.1)), but as the inductance in the backplane increases, the increase in the substrate noise is substantial.

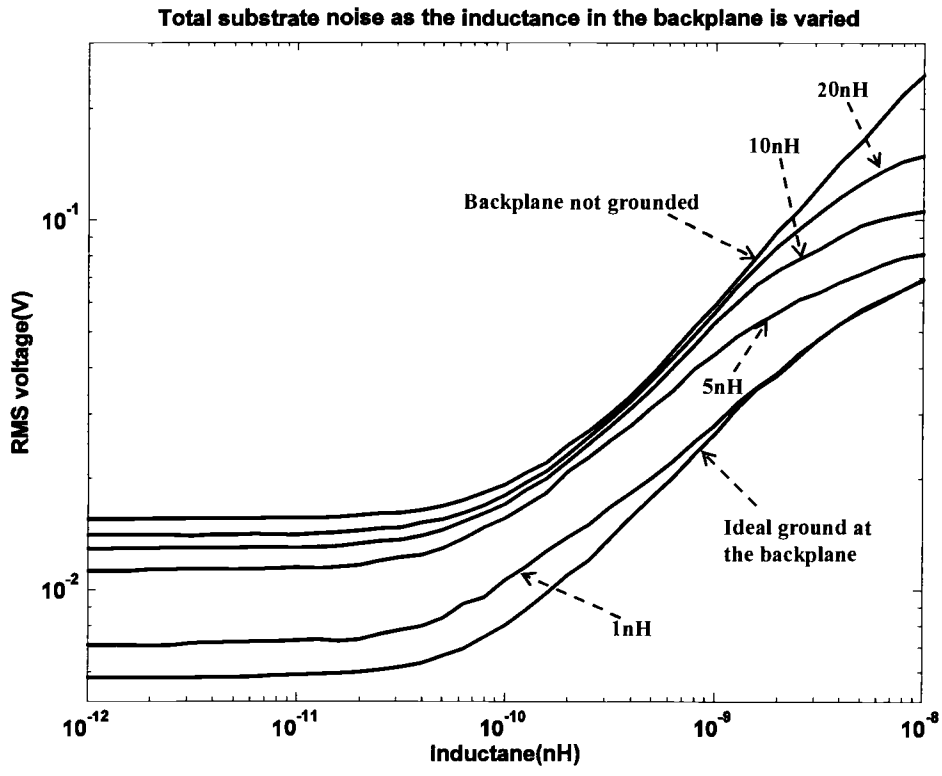


Figure 5.1 The total noise picked up by a sensor placed 5μm away from the stepped buffer increases as the inductance in the ground path of the backplane is increased. The input frequency of the stepped buffer is 10 MHz and the source and bulk nodes of the transistors in the buffer are tied together.

5.1.3 Simulations of noise suppression techniques

Simulations of the stepped buffer using the above noise reduction techniques are shown in Figure 5.2. For this figure, the pin parasitics were added in the supply lines and the stepped buffer was simulated for four different cases. First, the source and bulk nodes of the transistors were connected together and the die perimeter ring was not grounded. Then, the backplane was grounded by connecting the die perimeter

ring to a pin which is connected to ground. As a third case, the source and bulk nodes of the transistors were connected together, with the backplane not grounded, and finally, simulations have been done with the source and bulk nodes connected together and the backplane grounded. The rms values of the substrate noise picked up by a sensor placed 5μm away from the stepped buffer for all the four cases, when the inductance in the supply lines is varied from 1nH to 10nH, are shown in Figure 5.2. Input frequency of the stepped buffer is 10 MHz.

It can be seen from Figure 5.2 that when the bulk and source nodes are separated, the amount of substrate noise picked up by the sensor reduces by around 6 dB, at high inductor values. Since the supply noise gets coupled through the source and bulk, isolating the bulk and source nodes reduces the effect of supply noise on the substrate noise. When the bulk and source nodes are tied together, the amount of substrate noise reduction on the backplane, when the die perimeter ring is grounded is 6dB for low inductor values. Since the switching noise is affected by an imperfect ground, using a grounded die perimeter ring reduces the switching noise coupled to the substrate. On the whole, there is around 15dB reduction in substrate noise when the source and bulk nodes of the transistors are tied separately and when the backplane is grounded, compared to when the source and bulk nodes are tied together and the backplane is not grounded.

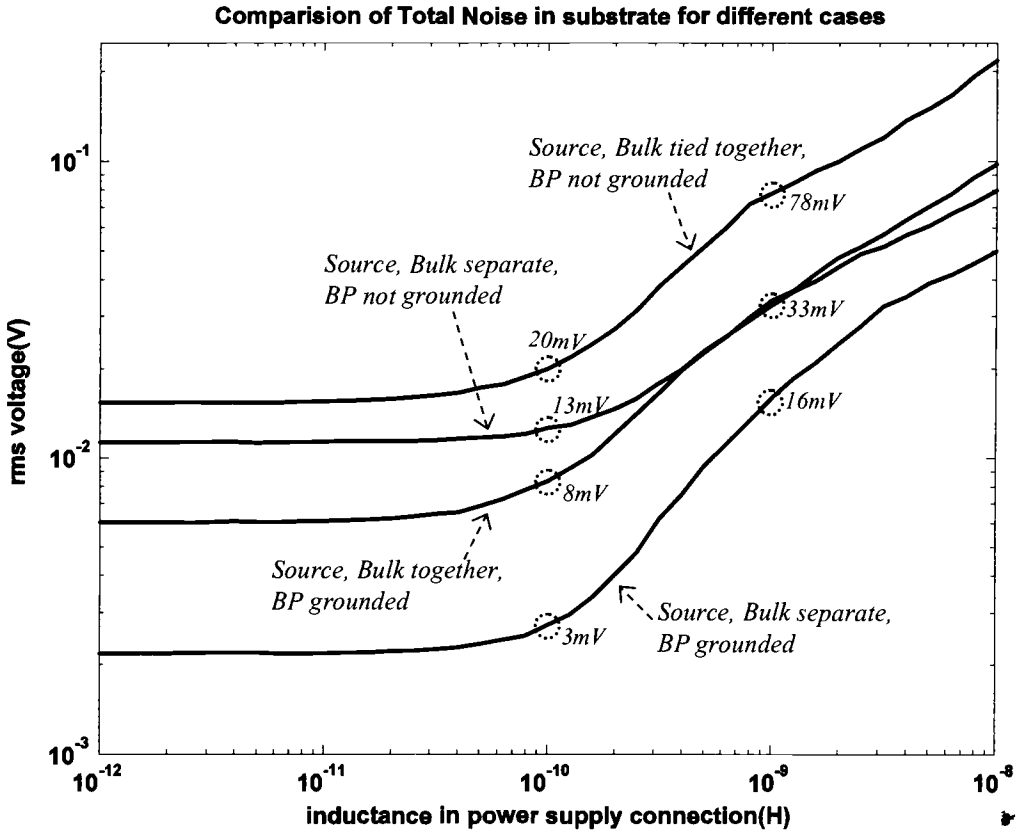


Figure 5.2 Comparison of total substrate noise picked up by a sensor placed 5μm away from the stepped buffer for different cases: bulk and source nodes tied together and separate, back plane grounded and not grounded. The input frequency of the buffer is 10 MHz.

The techniques discussed in the previous sections were used on the mixed-signal test chip fabricated in the TSMC CMOS 0.35μm process. This is a heavily doped process. The chip is packaged in a PGA121 package. The die photo of the chip is shown in Figure 5.3. In this chip, the stepped buffer has four different supplies, power and ground for the sources of the transistors and power and ground for the bulk nodes. The chip also contains a die perimeter ring. The measurements using the grounded die perimeter ring are shown in [8]. Measurements of the substrate noise to

verify the reduction in substrate noise due to separate bulk and source connection is discussed in the following section.

5.2 Measurement Technique

In [8], a variety of techniques to measure substrate noise, such as the wide band opamp, a p+-contact are discussed. In this thesis, substrate noise is obtained by measuring noise on a p+-contact. Noise on a p+-contact can be measured by connecting a probe-pad to it. Different probing solutions are available to measure the voltage on the probe-pad. Some of them use a low impedance probe, a high impedance probe or a low impedance GSG probe. Although the low impedance GSG probe provides good shielding from environmental noise during measurements, it loads the p+-contact due to its low impedance. For this reason, a high impedance active probe has been used to measure the noise on the p+-contact. The probe has an impedance of 1MΩ, shunted by a 0.1pF capacitance. The probe, when used with a high input impedance oscilloscope gives a 10:1 attenuation. So, the output seen on the oscilloscope needs to be multiplied by a factor of 10 to get the correct amount of substrate noise.

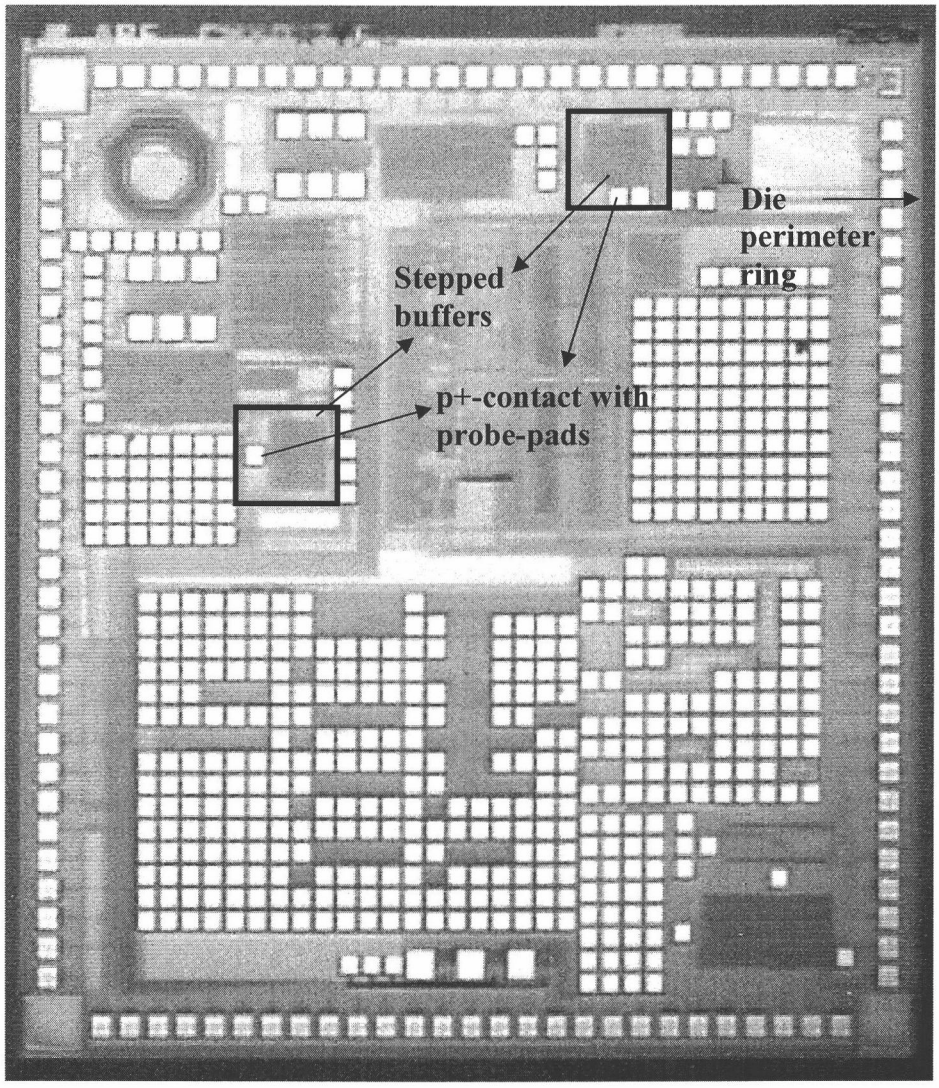


Figure 5.3 Die photo of the mixed-signal test chip showing the stepped buffers, the die perimeter ring and the p+-contacts used to measure substrate noise.

To measure the amount of substrate noise generated, the stepped buffer is given an input of 780 kHz. Frequencies higher than this were not chosen because the output of the stepped buffer becomes less ideal as the frequency of the input increases.

A p+-contact, placed at a distance of 5μm from the stepped buffer is probed with a high impedance probe. The transient waveform of the measured substrate noise is shown in Figure 5.4. Simulations for the stepped buffer are in good agreement with these measurement results. In these simulations the measurement setup needs to be emulated. The impedance of the wires providing supplies to the stepped buffer has to be included in the simulations. Also, the impedance of the probe used to obtain the noise voltage on the probe pad has to be included in the simulation setup. The characteristics of the input signal generator and the oscilloscope (impedance termination and capacitances) used to obtain all the transient waveforms are also included. The transient waveform of the noise obtained from a p+-contact in simulations is shown in Figure 5.4.

The peak-to-peak values [††] of the substrate noise in simulations and measurements are shown in Figure 5.4. The close agreement of the simulations with the measurements shows that including pin parasitics in digital noise simulations can accurately predict substrate noise.

[††] $V_{Ap\text{-}p}$ and $V_{Bp\text{-}p}$ are the peak-to-peak values of the substrate noise from simulations, $V_{Cp\text{-}p}$ and $V_{Dp\text{-}p}$ are the peak-to-peak values of substrate noise from measurements

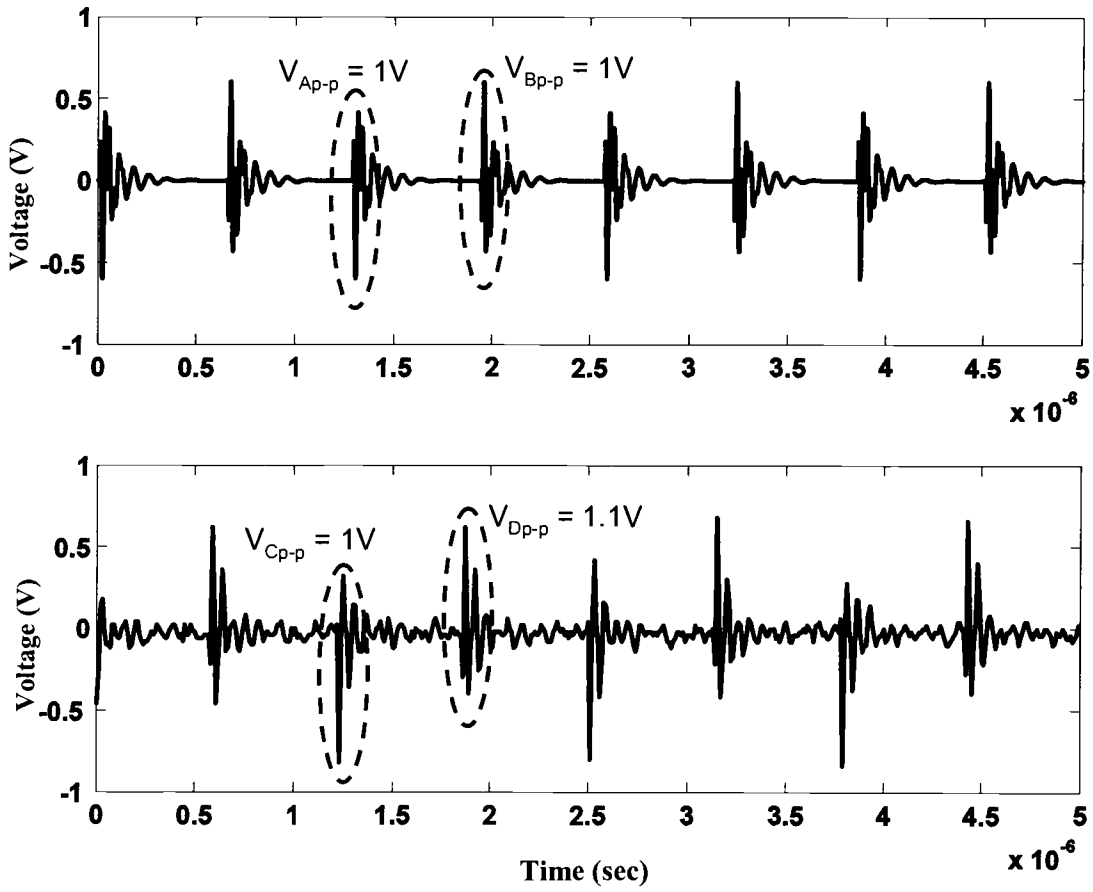


Figure 5.4 Simulation (top) and measurements (bottom) of the noise in the substrate when the stepped buffer is running at 780 kHz. The size of the first stage inverter is $(W/L)_n$ = 5μm/0.6μm and $(W/L)_p$ = 10μm/0.6μm. The source and bulk nodes of the transistors in the buffer are isolated.

Since the measurements and simulations of substrate noise match, simulations can be used to estimate the substrate noise generated by the stepped buffer when the bulk and source supplies are not separated. Simulations were done by connecting the source and bulk nodes of the transistors in the stepped buffer and keeping the rest of the setup the same as when the source and bulk nodes are separated. The transient

waveform of the substrate noise observed on a p+-contact, when the source and bulk supplies are tied together is shown in Figure 5.5. From the figure, it can be seen that both the positive and the negative peaks of the substrate noise waveform have increased by tying the source and bulk together. The peak-to-peak noise measured in this case is 1.4V.

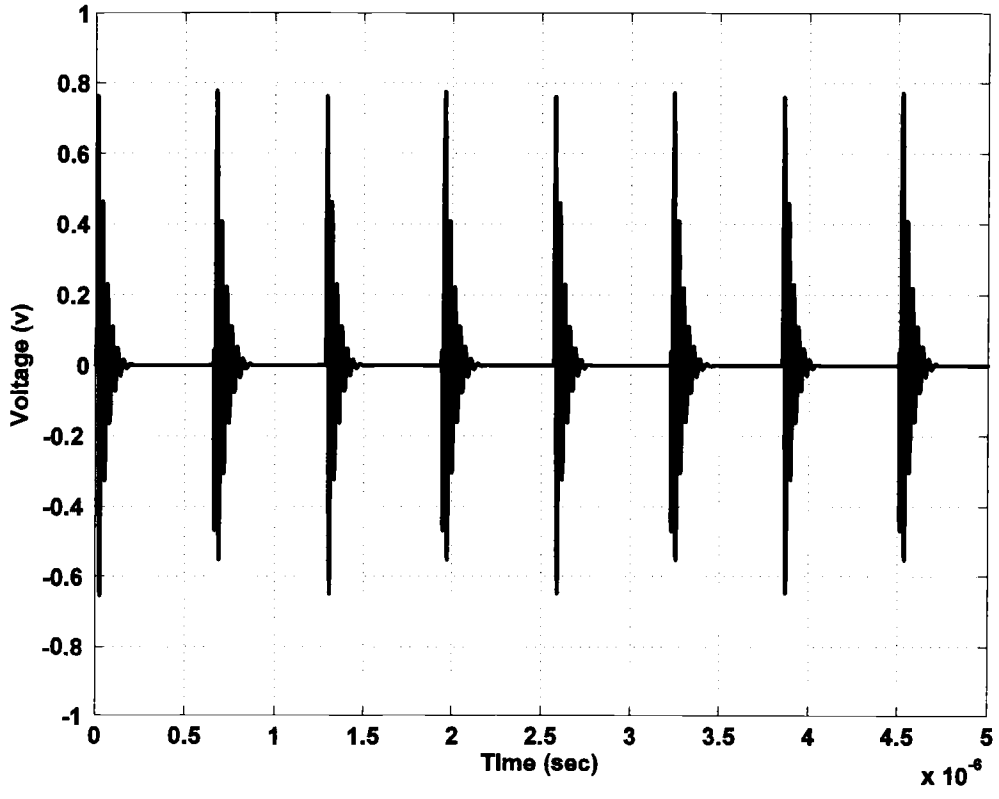


Figure 5.5 Simulation of substrate noise generated by a stepped buffer running at 780 kHz when the bulk and source are tied together.

6. CONCLUSIONS AND FUTURE DIRECTIONS

In this thesis, a simulation approach to isolate the supply and switching noise components from the substrate noise has been discussed and the importance of including pin parasitics in digital noise simulations has been elaborated. Simulations of the substrate noise generated by a stepped buffer were shown with and without including the pin parasitic model for the PGA121 package, which was used to package the mixed-signal test chip. Measurements of the substrate noise on the test chip matches with the simulations when pin parasitics were included. Thus, proper electrical models for the package need to be included for digital noise simulations. Noise suppression techniques were presented and measurements to validate these techniques were shown. Simulations and measurements on the stepped buffer show that there can be a substrate noise reduction of around 8dB when the source and bulk for a digital circuit are connected to separate pins.

Since the techniques for noise suppression have been proven to work on a heavily doped substrate [8], future work in this area can be directed towards their validation on a lightly doped process, such as an IBM 0.18 μm process. For a heavily doped substrate, the backplane can be considered as a single point, due to its low resistivity. In lightly doped substrates, this assumption may not be valid, due to the high resistive substrate. The issue of representing the backplane in simulations for lightly doped substrates needs to be looked into.

BIBLIOGRAPHY

[1] N. K. Verghese, T. J. Schmerbeck, and D. J. Allstot, *Simulation Techniques and Solutions for Mixed-Signal Coupling in Integrated Circuits*, Kluwer Academic Publishers, 1995.

[2] A. Samavedam, A. Sadate, K. Mayaram, and T. S. Fiez, "A scalable substrate noise coupling model for design of mixed-signal IC's," *IEEE J. Solid-State Circuits,* vol. 35, no. 6, pp. 895 – 904, June 2000.

[3] N. K. Verghese and D. J. Allstot, "Computer-aided design considerations for mixed-signal coupling in RF integrated circuits," *IEEE J. Solid-State Circuits*, vol. 33, no. 3, pp. 314-323, March 1998.

[4] R. Gharpurey and R. G. Meyer, "Modeling and analysis of substrate coupling in integrated circuits," *IEEE J. Solid-State Circuits*, vol. 31, no. 3, pp. 344-352, March 1996.

[5] B. R. Stanisic, N. K. Verghese, R. A. Rutenbar, L. R. Carley and D. J. Allstot, "Addressing substrate coupling in mixed-mode IC's: simulations and power distribution synthesis," *IEEE J. Solid-State Circuits*, vol. 29, no. 3, pp. 226-238, March 1994.

[6] A. J. van Genderen, N.P. van der Meijs and T. Smedes, "Fast Computation of Substrate Resistances in Large Circuits," *Proc. European Design and Test Conference*, pp. 560-565, Paris, France, March 1996.

[7] I. L. Wemple and A. T. Yang, "Integrated Circuit Substrate Coupling Models Based on Voroni Tessellation," *IEEE Trans. Computer-Aided Design,* vol. 14, no. 12, pp. 1459-1469, December 1995.

[8] B. Owens, P. Birrer, S. Adluri, R. Shreeve, S. A. Arunachalam, H. Habal, S. Hsu, A. Sharma, K. Mayaram, and T. Fiez, "Strategies for simulation, measurement and suppression of digital noise in mixed-signal circuits," *Proceedings of IEEE Custom Integrated Circuits Conference*, pp. 361 – 364, September 2003.

[9] D. Ozis, T. Fiez, and K. Mayaram, "A comprehensive geometry-dependent macromodel for substrate noise coupling in heavily doped CMOS processes," *Proceedings of IEEE Custom Integrated Circuits Conference*, pp. 497-500, May 2002.

[10] M. van Heijningen, J. Compiet, P. Wambacq, S. Donnay, M. G. E. Engels, and I. Bolsens, "Analysis and experimental verification of digital substrate noise generation

for epi-type substrates," *IEEE J. Solid-State Circuits*, vol. 35, no. 7, pp. 1002 – 1008, July 2000.

[11] J. Briaire and K.S. Krisch, " Principles of substrate crosstalk generation in CMOS circuits," *IEEE Trans. Computer-Aided Design*, vol. 19, no. 6, pp. 645-653, June 2000.

[12] D. K. Su, M. J. Loinaz, S. Masui and B. A. Wooley, "Experimental results and modeling techniques for substrate noise in mixed signal integrated circuits," *IEEE J. Solid-State Circuits*, vol. 28, no. 4, pp. 420-430, April 1993.

[13] Y. Zinzius, E. Lauwers, G. Gielen, and W. Sansen, "Evaluation of substrate noise effect on analog circuits in mixed-signal designs," *2000 Southwest Symposium on Mixed-Signal Design*, pp. 131-134, February 2000.

[14] B. Razavi, *Design of Analog CMOS Integrated Circuits*, New York: McGraw-Hill, 2002.

[15] The Center for High Performance Integrated Technologies and Systems, "Conventional Packaging Styles," http://www.eleceng.adelaide.edu.au/Personal/alsarawi/Packaging/node10.html#figthedifferencebetweenSMTandTHV, accessed August 2003.

[16] Amkor Technologies, "Flip Chip Packaging Technology Solution," http://www.amkor.com/enablingtechnologies/FlipChip/index.cfm, accessed August 2003.

[17] IC24h.com, "SOIC Packages," http://www.ic24h.com/, accessed August 2003

[18] Mitsubishi Integrated Circuit Packages, "IC Package Electrical Characteristics," http://www.mitsubishi.com/, accessed August 2003.

[19] CADENCE, Substrate Coupling Analysis User Guide, Product Version 4.4.6, April 2001.

[20] The Mosis Service, "Ceramic Packaging Options," http://www.mosis.org/Technical/Packaging/Ceramic/menu-pkg-ceramic.html, accessed August 2003.

[21] P. R. Gray and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*, New York: Wiley, 1995.

[22] R. Traylor, ECE574: System VLSI Design class notes, Oregon State University, Spring 2003.

[23] L. Liu, *Electrical Modeling Report: ASAT 32 Lead LPCC (5x5x0.9mm)*, ASAT Inc., June 2000.

[24] S. M. Kang, Y. Leblebici, *CMOS Digital Integrated Circuits Analysis and Design*, New York: McGraw-Hill, 1999.

APPENDICES

APPENDIX A. Simulation flow to study the effect of supply inductance on the various components of substrate noise.

To simulate the effect of supply inductance on the various components of substrate noise for a given digital circuit, the substrate network and the package parasitics need to be added to the digital circuit. These simulations are performed using HSPICE, because it is easier to obtain data files in HSPICE and port them to Matlab. The simulation flow is described below and also shown in Figure A.1.

Step 1: Obtain a netlist of the digital circuit in HSPICE format.

Step 2: Obtain a resistive network of the digital circuit using SCA, by labeling the active areas in the layout of the digital circuit (using Perl script 'make_portdata.pl' and SCA from command line).

Step 3: Modify the netlist of the digital circuit so that the resistive network can be appended to it, using Perl scripts 'clean_netlists.pl' and 'prog_addbulks.pl' (the bulk nodes of the transistors should be modified, so that they will be the same as the labeled active areas, which are used to obtain the resistive network).

Step 4: Add pin parasitics at the supply pins and generate netlists to simulate total, supply and switching noise as explained in the simulation approach to isolate the various substrate noise components in Chapter 3.

Step 5: Obtain the rms values of the noise present on the sensor, which is placed at a distance of 5μm from the digital circuit and plot the results in Matlab.

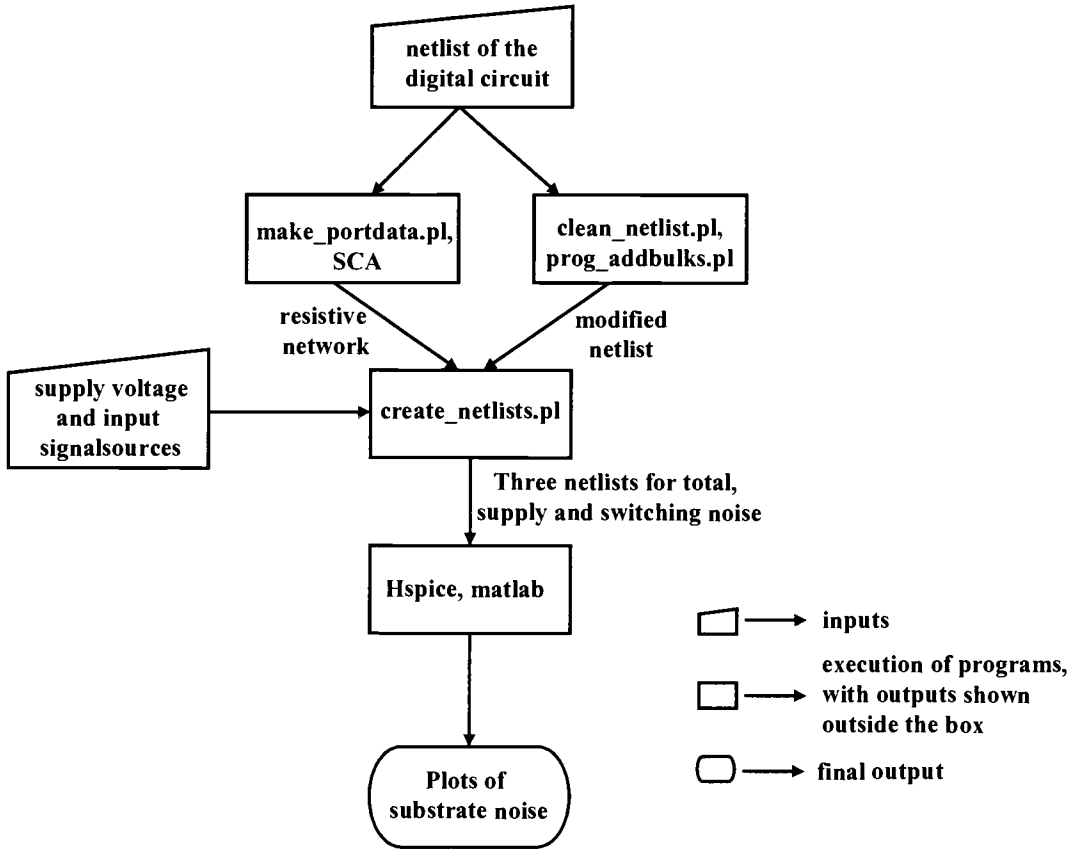


Figure A.1. Flow chart of the simulation process used to study the effect of supply inductance on the various substrate noise components.

Assumptions:

1) Only NMOS transistors are taken into consideration for extracting the resistive network. There is no resistive network for PMOS transistors.
2) The ground in the netlist is given a name and is not labeled as node '0'.

Inputs required for the simulation:

1. SPICE netlist of the digital circuit.
2. The power supply nodes need to be specified in an order in a file named 'supplies.txt'. The order is as follows: vdd to the source nodes of the PMOS transistors, vdd to the bulk nodes of the PMOS transistors, vss to the source nodes of the NMOS transistors, and vss to the bulk nodes of NMOS transistors. This is needed to identify the nodes to which the package parasitics are to be added.
3. The input signal source based on whose period, rise and fall times, transient analysis needs to be done.
4. A substrate profile, for SCA to extract the resistive network.

Flow Automation:

The SPICE netlist of the digital circuit is fed to Perl scripts 'clean_netlist.pl' and 'prog_addbulks.pl', which modify the netlist in such a way that the resistive network can be connected to the bulk nodes of the transistors. Usually, the bulks of the NMOSFET transistors are connected to ground and those of the PMOSFETs are connected to vdd. These scripts remove these connections and each of the bulk nodes

is named after its transistor number. These numbers are later used to label the active areas of the transistors in the layout.

Next, SCA is used to obtain the resistive network. SCA can be used in two ways. One way is to draw the layout of the digital circuit, extract it and then use SCA on the extracted layout. SCA internally generates a file which contains the layout information of the circuit (co-ordinates, layer information), called 'port.data' file and uses this file to generate the resistive network. Another way of using SCA is from the command line, which requires the 'port.data' file as an input. SCA is used from command line in this simulation flow and the 'port.data' file is created by a Perl script 'make_portdata.pl', using the transistor information from the SPICE netlist of the digital circuit.

Now, taking the modified netlist and the resistive network, three different files 'totalnoise.sp', 'supplynoise.sp' and 'mosnoise.sp' are created in a separate directory, using the script 'create_netlists.pl'. These files are created using the simulation approach to isolate supply and switching noise components described in Chapter 3. These .sp files are simulated using HSPICE to obtain the variation of the rms noise picked up by a sensor placed at a distance of 5μm from the digital circuit, when the inductance in the power supply lines is varied. The rms values and the inductance values are obtained from the .lis files generated by HSPICE and the plots showing the variation in substrate noise with respect to supply inductance are obtained using Matlab.

APPENDIX B. Perl and Matlab code used in the simulation flow to study the effect of supply inductance on the various components of substrate noise

clean_netlist.pl

 # This file takes the netlist given as an input and makes the netlist compatible to #HSPICE. For example, netlist extracted from Spectre HSPICE will have the #MOSFET model name as "nch" or "pch", this code converts it into "cmosn" or cmosp" etc.

```
$csv_file = $ARGV[0];# File with the netlist
$outfile=$ARGV[1]; # The cleaned netlist is written to this file
$voltagefile='voltagesources_in_netlist.txt';
# File used to store all the voltage sources present in the netlist

open(CSV, "<$csv_file") || die "Can't open $csv_file !";
open(OUT,">$outfile") || die "can't open outfile: $outfile ! ";
open(OUT1,">$voltagefile") || die "can't open outfile: $voltagefile ! ";
while($line = <CSV>)
{
        # Obtains all the voltage sources present in the netlist
        if($line=~m/^(v)/i)
        {
                print OUT1 $line;
        }
        else
        {
                if(($line=~m/^(\*)/) or ($line=~m/^(\.)/))
                {
                        $temp=$1;

                }
                else
                {
                        if(($line=~m/^(\+)/) and ($temp eq '.'))
                        {
                        }
                        else
                        {
                                if($line =~ m/^m/i)
                                {
                                        @fields = split(/\s+/, $line);
```

```
                                        if((lc(@fields[5]) eq nch) or (lc(@fields[5]) eq
nch_9) or (lc(@fields[5]) eq n))
                                        {
                                                @fields[5]= 'cmosn';
                                        }
                                        if((lc(@fields[5]) eq pch) or (lc(@fields[5]) eq
pch_9) or (lc(@fields[5]) eq p))
                                        {
                                                @fields[5]='cmosp';
                                        }
                                        print OUT join(" ",@fields),"\n";
                                }
                                else
                                {
                                        print OUT $line;
                                }
                        }

                }
        }
}
close CSV;
close OUT;
```

prog_addbulks.txt

```
#This code takes a netlist and makes the n-bulk nodes floating, making it possible to
#connect a resistive network later on. Requirements: an NMOS transistor is
#identified either by nch or nch_9 or cmosn

#!/usr/local/bin/perl
$csv_file = $ARGV[0]; # Netlist to be modified
$out_file = $ARGV[0].'.out'; # Output netlist
open(CSV, "<$csv_file") || die "Can't open $csv_file !";
open(OUT, ">$out_file") || die "Can't open output file: $out_file !";

while($line = <CSV>) {
        @fields = split(/\s+/, $line);
        # Pick a line of code which starts with m and then a number- indication of a
        #transistor
        if(($line =~ m/^m([a-z]+\d+)/i) or ($line =~ m/^m(\d+)/i)){
                $transnumber=$1; # get the number of the transistor
                if((lc(@fields[5]) eq nch) or (lc(@fields[5]) eq nch_9) or
                (lc(@fields[5]) eq cmosn) or (lc(@fields[5]) eq n)){
                        @fields[4]=ngate.${transnumber};
                        #print OUT $line,"\n";
                }

        }
        print OUT join(" ",@fields),"\n";
}
close CSV;
close OUT;
print "job done";
```

append_res_network.pl

```
# This file obtains the bulk nodes and the substrate contact names and creates files
#called ngates.txt and nbulks.txt  which will be used in "create_netlists.pl" to connect
#the bulk nodes and substrate contacts to appropriate voltages  for the 3 cases, total,
#supply and switching noise.

#!/usr/local/bin/perl
$csv_file = $ARGV[0]; # Cleaned netlist obtained from clean_netlist.pl
$res_file = $ARGV[1]; # Resistive network for the digital circuit

open(RES, "<$res_file")|| die "Can't open $res_file !";
$i=0;
%j=0;
$flag=0;
while($line = <RES>)
{
        @fields = split(/\s+/, $line);
        for($i=0;$i<scalar(@fields);$i++)
        {
                if(@fields[$i] =~ m/^(nbulk)/i )
                {
                        @bulks[$j]=@fields[$i];
                        $j++
                }
        }
}

for($i=0;$i<scalar(@bulks);$i++)
{
        for($j=0;$j<scalar(@bulks),$j!=$i;$j++)
        {
                if(@bulks[$i] eq @bulks[$j])
                {
                        @bulks[$j]= 0;
                }
        }
}

$j=0;
for($i=0;$i<scalar(@bulks);$i++)
{
        if(@bulks[$i] ne '0')
        {
```

```
                @newbulks[$j]=@bulks[$i];
                $j++;
        }
}
close RES;
open(OUT1,">nbulks.txt");

for($i=0;$i<scalar(@newbulks);$i++)
{
                print OUT1 @newbulks[$i],"\n";
}
close OUT1;
#For this, the bulk nodes(named ngate...) should be connected to a quiet gnd. So, first
#get all the nodes, obtain the bulk nodes from the resistive network file, then add
#appropriate voltage sources or resistors to connect them appropriately for each of
#the 3 cases

open(CSV, "<$csv_file") || die "Can't open $csv_file !";
$i=0;
$j=0;
$flag=0;
while($line = <CSV>)
{
        @fields = split(/\s+/, $line);
        for($i=0;$i<scalar(@fields);$i++)
        {
                if(@fields[$i] =~ m/^(ngate)/i )
                {
                        @gates[$j]=@fields[$i];
                        $j++;
                }
        }
}

for($i=0;$i<scalar(@gates);$i++)
{
        for($j=0;$j<scalar(@gates),$j!=$i;$j++)
        {
                if(@gates[$i] eq @gates[$j])
                {
                        @gates[$j]= 0;
                }
        }
}
```

```
$j=0;
for($i=0;$i<scalar(@gates);$i++)
{
        if(@gates[$i] ne '0')
        {
                @newgates[$j]=@gates[$i];
                $j++;
        }
}
close CSV;
open(OUT1,">ngates.txt");
for($i=0;$i<scalar(@newgates);$i++)
{
        print OUT1 @newgates[$i],"\n";
}
close OUT1;
```

create_netlists

```
# Creates three netlists 'totalnoise.sp' , 'supplynoise.sp' and 'mosnoise.s'p to be
#simulated in HSPICE

$csv_file = $ARGV[0]; # Ffile with the netlist of the digital circuit
$out_file1 = 'totalnoise.sp'; # Netlist to be created to simulate total noise
$out_file2 = 'supplynoise.sp'; # Netlist to be created to simulate supply only noise
$out_file3 = 'mosnoise.sp';# Netlist to be created to simulate mos only noise
$res_file = $ARGV[1]; # Resistive network to be conneceted to NMOSFET bulks
$supplies_file=$ARGV[2];
#Ffile which contains supply node information in the order: sourcevdd bulkvdd
#sourcegnd  bulkgnd

$source_file='voltagesources_in_netlist.txt';
# File of voltage sources obtained while cleaning the original  netlist

$netl_dir=$ARGV[4];
#Specifies the directory in which all the netlists have to be created

$input_file=$ARGV[3];
@outfile[0]=$out_file1;
@outfile[1]=$out_file2;
@outfile[2]=$out_file3;

mkdir $netl_dir;

for($i=0;$i<3;$i++)
{
        open(CSV,"<$csv_file") || die "Can't open $csv_file !";

        chdir $netl_dir;
        open(OUT, ">@outfile[$i]")|| die "Can't open ciffile:@outfile[$i] !";
        chdir '..';

        # Print the part of the netlist which is common to total, supply and mos netlists


        print OUT ".op\n.options post nomod ACCT RELTOL=1e-09 ABSTOL=1e-
15      VNTOL=1e-09\n";
        while($line=<CSV>)
        {
        print OUT $line;
        }
```

```
        print OUT ".include .\Vbsim3.mdl\n";
        print OUT ".include ", $res_file,"\n";

        #This function is used to identify input sources from a set of voltage sources
        #obtained while cleaning the netlist and print them... doesn't print supply
        #voltage sources

        printsources($source_file,$supplies_file);

        #This function creates pin parasitics between the supply node of the
        #circuit(given in supplies_file and it's voltage source given in source_file

        printsupplies($supplies_file,$source_file);

        #Depending on which file is open, it appends to the netlist accordingly
        printspecifics($supplies_file,$i);

        print_tran($input_file, $source_file);
        print OUT ".end\n";
        close OUT;
        close CSV;
}

sub printsources
{
        $source_file=@_[0];
        $supplies_file=@_[1];
        open(SOU,"<$source_file")|| die "Can't open $source_file!";
        while($linesource=<SOU>)
        {
                $flag=1;
                if($linesource =~ m/^(v)/i)
                {
                        @sources = split(/\s+/, $linesource);
                        open(SUP,"<$supplies_file")|| die "Can't open
                                $supplies_file !";
                        while($linesup=<SUP>)
                        {
                                if($linesup =~ m/^(\*)/i)
                                {
                                }
                                else
                                {
                                        @supplies = split(/\s+/, $linesup);
```

```
                                        checksupplies:
for($j=0;$j<scalar(@supplies);$j++)
                                        {
                                                if(lc(@sources[1]) eq @supplies[$j])
                                                {
                                                        $flag=0;
                                                        last checksupplies;
                                                }
                                        }
                                }
                        }
                        close SUP;
                }
                if($flag==1)
                {
                        print OUT $linesource;
                }
        }
        close SOU;
}

sub printsupplies
{
        $supplies_file=@_[0];
        $source_file=@_[1];

        open(SUP,"<$supplies_file")|| die "Can't open $supplies_file !";
        print OUT ".param l1=5n\n";
        while($linesup=<SUP>)
        {
                if($linesup =~ m/^(\*)/i)
                {
                }
                else
                {
                        @supplies = split(/\s+/, $linesup);
                        open(SOU,"<$source_file")|| die "Can't open $source_file!";
                        while($linesource=<SOU>)
                        {
                                if($linesource =~ m/^(v)/i)
                                {
                                        @sources = split(/\s+/, $linesource);
                                        for($j=0;$j<scalar(@supplies);$j++)
                                        {
```

```
                                        if(lc(@sources[1]) eq lc(@supplies[$j])
and                                     (lc(@supplies[$j]) ne lc(@supplies[$j-
1])))
                                        {
                                                print OUT "\nv",@supplies[$j],"
so", @supplies[$j], " 0 ", @sources[scalar(@sources)-1],"\n";
                                                print OUT "l",@supplies[$j],"
so", @supplies[$j], " intp",@supplies[$j], " 11\n";
                                                print OUT "r",@supplies[$j],"
intp",@supplies[$j]," ", @supplies[$j], " 0.5\n";
                                                print OUT "c",,@supplies[$j],"
so",@supplies[$j], " 0 2p\n";
                                        }
                                }
                        }
                }
                close SOU;
                }
        }
        close SUP;
}

sub printspecifics
{
        $supplies_file=@_[0];
        $j=@_[1];
        open(SUP,"<$supplies_file") || die "Can't open file: $supplies_file !";

        while($linesup = <SUP>)
        {
                if($linesup =~ m/^(\*)/i) # Implies ignore comments
                {
                }
                else
                {
                        @supplies = split(/\s+/, $linesup);
                        if($j==0)
                # If the file open is totalnoise netlist, connect all the bulk contacts to
                #the bulksupply
                        {
                                open(BUL,"<nbulks.txt");
                                while($linebulk = <BUL>)
                                {
                                        @fields = split(/\s+/, $linebulk);
```

```
                                        if(scalar(@supplies) > 2)
                # This is written so that the line doesnot get repeated for a blank line
                # in supplies_file
                                        {
                                                print OUT "v",@fields[0]," ",@fields[0],
" ", @supplies[3], " 0\n";
                                        }
                                }
                                close BUL;
                        }
                        elsif ($j==1)
                # If the file open is supply netlist, connect the bulk contacts to the bulk
                # supply and the bulk nodes to quiet supply
                        {
                                open(BUL,"<nbulks.txt");
                                while($linebulk = <BUL>)
                                {
                                        @fields = split(/\s+/, $linebulk);
                                        if(scalar(@supplies) > 2)
                # This is written so that the line doesnot get repeated for a blank line
                # in supplies_file
                                        {
                                                print OUT "v",@fields[0]," ",@fields[0],
" ", @supplies[3], " 0\n";
                                        }
                                }
                                close BUL;

                                open(GAT,"<ngates.txt");
                # Connecting bulk nodes to quiet supply
                                while($linegate = <GAT>)
                                {
                                        @fields = split(/\s+/, $linegate);
                                        if(scalar(@supplies) > 2)
                # This is written so that the line doesnot get repeated for a blank line
                #in supplies_file
                                        {
                                                print OUT "v",@fields[0],"
",@fields[0]," 0 0\n";
                                        }
                                }
                                close GAT;
                        }
                        elsif($j==2)
```

```
                # If the file open is mos netlist, connect the bulk contacts to quiet
                #bulksupply
                        {
                                open(BUL,"<nbulks.txt");
                                open(SOU,"<$source_file")|| die "Can't open
                                        $source_file!";
                                while($linesource=<SOU>)
                                {
                                        if($linesource =~ m/^(v)/i)
                                        {
                                                @sources = split(/\s+/, $linesource);
                                        }
                                        if(lc(@sources[1]) eq lc(@supplies[3]))
                                        {
                                                print OUT "vq", @supplies[3], " q",
@supplies[3], " 0 ", @sources[scalar(@sources-1)], "\n";

                                        }
                                }
                                close SOU;
                                while($linebulk = <BUL>)
                                {
                                        @fields = split(/\s+/, $linebulk);
                                        if(scalar(@supplies) > 2)
                                        {
                                                print OUT "v",@fields[0], " ",
@fields[0], "q", @supplies[3], " 0\n";

                                        }
                                }
                                close BUL;
                        }

                }
        }
        close SUP;
}

sub print_tran
{
        $source_file=@_[1];
        $input_file=@_[0];

        open(SOU,"<$source_file");
```

```
	while($linesource=<SOU>)
	{
		if($linesource=~ m/^(v)/i)
		{
			open(INF,"<$input_file");
			while($lineinput=<INF>)
			{
				if($lineinput=~ m/^(v)/i)
				{
					@sources=split(/\s+/, $linesource);
					@inputs=split(/\s+/, $lineinput);
					if(lc(@sources[0]) eq lc(@inputs[0]))
					{
						$tran_last=@sources[scalar(@sources)-
1]+@sources[scalar(@sources)-5];
						$period=@sources[scalar(@sources)-1];
						print OUT ".tran ",
@sources[scalar(@sources)-1]/10, " ", $tran_last," sweep ll DEC 10 1e-12 1e-08\n";
						print OUT ".measure tran rmsnoise rms
v(sensor) from= ",$tran_last-$period, " to= ", $tran_last,"\n";
						print OUT ".measure tran rmsnoisebp
rms v(bp) from= ",$tran_last-$period, " to= ", $tran_last,"\n";

					}

				}
			}
		}
		close INF;
	}
	close SOU;
}
```

count_fingers.pl

```
#This code is used on the netlist to obtain the number of fingers for each transistor
#and it's length and width, this information will be used to create the port.data file to
#be given to SCA to extract the resistive network

$csv_file = $ARGV[0]; # input netlist
$outfile=$ARGV[1];
# Creates a file which contains the transistor number, the number of fingers, its
length #and width.

open(CSV, "<$csv_file") || die "Can't open $csv_file !";
open(OUT,">$outfile") || die "can't open outfile: $outfile ! ";

#Previous transistor
$transprev=a;
$i=0;
$j=1;
$gate=0;
$flag=1;


while($line=<CSV>)
{
        @fields=split(/\s+/,$line);

        #Find out the number of fingers for each transistor, it's length and width, only
        #for NMOSFET transistors
        if((($line=~m/^m(\d+)_\d+/i) or ($line=~m/^m(\d+)/i) or
($line=~m/^m(load\d+)/i)) and ((lc(@fields[5]) eq nch) or (lc(@fields[5]) eq nch_9)
or (lc(@fields[5]) eq cmosn) or (lc(@fields[5]) eq n)))
        {

                $transnumber=$1;

        # If the transistor now is equal to the previous transistor, then it is a finger of
        #the same transistor. so, increment the no. of fingers
                if(lc($transprev) ne lc($transnumber))
                {
                        if($gate!=0)
                        {
                                $gate++;
                        }
                        $transprev=$transnumber;
```

```
                @gatenumber[$gate++]=$transnumber;
                @gatenumber[$gate++]=@fields[6];
                @gatenumber[$gate++]=@fields[7];
                @gatenumber[$gate]=number.1;
                $j=1;
        }
        else
        {
                $j++;
                @gatenumber[$gate]=number.${j};
        }
    }

}
print OUT join(" ",@gatenumber),"\n";
close CSV;
```

make_portdata.pl

```
#This file takes the information given in the input file about the transistors, their
#lengths, widths and number of fingers to generate the cif file. All dimensions are in
#microns


$csv_file = $ARGV[0]; # File containing the information of all the transistors
$outfile=$ARGV[1]; # port.data file

open(CSV, "<$csv_file") || die "Can't open $csv_file !";
open(OUT,">$outfile") || die "can't open outfile: $outfile ! ";

#The x and y co-ordinates of the center of each layer
$xpos1=0;
$ypos1=0;
$xpos2=0;
$ypos2=0;

#Minimum spacing between each layer
$minspacing=1.1;

#Minimum length for each of the source and drain region
$minsource=1;

while($line=<CSV>)
{
        print OUT "userUnits \"micron\"\n";

        @fields=split(/\s+/,$line);

        for($i=0;$i<scalar(@fields);$i=$i+4)
        {
        #Take the length and width and retain only the numbers
                @fields[$i+1]=~ s/L=//;

        #This will give the length in centi microns. So, need to be converted into
        #microns
                @fields[$i+1]=~s/E-9//;
                @fields[$i+1]=@fields[$i+1]/1000;
                @fields[$i+2]=~ s/W=//;

                if(@fields[$i+2]=~ m/E-6/i){
```

```
                @fields[$i+2]=~s/E-6//;
        }
        elsif(@fields[$i+2]=~ m/E-3/i){
                @fields[$i+2]=~s/E-3//;
                @fields[$i+2]=@fields[$i+2]*1000;
        }
        elsif(@fields[$i+2]=~ m/E-9/i){
                @fields[$i+2]=~s/E-9//;
                @fields[$i+2]=@fields[$i+2]/1000;
        }

        @fields[$i+3]=~s/number//;

        $fingers=@fields[$i+3];
        $length=@fields[$i+1];
        $width=@fields[$i+2];

        if($i!=0)
        {
                $xpos1=$xpos2+$minspacing;
        }
        else
        {
                $xpos1=0;
        }

        $xpos2=$xpos1+ ($fingers*$length + ($fingers+1)*$minsource);

        $ypos2=$width;

        print OUT "net=ngate",@fields[$i]," layer=scab depth=0 rect
((",$xpos1," ",$ypos1,") (",$xpos2," ",$ypos2,"))\n";
        print OUT "net=nbulk",@fields[$i]," layer=scab depth=0 rect
((",$xpos1," ",$ypos1-2.1,") (",$xpos2," ",$ypos1-1.4,"))\n";
    }
    print OUT "net=sensor layer=scab depth=0 rect ((",$xpos2/2-5," ", $ypos1-
8.4,") (",$xpos2/2+5," ",$ypos1-7.7,"))\n";
}
```

doeverything

 # Shell script to execute the commands required to create the three netlists: #'totalnoise.sp', 'supplynoise.sp' and 'mosnoise.sp'

```
perl clean_netlist.pl netlist.txt cleannetlist
perl prog_addbulks.pl cleannetlist  # generates a file called cleannetlist.out
perl append_res_network.pl cleannetlist.out resistivenetwork.txt
perl create_netlists.pl cleannetlist.out resistivenetwork.txt supplies.txt inputsource.txt
simulationfiles
cp resistivenetwork.txt simulationfiles/resistivenetwork.txt
rm cleannetlist
rm cleannetlist.out
rm nbulks.txt
rm ngates.txt
rm voltagesources_in_netlist.txt
```

do_final

#Shell script to run all the commands to create, execute and view the results of all the #three netlists created.

```
cp steppedbuffer_small.txt netlist.txt
cp netlist.txt layout_auto/netlist.txt

# To generate the resistive network
cd layout_auto
perl count_fingers.pl netlist.txt gatecount.txt
perl make_portdata.pl gatecount.txt port.data
cd ../../..
subx -procfile Research/Perl/layout_auto/sample_srp.proc -portfile
Research/Perl/layout_auto/port.data -modelfile
Research/Perl/layout_auto/substrate.scs -modelname substrate -acculevel high >>
Research/Perl/layout_auto/sca.log
cd Research/Perl/layout_auto
cleansca
cp substrate.scs ../resistivenetwork.txt

#To create the three netlists
cd ..
doeverything
rm netlist.txt
```

```
#To simulate the netlists in HSPICE
cd simulationfiles
hspice totalnoise.sp > totalnoise.lis
hspice supplynoise.sp > supplynoise.lis
hspice mosnoise.sp > mosnoise.lis
cleanup

matlab -nosplash -nojvm -r simul
```

simul.m

```
clear all;
close all;

set(0,'DefaultLineLineWidth',3);
set(0,'DefaultAxesFontWeight','bold');
set(0,'DefaultAxesFontSize',15);


format long e;
load('totalnoise.txt');
load('inductance.txt');
load('supply.txt');
load('mos.txt');
ytotal=totalnoise(:,1);
x=inductance(:,1);
ysupply=supply(:,1);
ymos=mos(:,1);
figure(100);
loglog(x,ytotal,'r');hold on;
loglog(x,ysupply,'g');hold on;
loglog(x,ymos,'b');
axis([1e-12 1e-8 1e-5 1]);
xlabel('inductance in power supply connection');
ylabel('RMS value of the substrate noise');
title('RMS values for Noise in a sensor');
grid on;
```

APPENDIX C. Layout of the stepped buffer