Abstract

Substrate noise is considered a critical parasitic in mixed-signal integrated circuits. Power supply noise is the dominant source of substrate noise. Various attempts have been made at both the circuit level and software level to estimate this noise. Software level noise estimation is especially important as software is becoming an increasingly integral part of future systems on a chip. In this paper, we propose a new model for estimating the di/dt noise and incorporate it into a publicly available power simulator (SimplePower[8]) for an embedded processor core. Next, we investigate how an ADC can be designed to adapt its resolution in the presence of the substrate noise generated when the SimplePower processor core is embedded on the same chip. The proposed strategies prevent unexpected performance degradation of the ADC.

1. INTRODUCTION

Analog circuits are being implemented in deep sub-micron CMOS technology and integrated along with high-speed digital circuits such as microprocessor cores for multimedia applications. In these mixed-signal integrated circuits (ICs), high-speed digital circuits generate noise when they are switching. This noise, called substrate noise, is propagated to the analog circuits through a shared substrate and deteriorates the performance of the analog circuits. This substrate noise is generated by major two sources, noise coupling from the switching transistors and noise coupling from the power-supply known as di/dt noise. As the di/dt noise is the dominant source of substrate noise[1], the estimation of di/dt noise is an important issue for designing the reliable mixed-signal IC.

In order to estimate the di/dt noise accurately, many attempts have been made at the circuit level [2,3]. However, these are not appropriate for estimating the noise that a software program may cause. This was the focus of recent work presented by Grochowski et. al [4]. By modifying the Wattch

* This work was supported in part by NSF 0082064, CAREER 0093085 and MARCO 98-DF-600 GSRC
toolset[5], they extract the current profile in advance and estimate the noise. The magnitude of pulse was determined by the power consumption in a single clock cycle and the current shape was modeled as a rectangular pulse. This approximation is a simple and effective way to quickly calculate the noise effect that software causes. However, this approximation can underestimate or overestimate the noise since a fixed current step is ideal, while in reality the shape of currents are often quite varied [6]. Therefore, a more accurate approximation is needed.

In this paper, we propose a new model for estimating the \( \text{di/dt} \) noise and incorporate it into a publicly available power simulator (SimplePower[8]) for an embedded processor core. Next, we investigate how an ADC can be designed to adapt its resolution in the presence of the substrate noise generated when the SimplePower processor core is embedded on the same chip. The proposed strategies prevent unexpected performance degradation of ADC.

The remainder of this paper is organized into three major chapters. In chapter 2, we describe the new current model; apply it to SimplePower; and explain how to measure the noise using the modified SimplePower. In chapter 3, we briefly explain the adaptive resolution ADC; analyze the performance degradation of the ADC due to the substrate noise; and propose strategies to avoid serious performance degradation. Finally, we provide concluding remark in chapter 4.

2. NOISE MODELING USING SWITCHING CAPACITANCE

Switching capacitance is one of the crucial factors that determines dynamic power consumption. Many power estimation tools extract the switching capacitance through different ways. Those methods are categorized by simulation levels, such as circuit, logic, architecture, behavioral, and application level. Among those, the architecture level simulator is appropriate for estimating power consumption of the microprocessor when software programs are running. It can be used to support architectural compiler, operating system, and application level experimentation. Sente’s WattWatcher, Synopsys’s DesignPower and PowerCompiler are the commercial tools and Wattch[5] and Simplepower[8] are the prototype academic tools. The variation of current is key information to estimate \( \text{di/dt} \) noise. If the relationship between the switching capacitance and the variation
of current is reliably defined, architectural level power estimation tools can be evolved as noise estimation tools. In the next section, we discuss how a more accurate current model is built.

### 2.1 Switching capacitance model

Our discussion starts with the case of a single CMOS gate. Whenever a CMOS gate is switching, the load capacitance is charged or discharged. It forms a current path from the external power supply to a CMOS gate. Figure 1(a) shows the current path that is established by a single CMOS inverter’s discharging activity. The noise is generated through the current path as in Figure 1(a) [9]. The shape of the current traveling through this path can be simply modeled as a triangle as shown in Figure 1(b) [6]. \( I_{\text{peak}} \) is the peak current through the current path in Figure 1(a) and defined by Eq-1. The parameter, \( D \), is the delay time of a CMOS gate.

\[
I_{\text{peak}} = \frac{2V_{\text{dd}} \times C_{\text{sw}}}{D} \quad \text{– Eq-1}
\]

![Figure 1. A single CMOS model](image)

Now the discussion is extended to multiple CMOS gates. Circuit level experiments of different types of digital circuits are performed to check whether the current shape of a single CMOS gate can be directly applied to multiple CMOS gates. Digital circuits are categorized by two types of logic, combinational logic such as adders, multipliers, and decoders and synchronous logic such as latches, flip-flops, and registers. We chose five sample circuits. For combinational logic, a single array multiplier cell and a 16x16 array multiplier are chosen. For synchronous logic, a single flip-flop and an 8-bit register are chosen. The last circuit is the combination of 4-bit flip-flops and a single array multiplier cell. These circuits were designed in 0.18 micron technology and simulated by HSPICE. Figure 2 shows the current shapes of these circuits. Table 1 contains the data that is
required for calculating $I_{\text{peak}}$ in Eq-1. The solid lines are the result of HSPICE simulation and the dotted lines are the approximated results of Eq-1 and Figure 1(b).

Figure 2. Current shape of five sample circuits

(a) Current shape of a single FF  (b) Current shape of a 8-bit register

(c) Current shape of a single array multiplier cell  (d) Current shape of 16x16 array multiplier

(e) 4-bit registers and a single array multiplier

Figure 2. Current shape of five sample circuits
As shown in Figure 2(a) and (b), the approximated current shape of synchronous circuits are close to the actual shape. Also, Figure 2(b) indicates that the current of multiple flip-flops are the sum of the single flip-flops’ current (flip-flop’s Ipeak=3.3mA and 8-bit register’s Ipeak=27mA). This is because the switching of a flip-flop is concentrated on the instance of the clock edge and most logic components in a flip-flop are activated within a short period time. However, the results from combinational circuits show different aspects from those of synchronous circuitry. A relatively simple circuit like the single array multiplier cell is appropriate for this approximation, but a more complex circuit is not apt to follow this approximation as shown in Figure 2(d) and 2(e).

Thus, we must consider an alternative to model the current shape for such circuits. If the circuit is composed of synchronous circuits and combinational circuits as in Figure 2(e), the current shape of this circuit will be modeled as a sum of two different parts, a synchronous part based on the synchronous switching capacitance and a combinational part based on the combinational switching capacitance. The synchronous part is modeled as a triangular shape as Figure 2(a) and the combinational part is modeled as a fixed current step as in [4]. In Figure 2(e), the dotted line(B) represents this model. Compared to the dotted line(A) that is the result of triangular approximation, it is closer to the actual current shape.

2.2 The methodology of Noise analysis

The previous section shows that an architectural power estimation tool that measures switched capacitance can be modified to measure the di/dt noise. We used SimplePower as the architectural power estimation tool as it provides cycle-accurate switching capacitance of a microprocessor core. It has the

<table>
<thead>
<tr>
<th>Circuits</th>
<th>D (ns)</th>
<th>Csw (pF)</th>
<th>Ipeak (mA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single FF</td>
<td>0.25</td>
<td>0.18</td>
<td>2.9</td>
</tr>
<tr>
<td>8-bit Reg</td>
<td>0.25</td>
<td>1.47</td>
<td>23.6</td>
</tr>
<tr>
<td>Single multiplier</td>
<td>0.85</td>
<td>0.17</td>
<td>0.78</td>
</tr>
<tr>
<td>16x16 multiplier</td>
<td>5.6</td>
<td>135.24</td>
<td>96.6</td>
</tr>
<tr>
<td>4-bit Reg. /single multiplier</td>
<td>0.87</td>
<td>1.11</td>
<td>5.1</td>
</tr>
</tbody>
</table>

Table 1. Measured Csw and calculated Ipeak (Vdd = 2V)
architecture of a five-stage single issue pipelined datapath that is composed of the fetch stage IF, the instruction decode stage ID, the execution stage EXE, the memory access stage MEM, and the write-back stage WB. At each clock cycle, it extracts the switching capacitance of each stage through pre-calculated values characterized from actual designs. It is a transition-sensitive power estimation tool that reports the different switching capacitances according to the different input vectors [8]. Using this tool, we can accurately separate the switching capacitance of synchronous circuits and combinational circuits at every clock cycle.

As mentioned, the synchronous circuits are concurrently switching at a clock edge and within a very short period of time. Therefore, the current shape of synchronous circuits within one clock cycle can be accurately estimated through the switching capacitance of synchronous part as discussed in Figure 2(a). It should be noted that the combinational circuits of a microprocessor core are more complex than the multipliers that were discussed in section 2.1. Their switching activities are randomly distributed within a clock cycle. Therefore, we need to make an assumption that the current shape of combinational circuits within one clock cycle is uniformly distributed. In other words, it has a fixed current step as in [4]. Figure 3 shows the current profile that is generated by running a benchmark, bsrch.c, that performs binary search. The right-hand side graph is the enlarged current shape of a few clock cycles. The triangular shape is for the synchronous circuits and rectangular shape is for the combinational circuits.

![Figure 3. Current profile of bsrch.c (100 cycles)](image)

From the current profile as in Figure 3, two different ways of analysis are available on two different domains, time and frequency. Figure 4 is the block diagram of the noise transfer function. As in [4], H(s) includes the power distribution network, package, and bond wire. However, we only assume the bondwire model.
for simplicity. Figure 4 shows the simplest bond wire model that is a RL circuit. The value of the RL circuit is assumed to be 0.15 ohm and 1.5nH based on a ceramic pin grid array (CPGA) package [10].

![Block diagram of noise transfer function](image)

Figure 4. Block diagram of noise transfer function

First, the time domain analysis is discussed. When the current is running through the bondwire as in Figure 1(a), Vss and Vdd have symmetric noise such that \( V_{ss} = V_{ss} + RI(t) + L\frac{dI}{dt} \) and \( V_{dd} = V_{dd} - (RI(t) + L\frac{dI}{dt}) \). Also, the RL circuit has the property of linearity such that \( f(ax+by) = af(x)+bf(y) \) because it is a linear circuit. This property makes the analysis easier; we can analyze each clock cycle and add up all results at the end of simulation. Figure 5(a) shows the time response of the noise for benchmark bsrch.c for the first 100 cycles. From this noise waveform, the Root Mean Square (RMS) of each clock cycle’s noise can be calculated. The RMS value is a widely used index to express the level of noise. Figure 5(b) is the histogram of RMS. This graph gives us the information about the distribution and level of noise. The influence of the distribution of RMS on the analog circuit is discussed later in Chapter 3.

![Time response of noise and histogram of RMS values](image)

Figure 5. The time response of noise and the histogram of RMS values (bsrch.c)

Next, the frequency domain analysis is discussed. In [11], the problem of the resonant frequency of the supply was stated. The resonant frequency of a supply amplifies the level of noise. To resolve this problem, a
reliable frequency analysis should be provided. Using our approach, the frequency analysis is available. To begin with, the Discrete Fourier Transform (DFT) is performed on the current profile and the DFT of the transfer function is multiplied by it. The frequency response of noise is then obtained as shown in Figure 6. In Figure 6(b), the frequency response of noise is shown. Based on this result, we can analyze the response around critical frequency range. For example, a new algorithm can be applied to reduce the noise due to the resonant frequency as in [11]. Before applying it, we can estimate the response of the resonant frequency and compare the results with that generated by a new algorithm.

Figure 6 Frequency response of noise

3. SOFTWARE-DRIVEN ADAPTIVE RESOLUTION ANALOG-TO-DIGITAL CONVERTER (ADC)

In this chapter, we discuss how the noise affects the performance of an analog circuit and how these effects can be controlled using noise analysis presented in the previous chapter. An ADC is an analog component commonly embedded with a microprocessor core on a single chip. The baseband modem that is used in wireless terminals is a good example of this. In this paper, we assume that SimplePower’s microprocessor core is embedded in the same chip as an adaptive resolution ADC called Power and Resolution Adaptive ADC (PRA-ADC) designed in 0.18 micron CMOS technology[7]. We analyze how the noise deteriorates the performance of ADC and suggest how this problem can be avoided.

3.1 Adaptive resolution analog-to-digital converter
To analyze the effect of noise, we first discuss the properties of the comparator used in the PRA-ADC. PRA-ADC uses a quite different type of voltage comparator that is known as the Threshold Inverter Quantization (TIQ). The TIQ comparator uses two cascaded CMOS inverters as a comparator for implementing a high-speed flash ADC. The principle of this comparator is related to the inverter threshold voltage denoted to $V_m$ and defined as the $V_{in}=V_{out}$ point in the VTC of an inverter. It is defined by Eq-2 mathematically. In Figure 7, $V_m$ plays the same role as the threshold voltage of the differential amplifier (Figure 7(a)). Also $V_m$ is adjusted by changing the ratio of $(W/L)_p$ and $(W/L)_n$. Using Eq-2, the threshold of comparators is determined.

$$V_m = \frac{r(V_{DD} - |V_p| + V_n)}{1 + r} \quad \text{with} \quad r = \frac{\mu_p W_p}{\mu_n W_n} \quad - \text{Eq-2}$$

(\text{where} \quad \nu_p : \text{PMOS threshold}, \quad \nu_n : \text{NMOS threshold}, \quad \mu_p \text{ and } \mu_n \text{ are hole and electron mobility})

![Figure 7. The comparator of PRA-ADC](image)

\[ a) \text{ Conventional Comparator} \quad b) \text{ TIQ Comparator} \]

### 3.2 The analysis of error caused by noise

![Figure 8. The substrate interconnection between analog circuits and digital circuits](image)

As Figure 8 illustrates, the external power and ground lines of analog circuitry is separated from those of digital circuitry. However, internal grounds (AGnd and Gnd) are connected by the low ohmic substrate (less than 1ohm for large digital circuit) [1]. Therefore, ground bounce is directly propagated to the analog circuit. This
noise is the dominant source of substrate noise. This substrate noise results in the reduction of the noise margin of analog circuits.

It should be noted that, $V_m$ can fluctuate when the threshold voltage of PMOS or NMOS changes due to the substrate noise. In general, the threshold voltage is determined by Eq-3. $V_T$ is a function of $V_{SB}$ because other parameters have constant values determined by the semiconductor technology. This $V_{SB}$ makes the threshold voltage fluctuate, as a consequence, $V_m$ changes as a result of the variation. The variation of $V_m$ is directly related to the substrate noise. Therefore, the TIQ based comparator is vulnerable to substrate noise. Also, we can easily estimate the substrate noise by an analytical model using Eq-2 and Eq-3.

$$V_T = V_{T0} + \gamma \left( \sqrt{\phi + V_{SB}} - \sqrt{\phi} \right) \quad \text{Eq-3}$$

($\gamma$: body-effect coeff., $\phi$: the surface inversion potential, and $V_{T0}$: the threshold voltage for $V_{SB} = 0V$)

Before simulating the variance of $V_m$, $V_{SB}$ is modeled as Eq-4 [12].

$$V_{SB} = \sum_{n=1}^{\infty} A_n \cdot \sin(\omega_n \cdot t + \varphi) \quad \text{Eq-4}$$

($A_n$: a random variable with uniform distribution for the magnitude of $V_{SB}$, $\omega_n$: the harmonic of digital switching frequency, $\varphi$: a random variable for the phase shift and uniformly distributed from 0 to $2\pi$)

<table>
<thead>
<tr>
<th>$V_{SB}$ (RMS)</th>
<th>22.9mV</th>
<th>32.6mV</th>
<th>65.2mV</th>
<th>98.6mV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean (V)</td>
<td>0.757159</td>
<td>0.757239</td>
<td>0.756715</td>
<td>0.755619</td>
</tr>
<tr>
<td>Variance</td>
<td>0.000038</td>
<td>0.000077</td>
<td>0.000312</td>
<td>0.000733</td>
</tr>
</tbody>
</table>

Table-2. The mean and variance of $V_m$ along with $V_{SB}$

The mean and variance of $V_m$ of the least significant bit (LSB) of the comparator are simulated by Eq-2 and Eq-4. In this simulation, 0.18 micron technology is used and supply voltage is assumed to be 1.8V. We simulate it for 10 usec (10000 samples) at 1GHz sampling. Figure 9 shows the distribution of $V_m$ along with the level of noise. All magnitudes of noise are RMS values. Table-2 summarized the mean and variance values. From these results, the mean is varying moderately and the variance is increasing as $V_{SB}$ is increasing. It implies...
that the larger $V_{SB}$ generates the larger errors. The distributions are approximately same as Gaussian distribution. For simplicity, we assume the distribution of $V_m$ as Gaussian distribution and that each comparator’s distributions are independent.

Figure 10(a) shows the distributions of three comparators in ADC. $V_m(i-1)$ is the lowest and $V_m(i+1)$ is the highest bit among three comparators. Three comparators’ $V_m$ are changing with noise independently as Gaussian distributions. The assumption is that the input voltage, $V_{in}$ is less than the mean of $V_m(i-1)$. When $V_m(i)$ is in the region 1, the $i^{th}$ comparator produces output 1. This is obviously an error and makes the ADC overestimate the output value. The contrary case makes the ADC underestimate the output value. These errors contribute to the quantization error. This probability of error is determined by Eq-5. Figure 10(b) shows the probability of error defined by Eq-5 in a 6-bit TIQ ADC. We simulated 30 comparators from a LSB comparator. It can be clearly seen that the larger the noise, the greater the probability of error.

$$P_e = 0.5(P_e(\text{region 1}) + P_e(\text{region 2}))$$  
- Eq-5

![Graph of the distributions of $V_m$ and the probability of error](image)

3.3 The strategies for software-driven ADC

In the previous section, the probabilistic error analysis of ADC was discussed. The noise model is set by Eq-4. Also, the probability of error was calculated by Eq-5. From the results, it is clear that the level of noise measured by RMS greatly affects the performance of the ADC. In this section, we discuss how we can control the resolution of the PRA-ADC to avoid the serious performance degradation due to noise. The probabilistic approach previously discussed is identically used in this section except that the noise model of Eq-4 is replaced by a deterministic model extracted from SimplePower as in Figure 5(a). This replacement is possible since the
substrate is modeled as a single low ohmic resistor (less than 1ohm for large digital circuits). Therefore, all ground bounce generated by digital circuits is directly coupled to the substrate [1]. In here, we consider the ground bounce as the only source of the substrate noise; in fact, this ground bounce dominates the total substrate noise[1].

We chose three benchmarks that have different distributions of RMS value. In Figure 11, the RMS of hanoi.c is concentrated on the range from 0.01 to 0.02 and the RMS of heap.c is concentrated on the range from 0.01 to 0.03. The RMS of matmult.c is concentrated on the range from 0.01 to 0.02 as hanoi.c but it has a critical noise range from 0.1 to 0.2. From the result of previous section, we can expect that hanoi.c causes less error than heap.c. In case of matmult.c, the overall probability of error is less than heap.c but it has critical points that may cause the error.

![Figure 11. Noise distributions](image)

In Figure 12, the probabilities of error for each benchmark are sorted by the various resolutions. PRA-ADC uses 5-bit, 6-bit, 7-bit, and 8-bit resolution. We assume that the serious performance deterioration takes place when the probability of error is over $10^{-4}$ (chosen based on bearable error rates in cellular systems [13]). Figure 12(a) is the probability of error for 5-bit resolution and all error rates are under $10^{-9}$. It implies that 5-bit resolution is always available when executing the three benchmarks. Figure 12(b) is the probability of error for 6-bit resolution. In this case, only matmult.c meets this requirement. Other benchmarks should choose 5-bit resolution. Figure 12(c) and (d) are the results of 7-bit and 8-bit resolution. These benchmarks cannot use either of the two resolutions. It implies that PRA-ADC can only support the 5-bit and 6-bit resolution, when it is embedded with SimplePower’s microprocessor core.
However, SimplePower has a pipeline gating mode. When this mode is turned on, SimplePower selectively gates subsets of the pipeline registers that are not used. This scheme is introduced to reduce the power of pipeline registers that takes up to 40% of datapath. Also, this can reduce a great amount of noise; as the synchronous circuits also contribute to a large part of the noise. Figure 12(e) and (f) shows the probability of error of 7-bit and 8-bit resolution when register gating is used. When this mode is activated, we can use always the 7-bit and 8-bit for matmult.c. Thus, we can determine the resolution of the ADC before running software.
Using the augmented SimplePower tool, this approach prevents unexpected errors as well as unexpected power consumption. We call this approach “fixed mode” since the ADC resolution is fixed before running the software.

![Graphs showing probability of error over cycle](image)

**Figure 13. Variation in Probability of error with time**

If we can change the resolution in real time instead of fixing the resolution before running the software, we can obtain more performance from the ADC when higher resolution is required by certain software. Figure 13 represents the probability of error along with the clock cycle. From the result of Figure 12(b), we cannot use 6-bit resolution for heap.c. In Figure 13(a), the error profile of heap.c is shown. If we can dynamically select the resolution based on this profile, we can use the 6-bit resolution except for several durations that are over $10^{-4}$. In case of matmult.c, 7-bit also can be used except during a few durations. In this case, the noise profile generated from the augmented SimplePower tool indicates when the ADC can be used at desired resolution during the execution of a program.

4. CONCLUSION

We have presented new noise managing strategies for adaptive resolution ADC. These strategies are based on the noise analysis using modified SimplePower. Using these strategies, we can reliably control the ADC in the existence of substrate noise. Also, a new model of current was presented to estimate the di/dt noise more accurately. From the results, we show the possibility to control the substrate noise in software level when the analog circuits are integrated with microprocessor core.

In this paper, we assumed that the substrate is a single low ohmic resistor, while in reality the substrate model is varied with the placement schemes of transistors. Using more realistic substrate model base on
the placement schemes, we can provide more useful guidance to design noise-tolerant analog circuits. This will be discussed in our future work.

[REFERENCES]


