READING AND WRITING FLASH MEMORY USING DELTA SIGMA MODULATION

by

Jennifer Taylor

A thesis

submitted in partial fulfillment

of the requirements for the degree of

Master of Science in Electrical Engineering

Boise State University

November 2004

The thesis presented by Jennifer Taylor entitled Reading and Writing Flash Memory using Delta Sigma Modulation is hereby approved:

R. J. Baker 10/27/04

R. Jacob Baker — Date
Advisor

Sin Ming Loo 10/27/04

Sin Ming Loo — Date
Committee Member

Jeff Jessing 10/27/04

Jeff Jessing — Date
Committee Member

John R Pelton 11-12-04

John R. Pelton — Date
Dean, Graduate College

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION TO FLASH MEMORY AND SENSING

The state of a flash memory cell, whether it is a logic high or low, is determined by sensing the current through the cell.[1-3] Today's common design practice is to convert the current to a voltage and then use a differential amplifier or latch to determine whether the flash memory cell has been programmed or erased.[1-9] The differential amplifier or latch scheme for data sensing has a major shortcoming; it is sensitive to process variations and noise, demanding wide threshold voltage margins between the programmed and erased memory cell states. As a result, programming and erase times are long, and it is difficult to store more than one bit of data per memory cell.[10]

This thesis presents a new and improved sense amplifier design based on delta sigma modulation. Flash cell current is converted from analog to digital using a one-bit $\Delta\Sigma$ modulator. The delta sigma modulator cancels out noise and outputs an accurate measurement of average cell current. The data conversion circuitry is simple and less sensitive to noise and process variations, with less margin required between programmed and erased states. This new design approach leads to both more precise sensing and more economical memory chips.

1.1 Flash Cell Structure and Operation

One cell of a flash memory array consists of a single n-channel transistor with two polysilicon gates stacked on top of each other. The bottom gate, which is called the

floating gate, is completely surrounded by dielectrics. This dielectric insulation gives the floating gate the ability to store charge. The bottom dielectric is silicon dioxide, and is called the tunnel oxide. Between the two poly gates is an *interlevel* dielectric consisting of an oxide-nitride-oxide sandwich. The drain junction is intentionally abrupt to facilitate the programming operation, which typically occurs through hot carrier injection. The source junction is intentionally graded to suppress band-to-band tunneling which might occur during the Fowler-Nordheim erase operation. The cross-section and schematic of a single flash cell are illustrated in Figure 1.1.

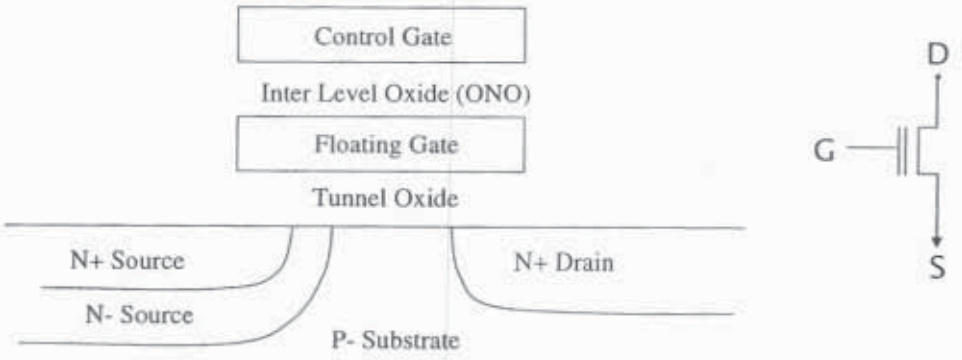


Figure 1.1. Flash Memory Cell Cross-Section (left) and Schematic (right).

Electrons can be stored on the floating gate by a programming operation, and they can be removed from the gate by an erase operation. Programming and erase are accomplished by applying voltages to the control gate, source, substrate, and drain. The state of the flash cell (programmed or erased) is defined by whether or not the floating gate contains electrons. A programmed cell contains electrons on the floating gate, and an erased cell does not contain electrons, as illustrated in Figure 1.2.[1-3]

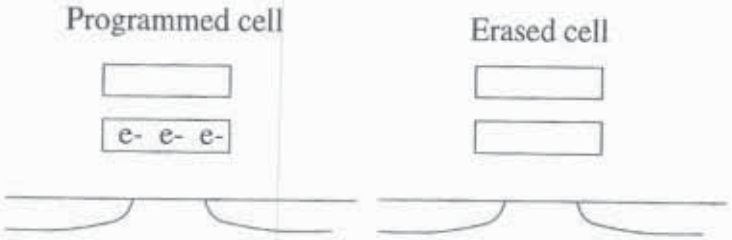


Figure 1.2. Programmed and Erased Flash Memory Cells.

Figure 1.3 illustrates the oxide capacitances associated with the flash cell. Assuming the capacitances of the tunnel oxide and the interpoly dielectric are roughly equivalent and equal to C_{ox}, the total capacitance from the gate to the substrate will be $C_{ox}/2$ (due to the two C_{ox} in parallel). The threshold voltage of a single flash cell can be determined by the following equation:

$$V_{THN} = -\Phi_{ms} - 2\Phi_{FP} + 2 \bullet \frac{Q_{b0}}{C_{ox}} \qquad (1.1)$$

where Φ_{ms} is the contact potential between the bulk and the gate poly, Φ_{FP} is the electrostatic potential of the p-type substrate, Q_{b0} is the charge contained in the channel region, and $C_{ox}/2$ is the gate to substrate capacitance.[11]

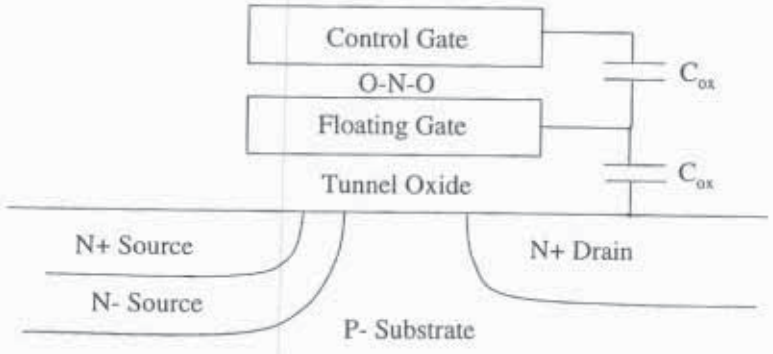


Figure 1.3. Flash Cell Oxide Capacitances

When electrons are stored on the floating gate, the threshold voltage is increased with the trapped charge (Q_{poly1}), as indicated in equation 1.2:

$$V_{THN,prog} = -\Phi_{ms} - 2\Phi_{FP} + 2 \bullet \left(\frac{Q_{b0}}{C_{ox}} + \frac{Q_{poly1}}{C_{ox}}\right) \qquad (1.2)$$

This difference in threshold voltage is used to determine whether a flash cell is programmed or erased.[11]

A flash cell is read by applying approximately 5V to the control gate, 1V to the drain, 0V to the source and substrate, and measuring the current from drain to source. The read operation is illustrated in Figure 1.4. An erased cell's threshold voltage is designed to be less than the 5V applied to the control gate during a read. Therefore, under read conditions, an erased cell will turn on and will draw current from drain to source. In contrast, the electrons added to the floating gate of a programmed cell cause its threshold voltage to be greater than the 5V read voltage. When a programmed cell is read, the cell is turned off, and no current flows from drain to source. Figure 1.5 shows typical I_{DS} curves of an erased bit and a programmed bit.[1-3,11]

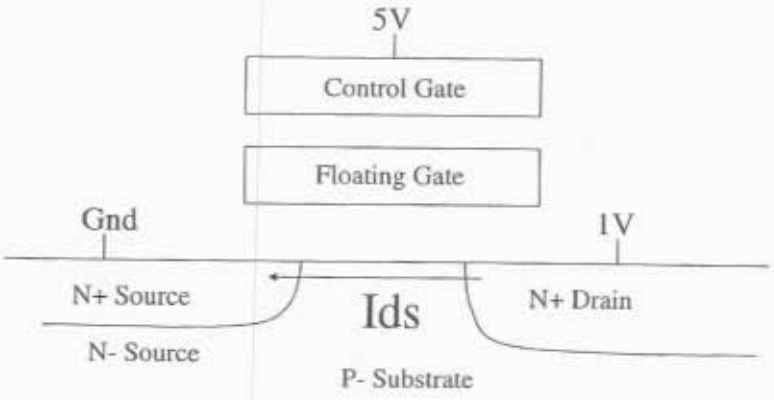


Figure 1.4. Flash Cell under Read Conditions.

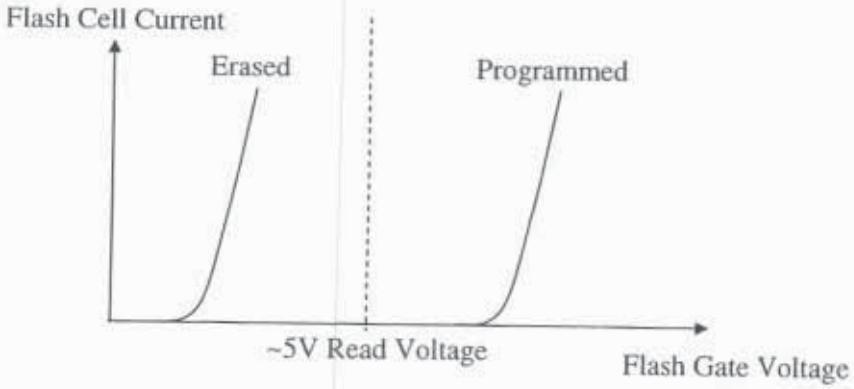


Figure 1.5. Flash Cell I_{DS} Curves.

Figure 1.6 contains an example of the flash cell current over time. The cell current varies due to injected noise, including capacitive coupling from adjacent circuits and V_{DD} or ground fluctuations. As a result of the inherent variation in flash cell current, the margin between programmed and erased states must be sufficiently large in order to prevent a sensing error. The margin between the erased cell's and the programmed cell's threshold voltages is typically 2V or higher. [12]

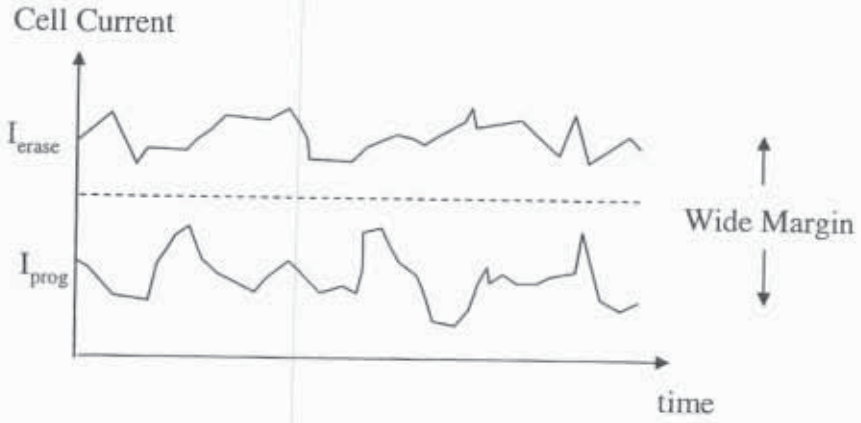


Figure 1.6. Flash Cell Current Variation over Time

Flash cells are arranged in a NOR memory array by columns and rows. The control gates are connected through wordlines, and the drains are connected through bitlines. All of the sources are connected together. The sense amplifiers exist at the bottom of each bitline. When a flash cell is read, appropriate voltages are applied to its wordline and bitline, while the other wordlines and bitlines remain at ground. The flash cell being read will supply current on its bitline if it is erased, and will supply no current if it is programmed. The sense amplifier then detects the current on the bitline and compares it to a known level to decide whether the cell is programmed or erased.[1-3]

A NAND memory array has a similar architecture, with the exception that each flash cell is not connected directly to a bitline, so the flash cell current must be passed through other flash cells to get to the bitline. As a result, in a NAND array, a very high pass voltage is applied to all wordlines except the cell under test's wordline. The cell under test's wordline receives a read voltage that will turn on an erased cell but will not turn on a programmed cell. If the cell is erased, current will flow through all cells and be detected by the sense amp. If the cell is programmed, no current will flow.[11] The NOR and NAND architectures are illustrated in Figures 1.7 and 1.8.

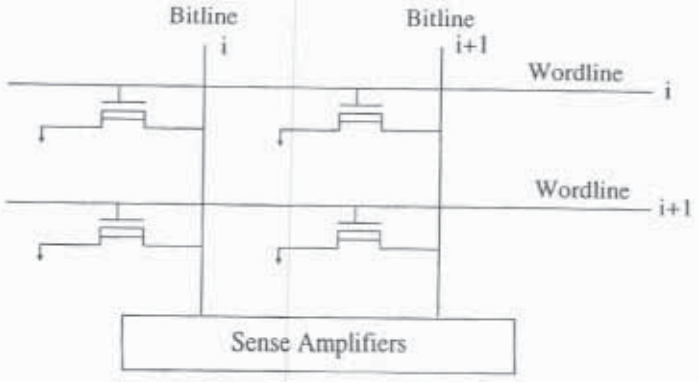


Figure 1.7. NOR Architecture.

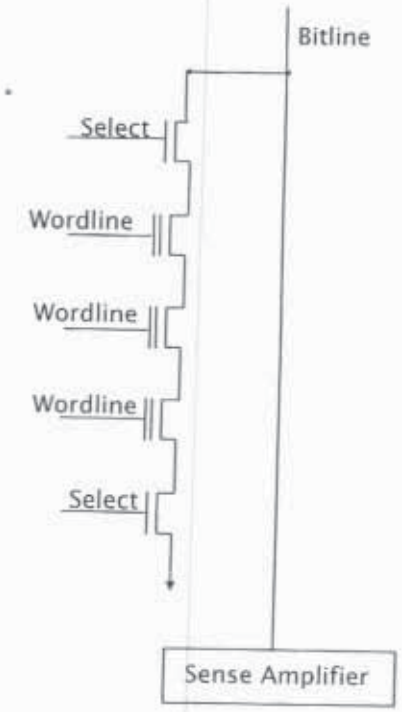


Figure 1.8. NAND Architecture.

1.2 Flash Memory Sense Amplifiers

Common sense amplifiers for NOR flash memory chips are single-ended sensing schemes that compare the potential difference between a voltage generated by the flash cell current and a reference voltage through the use of a differential amplifier.[1-5,6-9] The basic differential amplifier design is illustrated in Figure 1.9. Once the SENSE signal is enabled, the differential amplifier compares the flash bitline voltage to a reference voltage. If the flash bitline voltage is higher than the reference voltage, the DATA OUT signal will be pulled to a logic high. If the flash bitline voltage is lower than the reference voltage, DATA OUT will be pulled to a logic low.

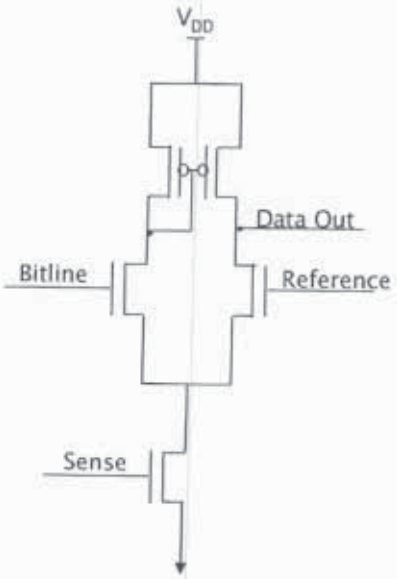


Figure 1.9. Differential Amp Sensing Scheme

One example of a sense amplifier based on a differential amplifier sensing scheme is illustrated in Figure 1.10. This sense amplifier uses parasitic capacitance to integrate the difference between the flash cell current and a reference current. The reference current is generated from another flash cell that has its threshold voltage programmed to a voltage level in between the fully programmed and fully erased states. The parasitic capacitance's voltage is then compared to a reference voltage by a differential amplifier.[9]

This type of sense amp is open-loop and makes its decision at a set point in time. As previously discussed, the flash cell current varies over time due to injected noise. The reference cell's current will also vary over time. The accuracy of this sense amplifier will be affected by many factors including the amount of noise affecting the cell and reference currents, and the sensitivity, gain, offset, and noise immunity of the

differential amplifier. Since this sense amp is based on analog currents and voltages, the circuit elements must be very precise. Further, the amount of margin between the expected flash cell currents and reference current must be sufficiently large to prevent sensing errors.

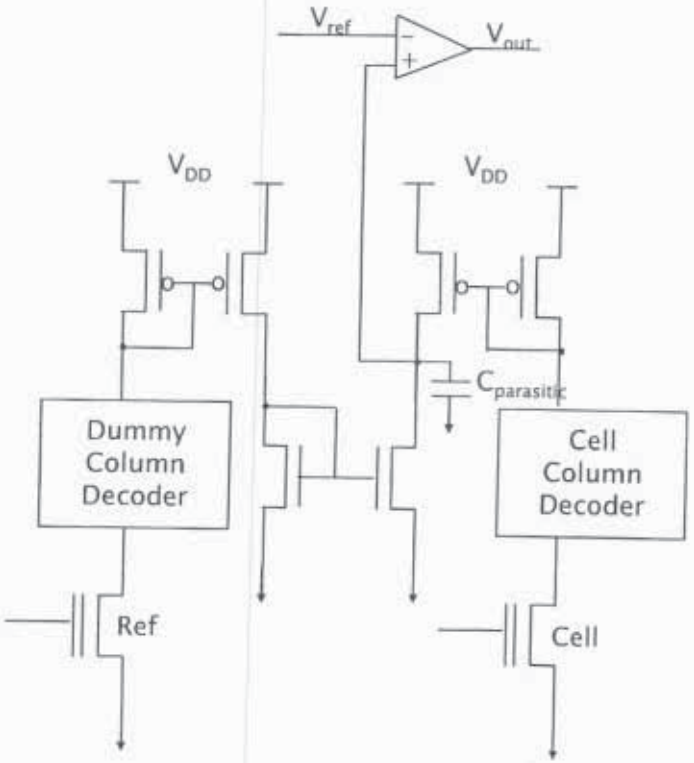


Figure 1.10. Example Sense Amplifier [9]

The commonly used sense amp in NAND architectures consists of a basic latch, as illustrated in Figure 1.11. During the set-up for the sense, the SENSE RESET signal is asserted and pulls the left side of the latch to a logic low. Once the SENSE signal is asserted, if the NAND cell under test is erased, current will flow and discharge the flash cell's bitline, causing the state of the latch to flip.[6] This type of sensing scheme is also open loop, making its decision at an isolated point in time. The sense amp must be

designed with very good noise immunity to prevent injected noise from causing the latch to incorrectly flip states and result in a sensing error.

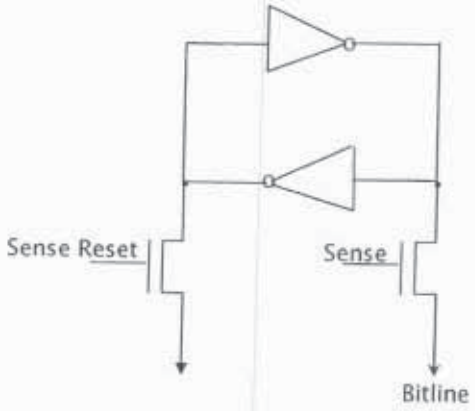


Figure 1.11. Latch Sensing Scheme.

The remainder of this thesis discusses the use of ΔΣ modulation to measure a flash cell current. The ΔΣ modulation cancels out the noise causing cell current fluctuations and provides a measurement of the average cell current, as depicted in Figure 1.12. Use of ΔΣ modulation in the sense amp alleviates the need for high precision circuitry that is necessary to provide noise immunity and accurate analog comparisons in conventional sense amplifiers.

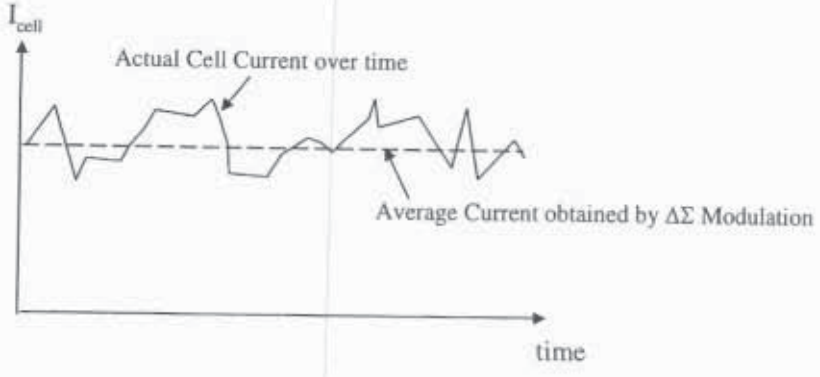


Figure 1.12. Actual Cell Current versus Average Current obtained by ΔΣ Modulation

CHAPTER TWO

THE BASICS OF ANALOG TO DIGITAL CONVERSION AND ΔΣ MODULATION

The sensing operation in a flash memory array is essentially an analog to digital conversion. The flash cell current is analog; it is continuous in time and may vary slightly over the sensing period. A sense amplifier must then convert this analog signal into a digital '0' or '1' (an erased state or a programmed state). This chapter reviews the basics of analog to digital conversion and ΔΣ modulation.

2.1 Analog to Digital Conversion

The block diagram in Figure 2.1 illustrates the analog to digital conversion process, and Figures 2.2-2.4 contain plots of the signals along each point in this block diagram. The ADC process consists of a sample and hold block to sample the signal and prevent it from fluctuating while the conversion is taking place, and then the actual analog to digital converter.[13,14]

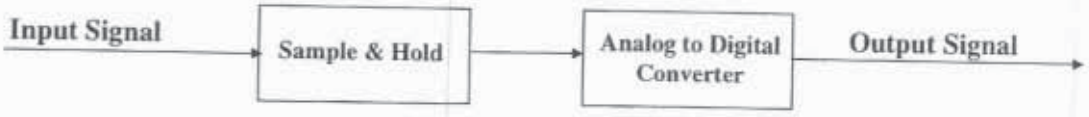


Figure 2.1. Block Diagram of Analog to Digital Conversion Process.

The plot in Figure 2.2 shows the analog input signal (flash cell current) that will be converted to a digital signal. For ease of simulations, the flash cell current can be

converted to a voltage prior to the analog to digital conversion. The input signal should not vary significantly; however for the purposes of this discussion we will assume the analog signal varies between 0 and 1.024V in a sinusoidal function. The value of 1.024V is purposely chosen to correspond to a 10-bit digital representation.

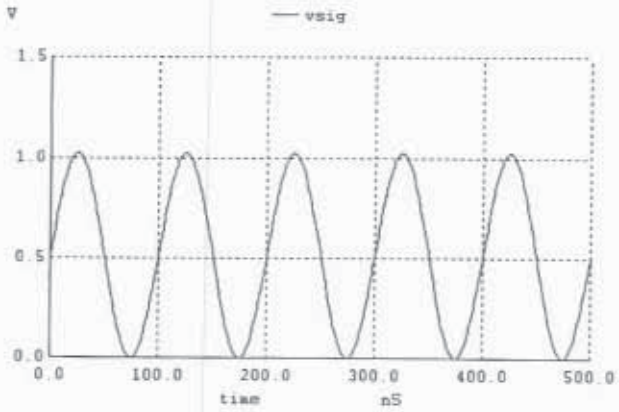


Figure 2.2. Example input signal varying between 0 and 1.024V.

The plot in Figure 2.3 shows the signal after passing through the sample and hold block. The sample and hold allows the signal to change only at periodic intervals corresponding to a clock signal. Changes in the input when the clock signal is OFF are ignored (because the signal is held).

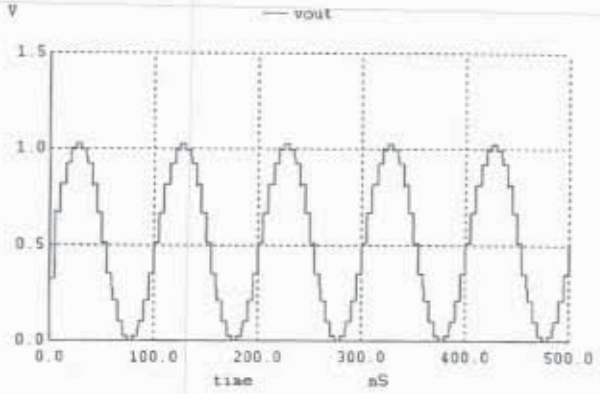


Figure 2.3. Example Output of Sample & Hold.

The ADC block then converts the static input signal into a digital representation, or an integer value between 0 and 1023 for each sampled signal as indicated by the flat portions in Figure 2.3. This digital conversion results in an error called the *quantization error*. As a specific example, analog signals of 0.5001V and 0.5004V will both be represented by the digital signal of 500. The difference, or resolution, between 0.5001V and 0.5004V is lost in this conversion.

Any digital representation can have a maximum error of +/- ½ LSB (Least Significant Bit), which is half the distance between adjacent quantization levels. For a 10-bit digital signal with 1.024V range, the quantization levels reside .001V apart, and the maximum error can be .0005V. Thus, the digital signal consists of the original analog signal plus the quantization error. The plot in Figure 2.4 shows the quantization error for this example. Quantization error normally will be white, or random, noise. In this case, the quantization error has a distinct pattern due to the input signal being a sinusoidal waveform. The quantization error can be reduced by increasing the number of bits in the digital sample. For example, a 12-bit digital signal with the same 1.024V range will result in a maximum error of .000125V versus an error of .0005V with a 10-bit digital signal.

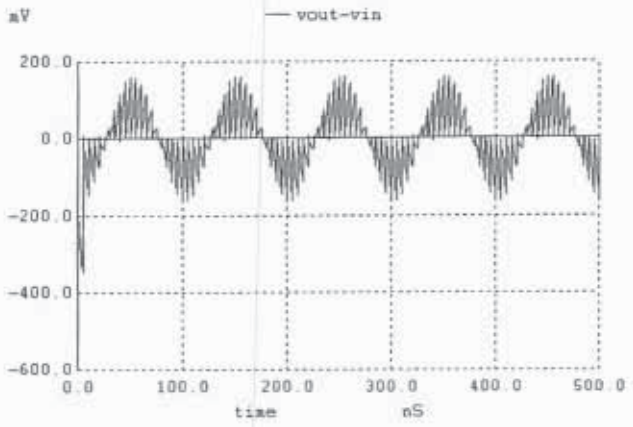


Figure 2.4. Example Quantization Error.

The analog to digital conversion process can be modeled as the summation of the input signal and the quantization noise. This is illustrated in Figure 2.5.

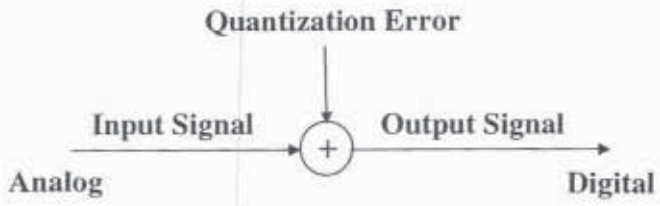


Figure 2.5. Model of ADC Process.

Analog to Digital Converters are typically characterized by their Signal to Noise Ratio (SNR). SNR is a measure of how well the converter passes the input signal while surpressing the quantization noise. Continuing our example with an input sinusoidal signal, the SNR is expressed as:

$$SNR_{ideal} = 20 \cdot \log \frac{V_p / \sqrt{2}}{V_{LSB} / \sqrt{12}} \qquad (2.1)$$

For our example, the SNR of our sinusoidal signal with a 10-bit digital representation can be calculated as:

$$SNR_{ideal} = 20 \cdot \log \frac{0.512V/\sqrt{2}}{0.001V/\sqrt{12}} = 62dB \qquad (2.2)$$

Another commonly referred to characteristic of a data converter is the number of effective bits, defined as:

$$N_{eff} = \frac{SNR_{meas} - 1.76}{6.02} \qquad (2.3)$$

This equation would be applied when the SNR is measured, instead of calculated as in the above discussion.

2.2 ΔΣ Noise Shaping Modulators

Quantization error is not the only error introduced into the signal during the analog to digital process. In a real circuit, the analog to digital conversion accuracy will be affected by the gain, offset, and linearity of the converter. Additionally, noise from adjacent circuit blocks may cause some amount of periodic corruption to the input signal. To overcome the possibility of an incorrect digital output, either the analog to digital converter must be designed with a high degree of precision, or alternatively a noise-shaping data converter, such as the ΔΣ converter, may be used.

The simplest ΔΣ modulator is the first-order lowpass modulator, shown in Figure 2.6. The input is an analog, continuous signal, and the output is a discrete-time, binary signal. Example input and output signals are shown in Figure 2.7.

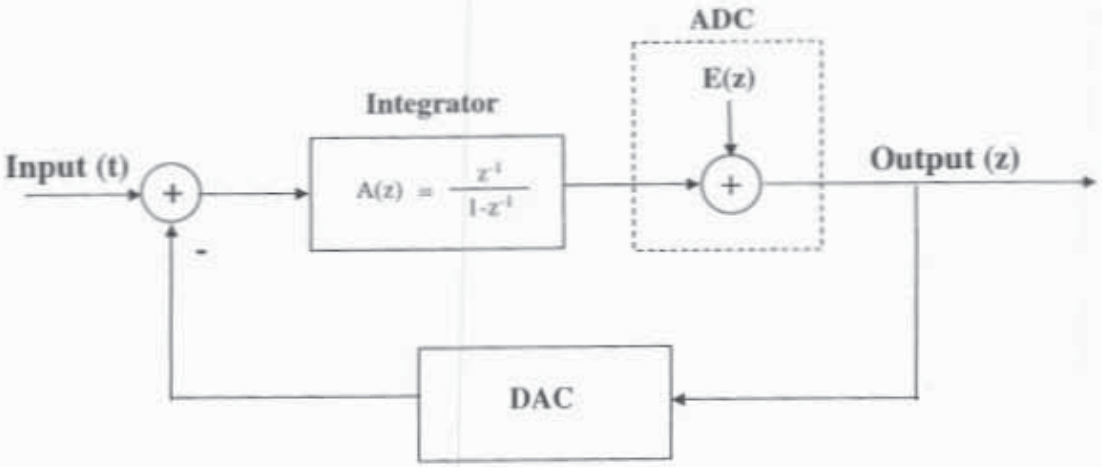


Figure 2.6. First order lowpass ΔΣ modulator.

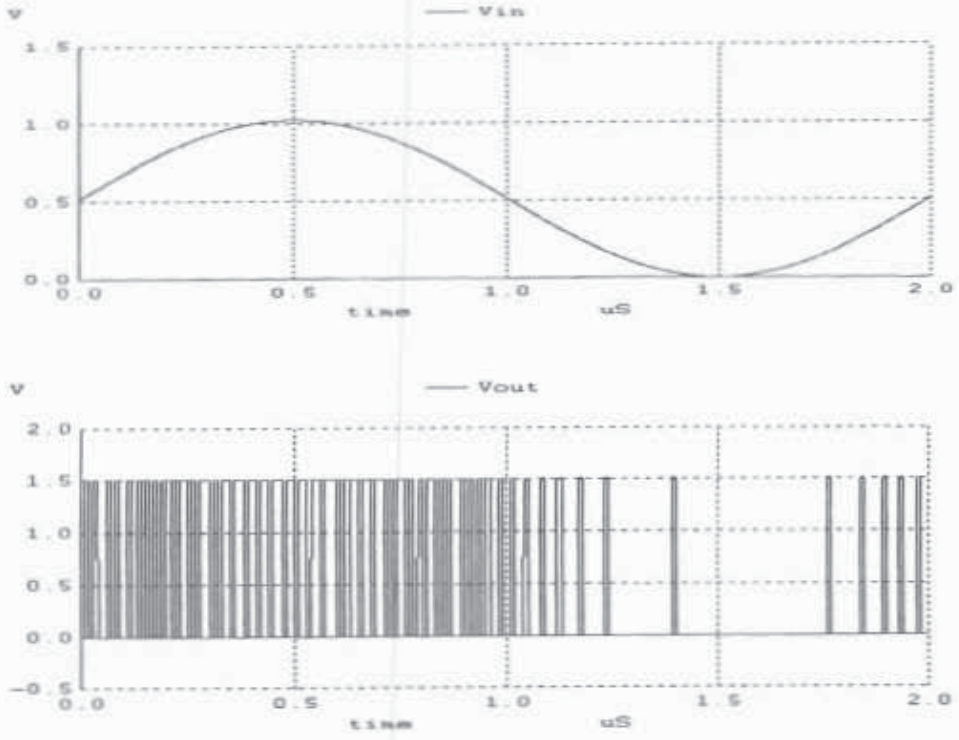


Figure 2.7. Typical input and output signals of a ΔΣ modulator.

From Figure 2.6, we can discuss the operation of the ΔΣ modulator. The circuit calculates the difference between the input signal and the delayed output signal, and integrates this difference. The output of the integration is then fed to a summer where it is added together with quantization noise from the ADC. The output of the summer is the output signal. The result is that the output tracks the average value of the input. From the diagram in Figure 2.6, the following equations can be written:

$$[In(z) - Out(z)] \bullet \frac{z^{-1}}{1-z^{-1}} + E(z) = Out(z) \qquad (2.4)$$

$$Out(z) = z^{-1}In(z) + (1-z^{-1})E(z) \qquad (2.5)$$

This shows that the output signal consists of:

- the input signal delayed by one clock cycle
- the quantization noise (error) differentiated.

The differentiation of the quantization noise acts to push the quantization noise to higher frequencies. This is why ΔΣ modulation is also often referred to as noise-shaping. The effect is a reduction of quantization noise in one frequency band at the expense of the remaining frequencies.

As discussed in section 2.1, the quantization noise for a time-varying input signal is random, and therefore its voltage spectrum is flat. The noise voltage spectral density can be calculated as

$$V_{QE} = \frac{V_{LSB}}{\sqrt{12 \cdot f_s}} \qquad (2.6)$$

where V_{LSB} is the voltage difference between adjacent quantization levels, and f_s is the sampling frequency. A graphical representation of the quantization noise spectral density is shown in Figure 2.8.

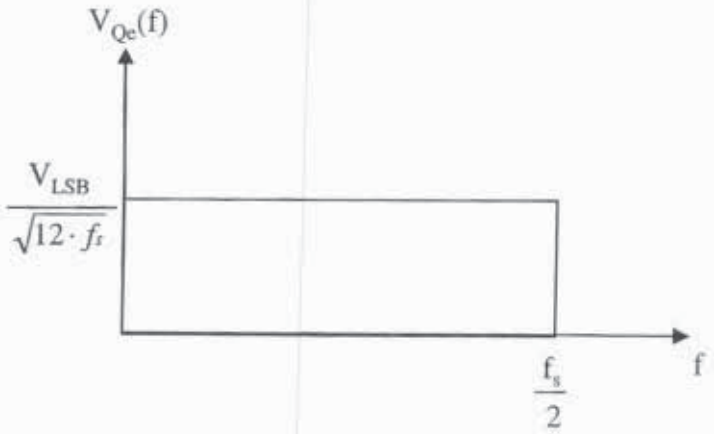


Figure 2.8. Quantization Noise Spectral Density

The differentiation of the quantization noise is called the modulation noise, and can be calculated as:

$$\text{Modulation Noise} = (1 - z^{-1}) \cdot V_{Qe}$$

$$= (1 - e^{-j2\pi(f/f_s)}) \cdot \frac{V_{LSB}}{\sqrt{12 \cdot f_s}} \qquad (2.7)$$

The power spectral density of the modulation noise can be written as:

$$= 2 \cdot (1 - \cos 2\pi \frac{f}{f_s}) \frac{V^2_{LSB}}{12 f_s} \qquad (2.8)$$

A graphical representation of the modulation noise spectral density is shown in Figure 2.9. Comparing Figures 2.8 and 2.9, we can see that the noise is pushed to higher frequencies, similarly to being processed through a highpass filter.

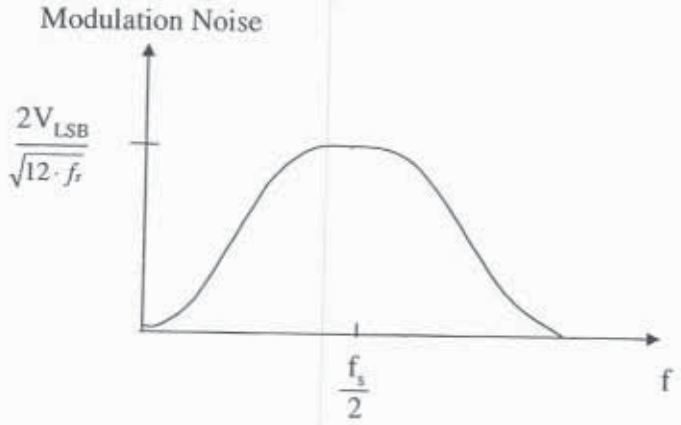


Figure 2.9. Modulation Noise Spectral Density

At frequencies approaching DC, the modulation noise is reduced to zero. This is ideal for the case of a flash sense amplifier, because the input signal to be digitized is expected to be a DC signal.

The noise value of $V_{LSB}/\sqrt{12}$ is applicable only for the specific case of a time-varying input signal with random quantization noise in continuous operation. Chapter 3 discusses the use of a $\Delta\Sigma$ modulator with a single bit output to sense a flash cell current. In the simple $\Delta\Sigma$ modulator design proposed in chapter 3, the quantization noise, or minimum resolution, will be V_{SIG}/K, where K is the number of clock cycles, which is also the number of data samples collected.

The SNR of the single bit output $\Delta\Sigma$ modulator used in a sensing operation can then be written as

$$SNR \approx 20 \cdot \log \frac{Signal}{Signal / K} \qquad (2.9)$$

Suppose the ΔΣ modulator makes a decision after 500 clock cycles, or after 500 data points have been collected. The SNR will then be

$$SNR \approx 20 \bullet \log \frac{Signal}{Signal / K} = 20 \bullet \log (500) = 54dB \qquad (2.10)$$

In order to get a relative comparison of performance between the single bit output ΔΣ modulator with conventional ADCs, we can calculate Number of Effective Bits to see that with only one bit, the simple ΔΣ modulator is able to achieve a similar SNR value as a more complex ADC with a digital output of over 8 bits!

$$N_{eff} = \frac{SNR_{meas} - 1.76}{6.02} = 8.7 \text{ bits} \qquad (2.11)$$

An additional advantage of the single bit output ΔΣ modulator is that the analog to digital conversion is inherently linear. Since there are only two possible output codes with a one-bit converter ('1' and '0'), the two analog voltages representing these output codes will always fit a straight line. In a multi-bit converter, linearity can be a concern and an additional source of error. [13,14]

CHAPTER THREE

ΔΣ SENSE AMPLIFIER DESIGN AND OPERATION

In this chapter, the design of a sense amplifier for flash memory using ΔΣ modulation is discussed. A very simple topology is presented to illustrate how the ΔΣ noise shaping can compensate for an imprecise comparator, in contrast to the accurate sensing circuits needed without noise shaping. The circuit operation is explained, and several simulations are performed to demonstrate the sense amp's capability in the presence of various noise sources.

3.1 Sense Amplifier Block Diagram

A block diagram of the sense amplifier circuit is shown below in Figure 3.1. As discussed in chapter two, a ΔΣ noise modulation topology consists of an integrator, an analog to digital converter, and a digital to analog converter. For this sensing circuit, the integrator is implemented as a capacitor, the ADC is implemented as a simple clocked comparator, and the DAC is implemented as a current source and switch controlled by the output of the ADC.

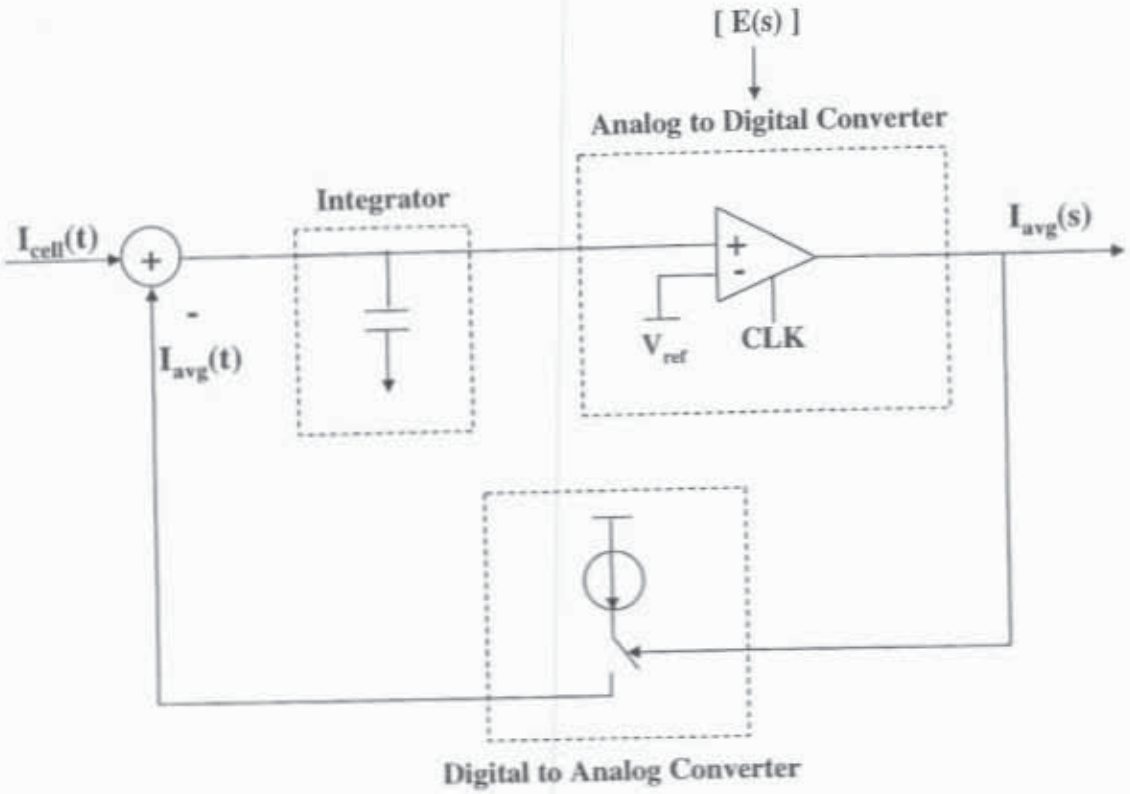


Figure 3.1. Block Diagram of ΔΣ Sensing Circuit.

From Figure 3.1, the following equations can be derived:

$$\frac{I_{cell} - I_{avg}}{sC} + E(s) = I_{avg} \tag{3.1}$$

$$\frac{I_{cell}}{sC} + E(s) = I_{avg}\left(1 + \frac{1}{sC}\right) \tag{3.2}$$

$$I_{avg} = \frac{I_{cell}}{sC + 1} + E(s) \cdot \frac{sC}{sC + 1} \tag{3.3}$$

Thus as expected, the output of the ΔΣ sense amp, I_{avg}, is equivalent to the desired cell current subjected to a lowpass filter, combined with the quantization noise subjected to a highpass filter.

3.2 Sense Amplifier Circuit Design

A simple transistor-level implementation for the block diagram of Figure 3.1 is shown below in Figure 3.2.

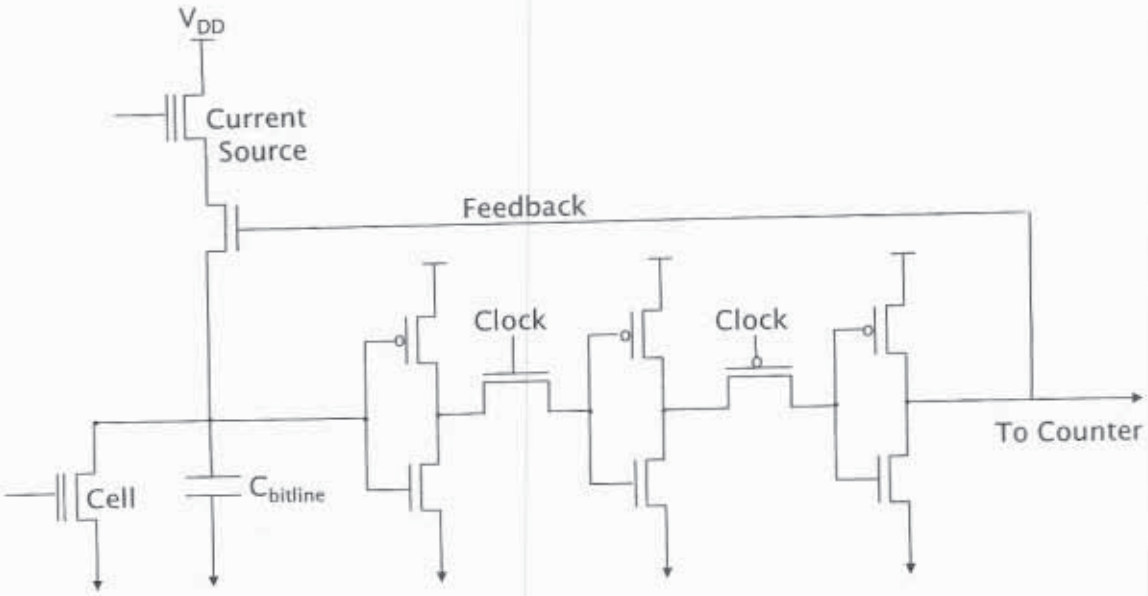


Figure 3.2. Transistor Level Diagram of ΔΣ Sensing Circuit

3.2.1. Integrator

The integrator consists simply of the bitline capacitance of the flash memory array. While the bitline capacitance of a memory array will vary depending on cell design, process technology, and number of cells per column, the bitline capacitance is assumed to be 5pF for this design. This capacitance is initially charged up to a reference voltage of 0.5V, which is chosen to be close to the switching point of the clocked comparator. The bitline capacitance does not need to be pre-charged, as the ΔΣ sensing circuit itself can charge up the bitline capacitance to the comparator switching point; however, this would add some delay to the sensing operation.

At the beginning of the sense, the flash cell current will begin to discharge the bitline capacitance until the voltage at that node reaches the switching point of the comparator. Once the comparator switches, the feedback circuit (discussed in more detail in section 3.2.3) will supply sufficient current to cause the bitline capacitance to charge back up to a voltage level above the comparator switching point. Expected discharge times for a few flash cell current values are shown below, by solving:

$$\frac{I_{cell}}{C_{cell}} = \frac{\Delta V_{bitline}}{T} \qquad (3.4)$$

with C_{cell} = 5pF, and $\Delta V_{bitline}$ = 0.5V (complete discharge).

I_{cell}	Discharge Time
1μA	2.5μs
30μA	83ns
60μA	42ns

In order to avoid completely discharging the bitline capacitance (and losing the benefit of averaging over time), the circuit cannot be allowed to discharge for more than 42ns in one clock cycle, for a maximum flash cell current of 60μA. For this design, a 100MHz clock is chosen, giving over 30ns of margin. While the discharge rate of the capacitor sets the minimum clock frequency, there is no limitation for maximum clock frequency.

It is worth noting that the voltage on the bitline, and consequently the voltage applied to the drain of the flash cell, will vary over time as the capacitance charges up and discharges around the comparator switching point. This will cause the flash cell current to fluctuate slightly. This is not a problem since the *average* value of the bitline

capacitance will be a constant, and we are seeking to determine the corresponding average value of the flash cell current.

3.2.2. Analog to Digital Converter

The clocked comparator consists of three inverters and two switches. This simple comparator is used to illustrate the point that the comparator does not have to be very precise for the ΔΣ noise shaping to work. The comparator can occasionally make wrong decisions, and the error will average out over the sensing time.

The NMOS and PMOS switches are sized as 10/1 and 20/1 respectively. The transistors in the inverters are sized as 10/10 NMOS and 20/10 PMOS. These transistors are made slightly long in order to keep power consumption lower. The comparator's decision point will correspond to the switching point of the first inverter. In this design, the switching point is approximately 0.5V, as shown in the SPICE simulation in Figure 3.3 below.

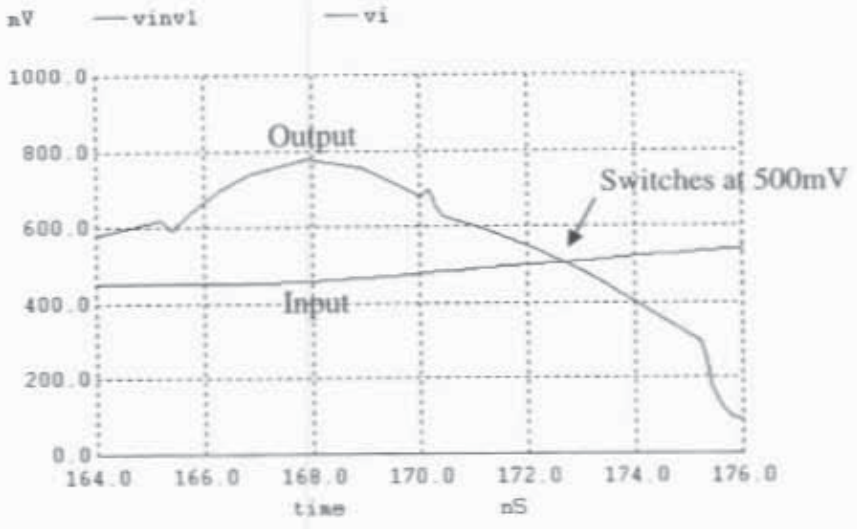


Figure 3.3. SPICE simulation showing comparator switching point.

The two switches controlled by CLOCK act together to serve as an edge-triggered latch, and are needed to provide a digital signal. The three inverters provide sufficient gain to restore the output signal to full voltage levels. Figures 3.4-3.6 show the voltage after each inverter stage of the comparator. As can be seen, the output has reached fairly clean logic levels by the final inverter.

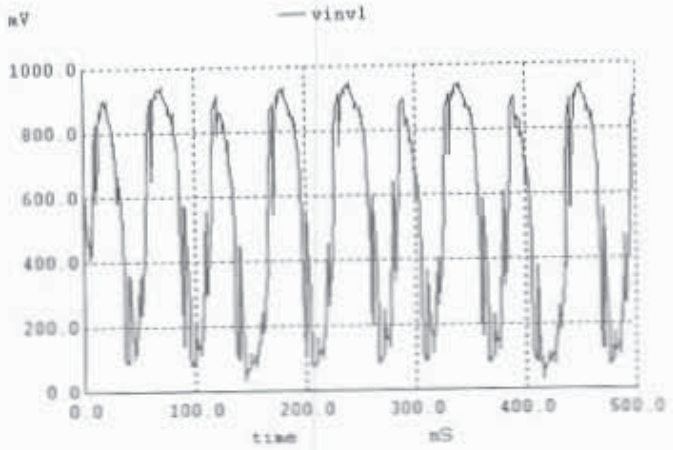


Figure 3.4. Voltage at the output of first comparator stage.

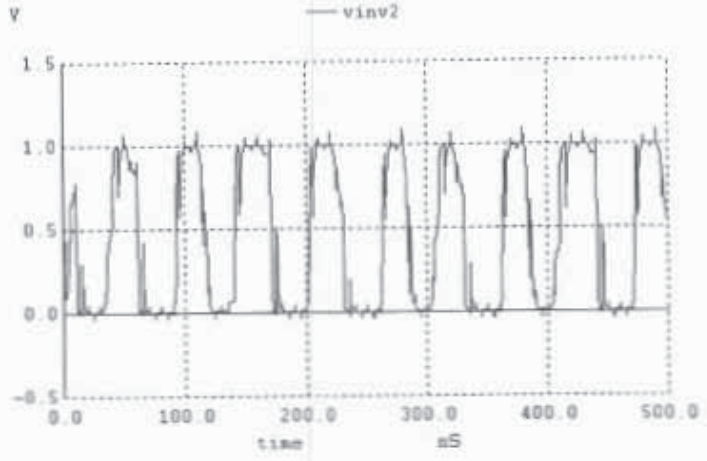


Figure 3.5. Voltage at the output of second comparator stage.

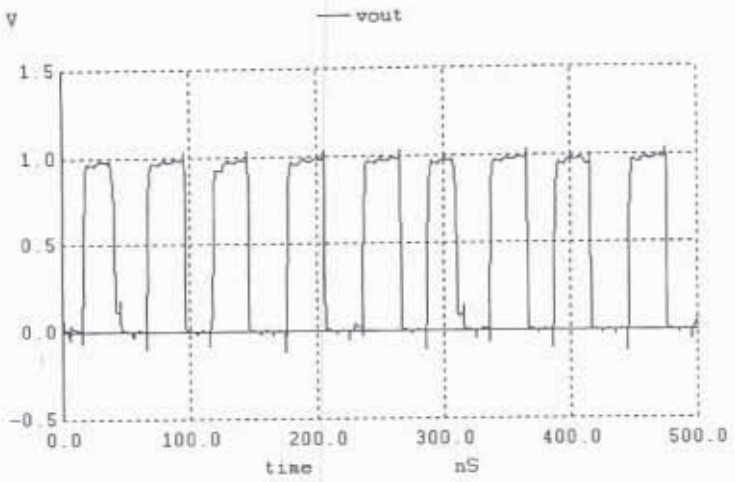


Figure 3.6. Voltage at the output of third and final comparator stage.

3.2.3. Digital to Analog Converter (Feedback Circuit)

The feedback circuit consists of a flash cell for the current source, and a transistor controlled by the output of the clocked comparator. When the sensing is initiated, the current source is disconnected and the flash cell being sensed causes the bitline capacitance to discharge. Once the bitline capacitance voltage reaches the switching point of the clocked comparator, the feedback switch is enabled and the feedback current is connected to the bitline capacitance, causing it to charge back up to a voltage level higher than the comparator switching point. Thus the feedback current must be selected at a level at least as high as the highest flash cell current expected. For this design, a feedback current of 60μA is selected, and the expected flash cell current values will range from 0μA to ~50μA. The precise value of this current is not important, nor is it important to be consistent across a wafer.

For processing simplicity, a flash cell is used in this design for the feedback current. This flash cell would need to be erased to the desired level during wafer probe testing. Another choice for feedback current would be a switched capacitor. A switched capacitor would result in lower power, but requires two non-overlapping clock signals (which are not otherwise required), and may also require increased layout area.

3.3 Sense Amplifier Operation

3.3.1. Sense Amplifier Operation

In this design, the sense amplifier is clocked at 100MHz. Thus, every 10ns, there is a constant amount of charge removed from the bitline capacitance due to the current from the flash cell being sensed. This charge can be calculated by:

$$Q_{bit} = I_{cell} T \qquad (3.5)$$

For example, Q_{bit} for a cell drawing 30μA of current is 300fC when T = 10ns.

The rate at which charge is added to the bitline capacitance by the feedback current is not constant during each clock cycle, since the feedback current is sometimes enabled and sometimes disabled. If the bitline capacitance voltage is less than the switching point of the comparator, then the feedback circuit will be enabled and the feedback current will be added to the bitline capacitance for one clock cycle. If the bitline capacitance voltage is greater than the switching point of the comparator, then the feedback circuit will be disabled, and the flash cell current will continue to discharge the bitline capacitance. A decision is made against the comparator switching point every clock cycle (every 10ns). If N is the total number of clock cycles used in the sense, and

M is the number of times the feedback current is enabled, then the charge added to the bitline capacitance can be calculated as:

$$Q_{feedback} = I_{feedback} \cdot \frac{M}{N} \cdot T \tag{3.6}$$

If the circuit is operating properly, the average feedback current will equal the flash cell current, so we can write:

$$Q_{feedback} = I_{feedback} \cdot \frac{M}{N} \cdot T = Q_{bit} = I_{cell} \cdot T \tag{3.7}$$

$$\frac{I_{cell}}{I_{feedback}} = \frac{M}{N} \tag{3.8}$$

$$I_{cell} = \frac{M}{N} \cdot I_{feedback} \tag{3.9}$$

Using the above equations, we can see that the minimum I_{cell} value that can be sensed, and the minimum resolution, corresponds to M=1 (one high output code during the entire sensing time). For this design, the sensing time was selected as 5μs, and the feedback current is 60μA. Thus the minimum resolution is:

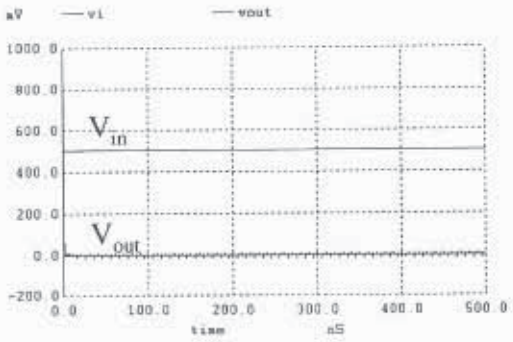
$$\frac{1}{500} \cdot 60\mu A = 120nA \tag{3.10}$$

3.3.2 Sense Amplifier Simulations

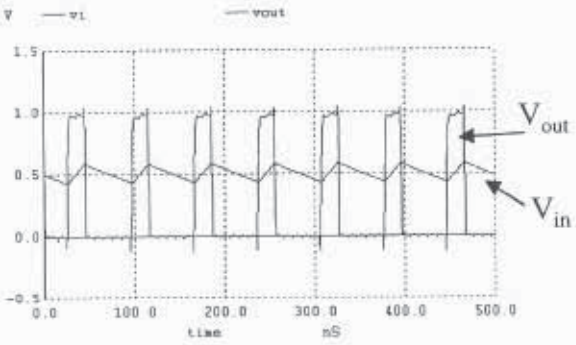
Simulations for the design in Figure 3.2 were performed with SPICE using 50nm BSIM4 models and a V_{DD} supply voltage of 1.0V. The appendix contains the netlist used to generate the simulations contained in this chapter.

The first set of simulations show the bitline capacitance voltage (vi) and the output of the comparator (vout) for several different flash cell current values. As seen in Figure 3.7 below, when the bitline capacitance voltage drops below approximately 0.5V, the comparator output is enabled, and the bitline capacitance charges back up to a value above the comparator switching point. The higher the flash cell current, the faster the bitline capacitance discharges, and the more often the feedback current must be enabled to maintain the constant ~0.5V bitline capacitance voltage. For the case of 0μA flash cell current, the bitline capacitance does not discharge within the sensing time, and therefore the feedback circuit is never enabled.

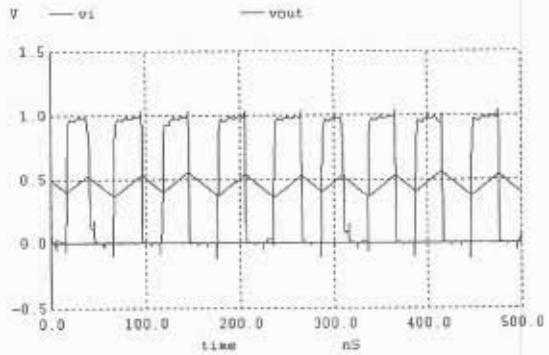
I_{cell} = 0μA



I_{cell} = 15μA



I_{cell} =30μA



I_{cell} = 45μA

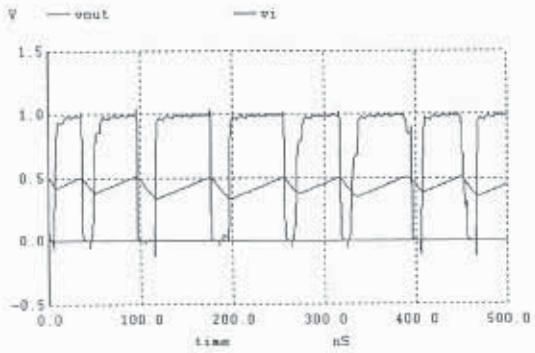


Figure 3.7. SPICE simulations of circuit operation with varying flash cell currents.

Additional simulations were performed with varying flash cell currents out to 5000ns to obtain the graph in Figure 3.8. The relationship between the number of times the comparator is enabled and the flash cell current is linear.

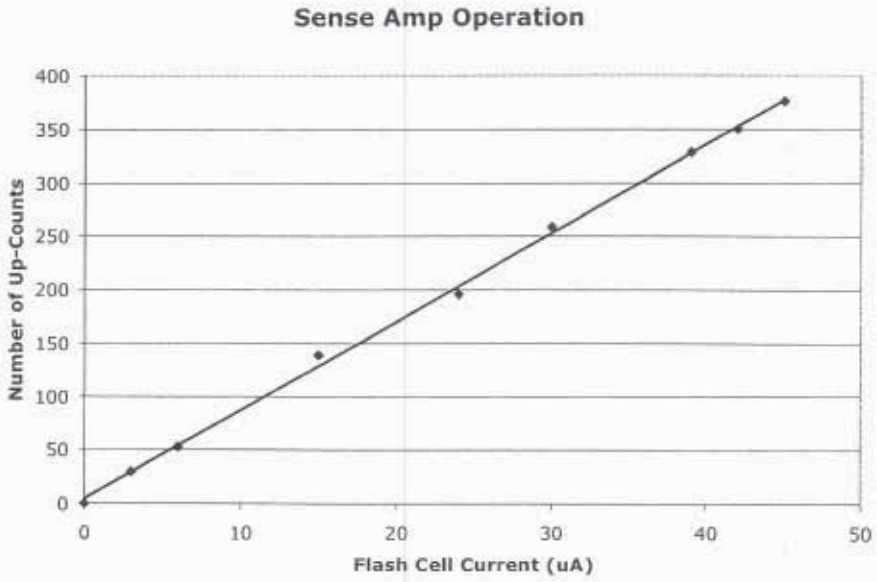


Figure 3.8. Number of times feedback is enabled versus flash cell current.

Table 3.1 contains the simulated cell currents, resulting number of comparator pulses, and calculated cell currents based on the formula:

$$I_{cell} = \frac{M}{N} \cdot I_{feedback} \qquad (3.11)$$

The error can be decreased by increasing the sensing time.

Table 3.1. ΔΣ sense amp operation for varying cell currents.

Cell Current (uA)	Simulated # Pulses	Calculated Cell Current (uA)	Error (uA)
0	0	0	0
3	30	3.7	0.7
6	53	6.5	0.5
15	139	17	2
24	196	24	0
30	259	31.7	1.7
39	328	40.1	1.1
42	350	42.9	0.9
45	376	46	1

The current consumed by this sense amplifier was simulated at around 10μA average; with peaks as high as 35μA, as shown in Figure 3.9. The current shown in the plot is consumed by the comparator and does not include the constant flash cell current or the feedback current.

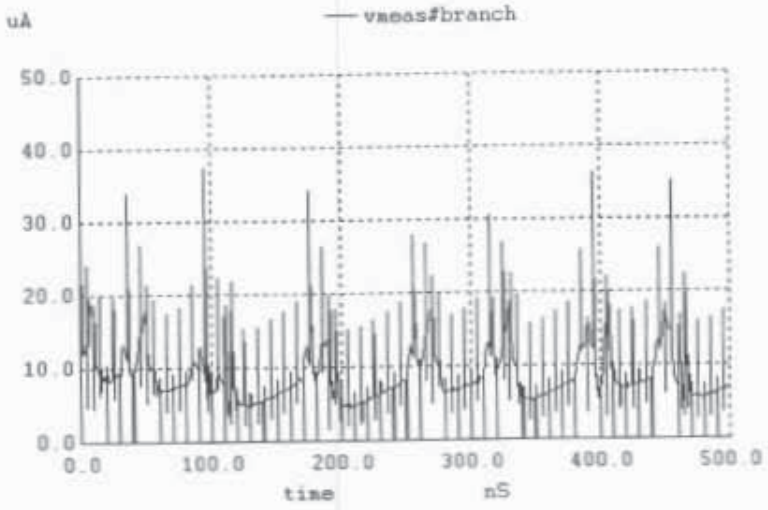


Figure 3.9. Current consumption during sensing operation.

3.3.3 Noise Simulations

3.3.3.1 White Noise

To demonstrate the effectiveness of the ΔΣ noise shaping in this sense amplifier, SPICE simulations were performed with the addition of noisy signals. The first case investigated is the injection of white thermal and flicker noise from surrounding circuits. For the purposes of modeling in SPICE, this noise was simulated by setting the flash cell current to a sinusoidal signal instead of a DC value. Because the average of a sinusoidal signal equals zero, it is easy to predict that the ΔΣ sense amplifier will function very well under white noise conditions, since the sense amplifier functions by averaging the flash cell current over the entire sensing time.

Simulations were performed with sinusoidal flash cell currents ranging from 0uA to twice the intended DC value at 100MHz, as illustrated in Figure 3.10. This amount of noise is obviously much more severe than would occur in a flash memory circuit, but demonstrates the sense amplifier's capability to filter out random noise. Table 3.2 includes the simulation results for two different currents; in both cases, the error produced by the sinusoidal noise is less than 1μA. A similar noise source applied to a current being sensed with a differential amplifier or latch would cause a sensing error, unless there was so much margin built in between the programmed and erased states that the diff amp or latch would not be incorrectly flipped.

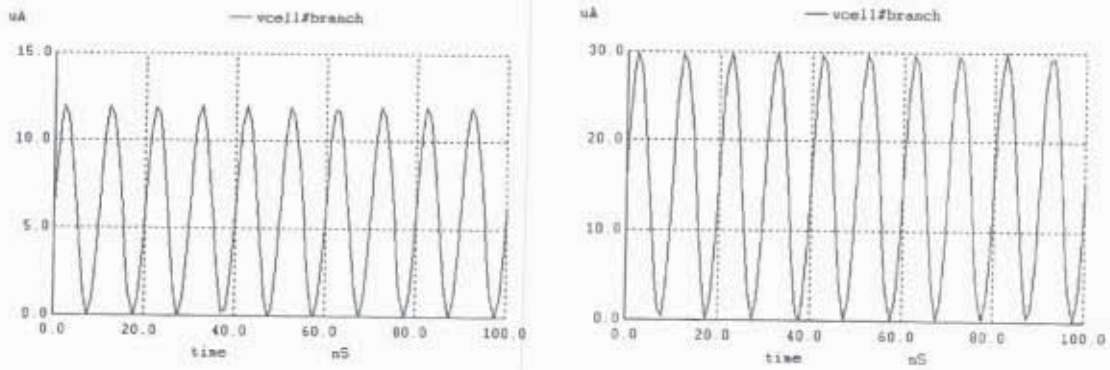


Figure 3.10. Flash cell currents modeled as sinusoidal signals (12μA and 30μA peak-peak)

Table 3.2. ΔΣ Sense amp outputs for "white noise" simulation.

Cell Current	Noise Added	Simulated # Pulses	Calculated Current	Current without noise
6uA	+/- 6uA	58	7.1uA	6.5uA
15uA	+/- 15uA	142	17.4uA	17uA

3.3.3.2 V_{DD} and Ground Bounce

Another common noise signal affecting sense amplifier operation is V_{DD} and ground bounce. This noise condition was simulated by adding a 20mV to 50mV 100MHz sinusoidal signal to the V_{DD} and ground signals, as shown in Figure 3.11.

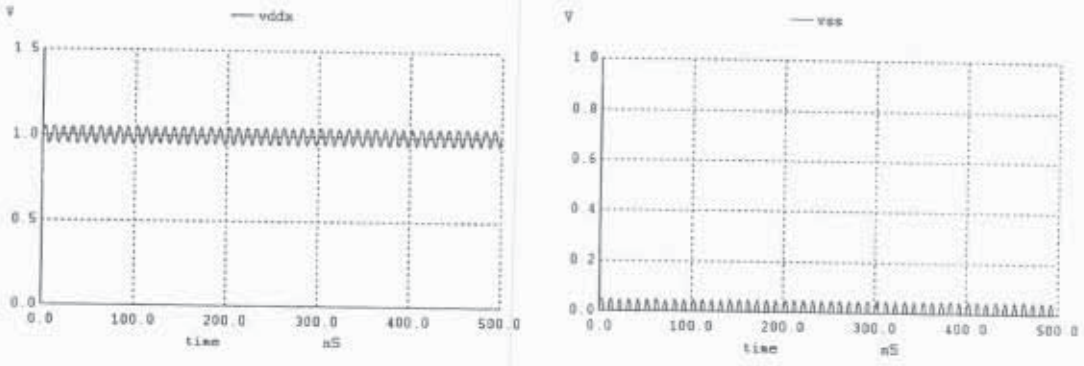


Figure 3.11. Simulations of V_{DD} and ground bounce.

The sense amplifier performed very well with 50mV V_{DD} and ground bounce, as the results in table 3.3 demonstrate. The 50mV noise added between 100-300nA noise to the signal, which is certainly acceptable even if the desired resolution is on the order of a 3-4 μA.

Table 3.3. SPICE results with simulated V_{DD} and ground bounce.

Simulation	Cell Current (uA)	Simulated # Pulses	Calculated Current (uA)	Current with no noise (uA)
20mV VDD bounce	3	31	3.8	3.7
50mV VDD bounce	3	33	4	3.7
50mV VDD bounce	42	352	43.1	42.9
20mV GND bounce	3	31	3.8	3.7
50mV GND bounce	3	31	3.8	3.7
50mV GND bounce	42	352	43.1	42.9

3.3.3.3 Bitline Capacitive Coupling

A third common source of noise is capacitive coupling of the bitline. This was simulated by briefly connecting the bitline to a "noise" source during the sense. The modified circuit used for the simulations is shown in Figure 3.12.

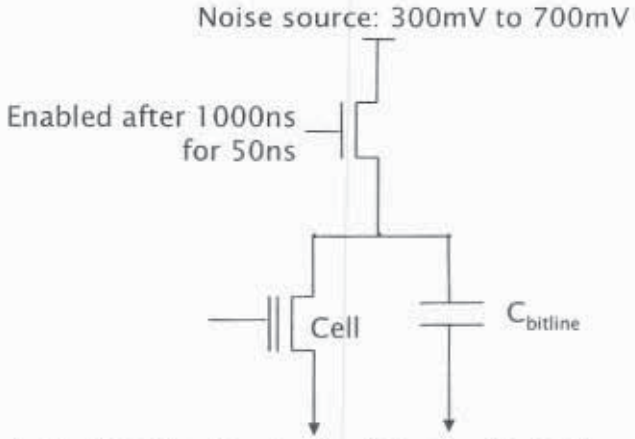


Figure 3.12. Modification to circuit to simulate bitline capacitive coupling.

The simulations in Figures 3.13 and 3.14 show the results of adding a positive and negative noise source to the bitline. For the first case, the positive noise source causes the bitline to be recharged even though the feedback was not enabled. For the second case, the negative noise source causes the bitline to briefly discharge faster than expected for the flash cell current. If the sense amplifier was designed with a differential amplifier or a latch, a significant noise jump in the signal could cause the sense amp to output an incorrect result. However with the $\Delta\Sigma$ sensing methodology, the error will average out over time. Improved accuracy can be obtained with longer sensing times. Figures 3.11 and 3.12 illustrate the circuits behavior with the simulated capacitive coupling, and Table 3.4 contains the results for 50mV and 200mV noise sources with a 5µs sense. The 50mV noise source results in practically no difference to the calculated current.

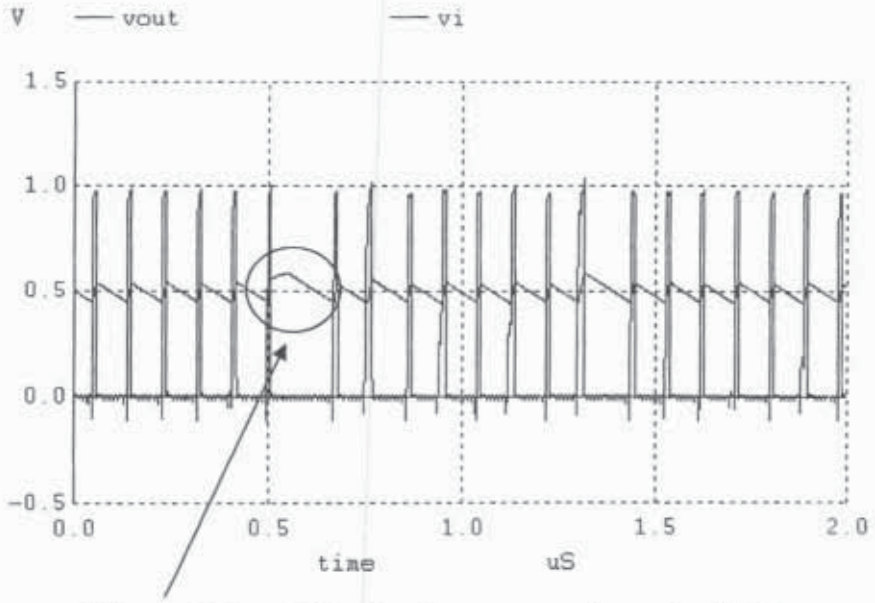


Figure 3.13. +200mV noise source: charge is added to the bitline when feedback is not enabled.

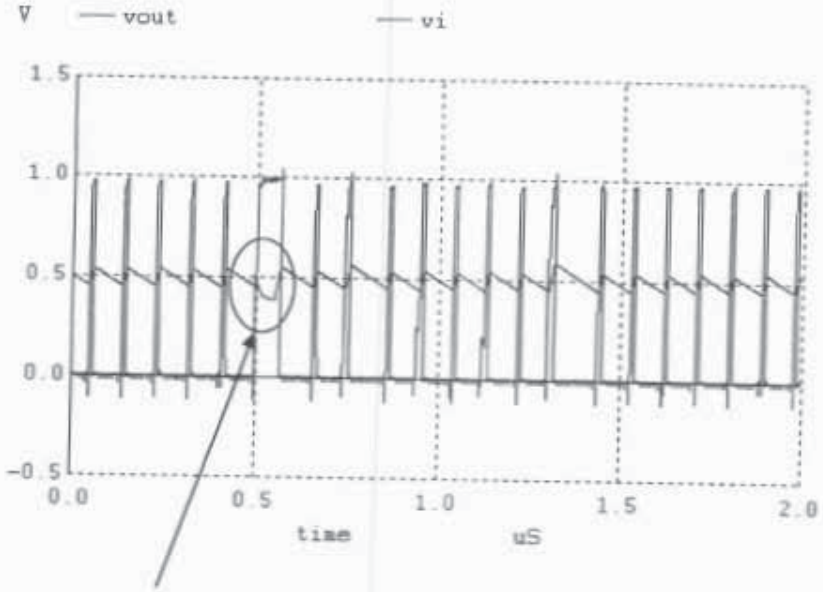


Figure 3.14. -200mV noise source: charge is removed from the bitline faster than expected with 1uA flash cell current.

Table 3.4. SPICE simulation results for bitline capacitive coupling.

Noise Voltage	Cell Current	Simulated # Pulses	Calculated Current
0	1uA	13	1.6uA
+ 50mV	1uA	13	1.6uA
- 50mV	1uA	14	1.7uA
+ 200mV	1uA	10	1.2uA
- 200mV	1uA	18	2.2uA

CHAPTER FOUR

USING THE ΔΣ SENSE AMPLIFIER FOR PROGRAMMING

In chapter three, the design of a sense amplifier using ΔΣ noise shaping was discussed. The ΔΣ sense amplifier with a 100MHz clock and 5μs sensing time was shown to reasonably sense flash cell currents with an accuracy of less than 2-3μA, even under white noise, V_{DD}/ground bounce and bitline capactive coupling conditions. While this degree of accuracy is interesting, it is not beneficial to use with modern flash memory arrays due to the wide threshold voltage separation between programmed and erased bits that is commonly implemented. This chapter discusses implementation of ΔΣ sensing in flash memory programming, in order to take advantage of the high degree of resolution obtainable.

4.1 Conventional Flash Cell Programming

Flash cells may be programmed through either Fowler-Nordheim tunneling or channel hot carrier (CHE) injection. CHE programming is typically used for NOR flash architecture, and is achieved by applying a high voltage to both the control gate (wordline) and the drain (bitline). The source and the substrate are grounded. The high voltage on the control gate and drain causes a high channel current to flow between drain and source, and also causes a channel field that generates hot electrons. The voltage on the control gate is coupled onto the floating gate, attracting hot electrons to the floating

gate. High voltage pulses on the order of a few μs are applied to the cell, and in between each pulse the cell is read against a set current level. The diagram in Figure 4.1 illustrates this programming method.[2,3,15]

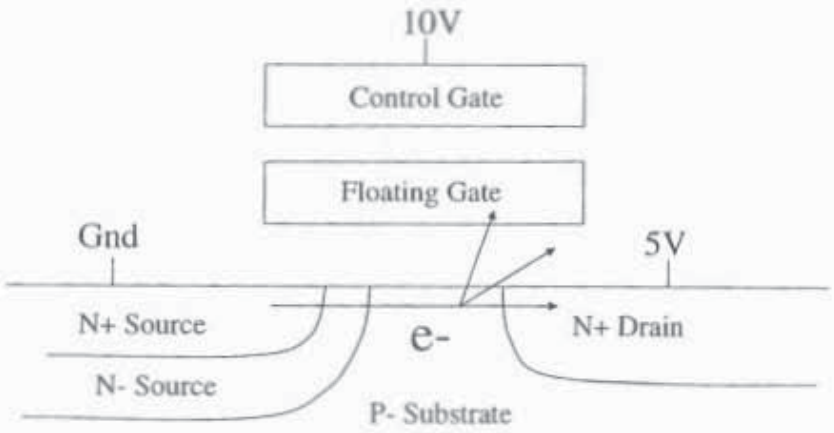


Figure 4.1. CHE programming of a flash cell.

In the case of Fowler-Nordheim tunneling (typically used for NAND flash architecture), a high voltage is applied to the control gate (wordline), which couples a high voltage onto the floating gate. The bitline, source, and substrate are tied to ground. Similarly to CHE programming, the high voltage is applied to the cell in pulses; however the duration is typically several hundred μs, which is at least one order of magnitude larger than for CHE programming.[11,16] Fowler-Nordheim programming is illustrated in Figure 4.2.

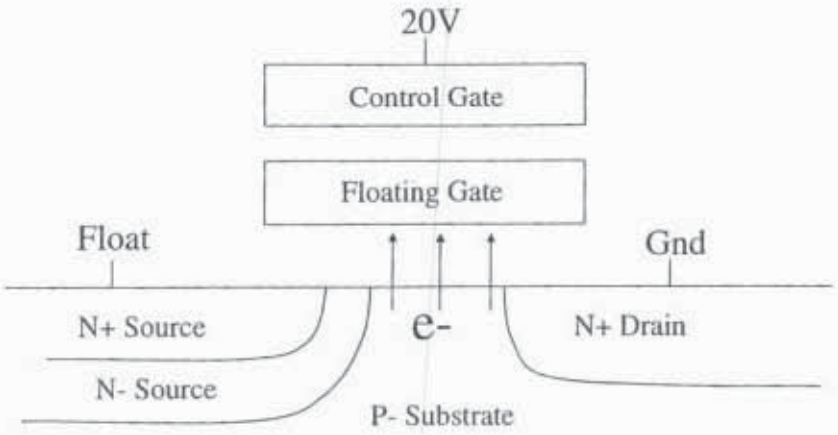


Figure 4.2. Fowler-Nordheim programming of a flash cell.

In both cases, high voltages are applied to the cell for a set period of time (one pulse), and then the cell is compared against a verify level. If the programmed cell's current is greater than the reference current, the cell has not been programmed sufficiently and another pulse will be applied. If the programmed cell's current is less than the reference current, the program operation is ceased and successful. Generally, the same sense amplifier circuitry used for a cell read is used during the verification step, except with a different reference current.

Due to the inherent imprecision of traditional sense amp circuitry, the flash cells are programmed and erased such that there is a wide margin between the programmed and erased current levels. For example, a typical NOR flash memory cell is programmed to a threshold voltage greater than 6V, and erased to a threshold voltage less than 4V.[12] This may result in a programmed/erased current differential of 30μA or higher. This margin is necessary so that the read sense amp makes correct decisions. Further, since

this much margin is used, the accuracy of the programmed and erased currents is not precisely controlled. As long as the programmed level is at least as low as the appropriate reference current, it is considered good enough.

In order to reduce costs, several flash manufacturers are using a multi-level cell technology, which stores more than one bit's worth of data in one cell.[6,7,10,12] For example, a two-bit per cell technology will contain four voltage levels corresponding to 00, 01, 10, and 11 codes, instead of just two voltage levels corresponding to a 0 or 1 code. In order to accomplish this, voltage ranges are established for each digital code, as shown in Figure 4.3.

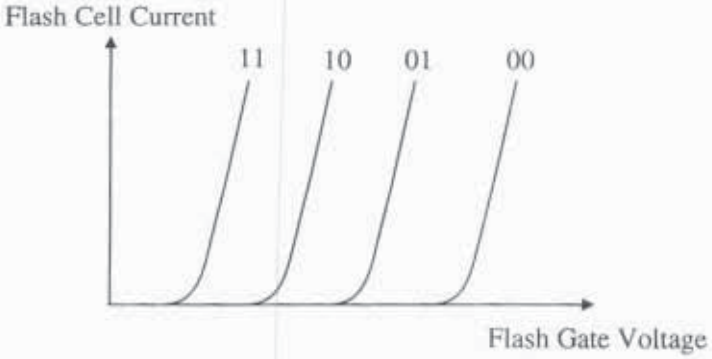


Figure 4.3. Flash IV curves demonstrating 4 states per cell.

In order to use a multi-level cell technology, the flash cell currents must be controlled more precisely. The distances between reference currents for a two-bit per cell technology may approach 10μA, versus the 30μA or more differential that can exist with a one-bit per cell technology. Typically the cell's charge is more accurately controlled during programming by limiting the gate voltage to lower levels while

programming the higher current states, and by using smaller pulse widths and as a result more frequent verifies.

Several issues arise with this programming methodology for multi-level cell technologies. The smaller pulse widths and more frequent verifies leads to a longer programming time, which may or may not be acceptable to customers. The margin between each programmed current level must still be large enough to accommodate the resolution possible with the sense amp, which limits the number of bits that can be implemented per cell. Further, since the cells are programmed until they reach a set current level, and the actual cell current is not accurately monitored, the result may be that the cell's current is on the edge between two logic states, and there is virtually no margin for charge gain or loss during field use.

4.2 Flash Cell Programming with ΔΣ Modulation

The ΔΣ sense amplifier discussed in chapter 3 could be used for verification during the flash programming operations in order to eliminate many of the problems discussed in the previous section. This methodology would be most successful in the case of NAND flash, where the programming times are significantly longer than the sensing times – typically several hundred μs of programming time, versus less than 5μs for a sense within a 1-2μA resolution.[16] An implementation of this idea is shown in Figure 4.4. The controller will consist of some type of counter or filter to convert the ΔΣ modulator output to a cell current value, and logic that switches the output between +20V and 0V depending on whether or not the pre-determined cell current level has been reached.

Assuming the flash cell is erased prior to commencement of the programming operation, the cell will be drawing a high current (for example, 60μA). At the beginning of the programming operation, the controller output will be high (+20V) to enable programming. As the +20V is applied to the flash cell's wordline, the cell's threshold voltage will increase and the current will drop. The current is continuously monitored by the ΔΣ modulator. Once the current reaches the pre-determined level (for example, 30μA), the controller's output will go low (0V) and the programming operation will cease.

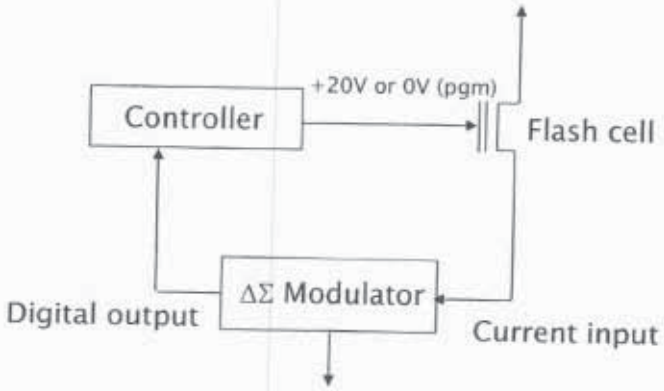


Figure 4.4. Method to program NAND flash cell with ΔΣ modulation.

As shown in chapter 3, the ΔΣ sense amplifier with a 5μs sensing time can measure a current's value to within a few μA. For the current reference of 60μA, sixteen states can be defined with 4μA distance between each state. This would allow a four bit per cell technology – twice as dense as the two bit per cell arrays which are commonly used in the industry. Further, only one current reference is required for programming all

sixteen states; with traditional multi-level cell programming techniques, fifteen current references would be required.

Since the ΔΣ methodology allows a precise measurement of actual flash cell current, the array cells could be programmed to a known value instead of just within a window. The cell current value could be well controlled by adjusting either the voltages and/or pulse durations during each programming operation based on how far along the cell has been programmed. For example, in the case of Fowler-Nordheim tunneling, there could be three different conditions for programming: large, medium, and small, as described in Table 4.1.

Table 4.1. Example of possible programming conditions for NAND flash cell with ΔΣ methodology.

Programming Method	Pulse Width
Large	100us
Medium	50us
Small	10us

The small programming pulses would be defined such that virtually no over-programming occurs. Shifting from large to small programming pulses would not be possible with standard pass/fail verification against a reference current, since the chip's state machine would have no way of knowing at what point the small programming pulses should begin. In order to hit a small window with standard programming, all of the programming pulses would need to be small to prevent over-programming. By using ΔΣ sensing and enabling the possibility to shift from large to small pulses, the

programming operation will be much more time efficient. This methodology is shown in Figure 4.5.

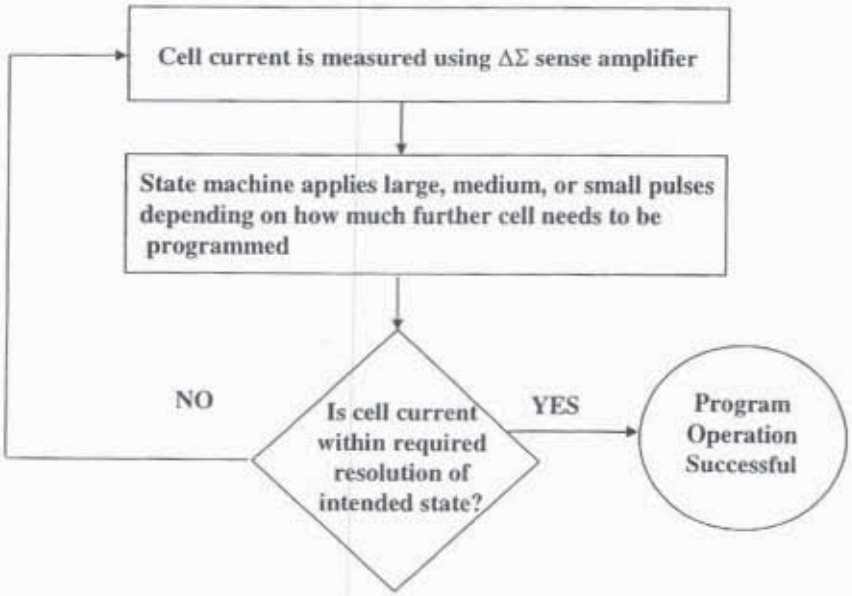


Figure 4.5. Block diagram of ΔΣ programming operation.

The same method can be used to program a NOR flash cell. However, the NOR cell programming with hot carrier injection is accomplished much more quickly – typically on the order of several μs. For best resolution with a NOR architecture, the controller should limit its output high pulses to one 8 to 10V pulse every 5μs, to avoid over-programming.

CHAPTER FIVE

CONCLUSIONS

Sense amplifiers for flash memory arrays are commonly based on differential amplifiers or latches. These circuits have several shortcomings including susceptibility to noise, causing incorrect senses of the memory cell state. To improve noise immunity, a new sense amplifier has been designed using $\Delta\Sigma$ noise modulation.

The new sense amplifier consists simply of three inverters, three switches, and a reference current source. The simple topology increases manufacturability ease, yet due to its noise shaping ability, it can detect a current with approximately 1μA resolution. The new sense amp can withstand V_{DD}/ground bounce of over 50mV and signal noise of over 200mV. The circuit can also be used during programming to achieve higher densities at low cost by storing four bits of data per memory cell.

REFERENCES

[1] Betty Prince, *Semiconductor Memories*, John Wiley & Sons Ltd, 2001. ISBN 0-471-92465-2.

[2] William D. Brown and Joe E. Brewer, *Nonvolatile Semiconductor Memory Technology*, IEEE Press, 1998. ISBN 0-7803-1173-6.

[3] Intel Flash Memory Articles, *What is Flash Memory?*, October 2002. <http://www.intel.com/design/flcomp/articles/298312.pdf>

[4] A.A.M. Amin, "Design and analysis of a high-speed sense amplifier for single-transistor nonvolatile memory cells," IEE Proceedings, April 1993, pp. 117-122.

[5] S. Atsumi et al., "A Channel-Erasing 1.8-V-Only 32-Mb NOR Flash EEPROM with a Bitline Direct Sensing Scheme," IEEE Journal of Solid-State Circuits, Nov. 2000, pp. 1648-1654.

[6] T. Cho et al., "A Dual-Mode NAND Flash Memory: 1-Gb Multilevel and High-Performance 512-Mb Single-Level Modes," IEEE Journal of Solid-State Circuits, Nov. 2001, pp. 1700-1706.

[7] M. Ohkawa et al., "A 98 mm2 Die Size 3.3-V 64-Mb Flash Memory with FN-NOR Type Four-Level Cell," IEEE Journal of Solid-State Circuits, Nov. 1996, pp. 1584-1589.

[8] G. Torelli, A. Manstretta, R. Gastaldi, and P. Rolandi, "Mixed serial-parallel sensing scheme for a 64-Mbit 4-bit/cell factory-programmed OTP-ROM," Microelectronics Journal 30, 1999, pp. 875-886.

[9] J. Bell, J. Bruce, B. Blalock, P. Stubberud, "CMOS current mode flash analog to digital converter", Proc. IEEE Midwest Symposium on Circuits and Systems 2001, vol. 1., August 2001, pp. 272-275.

[10] A. Modelli, A. Manstretta, and G. Torelli, "Basic Feasibility Constraints for Multilevel CHE-Programmed Flash Memories," IEEE Transactions on Electronic Devices, Vol. 48, No. 9, Sept 2001, pp. 2032-2042.

[11] *R. Jacob Baker, Harry W. Li, and David E. Boyce, CMOS: Circuit Design, Layout, and Simulation*, Wiley-IEEE, 1998. ISBN 0-7803-3416-7.

[12] Intel Technology Journal, *Intel StrataFlash Memory Technology Overview*, Q4'97. <http://www.intel.com/design/flash/isf/overview.pdf>

[13] R. Jacob Baker, *CMOS Mixed-Signal Circuit Design*, Wiley-IEEE, 2002. ISBN 0-471-22754-4.

[14] Steven W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, California Technical Publishing, 1997. ISBN 0-9660176-3-3.

[15] Intel StrataFlash Memory Datasheets, July 2004, <http://www.intel.com/design/flcomp/datashts/290667.htm>

[16] Samsung NAND Flash Memory Datasheets, May 2004. <http://www.samsung.com/Products/Semiconductor/Flash/NAND/2Gbit/K9F2G08U0M/ds_k9f2g08u0m_rev06.pdf>

APPENDIX

SPICE Netlist for Proposed Delta Sigma Modulator

```
.option scale=50n
.tran 1n 2000n 0 1n UIC

* Input Signals (Power, Clock)
VDD    VDD 0 DC 1
Vclk   clk 0    DC 0  pulse 0 1 0 200p 200p 5n 10n
VSS    VSS 0 DC 0

* Flash Cell being sensed (modeled as Ideal current source)
Ibit   Vi VSS 6u

* Integrator
Cb     Vi VSS 5p  IC=0.5

* Clocked Comparator
ML1 Vinv1 Vi VDD VDD PMOS L=10 W=20
ML2 Vinv1 Vi VSS VSS NMOS L=10 W=10
MS1 Vinv1 clk Vx1 VSS NMOS L=1 W=10
ML5 Vinv2 Vx1 VDD VDD PMOS L=10 W=20
ML6 Vinv2 Vx1 VSS VSS NMOS L=10 W=10
MS2 Vinv2 clk Vx2 VDD PMOS L=1 W=20
ML9 Vout Vx2 VDD VDD PMOS L=10 W=20
ML10 Vout Vx2 VSS VSS NMOS L=10 W=10

* Feedback
Icup   VDD    Vcup   60u
MF1 Vi Vout Vcup VSS NMOS L=1 W=20


* 50nm BSIM4 models
*
* Don't forget the .options scale=50nm if using an Lmin of 1
* 1<Ldrawn<200   10<Wdrawn<10000 Vdd=1V
* Change to level=54 when using HSPICE

.model  nmos  nmos  level = 14

+binunit = 1          paramchk= 1          mobmod  = 0
+capmod  = 2          igcmod  = 1          igbmod  = 1          geomod  = 1
+diomod  = 1          rdsmod  = 0          rbodymod= 1          rgatemod= 1
+permod  = 1          acnqsmod= 0          trnqsmod= 0

+tnom    = 27         toxe    = 1.4e-009   toxp    = 7e-010     toxm    = 1.4e-009
+epsrox  = 3.9        wint    = 5e-009     lint    = 1.2e-008
+ll      = 0          wl      = 0          lln     = 1          wln     = 1
+lw      = 0          ww      = 0          lwn     = 1          wwn     = 1
+lwl     = 0          wwl     = 0          xpart   = 0          toxref  = 1.4e-009

+vth0    = 0.22       k1      = 0.35       k2      = 0.05       k3      = 0
+k3b     = 0          w0      = 2.5e-006   dvt0    = 2.8        dvt1    = 0.52
+dvt2    = -0.032     dvt0w   = 0          dvt1w   = 0          dvt2w   = 0
+dsub    = 2          minv    = 0.05       voff1   = 0          dvtp0   = 1e-007
+dvtp1   = 0.05       lpe0    = 5.75e-008  lpeb    = 2.3e-010   xj      = 2e-008
+ngate   = 5e+020     ndep    = 2.8e+018   nsd     = 1e+020     phin    = 0
+cdsc    = 0.0002     cdscb   = 0          cdscd   = 0          cit     = 0
+voff    = -0.15      nfactor = 1.2        eta0    = 0.15       etab    = 0
+vfb     = -0.55      u0      = 0.032      ua      = 1.6e-010   ub      = 1.1e-017
+uc      = -3e-011    vsat    = 1.1e+005   a0      = 2          ags     = 1e-020
+a1      = 0          a2      = 1          b0      = -1e-020    b1      = 0
+keta    = 0.04       dwg     = 0          dwb     = 0          pclm    = 0.18
+pdiblc1 = 0.028      pdiblc2 = 0.022      pdiblcb = -0.005     drout   = 0.45
+pvag    = 1e-020     delta   = 0.01       pscbe1  = 8.14e+008  pscbe2  = 1e-007
+fprout  = 0.2        pdits   = 0.2        pditsd  = 0.23       pditsl  = 2.3e+006
+rsh     = 3          rdsw    = 150        rsw     = 150        rdw     = 150
+rdswmin = 0          rdwmin  = 0          rswmin  = 0          prwg    = 0
```

```
+prwb    = 6.8e-011     wr       = 1          alpha0  = 0.074      alpha1  = 0.005
+beta0   = 30           agidl    = 0.0002     bgidl   = 2.1e+009   cgidl   = 0.0002
+egidl   = 0.8

+aigbacc = 0.012        bigbacc = 0.0028      cigbacc = 0.002
+nigbacc = 1            aigbinv = 0.014       bigbinv = 0.004      cigbinv = 0.004
+eigbinv = 1.1          nigbinv = 3           aigc    = 0.017      bigc    = 0.0028
+cigc    = 0.002        aigsd   = 0.017       bigsd   = 0.0028     cigsd   = 0.002
+nigc    = 1            poxedge = 1           pigcd   = 1          ntox    = 1

+xrcrg1  = 12           xrcrg2  = 5
+cgso    = 6.238e-010   cgdo    = 6.238e-010  cgbo    = 2.56e-011  cgdl    = 2.495e-10
+cgsl    = 2.495e-10    ckappas = 0.02        ckappad = 0.02       acde    = 1
+moin    = 15           noff    = 0.9         voffcv  = 0.02

+kt1     = -0.21        kt11    = 0.0         kt2     = -0.042     ute     = -1.5
+ua1     = 1e-009       ub1     = -3.5e-019   uc1     = 0          prt     = 0
+at      = 53000

+fnoimod = 1            tnoimod = 0

+jss     = 0.0001       jsws    = 1e-011      jswgs   = 1e-010     njs     = 1
+ijthsfwd= 0.01         ijthsrev= 0.001       bvs     = 10         xjbvs   = 1
+jsd     = 0.0001       jswd    = 1e-011      jswgd   = 1e-010     njd     = 1
+ijthdfwd= 0.01         ijthdrev= 0.001       bvd     = 10         xjbvd   = 1
+pbs     = 1            cjs     = 0.0005      mjs     = 0.5        pbsws   = 1
+cjsws   = 5e-010       mjsws   = 0.33        pbswgs  = 1          cjswgs  = 3e-010
+mjswgs  = 0.33         pbd     = 1           cjd     = 0.0005     mjd     = 0.5
+pbswd   = 1            cjswd   = 5e-010      mjswd   = 0.33       pbswgd  = 1
+cjswgd  = 5e-010       mjswgd  = 0.33        tpb     = 0.005      tcj     = 0.001
+tpbsw   = 0.005        tcjsw   = 0.001       tpbswg  = 0.005      tcjswg  = 0.001
+xtis    = 3            xtid    = 3

+dmcg    = 0e-006       dmci    = 0e-006      dmdg    = 0e-006     dmcgt   = 0e-007
+dwj     = 0.0e-008     xgw     = 0e-007      xgl     = 0e-008

+rshg    = 0.4          gbmin   = 1e-010      rbpb    = 5          rbpd    = 15
+rbps    = 15           rbdb    = 15          rbsb    = 15         ngcon   = 1

.model  pmos  pmos  level = 14

+binunit = 1            paramchk= 1           mobmod  = 0
+capmod  = 2            igcmod  = 1           igbmod  = 1          geomod  = 1
+diomod  = 1            rdsmod  = 0           rbodymod= 1          rgatemod= 1
+permod  = 1            acnqsmod= 0           trnqsmod= 0

+tnom    = 27           toxe    = 1.4e-009    toxp    = 7e-010     toxm    = 1.4e-009
+epsrox  = 3.9          wint    = 5e-009      lint    = 1.2e-008
+ll      = 0            wl      = 0           lln     = 1          wln     = 1
+lw      = 0            ww      = 0           lwn     = 1          wwn     = 1
+lwl     = 0            wwl     = 0           xpart   = 0          toxref  = 1.4e-009

+vth0    = -0.22        k1      = 0.39        k2      = 0.05       k3      = 0
+k3b     = 0            w0      = 2.5e-006    dvt0    = 3.9        dvt1    = 0.635
+dvt2    = -0.032       dvt0w   = 0           dvt1w   = 0          dvt2w   = 0
+dsub    = 0.7          minv    = 0.05        voff1   = 0          dvtp0   = 0.5e-008
+dvtp1   = 0.05         lpe0    = 5.75e-008   lpeb    = 2.3e-010   xj      = 2e-008
+ngate   = 5e+020       ndep    = 2.8e+018    nsd     = 1e+020     phin    = 0
+cdsc    = 0.000258     cdscb   = 0           cdscd   = 6.1e-008   cit     = 0
+voff    = -0.15        nfactor = 2           eta0    = 0.15       etab    = 0
+vfb     = 0.55         u0      = 0.0095      ua      = 1.6e-009   ub      = 8e-018
+uc      = 4.6e-013     vsat    = 90000       a0      = 1.2        ags     = 1e-020
+a1      = 0            a2      = 1           b0      = -1e-020    b1      = 0
+keta    = -0.047       dwg     = 0           dwb     = 0          pclm    = 0.55
+pdiblc1 = 0.03         pdiblc2 = 0.0055      pdiblcb = 3.4e-008   drout   = 0.56
+pvag    = 1e-020       delta   = 0.014       pscbe1  = 8.14e+008  pscbe2  = 9.58e-007
+fprout  = 0.2          pdits   = 0.2         pditsd  = 0.23       pditsl  = 2.3e+006
+rsh     = 3            rdsw    = 250         rsw     = 160        rdw     = 160
+rdswmin = 0            rdwmin  = 0           rswmin  = 0          prwg    = 3.22e-008
```

```
+prwb    = 6.8e-011     wr       = 1          alpha0   = 0.074      alpha1   = 0.005
+beta0   = 30           agidl    = 0.0002     bgidl    = 2.1e+009  cgidl    = 0.0002
+egidl   = 0.8

+aigbacc = 0.012        bigbacc  = 0.0028     cigbacc  = 0.002
+nigbacc = 1            aigbinv  = 0.014      bigbinv  = 0.004     cigbinv  = 0.004
+eigbinv = 1.1          nigbinv  = 3          aigc     = 0.69      bigc     = 0.0012
+cigc    = 0.0008       aigsd    = 0.0087     bigsd    = 0.0012    cigsd    = 0.0008
+nigc    = 1            poxedge  = 1          pigcd    = 1         ntox     = 1

+xrcrg1  = 12           xrcrg2   = 5
+cgso    = 7.43e-010    cgdo     = 7.43e-010  cgbo     = 2.56e-011 cgdl     = 1e-014
+cgsl    = 1e-014       ckappas  = 0.5        ckappad  = 0.5       acde     = 1
+moin    = 15           noff     = 0.9        voffcv   = 0.02

+kt1     = -0.19        kt11     = 0          kt2      = -0.052    ute      = -1.5
+ua1     = -1e-009      ub1      = 2e-018     uc1      = 0         prt      = 0
+at      = 33000

+fnoimod = 1            tnoimod  = 0

+jss     = 0.0001       jsws     = 1e-011     jswgs    = 1e-010    njs      = 1
+ijthsfwd= 0.01         ijthsrev= 0.001       bvs      = 10        xjbvs    = 1
+jsd     = 0.0001       jswd     = 1e-011     jswgd    = 1e-010    njd      = 1
+ijthdfwd= 0.01         ijthdrev= 0.001       bvd      = 10        xjbvd    = 1
+pbs     = 1            cjs      = 0.0005     mjs      = 0.5       pbsws    = 1
+cjsws   = 5e-010       mjsws    = 0.33       pbswgs   = 1         cjswgs   = 3e-010
+mjswgs  = 0.33         pbd      = 1          cjd      = 0.0005    mjd      = 0.5
+pbswd   = 1            cjswd    = 5e-010     mjswd    = 0.33      pbswgd   = 1
+cjswgd  = 5e-010       mjswgd   = 0.33       tpb      = 0.005     tcj      = 0.001
+tpbsw   = 0.005        tcjsw    = 0.001      tpbswg   = 0.005     tcjswg   = 0.001
+xtis    = 3            xtid     = 3

+dmcg    = 5e-006       dmci     = 5e-006     dmdg     = 5e-006    dmcgt    = 6e-007
+dwj     = 4.5e-008     xgw      = 3e-007     xgl      = 4e-008

+rshg    = 0.4          gbmin    = 1e-010     rbpb     = 5         rbpd     = 15
+rbps    = 15           rbdb     = 15         rbsb     = 15        ngcon    = 1

.end
```