8-bit Charge-Redistribution ADC

Jonathan Choy
SID 13399964
jchoy@cory.eecs.berkeley.edu

08 May, 2001
Contents

Introduction ......................................................... 3
  1.1 Overview .................................................. 3
  1.2 High-Level Description & Block Diagram ..................... 3
  1.3 Timing .................................................... 4
    1.3.1 State Transitions .................................... 4
    1.3.2 Timing Diagrams ..................................... 5
Implementation Details ........................................... 7
  2.1 Design Goals ............................................. 7
    2.1.1 Power Issues ......................................... 7
  2.2 Amplifier Design Issues ................................... 7
    2.2.1 Choosing an appropriate $V_{\text{ref}}$ .................. 8
    2.2.2 Using the op-amp as a comparator ....................... 8
    2.2.3 Using the op-amp for the resettable-gain circuit ......... 8
    2.2.4 Amplifier Design Flow ................................ 8
    2.2.5 Amplifier Performance Metrics ......................... 9
  2.3 Switch Design Issues ..................................... 11
    2.3.1 Settling Time ......................................... 11
    2.3.2 Charge Injection ..................................... 12
ADC Performance .................................................. 13
  3.1 Waveforms ............................................... 13
  3.2 Disclaimer ............................................... 13
Conclusion ......................................................... 17
  4.1 Improving on this Design in the Future ....................... 17
  4.2 Tradeoffs and Other Design Issues ........................ 17
Appendix A - Hand Calculations

5.1 Parasitic Capacitance seen at the input of an op-amp .......................... 18
5.2 Switching and Charge Injection ......................................................... 18
  5.2.1 Settling Time .............................................................................. 18
  5.2.2 Charge Injection ........................................................................ 20
5.3 Op-Amp and Bias Circuitry Device Sizes ............................................. 22
  5.3.1 The Folded Cascode Op-Amp ...................................................... 23
  5.3.2 Supply-Independent Bias Circuitry ............................................. 24
5.4 Op-Amp Performance Metrics ............................................................... 24
  5.4.1 Low-Frequency Gain ................................................................. 25
  5.4.2 Dominant Pole Frequency ......................................................... 25
  5.4.3 Unity Gain Frequency .................................................................. 25
5.5 Settling Time Calculations ................................................................. 26
  5.5.1 Bit-cycle period ......................................................................... 26
  5.5.2 Sampling Period ......................................................................... 26
5.6 Input-Capacitor and Pre-Amplifier Stages .......................................... 27
  5.6.1 Input Range and Pre-Amplifier Gain ........................................... 27

Appendix B - SPICE Stuff .................................................................... 27

6.1 SPICE Output: proof that I really did run this in SPICE .................... 28
6.2 Miscellaneous MetaWave Graphs ......................................................... 29

Appendix C - Miscellaneous Schematics ............................................... 29

7.1 Input Stage ...................................................................................... 31
7.2 Resettable Gain Pre-Amplifier Stage ............................................... 32
7.3 Capacitor Array .............................................................................. 33
Introduction

1.1 Overview

This project involved the design of an 8-bit charge-redistribution analog-to-digital converter (ADC).

Obviously, the main goal is to meet the given specifications, which are as follows:

- Supply voltages of ±2.5V, with a tolerance of +10%
- \( \leq \) 1mW static power consumption
- No ideal circuit components
- Minimum gate length of 1\( \mu \)m

While some of the processes involved in the project’s development were not too pretty, here is the status of the project as submitted:

- Functional ADC operation at 100kSample/s (or 10\( \mu \)s/Sample) for ±2.5V. Digital output at ±2.75V is inconsistent with the output at ±2.5V, but its “functionality” has not been investigated in depth.
- \( \approx \) 750\( \mu \)W static power consumption at ±2.5V supply; \( \approx \) 1.01mW static power consumption at ±2.75V supply.
- All bias voltages are created with transistors and one accurate resistor; ideal voltage references are used only for powering the supply rails and for generating external signals (i.e., clocks) used in simulation.
- Except for digital switches, all lengths are \( \geq \) 1\( \mu \)m.

1.2 High-Level Description & Block Diagram

As illustrated in Figure 1.1, the circuitry used for this project can be divided into six major modules; however, only the following five are implemented:

1. **Input Stage**: This stage contains the variable sense capacitor \( (C_S) \) which has an optimal capacitance of 1pF (±5%), and a reference capacitor \( (C_R) \). It produces an analog voltage that is a mathematical ratio between \( C_S \) and \( C_R \) as given by the capacitive voltage divider equation.

2. **Pre-Amplifier**: Since ±5% variations of a 1pF capacitor do not produce enough variation of voltage as needed for eight bits of resolution, the analog signal produced by the input stage is subsequently amplified such that the ±5% will cover the range of voltages between 0V and \( V_{ref} \), the reference voltage used in the capacitor array.
3. **Capacitor Array**: After the input voltage is “scaled” by the pre-amplifier stage, it is then passed to the capacitor array which will pinpoint its location within the range of 0V to \( V_{ref} \). This is done using charge-redistribution to determine iteratively what each bit of a voltage’s digital representation (relative to \( V_{ref} \)) should be. In order to guess this value accurately, it needs sequential feedback as to whether its “current” guess is too high or too low.

4. **Comparator**: The comparator’s only purpose is to digitize the accuracy of the capacitor array’s current “guess.” Ideally, this digital value will be fed to some logic that will use this information to set one bit of the digital output and provide feedback to the capacitor array. For the purposes of this project, it is implemented using an op-amp with a high open-loop gain.

5. **Supply-Independent Bias**: This is a biasing circuit that produces appropriate reference voltages to ensure the functionality of the internal circuitry of the pre-amplifier, capacitor array, and comparator blocks. For both the pre-amplifier and comparator, it provides bias voltages for op-amps; and for the capacitor array, it provides the value \( V_{ref} \) to be used in charge redistribution.

The sixth major module is the Successive Approximation Register (SAR), which is not covered in this report for two reasons: (1) it was given to us and pre-designed by someone else, and (2) it was not used in simulating the ADC’s functionality. Ideally, this is the “logic” that interprets the output from the comparator and does two things: (1) sets the output bits of the digital representation appropriately, and (2) provides feedback to the capacitor array so that its next guess will get closer to the actual voltage value.

### 1.3 Timing

Functionally, this particular implementation of an ADC can be thought of as a finite state machine that has two states: “sample” and “process.” As can be easily guessed, during the “sample” state, an analog value is sampled and stored into some internal structure. Following this, the system enters the “process” state in which the stored value is digitized. This process is repeated for each sample that is taken.

#### 1.3.1 State Transitions

To control all of this stuff, three clock-control signals and eight bit-control signals are to be provided externally to drive the circuitry. They are as follows:
1. **AMPReset**: This signal drives the input stage and the pre-amplifier. When high, both these stages are “pre-charging” into an initial state and no voltages are ready to be evaluated.

2. **AMPSample**: This signal drives the input stage and the pre-amplifier. When high, both of these modules are in a “sample” state where the circuitry is switched “on.” It is in this state that the voltage is generated at the input, passed through the amplifier, and then sampled and stored by the capacitor array. Further, **AMPSample** and **AMPReset** are non-overlapping.

3. **ADC Sample**: This signal controls the capacitor array: when high, the capacitor array is in a “listen” (sample) mode in which it is connected to the incoming circuitry. This way, it will be able to sample a voltage according to what’s coming in. When low, the capacitor array is magically “disconnected” from the previous modules, and the system enters the “process” state where the capacitor array does its iterative guessing (with feedback from the comparator and SAR) to arrive at the final digital value.

4. **Bitline control signals**: Ideally, these should be signals fed back into the capacitor array from the SAR. However, since I did not use a SAR, these signals were generated manually. Basically, they alternate as digital “1” and “0” depending on the output of the comparator. They are fed back into the capacitor array and used to “shape” the redistribution of charge and lead the capacitor array to eventually guess the correct voltage value (or something close to it).

### 1.3.2 Timing Diagrams

![Timing Diagrams](image-url)
These are the control signals used to drive the circuitry. As calculated in Appendix A, the signal of importance during the Sample state needs a settling time of 3.75µs. Just to be paranoid, a full 4µs is allotted for sampling at the ADC.

Also, the settling time needed for the op-amp-acting-as-a-comparator is calculated to be 380ns. However, since the actual op-amp out-performs the conceptual one used in the calculation of this number (in terms of open-loop gain), only 300ns is allotted for each bit to settle to its final value, and as seen in simulation outputs, this is more than enough settling time.

Put together, these two settling times account for the timing of the clocks as presented above.

To avoid any potential ill-effects due to charge injection, most rise and fall times of signals that affect “sensitive” nodes (i.e. capacitively isolated nodes) are set to an extremely long 1µs. For the bit-switching, a rise and fall time of 100ns is used since charge injection is not as big a deal on the capacitor array.

The only signal that looks funky is the AMPReset signal - it has a rise time of 100ns and a fall time of 1us; since its rising edge signals the beginning of the reset signal, any switch controlled by this signal will be shorted to a power rail, and thus any injected charge on a node will have plenty of time to dissipate. For the fall time of this signal, however, some switches get turned off with the falling edge, and it is desirable to minimize the trapped charge by using a slow falling edge.
Implementation Details

This section outlines the design flow by which the final values of the design were calculated. All the gory details and calculations are explained in Appendix A.

2.1 Design Goals

To meet the “minimum” specifications for full credit, I chose to implement an eight-bit design that could handle 100kSample/second. As stated in the project description, this accounts for up to 80% of a final grade. I hope that this report and analysis takes care of the remaining 20%.

2.1.1 Power Issues

“Power kills. Absolute power kills absolutely.”

— Rudolph Rummel

Of all the constraints given in the project, the limiting one (to me, at least) was that of power. For this design, static power consumption is limited to 1mW. With a ±2.75V worst-case supply, this translates to a maximum total of 180µA of current available for all amplifiers and biasing circuits.

Considering the additional power consumption that occurs as a result of the occasional inverter used in the circuitry, I set out to design an op-amp and bias circuitry that together would consume less than one-third of this (or, 60µA). Since I needed at least two amplifiers (one for the comparator and one for the pre-amplifier), this should give me some overhead to play with in optimizing the design at the end.

2.2 Amplifier Design Issues

To achieve good swing, a folded-cascode topology was chosen for the opamp implementation. What this means is that the power consumption would take a hit: namely, the total current running through the amplifier (without considering biasing circuitry) is four times the current running through one of the input devices. Given this and the previously stated 60µA target for op-amp-plus-bias current, a nominal 10µA current through the input devices is chosen.

Along with this current, a $V_{DSAT}$ of 0.15V is chosen for the input device to achieve a relatively high output swing. With this information, the first thing to do is figure out whether this $I_D$ and $V_{DSAT}$ is enough – in other words, is it possible to use this op-amp for a comparator and/or a resettable-gain circuit while still allowing a sample frequency of 100kHz?

The following sections explain the thought processes behind the particular implementation that I came up with.
2.2.1 Choosing an appropriate $V_{ref}$

The reference voltage needed for comparison in the capacitor array is arbitrarily chosen to be $V_{DD} - (V_{DSAT} + V_T)$ (or 1.35V), since can be easily generated by the bias circuitry. When considering a choice of a reference voltage, one of the first issues to deal with is that of voltage swing: since the input voltage to the capacitor array to be compared to this reference voltage comes out of an amplifier, it is limited by the amplifier’s output swing. I figured that choosing a lower $V_{ref}$ would avoid this problem.

2.2.2 Using the op-amp as a comparator

The first step in using this op-amp as a comparator is to figure out how much open-loop gain it needs. With a $V_{ref}$ of 1.35V, the LSB at the comparator input is 5.27mV. Given this value, the op-amp must have an open-loop gain of at least 948.

To be conservative, the necessary gain is estimated to be 1000. With this gain, and the previously chosen $I_D$ and $V_{DSAT}$ values, the minimum bit-cycle time (time for the output of the op-amp as a comparator to reach its final value) is found to be 0.38µs. So to sample eight bits, the system would need about 4µs (accounting for rise and fall times).

2.2.3 Using the op-amp for the resettable-gain circuit

First, the desired closed-loop gain is calculated to be around -11. Then, just to be safe, this number is increased to -13 (to get a longer, safer settling time) and the settling time of the resettable-gain pre-amplifier stage is then calculated to be 3.75µs.

Putting these two together, it appears that the total time needed to process a single analog input and turn it into an 8-bit digital representation is at least 8.5-9µsec (including all rise and fall times). This corresponds to a sampling rate of 111kSample/s, which exceeds the desired 100kSample/s condition stated above. This is a good thing. But just to be safe, the initial design is run at 100kSample/s.

2.2.4 Amplifier Design Flow

Since an input-device current of 10µA is sufficient to meet the specs, it is now safe to proceed with designing the rest of this amplifier.

Bias Circuitry & Multiple-$V_{DSAT}$ Implementation

The schematic of the final design (with bias circuitry) used is shown in Figure 2.1. The left half of the figure depicts a “wide-swing” supply-independent bias circuit which provides three bias voltages. In designing the bias circuitry, I was faced with this problem: the supply-independent bias as I knew it (from the homework) provided at most two bias voltages – one for PMOS devices, and another for NMOS devices. And at a first glance, a folded-cascode topology requires four bias voltages: one for each of the PMOS cascode pairs, one for the wide-swing NMOS cascode, and one for the tail device pulling current through the differential pair. So creating a supply-independent bias for the folded-cascode device is not trivial.

The simple and straight-forward way to bias this amplifier is to pull out four branches (in addition to the two already there for the classical supply-independent bias) of current – one for each of the needed biases – with one “mirror” device on each one. However, each branch that gets pulled out will consume static power, which is a bad thing since this design is shooting for low power consumption.

Instead, I pulled out only one branch of current to generate a third bias voltage, while feeding through the two generated by the initial circuitry. Since the PMOS cascode devices of the NMOS-input folded-cascode
Figure 2.1: Transistor-level schematic of the designed folded-cascode op-amp with bias circuitry.

should be wide-swing, I implemented a similar wide-swing topology on the one branch that was pulled out. For this to work well, $V_{DSAT}$ of all these wide-swing PMOS devices should be the same.

To take care of biasing the NMOS devices, I had two options: I could pull out a second branch and do the same wide-swing biasing trick that was done with the PMOS devices, or I could just play some $V_{DSAT}$ tricks so that the bias voltage needed for the NMOS cascode and the NMOS tail device were the same (or at least close enough to the same). Having chosen $V_{DSAT}$ of the input devices of the amplifier to be 0.15 V, I went ahead and arbitrarily set the $V_{DSAT}$ of all the cascode devices in the amplifier to that value too. What this also means is that the $V_{DSAT}$ of the NMOS tail device needed to be at least twice that if it is to share a bias voltage. So leaving some room to play with later on, I arbitrarily assigned a $V_{DSAT}$ of 0.35 V to the tail device, which also became the $V_{DSAT}$ used in the supply-independent bias circuitry.

Device Sizes

The right side of Figure 2.1 shows the classical folded-cascode topology and all the device sizes. When doing hand calculations, I actually took channel-length modulation into account when choosing the device sizes. This would explain the somewhat unconventional widths of various devices in the amplifier.

As mentioned previously, the nominal current through the input devices (5A and 5B) is 10 μA. Consequently, the bias circuitry was designed to pull a nominal current of 5 μA. The next section gives more details about the computations and design decisions.

2.2.5 Amplifier Performance Metrics

See Appendix B for the SPICE output files. Only the results are mentioned here.

After designing the op-amp (as mentioned above), I tested it extensively by itself (one amplifier circuit hooked up to one biasing circuit) to make sure that it satisfied the previously assumed constraints: that its
open-loop gain is at least 1000, and that the input device has a $V_{dsat}$ of 0.15 with an $I_D$ of 10μA.

Table 2.1: Hand calculations vs. SPICE output for operating point parameters of bias circuitry MOSFETs.

<table>
<thead>
<tr>
<th>Device</th>
<th>$I_D$ (μA)</th>
<th>$V_{DSAT}$ (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1a</td>
<td>Hand</td>
<td>Hand</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5.027</td>
</tr>
<tr>
<td>M1b</td>
<td>5</td>
<td>5.079</td>
</tr>
<tr>
<td>M2a</td>
<td>5</td>
<td>5.027</td>
</tr>
<tr>
<td>M2b</td>
<td>5</td>
<td>5.079</td>
</tr>
<tr>
<td>M3a</td>
<td>5</td>
<td>5.027</td>
</tr>
<tr>
<td>M3b</td>
<td>5</td>
<td>5.079</td>
</tr>
<tr>
<td>M2c</td>
<td>5</td>
<td>5.017</td>
</tr>
<tr>
<td>M3c</td>
<td>5</td>
<td>5.017</td>
</tr>
<tr>
<td>M9c</td>
<td>5</td>
<td>5.017</td>
</tr>
</tbody>
</table>

Table 2.2: Hand calculations vs. SPICE output for operating point and performance parameters of op-amp MOSFETs.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>M4</th>
<th>M5</th>
<th>M9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_D$ (μA)</td>
<td>(Hand)</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>(SPICE - ±2.5V)</td>
<td>9.08</td>
<td>9.08</td>
<td>9.08</td>
<td>20.05</td>
<td>10.98</td>
</tr>
<tr>
<td></td>
<td>(SPICE - ±2.75V)</td>
<td>9.27</td>
<td>9.27</td>
<td>9.27</td>
<td>20.65</td>
<td>11.38</td>
</tr>
<tr>
<td>$V_{DSAT}$ (V)</td>
<td>(Hand)</td>
<td>0.15</td>
<td>0.15</td>
<td>0.15</td>
<td>0.15</td>
<td>0.15</td>
</tr>
<tr>
<td></td>
<td>(SPICE - ±2.5V)</td>
<td>0.143</td>
<td>0.143</td>
<td>0.101</td>
<td>0.106</td>
<td>0.163</td>
</tr>
<tr>
<td></td>
<td>(SPICE - ±2.75V)</td>
<td>0.144</td>
<td>0.145</td>
<td>0.100</td>
<td>0.108</td>
<td>0.164</td>
</tr>
</tbody>
</table>

Table 2.3: Hand calculations vs. SPICE output for operating point performance parameters of op-amp (with 50kΩ load).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Hand Calculation</th>
<th>SPICE Output at ±2.5V</th>
<th>SPICE Output at ±2.75V</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_{v0}$ (V/V)</td>
<td>9230</td>
<td>8382</td>
<td>8608</td>
</tr>
<tr>
<td>$\omega_{3dB}$ (kHz)</td>
<td>45.9</td>
<td>50.5</td>
<td>50.1</td>
</tr>
<tr>
<td>$\omega_{0}$ (MHz)</td>
<td>423.3</td>
<td>378.5</td>
<td>386.7</td>
</tr>
<tr>
<td>$R_o$ (MΩ)</td>
<td>69.4</td>
<td>63.2</td>
<td>63.1</td>
</tr>
<tr>
<td>phase margin</td>
<td>55</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td>power (μW)</td>
<td>275</td>
<td>256</td>
<td>289</td>
</tr>
</tbody>
</table>

As shown in Table 2.1, the current through the biasing circuitry devices is close enough to the nominal value of 5μA. The $V_{DSAT}$ values are all close to the “ideal” hand-calculated values too, except for the PMOS devices. But since this does not seem to hinder the performance of the op-amp in any way and is also above the “minimum” $V_{DSAT}$ of 0.1V, it is acceptable.

Correspondingly, the SPICE results for the operating point of the op-amp seem to correspond rather well with the hand calculations (see Tables 2.2 and 2.3). The only discrepancies between SPICE and the hand calculations can be attributed to the fact that not all the resistances are taken into account when calculating the output resistance by hand. As such, the hand-calculated output resistance should be a little higher than the actual resistance, which results in an overestimated gain and underestimated pole frequency.
2.3 Switch Design Issues

All switches used in this circuit are minimum-sized devices (namely, W/L = 1μm/0.5μm). This is chosen to reduce the amount of error produced by charge injection when the MOS switches turn off. All the analysis done here is for NMOS switches; in the actual implemented circuit, full complementary switches are occasionally used when the signal on one end of the switch might rise to within V_T of the top rail. Otherwise, the performance of the complementary switches is at worst, the same as the performance of an NMOS-only switch.

2.3.1 Settling Time

With this in mind, and assuming a paranoid settling time of 6τ_{RC} (modeling the switch onto a node as an RC delay), the time required for a value to settle after a switch turns on can be calculated to be 6τ_{RC}, where τ_{RC} = R_{ds}C_L, where R_{ds} is the equivalent channel resistance, and C_L is the load capacitance on the output node. Obviously, this is just modeling the switch and the node’s capacitance as a simple RC circuit.

For a minimum-sized NMOS device, the worst-case equivalent channel resistance R_{ds} is 625Ω.

The equivalent C_L value depends on the particular node that is being looked at. For this design, three nodes are of particular importance with regard to charge injection and settling time: (1) the input voltage coming out of the sense capacitor, (2) the input node of the op-amp being used for the resettable-gain pre-amplifier stage, and (3) the input node to the comparator in the capacitor array.

At the input node from sense capacitor

As designed, the sense capacitor (C_S) has a nominal value of 1pF, with a reference capacitor (C_R) at 1.05pF, and a “next-stage” capacitance (C_i) of 325pF. This particular input node is influenced by three switches: the one that “resets” it to ground, the one that “resets” the other end of C_S to ground, and the one that “resets” the other end of C_R to ground. Since all three are clocked on the same clock, the settling time on the node can be approximated intuitively as the longest of these three individual settling times.

The only difference that can come about in settling time must come from the equivalent capacitance (C_{eq}) seen by each switch (since they are all minimum-sized NMOS devices with equivalent resistances of 625Ω). So accordingly:

- The switch grounding the non-V_{in} node of C_S sees an equivalent capacitance of 579pF.
- The switch grounding the non-V_{in} node of C_R sees an equivalent capacitance of 586pF.
- The switch grounding V_{in} during the reset state sees an equivalent capacitance of 2.375pF. Clearly this is the winner.

So a simple plug-and-chug gives a 6τ settling time of 8.91ns.

In general, the switch that is directly connected to a node will see the most capacitance (since it sees all of them in parallel). Other switches only see an equivalent capacitance that involves some series combination of capacitors (which dramatically reduces the total capacitance). This reasoning is used in the two remaining nodes.

At the input node of the pre-amplifier’s op-amp

The equivalent capacitance seen at this node (the “minus” input of the opamp in the resettable-gain configuration) is equal to the parallel combination of the input, feedback, and parasitic input capacitances. These
numbers are derived and computed in Appendix A. For this analysis, we just need to know that the final value of this sum is 372\(\mu\)F, which gives a settling time of 1.4ns.

**At the input node to the comparator**

This is the functional “output” of the capacitor array which gets fed to the comparator (to be parsed to a digital output to drive the SAR). It sees all the capacitors in the array (all 2.56pF of it) plus the parasitic capacitance at the input of the op-amp being used as the comparator. The equivalent capacitance for this node is 2.582pF, which results in a \(6\tau\) settling time of 10ns.

### 2.3.2 Charge Injection

A full treatment of charge injection (best case, worst case, where it all comes from, etc.) is given in excruciating detail in Appendix A. But in summary here is what can be expected to happen to node voltages due to charge injection:

- **At** \(V_{\text{in}}\): Worst case: -0.789V; Best case: -0.243V
- **At** \(V_{\text{in-}}\) of the pre-amplifier op-amp: Worst case: -50.3mV; Best case: -13.39mV
- **At** \(V_x\), the input to the comparator: Worst case: -0.726mV; Best case: -0.194mV
ADC Performance

To show that the ADC works (or at least appears to work), six different test cases were made to produce single-bit transitions at the high, middle, and low end of the 8-bit output range. The output waveforms, along with the total static power consumption of each simulation, follow. For the details of the power calculation, see Appendix B, which contains the SPICE output files.

3.1 Waveforms

The input stage was designed to produce 0x00 when $C_S$ is at 1.05pF, and 0xFF when it is at 0.95pF. All ranges in between should scale semi-linearly, as given by the capacitive voltage divider equation:

$$V_{in} = V_{ss} + \frac{C_S}{C_R} (V_{dd} - V_{ss})$$

(3.1)

Correspondingly, this is how some of the strange $C_S$ values came about: taking the full capacitance variation range and dividing it by 256 gives a linear approximation of 0.391F/LSB. This amount is used to “tweak” the transitions. While it is not a perfect method, it gave numbers that were close enough and could be tweaked to generate the appropriate digital output.

In each of the graphs (shown in Figures 3.1 to 3.6), two waveforms are plotted: the solid line shows the (digital) output of the comparator (which will feed the SAR), and the dashed line shows the voltage at the input of the comparator.

Ideally, the sequence of digital output values coming out of the comparator represents the eventual final digital representation, MSB first.

3.2 Disclaimer

When taking the supply rails to $\pm2.75V$, these very same input capacitances on $C_S$ do not produce the same output. However, since the $V_{ref}$ generated internally by the supply-independent bias network is relative to the top voltage rail, this value will fluctuate with a change in $V_{DD}$. In theory, since the variations on the sense-capacitor should merely give a fractional change at the input node between the rails, the output should always be the same. However, due to some inconsistencies, whether it be in my input stage, or in the biasing of the bias network that generates $V_{ref}$, the digital output at $\pm2.75V$ is not consistent for non-edge values.

However, I am sure that despite the grossly mismatching output waveforms, even at $\pm2.75V$, the circuit does measure changes in $C_S$ and produces output proportional to the amount of variation. If given another week or two for the project, I would have first of all tried to improve on the design so that the output would be the same; the first step in doing this would be to determine exactly what output is being given at $\pm2.75V$ supply rails, and/or determine what input capacitance values would produce the exact same transitions that were given here.
Figure 3.1: To generate 0x00: $C_S=1.05\text{pF}$; power = $757\mu\text{W}$

Figure 3.2: To generate 0x01: $C_S=1.04656\text{pF}$; power = $758\mu\text{W}$
Figure 3.3: To generate 0x7F: $C_S=0.9988059p$; power = 746\mu W

![Graph](image)

Figure 3.4: To generate 0x80: $C_S=0.9974802p$; power = 752\mu W

![Graph](image)
Figure 3.5: To generate 0xFE: $C_S=0.9503\mu F$; power = 739\mu W

Figure 3.6: To generate 0xFF: $C_S=0.95\mu F$; power = 738\mu W
Conclusion

In this project, the main goal was to try to get the circuitry working functionally, first as individual blocks, then together as a whole system. Following that, I set out to write up this report and generate the appropriate pictures (schematics, waveforms, etc.) needed to demonstrate that the circuit indeed does work.

However, this alone took most of the time allotted for the project; given more time, I would work more extensively with the conditions of increased supply voltages and refine the design to make it more robust and work the same under both conditions.

4.1 Improving on this Design in the Future

As can be seen in the output waveforms showing the digital output switching, the settling time allowed for each digital bit is much longer than the time it takes for the signal to reach this value. One of the next steps in “improving” this design would be to reduce the bit-cycle period $\tau_u$ and see if the circuit can sample more than the 100kSample/s as it does now.

In addition, the settling time given to the resettable-gain pre-amplifier stage is also much greater than both the hand-calculated value and the settling time of that signal as observed by SPICE output. As with the bit-period, $\tau_{amp}$ can be reduced to give even more frequency improvement.

Also, for performance issues, the $V_{DSAT}$ used in the amplifier design was not the minimum value of 0.1V allowed in the process. Reducing this would improve the amount of settling time needed (all else being equal), which would again improve the sampling frequency.

4.2 Tradeoffs and Other Design Issues

Speed, accuracy, and power consumption wage a battle over any design; an improvement in one comes at the expense of the others. First, the speed of a circuit is directly related to the amount of drive current in it. Increasing speed will increase the current, which will make the circuit consume more power. In addition, desiring a certain sampling frequency severely limits the accuracy (as represented by the number of bits that can be generated in a sample period) that can be done for a given amount of current – to be able to successively generate more bits of a digital representation would require either a dramatic increase in drive current (which consumes more power) to allow for the fast bit-cycling, or an increase in sampling period, which affects the overall speed and frequency of sampling. Finally, power consumption limits the available speed and accuracy: using less power usually involves pulling less current through the devices. Unfortunately, this slows down the speed of the devices, and the frequency of sampling a single bit, which affects the speed and accuracy of an analog-to-digital conversion. In short, while meeting the specs was not too difficult in this project, further optimization to get the “best” implementation of an ADC is quite complicated and requires more involved analysis.
Hand Calculations

In considering whether I should just have stapled all those pages of scribbled equations and number-crunchings into an appendix (as “hand calculations”), I decided that writing it all out would be neater and make more sense (even though it may use a ton of extra paper and some of the intermediate number-crunchings will be omitted in favor of a brief explanation).

5.1 Parasitic Capacitance seen at the input of an op-amp

This number is estimated at 22fF in many of the following calculations. This is because a ballpark figure was needed before actually designing the op-amp itself. I assumed a W/L of 5/1 and just added the resulting $C_{gd}$ and $C_{gs}$ values.

5.2 Switching and Charge Injection

The following calculations were made in the switch design (to make sure that the minimum-sized NMOS transmission gate can act as a reasonable switch.

5.2.1 Settling Time

Assume that the settling time needed is $6\tau_{RC}$, modeling the switch and output node as a simple RC circuit. Then $\tau_{RC}$ is just the product of the equivalent channel resistance and the load capacitance at each node.

First, to find the equivalent channel resistance, just take the partial derivative of the current (in linear mode) with respect to $V_{DS}$, as follows:

\[
(R_{ds})^{-1} = \frac{\partial I_D}{\partial V_{DS}}
= \frac{\partial}{\partial V_{DS}} \left[ \mu_n C_{ox} \frac{W}{L} V_{DS} \left( V_{GS} - V_T - \frac{V_{DS}}{2} \right) \right]
= \mu_n C_{ox} \frac{W}{L} (V_{GS} - V_T - V_{DS})
\]

Therefore:

\[
R_{ds} = \frac{1}{\mu_n C_{ox} \frac{W}{L} (V_{GS} - V_T - V_{DS})}
\] (5.2)

Plug in the numbers corresponding to a minimum-sized NMOS device, and in the worst-case scenario (where $V_{DS} = 0$ and $V_{GS} = 5V$), the equivalent channel resistance is 625Ω.
Second, finding the equivalent capacitance at a node depends on where the node is in relation to the capacitances that it sees. For instance, the relevant switch and the capacitors it sees can be modeled as shown in Figure 5.1. Simple circuit analysis shows the following:

![Diagram](image)

Figure 5.1: Model schematic used to calculate charge injection on node $V_{in}$ due to switches on $C_S$ and $C_R$

- The equivalent capacitance seen at node $V_{in}$ is just the three capacitors in parallel, or

$$C_{eq} = C_S + C_R + C_i.$$  

- The equivalent capacitance seen at the “other” end of $C_S$ (namely, node A in the figure) is given by the series capacitance equation (where $C_S$ is in series with the sum of $C_R$ and $C_i$):

$$C_{eq} = \frac{C_S(C_R + C_i)}{C_S + C_R + C_i}$$  \hspace{1cm} (5.3)

- Correspondingly, at the “other” end of $C_R$:

$$C_{eq} = \frac{C_R(C_S + C_i)}{C_S + C_R + C_i}$$  \hspace{1cm} (5.4)

With this design’s numbers ($C_S = 1pF$, $C_R = 1.05pF$, $C_i = 325fF$), these equivalent capacitances can be found at the input stage:

- At $V_{in}$ directly, $C_{eq} = 2.375pF$
- At $C_S$, $C_{eq} = 579fF$
- At $C_R$, $C_{eq} = 586fF$

This technique is applied in a similar fashion at the input node of the pre-amplifier stage (where the three capacitances in question are $C_f = 25fF$, $C_i = 325fF$, and the parasitic $C_p = 22fF$) to derive the following input capacitances:

- At $V_{in-}$ directly, $C_{eq} = 372fF$
- At the other end of $C_i$, $C_{eq} = 41fF$
- At the other end of $C_f$, $C_{eq} = 23.3fF$

And at the input to the comparator, the total capacitance is just the sum of all the capacitors in the capacitor array (2.56pF) plus the parasitic capacitances at the op-amp input ($C_{gs} + C_{gd}$), which sums to a grand total of $C_{eq} = 2.582pF$.

Given these, the RC product at any node of an NMOS switch can be found quite trivially by selectively multiplying the appropriate equivalent capacitance with the channel resistance of the device.
5.2.2 Charge Injection

At the input node from the sense capacitor

Despite all the fun that was had in calculating all the different $C_{eq}$ values around the $V_{in}$ node, the only switch whose charge injection has a lasting effect on $V_{in}$ is the one that is directly connected to ground. While the injection of charge onto the nodes on the other side of $C_S$ and $C_R$ will temporarily change $V_{in}$ as given by the capacitive voltage divider equation, these nodes will then be tied to a power rail and will discharge this extra charge long before the voltage at $V_{in}$ matters.

Worst Case:
The amount of charge injected into a node is approximately one half of the total charge stored in the channel while the switch is on. In other words,

\[
\text{since } Q_{\text{channel}} = WLC_{ox} \left( V_{gs} - V_T \right), \\
Q_{\text{injected}} = \frac{1}{2} WLC_{ox} \left( V_{gs} - V_T \right)
\]

(5.5)

In the above equations, $V_{gs}$ refers to the gate-source voltage while the device is on (i.e., when the gate voltage is $V_{DD}$). Plugging in the appropriate numbers for the channel width and length and our process parameters, $Q_{\text{injected}} = -1.875 \times 10^{-15} \text{ coul}$.

Using the relation $Q = CV$, we can figure out the amount that this amount of charge will change a voltage given the amount of capacitance on the node.

For the switch connecting $V_{in}$ directly to ground during the reset phase, we can get the previously calculated $C_{eq}$ value and make the following calculations:

\[
\Delta V = \frac{\Delta Q}{C_{eq}} \\
= \frac{Q_{\text{injected}}}{C_S + C_R + C_i} \\
= \frac{-1.875 \times 10^{-15} \text{ coul}}{2.375 \text{ pF}} \\
= -0.789 \text{ mV}.
\]

(5.6)

In determining whether this amount of charge injection is tolerable, we look at the “LSB value” of the desired input range from 0 to -0.125V (which is an estimated swing for $V_{in}$ as $C_S$ swings between 1.05pF and 0.95pF). For an 8-bit design, the necessary resolution is $1/256$ of this range, which is -0.488mV! Clearly, the worst-case charge injection is a problem, as it is nearly twice the desired resolution.

Best Case:
The preceding worst-case charge injection numbers occur when the switches are clocked with a small rise and fall time (fast switching). In the best case, using a slow rising and slow falling control signal, the analysis can be repeated with most of the exact same numbers. The only difference is that $Q_{\text{inj}}$ is considerably less.

For the case of a slow-switching clock:

\[
\Delta Q_{\text{injected}} = C_{gd} V_T
\]

(5.7)

Using our EE140 process parameters, $C_{gd}$ is found to be 500aF for a minimum-sized MOS device. Using a $V_T$ of 1V, the total charge injected onto a node by one minimum-sized switch is $-500 \times 10^{-18} \text{ coul}$. Using this as the value of $Q_{\text{inj}}$ and running through the analysis again gives a $\Delta V_{in}$ of 0.243mV. This value is less than half of $V_{LSB}$, and is good enough to give the desired resolution.

20
At the input node of the pre-amplifier’s op-amp

The total charge injection considerations in this case must take into account the effects of three switches—two of which affect the node through a capacitive voltage divider, and a third that is directly connected to the node—and three capacitors: input, feedback, and parasitic. Unfortunately, since none of the “indirect” nodes that are reset during the reset phase are subsequently connected to low-impedance voltage sources (as happened with $V_{in}$ on the input stage), we cannot simply neglect the effect of any of these switches.

Worst Case:

Since each of these NMOS devices is minimum sized (as was the case in the input stage), the same amount of charge is stored in the device’s channels when they are conducting, which is the same $-1.875 \times 10^{-15}$ coul of charge as calculated before in equation (5.5).

Using this value, and equation (5.6), which relates the change in voltage to the change in charge, we can selectively calculate the change in charge at three nodes: (1) directly at $V_{in-}$ due to its reset switch, (2) on the “other” side of $C_i$, and (3) the “other” side of $C_f$.

First, taking up the “feedback bypass” switch that shorts the input and output of the opamp during the reset phase:

$$\Delta V_{in-} = \frac{Q_{inj}}{C_i + C_f + C_p}$$
$$= \frac{-1.875 \times 10^{-15}}{372 \times 10^{-15} \text{F}}$$
$$= -5.04 \text{mV}$$

Next, the switches that ground $C_i$ and $C_f$ are considered. Similar to the devices that grounded $C_S$ and $C_R$ in the input stage, the charge injected by these devices turning off only affects the node in question indirectly, through a capacitive voltage divider. A two-stage process is used to calculate the effect of the injected charge on $V_{in-}$.

For the device that grounds $C_i$:

$$\Delta V_i = \frac{Q_{inj} (C_i + C_f + C_p)}{C_i (C_f + C_p)}$$
$$= \frac{-1.875 \times 10^{-15} (372 \times 10^{-15})}{325 \times 10^{-15} (47 \times 10^{-15})}$$
$$= -45.7 \text{mV}$$

And by the capacitive voltage divider equation:

$$\Delta V_{in-} = \frac{C_i}{C_i + C_f + C_p} \Delta V_i$$
$$= \frac{-45.7 \text{mV} (325\text{F})}{-39.9 \text{mV}}$$
$$= -45.7 \text{mV}$$

A similar analysis can be done by reversing the $C_i$ and $C_f$ terms to get the effect of the charge injection due to the device that grounds the feedback capacitor. This analysis gives a $\Delta V_{in-}$ of -5.40 mV.

In total, the worst-case charge injection would inflict an astounding 50.3 mV of “noise” on this input!

Best Case:

With a slow-switching reset signal, $Q_{inj}$ can be reduced to $-500 \times 10^{-18}$ coul (this analysis is the same as was done previously for the minimum-sized device in equation (5.7)). Using this as the value of $\Delta Q_{inj}$ and running through the entire $\Delta V_{in-}$ analysis again results in the following:

21
• From the feedback switch: $\Delta V_{in-} = -1.344\text{mV}$
• From the switch on $C_i$: $\Delta V_{in-} = -10.6\text{mV}$
• From the switch on $C_f$: $\Delta V_{in-} = -1.441\text{mV}$

The total voltage change due to charge injection comes to -13.39mV, which is much bigger than $V_{LSB}$ at the input.

To reduce the effect of this noise, the technique of “mirroring” the capacitances on the other input ($V_{in+}$) will be done to “cancel out” some of the charge injection by making sure the same amount of injection (or close to it) ends up on both input terminals. Assuming the op-amp is good enough, an equal “noise” signal (such as this charge injection) that shows up on both inputs is discarded.

**At the input node to the comparator**

For this node, the charge injection to be concerned with comes from the feedback switch of the comparator, which is closed during the sample state. As before, assuming a minimum-sized NMOS device for this switch, the voltage variation due to charge injection can be calculated.

The main difference in the consideration of charge injection on this node is that $V_{LSB}$ is now $1/256$ of $V_{ref}$; with a $V_{ref}$ of 1.35V, this is 5.27mV.

**Worst Case:**

In this case, $C_{eq} = 2^n C_0$, where $n$ is 8 (for this 8-bit design) and $C_0$ is arbitrarily chosen to be 10fF. Or in this specific case, $C_{eq} = 2.56$pF. Add in the parasitic capacitance at the input stage (of about 22fF), and the grand total $C_{eq}$ comes to 2.582pF.

$$\Delta V_x = \frac{Q_{mj}}{C_{eq}} = \frac{-1.875 \times 10^{-15}}{2.582 \times 10^{-12}} = -0.726\text{mV}$$

Even in the worst case, the voltage change due to charge injection is a factor of seven less than the LSB voltage. So this means that the signal on this switch can rise and fall quickly without too much damage...

**Best Case:**

Again, the difference between worst and best case is the amount of charge injected from the device – instead of $-1.875 \times 10^{-15}$ coul, the charge from a slow-switching clock is $-500 \times 10^{-18}$ coul. Redoing the calculations yields a $\Delta V_x$ of -194μV.

This value is even less of a fraction of $V_{LSB}$!

### 5.3 Op-Amp and Bias Circuity Device Sizes

This section details the equations used in deriving the device sizes of the transistors used in the folded-cascode op-amp and the “wide-swing” supply-independent bias circuit.
5.3.1 The Folded Cascode Op-Amp

As stated in the section on amplifier design issues, a $V_{DSAT}$ of 0.15V is chosen for all devices in the op-amp except for the tail current device. That one has a $V_{DSAT}$ of 0.35V. Given that the nominal current running through the input device is 10µA, this means that 20µA runs through each of the PMOS “current source” devices (M4A/M4B), and the tail device (M9). Knowing the $V_{DSAT}$ and $I_D$ values, a reasonable calculation of the desired sizes of each device can be made.

But for better accuracy, channel-length modulation is factored into the equations. What this means is that all the $V_{ds}$ values (operating point values) are figured out.

Looking at the topology (see Figure 2.1), we see that all devices M1-M4 are set up in a wide-swing cascode configuration. This means that the devices closest to the rails are right at the edge of saturation (i.e., $V_{ds} = V_{DSAT} = 0.15V$). The cascoded devices, on the other hand, have their drains tied to the gate of these previously mentioned “near-the-rail” devices; that is, the drain of M3 is tied to the gate of M4, and the drain of M2 is tied to the gate of M1. As such, these nodes are at $V_T + V_{DSAT}$ away from the rails, resulting in a $V_{ds}$ equal to $V_T$ for M3 and M2. In our process, this is 1V. Finally, for the NMOS tail device (M9), we want it to be right at the edge of saturation (to maximize the input swing), so assume that it’s $V_{ds} = V_{DSAT} = 0.35V$.

Given the voltages at these nodes, the following equation is used to determine the desired device size of each transistor in the op-amp:

$$\frac{W}{L} = \frac{2I_D}{\mu C_{ox} (V_{DSAT})^2 (1 + \lambda V_{ds})} \quad (5.8)$$

So for each device, we can plug in the appropriate parameters and use this equation to come up with the desired W/L ratio, as shown in Table 5.4.

<table>
<thead>
<tr>
<th>Device</th>
<th>$I_D$</th>
<th>$V_{DSAT}$</th>
<th>$\mu C_{ox}$</th>
<th>$V_{ds}$</th>
<th>W/L</th>
</tr>
</thead>
<tbody>
<tr>
<td>M9</td>
<td>20µA</td>
<td>0.35V</td>
<td>200µA</td>
<td>0.15V</td>
<td>1.577</td>
</tr>
<tr>
<td>M5</td>
<td>10µA</td>
<td>0.15V</td>
<td>200µA</td>
<td>4.50V</td>
<td>3.06</td>
</tr>
<tr>
<td>M4</td>
<td>20µA</td>
<td>0.15V</td>
<td>100µA</td>
<td>0.15V</td>
<td>17.5</td>
</tr>
<tr>
<td>M3</td>
<td>10µA</td>
<td>0.15V</td>
<td>100µA</td>
<td>3.70V</td>
<td>6.488</td>
</tr>
<tr>
<td>M2</td>
<td>10µA</td>
<td>0.15V</td>
<td>200µA</td>
<td>1.00V</td>
<td>4.04</td>
</tr>
<tr>
<td>M1</td>
<td>10µA</td>
<td>0.15V</td>
<td>200µA</td>
<td>0.15V</td>
<td>4.379</td>
</tr>
</tbody>
</table>

In addition, the appropriate bias voltages need to be provided to keep all these devices in saturation are as follows:

- On the gate of M4 ($V_{bp1}$), the voltage needs to be at least $V_T + V_{DSAT}$ below the top rail; at $\pm 2.5V$, this translates to $V_{bp1} \leq 1.35V$.

- On the gate of M3 ($V_{bp2}$), the voltage must be no greater than $V_T + 2V_{DSAT}$ below the top rail; at $\pm 2.5V$, this translates to $V_{bp2} \leq 1.2V$.

- On the gate of M2 ($V_{bn}$), the voltage must be at least $V_T + 2V_{DSAT}$ above the bottom rail; at $\pm 2.5V$, this translates to $V_{bn} \geq -1.2V$.

- On the gate of M9 (also $V_{bn}$ by design), the voltage must be at least $V_T + V_{DSAT}$ above the bottom rail; at $\pm 2.5V$, this translates to $V_{bn} \geq -1.15V$.

So for the shared biasing voltage $V_{bn}$, so long as the condition for M9 is satisfied, then M2 and M1 will remain in saturation also.
5.3.2 Supply-Independent Bias Circuitry

For reasons of bias simplicity, the folded-cascode op-amp mentioned above was designed to have both of its NMOS bias voltages shared (as implemented by the multiple-$V_{DSAT}$ design). Still, three bias voltages need to be pulled out of whatever bias topology is chosen: two PMOS biases (to facilitate the wide-swing PMOS cascode), and one NMOS bias (since the wide-swing design is already taken care of in the op-amp’s topology itself in M1 and M2).

The final bias topology can be thought of in two parts: the “classical” supply-independent bias setup, and the “third branch” used for wide-swing. For power reasons, only 5μA of current runs through each branch of the bias circuit.

The “classical” supply-independent bias

To determine the optimal device sizes, $V_{DSAT}$ must first be chosen. For the purposes of matching the NMOS tail device in the op-amp, all $V_{DSAT}$ values are chosen to be 0.35V. It will be the job of the third branch to split this up and create PMOS bias voltages for the PMOS cascode devices in the opamp. Furthermore, since this bias circuit will be used only for steady voltage generation, a length of 10μm is chosen to minimize the effects of channel-length modulation.

The “third branch”

As with the other two branches of the bias network, an optimal 5μA of current runs through this branch too. The only difference is that since this branch will be biasing the wide-swing devices, its PMOS wide-swing devices have a $V_{DSAT}$ of 0.15V (for matching purposes). Further, a blind length of 10μm is not used here to improve device matching.

Plug-and-Chug

Again, equation (5.8) can be used to determine the sizes of each device. Taking into account channel-length modulation as before, the inputs to the equation and resulting device sizes are shown in Table 5.5.

Table 5.5: Results of plug-and-chug calculations to figure out bias circuit device sizes. (*Note that channel length modulation is ignored for the classical source branches)

<table>
<thead>
<tr>
<th>Device</th>
<th>$V_{DSAT}$</th>
<th>$\mu C_{ox}$</th>
<th>$V_{ds}$</th>
<th>$W/L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1A</td>
<td>0.175V</td>
<td>200μ</td>
<td>*</td>
<td>16.327/10</td>
</tr>
<tr>
<td>M1B</td>
<td>0.350V</td>
<td>200μ</td>
<td>*</td>
<td>4.082/10</td>
</tr>
<tr>
<td>M2AB</td>
<td>0.350V</td>
<td>200μ</td>
<td>*</td>
<td>4.082/10</td>
</tr>
<tr>
<td>M3AB</td>
<td>0.350V</td>
<td>100μ</td>
<td>*</td>
<td>8.163/10</td>
</tr>
<tr>
<td>M3C</td>
<td>0.150V</td>
<td>100μ</td>
<td>1.6V</td>
<td>4.04/1</td>
</tr>
<tr>
<td>M4C</td>
<td>0.150V</td>
<td>100μ</td>
<td>0.15V</td>
<td>4.38/1</td>
</tr>
<tr>
<td>M9C</td>
<td>0.350V</td>
<td>100μ</td>
<td>3.85V</td>
<td>1.085/3</td>
</tr>
</tbody>
</table>

5.4 Op-Amp Performance Metrics

The following calculations were made to determine the frequency response of the folded-cascode op-amp as designed:
5.4.1 Low-Frequency Gain

For a folded-cascode topology, the gain is just \( G_m R_o \), where \( G_m \) is just the \( g_m \) of the input device (M5), and \( R_o \) is the equivalent resistance seen at the output node.

To get \( R_o \), note that for the folded cascode, it can be simplified to \( R_{up} || R_{dn} \), which is given by the following:

\[
R_{up} = r_o [1 + g_m (r_o || r_{o5})] \\
R_{dn} = r_o (1 + g_m r_o)
\]

For those equations, we can find \( g_m \) and \( r_o \) by using these equations (which take into account channel-length modulation):

\[
g_m = \sqrt{2I_D J C_{ox} \frac{W}{L} (1 + \lambda V_{ds})} \quad (5.9) \\
r_o = \frac{1 + \lambda V_{ds}}{\lambda I_d} \quad (5.10)
\]

Plugging in the appropriate numbers yields this:

<table>
<thead>
<tr>
<th>device</th>
<th>( I_D ) (( \mu ))</th>
<th>( \mu C_{ox} )</th>
<th>( W/L )</th>
<th>( V_{ds} )</th>
<th>( g_m )</th>
<th>( r_o )</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>10 ( \mu ) 200 ( \mu )</td>
<td>4.379</td>
<td>0.15 V</td>
<td>*</td>
<td>1.015 M( \Omega )</td>
<td></td>
</tr>
<tr>
<td>M2</td>
<td>10 ( \mu ) 200 ( \mu )</td>
<td>4.04</td>
<td>1.00 V</td>
<td>9.43 m( \Omega )</td>
<td>1.10 M( \Omega )</td>
<td></td>
</tr>
<tr>
<td>M3</td>
<td>10 ( \mu ) 100 ( \mu )</td>
<td>6.488</td>
<td>3.70 V</td>
<td>133 ( \mu ) ( \Omega )</td>
<td>1.37 M( \Omega )</td>
<td></td>
</tr>
<tr>
<td>M4</td>
<td>20 ( \mu ) 100 ( \mu )</td>
<td>17.5</td>
<td>0.15 V</td>
<td>*</td>
<td>507.5 k( \Omega )</td>
<td></td>
</tr>
<tr>
<td>M5</td>
<td>10 ( \mu ) 100 ( \mu )</td>
<td>3.06</td>
<td>4.50 V</td>
<td>133 ( \mu ) ( \Omega )</td>
<td>1.45 M( \Omega )</td>
<td></td>
</tr>
</tbody>
</table>

So we know that \( G_m \) of the entire amplifier is just \( g_{m5} \) of the input device, or 133 \( \mu \) \( \Omega \). \( R_{up} \) is calculated to be 69.87 M\( \Omega \), and \( R_{dn} \) is a whopping 10.53 G\( \Omega \). Taken in parallel, the total \( R_o \) is 69.4 M\( \Omega \). So the low frequency gain is found by multiplying \( G_m \) and \( R_o \), which gives 9230.

5.4.2 Dominant Pole Frequency

This can be found as just the inverse of the product \( R_o C_L \), where \( C_L \) is the load capacitance on the output node. Since for our purposes, the op-amp as a comparator will be driving digital logic, an estimate of the load capacitance would be 50\( \text{\mu F} \). Given this, the pole frequency can be found as \( 1/R_o C_L \), which is 45.86 kHz.

5.4.3 Unity Gain Frequency

Using previously-calculated values, the following can be done to find the unity-gain frequency:

\[
\omega_u = A v_0 \omega_3 dB \\
= 9230 \times 45.86 kHz \\
= 423.3 MHz
\]
5.5 Settling Time Calculations

The following settling-time calculations were used in determining the feasibility of an 8-bit design at 100kHz sample/s. These numbers also served as a starting point for the exact timing of the signals that drive the circuits.

5.5.1 Bit-cycle period

With this nominal 10μA current chosen, we can go ahead and calculate the minimum bit-cycle period (if the amplifier is to be used as a comparator to feed the SAR). Modeling the output needed as an RC delay, and knowing that only 1 RC-delay would be needed for the input to a digital circuit (the SAR) to be happy, the time (τ_{bit}) needed for the op-amp acting as a comparator to produce the correct value would be R_0C_L, which is also conveniently equal to \( \frac{1}{\omega_{dB}} \).

Here is some math used to rewrite this expression:

\[
\tau_{bit} = \frac{1}{\omega_{dB}} = \frac{A_{V_0}}{\omega_u} \approx \frac{A_{V_0}C_L}{g_m} = \frac{A_{V_0}C_LV_{DSAT}}{2I_D} \tag{5.11}
\]

Above, the \( I_D \) and \( V_{DSAT} \) values correspond to a single input device of the amplifier. So now these parameters need to be specified:

1. \( V_{DSAT} \): 0.15V for the input device.
2. \( C_L \): If it is driving some digital circuitry, choose 50pF for a rough ballpark figure.
3. \( A_{V_0} \): This is the minimum gain needed to rail one LSB of the capacitor array. In theory, the general form for this is as follows:

\[
A_{V_{min}} = \frac{V_{DD} - V_{SS}}{V_{LSB}} = \frac{V_{DD} - V_{SS}}{V_{sat}} = \frac{2^n(V_{DD} - V_{SS})}{V_{ref}} \tag{5.12}
\]

Considering an 8-bit design with a \( V_{ref} \) of 1.35V and ±2.5V supply rails, the minimum gain is found to be 948.

To be conservative and account for fractions of an LSB that may show up on the differential input, set the gain to 1000. Using this, and the previously determined values, we can plug and chug through equation (5.11) and find a minimum bit-cycle period of 0.38μsec. For eight bits, a conservative estimate would reserve a total of 4μsec just for bit-cycling.

The actual numbers derived indicate that less than this amount of time is needed for a bit-cycle period; with an open-loop gain of at least 8000, the output of the op-amp as a comparator will tend to rise and hit the top or bottom rail faster than expected with a gain of only 1000.

5.5.2 Sampling Period

In addition to using this amplifier as a comparator, since I’m lazy, I’d like to see how well this same amplifier would work as the guts of a retablet-gain amplifier. Following a similar analysis as was done above in deriving equation (5.11), a similar expression for \( \tau_{sample} \) is found to be:

\[
\tau_{sample} = \frac{\ln(2^n)A_{CL}2^nC_0V_{DSAT}}{2I_D} \tag{5.13}
\]

Using the same \( I_D \) and \( V_{DSAT} \) as above in the comparator, the remaining parameters are chosen conservatively as follows:
1. $C_0$ is chosen at 10fF.

2. $A_{CL}$ is chosen to be 13, which is a conservative guess based on the needs of the input stage and $V_{ref}$.

Working these numbers through equation (5.13) gives a $\tau_{sample}$ of 3.75μsec.

5.6 Input-Capacitor and Pre-Amplifier Stages

These calculations were performed prior to any actual simulations of the final design. They are as follows:

5.6.1 Input Range and Pre-Amplifier Gain

The analog input to be digitized comes from the variation of a sense capacitor $C_S$. Varying this capacitance will produce a voltage perturbation on the output node of this input stage, $V_{in}$. Additionally, since the charge-redistribution capacitor array is configured to work only for positive analog values, we would want this $V_{in}$ value to always be negative.

$$\frac{V_{in}}{V_{DD} - V_{SS}} = \frac{C_S}{C_S + C_R} \quad (5.14)$$

For this to work absolutely, $C_R$ (the reference capacitor) is chosen to be 1.05pF, since one extreme has $C_S$ maxing out at 1.05pF (which will correspond to a $V_{in}$ of 0V).

Using equation 5.14, we can find the other extreme of the input range (which should produce all 1s as a digital output): since $C_S$ is never smaller than 0.95pF, we plug this value in and solve for $V_{in}$. With ±2.5V rail voltages, we get the following:

$$V_{in} = V_{SS} + \frac{C_S}{C_S + C_R} (V_{DD} - V_{SS})$$

$$= -2.5V + \frac{0.95}{0.95 + 1.05}(5V)$$

$$= -0.125V$$

In order to get a 11111111 digital output corresponding to this value, this -0.125V needs to be ramped up by the pre-amplifier to a full 1.35V (which is the value of $V_{ref}$ of the capacitor array). A quick division (1.35V/-0.125V) gives a necessary pre-amplifier gain of -11. When used in timing analysis, a more conservative value that will allow for more settling time would be -13 or -14.
SPICE Stuff

All of the SPICE decks are included on the accompanying floppy disk, and are therefore left out here. In addition, just in case my floppy drive does funky things to disks, the contents are also available online at http://www-inst.eecs.berkeley.edu/jchoy/project/ee140/proj.tar.gz.

Any questions or concerns regarding the electronic copy of these files can be addressed to jchoy@cory.eecs.berkeley.edu.

A few notes about the contents of the disk/tarball:

1. three directories are created: “decks,” “include,” and “out.” The include directory contains subcircuit definitions only and cannot standalone and be run in SPICE. The decks directory contains all the “top-level” code that gets parsed through SPICE and which depends on files in the include directory. The out directory contains all the SPICE output files (useful to find device parameters and power measurements) that are referenced in this report. In theory, they can be generated again by just re-running the appropriate SPICE deck out of the “decks” directory.

2. Within the “decks” directory are several files called topXX.sp. Each one of these is a derivative of my top-level SPICE deck; the difference between them is the value used for the sense capacitor and the bitline control signals used in “faking” the behavior of a SAR on the capacitor array.

6.1 SPICE Output: proof that I really did run this in SPICE

Following this little blurb here, I will attach a ton of SPICE output files. The relevant sections of each output file are highlighted and annotated. However, even these attachments are a condensed version of the total amount that I could have put here.

The attached are as follows:

- Complete output file from simulating the folded-cascode amplifier with supply-independent biasing (serves two purposes: to verify the currents and $V_{DSAT}$ of the topology, and to generate the Bode plots and compare performance parameters), in the file called “foktest.out.”

- Sample SPICE deck used in running full simulations (this is top7F.sp). Of course, all decks are included in the tarball.

- Abbreviated output file from running a full simulation to generate 0x00 output - this contains a full output file except for the device parameters.

- First and last pages of each of the other five full simulations (0x01, 0x7F, 0x80, 0xFE, 0xFF): in each, the only thing that matters is the value used for $C_S$, and the static power consumption

- First and last page of full simulation (0x00) using 2.75V rails. This is merely to show power consumption.
This list is far from complete, but it contains enough to justify the SPICE comparison numbers given in earlier sections of this report. Again, all the full output files (hundreds of pages, I imagine) can be found on the included disk or in the tarball available for download.

6.2 Miscellaneous MetaWave Graphs

In testing the stability of my folded-cascode amplifier, I ran it through a transient analysis in unity-gain feedback, and also ran an AC analysis to generate Bode plots. All of this was run with the foldtest.sp deck included on the disk. The plots are shown in Figures 6.1 and 6.2.

Figure 6.1: Bode Magnitude and Phase plot for the folded-cascode op-amp. Unity-gain frequency and phase margin are identified on the graph
Figure 6.2: Transient behavior of folded-cascode op-amp in unity-gain feedback. Notice that this looks like only one waveform – there are really two curves there; but since the phase margin is large enough, the amplifier is stable.
Miscellaneous Schematics Not Included Elsewhere

Some of the following have been described, but their transistor-level schematics have not shown up; so they are included now.

7.1 Input Stage

This is the switched-capacitor input. Depending on the values of sample and reset, the voltage at node $V_{in}$ will either be zero, or given by the capacitive voltage divider equation.
7.2 Resettable Gain Pre-Amplifier Stage

This is the resettable-gain pre-amplifier stage. It is a bit more complex than a traditional resettable-gain circuit to minimize charge injection: while all that stuff connected to the positive terminal of the op-amp may look useless (after all, it's switching between ground and ground), it closely tracks the amount of charge injection that occurs on the negative terminal to counteract the net effect of charge injection.
7.3 Capacitor Array

This is the charge-redistribution ADC capacitor array as implemented in this project. Actually, full complementary CMOS gates were used to connect the capacitors to $V_g$, but that is not shown above for lack of space.