Michael Lorek

Alberto Puggelli

11/30/2010

<u>**Fall 2010 EE247 Project Report: 6 Bit 300MS/s Flash ADC**</u>

1) Analysis and Hand Calculations

a) Overall Architecture

We chose to implement a flash ADC for our project, mainly because it seemed to be a straightforward architecture to implement. We also sought to avoid circuit functionality dependence on software algorithms, as is inherent in the SAR architecture's logic block. As we got moving in our ADC design, we made some design choices to enable better performance and lower power, albeit at the cost of increased complexity. This presented us with some design challenges – these issues, their solutions, and other aspects of our design process will be presented in the following text.

To minimize power consumption in our ADC, we decided to use an interpolation scheme. This enables lower power consumption in two ways. Most obviously, it reduces the number of preamplifiers, drastically reducing static power. The use of interpolation also makes the effective LSB at the preamplifier input increase by a factor of 2, thus relaxing the input-referred offset specifications. This enables the use of smaller preamplifier input transistors, resulting in less parasitic capacitance and lower g_m (power) circuits. Considering a large interpolation factor of 8, the benefits did not become extremely clear. The power benefits of lower preamp gain, limited by comparator offset attenuation, may be diminished by larger

preamplifier output swing linearity requirements, as a result of the larger LSB input. Therefore, we chose a moderate interpolation factor of 4 for our design. This reduces the number of comparators in our design from 64 to 17.

We also chose to implement the auto-zero offset cancellation technique (Mehr JSSC 1999) to further reduce our input-referred offset. Using this technique allowed us to cancel a large portion of the preamplifier offset. See Figure 1 below as well as the proceeding calculations. This, in turn, enabled lower preamplifier gain as the residual input-referred comparator offset that can be tolerated becomes greater. Lower preamplifier gain directly translates into lower g_m, current, and power.
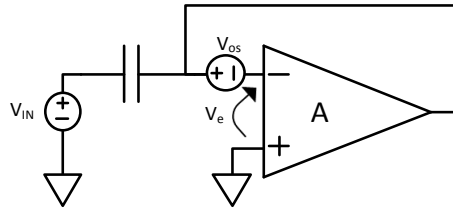

Figure 1: Amplifier Offset Cancellation Error

$$Q_{cap} = C(V_{in} - (V_{os} + V_e))$$
$$-AV_e = V_{os} + V_e \Rightarrow V_e = -\frac{V_{os}}{A+1}$$
$$Q_{cap} = C\left(V_{in} - \left(\frac{AV_{os}}{A+1}\right)\right)$$

The final top-level implementation of these techniques can bee seen in Figure 2 below.

b) *Comparator Design*

Seeking to minimize power dissipation, we chose to use a StrongARM latch for our comparator. This comparator topology uses no static power and outputs full rail-to-rail voltage levels. Since no load capacitance was specified for our ADC

output, the comparator is strictly self-limited. Therefore, small devices were used to increase the speed of the circuit. We chose W_{min} = 100nm for use throughout the ADC design; we were advised against using devices with L<100nm due to poor matching. Our comparator design process began by using all minimum size devices and this worked very well. These device sizes yielded a great regenerative response, although running a comparator overdrive test revealed hysteresis problems. Hysteresis was eliminated by adding pre-charge switches to the inverter "ground" nodes. The improved overdrive recovery test results are shown in Figure 7. It can be seen that the comparator outputs settle in much less than a half clock cycle for f_{clk} = 300MHz – this was exploited in the time allocation to different clock phases. The final comparator schematic can be seen in Figure 4.

c) *Preamplifier Design*

In order to design the process-sensitive preamplifier circuit effectively, we characterized some aspects of the 65nm CMOS process. All transistor devices were sized using the constant current density (V*) sizing method, with plots generated using Spectre simulation. From simulation we found that NMOS input devices biased at V*=120mV results in a Gain-Bandwidth product GBW~45GHz, V*=110mV bias results in GBW~28GHz, and V*=100mV bias yields GBW~15GHz. These simulations were run with L=100nm, and an additional 200aF load capacitance to model the comparator input capacitance. This information proved useful when considering power-speed tradeoffs in the preamplifier design.

Our initial preamplifier design iterations used PMOS diode loads, however we realized that these transistor loads were creating non-linear effects given the

relatively large output swings required. Therefore, our final design uses linear resistor loads and maintains good linearity. Much care was also taken in the choice of output common-mode level. The preamp input and output common-model levels are directly coupled during the auto-zero offset sampling phase as the preamp is tied into unity gain feedback.

The preamp output impedance must also be carefully considered in the design. The preamp outputs ideally act as voltage sources that drive the interpolation resistors. Therefore, if the preamp output impedance is too high, the voltage division that occurs between preamp outputs attenuates the desired comparator input signal and results in loss of dynamic range. To help mitigate this problem, we used the maximum interpolation resistance value allowed of 10KΩ. In common flash architectures, the value of the resistor string at the input of the latches is also determined by the kick-back charge injected on the resistive taps when the latches are enabled. In our differential design, the contribution of kick-back charge is not critical. Since the two inputs of the latches see equal impedance looking into the interpolation resistors, the kick-back charge results only in a common mode voltage shift. Therefore, we were able to choose the maximum value for the interpolation resistors without affecting the overall functionality.

The design details of our final preamplifier follow. A V*=120mV was chosen for low-power operation without the drastically reduced speeds associated with biasing the input devices with a lower V*. We found that our system maintained adequate SNR with a preamp load resistance up to 10kΩ. This was maximized to minimize the g_m and power. We designed for a voltage gain A_v=5; this was

approximately the minimum gain to maintain <0.5LSB of input-referred voltage offset.

$$A_v = g_m R_{out} = g_m \left(R_L \parallel r_o \right) \cong g_m R_L$$

$$5 = g_m (10 k\Omega) \Rightarrow g_m = 500 \mu S$$

$$I_D = \frac{g_m (V^*)}{2} = \frac{500 \mu S (120 mV)}{2} = 30 \mu A$$

For V*=120mV, from Spectre: $I_D = 1.5898 {\mu A}/{0.1 \mu m}$

$$W_{input} = \frac{30 \mu A}{1.5898 \mu A} * 0.1 \mu m = 1.887 \mu m$$

$$V_{CM,in} = V_{CM,out} = V_{DD} - I_D R_L = 800 mV$$

$$V_{swing,out} = V_{DD} - V_{CM,out} = 300 mV > A_v LSB_{Preamp,in}$$

From Spectre AC Analysis: $A_v = 4.507$

$$\sigma_{offset,preamp} = \frac{8mV}{\sqrt{1.887 \mu m (0.1 \mu m)}} = 18.416 mV$$

$$\sigma_{offset,input,total} = 3 \sqrt{\left[\sigma_{offset,preamp} \left(1 - \frac{A_{preamp}}{1 + A_{preamp}} \right) \right]^2 + \left(\frac{\sigma_{offset,latch}}{A_{preamp}} \right)^2}$$

$$\sigma_{offset,input,total} = 19.43 mV < 20 mV < 0.5 LSB$$

The AC response of the final preamplifier design, with 200aF load capacitance can be seen in Figure 6, and the final schematic is shown in Figure 5.

d) *Clock Phase Design*

Since the maximum input signal frequency is $f_{in,max}$ = 150MHz, we designed our ADC to operate at a frequency of approximately f_s = 2 $f_{in,max}$ = 300MHz. This results in a sampling period of 3.33ns. For our system to operate as desired, the clock period needs to be divided into 3 phases. Phase 1 corresponds to the auto-zeroing phase, in which both the input signal and offset voltages are sampled onto the decoupling capacitor. During this phase, the amplifiers operate in unity-gain

feedback with the feedback switches closed. In the second phase, the decoupling capacitors are connected to the resistive DAC and the ideal preamp input differential voltage becomes $V_{diff} = V_{ref} - V_{in}$. In this phase, the amplifiers operate in the open-loop configuration and amplify the inputs. This sampling scheme introduces a 180° phase shift, but this can easily be compensated by introducing a sign inversion later in the differential chain. In phase 3, the comparator clocks are toggled and they generate the thermometer code for conversion by the digital backend.

In order to devise the optimal partitioning of the clock period into the three phases, we made the following considerations:

i) In phase 1, the preamps operate in unity gain configuration; their bandwidth is approximately A times larger than during the open-loop operation, where A is the open-loop gain of the preamplifier.

ii) Using the overdrive recovery test, our comparator is able to resolve a worst-case input signal sequence in less than 250ps.

From this analysis, we allocated t_3=300ps for the comparator decisions in phase 3. The remaining clock period was split between the first and second phases with a ratio of 1:5. This dissimilarity is to account for the increased bandwidth of the amplifier in unity-gain feedback – our preamplifier open-loop gain is around 5. We thus chose t_1 = 600ps and t_2 = 2.6ns.

The final timing is obtained by considering two non-ideal effects:

i) The end of phase 1 and the beginning of phase 2 should be separated to avoid loss of charge from the decoupling capacitor;

ii) The feedback path switch should be opened before the switch connected to the input to avoid creating input-dependent switch charge injection. We note that the feedback switch always operates at the same operating point, so the charge that it injects on the decoupling capacitor results in an offset contribution that is canceled by the differential architecture.

We thus open the switch connected to the input 50ps after the switch in the feedback path and wait another 50ps before beginning phase 2. Finally, we made phase 3 completely overlap with phase 2 and the two phases finish at the end of the conversion cycle. This overlap helps to provide a larger differential comparator input voltage for inputs near the latch threshold and avoids charge injection from the auto-zero preamplifier switches.

e) *Decoupling Capacitor Design*

The decoupling capacitor acts as a track and hold capacitor during phase 1. Its size is thus bounded by two requirements:

i) At least $t_1 = 600ps = 5\tau$ is required to correctly sample the input to within 0.5LSB of accuracy. This value comes from the following calculation:

$$settling\ error = V_{FS}e^{-\frac{t}{\tau}} = 5mV \Rightarrow t = 4.85\tau \Rightarrow \tau = 123ps$$

This considers the worst-case scenario when a voltage step equal to V_{FS} is applied to the decoupling capacitor for two consecutive sampling periods. The input impedance is equal to 50Ω, and we designed the input switches to have an on resistance of 50Ω. The resulting upper bound for the value of the capacitor is:

$$\tau = 17C(R_{on} + R_{sw}) \Rightarrow C < 70fF$$

ii. kT/C noise sampled by the input track and hold function. Limiting the RMS value of the noise to be <0.5LSB, the minimum value of the capacitor becomes:

$$5mV = \sqrt{\frac{kT}{C}} \Rightarrow C > 0.1656fF$$

Ideally, any value of C within the above range would guarantee a correct functionality. Moreover, the matching of the capacitors is not important; the functionality of the circuit ideally depends only on the voltage drop across the input capacitor plates, while the actual charge stored is not relevant.

However, there is a differential loss in the charge across the decoupling capacitors when these capacitors are connected to the resistive DAC. This loss is due to the fact that the inputs of the preamplifier need to be charged differentially to $V_{diff} = V_{ref} - V_{in}$ since they were pulled to virtual ground during the previous phase. Since the inputs of the preamplifiers are high-impedance nodes during phase 2, they can retrieve the required charge only from decoupling capacitors, thus introducing an error in the voltage stored on the capacitors. This error is proportional to the differential swing at the input of the preamplifiers, V_{diff}. However, preamplifier number P_n whose reference voltage is equal to the input voltage is not affected by this error. Moreover, we are interested in quantifying the error only at the input of the amplifier $P_{(n-1)}$ and $P_{(n+1)}$ since these are the amplifiers that set the important voltage levels at the interpolation string. The other amplifiers operate in the non-linear gain region and thus error at their input is not relevant. The charge that is required at the input of the preamplifiers is equal to $Q=C_{in}V_{diff}$, where $C_{in}=3fF$, considering the parasitics of the feedback switch with $V_{diff} = 40mV$. This charge is

removed from the decoupling capacitors and introduces error. We designed for less than half an LSB of error, so the lower limit for the value of the decoupling capacitor is:

$$\frac{Q}{C} < 0.5 LSB \Rightarrow C > 24fF$$

We chose a value of C=50fF between the maximum (settling) and minimum (charge loss) values. This resulted in a higher ENOB with respect to the minimum value of C=26fF.

f) Resistor String DAC Design

The resistive DAC provides the reference voltages for comparison with the input signal. Figure 3 shows the wire connections that were used to get the 17 required references. We managed to use only 8 resistors for the resistive DAC – opposite differential voltages can be obtained simply by swapping the wires connected to the same tap. Moreover, the taps move by one resistor per differential reference, instead of two resistors (of half the value) in typical implementations. Ideally, a change in the common mode of the differential references should not affect the functionality of the circuit since the preamplifiers are AC coupled from the DAC. Although capacitive coupling might still result in swing of the common mode at the input of the amplifiers, this effect is mild since the common mode swings only by 20mV at the taps; we did not notice any change in performance by adopting the proposed architecture. On the other hand, fewer resistors result in less parasitics and DAC larger resistors can be used. When the DAC is connected to the rest of the circuit at the beginning of phase 2, the charge injected by the switches and the difference in common mode with respect to the input signal cause a voltage drop on

the DAC string. We aimed at recovering completely from this drop within 500ps, to provide enough time for the preamplifier to set their outputs to the correct value. For 8 resistors, the total parasitic capacitance is equal to C_{par}=8(3.5fF)=28fF. We thus get:

$$500ps > 5\tau = 28fF * 0.25 * R \Rightarrow R < 14285\Omega$$

where R represents the total resistance of the DAC. We chose R=6600Ω, for a total dissipation in the DAC equal to I_{DAC}=167μA. Figure 3 shows the actual ratio between the resistors to get the desired voltage references at the input of the preamplifiers.

2) Circuit Schematics
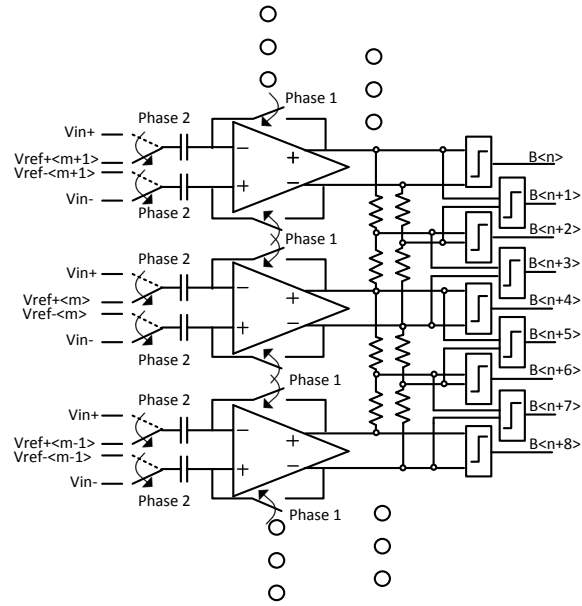
Figure 2: Auto-Zero and Interpolation Scheme
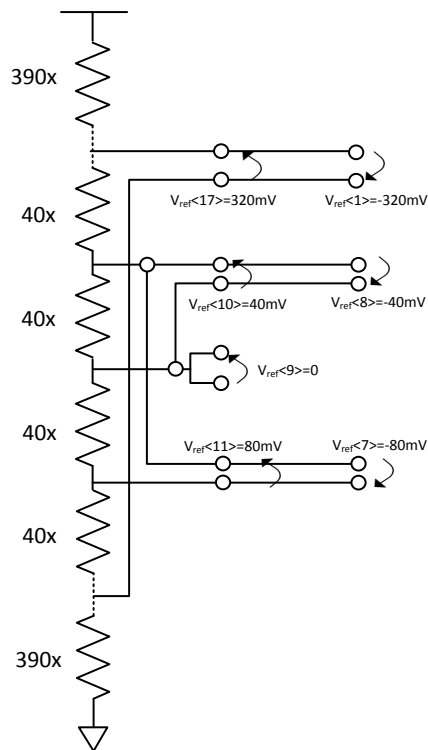
Figure 3: Resistive DAC Schematic
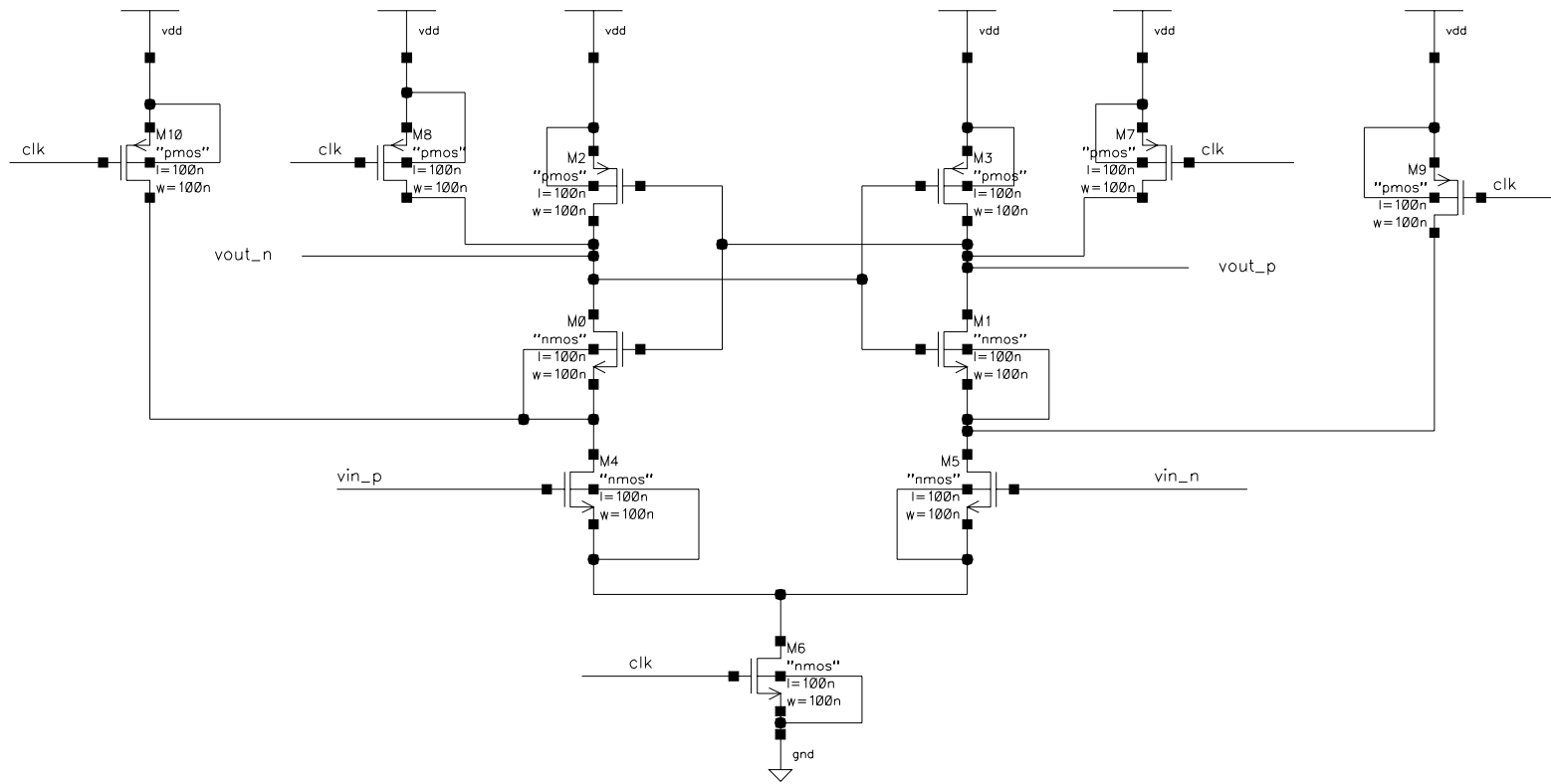
Figure 4: Comparator Schematic
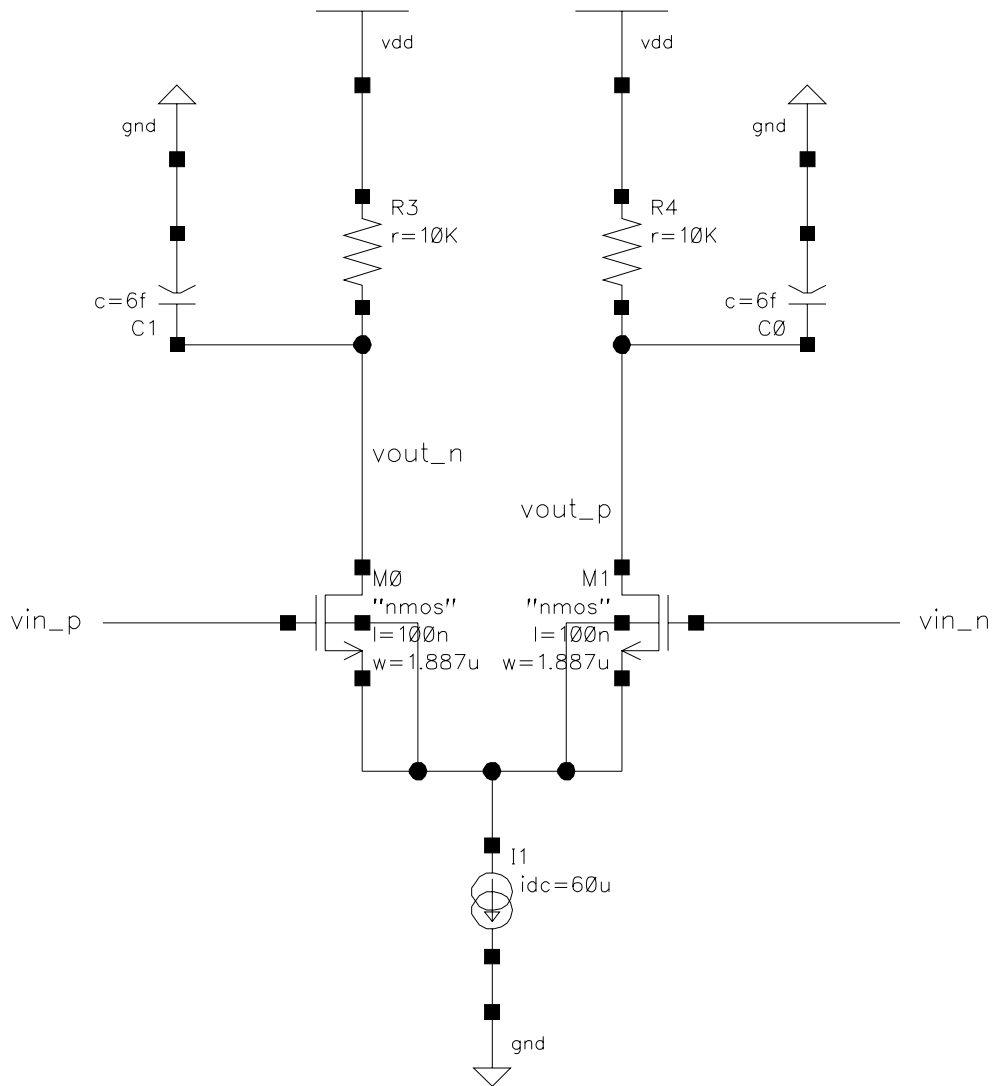
Figure 5: Preamplifier Schematic
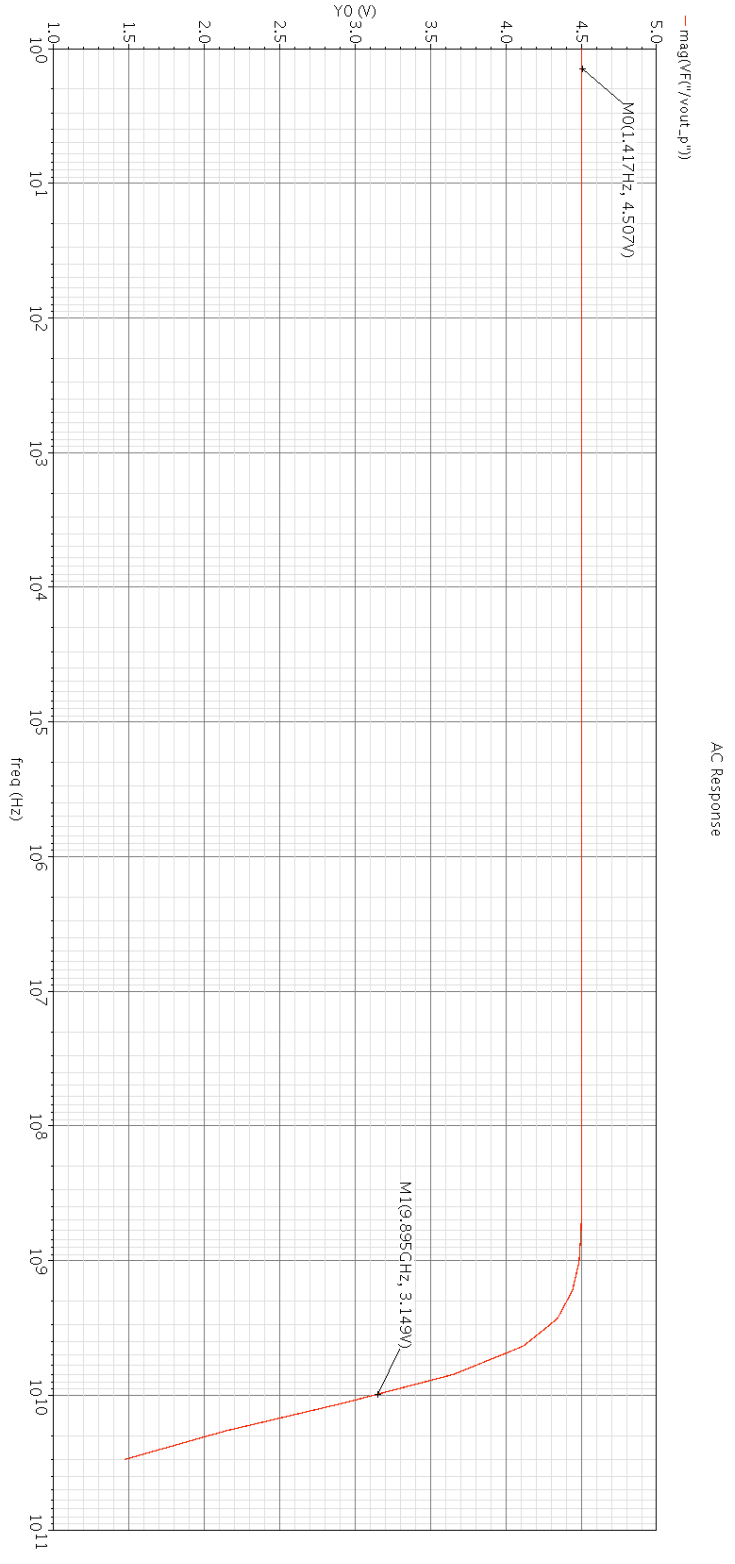
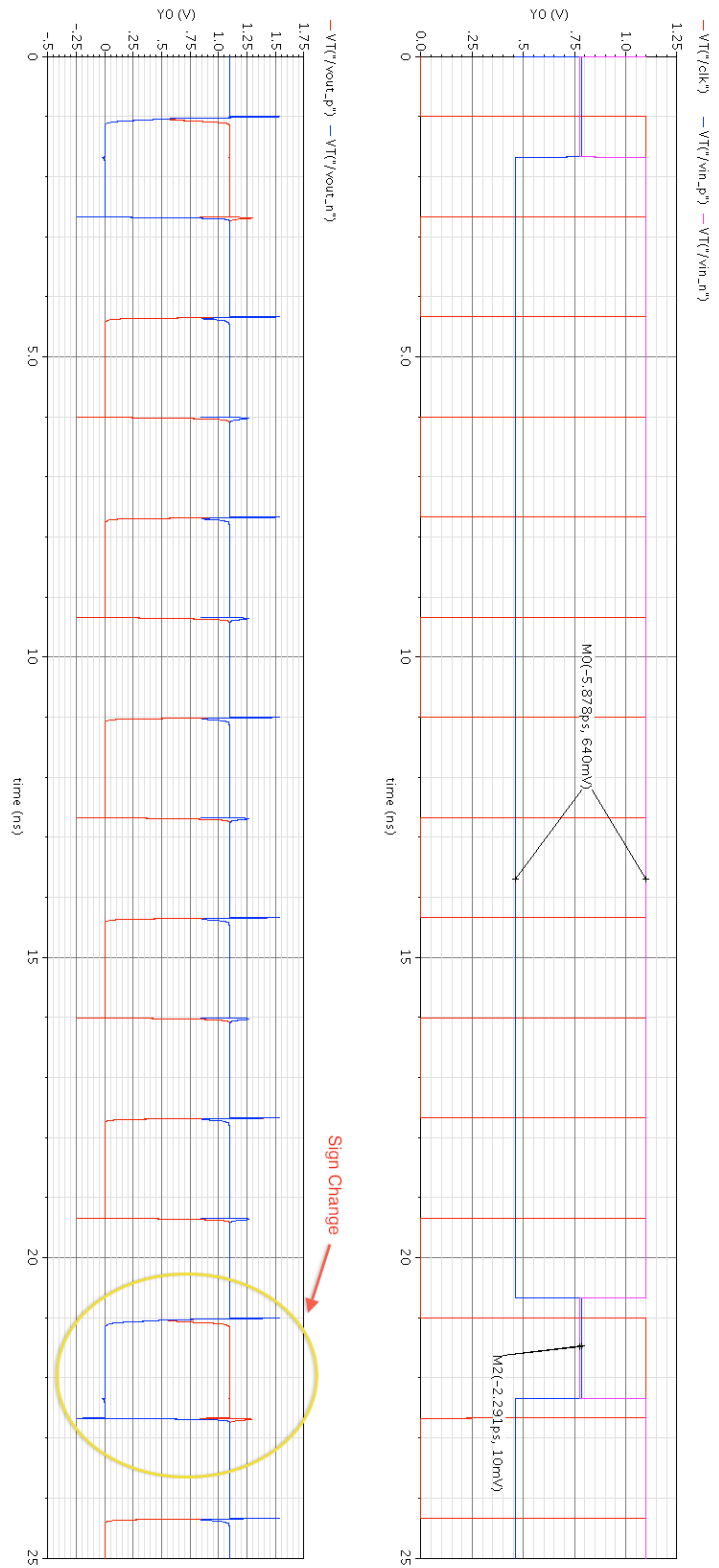## 3) Simulation Results

Figure 6: Preamplifier AC Response

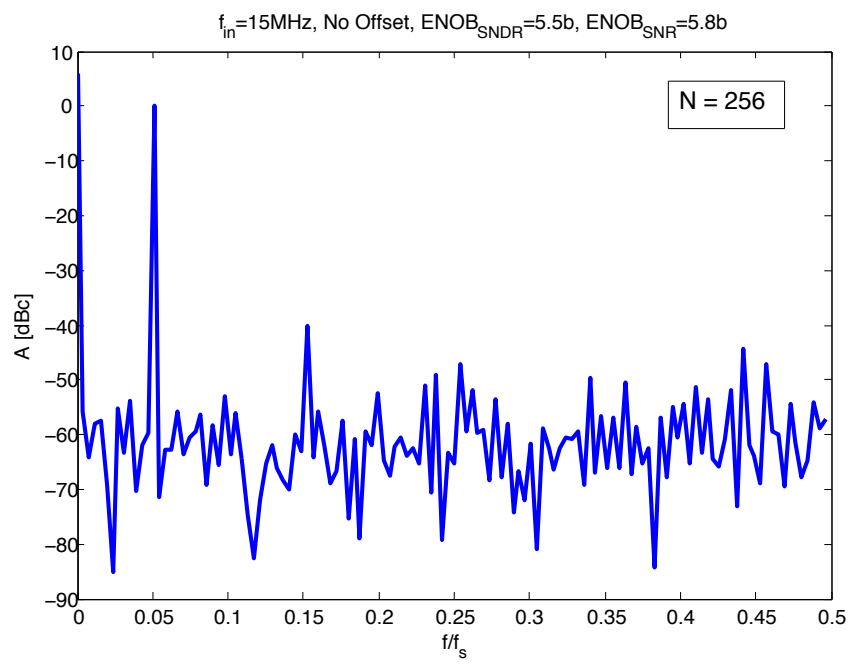**Figure 7: Comparator Overdrive Recovery Test Results**
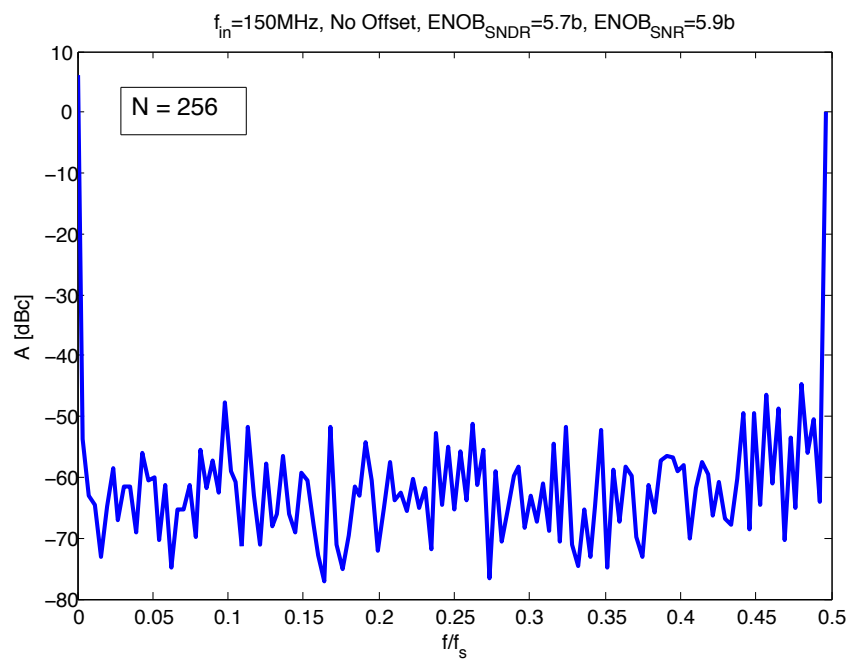
Figure 8: Low-Frequency, No Offset FFT Plot

Figure 9: High-Frequency, No Offset FFT Plot

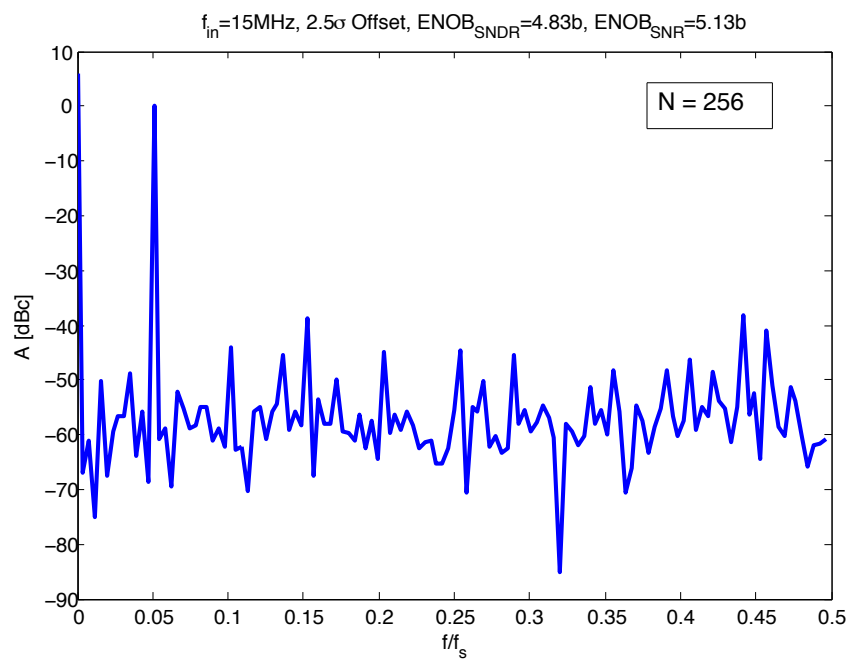**Figure 10: Low-Frequency, 2.5σ Offset FFT Plot**

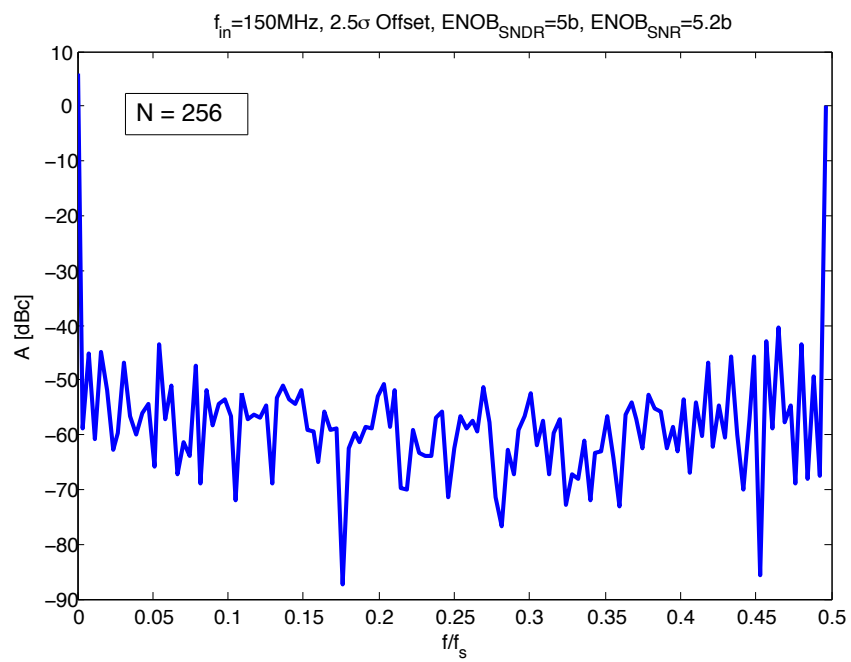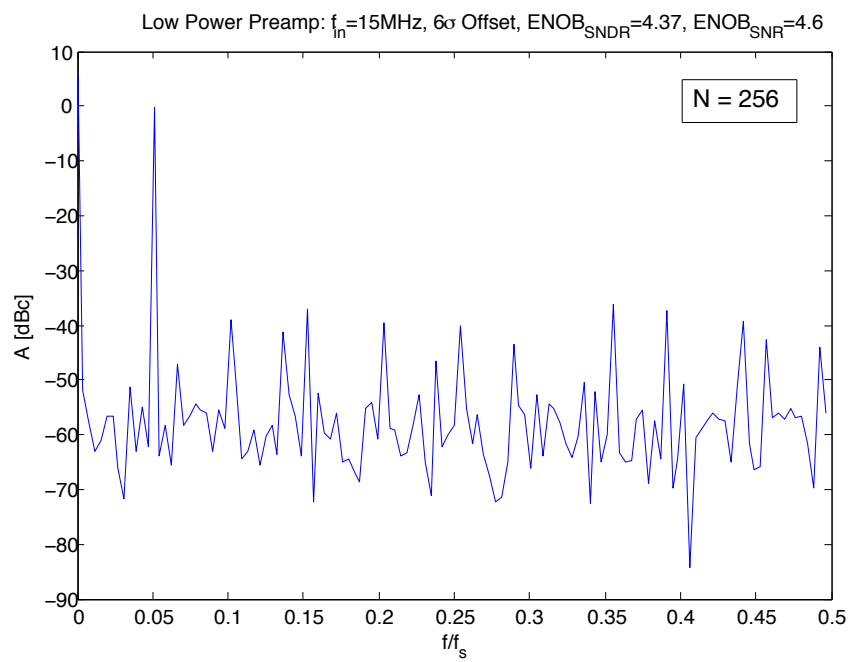**Figure 11: High-Frequency, 2.5σ Offset FFT Plot**

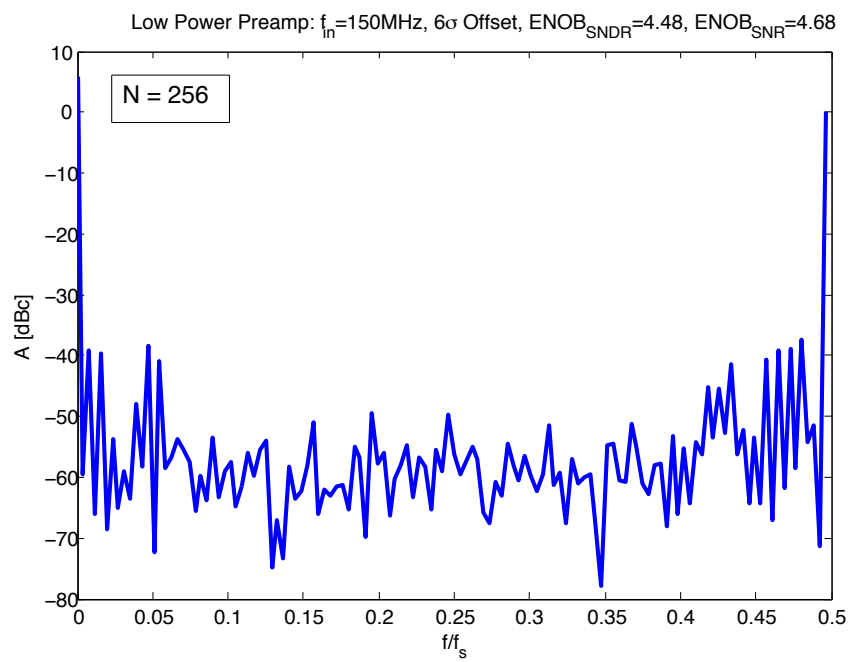**Figure 12: Low-Frequency, 6σ Offset FFT Plot**

Figure 13: High-Frequency, 6σ Offset FFT Plot

4) Final Results

a) *Power Consumption*

The final circuit power consumption is given below. The dynamic power consumption was found as follows: the transient current waveform is averaged over ten sampling cycles with f_{in} = 150MHz, V_{in} = V_{FS}.

$$P_{static} = 1.3mW$$
$$P_{dynamic} = 88.2\mu W$$
$$P_{total} = 1.3882mW$$

b) *Conclusions*

We believe that our design and architecture are very optimized for low power consumption. Our StrongARM comparator design uses minimum size transistors for low dynamic power, and dissipates no static bias current. This also results in a very fast settling response, therefore not increasing previous stage bandwidth requirements by much. We do not believe this power consumption can be reduced much further, without hysteresis concerns. However, smaller devices could potentially be used here, as we were unsure of our choice for W_{min} = 100nm.

The preamplifier circuit design merit is also critical for low power consumption – the amplifier dissipates static power and there are 17. We believe that our preamplifier gain is as low as it can be, with the largest output impedance tolerable given our already maximum sized interpolation resistors. This translates into the lowest g_m given the aforementioned specs. The preamplifier input pair is biased at the correct V* to minimize power without drastically decreasing bandwidth. This results in, what we believe to be, the lowest power preamp for our architecture.

We also have a higher-power preamp design that results in an overall increase in SNDR ENOB of 0.3-0.4 bits, however the total system burns 0.5mA more power. This is a tradeoff that was unclear based on the project description, so we decided to present our results for the lower power design. We did not include all data for both designs due to lengthy simulation times associated with generating the FFT plots.

5) Appendix

a) MATLAB FFT Code

```
6) % The first column stores the time steps, so it needs to be discarded.
7) latched = data(:,2:end);
8) N = size(latched);
9) N = N(1)
10)     pos = zeros(N,1);
11)
12)     % Compute how many latches toggle to 1 at each conversion cycle.
13)     for k = 1:N
14)         pos(k) = length(find(latched(k,:) > 0));
15)     end
16)
17)     % Plot the number of on latches, which should resemble a sinusoid
   shape.
18)     % This test helps in visualizing the linearity performance and
   the DR of
19)     % the ADC.
20)     figure(1)
21)     stem(pos)
22)
23)     % Compute the FFT, with the code taken from the course slides.
24)     figure(3)
25)     Afs = max(pos)/2;
26)     %s = 20*log10(abs(fft(pos)/N/Afs*2));
27)     s = abs(fft(pos)/N/Afs*2);
28)     s = s(1:N/2);
29)     f = (0:length(s)-1) / N;
30)     logS = 20*log10(s);
31)     plot(f,logS)
32)
33)
34)     [sorted,index] = sort(s,'descend');
35)
```

```matlab
36)      As = 20*log10(s(index(2)))
37)      % Get rid of the DC and signal component, to compute the SNDR
38)      s(index(1:2)) = 0;
39)      An = 10*log10(sum(s.^2))
40)      SNDR = As -An;
41)      ENOB_SNDR = (SNDR-1.76)/6.02
42)      % Get rid also of the 3rd and 5th harmonics to compute the SNR
43)      s(index(3:4)) = 0;
44)      An = 10*log10(sum(s.^2))
45)      SNR = As -An;
46)      ENOB_SNR = (SNR-1.76)/6.02
```