You are encouraged to work on the project in teams of two. In case you cannot find a partner, it is OK to work alone. It is not permitted to work in teams of three or more.

In Part I of this project, you are to design the first stage of a fully-differential 1.5-bit-per-stage pipelined ADC using generic circuit building blocks that we provide, including operational transconductance amplifier (OTA), switch, comparator, clock generator, single-to-differential converter, and differential-to-single converter. The detailed description for each of these building blocks is provided in the appendix. You can finish Part I without using NMOS and PMOS transistors. It is in Part II of this project that you will use real transistors to implement all these circuit blocks.

1. Design Target

The specifications for the first pipelined stage are shown in the following table.

<table>
<thead>
<tr>
<th>Specification</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power supply</td>
<td>1.8V</td>
</tr>
<tr>
<td>Differential input signal range</td>
<td>2V Vpp</td>
</tr>
<tr>
<td>Sampling rate</td>
<td>100 MS/s</td>
</tr>
<tr>
<td>Input referred noise</td>
<td>200 µV rms</td>
</tr>
<tr>
<td>THD for input at 31/64×100 MHz</td>
<td>−70 dB</td>
</tr>
<tr>
<td>Power consumption</td>
<td>As small as possible</td>
</tr>
</tbody>
</table>

This ADC does not use a dedicated front-end sample-and-hold circuit. In other words, the first pipelined ADC stage directly sees a continuous input.

The single-ended 1.5-bit stage architecture is shown below. It is the standard flip-over 1.5-bit structure. Please use the same capacitor sizes for the three capacitors. You will need to develop a fully differential version of the 1.5-bit stage, in which you will have 6 capacitors with equal size.
You are allowed to use ideal voltage and current sources in your design, but you need to make sure that the output of the voltage source is between 0V and 1.8V. Although the OTA model works for any input and output common mode voltages, you need to make sure that they are reasonable. This will facilitate your design in Part II. Remember that a real OTA built by transistors will not function correctly if the common mode voltages are too small or too large. Last but not least, please check that all of the circuit nodes are between 0V and 1.8V.

2. Design Flow

Below is the suggested design procedure. You do not have to follow it. Feel free to use your own design methodology as long as they are logical.

a) Decide OTA gain

You can infer how much OTA gain you need in order to achieve the target THD.

b) Decide capacitor size

Given input referred noise spec and the OTA noise model (see appendix), you can estimate the capacitor size by hand.

c) Decide OTA transconductance

From the THD spec, you can estimate how many loop constants you need. Then you can calculate by hand how much gm you need for the OTA based on the capacitor size from b).
d) Assemble the 1.5-b stage

You have already calculated the parameters in a), b) and c), now it is time to assemble the 1.5-bit stage. It may be useful to first assemble a single-ended version using the single-ended OTA (see appendix), as it is easy to debug. Once you finish the single-ended version, you can then easily extend it to a differential version.

Once the 1.5-bit stage is assembled, run a transient simulation with a full-swing input. Take a look at the output waveform to see whether the output has settled. If not, you probably want to increase the gm of the OTA to make sure that the OTA settles completely. After that, you should set the input to be at $31/64 \times 100 \text{ MHz}$ and perform a 64-point DFT on the 1.5-bit stage output to see whether the 1.5-bit stage meets the THD spec. Remember that to obtain the correct result, you need to sample $V_{\text{res}}$ and $d_1$ (using the sample function in Cadence calculator) at correct time instances and add them together. A common mistake is to grab samples before the circuit settles completely. We recommend you to export transient simulation results from Cadence to Matlab, and write a script to perform DFT and measure THD. If the THD spec is not met, figure out the reason and adjust the circuits appropriately.

Once the THD spec is satisfied, measure the total input referred noise of the 1.5-bit stage using cadence. You need to perform noise simulation using two methods: AC noise and transient noise. The results should be close to each other. If the noise spec is not met, identify the problem. It may be because your capacitor size calculated from b) is not large enough. Note that when you increase the capacitor size, you will need to increase OTA gm accordingly, as otherwise the circuit may not settle completely and the THD spec may not be met.

3. Recommendations

Whenever you run simulations, keep in mind that the specs we are demanding in this project are pushing the simulation engine close to their accuracy limits. Poor simulation settings may lead to erroneous results. You need to carefully adjust simulation settings, such as conservative or moderate, voltage/current tolerance, and time steps. You do not want to ask for an unnecessarily high precision, as it leads to long simulation time; you also do not want to make it too loose, as otherwise the results are not accurate enough. Seek a balance. When your result shows that you have met the specs, try re-running the simulation with a higher precision level to make sure that what you see is not a numerical artifact.

You may encounter problems during the design. In fact, the majority of your time is likely to be spent on debugging. This is a very important part of your learning process (well, you learn better from your mistakes). Indeed, what truly define an excellent designer are not just brilliant ideas,
but the ability to solve problems and make the ideas work in real environment. When you see something is wrong, do not panic, look at voltages and currents at various circuit nodes, and try to figure out why it is wrong by yourself. Do not seek help immediately, as otherwise, you will lose the precious debugging opportunity. You will learn as much from the debugging process as from the design process, if not more. If you encounter problems that you could not solve by yourself, please feel free to come to office hours and discuss with me and the TA. Please be well prepared and bring all your design documents. This is to save both our time and your time. The better prepared you are, the more efficient our discussion will be and the more help you will receive.

You are encouraged to discuss with other students from other groups about the general design issues, but your design and report must be unique. It is not permitted to copy design files from others. Let your honesty guide you.

Do not wait until the last minute to start. This project involves a great deal of just plain old labor. It takes significant amount of time to run all the necessary simulations. The servers may slow down and even crash toward the deadline as people are rushing. You want to be an early bird.

To be frank with you, this project is demanding (especially Part II), but we have done our best to ease your progress by developing generic circuit blocks. We are confident that you will LEARN A LOT from this project. Your time and efforts on this project will surely pave your way to the pinnacle of analog circuit design.

Happy Projecting!!

4. Project Report for Part I

The purpose of Part I is to warm you up for Part II. You do not need to submit a fully-blown report for Part I, but we will check on the date specified on page 1 to see whether you have finished it. Please prepare a 5-minute in-class presentation on Part I of your project. You need to provide at least the following figures in your presentation:

- Circuit diagram for the fully-differential 1.5-bit stage
- Settling transients (zoom in and annotate appropriately)
- 64-point DFT plot for input at $31/64 \times 100$ MHz
- Noise integrals for both sampling and charge transfer phases
- Transient noise at the output of the charge transfer phase
Appendix I: Circuit Building Block Description

In this appendix, we explain the definitions of the generic circuit building blocks that we have developed to facilitate your design. In order to fully understand their functions, we recommend you to look at their schematics and build test bench circuits to play with them. Do NOT start using these circuit blocks without understanding them.

1. OTA

We provide two types of OTAs that model telescopic operational amplifiers. One is a single-ended version (OTA_Single), and the other is a differential-ended version (OTA_Diff).

The single-ended version has five pins, given by:

<table>
<thead>
<tr>
<th>Name</th>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vin, Vinb</td>
<td>Input</td>
<td>Positive and negative inputs</td>
</tr>
<tr>
<td>Vcmo</td>
<td>Input</td>
<td>Output DC voltage</td>
</tr>
<tr>
<td>Vout</td>
<td>Output</td>
<td>Single-ended OTA output</td>
</tr>
<tr>
<td>Vdd</td>
<td>Supply</td>
<td>Power supply</td>
</tr>
</tbody>
</table>

When you use this single-ended OTA, you need to connect Vdd to a 1.8V power supply and Vcmo to a voltage source that sets the output DC (or common mode) voltage.

There are four parameters that you can set for the OTA:

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gain</td>
<td>OTA DC gain</td>
<td>N/A</td>
</tr>
<tr>
<td>fT</td>
<td>Transistor cut off frequency</td>
<td>20G</td>
</tr>
<tr>
<td>gm/id</td>
<td>Transistor power efficiency</td>
<td>10</td>
</tr>
<tr>
<td>gm</td>
<td>Transconductance</td>
<td>N/A</td>
</tr>
</tbody>
</table>

In this project, you are not allowed to change fT and gm/id. These parameters are set by the technology we use, which is 0.18um CMOS. You can change the gain and gm.

The input referred noise of this single-ended OTA is given by:

\[
\frac{v_n^2}{\Delta f} = 8kT/gm
\]

The differential-ended OTA is almost identical to its single-ended counterpart. One difference is that it produces differential outputs, V_{out} and V_{outb}. It has an internal common-mode-feedback
(CMFB) loop that ensures that the output common mode voltage, \((V_{out}+V_{outb})/2\), is equal to \(V_{cmo}\).

2. Comparator

The comparator (Comp) is a model for the strong arm latch. It compares its input vip with vim, and generates the digital output d and db. \(d = 1.8V\) and \(db = 0V\) for \(vip > vim\). \(d = 0V\) and \(db = 1.8V\) for \(vip < vim\).

The operation of the comparator is controlled by the clock CLK. When \(CLK = 0V\), the latch operates in the reset phase. When \(CLK = 1.8V\), the latch operates in the regeneration phase.

3. Clock generator

The clock generator (CLK_Gen) produces 8 clock phases. \(f1b\), \(f1eb\), \(f2b\), and \(f2eb\) are complementary clock phases for \(f1\), \(f1e\), \(f2\), and \(f2e\), respectively. This clock generator has 4 parameters:

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>period</td>
<td>Clock period</td>
<td>1/100M</td>
</tr>
<tr>
<td>t_early</td>
<td>The time difference between f1 and f1e</td>
<td>200 ps</td>
</tr>
<tr>
<td>t_edge</td>
<td>The rise and fall time</td>
<td>60 ps</td>
</tr>
<tr>
<td>t_nonoverlap</td>
<td>The non-overlapping time between f1 and f2.</td>
<td>200 ps</td>
</tr>
</tbody>
</table>

You are not allowed to change period and t_edge, but you can adjust t_early and t_nonoverlap. In case you need other timing signals that are not provided by this clock generator, feel free to add them, but you will need to set the period to be 1/100M and the edge time to be 60 ps.

4. Switch

The switch (Switch) has a resistance of 100\(\Omega\). It is closed when its control voltage is greater than 0.9V and is open otherwise.

5. Single-to-differential and different-to-single converters

The single-to-differential converter (S2D) is helpful when you want to generate differential input signals. The relationship between its inputs and outputs is given by:

\[
V_{out} = V_{cni} + V_{in}/2 \\
V_{outb} = V_{cni} - V_{in}/2
\]
The differential-to-single converter (D2S) helps you analyze the outputs of a differential circuit. It receives a differential signal, and calculates its common-mode voltage and differential mode voltage. The mathematical relationship is given by:

\[ V_{cm} = \frac{(V_{ip} + V_{im})}{2} \]
\[ V_{dm} = V_{ip} - V_{im} \]

Appendix II: EE382V Library Installation

Please follow the steps to install the EE382V library that contains the generic circuit blocks:

1) Create a library named EE382V_lib in Cadence
2) Download EE382V_lib.zip from blackboard; if you are using cadence 2009, please download EE 382V_lib_OA_zip from blackboard.
3) Unzip the file and copy its content to the EE382V_lib directory that you just created in your LRC account in step 1). The system may ask whether you want to overwrite the files that are originally there. Click ‘Yes’.

Now you should be able to see the generic circuit building blocks in EE382V_lib from Cadence. If you have any question, please feel free to let us know.
EE 382V Data Converter Project: Part II
Instructor: Nan Sun
Due: Wednesday, 05/04/2012
Copyright © 2012 by Nan Sun

With the completion of Part I, you should have a working 1.5-bit stage using the generic circuit building blocks. Now, in Part II of this project, your job is to implement the OTA, comparators, and switches by real transistors, resistors, and capacitors. You do not need to implement the clock generator, single-to-differential converter, and the differential-to-single converter; you can still use generic models for them.

1. Design Specifications
The specs are given in the following table.

<table>
<thead>
<tr>
<th>Specification</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power supply</td>
<td>1.8V</td>
</tr>
<tr>
<td>Differential input signal range</td>
<td>2V Vpp</td>
</tr>
<tr>
<td>Sampling rate</td>
<td>100 MS/s</td>
</tr>
<tr>
<td>Input referred noise</td>
<td>200 uV rms</td>
</tr>
<tr>
<td>THD for input at 31/64×100 MHz</td>
<td>−70 dB</td>
</tr>
<tr>
<td>Power consumption</td>
<td>As small as possible</td>
</tr>
<tr>
<td>ERBW</td>
<td>As large as possible</td>
</tr>
</tbody>
</table>

They are the same as in Part I. The difference is that now you need to design OTA, comparators, and switches to meet these specs. Please minimize the total power consumption of the OTA and the comparator as much as possible. You also want to enlarge the ERBW as much as you can. The power consumption and ERBW are important factors that we will consider when giving out grades and the design award.

2. Circuit Building Blocks

a) OTA

The OTA is probably the most challenging circuit building block in this project. Please formulate by yourself a list of specs for the OTA based on the 1.5-bit stage specs. The OTA specs should include DC gain, unity-gain frequency, output swing, phase margin, and input-referred noise. Choose an appropriate OTA architecture that can satisfy these requirements and make it work.

Since the OTA is fully differential, you will need to design a common-mode feedback (CMFB) circuit that stabilizes the OTA output common mode voltage. You are allowed to use ideal
controlled voltage/current sources from analogLib to build them. Yet, if you can implement the CMFB circuit using transistors, capacitors, and resistors, you will receive big bonus points.

For the OTA biasing circuit, you can use ideal voltage and current sources. Nevertheless, for the OTA core, you are not allowed to use any ideal voltage and current sources; you can only use transistors, resistors, and capacitors.

b) Comparator

Choose a comparator topology that dissipates low power. When measuring comparator power, make sure that you average its power over the entire clock period. In case you use a dynamic comparator, even though it does not dissipate static power, it does not mean that the comparator does not dissipate dynamic power.

c) Switch

There are many switches in the pipelined stage, but not all of them are critical in determining the linearity. Identify those are important and spend your time on them. For others, standard CMOS switches are sufficient (you can even use only one NMOS transistor or one PMOS transistor, depending on the voltage that the switch passes).

To increase ERBW, you probably want to consider using bootstrap circuit for the front-end switches that connect to the input. When you use them, remember that it takes a few clock cycles for the bootstrap circuit to initialize its internal capacitors and reach steady state. Make sure that you skip the first few samples for your DFT analysis.

Note that the generic switch that we provide to you does not take into account of charge injection and clock feedthrough. Thus, when you replace it with real transistors, you will observe common-mode voltage variations. You may need to adjust your common-mode voltages to make sure that the OTA input and output common-mode voltages are correct.

3. Recommendations

We highly recommend you to use your 1.5-bit stage from Part I as a test bench for the OTA, comparator, and switch that you design in Part II. When testing your OTA, keep the comparators and switches to be the generic ones, so that if any problem occurs, you know that is due to the OTA. Test comparator and switch following the same way. Do not test them altogether, as otherwise it will be very hard to debug. Another reason for testing them one by one is that it saves simulation time.
4. Project Report

Each team will submit one project report describing your design. Please turn in both an electronic copy and a hardcopy.

Please include a table that summarizes the OTA performance, including DC gain, unity-gain frequency, output swing, phase margin, and input-referred noise, and power consumption. You also need to include at least the following figures:

- Circuit diagrams
- Settling transients (zoom in and annotate appropriately)
- Circuit diagrams for OTA, comparator, and switches
- 64-point DFT plot for input at 31/64×100 MHz
- Noise integrals for both sampling and charge transfer phases
- Transient noise at the output of the charge transfer phase
- SFDR and SNDR versus input frequency

The target audience for your report consists of circuit designers that are generally familiar with the course material, but are assumed to know nothing about your particular circuits and design task. Your main objective is to convince readers that you have done a great job and you fully understand the involved design tradeoffs.

Your report must be in the format of an IEEE journal paper. You can find the IEEE template from:

http://www.ieee.org/publications_standards/publications/authors/authors_journals.html

The basic guidelines are as the following:

- Abstract: Briefly explain what your paper is about and what you have accomplished.
- Introduction: Explain from a high level perspective what you have done. Define the objectives, discuss the trade-offs, and mention the key idea. Do not go into circuit details.
- Core Sections: Present the details of your design in a clear and easy way. A good paper makes it crystal clear to readers how you achieved your specs.
- Conclusion: Conclude by summarizing what you said.
- Figures: Please make sure that the figures are clearly labeled, and the fonts are large enough. If you use Microsoft Word and you find it hard to embed figures in the text, please feel free to put all figures at the end of your report on separated pages.