Design of a CMOS OPAMP

Prof. Michiel Steyaert
Filip Tavernier
Koen Cornelissens
Wim Vereecken

Introduction

This text will guide you through making a design in a CMOS technology. The goal is to design an OPAMP that can be used as an active low-pass filter. As will be seen later on, the OPAMP is composed of an OTA and a voltage buffer.
First starting from the specifications for the filter, the specifications for the OPAMP will be derived. Then the voltage buffer is designed, followed by the design of an appropriate OTA. Finally the designed OPAMP will be verified by simulating the entire filter.

Initialization

We will use the HSpice simulator for this design, working in UNIX and AvanWaves as waveform viewer. The online help for HSpice can be started with the command:
`/imec/software/synopsys/syn_2004.12-SP1/sold &`.

1. Make a subdirectory `opamp_design` on your account.
2. First the transistor model files must be copied and unzipped in this directory. They can be found at: `/imec/other/umc18/UMC/UMC18_documentation/UMC18_LOGIC_GII/G-05-LOGIC18-1.8V-GENERICII-SPICE.zip`.
3. Next the template files must be downloaded and unzipped. These can be found at `http://homes.esat.kuleuven.be/~ftaverni/`
4. Since HSpice works under UNIX, a UNIX session must be started.
5. Then the necessary script has to be sourced, type:
   `source /imec/software/meta/W-2005.03/hspice/bin/cshrc.meta`. This must be done each time a new session is started.
6. To execute a simulation, type `hspice netlist.sp > netlist.lis`
   The `netlist.lis` file contains eventual errors and warnings, the DC-operation point and the small-signal parameters of all transistors.
7. When doing an AC analysis, a file `netlist.ac0` is created.When doing a transient analysis, a file `netlist.tr0` is created. Both can be viewed using `awaves`.

Design Specifications

- Low frequency gain of the active low-pass filter = $20\ \mathrm{dB}$
- Error on the low frequency gain = $1\ \%$
- Bandwidth of the active low-pass filter = $1\ \mathrm{MHz}$
- f_{nd} of the active low-pass filter = $100\ \mathrm{MHz}$
- Resistor R_2 of the low-pass filter = $1\ \mathrm{k\Omega}$
- OTA must be capable of operation in unity feedback configuration
- OTA slewrate = $100\ \mathrm{V}/\mu\mathrm{s}$
- Minimal power consumption

1 Design of the first order active low-pass filter

1.1 Ideal OPAMP

Figure 1 shows the schematic of an active low-pass filter. The circuit uses feedback to obtain a low-pass filter characteristic. Starting from figure 2, the loop gain is given by

$$L = GH \tag{1}$$

and the closed loop transfer function is equal to

$$T = \frac{G}{1+GH} \approx \frac{1}{H} \tag{2}$$

- Convert the schematic of the low-pass filter to a block diagram that clearly shows the feedback nature of this system.

We will first assume that the OPAMP is ideal, i.e. it has an infinite gain and GBW.

- Find the transfer characteristic (v_{OUT}/v_{IN}) of the filter under the assumption that G is very large. Draw a Bode-plot of this transfer function.
- Find the relation between the component values (R_1, R_2, C_1) and the specifications of the filter (DC-gain and bandwidth).

1.2 Influence of OPAMP gain and GBW

In reality, the OPAMP will not be ideal, but will have a finite gain and GBW. If the OPAMP is approximated as a first-order low-pass system with finite gain and GBW, the transfer function of the OPAMP can be written as

$$G = \frac{A_{DC}}{1+j\frac{\omega}{\omega_d}} \tag{3}$$

with A_{DC} the low-frequency gain of the OPAMP and ω_d the bandwidth.

- Find the exact closed loop transfer function of the filter.
- Find the required minimum DC-gain of the OPAMP A_{DC} in order to have a DC-gain error below the specified value. The relative DC-gain error ($\varepsilon_{T(0)}$)is given by:

$$\varepsilon_{T(0)}(\%) = 100\frac{1+R_1/R_2}{1+R_1/R_2+A_{DC}} \tag{4}$$

- Calculate the influence of the finite GBW of the OPAMP on the transfer function of the active low-pass filter. See Appendix A for more information.

1.3 Influence of finite OTA slewrate

When the input voltage changes too rapidly and with high amplitudes, the possibility exists that the OTA enters slewing regime. This means that the OTA output is delivering the maximal possible current, to change the output voltage as fast as possible.

- For the OTA of figure 4, calculate the theoretical slewrate.
- If we look at slewrate as function of GBW, which design parameters become fixed?
- Evaluate the previous condition for the given specifications.

2 Design of the OPAMP

An OPAMP, like the one used in this active low-pass filter, has a low output impedance and a large voltage gain. In general, an OPAMP consists of an OTA, followed by a voltage buffer. The next step of this seminar is to design the OTA and voltage buffer in a $0.18\ \mu\mathrm{m}$ CMOS technology that operates at $1.8\ \mathrm{V}$.

2.1 Design of a transistor

In deepsubmicron CMOS technologies, it becomes difficult to use easy mathematical relations for I_{DS} and g_m as function of W, L and $V_{GS} - V_T$. To obtain the small signal parameters of a transistor for a given W/L and biasing point, it is best to simulate it. You can use the file `tor_test.sp` for this purpose. The resulting small signal parameters and transistor current can then be found in the output file.

2.2 Design of the voltage buffer

For the voltage buffer, we will make use of a common source amplifier, as shown in figure 3. The DC input voltage of this voltage buffer is equal to the DC output voltage of the OTA. To maximize the voltage swing of the OTA, the DC output voltage of the OTA is chosen equal to $V_{DD}/2$, i.e. $0.9\ \mathrm{V}$ for this $0.18\ \mu\mathrm{m}$ CMOS technology. The DC output voltage of the buffer is chosen to be $0.7\ \mathrm{V}$. The reason for this choice will be clarified in section 2.3.

- Obtain an expression for the voltage gain, output resistance and input capacitance of the buffer.
- Verify the design by simulating the buffer in HSpice. The netlist is given in the file `buffer.sp` and the corresponding testbench is `buffer_test.sp`.
  Use `hspice buffer_test.sp > buffer_test.lis` in order to simulate this buffer with Hspice.
- From the file `buffer_test.lis` the numeric values of the operating point can be found.
- Use `awaves` to view the AC simulation results.
- Compare the simulated voltage gain and bandwidth of the buffer, with the values obtained from the calculated expressions. Do the results match? If not, why ...?
- Obtain the equivalent input capacitance of this buffer from the `buffer_test.lis` file.

2.3 Design of the OTA

The schematic of the OTA is shown in figure 4. This is a so-called current mirror OTA with single-ended output.

- What is the inverting input and non-inverting input ?

As said before, the DC output voltage of the OTA is chosen equal to $V_{DD}/2$, i.e. $0.9\ \mathrm{V}$, for a maximum output swing.

- What is the maximum and minimum output voltage of the OTA? What is the limitation?

- At first sight, it might seem logical to choose the DC voltage at the gates of M1 also equal to 0.9 V. This however will cause problems for transistor . . . ?

Clearly, the DC input voltage of the OTA needs to be lower than 0.9 V, and is therefore chosen equal to 0.7 V. This explains why the DC output voltage of the buffer should also be 0.7 V, after all, the OPAMP will be used in a feedback configuration. The voltage buffer will take care of the DC voltage drop from 0.9 V (output of the OTA) to 0.7 V (output of the buffer).
The next step is to design this OTA (i.e. make a selection of the gate-width, gate-length and drain-current of each transistor) so it fulfills the required specifications. First of all, analyze the small signal behavior of the OTA.

- What is the load of this OTA?
- On which nodes are the dominant and non-dominant pole(s) located; find the impedance of each node of this circuit.
- Find expressions for the DC-gain, bandwidth, GBW and non-dominant pole as a function of the transistor small-signal parameters. Remember things like *Miller Effect* and *pole-zero doublets*. Try to find the influence of each node of the OTA on the transfer function.
- Use the values of the previous section for DC-gain, bandwidth, GBW, non-dominant pole and slewrate in order to get an idea of the required small-signal transistor parameters (g_m, r_o etc.).

Adjust the netlist `ota.sp` and `ota_test.sp` to your specifications. A possible design-flow could look as follows:

1. Make a reasonable selection for the currents in each branch of the OTA. You can use the small-signal parameters as a guideline for this initial guess.
2. From this rough selection of currents, calculate the required gate-width and gate-length of each transistor (eg. with the aid of `tor_test.sp`) and use these values in the netlist `ota.sp`.
3. Simulate the operating point of the OTA and ensure that all transistors are in saturation.
4. Simulate the AC response of the OTA and verify the OTA performance (gain, GBW, . . .) with AvanWaves.
5. Verify the results from Hspice and AvanWaves with your analytical formulas. If there is a discrepancy, try to find an explanation for it.
6. You can now change the transistor parameters (gate-width, gate-length, drain-current, gate-source voltage, . . .) to improve the performance of the OTA, but of course, maintain the symmetry in the circuit!

Before you change a transistor parameter, reason about it first. . . ! Don't just change something unless you have at least a (small) idea why it should be beneficial for your design. Try to have a answer to the question ***"why do you change something and what do you expect from it ?"*** And in case you wonder . . . this *is* analog design . . .

- Does the calculated GBW match the simulated GBW? If not . . . why?
- What is the maximum gain and/or GBW that you can achieve with this OTA? Are all specifications met? If not, what is (are) the limitation(s)?
- Summarize the trade-offs that occur in this OTA.

2.4 Cascode Transistors

If the specifications are still not met, add an nMOS *cascode transistor* at the output, see figure 5.

- Why do we add an nMOS cascode and not a pMOS cascode?
- What is the impedance at nodes n3c and n3d?
- How would you bias node ng3?
- Adjust the netlist and add the cascode transistor.
- Which parameter(s) is (are) improved by the cascode transistors? What is the limitation of this technique? Which specifications are met and which ones are not met?
- Does the calculated GBW match the simulated GBW? If not . . . why?
- Does this technique allow you to reduce the current consumption?

Of course, a pMOST cascode could be added as well. What will be the advantage of this additional transistor, besides a higher DC-gain? If you have some time left, you can design the OTA with both the nMOS and pMOS cascodes.

3 Simulation of the low-pass filter

Now you can do a simulation of the low-pass filter, with the designed OPAMP at transistor level.

- Create a new subcircuit for the entire OPAMP.
- What will be the effect of the negative gain of the voltage buffer?
- What will change with the circuit of figure 1 if you use an asymmetric supply between 0 and V_{DD}?
- Verify the simulated values with the original specifications.

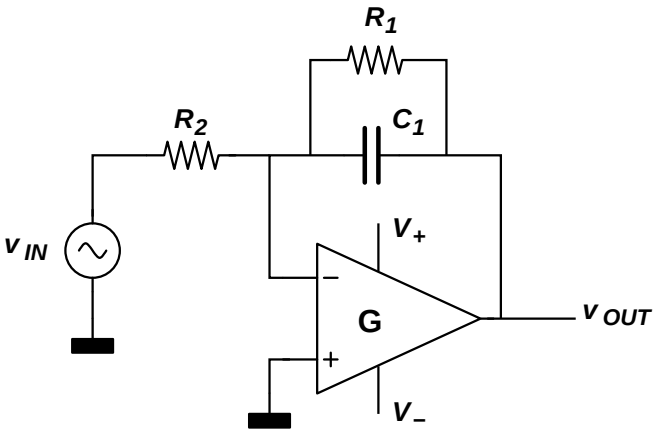

Figure 1: Low-pass filter.

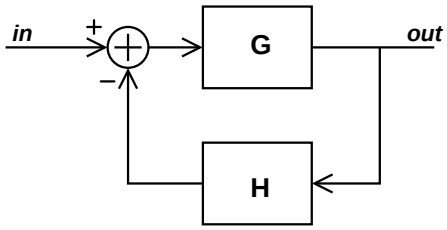

Figure 2: Typical feedback system.

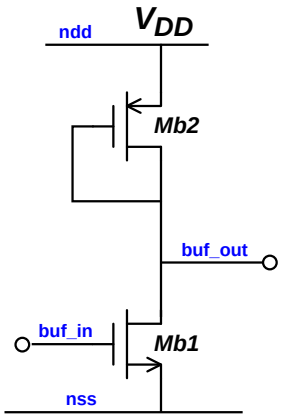

Figure 3: common-source output buffer.

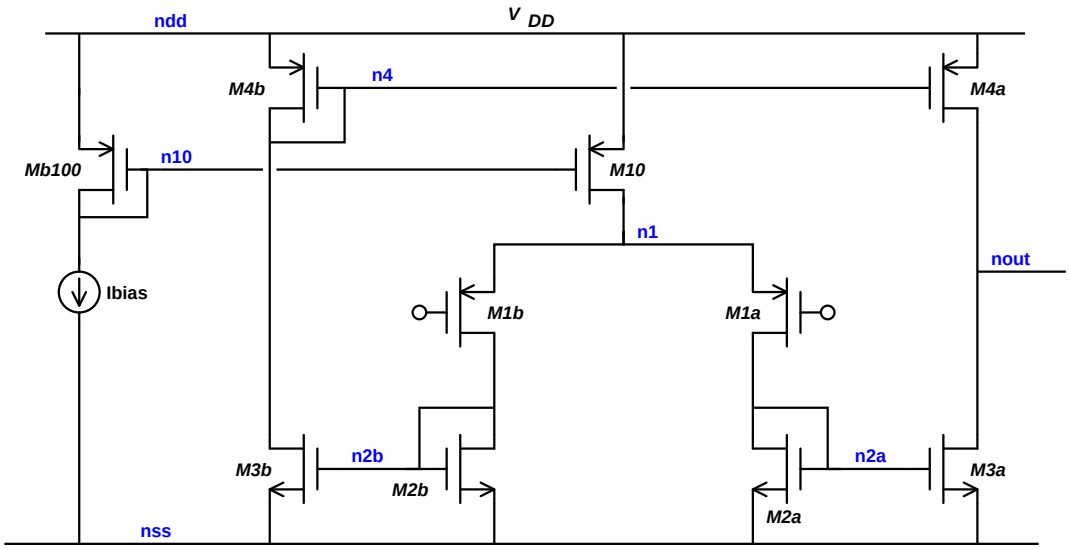

Figure 4: current mirror OTA with single-ended output.

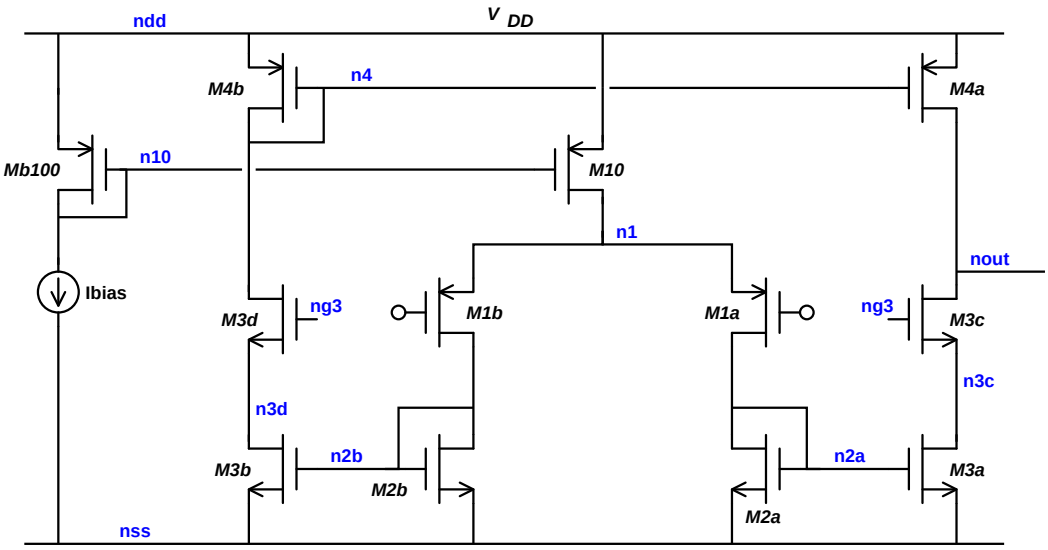

Figure 5: current mirror OTA with single-ended output and cascode output stage.

Appendix A: Second order systems

A second order system with two separate poles (τ_d and τ_{nd}) can be written as

$$T(j\omega) = \frac{1}{(1 + s\tau_d)(1 + s\tau_{nd})} \tag{5}$$

with $\omega_d = 2\pi f_d = 1/\tau_d$ and $\omega_{nd} = 2\pi f_{nd} = 1/\tau_{nd}$. The previous equation can be rewritten as

$$T(j\omega) = \frac{1}{(1 + s(\tau_d + \tau_{nd}) + s^2(\tau_d\tau_{nd}))} \approx \frac{1}{(1 + s(\tau_d) + s^2(\tau_d\tau_{nd}))} \tag{6}$$

under the assumption that $\tau_d >> \tau_{nd}$
In other words, if you have an expression

$$T(j\omega) = \frac{1}{(1 + a \cdot s + b \cdot s^2)} \tag{7}$$

then

$$\tau_d \approx a \tag{8}$$

and

$$\tau_{nd} \approx \frac{b}{a} \tag{9}$$