Electronics Design III
Laboratory work
2010 - 2011

Case: 2nd order SC-low pass filter designing and layout

Introduction

The aim of this work is to realize a 2nd-order SC low pass filter from given specifications by using Fleischer-Laker biquad -topology. The work includes all phases starting from realization of filter specifications ending at drawing a physical layout for the some part of the whole filter. The task is started from the specification of the filter and continues to designing the different components of the filter. The parts of the filter and also the whole filter will be checked by using different simulations. At the end the layout of the almost whole filter will be designed. The schematic and layout will be designed by using the hierarchical manner meaning that individual block will be simulated and drawn separately and after that those can be used to realize the whole filter. This work does not consist of placing the I/O pads and post layout simulations, which should be done before the the whole IC could be sent for fabrication.

The first part consists of sizing the capacitnaces of the filter, which will be done by using Matlab. After that filter will vbe designed by using IC-program called Cadence. The minimum gate length of the process, HitKit, used in this work is 90 nm.

Phases of the work:

1. Size the capacitance values of the Fleischer-Laker biquad SC-filter and print the amplitude response of the filter by using Matlab.
2. Generate the specification for an operational amplifier and switces used in the filter and simulate and print the results.
3. Simulate the transient response of the whole filter by using the frequency of your corner frequency and print the results.
4. Draw the layout of almost the whole filter so that components are well matched. Confirm the operation of your circuit by checking the design rules (DRC = Design Rule Check) and the equivalency between the layout and schematic by using LVS = Layout Versus Schematic. Print the layout and the results of checks.

Cadence

Cadence is used in linux enviroment, so it can be used directly in classrooms TS136-138 or with an emulator in in PC-classrooms TS134-135 or home.

PC: If you like to use Cadence for examble home, you can download an emulator, XLiveCD, from a webpage.

1) Download and save image-format file from:
   http://xlivecd.indiana.edu/
2) Open it and burn it in a CD.
3) Put CD into your PC, in which case the emulator starts up.
4) Click "Next" and choose mouse (2-buttoned) in the frame.
5) Program will open a command editor, make setups and open an unix window.
6) Login unix workstation for examble => ssh username@stekt1.oulu.fi
7) If a terminal asks if you want to continue : type yes
8) Write your password.
9) Now you can continue to use Cadence or do setup for Cadence as explained below.

Setup Cadence

Before you can use Cadence some setup has to be done: Generate a design directory under your home directory and named it for examble ed3. Write a command using terminal:

```
cd; mkdir ed3
```

Copy all design files and setup files aka file =>

```
sc_filter_linux.tar.gz
```

from the course webpage http://www.electronics.oulu.fi/Opetus/APIII/AP3_03/AP3.html, to your design directory. Decompress that file by using command

```
gzip -d sc_filter_linux.tar.gz; tar xvf sc_filter_linux.tar; rm sc_filter_linux.tar
```

Copy a printing file for Cadence .cdsplotinit, if you don't have that file by opening it and saving it by name (.cdsplotinit). Check that file name has .before name.

Write a command to login to linux

```
elcn1
```

Go to your home directory ed3

```
cd ed3
```

Execute the setup and start Cadence by using command

```
source csh_init ;icfb&
```

1. Designing Fleischer-Laker biquad SC-filter

Fleischer-Laker biquad filter has been presented in a book CMOS Analog Circuit Design (Allen&Holberg), which can be found in library Tellus. The short overview of the designing the filter has been presented here.

The task is to design 2nd order low pass SC-filter, which will meet the specifications. The specification of the filter will be sent to each group by me and the specifications consist of the corner frequency f_0, the quality factor Q and the amplification A of the filter. In that case the transfer function of the filter can be presented in s-domain as

$$H(s) = A\frac{s^2 + cs + d}{s^2 + as + b} \Rightarrow A\frac{\omega_0^2}{s^2 + s\cdot\frac{\omega_0}{Q} + \omega_0^2} \tag{1}$$

The clock frequency of the SC-filter f_{clk} is 1 MHz.

The corner frequency has to be prewarped, so that the corner frequency of the discrete transfer function is the nearest thing to the corner frequency given by specification. Equation below can be used to prewarp the frequency.

$$\omega = \frac{2}{\tau}\tan\left(\frac{\hat{\omega}\tau}{2}\right), \tag{2}$$

wher $\hat{\omega}$ is the corner frequency given by specification f_0 and $\tau = 1/f_{clk}$. The prewarped corner frequency is inserted to equation (1) and after that the s-domain transfer function is converted to z-domain by using a bilinear transformation as

$$s \rightarrow \frac{2}{T}\frac{1 - z^{-1}}{1 + z^{-1}} \tag{3}$$

Z-domain transfer function will be :

$$H(z) = \frac{\gamma + \varepsilon\cdot z^{-1} + \delta\cdot z^{-2}}{1 + \alpha\cdot z^{-1} + \beta\cdot z^{-2}} \tag{4}$$

The closed loop transfer functions of Fleisher-Laker biquad at the outputs of the first and the second operational amplifier are as presented below, respectively

$$T(z) = -\frac{DI + (AG - DI - DJ)z^{-1} + (DJ - AH)z^{-2}}{D(F + B) + (AC + AE - DF - 2DB)z^{-1} + (DB - AE)z^{-2}} \tag{5}$$

$$T'(z) = -\frac{(IC + IE - GF - GB) + (FH + BH + BG - JC - JE - IE)z^{-1} + (EJ - BH)z^{-2}}{D(F + B) + (AC + AE - DF - 2DB)z^{-1} + (DB - AE)z^{-2}} \tag{6}$$

Choose the type of the filter related to quality factor:

E-type => high-Q or F-type => low-Q filter.

Choosing A = B = D = 1 T(z) and T'(z) can be rewritten. Additionally, because the filter is low pass filter H = 0, I = J = γ(F + 1) and G = ε(F + 1) + 2γ (F + 1).

H(z) = T(z) => A – J can be solved.

The equation (5) has to be multiplied by amplification, if it is different than one.

Dynamic scaling => the amplifications of the pass bands of T(z) and T'(z) have to be equal This scaling is done by scaling A and D by factor μ.

$$\mu = \frac{|T(z)|}{|T'(z)|} \tag{7}$$

When the A - J has been solved => the capacitance values has to be chosen. The smallest factor (A-J) is set to two times larger than the minimum capacitance value, which can be realized in IC (can be also bigger, but then the largest capacitance can be too large to realize). Other capacitance values are scaled by the same factor. Scaling can be made separately with two groups:

Group 1: C, D, E, G and H
Group 2: A, B, F, I and J

Simulate and print the amplitude responses of a continues and discrete time transfer functions by using Matlab to see that they are correct. The examble of responses is shown in attachment 1.

Matlab

- Matlab can be started by command ***matlab.***
- Here are couple of links, which show show the amplitude responses can be plotted in Matlab:
  - http://www.ele.uri.edu/Courses/ele343/tutorials/matlab.bode.plots/
  - http://www.csupomona.edu/~zaliyazici/ece307/Frequency%20response%20of%20circuit.pdf – especially slide 10
  - The explanation of a command can be checked in Matlab by using command help <command>
  - An examble, which plots the continues time response in Matlab, is shown below:

```
f0 = 100e3;
q = 0.707;
Fs = 1e6;
w0 = 2*3.14159*f0;
count = 0;

for f = 1:100:Fs
  count = count +1;
  Hs(count) = w0^2/((j*2*3.14159*f)^2 + j*2*3.14159*f*w0/q + w0^2);
end

fx= 1:100:Fs;
semilogx(fx,20*log10(abs(Hs)),'r');
grid;
xlabel('f[Hz]');
ylabel('|H(f)| [dB]');
title('Continuous time response');
```

- Write the script by using text editor in Matlab and save it (***response.m)*** in your directory, where the Matlab was started. The script can be run by using a command ***response*** in command window. That should plot the response.
- ***semilogx***-command can be used to plot several responses in the same figure.
- You can also use some other program to plot the reponses, if you want to.

Required result in the report

- Hand calculation as shown above.
- The list of capacitance values of Fleischer-Laker biquad filter (for examble. C_A = 126 fF, C_B = 250 fF etc.)
- Amplitude responses of your continues and discrete time tarnsfer functions: |H(s)|, |T(z)| ja |T'(z)| plotted in the same figurep (y-axis in decibel and x-axis logarhitmic). See attachment 1.

2. Theory of designing OpAmp and switches for SC-filter

Operational amplifiers and switches has to be properly designed to be used in SC-filters. A book called"Analog MOS Integrated Circuits for Signal Processing" (R. Gregorian ja G. C. Temes) chapter 7 explains some of these rules. These things have been shortly presented in a course book too "Analog Integrated Circuit Design" (D. A. Johns ja K. Martin) pages 394 – 397. Some good tips can be also found from the laboratory work of electronics design II and exercise 5. Table 1 shows the process parameters of the used process.

Section 2.1 explains the theory of the nonlinearities of the OpAMp, Section 2.2 explains the nonlinearities of the switches, which affects the operation of the SC-filter and Section 2.3. presents the task, which have to be done in this work. Read the theory part first, so that you understand that and then start to make tasks.

Table 1. Proces parameters

	V_t (V)	$K_p = \mu \cdot C_{ox}$ ($\mu A/V^2$)	λ (1/V)
NMOS	0,460	130	0.13
PMOS	0,430	70	0.07

2.1 Nonlinearities of operational amplifier

SC-filter, which has been sized before by using Matlab, is realized by using lead compensated operational amplifiers. The place of the second pole of the OpAmp is affected by the load capacitance of the OpAmp. At first assume that the second pole is placed at the frequency of 20 MHz.

The OpAmp is compensated for by using Miller capacitance, which places a dominant pole and has an effect on a phase margin. The lead resitance is used to compansate for the zero of the transfer function of the OpAmp. An OpAMp is stable, if the phase margin is at least 60^o. From the SC-filter point of view the phase margin should be at least 10^o larger resulting to 70^o. In that case the SC-.filter will settle faster after transient. Too big Miller capacitance still increases the area and decreases the slew rate.

The amplification of an OpAmp has an effect on the accuracy of the transfer function of the SC-filter. Usually an error caused by the amplification has to be as small as possible, smaller than 0,1 %. If an OpAmp is connected to operate as a voltage follower, the output voltage as a function of an input voltage can be expressed as

$$V_{out} = V_{in}(1\text{-}1/gain) \qquad (1)$$

wher *gain* is the Dc gain of an OpAmp. This formula can be used to approximate the amplification of an OpAmp to achieve the required accuracy. (assuming that the bandwidth is infinity).

The unity gain bandwidth has an effect on how fast a SC-filter can charge. In the other words it gives the limit for a settling time. A tumb rule says that an unity gain bandwidth has to be at least 5 times larger than the clock frequency of SC-filter.

A Slew rate (SR) can also limit the settling time, so it has to be also adequate large. SR is usually limited by the Miller-C in a lead compensated OpAmp. If one stage OpAmp would be used, the SR is limited by the output stage and a load capacitance. The required SR for the

OpAmp depends also on the input signal of the used application. The maximum input signal step during a half clock period has to be calculated. Fig. 1 (Gregorian & Temes) below shows the clock signals and the output signal of the inverting integrator. t_{slew} is a rise time and t_{settle} is a settling time.

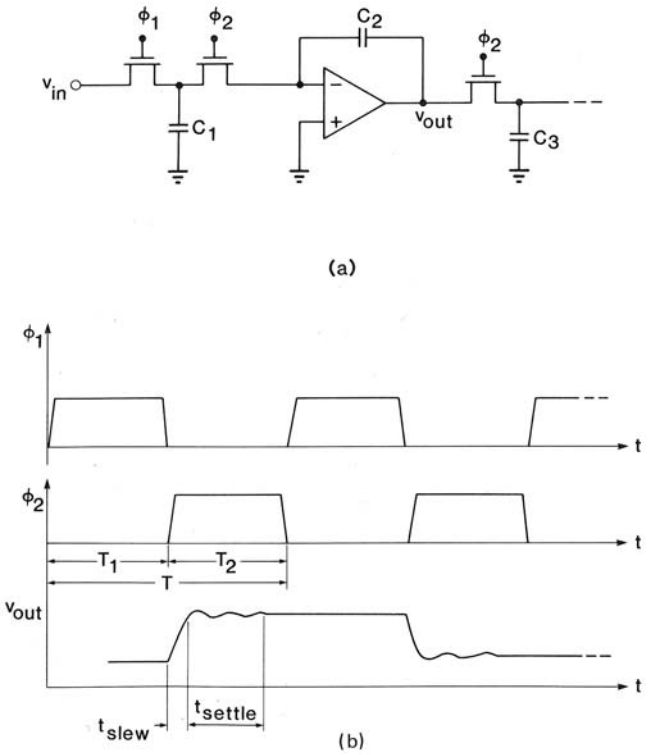


Fig. 1. (a) inverting integrator. (b) clock signals and output signal.

SR can be sized by using equation below (Gregorian & Temes):

$$SR \geq \frac{\Delta v_{out,\max}}{t_{slew}} \tag{2}$$

where t_{slew} is a rise time and $\Delta V_{out,max}$ is a maximum output step ($T= 1/f_c$) and can be expressed as

$$\Delta v_{out,\max} \approx \frac{|dv(t)|}{dt} \cdot T = \frac{\omega_B \cdot v_{\max}}{f_c}, \tag{3}$$

where ω_B is a corner frequency of the filter, v_{max} is the maximum amplitude of the input signal, and f_c is a clock frequency. The worst case scenario is used in calculation in which case the signal amplitude and frequency have maximum values (Fig. 2).
Inserting (3) to (2) we get

$$SR \geq \frac{\Delta v_{out,\max}}{t_{slew}} \approx \frac{\omega_B \cdot v_{\max}}{f_c \cdot t_{slew}}. \tag{4}$$

t_{slew} can be presented as a function of an active clock period => $x*T_2$ (x is the wanted fraction part) and T_2 is the active clock phase (approximately $½*(1/f_c)$) (see Fig. 1). In that case the

minimum SR can be expressed as

$$SR \geq \frac{\omega_B \cdot v_{\max}}{f_c \cdot x \cdot T_2} \approx \frac{2 \cdot \omega_B \cdot v_{\max}}{x}. \quad (5)$$

SC-filters are affected by the several other nonlinearities of the OpAmp, for examble the offset , output resistance and noise of the OpAmp, but these can be ignored in this work.

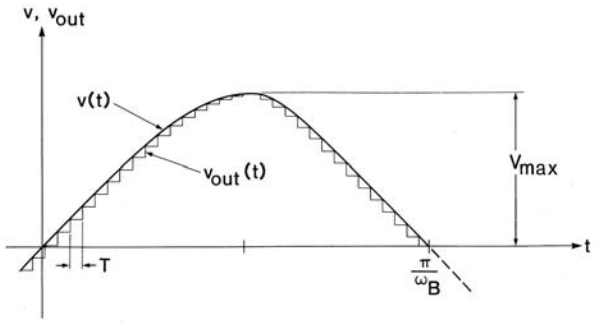


Fig. 2. The output of the integrator $v_{out}(t)$ and it's discrete waveform v(t).

2.2 Nonlinearities of switches

The operation of the SC-filter is also affected by the nonlinearities of the switches, that has to be taken into account designing the SC-filters. The nonlinearities of switches are an on-resistance, paracitic capacitances, leakage currents and 1/f noise among others.

The charging and discharcing speeds are affceted by the on-resistance of a switch. A clock signal causes cross talk through parasitic capacitances. Leakage currents discharges the cpacitances of the filter causing error. 1/f-noise can be minimized by using larger MOS-transistors in which case the cross talk is larger. 1/f noise and other noises and disturbances can be also minimized by using different type structures, for examble a fully differential structure and correlated double sampling (CDS technique).

In this work only the on-resistance of th switch has to be sized proberly.

Fig 3 (Gregorian & Temes) shows an inverting integrator, it's clock signals and an equivalent schematic, where the MOS switches are replaced by using ideal switches and serial resistances (on-resitances of switches).

The minimum on-resistance value for a SC-filter has been solved by using the schematic shown in Fig. 3 based on reference (Gregorian & Temes):

$$RC_1 \leq T/20 \quad (6)$$

where R is the on-resistance of switch, C_1 is a sampling capacitance and T is clock period ($1/f_c$). 0,1 % accuracy is achieved by using this formula to size an on-resistance in Fig. 3.

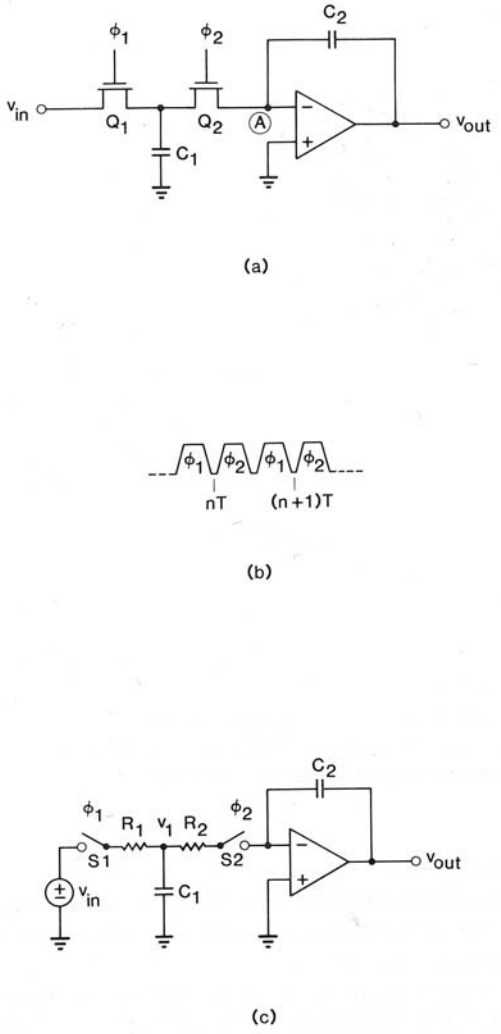


Fig 3. (a) inverting SC-integrator, (b) clock signals and (c) equivalent schematic with on-resistances.

2.3 Tasks

Next tasks will be made by using Cadence. The set-up of Cadence was explained at the beginning of this instruction. Remember to login to linux server and after that go to your design directory (ed3) *cd ed3*. Start Cadence =>

source csh_init; icfb&

The required OpAmp models can be found from library **es3_lib**. It consists of three OpAmp models and a bias circuit **bias.** The first OpAmp model (opamp_model_1) can be used to size the required frequency response and amplification (ideal model). The second model (opamp_model_2) has real transistors but an ideal Miller capacitor and lead resistance and it can be used to simulate transient and ac-responses. The third model (opamp_real) is real OpAmp, which can be fabricated by using CMOS process and it has a real Miller capacitance and NMOS-transistor operating as a lead resistance. The same OpAmp has been designed

during course Electronics Design II, so only the Miller capacitance and the Lead resistance have to sized here.

The opamp_model_2 and opamp_real require a bias-circuit **bias** to operate properly.

The model of **switch** can be foun from library **analogLib**.

When an operational amplifier and switches have been sized properly (R, C and switch), those components are realized by using components, which can be integrated into a chip..

Task 1: Size the DC-gain (gain) and gain bandwidth product (GBW) of an OpAmp so that it is stable and meet the requirements of SC-filter.

At first make your own design library. Use the pulldown menu of icfb-window *Tools -> Library Manager* and then the pulldown menu *Library Manager File -> New -> Library...* Give a name for your library and click OK => choose "*Attach to an existing techfile*" when a new popup-window arrives and then choose "*gpdk090*". To draw a schematic to your own library choose pulldown menu of *Library Manager File -> New -> Cellview...*and give name for your schematic and choose tool *Composer-Schematic*. Now you can draw a test schematic for the ac-simulation.

At first use **opamp_model_1 from es3_lib**. OpAmp consists of three parameters: the first pole (dominant) and second pole (*pole1, pole2*) in a frequency domain f and dc-gain (*gain*). Use the output load capacitance of the OpAmp in the simulation, which is seen at the output of the one or other of the two OpAmps during active clock phase of the SC-filter.

After you have drawn and saved the schematic open a simulator by using the pulldown menu of the schematic

Tools -> Analog Environment.

Before starting the simulation all the variables have to be set (pole1, pole 2 and gain) by using pulldow

Variables -> Copy From Cellview.

Set the second pole (*pole2*) at 20 MHz and size the *pole1* (dominant pole) and *gain* (dc-gain) so that your OpAmp is stable and meets the requirements of the SC-filter (see theory part)

After that the requirements of OpAmp is meant to be realized by using a Lead compensated OpAmp, whose schematic can be found from library **es3_lib.** Only the Lead resistor R, the Miller capacitance C and the resistor of the bias circuit have to sized properly to meet requirements.

Use now the **opamp_model_2,** add a bias circuit **bias** and place the bias resistor from the library **analogLib** between bias node and ground. Set the initial value of the bias resistor to 70kΩ. Use the output load capacitance of the OpAmp, which is seen at the output of the one or other of the two OpAmps during active clock phase of the SC-filter. Use supply voltage 3 V and groun 0 V (gnd from **analogLib**) and analog ground 1,5 V.

At first simulate the circuit with the values of R and C, *Rlead = 0 and Clead = 0*. What is the dc-gain of the circuit,?

Size the R and C (*Rlead and Clead)* based on "Analog Integrated Circuit Design" (D. A. Johns ja K. Martin) pp. 242 – 244. Simulate the circuit again and make some modification if needed. Change the resistor value of the bias circuit, if you cannot meet the specification by using any other modification.

After that replace the **opamp_model_2** for the **opamp_real** and size a NMOS transistor in triode region to replace R and C by using real **mimcap** (= metal-insulator-metal capacitor).

A hint: The resistance value of the NMOS transistor can be simulated by using Dc-analysis as follow pulldown menu *Analyses -> Choose...* (choose *dc* and *Save DC Operating Point*) and after simulation use pulldown menu *Results -> Print -> DC Operating Points* and choose the NMOS transistor. A resistance value can be found from the list *ron*.

Required result in the report

- Give the minimum requirements for the OpAmp.
- Give the values of *Rlead* and *W* and *L of NMOS transistor, Clead*, *gain* and *GBW and Rbias*).
- Print the frequency response of the OpAmp, where the PM and GBW and gain can red.

Task 2: Check/size the slew rate of the operational amplifier so that the rise time of the amplifier (tslew) is less than 10 % of the half of the clock period in which case the output of the amplifier has time to settle accurately during every sampling phase.

Calculate the positive and negative slew rates of the operational amplifier (*slewp*, *slewn*) based on specification by using equation (5). Draw the schematic for a transient simulation to be used to simulate slew rate as explained below. Use the OpAmp model **opamp_real**.

Use OpAmp as a voltage follower and connect the load capacitance at the output (the largest load during active clock phase at the output of the filter). Supply voltages are 3 V and 0 V (ground) and an analogical ground 1,5 V. Input signal is clock signal with an amplitude of 0.5 V, a dc-value of 1.5V. The easiest way to make this clock signal is to use **vpulse** –source from library **analogLib**. Use rise and fall times of source 1 ns in which case the input signal does not distort the result. A frequency can be 50 kHz. Simulate at least ower two clock periods in order to see both the rising and falling SRs. Determine the *slewp* and *slewn* from the result of the simulation. If the SR is smaller than the calculated values based on the specifications, try to find out if the output stage is too slow by decreasing the value of the load capacitance notably. Change your capacitance values of the filter or the size of the resistor of the bias-circuit depending on the limiting stage if you cannot meet the specification. **If you have to change the value of the bias-resistor, the first task has to be made again to assure the stability of the OpAmp!**

When the OpAmp meet the SR specifications, assure if the OpAmp has time to settle by sampling rate with the accuracy of 0.1%. Now calculate the $\Delta v_{out,max}$ by using (3) and use this value as peak-to-peak value of the input signal (DC-value 1.5 V) and the frequency of the clock frequency of the filter. If the output signal settles during a half period of the clock, the SR and also the phase margin are adequate. Note, that an adequate SR does not alone quarantee the proper operation, if a phase margin is poor. No problems should arrive, if phase margin had been adequate in AC-simulations.

Required result in the report

- Minimum required SR (*slewp* and *slewn*)calculated by using (5).
- Simulated waveforms at the input and the output of the OpAmp during at least two periods, which shows SRs and also settling.

Task 3: Sizing swirches. Size the on-resistance value of a switch so that a charge can be delivered adequate fast at the worst case situation.

Size the on-resistance of a switch by using (6). Use the maximum capacitance value of your filter or the sum of capacitances, if you have removed the redundant switches..

Draw the schematic for a transient simulation consisting of a voltage source, an ideal switch and a capacitance in series. Draw a one more switch in parallel with a capacitance. Connect the voltage source and capacitance to the analog groud. Use two non-overlapping clock signals, whose rise and fall times are 1 ns, to control the switches (see Fig 3b) and the non-overlapping time is 4 ns. Use **vpulse**-source to generate waveforms (from library **analogLib**). The frequency of the clock signal is same as the clock frequency of your filter and the maximum and minimum voltages are 3 V and 0 V, respectively.

Note! Switch is controlled through node ”Wx”, where x is the running number of the switch. The other control node is connected to ground.

Set the switching voltage to the middle of the power supplies and size the on-resistance based on (6) and set an off-resistance = 1GΩ. Set the last resistance value to 1 kΩ (the resistance of the inductance). Use two test signals: 1 V and 2 V DC-values, which are the minimum and maximum values at the input, output or middle of the filter. Simulations show how perfectly capacitances are charged or discharged with some on-resistance at the worst case situation. If on-resistance is sized properly, the capacitance is charged accurately to the test value during one clock phase and discharged accurately to the analog ground during another clock phase.

Simulate the same test with NMOS or (CMOS) switch sized properly. Simulate the test circuits also by using the maximum amplitude sine wave at the input and the frequency of the sine wave has to be same as your corner frequency of the filter..

Switch can be realized by using NMOS- or PMOS-transistor or CMOS-switch, which consists of both transistors. Choose the type of your switch and size the W and L of the transistor.

Note!When you are realizing new components/blocks, make a new schematic and symbol, which can be then used in a test bench, which consits of supply voltages, inputs and loads.

Hint: The resistance value of the switch transistor depends on the W/L and the control voltage V_{GS}. If the control voltage is large enough, the triode region of the transistor is achieved fastly in which case the source and drain potential are equal in a switch. The resistance value of the switch can be determined by using the schematic shown in Fig 4 below (NMOS-switch).

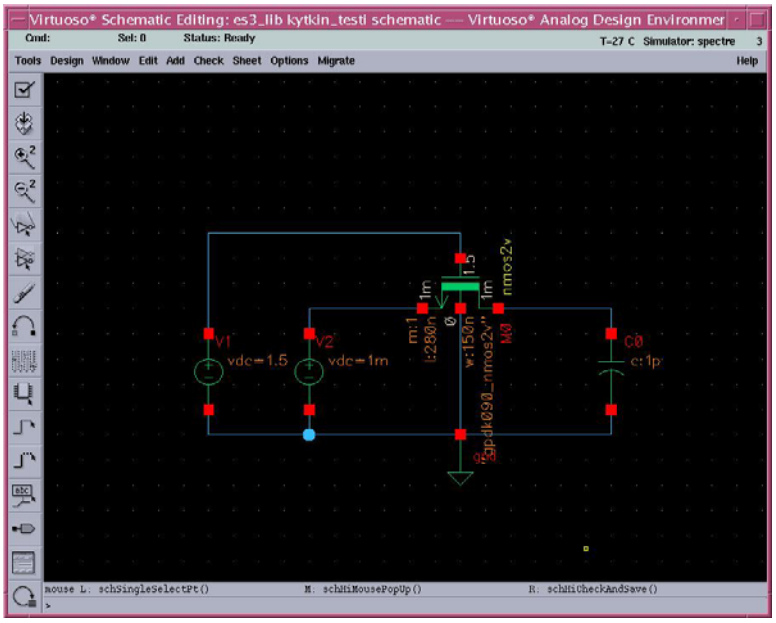


Fig 4.

The other end of the switch transistor is connected to (source or drain) dc-source and the other to a capcitance. In that case V_{DS} will settle to 0 V -> triode region. The resistance value can now be determined by using DC simulation. V_{GS}-voltage is set to the minimum possible value of your filter (depending on input voltage and gate voltage). On-resistance can be printed by using pulldown *Results -> Print -> DC Operating Points* after simulation. Choose the component in the schematic. Resistance value is named *ron* on the list.

Required result in the report

- How the on-resistance was sized and it's value.
- Size of the MOS switch and How did you determine that.
- The schematics used in simulations .
- Simulation results consisting of the waveforms of clock signals so that the non overlapping and symmetry can be seen, test voltages and a capacitance.

3. Confirming the operation of SC-filter

The whole SC-filter consisting of all components sized before is simulated to be sure that a response corresponds to that of the Matlab simulation.

Draw the schematic of the whole filter by using the operational sized before (**opamp_real)**, **bias** – circuit, MOS switches (sized before) and capacitors **mimcap** from library **gpdk090**. Drwa the terminals needed (inputs, outputs, clock signals etc.) by using pulldown menu *Add -> Pin...*), but don't draw any voltage sources. In that case you can make a symbol corresbonding to the schematic, which can be used in simulations. Use pulldown menu *Design -> Create Cellview -> From Cellview...,* accept the next window and you can draw the symbol. Draw the test bench for the simulation, to which you can add the symbol of the OpAmp and signal sources etc.

Use non-overlapping clock signals to control the switches as you did previously in task 3. The input signals has to be the same as in task 3: DC-input 1 V, DC-input 2 V and a sine wave with the frequency and amplitude of the maximum values of your specification.

Required result in the report

- The schematics of the filter and the test bench
- Simulation results: clock signals, inputs and outputs during at least two periods of the input signal (DCs and sine wave).

4. Drawing the layouts of the OpAmp and bias-circuit of the SC-filter

The layout of the SC-filter is drawn next. The layout consists of only two OpAmps and bias circuits. The layouts of capacitance arrays and switches has not to be drawn. The layout of the circuit descripes the real method of implementation of circuit. The hierarchial structure is used designing layout, where at first the individual components are drawn (opAmp and bias-circuit)and after that those components are put together by using higher level schematic consisting of only the symbols of the OpAmps and bias circuit. (the output of the 1. OpAmp and the inputs of the 2. OpAmp are not connected, but marked by using terminals, pins) The layout of switches and capacitance arrays could be designed later and included to realize the whole layout of the SC-filter.

Aims:

1. Draw a hierarchical layout by using Cadence.
2. Use the good principles of layout designing.
3. Check the design rules.
4. Check the correctness of the layout by comparing it to your schematic.

Realizing a layout

The typical phases of layout designing:

- Component placing
- Wiring
- Checking the layout

Cadence offers three different method to draw layouts:

- manual
- schematic driven
- automatic placing and wiring

In this work the best method for analogue cirrcuit is used mening the schematic driven layout. A manual method is also possible, but is labour-consuming. An automatic method is usually used to design digital circuits.

The instruction of the layout tools can be found from on-line manual, Cadence's icfb-window using (*Help -> Cadence Documentation -> Virtuoso XL Layout Editor -> Virtuoso XL Layout Editor User Guide -> Table of Contents*), or using link:

file:///elsoft1/IC5141USR3/doc/vxlhelp/vxlhelpTOC.html

The documents and design rules of a HitKit can be found from:

file:///elsoft1/gpdk090/gpdk090_v3.0/docs/gpdk090_DRM.pdf
An example of the layout drawing and checking can be found from a link:

http://vlsi1.engr.utk.edu/~scterry/hw7_test/homework7b.html

Couple of examples of the layout of an OpAmp are presented in attachment 2.

Good principles of layout designing

To achieve a working IC-circuit, it is important to use these good principles of layout designing, because the most of the nonlinearities of the layout cannot be predicted by using schematic simulations. **Good principles of layout designing consists of** :

1. **Matching rules** (minimizing the difference of two components)
2. **Shields** (Minimizing disturbances and the risk of latch-up)
3. **Good floorplan** especially in mixed-signal circuits (minimizing disturbances)
4. **placing pins** (minimizing disturbances and inductances of bonding wires)

These principles in more detail can be found in reference *The Art of Analog Layout* –book (see references). More information can be also found in the course books and other references.

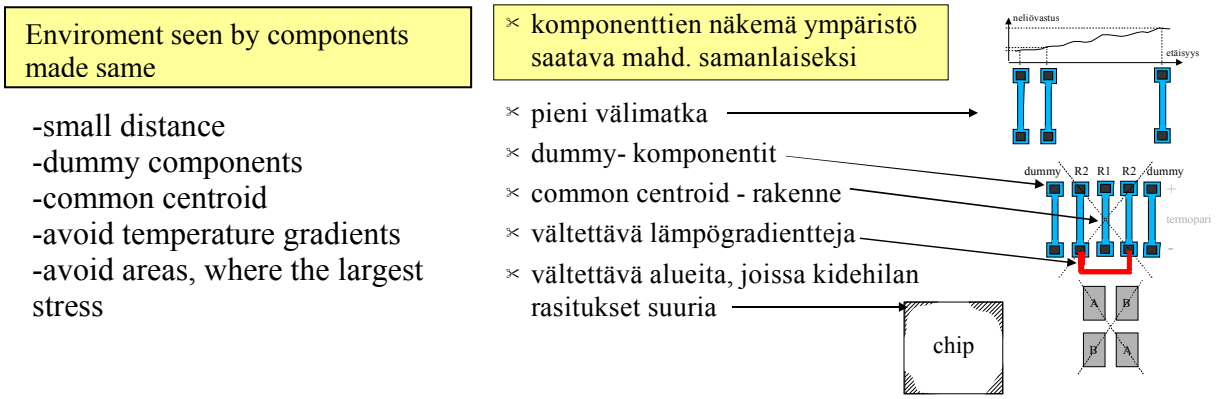


Fig. 5. Principles of matching.

Note!You have to use at least dummy- and common-centroid –structures in your layout! Divide the transistor to be matched to several parallel elements, which can be placed in common centroid structure.

An example of the common centroid structure of MOS-transistors are presented in Fig. 6. Assuming that two transistors, whose Ls are equal size and W1=2u and W2=16u, have to matched. The best possible matching is aschieved by dividing the transistor having the W of 16u by 8, in which case we have 8 transistor having W of 2u. In that case all the transistors have same size and the ratio of transistors Q_1 and Q_2 can be realized by connectin these eigth transistors in parallel. After that the transistors are placed so that they have the same geometric middle point as shown in Fig. 6. Now the wiring has to made so that all the drains, sources and gates of Q_2 have to be connected together, respectively.

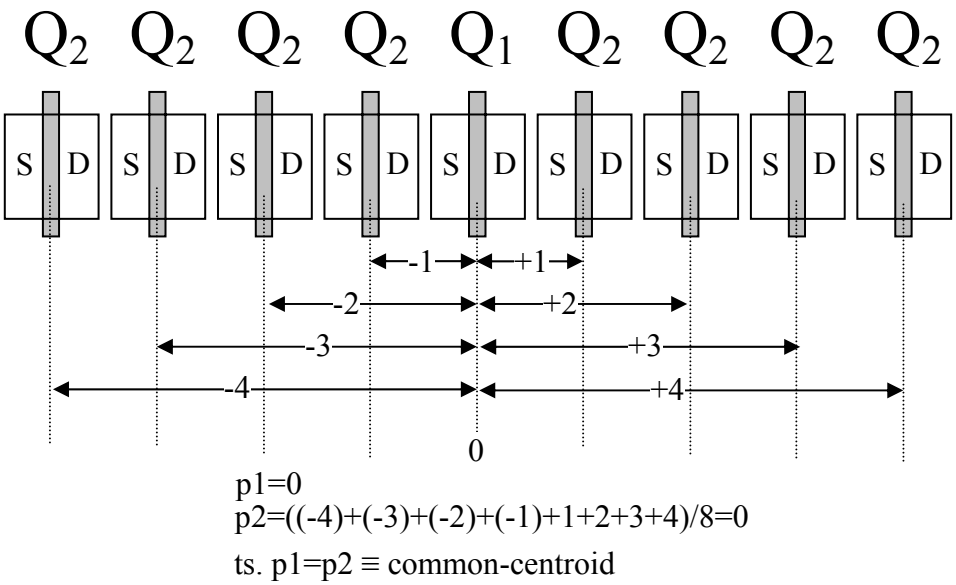


Fig 6. An example of common-centroid -structure.

A common centroid structure of MOS pair with dummy gates is presented in Fig. 7. Note! This structure can be used when the even number of components have to matched.

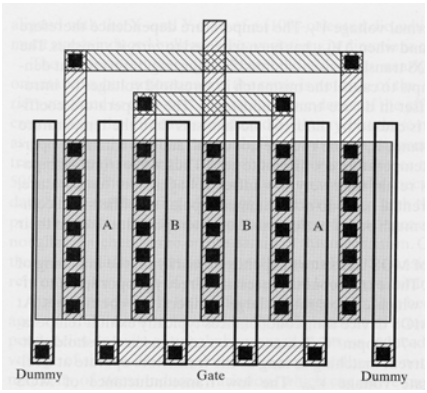


Fig. 7. Common-centroid -MOS-pair with dummy-gates.

Drawing a layout

The Cadence generates automatically all the components in a schematic to a layout when a schematic driven layout method is used. Additionally, the missing connections are marked by different colours lines. The places of the components have to be thougth also from the floorplan point of view. Drawing is still started from the bottom of the hierarchy and is continued to a top.

Generating a schematic driven layout

CAdence is started by using command

source csh_init_general;icfb&

The layout of a schematic can be generated by opening first the schematic of the circuit for

example **opamp_real** from **es3_lib** and starting Layout XL by using pull-down menu

Tools -> Design Synthesis -> Layout XL.

Choose (Startup option)

Create new

In the next window check that a component is editable and accept selections. After that all the components and terminals (pins) are updated to screen by using pull-down menu

Connectivity -> Update -> Components and Nets.

A new window appears, where you choose the sizes and layers of the pins. Usually metal1-layer is chosen (Choose Layer/Master -> Metal1 pn and then click apply).

After that you have to choose *Pin Label Shape: label*
Then choose *Pin Label options.. layer name->* choose the second *Metal1* and *Layer purpose->* choose *label.*
After that you can choose the height of the label. Choose for example 0.1. and click Ok and again OK.

The missing connections are shown by using pull- down menu

Connectivity -> Show Incomplete Nets -> Select All.

The layout editor shows only the highest layer as default. Other layers can be chosen by using pull-down menu

Options -> Display...

and writing in *Display Levels Stop* – window for example. 10.

A chosen component in a layout is also chosen in a schematic, which helps to recognice the different components. Any changes made in schematic can be updated in layout editor by using pull down menu

Connectivity -> Update -> Layout Parameters.

Moving a component

A component can be moved by choosing a component by clicking the left mouse button and keeping it down and moving a mouse. Components can be rotated or flipped by pushing the key m and after that F3. After that you can choose different options (rotate, flip sideways etc…)

Wiring

The wiring material is chosen by clicking the left mouse button on the top of the layer in the LSW window. Choose for example Metal1 dg (drw = drawing) and then use pull down menu

Create -> Path,

Click the left mouse button at the strating point and corner points and the path is ended by douple clicking the left mouse button or using enter. Default width of the path is minimum. That can be changed by using pull down menu
Edit -> Properties.

Pull down menu

Options -> Layout Editor -> Gravity on

can be turn off to help to draw path.

Path segments can streched by using pull down menu

Edit->Stretch

and then choose the segment and move the mouse.

One good rule is that horizontal paths are drawn by using for example metal1 and vertical paths by using metal2.

Substrate contacts

In additiond to components substrate contacts have to be drawn. These contacts connect the environment of the transistors to a node determined by schematic. In other words if the fourth node of the NMOS is connected to a ground in a schematic, the p-substrate of the IC has to be connected to a ground potential by using p-substrate contacts (M1_PSUB). If the fourth node of the PMOS is connected to a supply voltage in a schematic, the NWELL of the PMOS has to be connected to a supply voltage by using NWELL contacts (M1_NWELL). These contacts and also the vias between metals can be found by using pull down menu *Create ->Contact.* Choose only the type of the contact. Note all the NMOS transistors are connected in the same substrate (has to be connected ground), but PMOS transistors are placed in their own NWELLs so they can be connected also to the source of the PMOS (input diff pairi). Often it is practical to connect all the matched PMOS transistors with same NWELL connection to same NWELL, which is connected to a supply voltage or the source of the PMOS depending on the schematic
.

Checking a layout

Finished layout has to be checked by using=>

- Design rule check (DRC) Checks the distances and widths etc of different layers
- Comparison between schematic and layout (LVS)

Good instruction of DRC and LVS can be found from HitKit's on-line document:
file:///elsoft1/IC5141USR3/doc/divaref/chap14.html

The short overview of that instruction is presented below.

DRC

DRC will check that all the widths and distances of layers are correct. DRC is started by using

Virtuoso-window's pull-down mmenu

Verify -> DRC,

Choose the points below

- flat/hierarchical (flat is recommended, hierarchical can be made once in the while if the cicruit is large)
- full/incremental (full is recommended and has to be done finally, incremental can be made once in the while if the cicruit is large)
- Join nets with same name (don't use)
- Rules name and library "divaDRC.rul" and "gpdk090" usually defaults
- Set Switches, choose SUGGESTED_CHECK

Errors are shown by a white color. The explanation of the error can be plotted by using pull down menu

Verify -> Markers -> Explain

and clicking the white area or

Verify -> Markers ->find

And then choose *Zoom To Markers* and click next.

DRC-report can be printed by opening CDS.log – file in your home directory and copying DRC lines starting from text "DRC started....." to a text editor and printing that file.

Comparison between schematic and layout (LVS)

To check LVS, the layout has to be extracted by using pull down menu
Verify -> Extract.

After that open the extracted view of the circuit. LVS is started by using pull down menu

Verify -> LVS.

Give the names of extracted and schematic views (if not shown) Rules file => *divaLVS.rul* and Rules library => *gpdk090*i. LVS is started by clicking

Run.

The results of the LVS can be plotted by clicking *Output*. Errors can be search by using button *Error Display* in LVS-window.

LVS-report can be printed by copying the *Output*-file to a text editor and printing that or printing the si.log file, which locates in LVS directory generated by Cadence in your working directory.

Drawing hierarchical layout

When you have finished the lowest level layouts, you can start to draw the layout of the next level. Generate the schematic of the next level and after that use layout XL to generate the layout of that schematic.

If the circuit would send to be fabricated, post-layout simulations, I/O pad and packaging designing, and mask file generation should be made. This work does not consists of these parts. The good principles of layout designing consists of the use of common centroid and dummy structures meaning that the components have to be divided unity components, which can be used to generate bigger components.

The blocks of the layout should be also placed properly meaning good floorplan. Fig. 8 shows a floorplan of a SC-filter based on ”Analog Integrated Circuit Design” (D. A. Johns ja K. Martin) In this work the layouts of capacitances and switches are not designed so Fig. 8 only shows the hint how the layout of the whole SC-filter could be finished.

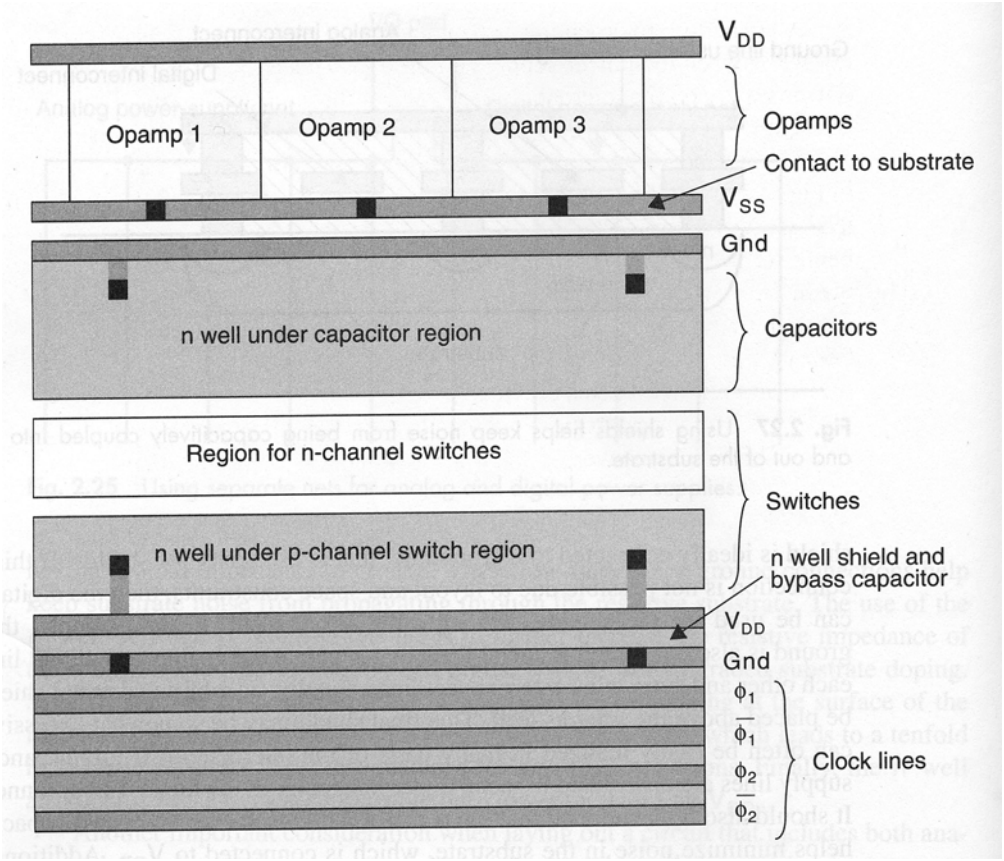


Fig 8. the floor plan of a SC-filter.

Required result in the report

- The figure of the whole layout consisting of OpAmps and bias circuit.
- The reports of LVS and DRC (the highest level).
- Used principles of good layout designing (refered to layout).

- Answers to the next questions:
    1. What does the good principles of layout designing mean and what is the aim of those?
    2. What is the middle point of the common centroid structure? Draw an example.
    3. What is the reason for the use of dummy structures?
    4. What are the substrate contacts? How many types of these contacts can be used? Why are these contacts used?
    5. What is a structure, which determines the points to be connected in a higher hierarchy level?
    6. What does DRC mean? What kind of checks are made by using DRC?
    7. What does LVS mean? If LVS is succesful, what it means.

References

[1] D.A. Johns & K. Martin, Analog Integrated Circuit Design, Wiley & Sons 1997, kpl 2.

[2] P. E. Allen, D. R. Holberg, CMOS Analog Circuit Design, Oxford University Press, 1987.

[3] R. Gregorian ja G. C. Temes, Analog MOS Integrated Circuits for Signal Processing, Wiley & Sons, 1986, kpl 7.

[4] M. Ismail & T. Fielz, Analog VLSI Signal and Information Processing, McGraw-Hill 1994, kpl 16.

[5] A. Hastings, The Art of Analog Layout, Prentice Hall 2001

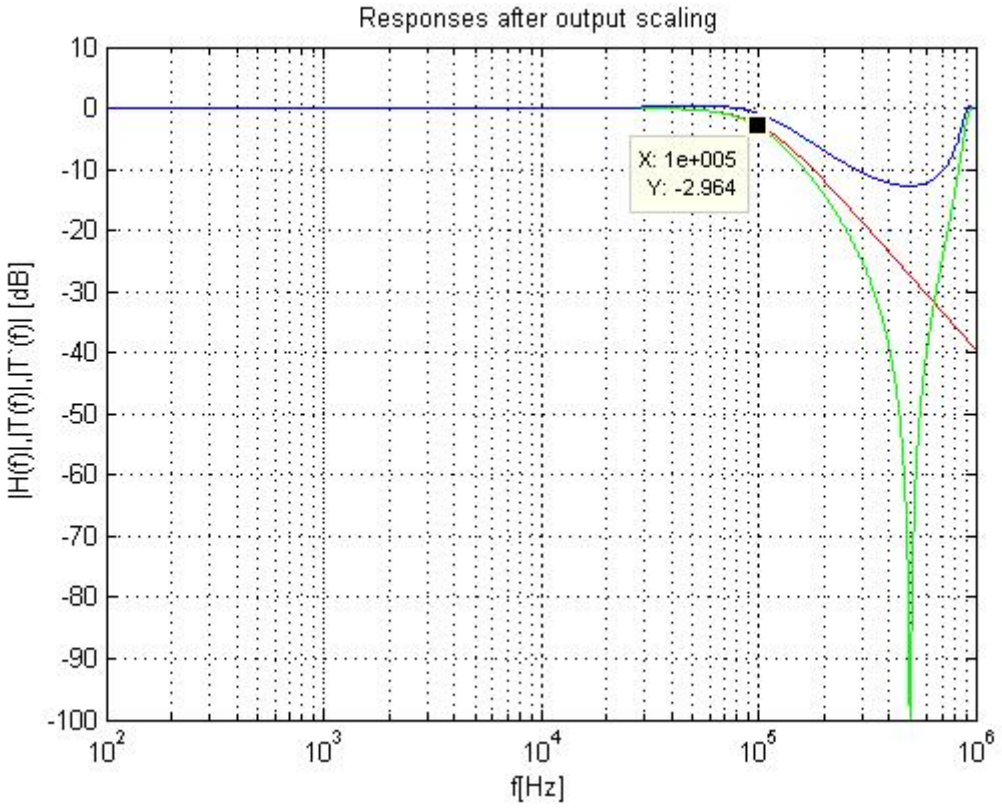
[6] Cadences layout-editor on-line manual: file:///elsoft1/IC5141USR3/doc/vxlhelp/vxlhelpTOC.html

[7] HitKit documentation: file:///elsoft1/gpdk090/gpdk090_v3.0/docs/gpdk090_DRM.pdf

[8] An example how to draw and check the layout of the OpAMp: http://vlsi1.engr.utk.edu/~scterry/hw7_test/homework7b.html

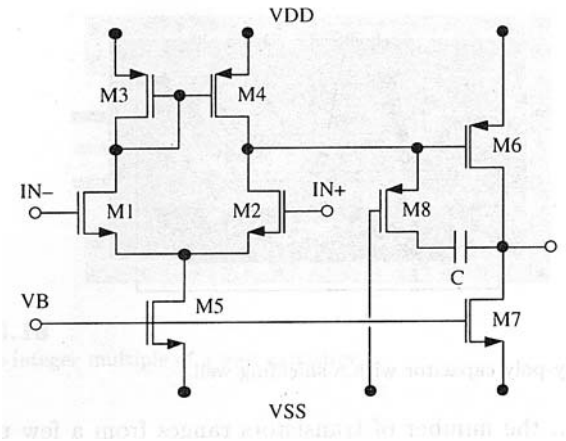
[9] AMS, Analogue & Mixed Signal Application Note, Device Matching Rules, file:///elsoft2/DesignKits/ams_3.40/www/appnotes/analog/matchingrules.html

Attachment 1. Magnitude responses of filter.

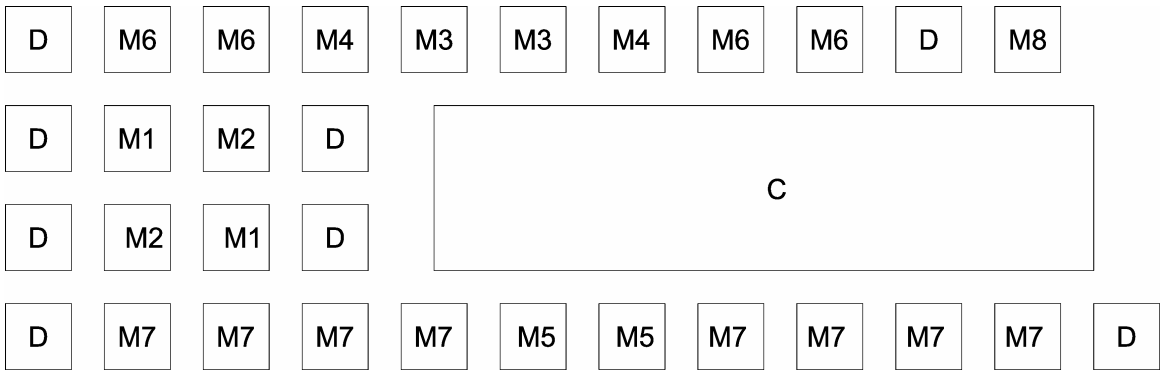


The specifications of the filtert: f_0 = 100 kHz, Q = 0,7, A = 1 (f_s = 1 MHz)

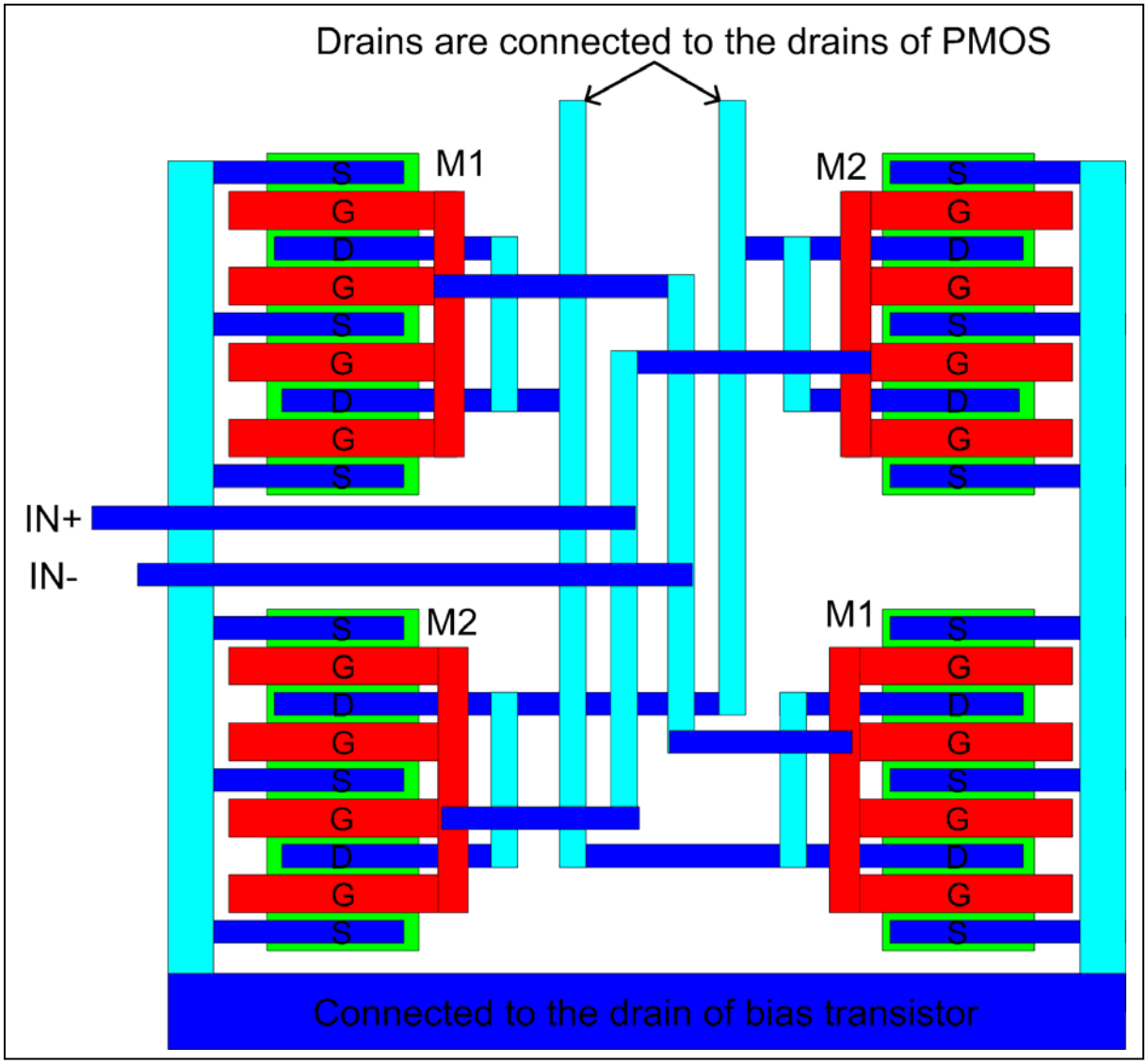
Attachment 2. an example of a layout of an OpAmp (Analog VLSI : signal and information processing / Mohammed Ismail, Terri Fiez):



Schematic.



Placed components using unity components and dummies.



The cross-coupled input pair of the OpAmp (blue is metal 1 and light blue metal 2). The wiring is also symmetrical.