Assignment 1: Simulation and Design of CT Integrators and Filters

Objectives

- To learn how to simulate the small-signal and large-signal response (including the linearity: THD, SFDR) of integrators with opamps and transconductors.
- To translate the specification of the 2nd order transfer function into a circuit-level implementation and to design the associated transconductors.

1. Preparation:

1) Study the course material, Chapter 12

2) Obtain access, sign NDA and setup the ST 65-nm CMOS design kit.

You can find setup instructions and agreement forms at

/CMC/kits/cmos65nm/==README.ECE1396-ECE412

/CMC/kits/cmos65nm/Student_Acknowledgement_Statement.pdf

The ST65nm CMOS Training documents are available at

/CMC/kits/cmos65nm.1.0/cmos065/

DK_cmos065lpgp_7m4x0y2z_50A28A_4.2.1/doc/Manuals/

Open the cds.lib file in the Cadence Analog Artist directory you have created and verify that the following two lines exist:

DEFINE ahdlLib /CMC/tools/cadence/IC/tools/dfII/samples/artist/ahdlLib

DEFINE analogLib $CDS_INST_DIR/tools/dfII/etc/cdslib/artist/analogLib

If one or both are missing, please add them and save the cds.lib file BEFORE you start the design kit.

2. Op-amp integrator simulation (10 points)

Create the opamp integrator testbench in Fig. 1. Set V_{DD} to 1.1V, *vref* = V_{DD}/2 = 0.55 V. R_1 =1 kΩ and C_1 = 100 fF.

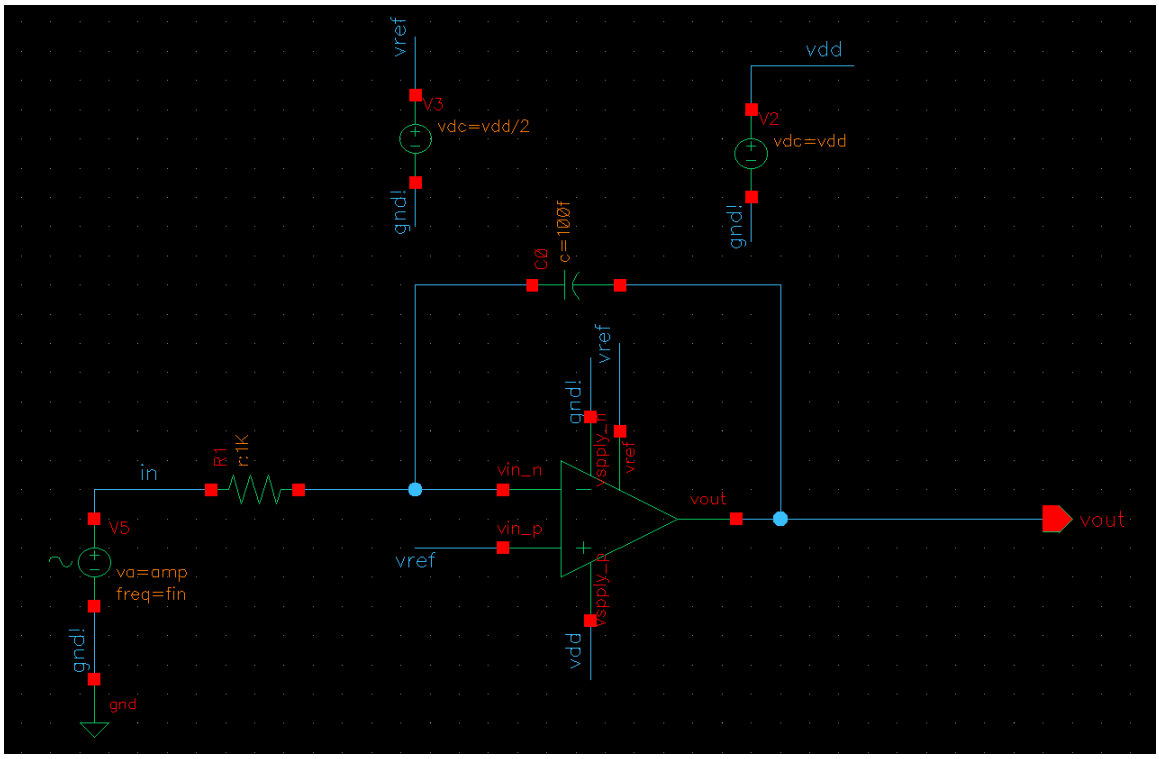

Fig.1 Opamp integrator testbench.

The opamp block can be found in the *ahdlLib*. Once you have inserted the opamp, query its properties and set them as in Fig.2. They show that the unity gain bandwidth, ω_{ta}, of the opamp is 10 GHz, the DC gain A_0 = 100, the input resistance is 10 MΩ, the slew rate is 0.1 V/ps and the output resistance is 50 Ω. The opamp offset voltage is left as a variable which you can set to 0 for now. *Ibias* and *Iin_max* are left as blank, which

means that they are set to the default values pre-programmed in the verilog-A model shown partly in Fig.3.

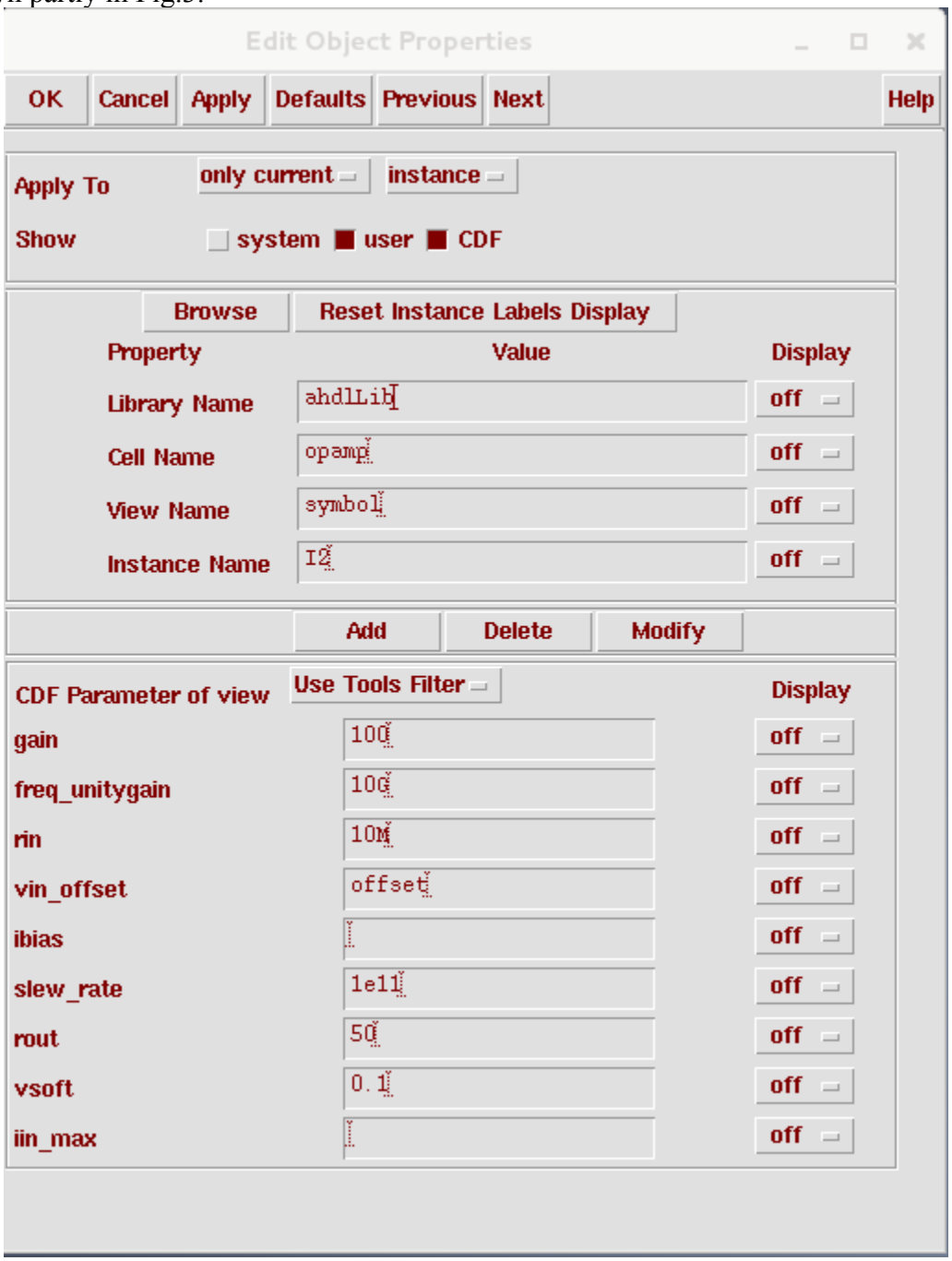

Fig.2 Opamp parameters.

VerilogA-Editor Editing: ahdlLib opamp veriloga

Design Help 3

```
`include "discipline.h"
`include "constants.h"

// $Date: 1997/08/28 05:45:21 $
// $Revision: 1.1 $
//
//
// Based on the OVI Verilog-A Language Reference Manual, version 1.0 1996
//
//

`define PI      3.14159265358979323846264338327950288419716939937511



//--------------------
// opamp
//
// -  operational amplifier
//
// vin_p,vin_n: differential input voltage [V,A]
// vout:        output voltage [V,A]
// vref:        reference voltage [V,A]
// vspply_p:    positive supply voltage [V,A]
// vspply_n:    negative supply voltage [V,A]
//
// INSTANCE parameters
//    gain           = gain []
//    freq_unitygain = unity gain frequency [Hz]
//    rin            = input resistance [Ohms]
//    vin_offset     = input offset voltage referred to negative [V]
//    ibias          = input current [A]
//    iin_max           = maximum current [A]
//    slew_rate      = slew rate [A/F]
//    rout           = output resistance [Ohms]
//    vsoft          = soft output limiting value [V]
//
// MODEL parameters
//    (none)
//

module opamp(vout, vref, vin_p, vin_n, vspply_p, vspply_n);
input vref, vspply_p, vspply_n;
inout vout, vin_p, vin_n;
electrical vout, vref, vin_p, vin_n, vspply_p, vspply_n;
parameter real gain = 835e3;
parameter real freq_unitygain  = 1.0e6;
parameter real rin = 1e6;
parameter real vin_offset = 0.0;
parameter real ibias = 0.0;
parameter real iin_max = 100e-6;
parameter real slew_rate = 0.5e6;
parameter real rout = 80;
parameter real vsoft = 0.5;
   real c1;
   real gm_nom;
   real r1;
   real vmax_in;
   real vin_val;

   electrical cout;


   analog begin

      @ ( initial_step or initial_step("dc") ) begin
```

Fig.3 Opamp verilog-A model.

All these parameters can be changed by you. You can open the verilog-A code of the opamp to understand how the opamp is modelled and the meaning of each parameter. The parameter *vsoft* smooths the clipping of the opamp output voltage when it approaches *vsoft* and V_{DD} -*vsoft*. Its value, along with that of the supply voltage, affect the linearity of the opamp and of the integrator.

a) Using the testbench in Fig.1, simulate the small signal response of the integrator from 1 kHz to 20 GHz. What are the values of k_1, k_0 and ω_0 of the corresponding first order transfer function implemented by this circuit? (2 points)

b) Apply an input sinusoidal signal at 1 MHz with 1mV amplitude and run a transient simulation for 20 µs. Plot the waveforms at the input and at the output of the circuit. Use the Cadence calculator to plot the Fourier transform of the output signal (***dft*** function) with 1024 samples and rectangular window (see Fig.4). Plot the results in dB => use the *db20(dft)* function. What is the ***SDR*** in dB ? What is the ***THD*** in %. (2 points). **Note:** ***SDR*** is the difference in dB between the fundamental and the harmonic with the largest power.

c) Repeat *b)* with the input amplitude set to 100 mV (2 points)

d) Replace R_1 with a n-MOSFET with 60nm gate length, 2µm gate width, 2 gate fingers each 1µm wide, as in Fig.5. Use the *nsvtgp* device from the *cmos065* library. Plot the

3dB bandwidth of the integrator as a function of control voltage on the gate of the MOSFET when the control voltage changes from 0 to 1.1 V. Repeat c) above with the control voltage set to 0.7 V.

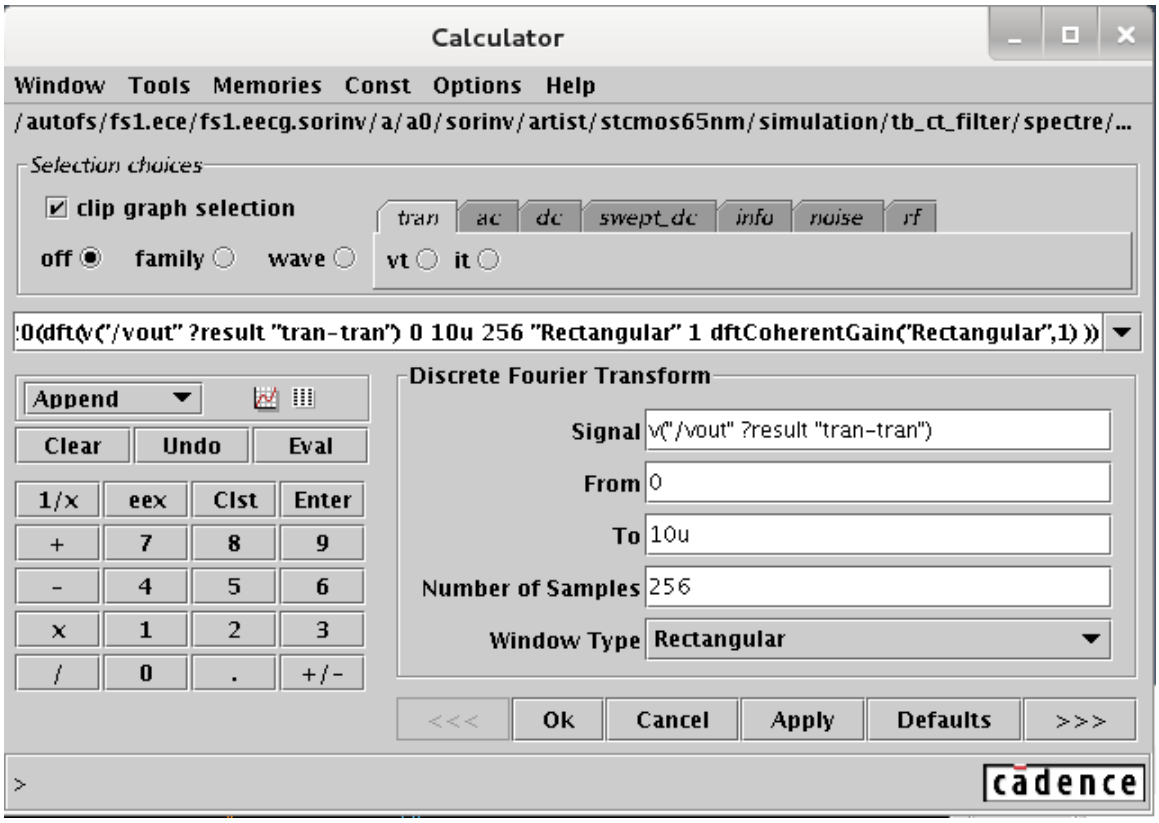

Fig.4 DFT setup.

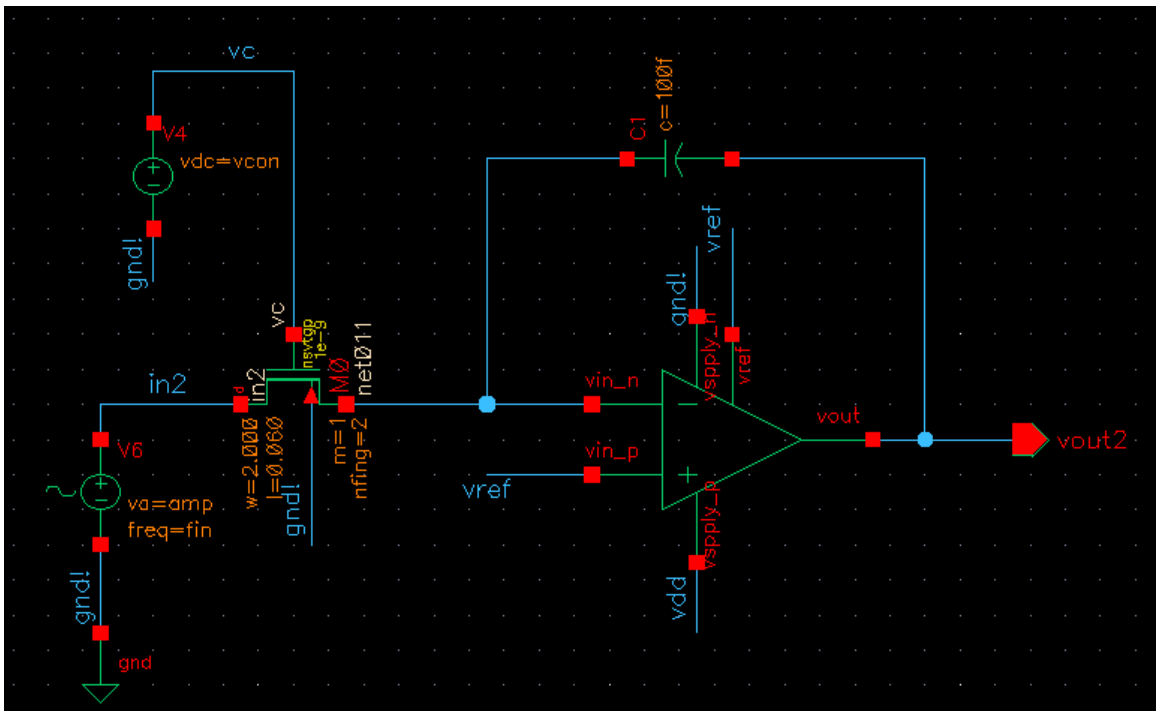

Fig.5 Tuneable MOSFET-C opamp integrator schematic.

3. Design of the second order bandpass filter centred on 200 MHz discussed during the lecture 3 (slide 13) using fully-differential transconductors (10 points)

You can use any of the transconductor topologies discussed in the book or in class.

You MUST use n-MOSFETs, p-MOSFETs, 3-terminal MOM capacitors and 3-terminal polysilicon resistors from the cmos065 library but you may use ideal current sources (only where they are used in the textbook schematics) from the *analogLib* library.

You will simulate the small-signal transfer characteristics and the ***SDR*** of the filter at the nominal temperature of 65 C and an input differential signal amplitude of 250 mV.

You will also simulate and plot them in the *slow* and *fast* corners and plot $|H(f)|_{dB20}$ in the *nominal*, *slow* and *fast* corners in the same graph to illustrate how sensitive your filter is to process variation.

The *slow corner* is defined by setting all MOSFETs in the *SSA* corner, the resistors to *Rmax,* and the capacitors to *Cmax*.

The *fast corner* is defined by setting all MOSFETs in the *FFA* corner, the resistors to *Rmin,* and the capacitors to *Cmin*.

Include transconductor, resistor and capacitor values in your report, the schematics and the simulation plots. Report the power consumption. The supply voltage cannot exceed 1.1 V.

The circuit with the best performance in terms of power consumption, ***SDR***, and process variation will get 2 bonus points to the final mark of the entire assignment.

Notes:

- Use MOSFETs from the cmos065 library.
- In the MOSFET form
  - Set “cell name” to “nsvtgp” or “psvtgp”(standard threshold)
  - Set “number of fingers” to the desired number of fingers *NF*
  - Set “width” to the desired total gate width *W* (it is in micrometers)
  - Set “length” to 0.06 (it is in micrometers)
- You must set up the Corners and Simulation Levels in the Tools form before you start simulations for the first time.