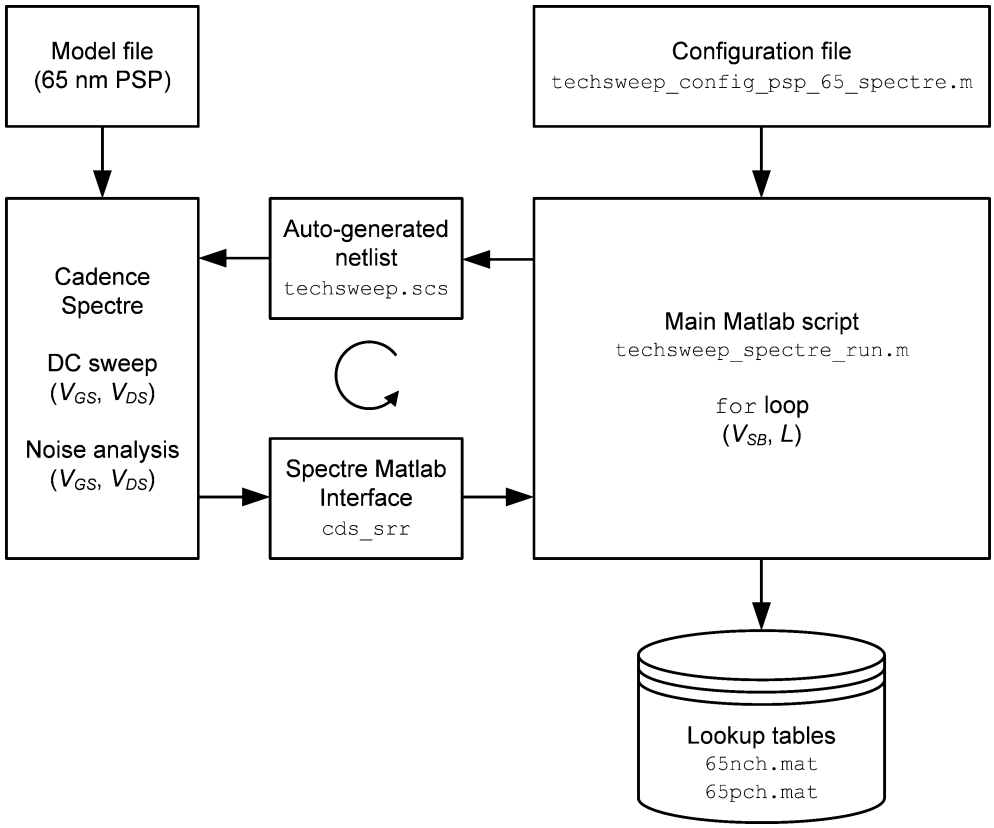
Matlab Scripts for g_m/I_D Lookup-Table Generation

Rev. 20170917
Boris Murmann,
Stanford University
murmann@stanford.edu

The scripts provided in this package store DC-sweep simulation data into Matlab files (.mat). These "lookup tables" can then be used for systematic circuit design in Matlab, as described in the book listed below. The book's companion website provides Matlab scripts for the design examples of each chapter, lookup tables for a 65-nm process, and a "lookup" function for convenient read-in and data interpolation. The scripts provided in this package are for those who want to generate their own lookup tables. Please refer to Appendix 2 of our book for a more detailed description of the lookup table generation flow (summarized in the figure below).

P.G.A. Jespers and B. Murmann, Systematic Design of Analog CMOS Circuits: Using Pre-Computed Lookup Tables, Cambridge University Press, 2017.
https://www.cambridge.org/core/books/systematic-design-of-analog-cmos-circuits/A07A705132E9DE52749F65EB63565CE0



What You Need

- A model file for the CMOS technology that you want to characterize. For example, a set of PSP, BSIM3, BSIM4 or BSIM6 model cards.
- A circuit simulator: Spectre (Cadence) or HSpice (Synopsys)
- A function for reading simulation data in into Matlab (pick one of these two options)
    - For Spectre: cds_srr. This function is part of the Cadence Spectre/RF MATLAB Toolbox, see "Virtuoso Spectre Circuit Simulator RF Theory Manual." You must ensure that your Matlab versions and MMSIM versions are compatible.
    - For HSpice: loadsig. This function is included in Michael Perrott's HSpice Toolbox for Matlab and Octave, http://www.cppsim.com/download_hspice_tools.html.
- It is assumed that you launch the provided code within a Matlab instance running in the same Unix/Linux environment as your simulator (to enable direct file exchange).
- It is assumed that you have some experience with setting up EDA tools. The provided scripts are far from "plug and play." You need to look at each line in the header of the configuration file and determine appropriate adjustments for your system.

File Descriptions

techsweep_spectre_run.m
techsweep_hspice_run.m
These are the top-level scripts that execute the lookup table generation. You should not have to edit these files, except for specifying the proper configuration file near the top.

techsweep_config_psp_65_spectre.m
techsweep_config_bsim4_28_spectre.m
techsweep_config_bsimcmg_16_spectre.m
techsweep_config_bsim3_180_hspice.m
These are example configuration files that are called by the top-level scripts. The "psp_65" file is what we used to generate the data files distributed with the book (65nch.mat and 65pch.mat). The other files are provided as examples for your convenience and can be edited to work with your own BSIM3, BSIM4 or BSIM-CMGmodels.

The configuration files define all tool and file paths and contain a netlist template near the bottom. This is where you define the content of the auto-generated simulation netlist (see above figure). Also, the configuration file defines how the raw simulation parameters are mapped into Matlab variables (for example: CGG = cgs + cgd + cdb).

It typically takes some time to get all the paths and settings right (especially for cds_srr). In our experience, things are more easily debugged when using HSpice & loadsig.

techsweep_spectre_debug.m
techsweep_hspice_debug.m
This is a utility script designed for debugging the configuration files. Basically, this is a simplified version of techsweep_*_run.m files that runs just one single simulation sweep and displays the parameters at mid-supply. Once this works, the full table generation can be kicked off and should work without any problems.