Drain Current Calculation using BSIM4 Model Equations

Robert Rice Brandt[1], Vincent Patrick Marie Bourguet, Raimundo Carlos Silvério Freire

[1]Universidade Federal de Campina Grande - UFCG, Campina Grande, Paraíba, Brazil, brandt.rr@gmail.com

Abstract: In this paper we describe the development of a C++ routine (and several auxiliary subroutines) to calculate the drain current of a MOSFET, based on the BSIM4V6 model equations from Berkeley University.

Keywords: BSIM4, MOSFET, current, C++, IP.

1. INTRODUCTION

In the modern world, Integrated Circuits (ICs) are becoming more and more commom place. As in many fields of knowledge, the design of new ICs depends greatly upon the experience aquired during the design of the previous versions. Most modern Integrated Circuits are extremely complex. They are called SoC (System On a Chip), because of the various modules that are joined together in the making of the ICs or chips. See figure 1.

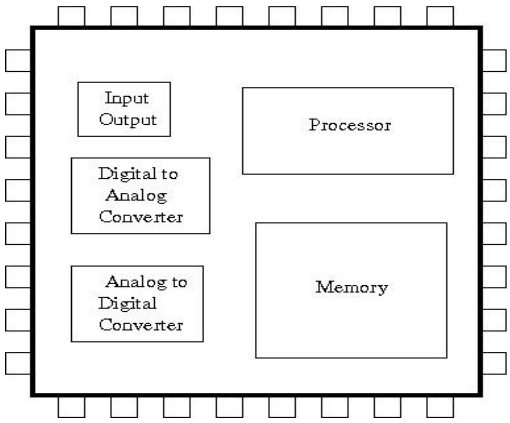

Fig. 1. - System on a Chip

In the realm of IC creation the term IP (Intelectual Property) has come to stand for “a reusable module” that can be joined with others to make a larger system. It can be a characterized layout, or even program code that allows to synthesize a characterized layout from an higher abstraction level. IPs can be bought or sold, and mixed with other IPs to build a complete system.

In the digital world, the definition and the synthesis of IPs is already pretty much run of the mill. The abstraction level, and hence the reuse rate, is very high. The transistors are mostly minimals. Most logic gates are simple functions, and are ready for use in new projects.

In the analog world, there are many more options and parameters needed to specify each block. The current in each individual transistor can vary enormously, which in turn varies its size. Parasitics also need to be taken into consideration. These factors, coupled with the fact that a change in the board layout can change the operating caracteristics and thus the funtionality of the chip, make analogue IP synthesis extremely complex.

New methods are constantly being developed to automate the analog IP synthesis. For each project, many parameters need to be taken into account. With the development of better automation tools, the time and cost needed to produce new ICs is greatly reduced. These methods can be classified into two broad categories: simulation based, and knowledge based.

- The **simulation based** analog IC design automation usually uses complex models which in turn produce very reliable results [1]. The major drawback is the time needed for several iterations through the simulation.
- The **knowledge based** IC design automation usually uses very elaborate design procedures, and is very fast [2,3]. But usually uses very simple models and couples that with the knowledge and experience of the designer.

Our paper is organized as follows: Section 2 presents an introduction to the BSIM Model. Section 3 briefly presents the CAIRO+ approach. The remaining sections describe a program under development to read the BSIM4 model parameters from a SPICE file and compute the drain current depending on transistor size, number of fingers (M), and bias (L, W, M, vgs, vds, and Temperature).

2. BSIM MODEL

Large signal transistor models fall into three main types: physical models, empirical models and tabular models.

Physical models are based on the physics of the transistor. They are based on the modeling of physical phenomena within a transistor. These models have become extremely complex and are not suited for circuit simulation.

Empirical models are based solely on curve fitting. The parameters in these models do not need to have any

fundamental basis, and will depend on the fitting procedure used.

Tabular models use look-up tables with a large number of values for the common transistor parameters, such as current and parasitics. Increasing the data points in the table increases the likelyhood of finding the values needed for a given transitor. A limitation of these models is that they work best for designs that use devices within the table (interpolation) and are not reliable for devices outside the table (extrapolation).

Researchers at the **University of California, Berkeley** have written a set of IGFET (**insulated-gate field-effect transistor**) transistor model equations for use in IC design. This set of model equations is called BSIM (Berkeley Short-channel IGFET Model), and has been in widespread use since the release of version 3.3 in September of 1995. Version BSIM4.6.3 was released in September of 2008. Improvements have incorporated many new parameters, taking into account several factors present in the manufacturing process, such as: stress effect, noise, temperature dependence, and component placement and dimensioning. The model currently in use has over 400 distinct parameters.

Over time, the BSIM3v3 model equations became an industry reference. Other models were measured in comparison to the BSIM3v3 model. Now that the BSIM4 version has been released, programmers that had used BSIM3V3 are modifying their code (or rewriting it entirely) to take into account all the new parameters of BSIM4. New programs should not be written based on the BSIM3V3 code. Instead, they should be written using either the Berkeley BSIM4 model or the Penn State Philips (*PSP*) transistor model.

3. CAIRO

CAIRO+ [3] is an environment for analog IC design automation developed by the LIP6 laboratory in Paris, France. It allows one to define and to synthesize analog IPs at the circuit level, where the block's netlist is known.

CAIRO+ is a French acronym that stands for “Circuits Analogiques Intégrés Réutilisables et Optimisés”. It is being implemented as a documented superset of C functions.

To that matter, CAIRO+ offers the following features to the designer:

- An analog description language suited to the writing of accurate sizing procedures;
- An access to integrated Bsim3v3 model equations;
- A library of generic basic device generators [4]. These generators can produce an optimized analog layout from device specifications and a design rules file. They take into account matching constraints and perform parasitics caracterization.

CAIRO+ also includes a privileged communication flow between the sizing engine and the layout engine. Thus, the number of sizing-layout iterations needed to reach specifications is significantly reduced. Moreover, thanks to the integrated model equations, the iterative process is faster.

The work presented in this paper aims to provide CAIRO+ with the BSIM4 model equations so that submicronic effects like stress effect can be taken into account.

4. OBJECTIVES

The BSIM4 Model is very complex. The code furnished by Berkeley is composed of many subroutines to calculate dozens of different parameters, based on the values of several hundred others. Our main objective is to extract only the sections of interest in the calculation of the drain current from the many routines offered by Berkeley, leaving out the sections relative to other parameters.

In writing the subroutines, we also saw the need to create routines to read the general SPICE model parameters, to read the specific transistor parameters, and to save the results to a file. The specific transistor parameters are L, W, Temperature, Vgs, Vds, Vbs, and Number of Fingers (M). All of these basic features can be controlled by command line parameters of the main program.

The program was suggested by the research team in France that is working on the CAIRO+ project. In the future it is expected that the subroutines used in this program will be added to the CAIRO+ programs under development at Paris 6 University in France.

5. METHODOLOGY

Our first step was to examine a similar project [5] by a French researcher called Laurent de Lamarre. His work was based on the BSIM3V3 model. We saw how he parsed the model parameter file, and how he saved the information in a structure in memory. Examining his work made it very clear the level of difficulty we would face when working on the BSIM4 model.

Our second step was to examine the BSIM4V6 code furnished by the Berkeley University team. This code was taken from their public web site [6]. We extracted the lines of code needed for the particular case of our project. We found no specific input or output routines, because these tend to be specific for each application, so we had to write these routines, also.

We decided to divide the main part of our code into small routines to calculate intermediate results. In each part where we need a specific parameter, we test to see if it has been furnished, or previously calculated. If so, we use that

value. If not, we calculate the value, then use it. Example code:

if(not struct->variable.set) calculate_variable(struct);

local_variable = struct->variable.val;

We also decided to use four different structures in memory for the different parameter values we needed. The first structure holds the model parameters, read and parsed directly from the SPICE file. See figure 2. The second structure holds the specific transistor (instance) data: temperature, voltages, and dimensions. The third structure is calculated from the first two as in figure 3.

The fourth structure holds the final calculated parameters. See figure 4. For the fourth structure, every parameter has a flag to indicate if the value has already been calculated. If it has been calculated, the value can be used directly. If the flag indicates that the value has not been calculated, it is first calculated, the flag is set, and the value is then used.

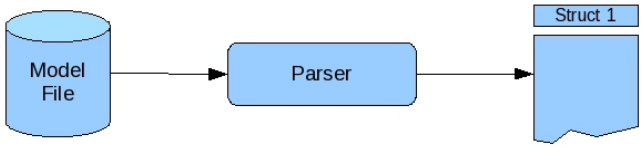

Fig. 2. - Parsing and initializing structure 1

We created a subroutine to parse the model parameter file, using standard yacc and lex. This routine reads the SPICE model parameters from a file and fills our first structure in memory. This subroutine is very similar to the one that was used by LAMARRE in his BSIM3V3 version program.

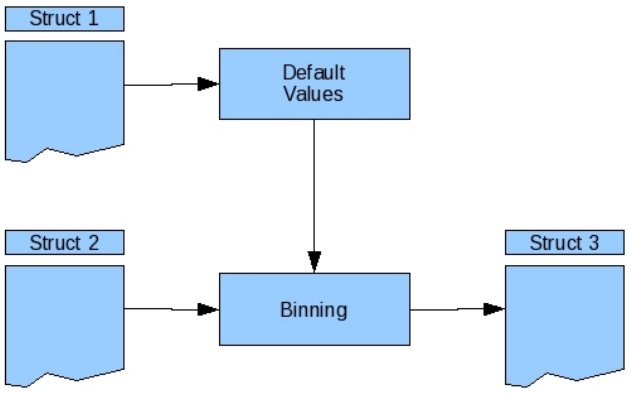

Fig. 3. - Check default values and calculate binning

These parameters are used to calculate our intermediate values and the results are stored in the third structure. In these calculations binning is also used to adjust some of the parameters. Binning is a technique used to make small adjustments in the calculated values of some parameters, based on the transistor dimensions.

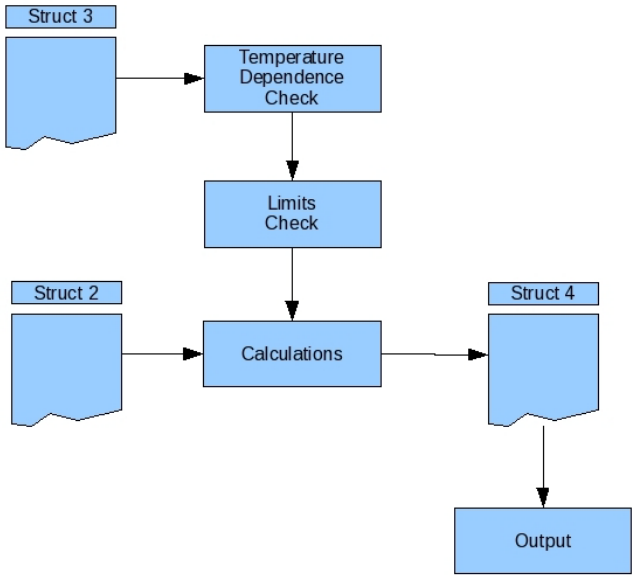

Fig. 4. - Calculate limits and output

These values, along with the specific transistor parameters (dimensions, voltages, and temperature, from the second structure) are used in the drain current calculations.

After all calculations are finished, the results are saved in an output file. This file name is specified on the command line, when the program is run.

6. RESULTS

After writing the programs and routines, we used the programs to calculate the current on a specific model, and then we compared the values with the results of the same model, calculated with Eldo. At the time of this writing it seems we still have a bug in the final Ids calculations. More time will be needed to find our error and correct it.

But, we also compared a few intermediate parameter values, that can easily be calculated in the “Eldo” program, by Mentor Graphics[7]. In table 1 we show some intermediate results using these values: Vds = 2.5V, Vgs = 1.5V, Vbs = 0.0V, Temperature = 27°C, W = 5µm, L = 0.5µm, M = 1.

Table 1. Typical Results

Variable	Eldo value	Our value
VTH	.2795162	.279516
PHI	.8934560	.893456
UEFF	6.1107m	6.11058m

7. CONCLUSION

BSIM4 is a useful base for making MOSFET transistor models. Extracting parts of the code can be helpful to calculate a few of the most used parameters. Work on BSIM is an ongoing project, and can render many subprograms.

The various routines we wrote in "C" to read the various MOSFET transistor parameters and calculate the value of the drain current and write the results in a file, gave us very satisfactory values.

Incorporating our BSIM4 based program into the CAIRO+ environment will permit design iterations to be faster since simulation is not further necessary in the loop. However, because of the quality of the model used, sizing accuracy is ensured.

AKNOWLEDGMENTS

I thank God for giving me life and health, and a wonderful family. I am also grateful for my tutors: Raimundo Carlos Silvério Freire, Benedito Antonio Luciano, and Vincent Patrick Marie Bourguet, for the incentive to work on this project.

REFERENCES

REFERENCES

[1] Gielen and Rutenbar, "Computer-Aided Design of Analog and Mixed-Signal Integrated Circuits", IEEE Proceedings, 88(12): 1825-1852, December 2000.

[2] R. Castro-Lopez et al., "Generation of Technology-Independent Retargetable Analog Blocks", Analog Integrated Circuits and Signal Processing, Kluwer Academic Publishers, 33: 157-170, 2002.

[3] M. Dessouky, "Design for Reuse of Analog Circuits. Case Study: Very Low-Voltage Sigma-Delta Modulator.", PhD Thesis, University of Paris 6, 2001.

[4] V.Bourguet et al., "A Layout-Educated Analog Design Flow", IEEE MWSCAS'04, Hiroshima, Japan, July 2004.

[5] Lamarre, Laurent, "Calculette MOS, Implémentation du modèle BSIM3V3", MSc thesis, University of Paris 6, 2002

[6] Official BSIM4 site: http://www-device.eecs.berkeley.edu/~bsim4/

[7] Official Mentor Graphics site: *http://www.mentor.com/*

Brandt, R. R. is a professor of the Electrical Engineering Department, Campina Grande Federal University in Campina Grande, PB, Brazil.