National University of Science and Technology POLITEHNICA Bucharest

Doctoral School of Electronics, Telecommunications and Information Technology

Decision no. from

PhD Thesis Abstract

Behavioral Modeling of Electronic Circuits

PhD student: **Eng. Laura-Alexandra GHEORGHE**

DOCTORAL COMMITTEE

Prof. Dr. Eng. Drăghici Florin Univ. Polytechnic University of Bucharest	President
Prof. Dr. Eng. Lidia DOBRESCU Univ. Polytechnic University of Bucharest	PhD supervisor
Prof. Dr. Ing. Univ. Polytechnic University of Bucharest	Reviewer
Prof. Dr. Ing. Univ. Polytechnic University of Bucharest	Reviewer
Dr. Ing.	Reviewer

BUCHAREST 2025

Introduction

The theme chosen for research within the Doctoral School is an incursion into a field of interest in the world of electronics because it addresses an innovative, efficient and fast method for the implementation and testing of electronic devices used in the semiconductor industry, and especially in the design of mixed analog-digital systems. Thus, the key word that defines this doctoral thesis is **digitalization** in the electronic field. As the title announces, the field of study is focused on **the modeling of electronic circuits**. The central reason why **behavioral modeling** is a significant branch of electronics is that it offers the possibility of emulating and testing both simple and complex applications or components, using different simulation programs. The main purpose is to evaluate the performance of an analog and/or digital circuit before it is manufactured.

In order to maintain an upward trend of technological progress, several conditions must be met: the **rapid realization** and **supply** of integrated circuits for customers, **the reduction** of manufacturing costs, **innovation** – the implementation of new functionalities meant to offer more benefits to users. At the same time, both on a large scale and in the automotive industry (most of the behavioral models studies presented in this thesis are dedicated to this industry) increasing the operating speed, switching to 48V supply voltages as well as reducing the amount of energy consumed are topics of great interest and represent a challenge for engineers.

As a definition, **behavioral modeling** is an abstraction of electronic circuits without using information related to the technology used or physical implementation, providing a system-level approach to their analysis and synthesis.

Behavioral modeling is based on two fundamental principles: defining inputs and outputs, describing the relationship between them, so that a certain functionality can be achieved. As will be shown in this thesis, the basic elements used in behavioral description are: controlled sources, dedicated SPICE functions.

By modeling the behavior of individual components and the interconnections between them, engineers can simulate their operation, or that of an **integrated circuit** or **system application**, before making a physical prototype. Also, simulation models can be used for conceptual verification but also for defining a product.

Basically, a behavioral model can be made both for a simple passive component and for complex systems. The difference between behavioral and device/structural modeling is that behavioral modeling is not based on mathematical equations that describe the functionality of a component (such as the exponential characteristic of a diode), but on abstracting its functionality, so that with a minimum set of lines of code the basic function is described, this method being useful to reduce the overall duration of a simulation.

This thesis will address various behavioral models, from simple to complex, starting with device modeling, then modeling integrated circuits, and finally addressing a system application.

The modeling process involves extracting the relevant information from the component data sheets and making them in a format compatible with simulation tools. The resulting

models can then be used to analyze circuit performance under different operating conditions, as well as optimize design parameters.

Any model is developed based on a programming language specific to each simulator. In this thesis, the models presented are described both in **SPICE language**, which allows the description of the analog part, and **in MATLAB/Simulink** language, in order to be able to combine the analog and digital parts and describe a system behaviorally.

The advantages and limitations of behavioral modeling will be presented below in order to define this field as clearly as possible.

Advantages of behavior modeling:

- Abstraction of the device/circuit/application using different methods (which generally do not take into account technology) and/or mathematical equations in order to observe its behavior;
- Significant reduction of simulation and implementation time;
- Merging the analog and digital parts (e.g. communication between a microcontroller and a peripheral via a serial communication interface);
- Promotion of the device in the design/production phase;
- Development or transfer of the model to any analog simulator;
- Identifying problems that may arise with a product before its physical realization;
- The ease of modifying a behavioral model – at the code level;
- Possibility to perform and simulate an unlimited number of scenarios/system tests;
- Evaluating and improving certain functionalities before the device is physically realized;
- Testing a behavioral model - nothing can be damaged compared to the verification of integrated circuits in the laboratory;

The limitations of behavioral modeling can be:

- The cost of licenses of different simulators;
- Simulator convergence errors;
- Lower accuracy compared to device-level modeling;
- Product idealization;
- Implementation of certain digital circuits in SPICE language.

The novelty of this scientific research lies in determining modeling methods so that the limitations of behavioral modeling are overcome – the integration between the analog and digital parts. In addition, the thesis also has a **didactic purpose**, that of defining an application platform for students supported within **the MACAD** discipline ("Advanced Modeling of Digital Analog Circuits") that is part of the master's degree program entitled ***Microelectronics and Nanoelectronics*** supporting this trend of digitalization, because the virtual environment of simulation is mature, sustainable and can be achieved from anywhere, not depending on physical presence.

The thesis will comprise eight chapters, presenting behavioral models for different branches of electronics (polarization circuits, power electronics, frequency conversion circuits for audio systems) and will address the following topics.

The first chapter is the introductory part and aims to explain the theme addressed in this thesis as well as the motivation and current state in terms of behavioral modeling.

Chapter 2 presents an introduction to computer-aided design of electronic circuits. The programs (PSpice and MATLAB/Simulink) used in the development of behavioral models will be presented.

Chapter 3 presents a method for optimizing the Widlar current reference using bipolar transistor and MOS models, basically presenting the modeling method for circuits that provide small currents. In this chapter, two techniques for reducing the total area for a Widlar current reference based on the sum of resistances, using both bipolar transistors and MOS, will be presented.

Chapter 4 also presents an optimization method, this time for an analog computational circuit (Gilbert Cell) that can perform mathematical operations in order to minimize computational error, based on two implementations, using SPICE models of bipolar transistors and nMOS.

Chapter 5 introduces the behavioral patterns of electronic components (passive and active), of some fundamental building blocks and explains how they work. The current state of behavioral modeling is also presented.

Chapter 6 presents behavioral modeling methods for an intelligent power switch with different protections: overcurrent, overtemperature, overvoltage. The validation of the behavioral model was performed using the worksheet.

Chapter 7 presents the BLDC motor behavioral model and its command behavioral model in SPICE language. The simulation results will be compared with the laboratory measurements. This chapter discusses behavioral models with high current capability.

Chapter 8 addresses system modeling, describes the serial communication interface (SPI) behavioral model developed in MATLAB/Simulink and the development of a system application that will contain an abstract microcontroller model, the SPI model and an intelligent power switch model (with different protections) in "*High-Side*" configuration. Moreover, the topic of co-simulation between a simulator with an analog engine and the SystemC language will also be addressed.

Widlar current reference optimization implemented using nMOS and bipolar transistor models

Comparison between the nMOS LEVEL1 model and the PTM model

Starting from the premise that production processes encounter difficulties as the scaling of integrated circuits tends towards nanometric dimensions in which, operating close to the thermal limit, the devices are sensitive to noise disturbances, the Widlar current reference implemented with predictive transistor models was performed, implemented and studied. Specific to these models is the very large number of parameters of the simulation model (in the order of hundreds), used to describe as accurately as possible the new effects of the MOS transistor.

The latest MOS models that emulate a lot of parasitic effects are the BSIM3 or BSIM4 versions generated by Berkeley. The BSIM4 model derived from the BSIM3 model has about 130 parameters, so about 18 times more electrical parameters than the classic MOS model [11]. These models guarantee sufficient bias current values dedicated to different circuits in different industries. The supply voltage is around 1V. The use of a low supply voltage is an advantage in terms of power dissipation.

The PTM models are chosen because the technological node is becoming smaller and smaller and the short-channel effects such as: *velocity saturation*, *threshold voltage variations* must be taken into account. New short-channel effects occur as its length becomes smaller.

Over time, approximately 64 different levels of MOSFET models used with HSPICE have been developed by Synopsys [12].

PTM (Predictive Technology Model) [13] provides libraries of accurate models, predictive and interconnection technologies. In this work, a **BSIM4 PTM HP NMOS level 54** model developed in **45nm** technology, known as High Performance 45nm Metal Gate, High-K/Strained-Si [14], is used.

On the other hand, if the analog part does not need to be emulated in detail, LEVEL 1 transistor models can be used (especially for simulating large digital circuits). They offer a high level of accuracy for measuring transition times but also reduced simulation time. For more accuracy, the LEVEL 6 model or one of the BSIM models (LEVEL 13, 28, 39, 47, 49, 53, 54, 57, 59 and 60) can be used for very faithful modeling.

The MOS LEVEL 1 transistor model can be used for devices with a gate length greater than 10μm (long channel), and the output size is basically a first-order approximation.

The Widlar current reference has been implemented with both Level 1 transistor models and BSIM4 predictable models. This type of analog circuit is generally used to provide an accurate, low-value output current (hundreds of μA), and PTM models are very suitable to achieve this characteristic, as shown in figure 3.1. In classical implementations, the current copy ratio is given by the multiplicity factor and the pairing of the two nMOS transistors. In the current study, there is no multiplicity factor, but it has been observed experimentally: the output current for the PTM implementation is better copied than the output current from the classical implementation.

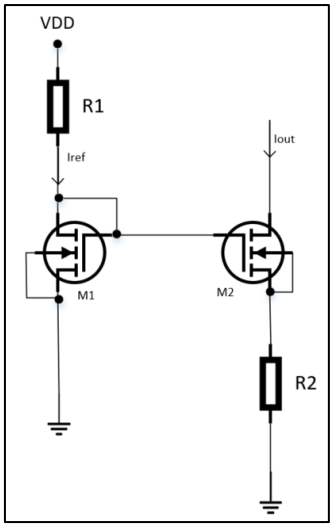

Figure 3.1 *Widlar Current Reference*

For testing, the transient simulation including Monte Carlo analysis was chosen.

The threshold voltage varies with a tolerance of ±5% for the classic transistor model. During the Monte Carlo simulation, the average output current value is calculated in the SPICE Error Log file for each point. It ranges from the minimum value of 44.5µA to the maximum value of 45.88 µA. The same study was carried out using advanced predictive MOS models. The simulation also contains 100 runs. The threshold voltage specific to this type of model is varied in the simulation with a tolerance of ±5%. The output current values range from 7.61µA to 8.27µA.

Figure 3.1.4 also shows the histogram of the output current using the PTM model for 250 runs, demonstrating that the values are gathered around the mean.

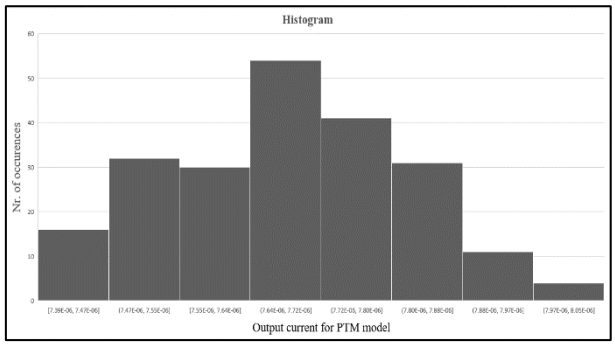

Figure 3.1.4 *Output current for PTM model: Vt0 = {mc(0.46893, tol)}, Vdd=1V, tol=0.05*

Optimization of the current reference implemented with MOS transistors

For the Widlar current reference implemented with both bipolar transistors and MOS, a mathematical study was carried out that aims to reduce the occupied area in integrated circuits.

As an example, after several mathematical calculations, for determining the values of the two resistances, the following formulas can be considered:

The output current sizing equation is shown below:

$$I_0 = \frac{1}{R2}\frac{kT}{q} ln \frac{I_{REF}}{I_O} \quad (3.3.6)$$

The reference resistor can be calculated from Kirchhoff's second law:

$$R_1 = \frac{V_{CC}-V_{BE}}{I_{REF}} \tag{3.3.7}$$

In order to minimize the area of silicon occupied by the resistors of the circuit, the expression of resistance R2 in equation 3.3.7 shall be substituted into the following function given in equation 3.3.8:

$$f(R) = R_1 + R_2 = R_1 + \frac{kT}{qI_O} ln \frac{I_{REF}}{I_O} \tag{3.3.8}$$

Substituting the reference current, the expression becomes:

$$f(R) = R_1 + \frac{kT}{qI_O} ln \frac{V_{CC}-V_{BE}}{R_1 \cdot I_O} = R_1 + \frac{kT}{qI_O} [ln \frac{V_{CC}-V_{BE}}{I_O} - lnR_1] \tag{3.3.9}$$

The minimum function is obtained when the derivative of the function is 0:

$$f'(R) = 0 \Rightarrow 1 - \frac{kT}{qI_O} \cdot \frac{1}{R_1} = 0 \tag{3.3.10}$$

Next, the resistance R1 will be dimensioned:

$$R_1 = \frac{kT}{qI_O} \tag{3.3.11}$$

For a desired output current of Io=1μA, taking into account kT/q=25.85mV, the value for $R_{1=25.85k\Omega}$. The value for R2 can be calculated from equation 3.3.8 with IREF calculated from equation 3.3.7 at a defined value for VBE and a supply voltage, VDC equal to 5V. Thus, the optimal value for R2 is 129.21 kΩ.

The optimized values of the resistors are used in a simulation, as shown in Figure 3.3.2, where an additional study on the temperature dependence of the resistor was performed.

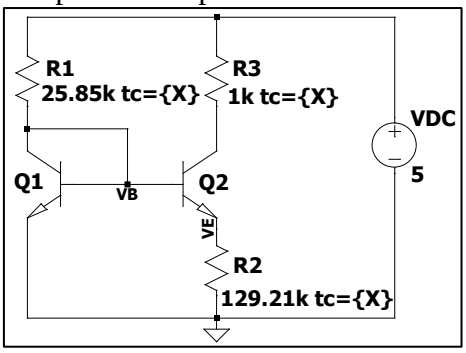

Figure 3.3.2 *Widlar current reference implemented with bipolar transistors with optimized resistances*

Optimization of an analog computing circuit using bipolar transistor and MOS models

Digital multiplication is a standard today due to the ease of design using number blocks or general-purpose processors. Analog multiplication is an alternative to ordinary digital calculations, in applications where a very high precision of more than 32 bits is not required [22, 23]. Analog multiplication has certain advantages: low power consumption, real-time operation and reduced number of devices, in contrast to the billions of transistors that perform multiplication in several clock cycles used in digital deployments.

A well-known analog multiplier circuit is the Gilbert cell [24, 25], which is widely used in mixed-signal circuits such as frequency mixers and mixers in radio frequency telecommunications. Although initially analog multiplication circuits used bipolar transistors, today modern MOS technology can also be used. The circuit shown in Figure 4.1.1 is derived from the Gilbert cell and is intended to perform analog multiplication calculations.

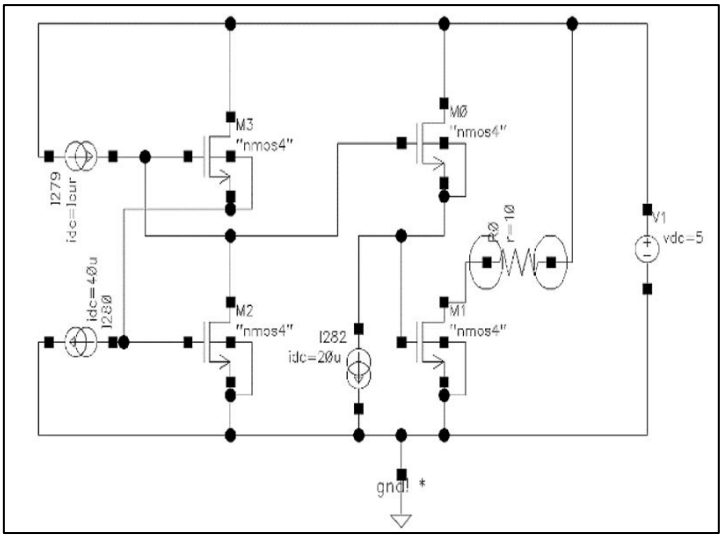

Figure 4.1.1 *Compute cell using MOS transistors*

The circuit operates considering the exponential dependence of the current on the control voltage, which means that the MOSFET transistors should operate in the respective region of the current-voltage characteristics, in the sub-threshold regime [8-11]. This is advantageous because both low currents and a low supply voltage imply low power consumption of the entire circuit.

The MOS field-effect transistor has two main operating regions: the sub-threshold region [26], where the gate voltage is lower than the threshold voltage (VT), and the above region, where the inversion in the channel occurs. The current-voltage dependence is represented in Figure **Error! No text of specified style in document.**.2, on a logarithmic scale for current. In the region below the threshold, the exponential dependence of the current on the gate voltage can be noted, being a straight line in this representation of the current. This allows a range of

currents to be selected for the correct operation of the analog multiplier circuit. The range of optimal current values is between 100pA and 1mA.

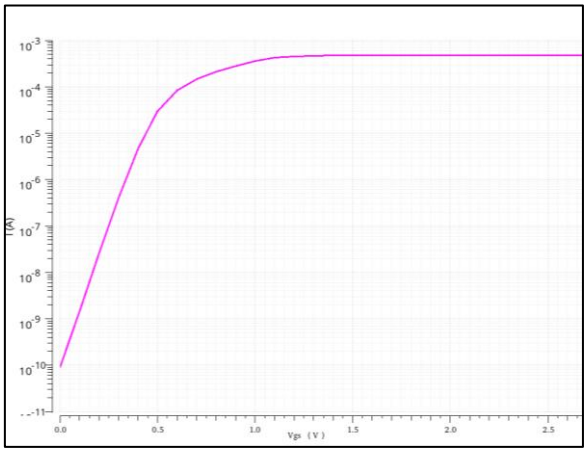

Figure 4.1.2 The current-voltage characteristic of a MOSFET simulated in Virtuoso

The output current of the multiplier circuit was simulated in 150 nm technology using the Virtuoso program [27].

The VT threshold voltage of the MOSFETs was varied in a Monte Carlo simulation with a nominal value of 1.6 volts and a standard deviation of 10%. Figure 4.1.4 shows the variation in the output current of the circuit due to these variations in the threshold voltage. Most values are within a range of less than 0.2 µA, resulting in a relative error of less than 0.1%. The temperature was varied using the Virtuoso program between -40°C and 125°C.

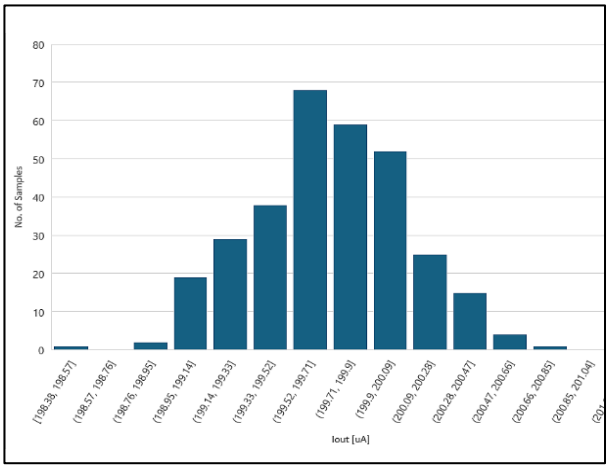

Figure 4.1.4 *Effect of threshold voltage variation on output current*

The variation with temperature showed that in a temperature range between 25ºC and 50ºC, the error is less than 1µA, or 0.5% the relative value of the error. As previously shown in Figure 4.1.4, for output current values less than 200 µA, e.g. in the current range of nA, the error is even smaller. The supply voltage of the multiplier circuit is equal to the sum V_{DS} and VGS in Figure 4.1.1. Since the devices operate in the region below the threshold [28], for example with VT = 1.6V, the supply voltage can be up to 3.3 volts.

The variation of the output current with the supply voltage was also studied and it was observed that for a lower threshold voltage VT, for example 0.5 V, the supply voltage has a minimal error and should be reduced to about 3 volts or less.

As a conclusion of this part, an analog multiplication circuit was studied and simulated, using the Virtuoso Studio program, in order to be able to make real-time calculations with a small number of devices. The circuit multiplies the input currents, and the output is also a current, obtained by dividing the product by a reference current of the circuit. MOSFET transistors operate in the region below the threshold, allowing for low currents and therefore low power consumption. The circuit has good output current linearity and low variations with parameters such as threshold voltage, temperature, and supply voltage, especially at low current levels. The output current error is less than 0.5% in the operating temperature range and less than 0.1% due to the technological variation of the threshold voltage of the MOS transistor.

Implementation of the multiplication circuit model

Figure 4.2.1 shows the implementation based on bipolar transistors of a computational circuit derived from the Gilbert computation cell used in the realization of the behavioral model [29,30].

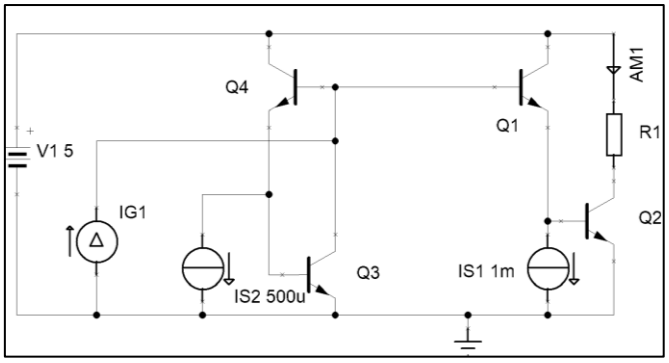

Figure 4.2.1 *Compute cell using bipolar transistors*

For the circuit in Figure 4.2.1, the sum of the emitter base voltages: VBE1 and VBE2 is equal to the sum of VBE3 and VBE4. This results in the multiplication of the IG1 and IS2 currents, according to the following calculations:

$$V_{be1} + V_{be2} = V_{th} * \ln\left(\frac{I_{C1}}{I_{C2}}\right) + V_{th} * \ln\left(\frac{I_{C2}}{I_{C1}}\right) = V_{th} * \ln\left(\frac{I_{C1} * I_{C2}}{I_S^2}\right) \quad (4.2.1)$$

$$V_{be1} + V_{be2} = V_{be3} + V_{be4} = V_{th} * V_{th} * \ln\left(\frac{I_{C3} * I_{C4}}{I_S^2}\right) \quad (4.2.2)$$

$$I_{C2} = \frac{I_{C3} * I_{C4}}{I_{C1}} = \frac{I_{G1} * I_{S2}}{I_{S1}} \quad (4.2.3)$$

where $V_{th} = kT/q$ is the thermal voltage, k is the Boltzmann constant, q is the elementary electric charge, and T is the absolute temperature. The emitter currents were approximated to the collector currents (the basic currents being neglected). Transistors are considered identical, having the same saturation current I_s.

The bias current IS1 (and for the current IS2 as a weight) is provided by a classic Widlar current reference.

Figure 4.2.5 shows the behavioral model implemented with MOS transistors

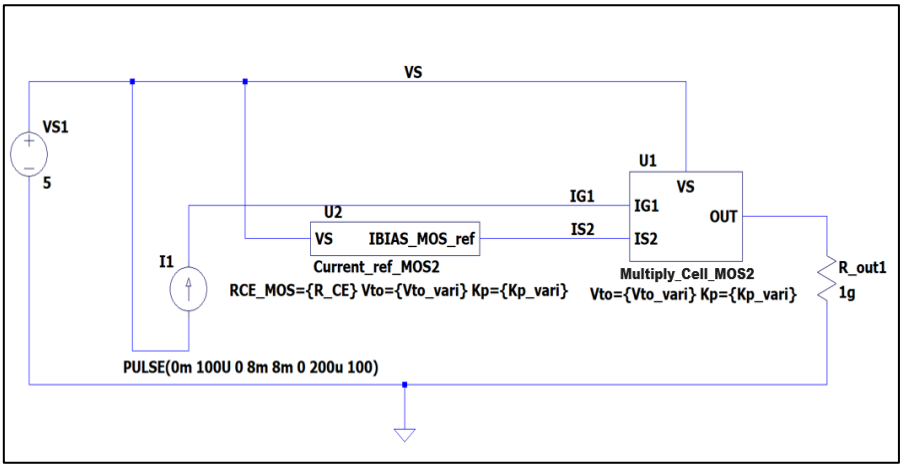

Figure 4.2.5 *Simulation model of current reference and multiplication cell [74]*

The simulation model proves the mathematical hypothesis presented at the beginning of this study, so the simulation results – the output current and the current formula are simulated and superimposed, as shown in Figure 4.2.7. The color black represents the calculation formula for the output current, and in blue the current is measured by the output resistance [74].

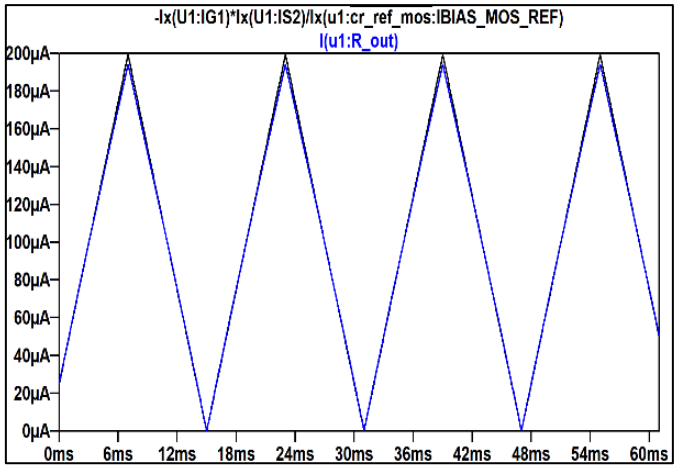

Figure 4.2.7 nMOS Results [74]

The same study was conducted using bipolar transistors. For both types of implementations, Monte Carlo simulations were run, the number of runs in 200, the tolerance of the various parameters being 10% and the varied parameters are presented in Table 4.3.1.

Table 4.3.1 *Varied parameters*

	Parameter	Definition	Default value
Model nMOS	Vth	Threshold voltage	1.6V
	Kp	Transconductance	1.1 uA/V^2
Bipolar model	IS	Saturation current	1fA
	β	Gain in current	75

Figure 4.3.5 shows a graphical comparison between the output current of the model for the multiplication circuit and the mathematical calculation, the results being close.

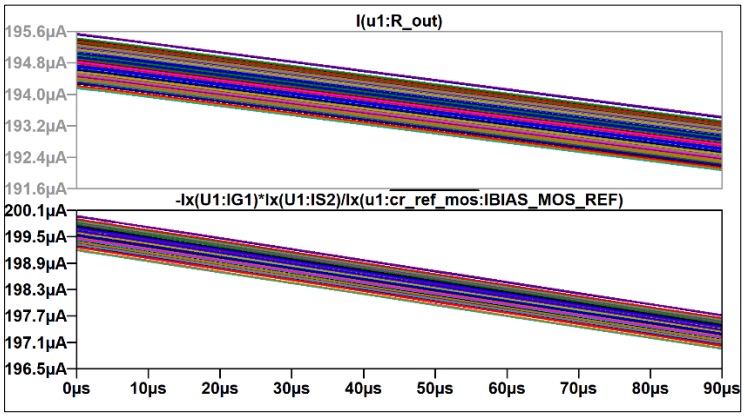

Figure 4.3.5 *Variation of the Kp parameter in Monte Carlo analysis – comparison between output current and mathematical calculation*

In Figure 4.3.5 it can be seen that the relative error for the output current is 0.72% for all variations of the Kp parameter.

Neural experiment based on the calculation cell model

The synapse of an artificial neural network performs the multiplication of the input signal from the previous layer (the arrows indicating the neurons of the hidden layer, for example) by the weight of that neuron. The mathematical model of the synapse is:

$$S_{out} = W_n S_{in} \qquad (4.4.1)$$

where, Sin is the signal received from the previous neuron, Wn is the weight of the neuron, and S_{out} is the signal delivered to the neuron. These three elements can be voltages or they can be currents. There are hardware implementations where the weight is the conductance of a resistor. The input is the IG1 current, and the weight is the IS2 current. The values for currents can be determined from Table 8.4.1.

Table 4.4.1 *Input current values*

	Network1	Network2	Network3	Network4	Network5
GI1	1µA	1µA	1µA	1µA	1µA
GI2	3µA	3µA	3µA	3µA	3µA
GI3	5µA	5µA	5µA	5µA	5µA
IS2	0.9µA	1.8 µA	3µA	3.95µA	4.853µA
IS1 (reference)	2µA	2µA	2µA	2µA	2µA

Therefore, experimentally, three values were chosen for the IG1, IG2, IG3 currents: 1µA, 3µA, 5µA. The two representations in Figure 4.4.2 are perfectly correlated, the three input currents being applied to the five neurons formed by three computing cells each [74].

Figure 4.4.1 shows only one neuron. The sum of the output currents shall be as close as possible to the sum of the three input currents, according to the formula shown in equation (8.2.3). Considering the values in Table 4.4.1, the experimental results are shown in Figure 4.4.2, where for a single neuron the sum of the currents provided by each calculation cell – represented by I(R1), I(R2) and I(R3) – were calculated, the currents were calculated using the formula from which the experiment started, then the results that are very close were superimposed.

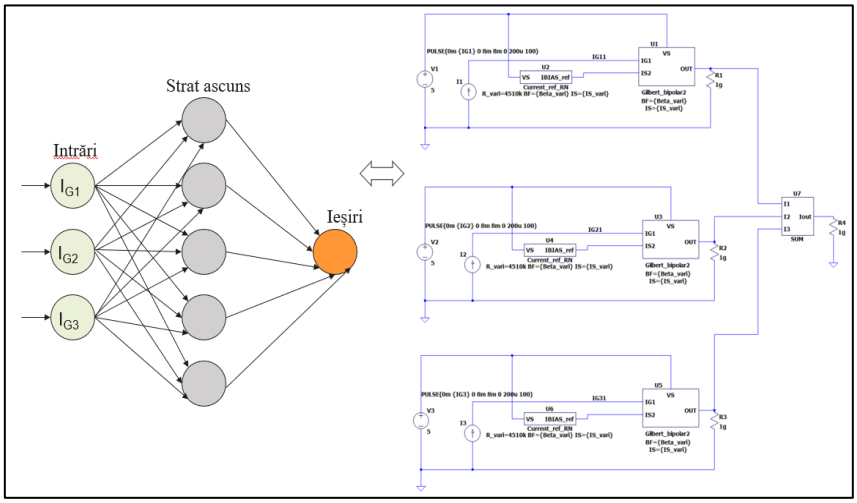

Figure 4.4.1*Neural network model*

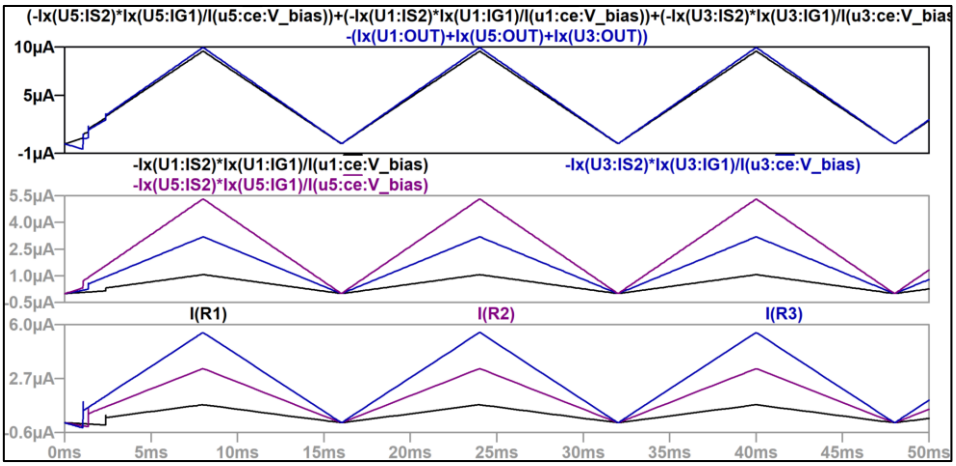

Figure 4.4.2 *The result of the network experiment*

The calculated relative error is less than 5%. Thus, it can be considered that the model of the computational cell used as part of a neuronal structure is robust, it can be easily integrated into such computational methods, in this case it can make multiplications in order to minimize computational error.

Description of the fundamental behavioral models

This chapter introduces the behavioral models of electronic components (passive and active), defining different basic implementation methods.

As modeling techniques, voltage- or current-controlled sources can be used to achieve that abstraction of an integrated circuit. Some analog circuits such as different types of op-amps can be described with the help of these sources. Using these types of components it is much easier to control the behavior of the output signals compared to using device models that are described based on complex mathematical equations, which most of the time hinder the overall simulation time.

Some types of abstraction will be shown below. Table 5.1.1 shows the comparison between the direct use of a resistor and its design using a current source controlled by the voltage at its terminals, divided by the value of a parameter that equals the resistor. Both transient analyses are parametric to obtain three current values (20mA, 25mA, 50mA) depending on the resistance values, according to Ohm's law. The output current values are identical and the simulation times are comparable.

Table 5.1.1 *Methods of modeling a resistance*

PSpice component (resistor)	Molded PSpice component (source G)
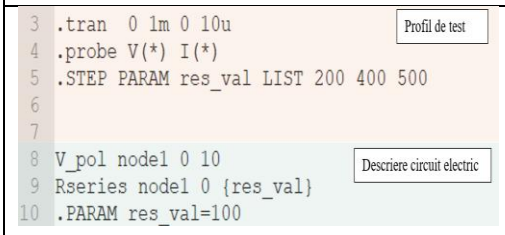	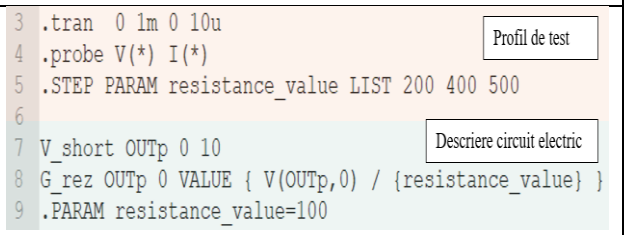
Figure 5.1.2 *Parametric simulation of the value of a resistance*	***Figure 5.1.3*** *Parametric simulation of the pattern of a resistance*
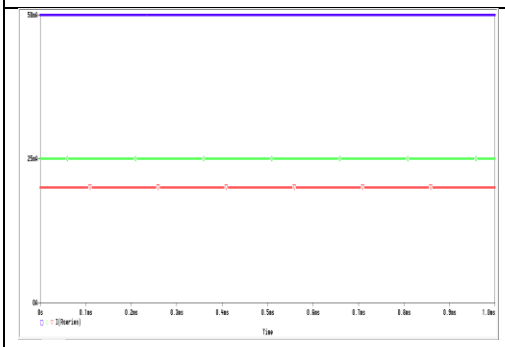	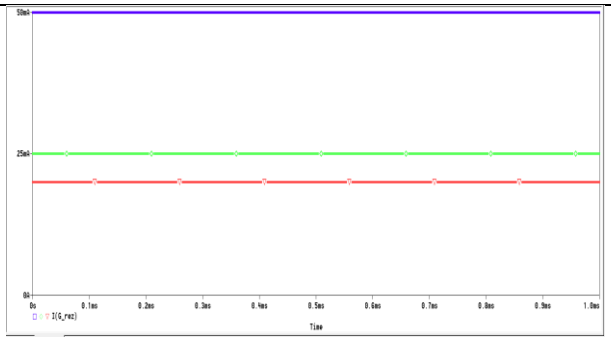
Figure 5.1.4 *Varied current values*	***Figure 5.1.5*** *Varying current values based on resistance model*

***Table** Error! No text of specified style in document..**1** Zener Diode Modeling*

PSpice component (Zener diode)	Molded PSpice component (source G)
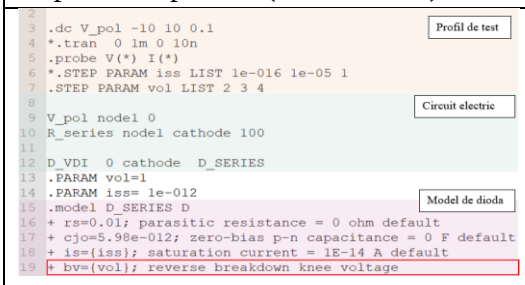	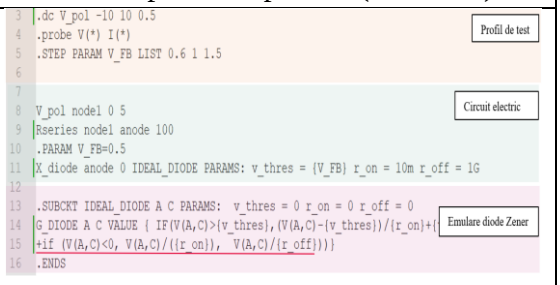
***Figure 5.1.11** The PSpice model of a Zener diode*	***Figure 5.1.12** Diode Behavioral Model*
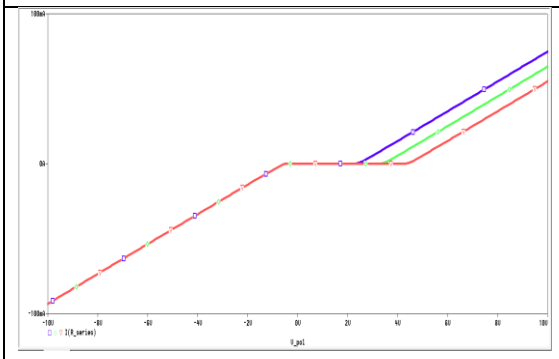	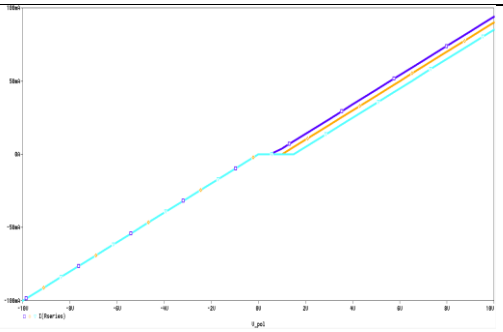
***Figure 5.1.13** Characteristic of a Zener diode*	***Figure 5.1.14** Parametric simulation of the Zener diode behavioral model*

Also in this chapter, types of delay models are defined. In this case, to achieve a delay on the decreasing front, a current source controlled by the voltage at its terminals was used to force the capacitor to charge quickly. In addition, an "OR" gate was used to form the output signal, then an "XOR" gate as illustrated in Figure 5.2.6 to achieve the delay only on the decreasing front of the input signal.

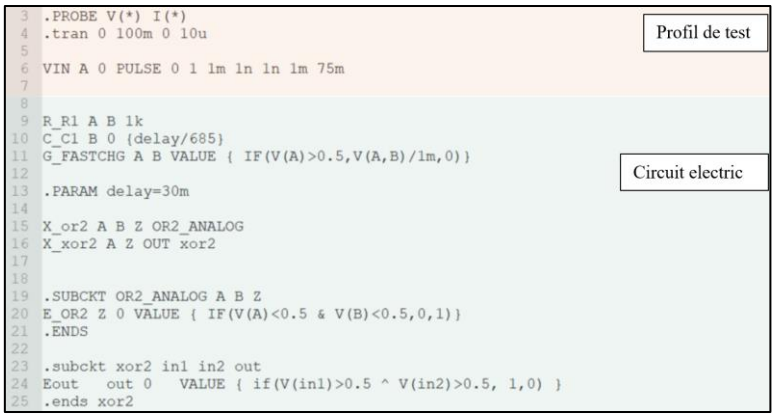

***Figure 5.2.6** RC delay pattern on decreasing front*

Figure 5.2.7 shows the step-by-step results, the charge current form of the capacitor, the preliminary result of the "OR" gate and the output signal delayed by 30ms.

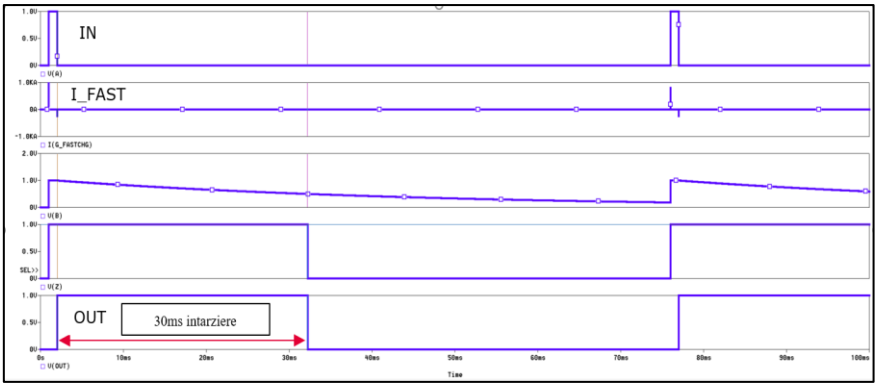

Figure 5.2.7 *Simulating the RC Delay Model on the Descending Front*

Another basic behavioral model is that of the hysteresis comparator that takes into account the input signal and the result of the positive output signal (compare the input signal with the higher threshold) in the formation of the negative signal and vice versa, thus creating a feedback loop, as shown in Figure 5.3.4.

Profil de test

Circuit electric

```
.probe V(*) I(*)
.tran 0 500u 0 10n

V_inp IN 0 pwl 0 0 50u 0 150u 100 250u 0


V_Vthh VTHH 0 dc 50 ; high threshold voltage
V_Vthl VTHL 0 dc 20 ; low threshold voltage

X_HYST IN VTHH VTHL OUTP OUTN 0 HYSTERESIS

**************************
*         HYSTERESIS     *
**************************
.SUBCKT HYSTERESIS IN VTHH VTHL OUTP OUTN GND

E_HYP OUTP 0 VALUE { IF(V(IN,GND)>=V(VTHH) | V(OUTN)<2.5,5,0)}
E_HYN OUTN 0 VALUE { IF(V(IN,GND)<=V(VTHL) | V(OUTP)<2.5,5,0)}

.ENDS HYSTERESIS
```

Figure 5.3.4 *Hysteresis Comparator Model*

From the point of view of simulation, in figure 5.3.5, one can see the ramp applied on the input signal in order to be able to check the imposed switching thresholds and the output signals (OUTP and OUTN) that switch first to the higher threshold and then to the lower one, this interval being known as the hysteresis window.

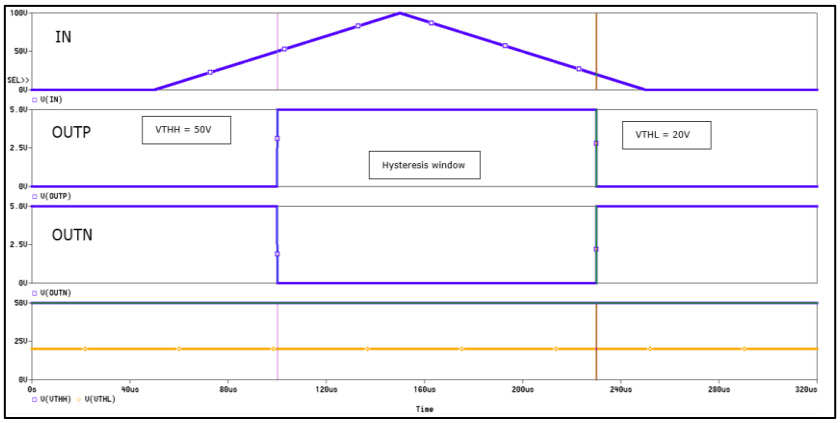

Figure 5.3.5 *Simulation of the hysteresis comparator model*

Modeling a Smart Power Switch

Smart switches have become more and more efficient products, are easier to use for switching electrical loads ON and OFF, provide better diagnostics and protection, are more reliable, and cost less compared to discrete solutions. Smart switches, consist of a power stage and control logic that, together with other blocks, provide advanced functionalities such as preventing damage caused by high temperature, overcurrent or overvoltage.

Figure 6.1 shows the block diagram of a classic smart power switch in "*Low-Side" configuration*, i.e. the load is placed between the power supply (Battery) and the output node (referred to as "OUT") [38]. In general, this type of device is used in automotive and industrial applications: **relays**, **fans** and **pumps**, **solenoids**, **valves**, **heating elements**, light bulbs and LEDs, power switches. The power block consists of the nMOS power transistor (which occupies the largest consumed area of the chip) which must be protected in the event of an undesirable failure event [39].

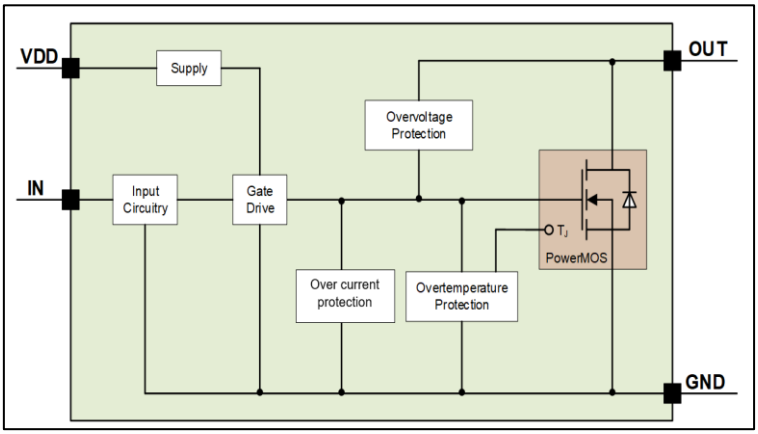

Figure 6.1 *Block diagram of a smart switch in "Low-Side" configuration [15]*

A PSpice model of a smart switch contains dedicated sub-circuits that describe each block specified in the datasheet and can be validated by developing scenarios, following the test conditions indicated. The command method for the nMOS transistor is modeled inside the block named "*Gate driver*"[34]. It is constructed of two voltage-controlled current sources that generate the corresponding electrical currents for the charging and discharging of CGS and CGD capacitors, thus obtaining the most representative parameters: **switching times**, as shown in Figure 6.1.7.

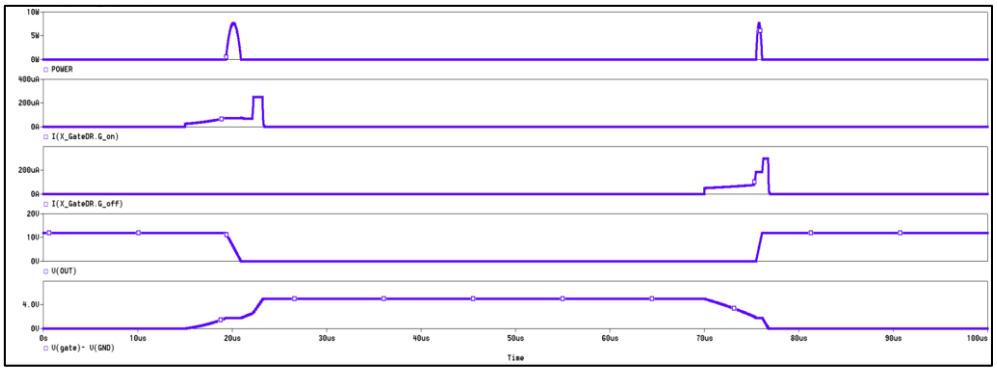

Figure 6.1.7 *Modeling switching times*

Next, the model of an overcurrent protection circuit will be presented. Overcurrent protections can occur due to short-circuit events or an overload event.

Figure 6.2.1 shows a simple method of designing an electrical circuit [42] with overcurrent protection: the nMOS transistor is used to switch the load during normal operation and overload conditions. To trigger the protection mechanism, the load current must be detected. This is done by monitoring the current through the shunt resistor, called R_{out}.

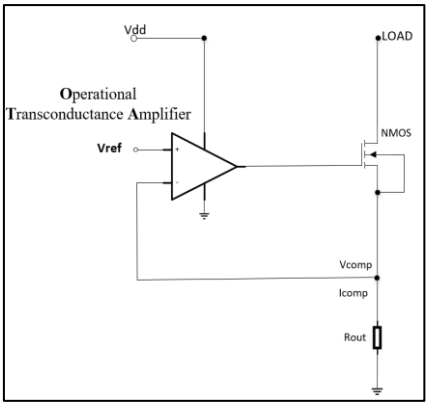

Figure 6.2.1 *Electric current value detection circuit [42]*

From the point of view of SPICE modeling, overcurrent protection can be implemented as in Figure 6.2.1 where, based on a simple comparator, the error signal is determined if the current limit has been reached, then the delay imposed by the data sheet is applied. The SR_FF instance is used as a "latch" model to not allow the circuit to start unless there is a reset signal.

1ms delay current limitation and auto reset: the current is limited as in the above case to a certain current value, and the device will be in the power-on state as long as the 1ms delay has not passed, as shown in Figure 6.2.7. After this delay, the device is turned off. If the input voltage is valid and the overcurrent event has passed, the device is automatically reset. The last waveform in Figure 6.2.7, monitors the temperature of the circuit. Thus, in this test scenario, if the input voltage ON/OFF is varied for a considerable period of time, the temperature will increase and in this way, apart from the overcurrent events illustrated, the overtemperature protection, described in the next subchapter, will also be achieved.

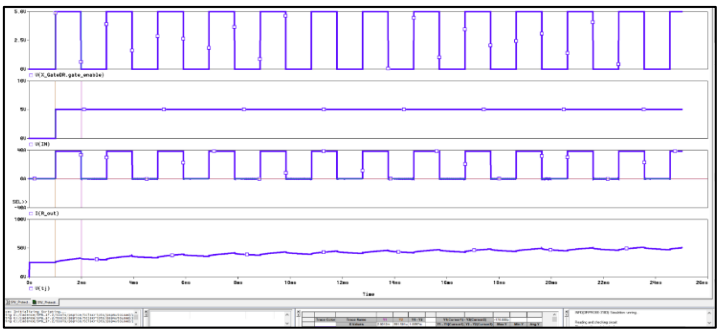

Figure 6.2.7 *1ms delay overcurrent protection and auto reset*

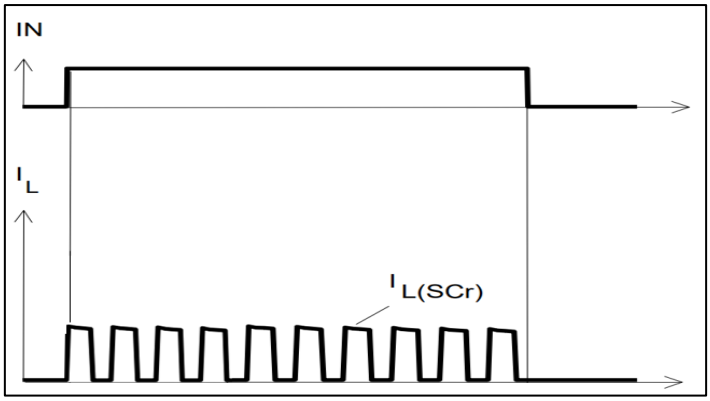

Figure 6.2.8 *1ms Delay Overcurrent Protection and Auto Reset Datasheet [45]*

Figure 6.2.7 shows the results obtained from the simulation of the behavioral model and their comparison with Figure 6.2.8 which contains the specifications in the catalog sheet.

The thesis also contains other types of protections such as over-temperature or over-voltage protection, and the results are also validated based on the catalog sheets.

Behavioral modeling of a control application using a brushless motor

An electric motor (with or without brushes) **converts** electrical energy into mechanical energy through the interaction of magnetic and electric fields. The magnetic field generated when electric current passes through a wire or coil can be used to create movement.

The parameters of the BLDC engine considered have been given in a data sheet and are shown in Table 7.3.1.

Table 7.3.1 *BLDC Motor Parameters*

Parameter	Description	Value	Unit of measurement
V_nominal	Rated Voltage of BLDC Motor	48	V
Pole_number	Number of poles per rotor	8	-
Phase_number	Number of phases	3	-
R_winding	Winding strength	6.5	Ω
L_winding	Winding inductance	0.3	Mh
KE	Constant EMF back – defines the relationship between the BEMF voltage and the rotational speed of the motor; *__BEMF__ was modeled as a sinusoidal signal – being closer to reality;	0.012	V*s/rad
KT	Torque constant - defines the torques of the motor according to the input currents of the phases;	0.012	Nm/A
J	Rotor inertia – is defined as the rotational force that must be overcome to start the engine; If J has higher values, the speed will be lower and the engine will start more slowly;	24	g*cm2
NL_SPEED	No-load speed – defines a maximum engine speed without using a load;	3300	Rpm

The PSpice model has two main output pins in addition to the 3 current phases: Shaft_speed [rpm/s] and Shaft_angle [rad] [57], used to monitor whether the PSpice model is starting and to determine if a full rotation has been performed.

The model parameters described in Table 7.3.1 were used in the adapted mathematical formula [57] to determine the rotational speed and rotational angle.

The inertia of the rotor is considered a capacitor, and the torque is a controlled current source described by a mathematical formula.

So, as the capacitor is larger, it will charge more slowly, modeling the inertia of the motor, using the PSpice program. Thus, the rotational velocity is the voltage per capacitor, and the rotational angle is the integral of the rotational velocity.

The implementation of motor control blocks is carried out by using purely analog modeling - switches, controlled voltage sources and mathematical equations. To obtain the three phases of the motor currents, without Hall sensors, three PWL voltage signals [58-60] were modeled. The PSpice command model contains three blocks, as shown in Figure 7.4.1.

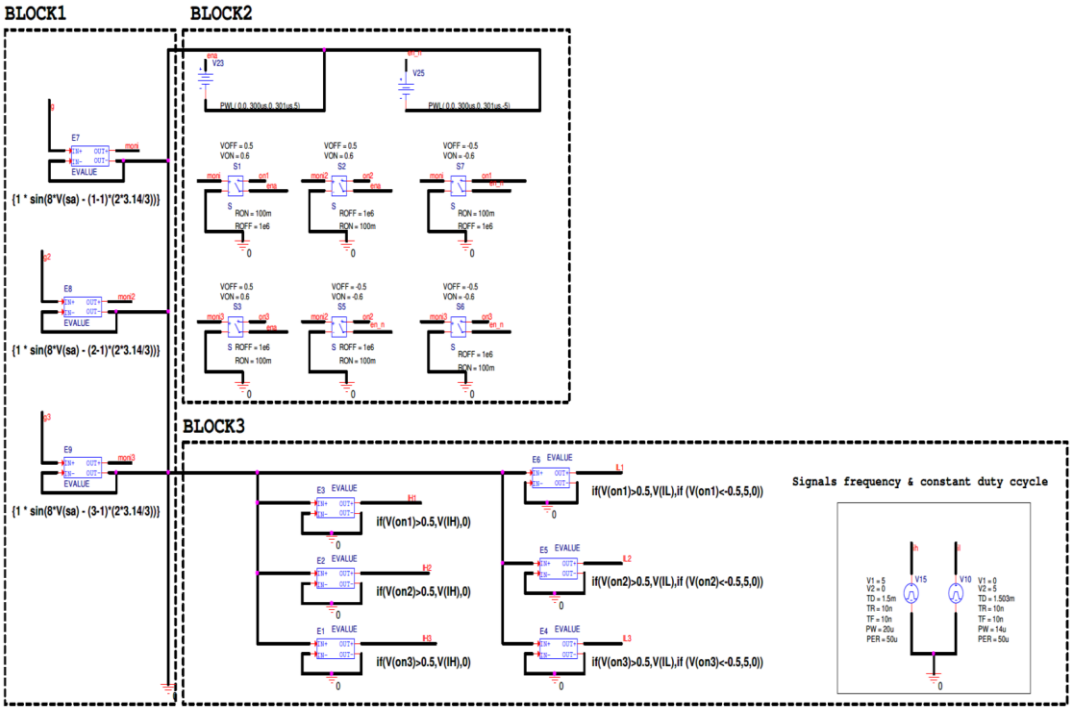

Figure 7.4.1 *The PSpice model of the control block [77]*

The control algorithm is implemented based on a feedback loop given by the angle of rotation, defining the three sinusoidal signals. This part is represented in the first component block.

The second block processes the sinusoidal signals, transforming them into rectangular, forming the envelope of the signals generated by the PWM sources that generate the operating frequency.

The third block generates the rectangular signals, used to control the MOSFET half-bridges, as shown in Figure 7.4.2.

The PWM voltage generator that drives the transistors in "*High-Side"* and "*Low-Side" configuration* has a frequency of 20kHz. The duty cycle factor of the transistor used in the "*High-Side"* configuration is 40%, and for the one used in the "*Low-Side" configuration* it is 28%, ensuring that there is no overlap between the PWM signals.

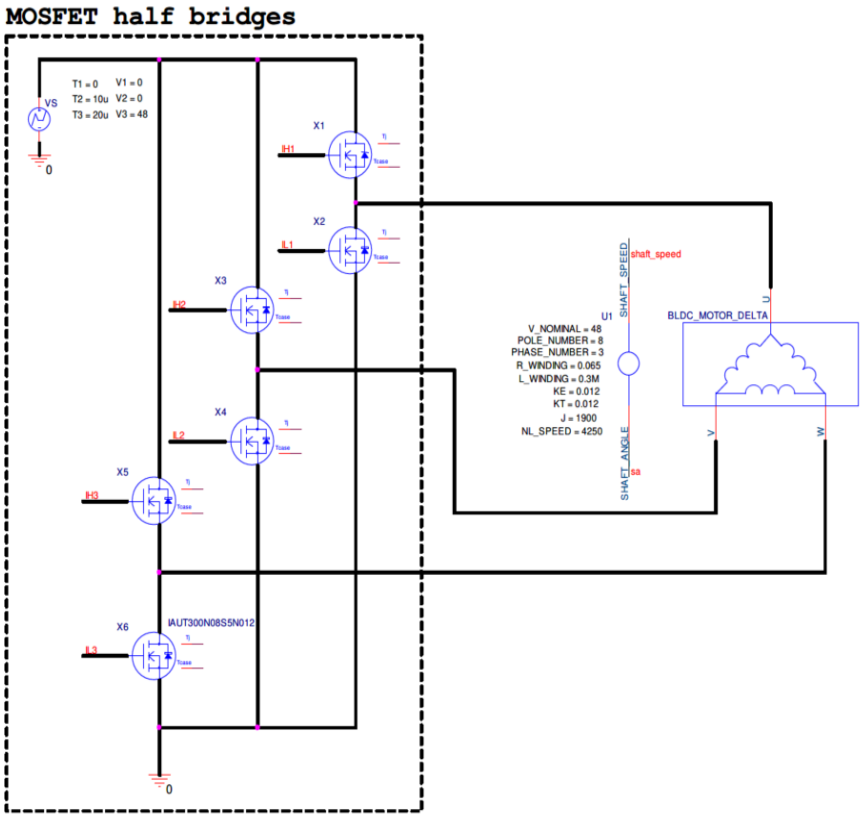

Figure 7.4.2 *MOSFET Half-Axles Used to Control a BLDC Motor [77]*

In order to validate the behavioral model, the following laboratory scenario presented in Figure 7.6.1 was performed. The simulation results are shown in Figure 7.6.2, showing the same waveforms.

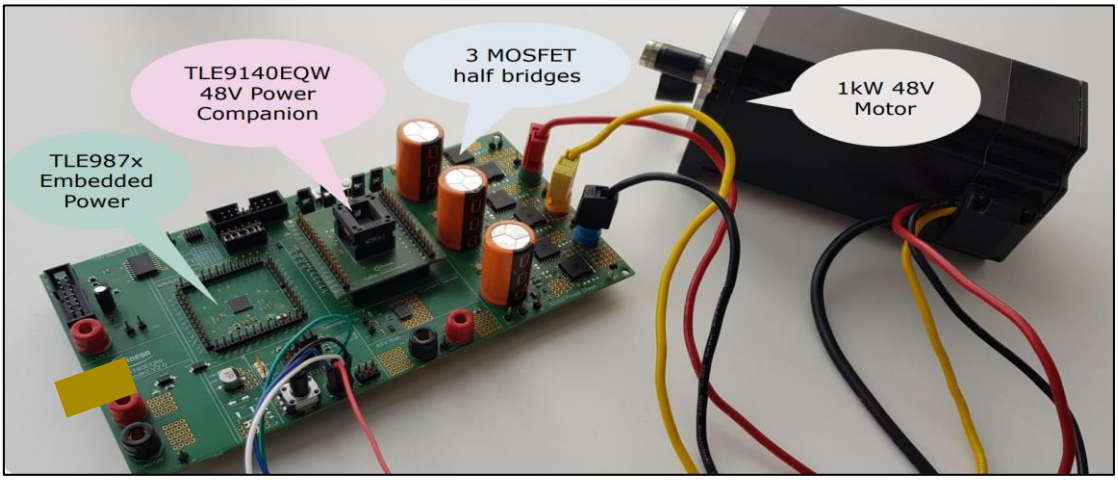

Figure 7.6.1 *Laboratory application*

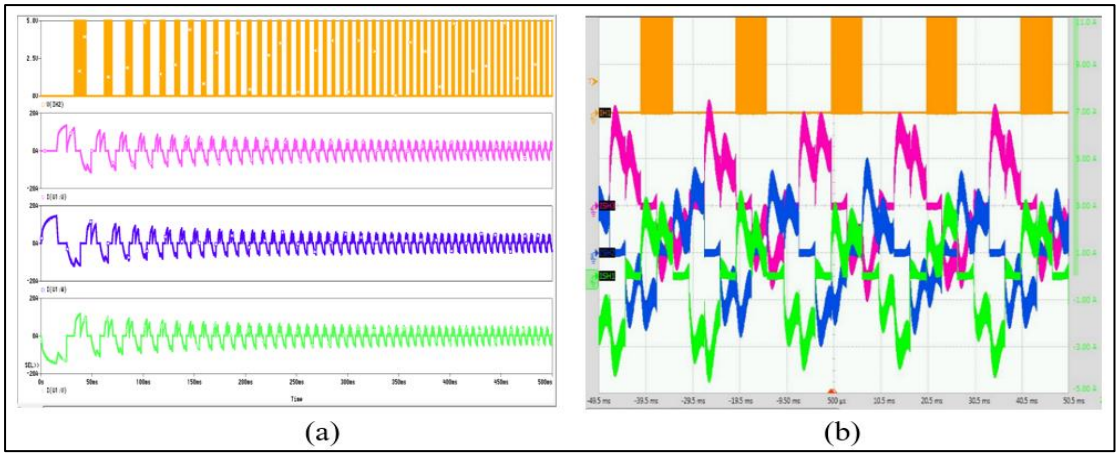

Figure 7.6.2 *PSpice simulation results (a) – output currents (10A) and oscilloscope results (b) – output currents (7A)*

Therefore, it has been demonstrated that through simulation, BLDC motors can also be modeled in SPICE language as proposed by this paper to optimize verification and validation processes. Thus, the engine model is tested in an application dedicated to the automotive industry based on a dedicated algorithm for controlling three half-bridges made up of MOSFET transistors. The algorithm is based on a feedback loop, provided by the angle of inclination of the motor. The SPICE engine model was developed using the real parameters of the BLDC engine, and simulations demonstrated its ability to accurately reproduce the real behavior of the BLDC engine under dynamic conditions. Engine model-specific benchmarks are **angle** and **speed**. They are used to monitor whether a full rotation is performed.

System modeling

The synergy between digital and analog simulation has become an increasingly studied topic, and this is possible through co-simulation. Integrated circuits are becoming more and more complex, and some of the analog functions are being replaced by their equivalent in the digital field because it is cheaper.

Therefore, the behavioral model must also keep up with the design, finding alternatives to describe digital functions. As could be seen in this paper, the current state of SPICE syntax consists of active and passive electrical components, controlled sources, logical expressions, different logical functions and operations, elements that are very suitable for describing analog behavior, but not sufficient for describing complex digital functions.

SystemC or MATLAB/Simulink are versatile languages for system-level modeling and allow architecture exploration, and SPICE is famous for its precision in simulating analog circuits, thus merging them marks a significant advance in simulation technology.

Development of the SPI model

For the development of the SPI block in Simulink, Infineon's product documentation BTS71220-4ESA [65], an intelligent power switch, was followed.

The serial peripheral interface model was implemented using components from the Simulink library and contains a clock pulse generator, transmit block, and receive block.

Therefore, the generic model of the communication interface has the following configurable parameters:

- Polarity and phase of the clock signal
- Clock Signal Period
- Transmission method: LSB or MSB
- Data length (number of bits)

The configurability of simulation models is very important because it offers the possibility of reusing a block or even adapting a behavioral model, making the development process much faster.

The data transfer is described as follows: the data is transferred on the SI and OS lines synchronized by the SCLK. When the CSN signal transitions from 1 to 0, a new communication can begin. The data is received by the peripheral on the SI line and transmitted by it on the SO line. Every communication must be terminated when the CSN signal transitions from logical 0 to logical 1 [66].

When the CSN signal is in the state of logic 1, the signals from the SCLK and SI pins are ignored and the SO signal will be forced into a high impedance state. On the SI pin the data is received synchronously with the clock periods. The information is transmitted to the similar OS output pin. The number of clock periods in each frame is given by the length of the bits transmitted [66].

When the CSN signal is in the logical 0 state, the clock signal pulses are transmitted. In order to start a communication, the signals given by the master, CSN and SCLK, must have certain times from when the CSN decreases until the clock signal starts and the same at the end of the communication, from the last period of the clock ends, until the CSN transitions into logical 1. The SCLK signal shall consist of the number of periods indicated by the length of the bits. Therefore, a communication framework is presented in Figure 8.2.1 [76].

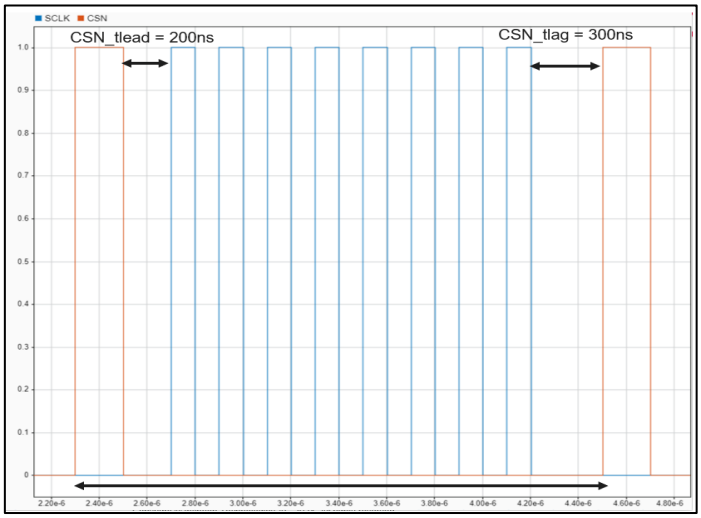

Figure 8.2.1 *CSN and SCLK signal generation*

Figure 8.2.2 is the reference according to which the two signals CSN and SCLK were modeled and synchronized, respecting the two delays.

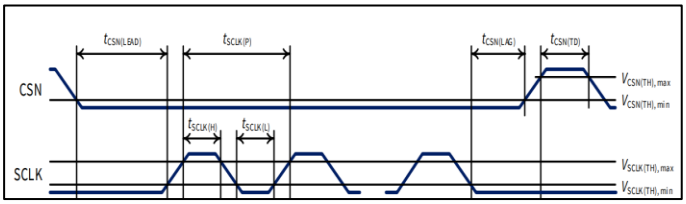

Figure 8.2.2 CSN and SCLK synchronization [65]

The system application is shown in figure 8.2.12 and contains some of the behavioral patterns described in this sentence but in Simulink language:

Microcontroller emulation consists of:

- **The data transmission block** on the SPI containing the binary message for opening and closing the power transistor;
- **The SPI data reception block**;
- **SCLK and CSN signal generator**.

The power switch consisting of:

- **The receiving block** that receives the commands from the microcontroller and sends them on to the decoder;
- **The decoder of the messages** that interprets the data and sends the commands to the switch;
- **The model of the intelligent power switch** consisting of the control circuit, the power transistor and the protection blocks; basically what was presented in the fourth chapter of this thesis, but this time in Simulink language;
- **The decoder block** the error signals and sends the messages;
- **The transmission block** that sends bitwise messages from the peripheral to the microcontroller.

The test profile of the application involves switching various resistive loads starting with normal operation and later during simulation, arriving at an overcurrent event, which can be observed by monitoring the PROT register.

At this point the commands that come to the power switch are executed sequentially without being prioritized and the messages sent to the microcontroller belong to the PROT register, containing only status bits. In reality, there are several separate registers for each functionality, and the master needs two SPI frames to receive the information.

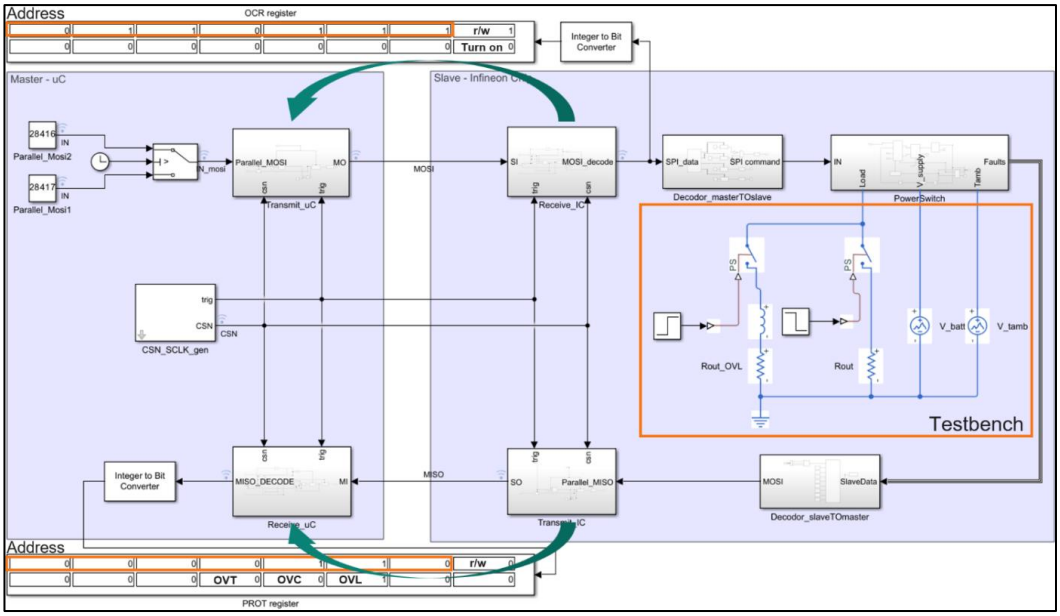

Figure 8.2.12 *System application [76]*

That being said, the SPI model [70] developed in MATLAB/Simulink meets all the requirements of the datasheet, being validated based on it, but also on the basis of different use cases to verify that all parameters such as: SCLK frequency, bit transmission (MSB or LSB first), phase and polarity of the clock signal, data length are respected.

Functionalities such as "daisy chain", protocol error detection, etc. were not considered.

In conclusion, the SPI model is fully configurable (this being the biggest advantage), meets internal requirements regarding simulation speed, can be integrated into many types of system applications, thus proving its robustness.

As future development directions, two new models are needed to be able to faithfully replicate the entire system application: a Simulink microcontroller model (used to generate SCLK and CSN signals) and a memory block (together with a decoding block) to be able to interact with the set of registers, given their type (read, writing, reading-writing, etc.).

SystemC integration with SPICE models

This research addresses the second method of modeling complex mixed-signal systems, in which the digital part, SystemC, and the analog part, Spice, must be spliced together and work precisely.

Implementing SystemC begins with establishing a development environment, using tools such as Visual Studio and the SystemC library.

The aim is to create a SystemC model, which is part of the digital system. This model is developed using a combination of C++ and SystemC coding languages, emphasizing the modular and hierarchical structure of SystemC to encapsulate the complex behaviors of a system [75].

After the SystemC model has been made, the process moves towards integrating this SystemC model with OrCAD, using the SPICE simulation environment. This integration is facilitated by the program in the OrCAD CADENCE package, called Device Model Interface (DMI), which allows the dynamic connection of the SystemC model in SPICE simulations. Key steps include configuring the DMI to recognize the SystemC model, compiling the model

into a compatible format, and establishing communication channels between the SystemC environment and SPICE to enable co-simulation.

The library that allows communication between SystemC [71] and Spice is automatically generated as part of the process when the Dynamic Link Library is created.

A state automaton (FSM) shown in Figure 8.3.1 of an intelligent power switch was made both in PSpice language using logic gates and in SystemC. The comparison between the two implementations will be presented demonstrating how fast and useful the method based on the SystemC description is.

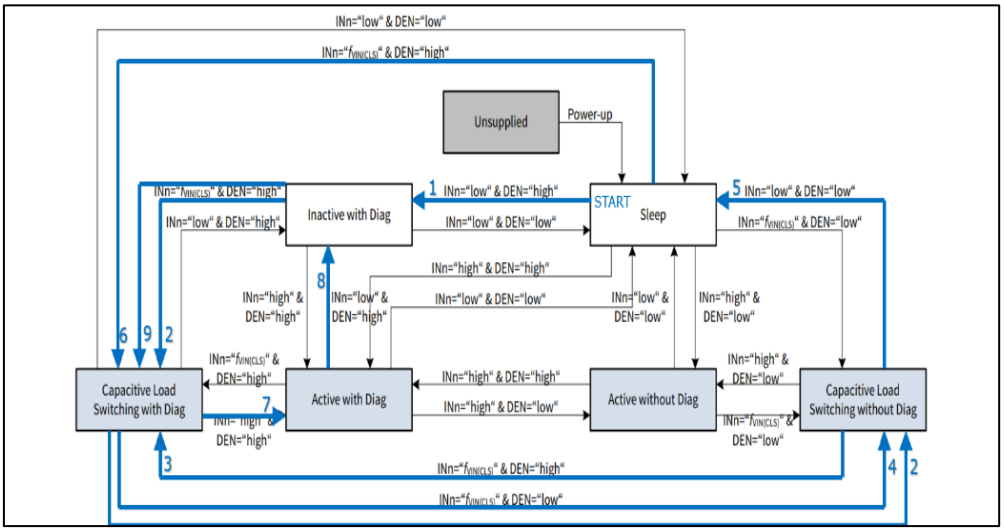

Figure 8.3.1 *Automatic states [72]*

Figures 8.3.2 and 8.3.3 show the two implementation methods.

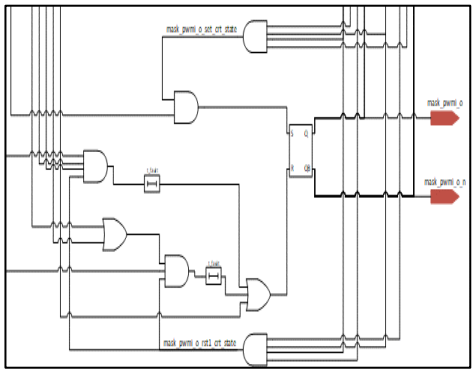

Figure 8.3.2 *Implementation of FSM PSpice [75]*

```
switch(state) {
  case 1:
        if(in[2] == 1){
      if(in[0] == 1)
        state = 32; //CLS_DIAG
      else
        state = 15; //CLS_NO_DIAG
        }
        else {
      if(in[1] == 1 && in[0] == 0)
        state = 8; // ACTIVE_NO_DIAG
      if(in[1] == 1 && in[0] == 1)
        state = 4; // ACTIVE_DIAG
      if(in[1] == 0 && in[0] == 1)
        state = 2; // INACTIVE_DIAG
      }
      break;
```

Figure 8.3.3 *FSM SystemC Implementation [75]*

Figure 8.3.4 shows the simulation results obtained using the automated states implemented using SPICE code on the one hand, and SystemC code on the other. It can be seen that the results are similar and follow the catalog sheet.

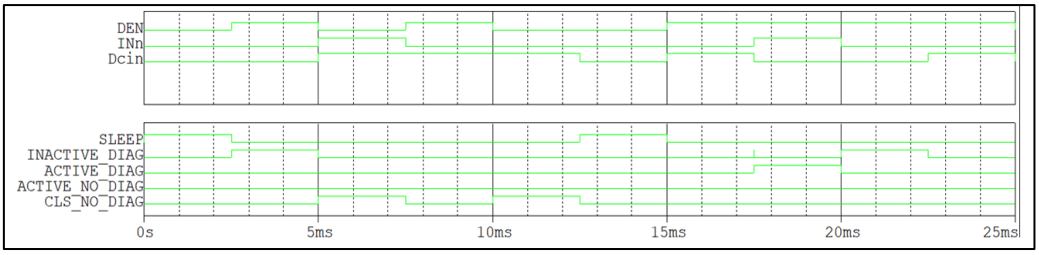

Figure 8.3.4 *Results of simulations using SystemC/SPICE [75]*

As a conclusion of this study, for the SystemC implementation, the run time is 0.59 seconds, much less compared to the SPICE implementation, which took 8 seconds.

So, as the complexity of the circuit/system increases, the number of states and transitions within FSMs usually increases, which can lead to an exponential increase in computing power. With traditional simulation methods, such as those offered by the Spice model, simulation times can be greatly extended, making the process more difficult to implement because a product contains many more integrated circuits in addition to the state controller. On the other hand, the SPICE language is the best option for modeling analog behaviors with the highest fidelity. Simulation times can become a bottleneck, slowing down the development cycle.

In contrast, the implementation of SystemC [73] represents a more modern approach to system simulations, being a top-level modeling language designed specifically for the description of systems and hardware, allowing a faster simulation of complex systems by abstracting insignificant details that are often at the heart of SPICE simulations. This abstraction means that the simulation time is significantly reduced while maintaining the required accuracy [75].

Conclusions

This chapter presents the results obtained as well as the list of published works.

Original contributions

The present paper opens new perspectives in the field of simulation, demonstrating the usefulness of simulation models.

[CO1] Fundamental behavioral models of delay circuits on different fronts as well as the behavioral model of the hysteresis comparator have been made limiting themselves to the realization of basic functionality. Behavioral models of some active components as well as the modeling of a gate control circuit of an nMOS transistor were implemented. All of these blocks were used to model the behavior of a smart power switch with guards. In that chapter the modeling methods were explained, and it was shown that the model complies with the specifications of the datasheet both in terms of electrical parameters and waveforms.

[CO2] For the Widlar current reference implemented with both bipolar transistors and MOS, a mathematical study was carried out that aims to reduce the occupied area in integrated

circuits. On the other hand, a comparison of the output currents was made using classic models and those with short channel with improved performance. The threshold voltage, as the main nMOS parameter, has been varied using Monte Carlo analysis, demonstrating that predictive models (PTMs) are more accurate, but since the number of parameters describing the mathematical equations modeling short-channel effects is significant, the simulation speed will be affected if these models are used in an application with a high level of complexity.

[CO3] Another point of interest was the modeling of a control algorithm using a BLDC motor model as a load. In this study it was demonstrated that the motor can perform multiple rotations from $-\pi$ to π. On the other hand, the three phases of current were shown and compared with those obtained from laboratory measurements, and the results were very similar.

The chapter dedicated to this topic explains that the control algorithm is implemented based on a reaction loop given by the angle of rotation. From the point of view of the results obtained, it was demonstrated the modeling of the BLDC motor load and its control in SPICE language, as well as the validation of this application with the results from the laboratory.

[CO4] System modeling is a major point of interest because it is a new field, which allows the emulation and merging of the analog and digital parts. Thus, two studies were conducted:

- implementation of a system application (involving the SPI model and an intelligent power switch peripheral) developed in the MATLAB/Simulink program, modeling a configurable serial communication protocol (SPI), thus realizing data transfer at the simulation level;
- co-simulation between the SPICE language that can model the analog part very precisely with SystemC where a state automaton has been implemented. This state automaton was also modeled in SPICE language, showing that the running time for SystemC was 0.59s compared to PSpice where the simulation lasted 8 seconds.

[CO5] The optimization of the Gilbert compute cell was demonstrated by modeling an analog compute cell derived from the Gilbert cell, where an analog multiplication circuit was studied and simulated, in order to be able to make real-time calculations with a small number of devices. MOSFET transistors operate in the region below the threshold, allowing for low currents and therefore low power consumption. The output current error is less than 0.5% in the operating temperature range and less than 0.1% due to the technological variation of the threshold voltage of the MOS transistor. The main contributions are the achievement of low power consumption and behavioral modeling with Monte Carlo analysis, which shows the possibility of extending this circuit to the scale of the entire neural network with minimal errors.

As a general conclusion, behavioral modeling at the integrated circuit level is a relatively new field that will evolve over time towards a **system modeling** because the fundamental characteristic of this branch is to support digitalization. This is based on the premise that a real test board is no longer needed to be verified in the laboratory, but the use of a behavioral model in order to test various functionalities, such as control blocks/protections, state automatons, communication protocols. As general characteristics, a **behavioral model** must meet the following standards: run fast and robust, emulate the behavior of a device, circuit from an

analog and/or digital point of view, be as configurable and easy to use as possible, be able to be used in different simulators.

List of original works

I. Scientific articles indexed in International Scientific Journals

[1] **A1.** M.J. Cristea, **L.A. Gheorghe**, L.Dobrescu, "Behavioral Modelling of Analog Computing Circuit for Artificial Intelligence Neural Networks", ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY, 2025. **Category Quartile Q1**.

II. Scientific articles indexed in the Clarivate-Web of Science database

[2] **L. A. Gheorghe**, L. Dobrescu, S. S. Varzaru, M. Craciun, D. Dobrescu, "*Behavioral Modeling of Current Reference for LEDs Control Applications*", 229-232, International Semiconductor Conference (CAS), 2020. Publisher: IEEE, ISBN 978-1-7281-1073-8, ISSN 1545-827X, **WOS:000637264600052**

[3] P.A. Stefanescu, M.J. Cristea, **L.A. Gheorghe**, L.Dobrescu, "Analog-multiply current-mode circuit", International Semiconductor Conference (CAS), 2024. Publisher: IEEE **ISBN** 979-8-3503-5208-5979-8-3503-5207-8, ISSN 1545-827X, **WOS:001361909500074**.

III. Volumes of internationally recognized scientific events, organized in the country and abroad, indexed in other international databases – IEEE database

[4] **L.A. Gheorghe**, C.L. Gogu, R.C. Cavache, L. Dobrescu, D.Dobrescu, "Simulation Model for Brushless DC Motor Loads Controlling", International Semiconductor Conference (CAS), 2023. ISSN: 2377-0678 DOI: 10.1109/CAS59036.2023.10303714, Publisher: IEEE.

[5] A. Dragomir, **L.A. Gheorghe**, L. Gogu, D. Shikhar, Integration of SystemC with PSpice Simulations Models, ECAI, 2025, *forthcoming, IEEE indexing.*

[6] **L.A. Gheorghe**, E. Sorega, L. Dobrescu, SPI Digital Twin in MATLAB/Simulink, ECAI, 2025, *forthcoming, IEEE indexing.*

[7] **L. A. Gheorghe,** D. Dobrescu, I. Busu and L. Dobrescu, "A New Optimization and Statistical Analysis for Widlar Current Source using Predictive MOSFET Models," 2021 Publisher: IEEE, 32nd International Conference on Microelectronics (MIEL), Nis, Serbia, 2021, pp. 227-230, DOI: 10.1109/MIEL52794.2021.9569090, Publisher: IEEE.

IV. Volumes of internationally recognized scientific events, organized in the country and abroad

[8] **L. A. Gheorghe**, L. Dobrescu**,** *"Investigations on Behavioral Modeling – NEW View of the Electronic Circuits"* EPIEM - INNOVATION AND SUSTAINABILITY INTERNATIONAL SCIENTIFIC CONFERENCE, Sixth Edition, October 2020. ISSN 2501-6695, ISSN-L 2501-6695

[9] **L.A. Gheorghe**, D. Dobrescu, "*Application of behavioral modeling for ics investigation",* **INTERNATIONAL ARCHITECTURAL SCIENCES AND APPLICATION SYMPOSIUM,** IKSAD, **Turkey, 2021. ISBN: 978-625-7464-51-2**

Selective bibliography

[8] MATLAB Masking Fundamentals, https://www.mathworks.com/help/simulink/ug/block-masks.html?searchHighlight=mask&s_tid=srchtitle_support_results_2_mask , accessed on 31.05.2023.

[9] Model-Based Design, https://www.mathworks.com/solutions/model-based-design.html , accessed on: 22.03.2023.

[10] **L. A. Gheorghe**, D. Dobrescu, I. Busu, L. Dobrescu, *A New Optimization and Statistical Analysis for Widlar Current Source using Predictive MOSFET models*, Honey, 2021.

[11] W. Liu, *Mosfet Models for SPICE simulation, including Bsim3v3 and BSIM4*, Wiley & Sons, 2001.

[12] HSPICE Models Manuals, Version X-2005.09, September, 2005, Available at:. https://www.researchgate.net/file.PostFileLoader.html?id=59d32a3ff7b67eb36527afb2&assetKey=AS%3A545258368507904%401507011133933

[13] Y.Cao, *Predictive Technology Model for Robust Nanoelectronic Design*, Springer, 2011.

[14] Predictive Technology Model Available at: http://ptm.asu.edu/ accessed on 20.02.2022

[15] BSIM 4 Manual: https://cmosedu.com/cmos1/BSIM4_manual.pdf

[22] M. Cheng, L. Yang, Z. Ding, S. Li, J. Lei, *Circuit design of a novel analog multiplier*, 2023 9th International Conference on Control Science and Systems Engineering (ICCSSE), Shenzhen, China, pp. 386-390, 2023.

[23] T. Unuk, R. Arslanalp, S. Tez, *Design of current-mode versatile multi-input analog multiplier topology*, AEU - Intl. Jour. of Electronics and Comm., vol.160, 2023.

[24] B. Gilbert, *Design considerations for BJT active mixers*, Low-Power HF Microelectronics: A Unified Approach, Jul. 1996.

[25] Analog Devices Inc., *Analog Multipliers*, available online at: https://www.analog.com/media/en/training-seminars /tutorials/MT-079.pdf

[26] D. Dobrescu, B. Cretu, E. Simoen, A. Veloso, A. Voicu-Spineanu, L. Dobrescu, *Si GAA NW FETs threshold voltage evaluation*, Solid-State Electronics, vol.194, 2022.

[27] Cadence Virtuoso Studio, available online at: https://www.cadence.com/en_US/home/tools/custom-ic-analog-rf-design/virtuoso-studio.html

[28] A. Wand, B.H. Calhoun and A.P. Chandrakasan, *Sub-threshold Design for Ultra Low-Power Systems*, Springer, New York, 2006.

[29] P.A. DURAN, *A Practical Guide to Analog Behavioral Modeling for IC System Design*, Springer Science & Business Media, New York, 2012, pp.33–35.

[30] L. Dobrescu, D. Dobrescu, *Advanced Modeling of MOS Devices*, 104 pages, Printech Publishing House, ISBN 973-652-561-9, Bucharest, 2002.

[34] *Bridging Theory into Practice*, Published by Infineon Technologies AG Edition 2008.

[38] B.Razavi., M. Graw-Hill, *Design of Analog CMOS Integrated Circuits* INT ERNATIONAL EDITION, 2016.
[39] BTS3800SL Infineon Technologies Datasheet : https://www.infineon.com/dgdl/Infineon-BTS3800SL-DS-v01_01-EN.pdf?fileId=5546d4625a888733015aa9b0af2a361e accessed on: 22.05.2023.
[42] **L. A. Gheorghe**, L. Dobrescu, S. S. Varzaru, M. Craciun, D. Dobrescu, *Behavioral Modeling of Current Reference for LEDs Control Applications*, 229-232, International Semiconductor Conference (CAS), 2020.
[45] BTS436L2G Infineon Technologies Datasheet: https://www.infineon.com/dgdl/Infineon-BTS436L2G-DS-v01_00-EN.pdf?fileId=5546d4625a888733015aa9aff33e35e5
[57] PSpice Brushless DC Motor Model. Available online: https://www.pspice.com/resources/application-notes/brushless-dc-motor-model (accessed on 14.01.2024).
[58] P. Bauer, P. Duijsen, *Modeling and Simulation for Power Electronics and Electrical Drives*, Proceedings of the International Power Electronics Conference: PCIM 2001 Europe, Nürnberg, June 19 - 21, **2001.** ISBN: 3928643274, 9783928643276.
[59] B. K. Bose, *Voltage-Fed Converters*, Modern Power Electronics and AC Drives, Prentice Hall PTR, Upper Saddle River, **2002**. pp. 210-240.
[60] S. T. Jo, H. S. Shin, Y.G. Lee, J. H. Lee, J. Y. Choi, *Optimal Design of a BLDC Motor Considering Three-Dimensional Structures Using the Response Surface Methodology*, Energies, *15*, 461, 2022.
[65] Datasheet BTS71220-4ESA, Infineon Technologies AG, Rev. 1.10, 2021-03-23. https://www.infineon.com/dgdl/Infineon-BTS71220-4ESA-DS-v01_00-EN.pdf, 2021.
[66] I. I. b. Jamaludin, H. b. Hassan, *Design and Analysis of Serial Peripheral Interface for Automotive Controller*, IEEE Student Conference on Research and Development (SCOReD), Batu Pahat, Malaysia, 2020, pp. 498-501, 2020.
[71] O. S. Initiative, IEEE 1666-2005 *Standard SystemC Language Reference Manual*, 2005.
[72] BTG7090-2EPL datasheet. Available online: https://www.infineon.com/cms/en/product/power/smart-power-switches/high-side-switches/profet-load-guard-12v-automotive-smart-high-side-switch/btg7090-2epl, accessed on 19.04.2024.
[73] J. E. Hopcroft, R. Motwani, J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Second Edition, pp. 37-81, 2001.
[74] M. J. Cristea, **L. A. Gheorghe**, L. Dobrescu, "*Behavioral Modelling of Analog Computing Circuit for Artificial Intelligence Neural Networks*", ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY, DOI: 10.59277, pp. 51–62, 2025.
[75] A. Dragomir, **L. A. Gheorghe**, L. Gogu, D. Shikhar, Integration of SystemC with PSpice Simulations Models, ECAI, 2025, *forthcoming, IEEE indexing.*
[76] **L. A. Gheorghe**, E. Sorega, L. Dobrescu, SPI Digital Twin in MATLAB/Simulink, ECAI, 2025, *forthcoming, IEEE indexing.*
[77**] L.A. Gheorghe**, C.L. Gogu, R.C. Cavache, L. Dobrescu, D.Dobrescu, „Simulation Model for Brushless DC Motor Loads Controlling", International Semiconductor Conference (CAS), 2023. ISSN: 2377-0678 DOI: 10.1109/CAS59036.2023.10303714, Publisher: IEEE.