Capacitance and Power Modeling at Logic-Level

João Baptista Martins [*], **Ricardo Reis**
{batista,reis}@inf.ufrgs.br
II, UFRGS, Porto Alegre, Brazil

José Monteiro
jcm@inesc.pt
IST / INESC, Lisboa, Portugal

Abstract

Accurate and fast power estimation of CMOS circuits during the design phase is required to guide power optimization techniques employed to meet stringent power specifications. Logic-level power estimation tools, such as those available in the SIS and POSE frameworks are able to accurate calculate the switching activity under a given delay model. However, capacitance and delay modeling is crude. The objective of the work described in this paper is to investigate how close can logic-level power estimates get to estimates obtained with circuit-level simulators such as SPICE.

We propose new models for the input and output capacitances of logic gates, taking into account the gate´s internal capacitances and the interconnect capacitances extracted from layout. The results we present show an agreement of the logic-level estimates and SPICE, with less that 10% error.

1. Introduction

Power consumption is becoming one of the most important design challenges in the design of VLSI circuits. Optimization techniques for low power are being employed at all design levels of abstraction. In order to guide designers and optimization tools, there is a pressing need for accurate and fast power estimation tools.

During normal operation, the power dissipation of a CMOS circuit is directly related to the switching activity. For a well designed circuit, the total average power can be approximated by the switched-capacitance power [5]. This is an underlying assumption of almost all of the available power estimation tools at the logic gate and higher levels of abstraction. The average dynamic power consumption of a CMOS circuit is then given by:

$$Power = \frac{1}{2} . f . V_{dd}^2 . \sum_{i=1}^{n} C_i . \alpha_i \qquad (1)$$

Where f is the clock frequency, V_{dd} is the supply voltage, C_i and α_i are capacitance load and average switching of the logic gate i, respectively.

Significant amount of work has been carried out in developing efficient techniques to estimate the switching activity of a CMOS circuit [9]. These techniques can be divided into two classes: statistical techniques (also known as dynamic techniques), e.g. [1]; probabilistic (or static) techniques, e.g. [2].

Statistical techniques simulate the circuit repeatedly until the power values converge to an average power, based on some statistical measures. Probabilistic techniques propagate input statistics through the circuit to obtain the switching probability for each gate in the circuit. Probabilistic techniques are employed in the power estimation tools inside SIS [3] and POSE [4].

In this paper, we focus on the problem of capacitance modeling. This is typically done very simplistically at the logic level. We propose to build a more accurate model. At the logic-level, only the input and output nodes of the gates are available. We present an equivalent load capacitance model for each external node, computed from the internal transistor capacitances and the interconnection capacitance.

The interconnection capacitance is currently being extracted from layout information. We have performed several experiments on (necessarily) simple circuits where power estimates were obtained both using SPICE, and SIS using the capacitance values obtained through our model. The results are very promising, with power differences of less than 10%.

This paper is organized as follows. In Section 2, we briefly present the capacitance model used for a transistor MOSFET. In Section 3, we describe the methodology for calculating the equivalent capacitances associated the input and output nodes of logic gates. In Section 4, we describe the experimental setup and the tools used. Experimental results are presented in Section 5. In Section 6, we give some conclusions and discuss future research.

2. Mosfet Capacitance Model

The MOSFET transistors exhibit a number of parasitic capacitance [6] (Figure 1), which must be accounted for in circuit design: gate-to-source capacitance (C_{GS}), gate-to-drain capacitance (C_{GD}), gate-to-bulk capacitance (C_{GB}), source-to-bulk capacitance (C_{SB}) and drain to bulk capacitance (C_{DB}). In this work, our interest will be the switching region of the transistors and all the capacitances will be considered, with the exception of C_{GB}.

Although these capacitances are a nonlinear function

[*] This researcher was supported by UFSM and CAPES/Brazil.

of the voltage, the general approach is to assume them as a linear, time-invariant element. Given some technology parameters and the size the transistor (W/L), a value for each of these parasite capacitances can be computed [6].

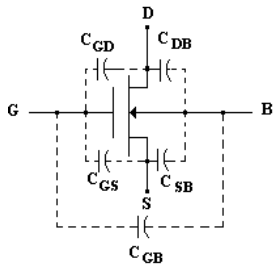


Figure 1- Parasites Capacitances of the MOSFET

For definition [6]:

$$C_G = C_{ox}LW + 2C_o \quad (2)$$

$$C_{GD} = G_{GS} = \frac{1}{2}C_{ox}(WL) + C_o \quad (3)$$

$$C_{DB,bot} = K(V_l)C_{j0}A_D \quad (4)$$

$$C_{DB,sw} = K_{1/3}(V_l)C_{jsw}l_D \quad (5)$$

$$C_{DB} = C_{DB,bot} + C_{DB,sw} \quad (6)$$

$$C_{SB,bot} = K(V_l)C_{j0}A_S \quad (7)$$

$$C_{SB,sw} = K_{1/3}(V_l)C_{jsw}l_S \quad (8)$$

$$C_{SB} = C_{SB,bot} + C_{SB,sw} \quad (9)$$

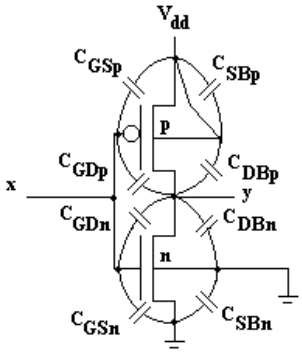
where:
L = channel length
W = channel width
C_o = overlap capacitance
C_{j0} = zero bias capacitance per unit area
C_{jsw} = zero-bias sidewall capacitance per unit perimeter
A_D = Area of dreno
l_D = perimeter of dreno
A_S = Area of source
L_S = perimeter of source

3. Modeling Gate Input and Output Capacitances

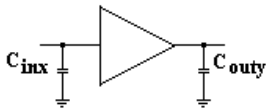
For circuits described at the logic level there is only access to the logic gates input and output nodes. Therefore, all capacitances internal to the gate have to be taken into account in concentrated capacitance models at these external nodes. In this section we describe how we compute the equivalent capacitance load for any logic gate from its transistor level description.

A. CMOS Inverter

The CMOS inverter (Figure 2) presents two externals nodes. In this case, equivalent capacitances at the input x. $C_{in\ x}$, and output y, $C_{out\ y}$, nodes can be simply computed using:



(a)



(b)

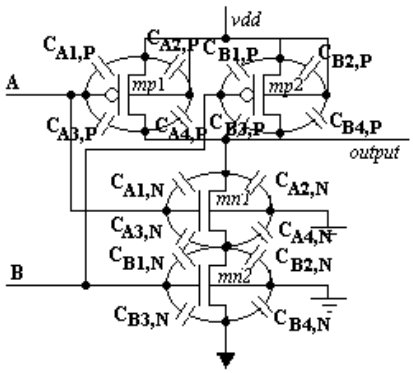
Figure 2 – Parasite capacitances of CMOS inverter
(a) Transistor-level representation
(b) Logic-level representation

$$C_{inx} = C_{GSp} + C_{GDp} + C_{GSn} + C_{GDn} \quad (10)$$

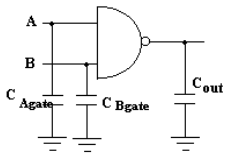
$$C_{outy} = C_{DBp} + C_{GDp} + C_{GDn} + C_{DBn} \quad (11)$$

B. 2 -Input CMOS NAND Gate

In the case of a CMOS 2-input NAND gate (figure 3), there are two inputs nodes with concentrate capacitance and one output node. The additional difficulty is that we now have internal capacitance modeling. We analyze the equivalent capacitance from superposition of the signals. For example, when calculate the equivalent capacitance to a input node, the inputs and output will be in the ground level. The capacitance of internal nodes depends on the logic value of the others inputs. To model this effect we are assuming a probability of 0.5 that all inputs are set to 1. That is, the NMOS transistors will be ON at half the time.



(a)



(b)

Figure 3 – Parasite Capacitances of 2-input NAND
(a) Transistor-level representation
(b) Logic-level representation

In the case NAND2, the input combination possible is 00, 01, 10 and 11.

Input "a":

For input combinations 00 and 10, the "mn2" transistor is OFF and equivalent capacitance node (C_{a1}) is equal:

$$C_{a1} = 0.5(C_{A1,N} + \frac{C_{A3,N} * C_{T,N}}{C_{A3,N} + C_{T,N}}) \quad (12)$$

$$C_{T,N} = C_{A4,N} + C_{B1,N} + C_{B2,N} \quad (13)$$

For input combinations 01 and 11, the mn2 transistor is ON and equivalent capacitance (C_{a2}) is equal:

$$C_{a2} = 0.5(C_{A1,N} + C_{A3,N}) \quad (14)$$

Then

$$C_{A,N} = C_{a1} + C_{a2}$$

$$C_{A,N} = 0.5(C_{A1,N} + \frac{C_{A3,N} * C_{T,N}}{C_{A3,N} + C_{T,N}}) + 0.5(C_{A1,N} + C_{A3,N}) \quad (15)$$

In terms probabilistics:

$$C_{A,N} = p_0(C_{A1,N} + \frac{C_{A3,N} * C_{T,N}}{C_{A3,N} + C_{T,N}}) + p_1(C_{A1,N} + C_{A3,N}) \quad (16)$$

and C_{Agate} is:

$$C_{Agate} = p_0(C_{A1,N} + \frac{C_{A3,N} * C_{T,N}}{C_{A3,N} + C_{T,N}}) + p_1(C_{A1,N} + C_{A3,N}) + C_{A1,P} + C_{A3,P} \quad (17)$$

where p_0 is probability second input to be equal the ZERO and p_1 is probability second input to be equal the ONE ($p_0 + p_1 = 1$).

Input "b"

In the same form, can analyses the second input.

For input combinations 00 and 01, the mn1 transistor is OFF and C_{b1} is equal:

$$C_{b1,N} = 0.5(C_{B3,N} + \frac{C_{B1,N} * C_{T1,N}}{C_{B1,N} + C_{T1,N}}) \quad (18)$$

For input combinations 10 and 11, the mn1 transistor is ON and C_{B2} is equal:

$$C_{b2,N} = 0.5(C_{B1,N} + C_{B3,N}) \quad (19)$$

Then

$$C_{B,N} = C_{b1,N} + C_{b2,N} \quad (20)$$

$$C_{B,N} = 0.5(C_{B3,N} + \frac{C_{B1,N} * C_{T1,N}}{C_{B1,N} + C_{T1,N}}) + 0.5(C_{B1,N} + C_{B3,N}) \quad (21)$$

$$C_{T1,N} = C_{A4,N} + C_{A3,N} + C_{B2,N} \quad (22)$$

In terms probabilistics :

$$C_{B,N} = p_0(C_{B3,N} + \frac{C_{B1,N} * C_{T1,N}}{C_{B3,N} + C_{T1,N}}) + p_1(C_{B1,N} + C_{B3,N}) \quad (23)$$

and C_{Bgate} is:

$$C_{Bgate} = p_0(C_{B3,N} + \frac{C_{B1,N} * C_{T1,N}}{C_{B1,N} + C_{T1,N}}) + p_1(C_{B1,N} + C_{B3,N}) + C_{B1,P} + C_{B3,P} \quad (24)$$

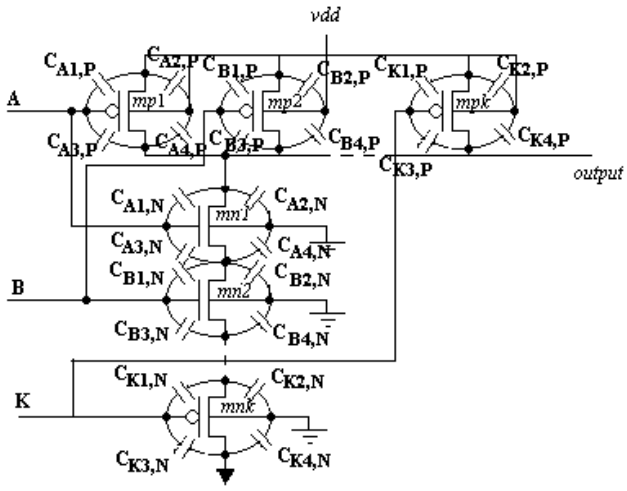
Where:
C_{Agate} and C_{Bgate} are equivalent capacitances for input A and input B, respectively.

And output is:

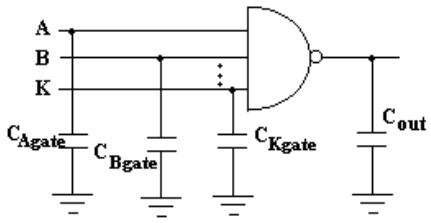
$$C_{out} = C_{A1,N} + C_{A2,N} + C_{A3,P} + C_{A4,P} + C_{B3,P} + C_{B4,P} \quad (25)$$

C. NAND with *n*-inputs

These considerations also are valid for CMOS NAND´s with *n* inputs, where *n* is an integer number and probability=0.5. This case one input is switching and others inputs assume *0* or *1* values. All combinations of input vectors will be analyzed.



(a)



(b)

Figure 4 – Parasite Capacitances of a k-input NAND
(a) Transistor-level representation
(b) Logic-level representation

Initial Considerations:
$C_{GD(0)} = C_{A1,N}$
$C_{DB(0)} = C_{A2,N}$
$C_{GS(0)} = C_{A3,N}$
$C_{SB(0)} = C_{A4,N}$

For *k*= *B* to *K*

$C_{GD(1)} = C_{B1,N}$	.....	$C_{GD(k)} = C_{K1,N}$
$C_{DB(1)} = C_{B2,N}$	.....	$C_{DB(k)} = C_{K2,N}$
$C_{GS(1)} = C_{B3,N}$	.....	$C_{GS(k)} = C_{K3,N}$
$C_{SB(1)} = C_{B4,N}$	.....	$C_{SB(k)} = C_{K4,N}$

a) Capacitance of the first input NMOS net(transistor connected to the output i = 0):

$$C(0) = C_{GD(0)} + \frac{2}{2^n} C_{GS(0)} + \frac{1}{2^n} \sum_{q=1}^{n-1} (2^{n-q} \frac{C_{GS(0)} * C_{Ti(q-1)}}{C_{GS(0)} + C_{Ti(q-1)}}) \quad (26)$$

$$C_{Ti(0)} = C_{SB(0)} + C_{GD(1)} + C_{DB(1)} \quad (27)$$

For q=1 to n-2

$$C_{Ti(q)} = C_{Ti(0)} + \sum_{i=1}^{q} (C_{GS(i)} + C_{SB(i)} + C_{GD(i+1)} + C_{DB(i+1)}) \quad (28)$$

The equivalent capacitance gate (C_{Agate}) is:

$$C_{Agate} = C_{(0)} + C_{A1,P} + C_{A3,P} \quad (29)$$

b) Capacitance of intermediates inputs (0 < i < n-1):

$$C(i) = \frac{2^{n-i}}{2^n} C_{GD(i)} + \frac{1}{2^n} \sum_{q=1}^{i} (2^{n-q} \frac{C_{GD(i)} * C_{T\_GD(q-1)}}{C_{GD(i)} + C_{T\_GD(q-1)}}) + \frac{2^{i+1}}{2^n} C_{GS(i)} + \frac{1}{2^n} \sum_{q=n-1}^{i+1} (2^q \frac{C_{GS(i)} * C_{T\_GS(n-1-q)}}{C_{GS(i)} + C_{T\_GS(n-1-q)}}) \quad (30)$$

$$C_{T\_GD(0)} = C_{DB(i)} + C_{GS(i-1)} + C_{SB(i-1)} \quad (31)$$

For q=1 to i –1

$$C_{T\_GD(q)} = C_{T\_GD(0)} + \sum_{i}^{i-q-1} (C_{GS(i-2)} + C_{SB(i-2)} + C_{GD(i-1)} + C_{DB(i-1)}) \quad (32)$$

$$C_{T\_GS(0)} = C_{GD(i+1)} + C_{SB(i)} + C_{DB(i+1)} \quad (33)$$

For q=1 to i - 1

$$C_{T\_GS(q)} = C_{T\_GS(0)} + \sum_{i}^{q} (C_{GS(i+1)} + C_{SB(i+1)} + C_{GD(i+2)} + C_{DB(i+2)}) \quad (34)$$

The equivalent capacitance gate (C_{Igate}) is:

$$C_{Igate} = C_{(i)} + C_{i1,P} + C_{i3,P} \quad (35)$$

For i = A, B… (K-1)

c) Capacitance of the last input (transistor connected to ground - i = n – 1):

$$C(n-1) = \frac{2}{2^n} C_{GD(n-1)} + \frac{1}{2^n} \sum_{q=1}^{n-1} (2^{n-q} \frac{C_{GD(n-1)} * C_{Tf(q-1)}}{C_{GD(n-1)} + C_{Tf(q-1)}}) + C_{GS(n-1)}) \quad (36)$$

$$C_{Tf(0)} = C_{GS(n-2)} + C_{SB(n-2)} + C_{DB(n-1)} \quad (37)$$

For q=1 to n-2

$$C_{Tf(q)} = C_{Tf(0)} + \\ + \sum_{i=1}^{q} (C_{GD(n-1-i)} + C_{DB(n-1-i)} + C_{GS(n-2-i)} + C_{SB(n-2-i)}) \quad (38)$$

The equivalent capacitance gate (C_{Kgate}) is:

$$C_{Kgate} = C_{(K-1)} + C_{K1,P} + C_{K3,P} \quad (39)$$
n = K

Where
C_{GD} = Capacitance *Gate-Dreno*
C_{GS} = Capacitance *Gate-Source*
C_T = Equivalent capacitance of analyzed node
i = node
n= inputs number of the logic gate

D. CMOS NOR's

The NOR logic gates are dual NAND logic gate. In this case the internal capacitance will be present in the transistor *pmos* network.

E. Interconnection Capacitances

The interconnection line capacitance calculated and it is included to output node of the considered gate. The next stage present an equivalent capacitance, whose value depends on the type of the logic gates and load, connected.

Then, the total node capacitance is the sum of the capacitance contributions to the node. The total capacitance of the output node of a gate *g* connected to *n* others gates is given by:

$$C_{TOTAL} = C_{line} + \sum_{i=1}^{n} C_{igate} \quad (40)$$

Where:
C_{line} = interconnection capacitance connected to the node.
C_{igate} = equivalent capacitance gate.
n = total number of logic gates connected of the node.

4. Experimental Setup

The block diagram in Figure 5 gives an overall view of the proposed method for validation of the proposed model. The method consists in extraction of capacitances and transistors from the layout of the circuit. We use the following tools in the validation process:

TROPIC2 [7] – This tool automatically generates the layout of a circuit with a "*linear-matrix multi-row*" layout style. It makes the partitioning, placement and routing of a circuit, generating a symbolic output file. Symbolic file is a textual description of a circuit without taking account of the design rules. Wire, contacts and transistors are indicating with a unitary dimension. The input file uses the SPICE format.

CADENCE – The Cadence system was used for layout compaction and capacitance extraction.

Power Estimate – This program is embedded into SIS. It calculates the switching activity circuit using various delays model: zero delay, unit delay and general delay.

HSPICE – This wellknow program does circuit analysis at the level transistor. It is very accurate and secure, however it use is limited to circuits with a small number of inputs.

The validation flow used is the following:

1) Read circuit in the format *blif* file;
2) Converter *blif* file to *sim* file. The *sim* file format is a SPICE logic gate level file;
3) the *sim* file is read by TROPIC2, that generates a symbolic layout file;
4) Circuit compaction is performed using the CADENCE tool. The interconnection capacitances and transistor are extracted. The result of this process step is a SPICE *netlist*;
5) With the *netlist* and *blif* files it is possible to calculate the concentrated capacitances at the nodes of each logic gate.
6) The capacitances file and logic gate library are read into SIS where we can run the *Power Estimate*, that calculates the average power of the circuit;
7) The HSPICE is the simulation tool used. The analysis was done in two ways:
   a) Distributed Capacitances: In the case, the interconnection capacitances are distributed over all the circuit;
   b) Concentrated Capacitances: Here, the interconnection capacitances are concentrated in the nodes of circuit. The nodes correspond to the inputs and output of logic gates;
8) The results were analyzed and compared.

5. Experimental Results

In this section, we present some power estimation results using the methods described in the previous sections. In Table I, we present some results for a set of standard logic gates: an inverter, AND gates, NAND and NOR with 2 to 4 inputs. The layout symbolic files were generated with TROPIC2 [7]. The interconnection capacitances and transistors were extracted using the CADENCE tools.

In order to confirm the accuracy of the approximations made, we compare the results obtained at logic level with the ones obtained with the HSPICE tool. We evaluated the average power using a sequence of input vectors covering all possible combinations of input transitions. For example, for a circuit with one input there is four possibles transitions and we use the sequence: 0, 0, 1, 1, and 0. For a circuit with two inputs, there are sixteen

possible transitions: 00, 00, 01, 00, 10, 00, 11, 01, 01, 10, 01, 11, 10, 10, 11, 11, and 00. In general, the total number input transition of a circuit with *n* inputs is 2^{2n}.

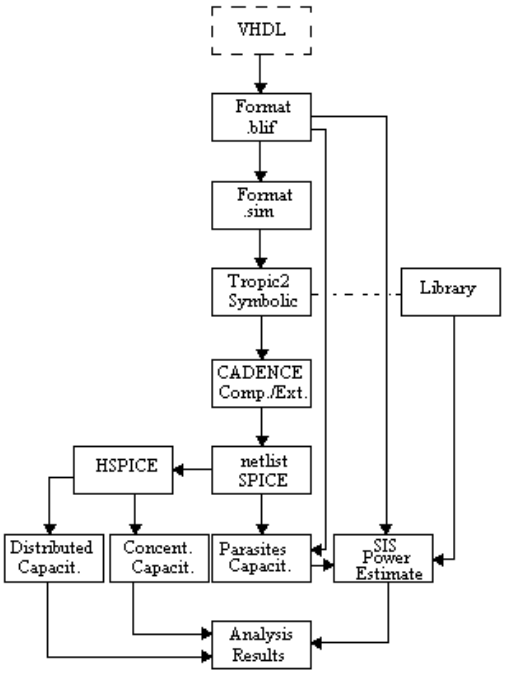


Fig. 5 - Fluxogram of the Experimental Setup

It was done the comparison between the proposed evaluation method and HSPICE results. The simulations were made with transistors using a 0.8 technology. The average power values are in microWatts, assuming a 5V-supply voltage and 20MHz-clock frequency. How example of capacitance file extracted we have the netlist layout file for NAND2 generated since TROPIC2 and CADENCE.

```
* net 0 = gnd!
* net 1 = /b
* net 2 = /a input
* net 3 = /vdd!
c0 2 1 2.15401e-16
c1 3 1 8.81069e-16
c2 3 2 5.53879e-15
c3 0 1 2.54454e-15
c4 0 2 2.0584e-15
c5 0 3 3.60421e-14
.model model2 pmos level=2 vto=-0.7 gamma=0.4
kp=1.5e-05 lambda=0.03 tox=6e-07
m6 1 2 3 3 model2 w=8.01u l=0.745342u ad=1.8e-11
as=1.76e-11 pd=1.5173e-05 ps=1.504e-05
.model model3 nmos level=2 vto=0.7 gamma=0.2 kp=3e-
05 lambda=0.02 tox=6e-07
m7 1 2 0 0 model3 w=8.01u l=0.745342u ad=2.08e-11
as=1.72e-11 pd=1.48e-05 ps=1.392e-05
```

The estimated power is presented below.

```
Node a Cap.=9.749183e-015 sw=0.500000
Power=1.218648e-006
Node b Cap.=8.842158e-015 sw=0.500000
Power=1.105270e-006
Node c{[5069]} Cap.=5.149386e-015    sw=0.380000
Power=4.891917e-007
Total Power:  2.813109e-006
```

The general results of these comparisons are presented in table I and III.

In the second and third column of table I, we show the estimations obtained our approach and with the logic-level estimator of the Power Estimate tool, respectively, without taking into account the interconnection capacitance. Columns 5 and 6 present the estimations made considering the interconnection capacitance. The 4 and 7 columns of Table I show the percentage error between Proposed Method and HSPICE. In the Table II we present the characteristics of five circuits MCNC '91benchmark: C17, cm138a, cm42a, decod and majority. In the Table III, we show the average power estimation and error. The circuits analyzed are limited to two conditions: number inputs and the number of logic levels. The first condition has to do with the number of input combinations that we have to simulate in HSPICE. The second is related to the fact that we are still not computing the delays of the logic gates with our method and therefore we can not compute an accurate glitching power.

The small errors of the evaluations presented in the last column encourage us to continue the investigation for larger circuits.

Name	**Without Interc.**			**With Interc.**		
	HSP	Prop. Meth.	Error (%)	HSP	Prop. Meth.	Error (%)
Inv	0,139	0,153	10,0	1,680	1,585	-5,6
Nand2	0,254	0,282	11,0	2,880	2,813	-2,3
Nand3	0,368	0,367	-0,2	4,758	4,608	-3,1
Nand4	0,419	0,410	-2,1	6,160	6,096	-1,0
nor2	0,275	0,298	8,3	3,220	3,205	-0,4
nor3	0,378	0,365	-3,4	4,388	4,223	-3,7
nor4	0,441	0,418	-5,2	5,874	5,686	-3,2

Table I – Power consumption of conventional logic gates.

Name	# Gates	#Trans	#nodes	# interc. Capac.
C17	6	24	19	60
Cm138a	13	96	56	231
Cm42a	16	100	56	241
Decod	22	146	80	288
Majority	7	46	30	124

Table II – Characteristics of the Circuits Benchmarks Analyzed

	Average	Power	(μW)	
Name	Level Logic	HSPICE Approx.(1)	Prop. Method (2)	Error (%)
C17	3	32.54	30.21	-7,16
Cm138a	3	99.40	96,94	-2,47
Cm42a	3	120.57	114.72	-4,85
Decod	3	234.24	221,77	-5,32
Majority	4	64.91	58.49	-9,89

Table III – Evaluations of Power Consumption for the MCNC'91 Benchmarks Circuits

6. Conclusion and Future Work

We presented an accurate method for capacitance modeling by computing equivalent concentrated capacitances at the external nodes of logic. The model is able to take into account the capacitance of internal nodes and the interconnect capacitance extracted from layout. The results of experimental application to standard logic gates validated the proposed procedure. With MCNC'91 benchmark circuits, the comparison between circuits with a number of logic levels less than three was done. For logic with more than three levels, delay must be considered in power a consumption evaluation. The next step of this research will consist in taking account of the delay of each logic gate and switching activity with variable probability.

References

[1] R. Burch, , F. Najm, P. Yang and T. N. Trick. A Monte Carlo Approach for Power Estimation. IEEE Transactions on Very Large Scale Integration, vol. I, no. 1, march 1993, pp. 63 – 71.

[2] A. Ghosh, S. Devadas, K, Keutzer and J. White. Estimation of Average Switching Activity in Combinational and Sequential Circuits, in Proceedings**:** 29th ACM/IEEE Design Automation Conference, June 1992, pp. 253-259.

[3] J. Monteiro e S. Devadas. Computer-Aided Design Techniques for Low Power Sequential Logic Circuits, Ed. Kluwer Academic Publishers, USA, 1997.

[4] S. Iman and M. Pedram. Logic extraction and Factorization for Low Power. In: 32 nd Design Automation Conference. Proceedings: IEEE Computer Society Press, June 1995, p. 248-253.

[5] L. Glasser and D. Dobberpuhl, *The Design and Analysis of VLSI Circuits*. Reading, MA: Addison-Wesley, 1985.

[6] J. P. Uyemura Circuit Design for CMOS VLSI, Ed. Kluwer Academic Publishers, USA, 1992.

[7] F. Moraes, N. Azemard , M. Robert and D. Auvergne. Flexible Macrocell Layout Generator, 4th ACM/SIGDA Physical Design Workshop, Los Angeles (USA), 1993, pp. 105-116.

[8] F. Najm, Transition Density: A New Measure of Activity in Digital Circuits, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 12, No. 2, Feb. 1993, pp. 310-323.

[9] F. Najm, A Survey of Power Estimation Techniques in VLSI Circuits, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 2, Dec 1994, pp. 446-455.