Fast Power Loss Calculation for Digital Static CMOS Circuits

S. Gavrilov, A. Glebov, S. Rusakov
Russian Academy of Sciences, Moscow, Russia.
D. Blaauw, L. Jones, and G. Vijayan
Motorola Inc., Austin, Texas

Abstract: *In this paper, we present a new dynamic power estimation method that produces accurate power measures at considerably faster run times. The approach uses an enhanced switch-level simulation algorithm that takes into account both short-circuit power and charge-sharing power effects. In benchmarks against a popular commercial power simulation tool, our approach yields power measurements on average within 3% of the commercial solution, while taking between 15 to 20 times less CPU time.*

1. Introduction

Fast and accurate power estimation is a critical tool in design and optimization of low power circuits. Over the last three years, low power IC design using static CMOS circuits has been of particular interest [1]. There are two broad approaches for power estimation of CMOS circuits. In the first approach, the power is computed based on switching frequencies at primary inputs. This approach was used at the logic synthesis stage for delay [2] and power [3,4] optimization of CMOS circuits. In this approach statistical switching probabilities at the inputs are propagated to the other nodes in the circuit. It has the advantage that it does not require explicit simulation patterns and that it is extremely fast. However, the probabilitic approach is limited in its accuracy to the extent of the accuracy of modeling spacial and temporal switching correlations. Also, it does not take into account for nodes that switch partially between 0 and VDD volts.

The second approach, which is considered in this paper, uses dynamic simulation of the circuit using input vectors. Since the actual switching times of the signals are available, this method takes into account the correlations between the signal switchings. Also, the switching of nodes internal to the gate are considered, as well as nodes that switch partially between 0 and VDD volts. The simulation-based power estimation approach therefore produces a much tighter estimate of power loss than the probabilitic approach. The disadvantage of this approach is that the run time tends to be much higher, since a full simulation of the input vectors is performed. This is particularly a problem when power estimation is performed repeatedly in the inner loop of an optimization tool.

To select a proper method of electrical simulation, a number of simulation techniques were considered. Clearly, detailed SPICE-like circuit simulation is too time consuming. Various simplified modifications of detailed circuit simulation were examined, including the ELOGIC algorithm [5]. However they were found unsuitable either from a speed or accuracy point of view.

Switch-level simulation [6,7] is a very fast simulation approach, but has the limitation that it only treats signals as logic zero and one and does not calculate accurate delays. Power estimation using standard switch-level simulation was reported in [8] using a commercially available simulator. However, exact values of node potentials were not calculated, and short-circuit power was not estimated.

The importance of obtaining the exact values of node potentials for power estimation is a consequence of the fact that only part of the nodes in static CMOS circuits are switching between VSS and VDD potentials. The rest of the nodes are switching between some intermediate potentials due to either a V_t drop across transistors or charge sharing between nodes that are isolated from VDD and GND. For example, in the 3-input NAND gate shown in Figure1, supposing a supply voltage of 5V and identical n- and p-transistors with threshold voltage of 1V, node 1 is switching between 0V and 4V due to a V_t drop across a n-transistor, and node 2 is switching between 0V and 2V due to charge sharing between nodes 1 and 2. In a standard switch-level simulator, all nodes would have switched between logic zero and one. Therefore, power estimation without taking into account exact node potentials may result in a significant error.

To obtain an efficient and accurate power estimation tool, we have extended switch-level simulation, such that we obtain accurate delay and voltage potentials needed for power estimation while maintaining the performance of the simulation. In benchmarks against a commercial power simulation solution, our approach performed more than one magnitude faster with comparable accuracy.

ED&TC '97 on CD-ROM

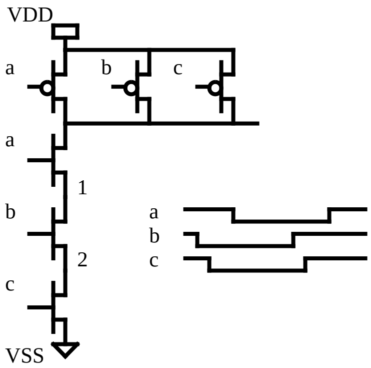


Figure 1. 3-input NAND gate and typical test.

2. Power Calculation

A circuit switching is defined as all processes in the circuit resulting from simultaneous switchings of one or more inputs of circuit fragment. The switch-level simulation for a single circuit switching results in old and new voltage potentials for every circuit node (i.e. the potential in the stationary state before and after switching), and also the time and duration of the switching for every circuit transistor. When taking into account both capacitive recharging and short-circuit power loss, energy loss in static CMOS circuit for one circuit switching is given by the formula:

$$E = \frac{1}{2}\sum_i C_i\left(U_i^2 - V_i^2\right) + V_{dd}\sum_j C_j(V_j - U_j) + V_{dd}Q \tag{1}$$

where Vdd is supply voltage, U_j and V_j are the old and new voltage potential of the j-th node. The first sum is over all circuit nodes and the second sum over circuit nodes that are dc-connected through conducting transistors with VDD after switching. The last term corresponds to short-circuit power loss, and the rest of the formula gives the power loss when neglecting short-circuit. The variable Q is the total charge flowing during one switching through short-circuit transistors and is formulated as follows:

$$Q = \int\left(\sum_m J_m\right)dt$$

where integration is over the time of switching, J_m is the current through the m-th transistor, and the sum is over all short-circuit transistors. Formula (1) is derived as follows.The instantaneous power loss in one transistor connected by k-th and l-th nodes is given by:

$$P_{kl} = (V_k - V_l)I_{k(l)} \tag{2}$$

where V_k is the potential of k-th node and $I_{k(l)}$ is the current flowing from k-th to l-th node. When summing equation (2) over all circuit transistors, one obtains:

$$P = \sum_k V_k I_k + V_{dd}I_{dd} \tag{3}$$

where the sum is over all circuit nodes except the VDD-node, I_k is the full current flowing from the k-th node through conducting transistors and I_{dd} is the full current flowing through the VDD-node into the circuit (i.e. power supply current). We assume that all circuit inputs with potential V_{dd} are also connected to the VDD-node. Given a grounded capacitance C_k at k-th node, we can write:

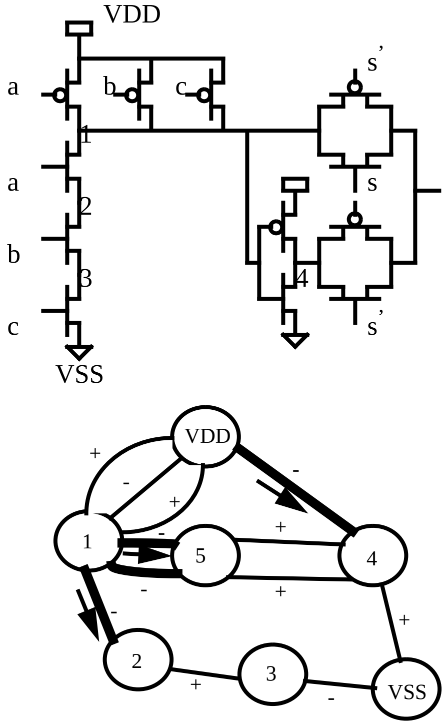
$$I_k = -C_k\frac{dV_k}{dt} \tag{4}$$

We can identify several processes in a DCCC (dc-connected component) during a single circuit switching. Some transistors are turning on while other transistors are turning off. In the stationary state after the circuit switching is completed, there is a part of the DCCC (set A of nodes), that are dc-connected with the VDD-node through conducting transistors. The rest of the DCCC (set B of nodes) is dc-isolated from the VDD-node. We consider the circuit graph with vertices for circuit nodes and edges for transistor channels. According to the stationary state after one circuit switching, we divide the circuit graph into two subgraphs: subgraph A generated by set A of nodes, and subgraph B generated by set B of nodes.

An example of a DCCC and its corresponding circuit graph is shown in Figure 2. The state after a specific circuit switching is shown. The conducting and non-conducting transistors are marked by plus and minus signs respectively. Subgraph A contains the nodes {VDD,1} and subgraph B contains the nodes {VSS,2,3,4,5}. There are also transistors in the circuit connecting subgraphs A and B. We call them short-circuit transistors. In the example of Figure 2, the four transistors shown by heavy lines are short-circuit transistors. We now give a rigorous definition for dividing the total power into two parts:

I. Capacitive recharging energy (CR-energy) for one circuit switching is the total energy loss calculated supposing that all short-circuit transistors are switching instantly at the beginning of a circuit switching.

II. Short-circuit energy (SC-energy) for one circuit switching is the total or actual energy minus CR-energy.



(a,b,c,s): (0,1,0,1) -> (0,1,0,0)

Figure 2. Example of a DCCC and its corresponding circuit graph.

To calculate the energy loss for one switching, the equation (3) should be integrated over the time of switching. Integration of the first term in (3), taking into account (4), gives: $\int\left(\sum_k V_k I_k\right)dt = \frac{1}{2}\sum_k C_k(U_k^2 - V_k^2)$ (5)

where U_k, V_k are old and new potential of k-th node respectively, and the sum is over all circuit nodes except the VDD-node. The total charge, that flows during the switching into the circuit through VDD-node is equal to the total charge increment in capacitances of all nodes dc-connected with VDD-node, i.e., all nodes of subgraph A except VDD-node, plus the total charge that flows from subgraph A through short-circuit transistors.Therefore, integration of the second term in (3) gives

$$\int V_{dd} I_{dd} dt = V_{dd}\sum_l C_l(V_l - U_l) + V_{dd}Q \qquad (6)$$

where the sum is over circuit nodes dc-connected with the VDD-node through conducting transistors, and the formula for Q is given above. By adding (5) and (6), one obtains formula (1).

3. Short-Circuit Power Calculation

To calculate SC-energy for one switching, we should identify the short-circuit transistors for this switching, and then calculate the total charge flowing through these transistors from subgraph A. It should be noted that short-circuit energy, as defined above, may be negative in some cases. This means that the energy stored by the node capacitances before the switching, is partially recycled during the switching. In the example of circuit switching given by Figure 2, the direction of current through short-circuit transistors are shown by arrows. For the circuit switching shown in Figure 2, SC-energy is positive, however, for another switching with the same final state, (0,0,0,1) -> (0,1,0,0), the SC-energy may be negative. This corresponds to partial recycling energy stored by nodes 1,4,5.

When calculating short-circuit energy, we process every switching of a gate input separately. It can be easily shown that, according to the definition of section 3, SC-energy is zero when a gate output is not switching. If a gate input switching results in an output switching, then Q in the formula for SC-energy is the total charge that flows through the transistor which are turning to a non-conducting state. If we neglect capacitances of all gate internal nodes, and account only for capacitance of gate output node, then we should simply calculate the average resistances of pull-up and pull-down networks during switching. We define R as the maximum resistance of a path in the pull-up network, connecting VDD-node and the switching transistor, r_0 as the resistance of the switching pull-up transistor, r_1,... as the resistances of the conducting transistors composing a path from the switching transistor to the gate output, C_i as the tree capacitance, i.e. the maximal capacitance of a tree that can be switched by switching of i-th transistor. We now formulate the SC-energy of one gate switching as follows:

$$E_{SC} = \frac{K\tau V_{dd}^2}{\tau_1 R_1}\left(d - \tau + \tau e^{-\frac{d}{\tau}}\right) \qquad (7)$$

where d is the input slope. If the output is rising, then $R_1 = R_n$ and $\tau_1 = \tau_p$, otherwise $R_1 = R_p$ and $\tau_1 = \tau_n$, where $\tau_n = R_n C$, $\tau_p = R_p C$, $\tau = (1/(1/R_n + 1/R_p))C$,

C is the output node capacitance, R_p and R_n are the average resistance of the pull-up and pull-down networks, K is the adjustable algorithm parameter.

4. Switch-Level Simulation

In order to calculate power consumption accurately, we have developed an extended version of the switch-level simulation algorithm. In this version, the exact value of voltage potential is calculated for every circuit node including internal nodes, for every stationary state of the circuit. The main distinctions between our algorithm of switch-level simulation and a conventional algorithm of this sort (for example, "SLS" algorithm described in [9]) are:

1. Our algorithm is developed specifically for static CMOS circuits; this results in a simplified way of circuit graph processing (each conducting transistor is typically passed only once and in one direction)
2. Our algorithm is developed specially for power loss calculation; this requires the exact potential calculation for each circuit node instead of using the {0,1,X} set of node states.

We use a simple switch-level model for a MOS-transistor. Each of the transistor contacts (gate, source, drain) is connected to ground through a linear capacitance, depending on transistor size and type. Transistor may be in either a conducting or non-conducting state, depending on the gate potential and transistor type. Power loss, when calculated neglecting short-circuit (CR-power), does not depend on specific behavior of the transistor in the conducting state (linear or non-linear channel resistance, etc.).

Let V_{tn} ,V_{tp} be the threshold voltages. We employ the fact that in every stationary state of a static CMOS circuit the circuit graph decomposes into three subgraphs:

1. Subgraph A, containing nodes dc-connected with VDD-node (potential value either Vdd or Vdd-V_{tn});
2. Subgraph C, containing nodes dc-connected with VSS-node (potential value either 0 or V_{tp});
3. Subgraph D, containing nodes isolated from both VDD and VSS (any intermediate value of potential).

Subgraphs C and D together compose the subgraph B from the previous section. Several nodes from subgraph D, when interconnected and isolated from the rest of nodes of this subgraph, compose what we call a CS-group (Charge Sharing group). During simulation, all circuit switchings are processed sequentially. When processing one circuit switching, only local decisions are made for the nodes from subgraphs A and C. This means that during traversal of the circuit graph, when passing through a conducting transistor, we calculate a new node potential, based only on the new potential of other nodes. Iterations until potential relaxation (repeated traversal of some parts of circuit graph) are needed in the following cases:

1. We apply a degraded voltage level of V_{tn} or VDD-V_{tp} to a node with an already calculated new potential of 0 or VDD;
2. There is a feedback loop within one DCCC (iterations within one DCCC);
3. There is a feedback loop including several DCCCs (iterations across DCCCs).

Once a CS-group is identified, the new potential for it is calculated using charge conservation, supposing instant switching of all transistors to their final states:

$$V = \left(\sum_i C_i U_i\right) / \left(\sum_i C_i\right) \qquad (8)$$

where the sum is over the nodes of CS-group.

When simulating a single switching for a DCCC, the corresponding circuit graph is traversed in the following sequence:

1. All two-terminal pull-down paths from subgraph C;
2. All two-terminal pull-up paths from subgraph A;
3. The rest of subgraph C;
4. The rest of subgraph A;
5. The subgraph D (CS-groups).

When dealing with large CMOS circuits, our algorithm starts with circuit partitioning into DCCCs and ordering DCCCs according to signal propagation [10]. Simulation of the whole circuit is event-driven, with use of global event schedule where an event is defined to be a node switching. When an event occurs at one of DCCC inputs:

1. Switch-level simulation for this DCCC with CR-energy calculation is performed as described above
2. If an event is generated at the DCCC output, then time of this new event is calculated with use of the Elmore delay model [12]
3. Similarly, a slope for the new event is calculated;
4. SC-energy for the DCCC switching is also calculated.

If for some DCCC, two input events are very close in time, then a partial glitch may be generated for the DCCC output. A partial glitch is composed of two events that are very close in time, such that each of these events is not actually a complete switching. In this case, energy loss for the current DCCC is calculated, but the partial glitch is not propagated through the circuit.

5. Switch-Level Simulation Using SP-BDD

Recently a new and effective BDD representation was proposed for digital circuits, which is called the SP-

BDD (Series-Parallel BDD) [12]. A SP-BDD is a special reduced ordered BDD. It gives a canonical representation of classical static CMOS gate with series-parallel and logically complementary pull-up and pull-down networks. A SP-BDD represents not only the boolean function of a gate, but also its topology. Every vertex of gate-BDD corresponds to a couple of transistors (one in pull-up and one in pull-down) with united gate contacts composing gate input. Besides, every vertex except root corresponds to internal node of either pull-up or pull-down network. Output node of a gate has no corresponding vertex in gate-BDD. This representation is convenient for many purposes, such as logic extraction, transistor reordering and logic resynthesis for low power. Basic algorithms for SP-BDD extraction and manipulation are given in [12]. A SP-BDD is a ROBDD (Reduced Ordered BDD [13]) with natural linear ordering of vertices. The power estimation algorithm proposed in this paper has been implemented in a very efficient manner using the SP-BDD circuit representation.

TABLE 1.

Ckt	# of Trans	I-Avg (our algor.)	I-Avg (comm. tool)	Run time(s) (our algor.)	Run time (s) (comm. tool)
c432	804	0.129	0.126	6.3	116
c1355	2308	0.400	0.406	28.6	452
c1908	3482	0.599	0.601	46.6	614
cla	1008	0.185	0.175	6.8	123
cla1	956	0.226	0.227	8.1	136
cla2	956	0.184	0.180	8.0	135
cnt_0	352	0.032	0.030	2.9	58
cnt_1	372	0.034	0.033	3.4	65
cnt_	350	0.028	0.028	3.8	56
cnt_	336	0.028	0.029	3.4	53
fdc	696	0.050	0.055	4.0	42
pra	1138	0.119	0.121	6.1	65

6. Experimental Results

The algorithm described in previous section has been implemented and tested on Unix-workstations. Typical testing results are shown in Table 1, in comparison with similar results obtained for a popular commercial power simulation tool. This commercial tool also performs fast power calculation for CMOS, but uses a more complex transistor model. For all tests shown in Table 1, randomly generated test sequence with length of 2000 test vectors were used. The results demonstrate very good accuracy and efficiency of our algorithm. The difference in average VDD-current between the commercial tool and our algorithm is within a few percent. Notably our algorithm is typically between 15 to 20 times faster. This makes our algorithm very suitable for both fast power estimation for large CMOS circuits and for use within power optimization tools.

7. References

1. A.P.Chandrakasan, S.Sheng, R.W.Brodersen. “Low-Power CMOS Digital Design”, IEEE Journ. of Solid-State Circuits, 1992, v.27, n.4, p.473.
2. B.Kick. “Timing Correction in Logic Synthesis”, Proc. of 1st Int. Conf. “VLSI and Computers”, 1987, p.299.
3. C.Y.Tsui, M.Pedram, A.M.Despain. “Technology Decomposition and Mapping Targeting Low Power Dissipation”, Proc. of 30th DAC, 1993, p.68.
4. V.Tiwari, P.Ashar, S.Malic. “Technology Mapping for Low Power”, Proc. of 30th DAC, 1993, p.74.
5. Y.H.Kim, S.H.Hwang, A.R.Newton. “Electrical-Logic Simulation and Its Applications”, IEEE Trans. on CAD, 1989, v.8, n.1, p. 8.
6. J.P.Hayes. “A Unified Switching Theory with Applications to VLSI Design”, Proc. IEEE, 1982, v.70, n.10, p.1140.
7. R.E.Bryant. “A Switch-Level Model and Simulator for MOS Digital Systems”, IEEE Trans. on Computers, 1984, v.33, n.2, p.160.
8. C.M.Huizer. “Power Dissipation Analysis of CMOS VLSI Circuits by Means of Switch-Level Simulation”, Proc. of 16th European Solid-State Circuit Conf., 1990, p.61.
9. Z.Barzilai, D.Beece, L.M.Huisman, V.Iyengar, G.Zilberman. “SLS - a Fast Switch-Level Simulator”, IEEE Trans. on CAD, 1988, v.7, n.8, p.838.
10. V.B.Rao, T.N.Trick. “Network Partitioning and Ordering for CMOS VLSI Circuits”, IEEE Trans. on CAD, 1987, v.6, n.1, p.128.
11. J.P.Caisso, E.Cerny, N.C.Rumin. “A Recursive Technique for Computing Delays in Series-Parallel MOS Transistor Circuits”, IEEE Trans. on CAD, 1991, v.10, n.5, p.589.
12. A.L.Glebov, D.Blaauw, L.G.Jones. “Transistor Reordering for Low Power CMOS Gates Using an SP-BDD Representation”, Intern. Symp. on Low Power Design, 1995, p.161.
13. R.E.Bryant. “Graph-Based Algorithms for Boolean Function Manipulation”, IEEE Trans. on Computers, 1986, v.35, n.8, p.677.