

# HEAT : Hierarchical Energy Analysis Tool\*

Janardhan H. Satyanarayana and Keshab K. Parhi

Department of Electrical Engineering  
University of Minnesota, 200 Union Street SE  
Minneapolis, MN 55455, USA  
E-mail: {jsatyana, parhi}@ee.umn.edu

## ABSTRACT

This paper presents an algorithm for the estimation of power in static CMOS digital circuits using a stochastic approach. The salient feature of this approach is that it can be used to estimate the power of reasonably large digital circuits in a very short time, due to its *hierarchical* nature. Here, the given circuit is first partitioned into smaller sub-circuits. Then, the sub-circuits are modeled using state transition diagrams (stds), and the steady-state probabilities associated with the various states are computed by treating them as irreducible Markov chains. Finally, the energy associated with each sub-circuit is computed, and the total energy of the circuit is obtained by summing up the energies of its constituent sub-circuits. In the proposed hierarchical approach, the energies associated with various edges in a sub-circuit are calculated only once using SPICE and these values are used several times; this results in large savings in computation time. Another advantage of the proposed approach is that we can accommodate switching activities at the transistor level and not necessarily at gate or higher levels. Experimental results show that the estimated power is in close agreement with the actual power obtained from exhaustive SPICE simulations, but the computation time required by the proposed approach is orders of magnitude less than that of SPICE.

## 1 INTRODUCTION

The rapid advancement in semiconductor technology in the last decade has made possible the integration of a large number of digital CMOS circuits on a single chip. Moreover, the desirability of portable operations of these circuits has necessitated the development of low power technology. This is because power consumption of digital circuits has a direct bearing on the life-time of the batteries. For example, if the power consumption of a digital circuit is reduced by a factor of two, the same number of batteries can be used for twice the number of hours. Even in the case of non-portable

systems, reduction in power consumption can greatly cut cooling costs.

In a typical CMOS digital circuit with negligible leakage current, power is dissipated only when there is a transition from zero to one (or one to zero) at the nodes constituting the circuit. The average number of transitions occurring at any given node in a circuit determines the switching activity of that node. In fact, it has been shown in [5] that the switching activity is a measure of the *stress* that can cause failures in digital circuits. Therefore, a major focus of low power design has been to reduce the switching activity of nodes in a given digital circuit. The problem of determining when and how often transitions occur at a node in a general digital circuit is difficult because it is highly input dependent. As a result, researchers have resorted to stochastic approaches, where the probability of transition of all nodes in the circuit is estimated [3]-[4], [6]-[11], [13]-[15], [17].

This paper presents a hierarchical approach for power estimation of digital circuits at the transistor-level. This is achieved by decomposing the digital circuit into sub-circuits, with each sub-circuit performing a particular operation. For example, a typical Booth [2] multiplier is decomposed into three types of sub-circuits; full-adders, encoders, and multiplexors. Each sub-circuit is then modeled with the help of a std, facilitated through the development of state-update equations for each node in the sub-circuit. Then, the energy associated with each edge in the std is computed using SPICE, and the total energy of a given sub-circuit is computed by summing the energies of all the edges in its std. This procedure is repeated for all the sub-circuits, and the final energy of the digital circuit is computed by summing the energies of the constituent sub-circuits. An estimate of the average power is then obtained by finding the ratio of the total energy to the total time over which it was consumed. The proposed approach greatly reduces the number of states in the std, thereby reducing the computation time by orders of magnitude.

The organization of this paper is as follows. Section 2 is concerned with the basic definitions and terminologies used throughout the paper. An algorithm for the proposed hierarchical approach to power estimation is presented in Section 3. A CAD tool called *HEAT* has been developed based on the proposed hierarchical approach and tested on various digital circuits. The results thus obtained are presented in Section 4. Finally, the main conclusions of the paper and future work are summarized in Section 5.

## 2 THEORETICAL BACKGROUND

Let a signal  $\hat{x}(t)$ ,  $t \in (-\infty, +\infty)$ , be a stochastic process [12] which makes transitions between logic zero and logic one at random times. A logic signal  $x(t)$  can then be thought of as a sample of the stochastic process  $\hat{x}(t)$ , i.e.,  $x(t)$  is

\*This research was supported in parts by the Office of Naval Research under contract number N00014-91-J-1008 and AT&T.

one of an infinite number of possible signals that make up the family  $\hat{x}(t)$ . In this paper, it is also assumed that the input processes are *strict-sense stationary* [12] implying that its statistical properties are invariant to a shift in the time origin.

In this section we redefine some discrete-time probabilistic measures, which will be used throughout the paper. The digital CMOS circuits under consideration are assumed to be operating in a synchronous environment, i.e., they are being controlled by a global clock. Let  $T_{clk}$  denote the clock period, and  $T_{gd}$  denote the smallest gate delay in the circuit. To capture the glitches in the circuit, the clock period is assumed to be divided into  $S$  time-slots [8], where

$$S \triangleq \frac{T_{clk}}{T_{gd}}. \quad (1)$$

Then, the probability of a signal  $x_i$  being one is defined as

$$p_{x_i}^1 = \lim_{N \rightarrow \infty} \frac{\sum_{n=1}^{N \times S} x_i(n)}{N \times S} \quad (2)$$

where,  $N$  represents the total number of clock cycles, and  $x_i(n)$  is the value of the input signal  $x_i$  between the time-slots  $n$  and  $n + 1$ . Consequently, the probability that the signal  $x_i$  is zero is

$$p_{x_i}^0 = 1 - p_{x_i}^1. \quad (3)$$

Let us assume that the signal  $x_i$  makes a transition from zero to one in successive time-slots. Then, the probability associated with this transition is defined as

$$p_{x_i}^{0 \rightarrow 1} = \lim_{N \rightarrow \infty} \frac{\sum_{n=1}^{N \times S} x_i(n)x_i(n+1)}{N \times S}. \quad (4)$$

The other transition probabilities can be obtained in a similar manner. It is easy to verify that

$$p_{x_i}^{0 \rightarrow 1} + p_{x_i}^{1 \rightarrow 0} + p_{x_i}^{1 \rightarrow 1} + p_{x_i}^{0 \rightarrow 0} = 1, \quad (5)$$

and

$$p_{x_i}^{0 \rightarrow 0} + p_{x_i}^{1 \rightarrow 0} = p_{x_i}^0 \quad (6)$$

$$p_{x_i}^{0 \rightarrow 1} + p_{x_i}^{1 \rightarrow 1} = p_{x_i}^1. \quad (7)$$

The conditional probabilities can be easily derived from the transition probabilities [10], where for example

$$p_{x_i}^{1/0} = \frac{p_{x_i}^{0 \rightarrow 1}}{p_{x_i}^{0 \rightarrow 1} + p_{x_i}^{0 \rightarrow 0}} \quad (8)$$

represents the probability that  $x_i(n+1) = 1$  given that  $x_i(n) = 0$ .

The signal characteristics can be completely determined once the conditional or transition probabilities are known.

### 3 HIERARCHICAL APPROACH TO POWER ESTIMATION

This section presents a hierarchical approach for power estimation of digital circuits. The salient feature of this approach is that it can be used to estimate the power of large

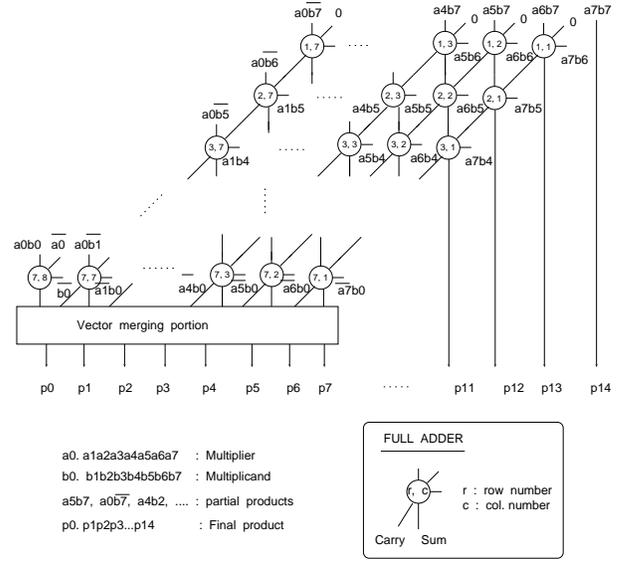


Figure 1: 8x8-b Baugh-Wooley multiplier

digital circuits including multipliers, dividers etc. in a short time. Consider the example of a 8x8-b Baugh-Wooley (BW) [1] multiplier array shown in Fig. 1. The array is treated as an interconnection of sub-circuits (full-adders) arranged in rows and columns. The energy of the entire circuit is then computed by summing the energies of the individual sub-circuits. The steps in the proposed approach are summarized in the following algorithm.

#### Algorithm

INPUT: # of rows, cols in the circuit, type of sub-circuits, parameters, i.e., signal, conditional probabilities of all input signals

OUTPUT: Estimated average power

est\_power () {

total\_energy = 0;

for r = 1 to rows

for c = 1 to cols

1. model the sub\_circuit(r,c) by using a std;

2. compute steady-state probabilities from the input signal parameters of sub\_circuit(r,c), by treating the std as an irreducible Markov chain;

3. compute energy(r,c) associated with the edges of the std using SPICE; /\* this step has to be executed only once \*/

total\_energy = total\_energy + energy(r,c);

4. compute the output signal parameters of sub\_circuit(r,c);

end;

end;

average power =  $\frac{\text{total\_energy}}{\text{time over which the energy was spent}}$  }

□

The remainder of this section is concerned with the implementation of each step in the above algorithm.

#### 3.1 STATE-TRANSITION DIAGRAM MODELING

Here, a systematic approach is presented to model digital circuits using state transition diagrams. The modeling is done by deriving analytic expressions for the state-update

of all nodes in the corresponding digital circuits.

#### A) Static CMOS NOR gate

Consider a typical static CMOS NOR gate shown in Fig. 2, where  $x_1$  and  $x_2$ , respectively, represent the two input signals and  $x_3$  represents the output signal. It is clear from Fig. 2 that there are basically two nodes  $node_2$  and  $node_3$ , which

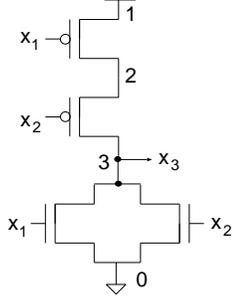


Figure 2: A Static CMOS NOR gate

have their values changing between 1 and 0. The presence of charging/discharging capacitances at these nodes enables us to develop the state-update *arithmetic* equations for the nodes in accordance with

$$node_2(n+1) = (1 - x_1(n)) + x_1(n) * x_2(n) * node_2(n) \quad (9)$$

$$node_3(n+1) = (1 - x_1(n)) * (1 - x_2(n)). \quad (10)$$

The above equations can be used to derive the std for the NOR gate as shown in Fig 3, where for example,  $S_1$  repre-

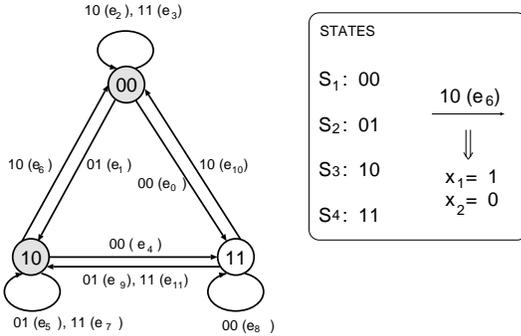


Figure 3: State Transition Diagram for a Static CMOS NOR gate

sents the state with node values  $node_2 = node_3 = 0$ , and the edge  $e_1$  represents a transition (switching activity) from state  $S_1$  to  $S_3$ .

A static CMOS NAND gate has also been modeled by developing state-update equations for the corresponding nodes. It turns out that the state-transition diagram thus obtained is identical to the one given in [8] where it had been obtained in an ad-hoc manner. Finally, a type-0 adder and a type-1 adder [16] have also been modeled using this approach.

#### B) An edge-triggered D flip-flop

One of the common approaches to increase speed in digital circuits is to pipeline it at various levels using D flip-flops. Therefore, it is interesting to consider the ramifications of pipelining on power consumption. To this effect, we propose a model for an edge-triggered D flip-flop which can be used to estimate power consumption in pipelined architectures.

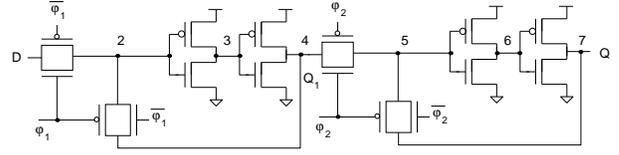


Figure 4: An edge-triggered D flip-flop

Consider an edge-triggered D flip-flop as shown in Fig. 4. Here,  $D$  represents the input signal,  $Q$  represents the output signal, and  $\phi_{1,2}$  represent the non-overlapping two-phase clock signals. It is clear from Fig. 4 that the D flip-flop can be viewed as a cascade of two identical latches controlled by different clocks. Therefore, for power estimation it is sufficient to model a single latch with the help of a std. The state-update *arithmetic* equations for the first latch are

$$node_2(n+1) = D(n) * \phi_1(n) + (1 - \phi_1(n)) * node_2(n) \quad (11)$$

$$node_3(n+1) = 1 - node_2(n+1) \quad (12)$$

$$node_4(n+1) = node_2(n+1). \quad (13)$$

Using the above equations, the std for the latch is derived as shown in Fig. 5. It is interesting to note that although

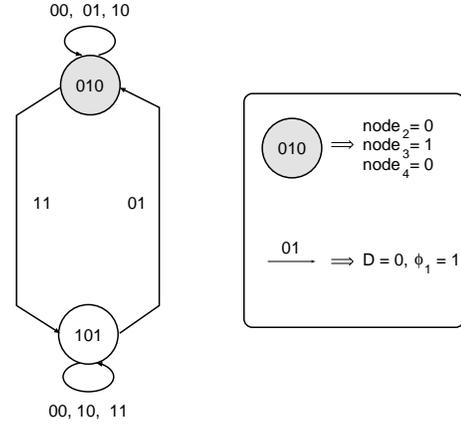


Figure 5: State-transition diagram for a latch

there are three nodes in the latch, there are only two states. Intuitively, this means that the presence of a latch reduces the glitching activity. This corroborates the fact that when sampling frequencies are unaltered, pipelining leads to a reduction in power consumption. The std for the second latch can then be easily obtained by replacing  $\phi_{i1}$  with  $\phi_{i2}$ , and  $D$  with  $Q_1$ .

### 3.2 COMPUTATION OF STATE PROBABILITIES

An approach based on irreducible Markov chains is used to compute the steady-state probabilities of the various states in the std. Consider the std of the CMOS NOR gate shown in Fig. 3. Here, assuming that the input signals  $x_1$  and  $x_2$  are independent, the probabilities  $p(e_j)$  associated with the various edges are computed in accordance with

$$p(e_j | e_j : x_m = q, x_n = r) = (q * p_{x_m}^1 + (1 - q) * p_{x_m}^0) \times (r * p_{x_n}^1 + (1 - r) * p_{x_n}^0) \quad (14)$$

where  $0 \leq j \leq 11$ ,  $1 \leq m, n \leq 2$ , and  $0 \leq q, r \leq 1$ . For example, the probability of edge  $e_1$  in the state transition diagram is  $p(e_1) = p_{x_1}^0 * p_{x_2}^1$ . These edge probabilities are then used to compute the state transition matrix  $\Pi_{nor}$  in accordance with

$$\Pi_{nor} = \begin{bmatrix} p(e_2) + p(e_3) & 0 & p(e_1) & p(e_0) \\ 0 & 1 & 0 & 0 \\ p(e_6) & 0 & p(e_5) + p(e_7) & p(e_4) \\ p(e_{10}) & 0 & p(e_9) + p(e_{11}) & p(e_8) \end{bmatrix} \quad (15)$$

Here, the  $\Pi_{nor_{ij}}$ -th element represents the transition probability from state  $S_i$  to state  $S_j$ , where  $1 \leq i, j \leq 4$ . It is intuitive to observe that the zeros in the matrix represent the non-existence of state  $S_2$ . Having modeled the transition diagram as an irreducible Markov chain, the steady-state probabilities are then computed by solving

$$P_S = P_S * \Pi_{nor} \quad (16)$$

where

$$P_S = [ P_{S_1} \ P_{S_2} \ P_{S_3} \ P_{S_4} ] \quad (17)$$

represents the steady-state probabilities of the different states. A simple approach to solve (16) is to first compute the eigenvalues associated with  $\Pi_{nor}^T$ . Then the normalized eigenvector corresponding to an eigenvalue of 1 (ignoring the trivial case) would be the steady-state probability vector  $P_S$ .

The steady-state probabilities computed by using the above Markov model are then used to compute the edge-activities  $EA_j$  ( $0 \leq j \leq 11$ ) as proposed by Lin et. al.[8]. For example, the edge-activity numbers for the NOR gate are computed in accordance with

$$EA_0 = P_{S_1} * N * S * P(00/10) + \quad (18)$$

$$EA_3 * (P(00/11) - P(00/10))$$

$$EA_1 = P_{S_1} * N * S * P(01/10) + \quad (19)$$

$$EA_3 * (P(01/11) - P(01/10))$$

⋮

$$EA_{11} = P_{S_4} * N * S * P(11/00) \quad (28)$$

where,  $P(11/00)$  for example, represents the conditional probability that  $x_1(n+1) = x_2(n+1) = 1$  given that  $x_1(n) = x_2(n) = 0$ . The error in the edge-activity numbers using the proposed approach was found to be less than 1.5% as shown in Fig. 6. The edges  $e_2, e_3, e_5, e_5, e_7$ , and  $e_8$  have larger activity than the other edges because for this example, the input signals to the NOR gate were allowed to change only in the beginning of each clock-cycle (i.e., once in every  $S$  time-slots). As a result, there is some switching activity only in the first time-slot in every clock-cycle, and the circuit remains idle for the remaining time-slots. However, this will not be the case if two or more NOR gates are cascaded.

### 3.3 EDGE ENERGY COMPUTATION

This section presents an algorithm for the computation of energy associated with each edge in the std using SPICE. The first step in the algorithm is the identification of the initial state, and the sequence of inputs leading to that state. Two flag vectors are defined; one for all the states, and one for all the edges, in the std. A state flag bit is set whenever that state is first encountered. An edge flag bit on the other hand is set whenever that edge is traversed. The variable

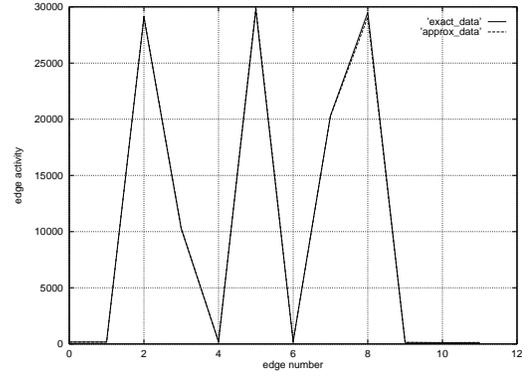


Figure 6: Exact and approximated (using Markov model) edge-activity numbers for a static CMOS NOR gate with  $N=2000$ ,  $S=60$ , and  $T_{clk}=20ns$

$i$  stores the state number, while the variable  $k$  stores the number of the input sequence. For example in Fig. 3,  $i$  can vary from 1 to 4 (corresponding to states  $S_1$  to  $S_4$ ), and  $k$  can vary from 1 to 4 (corresponding to the sequence of inputs 00, 01, 10, 11). A matrix called *edge\_mat* is formed, the rows of which store the sequence of inputs leading to the traversal of an edge in the std. The steps in the algorithm are summarized below.

#### Algorithm

INPUT: std of the sub-circuit; initial state (*init\_state*), number of inputs to the sub-circuit,

*num\_inputs*, initialized *edge\_mat*.

OUTPUT: energy of each edge in the std.

energy\_edge() {

reset state flags and edge flags to zero;

$i = \text{init\_state}; k = 1;$

while(all edge flag bits have not been set)

$m = \text{new state};$

if(edge flag bit corresponding to input  $k$  is not set)

set corresponding edge flag bit;

update *edge\_mat*;

if(flag bit corresponding to state  $m$  is not set)

set corresponding flag bit;

update *edge\_mat*;

$\text{prev\_state}(i) = i;$

$i = m; k = 0;$

end;

end;

$k = k + 1;$

if( $k > 2^{\text{num\_inputs}}$ )

$k = 1;$

$i = \text{prev\_state}(i);$

end;

end; }

/\* a matrix *edge\_mat* with rows containing the sequence of inputs leading to the traversal of edges in the std has been formed \*/

rows = number of rows in *edge\_mat*;

cols = number of columns in *edge\_mat*;

for  $j = 1$  to rows;

run SPICE for input sequence *edge\_mat*( $j, \text{cols}-1$ );

$\text{energy1} = \text{resulting energy};$

run SPICE for input sequence *edge\_mat*( $j, \text{cols}$ );

$\text{energy2} = \text{resulting energy};$

$W_j = \text{energy2} - \text{energy1};$

end;

□

Using the above algorithm, the initial state for the NOR gate shown in Fig. 3 was found to be 11, and the `edge_matrix` was found to be

$$edge\_mat = \begin{bmatrix} 00 & 01 & 10 & 00 \\ 00 & 01 & 10 & 01 \\ 00 & 01 & 10 & 10 \\ 00 & 01 & 10 & 11 \\ 00 & 01 & 00 \\ 00 & 01 & 01 \\ 00 & 01 & 10 \\ 00 & 01 & 11 \\ 00 & 00 \\ 00 & 01 \\ 00 & 10 \\ 00 & 11 \end{bmatrix} \begin{array}{l} \rightarrow e_0 \\ \rightarrow e_1 \\ \rightarrow e_2 \\ \rightarrow e_3 \\ \rightarrow e_4 \\ \rightarrow e_5 \\ \rightarrow e_6 \\ \rightarrow e_7 \\ \rightarrow e_8 \\ \rightarrow e_9 \\ \rightarrow e_{10} \\ \rightarrow e_{11} \end{array} \quad (29)$$

The energy associated with the sub\_circuit is then computed by taking a weighted sum of the energies associated with the various edges in the std representing the sub\_circuit, in accordance with

$$energy = \sum_{j=0}^{\# \text{ of edges } - 1} W_j * EA_j. \quad (30)$$

### 3.4 COMPUTATION OF OUTPUT SIGNAL PARAMETERS

The final step in the hierarchical approach to power estimation is concerned with the computation of the signal parameters at the output of the sub-circuits. This is best illustrated with the help of a simple example. Consider two NOR gates connected in cascade as shown in Fig. 7. Let  $x_1(n)$  and

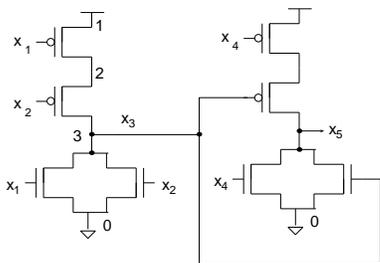


Figure 7: Two NOR gates connected in cascade

$x_2(n)$  represent, respectively, the binary values of the input signals  $x_1$  and  $x_2$  between time instances  $n$  and  $n+1$ . Then, from (10) one can compute the values of the signal  $x_3$  for all  $N \times S$  time-slots. Therefore, once  $x_3(n)$ , and  $x_4(n)$  are well-defined for all time-slots, the signal characteristics for the second NOR gate can be computed. To compute the energy of the second NOR gate using (30), we use the  $W_j$  values calculated previously for the first NOR gate and the new  $EA_j$  values obtained for the second NOR gate.

The above method is easily generalized to multipliers (dividers) which are designed using type-0 or type-1 adders cascaded in a specific manner.

## 4 EXPERIMENTAL RESULTS

A CAD tool called *HEAT* (*Hierarchical energy analysis tool*) has been developed based on the proposed approach. The tool has been developed mainly using C, though it has interfaces with both MATLAB and SPICE. All the experiments

were performed on a SUN SPARC 20 workstation, by generating random values (with probability 0.5) for the input signals. The results thus obtained are presented in Tables 1 and 2.

The entries in the first column in Table 1 represent the various kinds of basic static CMOS digital circuits for which power has been estimated. The entries in the second column represent the average power consumption computed by using both SPICE and *HEAT*, while those in the third column represent the corresponding computational times. The reduction in the number of states in the state transition diagram, obtained by using the proposed algorithm is elucidated in column four. Finally, the entries in the fifth column represent the error in power estimation using the proposed approach. It is clear from these entries that the values obtained by *HEAT* are in close agreement with the actual values obtained by performing exhaustive SPICE simulations. However, the computation time of *HEAT* is orders of magnitude less than that of SPICE.

The hierarchical approach for power estimation is exploited to obtain the results in Table 2. Here, the subscripts for the basic gates (e.g., NAND, NOR) represent the number of cells connected in cascade. For example,  $nand_6$  represents 6 nand-gates connected in cascade. The subscripts for the Baugh-Wooley (BW) and the hybrid (HY) multiplier [16] architectures represent the word length. The HY multiplier is designed by cascading type-1 adders and the architecture is similar to that shown in Fig. 1. As another example, for the  $16 \times 16$  BW and HY multipliers, the power consumption values were estimated to be 716639, and  $90584\mu W$  in approximately 900 and 620 seconds, respectively. It may be noted that these values can be computed using SPICE in about 16 to 20 hours.

The results show that the power consumed by the HY multiplier is much less than that consumed by the BW multiplier.

## 5 CONCLUSIONS

A hierarchical approach for power estimation in static CMOS digital circuits has been presented in this paper. The salient feature of this approach is that it can be used to estimate the power consumption of relatively large digital circuits in a very short time. Based on this approach, a CAD tool called *HEAT* has been developed and implemented in C. Experimental results show that the average power estimated using *HEAT* is very close to that obtained by performing extensive SPICE simulations. The computation time required by *HEAT* however is orders of magnitude less than that required by SPICE. Future work includes extension of this method to dynamic CMOS circuits and to the power estimation of other multipliers.

## References

- [1] BAUGH, C. R., AND WOOLEY, B. A. A two's complement parallel array multiplication algorithm. *IEEE Trans. on Computers C-22* (1973), 1045-1047.
- [2] BOOTH, A. D. A signed binary multiplication technique. *Q. J. Mech. Appl. Math.* 4 (1951), 236-240.
- [3] CHOU, T.-L., ROY, K., AND PRASAD, S. Estimation of circuit activity considering signal correlations and simultaneous switching. In *Proc. ICCAD* (1994), pp. 300-303.

Table 1: Comparison of average power for some basic static CMOS digital circuits,  $T_{clk}=20ns$ ,  $(W/L) = 3$  for all transistors

CMOS circuit	avg. power ( $\mu W$ )		computation time (sec)		max. # of states possible		error
	SPICE	HEAT	SPICE	HEAT	existing	HEAT	
nor	0.9939	0.9845	188	0.5	4	4	-0.94%
nand	1.4490	1.4107	219	0.5	4	4	-2.64%
type-0 adder	33.218	32.12	1289	2.5	$2^{14}$	$2^{14}$	-3.31%
type-1 adder	33.029	31.94	1295	2.5	$2^{15}$	$2^{15}$	-3.29%
D flip-flop	6.234	6.55	876	0.6	8	8	-5.06%

Table 2: Comparison of average power for larger circuits obtained by using the hierarchical approach,  $T_{clk} = 20ns$ ,  $(W/L)=3$  for all transistors

CMOS circuit	avg. power ( $\mu W$ )		computation time (sec)		max. # of states possible		error
	SPICE	HEAT	SPICE	HEAT	existing	HEAT	
$nor_2$	3.4029	3.2567	383	1	16	4	-4.30%
$nor_3$	5.7795	5.5429	571	1.4	64	4	-4.09%
$nor_6$	13.2548	13.8794	1266	3	4096	4	4.71%
$nand_2$	3.7608	3.8205	379	1	16	4	1.56%
$nand_3$	6.4374	6.1255	629	1.5	64	4	-4.85%
$nand_6$	13.7415	13.0554	1400	3.1	4096	4	-4.99%
$mult_{BW_4}$ [16]	980.28	900.34	4959	40	$2^{14*11} + 2^{7*9}$	$2^{14} + 2^7$	-8.15%
$mult_{BW_6}$	2928.3	3102.2	19007	215	$2^{14*27} + 2^{7*20}$	$2^{14} + 2^7$	5.94%
$mult_{BW_8}$	8847.4	8412.3	43438	289	$2^{14*51} + 2^{7*35}$	$2^{14} + 2^7$	-5.17%
$mult_{HY_4}$ [16]	672.49	627.79	4065	30	$2^{15*12}$	$2^{15}$	-6.64%
$mult_{HY_6}$	1297.6	1367.5	10011	115	$2^{15*30}$	$2^{15}$	5.11%
$mult_{HY_8}$	2802.7	3011.5	29052	220	$2^{15*56}$	$2^{15}$	7.45%

- [4] DEVADAS, S., KEUTZER, K., AND WHITE, J. Estimation of power dissipation in CMOS combinatorial circuits using Boolean function manipulation. *IEEE Trans. on Computer-Aided Design* 11, 3 (Mar. 1992), 373–383.
- [5] IYER, R., ROSSETTI, D., AND HSUEH, M. Measurement and modelling of computer reliability as affected by system activity. *ACM Trans. on Computer Systems* 4, 3 (Aug. 1986), 214–237.
- [6] JYU, H.-F., MALIK, S., DEVADAS, S., AND KEUTZER, K. W. Statistical timing analysis of combinatorial logic circuits. *IEEE Trans. VLSI Systems* 1, 2 (June 1993), 126–135.
- [7] KRISHNAMURTHY, B., AND TOLLIS, G. Improved techniques for estimating signal probabilities. *IEEE Trans. on Computers C-38* (July 1989), 1245–1251.
- [8] LIN, J.-Y., LIU, T.-C., AND SHEN, W.-Z. A cell-based power estimation in CMOS combinatorial circuits. In *Proc. ICCAD* (1994), pp. 304–309.
- [9] LIU, D., AND SVENSSON, C. Power consumption estimation in CMOS VLSI chips. *IEEE Jour. of Solid-State Circuits* 29, 6 (June 1994), 663–670.
- [10] MARCULESCU, R., MARCULESCU, D., AND PEDRAM, M. Switching activity analysis considering spatiotemporal correlations. In *Proc. ICCAD* (1994), pp. 292–297.
- [11] NAJM, F. N. Transition density: A new measure of activity in digital circuits. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems* 12, 2 (Feb. 1992), 310–323.
- [12] PAPOULIS, A. *Probability, random variables, and stochastic processes*, 2nd ed. McGraw-Hill, New York, 1984.
- [13] PARKER, K. P., AND MCCLUSKEY, E. J. Probabilistic treatment of general combinatorial networks. *IEEE Trans. on Computers C-24* (June 1975), 668–670.
- [14] POWELL, S. R., AND CHAU, P. M. A model for estimating power dissipation in a class of DSP VLSI chips. *IEEE Trans. on Circuits and Systems* 38, 6 (June 1991), 646–650.
- [15] SATYANARAYANA, J. H., AND PARHI, K. K. A hierarchical approach to transistor-level power estimation of arithmetic units. In *Proc. IEEE International Conf. Acoustic Speech and Signal Processing* (Atlanta, GA, May 1996).
- [16] SRINIVAS, H. R., AND PARHI, K. K. High-speed VLSI arithmetic processor architectures using hybrid number representation. *Journal of VLSI Signal Processing*, 4 (1992), 177–198.
- [17] TSUI, C.-Y., AND ET AL. Power estimation for sequential logic circuits. *IEEE Trans. VLSI Systems* 3, 3 (Sep. 1995).