Sequential Power Estimation using Probability Polynomials

José C. Costa
jccc@algos.inesc.pt

Luís M. Silveira
lms@inesc.pt

José C. Monteiro
jcm@inesc.pt

IST/INESC, 1000 Lisboa, Portugal

Abstract - **In this work, a new method for the power estimation of sequential circuits is proposed. Previously proposed techniques were limited in the size of the circuits they could handle. These methods required that BDDs for the next state lines be built in order to compute the present state lines probabilities. The new technique is based on an approximation approach proposed for combinational circuits in [3], which allows it to be applied to very large circuits. The results show that this technique can be very efficient, achieving less than 6% error in the switching activity estimates, with fast execution times.**

I. Introduction

Power dissipation has become a major parameter for many VLSI designs. Two independent factors have led for this power concern. One factor is the autonomy of portable devices. For a portable device to be successful in the market, it must have long autonomy and small size. Large batteries cannot be used in order to keep the size small, hence the need for low power designs. The other factor as to do with heat dissipation problems. As the feature size reduces, the design density increases, thus increasing the number of power dissipating devices per area. Also, the smaller sizes reduce the propagation delay, allowing for higher clock frequencies, which in turn increase power dissipation.

During the design process, several alternative designs may need to be evaluated and compared. In particular, when targeting low power circuits, a power estimation tool is required to obtain a fast estimation for the different designs.

Power estimation can be done at different levels of design. The higher the abstraction level, the faster the estimation process, yet the lower the accuracy. Gate-level power estimation is generally considered as achieving a good compromising between accuracy and execution time. Although the absolute power estimate must not be very accurate, the estimates are accurate enough in relative terms, permitting a safe comparison of different designs.

Under some generally accepted simplifying assumptions [4], power dissipation at the output of some gate i in a gate-level circuit description can be approximately computed using

$$P_i = \frac{1}{2} \cdot C_i \cdot V_{DD}^2 \cdot f \cdot N_i \tag{1}$$

where V_{DD} is the supply voltage, f is the clock frequency, C_i represents the load capacitance of the gate and N_i is the gate's switching activity, *i.e.*, the average number of gate output transitions per clock cycle. Under this model, the power estimation process reduces to computing the switching activity (N_i) of the gates in the circuit, as the other parameters can be easily extracted from the circuit.

Many different techniques have been proposed to compute the switching activity in combinational logic circuits [5]. There are two major approaches: *simulation-based* techniques, which simulate the circuit with as many input vectors as needed to achieve some predefined accuracy; *probabilistic* techniques, which propagate user-specified input probabilities through the circuit. Probabilistic techniques can in principle be more efficient since they only require the propagation of a single value through the circuit. However, issues such as spatial and temporal correlation may hinder the effectiveness of the methods.

In general, integrated circuits include some storage elements, such as registers, so they present a sequential behavior. A generic sequential circuit is depicted in Figure 1. Estimating power consumption of sequential circuit has the increased difficulty that the probability of the state lines has to be taken into account.

In this work, a new method for the power estimation of sequential circuits is proposed. Previously proposed techniques were limited in the size of the circuits they could handle. The new technique is based on an approximation approach proposed for combinational circuits in [3], which allows it to be applied to very large circuits. The approximation is based on ignoring some convergent signals (thus spatial correlation) if the reconvergence happens after l logic levels, where l is a used-specified parameter. Since it is based on the topology of the circuit, the errors are small. Larger values of l lead to a more accurate estimate, at the expense of a larger computation time and memory requirements. The results show that with $l = 2$, the error in the switching activity estimation is less than 6%, with very fast execution times.

The remainder of this paper is organized as follows. Section II describes the process of estimating power dissipation in sequential circuits. The approximate method for estimating switching activity in combinational circuits is outlined in Section III. The sequential power estimation being proposed is presented in Section IV. Section V presents some results obtained with this technique.

II. Sequential Power Estimation

To obtain accurate power estimates of logic circuits with sequential elements, the probability of the state lines as to be taken into account. As in the case of combinational circuits, both simulation- and probabilistic-based methods have been proposed.

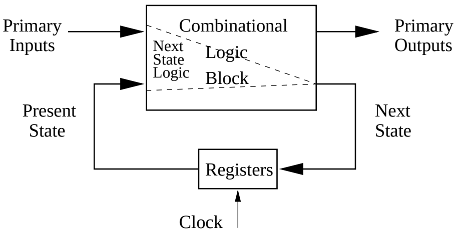


Fig. 1. Generic sequential circuit.

A. *Simulation-based Sequential Power Estimation*

Whereas in the case of combinational circuits, the number of input vectors required to guarantee some maximum error for the switching activity of every node in the circuit can be computed, for sequential circuits that is substantially more difficult since one must guarantee that all the possible states have been visited *in a representative manner*. Basically this means that the fraction of the time that the circuit was in some state during the simulation process has to be proportional to the stationary probability of that state. This is a value that is not known beforehand, may not be possible to compute, and in any case, is what simulation-based methods want to avoid computing.

Still, some techniques have been proposed [6], [2]. The approach is to have more than one simulation in parallel and average among them, so that several different branches of state traversal can be followed. Also, the setup phase (initial part of the simulation during which transitions are not accounted) is larger to allow the circuit to be close to its stationary regime.

B. *Probabilistic Sequential Power Estimation*

For probabilistic methods, the additional difficulty of estimating the power dissipation of a sequential circuit is that the state probabilities have to be computed beforehand. Once this has been done, the process is exactly the same as for combinational circuits, except that the state lines are now treated as primary inputs with probabilities given from the previous step.

In order to do this, one can setup and solve the Chapman-Kolmogorov system of equations [8]:

$$\begin{cases} p_{s_i} = \sum_{\forall j: s_j \rightarrow s_i} p_{PI_{ji}} p_{s_j}, 1 \leq i \leq K-1 \\ \sum_{i=1}^{K} p_{s_i} = 1 \end{cases} \tag{2}$$

where p_{s_i} represents the stationary probability of state s_i and $p_{PI_{ji}}$ the probability of the input condition that triggers a transition from state s_j to state s_i.

The problem is that this approach requires that the state transition graph (STG) has to be extracted from the circuit. For a circuit with N registers, it is possible that the number of states is $K = 2^N$. Therefore, for the majority of the circuits of interest the STG cannot be obtained.

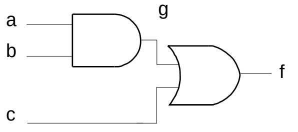
It has been proposed in [8] that the state *line* probabilities be computed instead of the state probabilities. The advantage is that in a circuit with N registers there are only N variables to compute (as opposed to 2^N). The disadvantage is that the spatial correlation among state lines is lost. However, it is shown in [8] that the error incurred with this approximation is less than 5% for all benchmark circuits.

Two different methods are proposed in [8]. The Picard-Peano method simply gets the next state logic block (see Figure 1), builds BDDs for all state lines, propagates the input and state line probabilities using these BDDs to obtain new values for the state line probabilities (using 0.5 as the initial value), repeating this process until all the state line probabilities converge. The Newton-Raphson method also extracts the Hessian matrix from the previous BDDs. The Newton-Raphson method converges in less iterations, but each iteration takes longer than the Picard-Peano.

The method proposed in this paper is based on the Picard-Peano method. However, the method to compute the probabilities does not require BDDs, therefore is applicable to larger circuits.

III. Modeling Switching Activity Using Polynomials

The computation of the switching activity is based on the approximate polynomial propagation method proposed in [3], which is a generalization of the Parker-McCluskey method [7]. This method uses polynomials to model the switching activity of individual gates, handling both spatial and temporal correlations and generic delays.



$$\begin{array}{rl} a: & P_a[0] \\ b: & P_b[0] \\ c: & P_c[0] \\ g: & P_g[0], P_g[1] \\ f: & P_f[0], P_f[1], P_f[2] \end{array}$$

Fig. 2. Unit-delay example circuit.

A. *Exact Method*

The Parker-McCluskey method generates a polynomial that represents the probability that the gate output is 1, for each gate in the circuit. These polynomials are computed from primary inputs to primary outputs in a topological manner. The variables of these polynomials are the input probabilities of the circuit. This method was generalized to work with transition probabilities [1]. In this extension, each input x_i has four probability values corresponding to the input staying low, making a rising transition, making a falling transition, and staying high. Thus, for each gate g, there are four polynomials P_g^{00}, P_g^{01}, P_g^{10}, and P_g^{11}, corresponding to the probability that the gate stays low, makes a rising transition, makes a falling transition, and stays high, respectively. These four polynomials were referred as the *polynomial group* for a gate.

Another important generalization was presented in [3] to handle gate delays. This leads directly to an exact power estimation algorithm, since one just has to sum up the values of appropriate polynomials to obtain the average switching activity at any gate in the circuit. At each output there will be a waveform of polynomials groups, termed a *polynomial waveform*, where each group represents the conditions at the gate output at a particular time instant. The polynomial group at gate g at instant t is denoted as $P_g[t]$. For example, for the simple circuit of Figure 2 with unit gate delays, there will be, for the various signals, the polynomial waveforms indicated in the figure which represent the different time instants where each input/gate can make transitions.

B. *Reducing the Size of Polynomials*

The Parker-McCluskey algorithm and the generalizations described above cannot be used on large circuits, since it involves "collapsing" the circuit into two levels. The method presented in [3] attacks this problem by using an approximation based on limited depth spatial correlation. In this approximation scheme the user specifies one parameter l. This parameter determines the depth in terms of logic levels from each node a that will be searched in order to determine if two paths starting at a will reconverge. Spatial correlation corresponding to two paths starting at a that reconvergence within l logic levels will be accurately taken into account. If reconvergent paths meet after l logic levels then they are assumed to be independent, thus the polynomials will be simplified by variable substitution and some error will be introduced. The rationale behind this approach is that ignoring spatial correlation for signals that reconverge after many levels of logic introduce negligible error.

IV. Sequential Power Estimation Based on Polynomials

The work proposed in this paper extends the method of polynomial simulation presented in Section III to work with sequential circuits. The method being proposed starts by computing the static state line probabilities of the next state logic block (see Figure 1). Using those static probabilities, the transition probabilities for the state lines are

```
nsLogic = Get_Next_State_Logic(Network);
Initialize $p^1_{NS_i} = 0.5 \ \forall i$ in nsLogic;
iter = 0;
do {
  $p^1_{PS_i} = p^1_{NS_i} \ \forall i$;
  Polynomial_Simulation(nsLogic);
  iter++;
} while ($\exists i$:  $p^1_{NS_i} - p^1_{PS_i} > \epsilon$)
for $PS_i \in$ {present state lines in nsLogic} {
  $PS_i = 0$;
  Compute $p_{NS_i}$
  $p^{00}_{NS_i} = (1 - p^1_{PS_i}) \cdot (1 - p^1_{NS_i})$
  $p^{01}_{NS_i} = (1 - p^1_{PS_i}) \cdot p^1_{NS_i}$
  $PS_i = 1$;
  Compute $p_{NS_i}$
  $p^{10}_{NS_i} = p^1_{PS_i} \cdot (1 - p^1_{NS_i})$
  $p^{11}_{NS_i} = p^1_{PS_i} \cdot p^1_{NS_i}$
}
Polynomial_Simulation(Network);
```

Fig. 3. Pseudocode for the approximate sequencial power estimation.

determined. The switching activity for all the nodes in the circuit is then computed. In these three steps polynomial simulation is used.

A. Static Probabilities of the State Lines

The static probabilities of the state lines are computed using the method of Picard-Peano, as described in Section II-B. However, instead of using BDDs for this computation, the new method is based on probability polynomials. Considering that the polynomial simulation method described in the previous section uses transition probabilities, this algorithm has to be modified to handle static probabilities. The way to do this is to assign static probabilities to the inputs of the next state block. Hence, we have the following transition probabilities at the inputs: $p^{00} = p^0$, $p^{01} = p^{10} = 0$, and $p^{11} = p^1$. By assigning the value zero to p^{01} and p^{10}, one guarantees that what will be propagated are static probabilities, and the polynomials resulting from the propagation give the static probabilities. Those polynomials are propagated through the next state block the number of times necessary for the static probabilities of the state lines to converge.

B. Transition Probabilities of the State Lines

The methods described above and in Section II-B can only compute the static state line probabilities. In order to account for temporal correlation, one as to use transition probabilities of the state lines. The transition probabilities consist of four values, p^{00}, p^{01}, p^{10} and p^{11}, which are respectively the probabilities of a signal staying at zero, making a low to high transition, a high to low transition and staying at one. In the case of the next sate logic block, the probability of a signal staying at zero, p^{00}, for example, is the product $(1 - p_{PS})(1 - p_{NS})$, which is equal to say that is the probability of the next state signal being zero knowing that the present state signal is zero.

C. Algorithm

The algorithm for switching activity estimation using polynomials in sequencial circuits is presented in Figure 3. It starts by computing the static probabilities of the state lines using the method by Picard-Peano. In computing the static probabilities of the state lines only the next state logic block is used. The next state logic is obtained by deleting in the original circuit the nodes that are not part of the next state lines support. The iterations start by initializing the present state line probabilities. The probability polynomials are then propagated through the next state logic. The resulting probabilities in the next state lines are then compared with the values in the present state lines. The algorithm will iterate until the difference between all corresponding present and next state lines is bellow ϵ.

TABLE I

STATE LINES STATIC PROBABILITY ERRORS.

Circ.	$l = 0$	$l = 1$	$l = 2$	$l = 3$	$l = \infty$
s1196	0.027	0.015	0.008	0.006	0.002
s1238	0.025	0.016	0.008	0.003	0.001
s3330	0.009	0.007	0.005	0.001	0
s382	0.013	0.002	0	0	0
s386	0.021	0.021	0.019	0.001	0.001
s400	0.013	0.002	0	0	0
s420	0.052	0	0	0	0
s444	0.014	0.002	0	0	0
s499	0.001	0.001	0.001	0.001	0
s510	0.059	0.043	0.035	0.029	0.011
s526	0.006	0.003	0	0	0
s635	0.001	0	0	0	0
s641	0.019	0.019	0.008	0.009	0.007
s713	0.019	0.019	0.007	0.01	0.008
s832	0.017	0.009	0.007	0.003	0
s938	0.026	0.004	0	0	0
s967	0.071	0.067	0.035	0.011	0.001
s991	0.118	0.048	0.004	0	0

The computation of the transition probabilities is the next step in the algorithm. For each state line, i, two passes are made through the next sate logic block. In the first, the input PS_i is set to zero and the other input probabilities have the values that resulted from the first part of the algorithm. The probabilities are propagated from the inputs to the outputs of the next sate logic block and the value of the static probability for NS_i, p_{NS_i}, is obtained. With p_{NS_i} we can get the transition probabilities $p^{00}_{NS_i} = (1 - p_{PS_i})(1 - p_{NS_i})$ and $p^{01}_{NS_i} = (1 - p_{PS_i})p_{NS_i}$ A similar computation is done to get the values p^{10}_{NS} and p^{11}_{NS}. In this case the input PS_i is set to one and after obtaining the value p_{NS_i} the remaining transition probabilities for NS_i, $p^{10}_{NS} = p_{PS_i}(1 - p_{NS_i})$ and $p^{11}_{NS} = p_{PS_i}p_{NS_i}$

The resulting transition probabilities are then applied at the primary inputs of the circuit. Finally, the probability polynomials are propagated through the original circuit, thus computing the switching activity of the circuit.

V. EXPERIMENTAL RESULTS

This section presents power and switching activity estimation results obtained using the approximation algorithm based on limited depth reconvergent path analysis applied to sequencial circuits. Results are presented for different values of l and are compared with the value obtained with the method of [8].

The results presented in Table I simply compare the static probabilities of the state lines obtained with the approximate polynomial simulation and the exact method using BDDs. The table shows the average of the difference of the probabilities for each state line. Values for l equal to 0, 1, 2, 3 and ∞ are presented. As it can be observed, the probability errors are very low, and reduce significantly with l.

Table II presents the power estimation results obtained with the approximation algorithm, the % difference to the method of [8] and the CPU time (in seconds) taken by this computation. A general delay model was used for all the examples and a supply voltage of 5V and clock frequency of 20MHz was assumed. A probability of 0.25 was used for all primary input events. An “NA” entry indicates that either the memory or CPU time limits were exceeded. At the bottom of the table is indicated the maximum and average error over all the circuits, for each value of l. The average error is relative to only those circuits

TABLE II

POWER ESTIMATION RESULTS.

Circ.	Symbolic		$l = 0$			$l = 1$			$l = 2$			$l = 3$			$l = \infty$		
Name	P	t	P	%	t	P	%	t	P	%	t	P	%	t	P	%	t
s1196	2971	187	2898	2.5	27	2911	2.0	34	2924	1.6	42	2926	1.5	390	NA		
s1238	2992	153	2958	1.1	28	2957	1.1	36	2961	1.0	38	2971	0.7	199	NA		
s1269	NA		4799	-	67	4170	-	137	4907	-	484	NA			NA		
s1423	NA		2039	-	78	1837	-	114	1852	-	126	1764	-	248	NA		
s1512	NA		1603	-	611	1365	-	422	1346	-	491	NA			NA		
s3271	NA		10285	-	120	13173	-	203	14375	-	230	NA			NA		
s3330	3221	464	3371	4.7	376	3279	1.8	508	3293	2.2	520	3242	0.7	649	NA		
s382	334	8	327	2.0	13	314	6.1	12	316	5.2	13	316	5.5	21	316	5.5	21
s386	454	6	426	6.2	8	438	3.6	9	432	4.9	8	431	5.2	102	427	5.9	17
s400	345	9	346	0.3	16	324	6.0	17	324	6.2	14	323	6.4	69	323	6.4	20
s420	327	8	745	127.7	49	279	14.7	7	285	13.0	7	285	13.0	7	285	13.0	249
s444	324	8	323	0.1	13	303	6.4	16	302	6.6	13	302	6.8	20	302	6.8	20
s499	375	15	375	0.0	12	390	4.0	15	390	3.8	15	389	3.8	24	390	3.8	204
s510	927	13	1391	50.1	19	952	2.7	21	952	2.8	22	949	2.4	87	1003	8.3	158
s526	281	10	281	0.1	14	268	4.7	16	264	6.1	16	264	6.0	23	264	6.0	32
s635	178	24	180	0.8	14	173	2.9	18	174	2.5	18	174	2.5	18	174	2.5	172
s641	649	191	708	9.0	12	670	3.2	16	670	3.3	17	662	1.9	18	NA		
s6669	NA		33884	-	369	32708	-	808	39823	-	904	NA			NA		
s713	694	170	742	7.0	12	703	1.3	17	711	2.5	18	708	2.1	19	NA		
s832	921	22	987	7.1	19	930	1.0	21	932	1.2	21	933	1.3	81	928	0.7	699
s938	483	20	1592	229.4	109	435	10.0	84	457	5.4	18	455	5.8	22	NA		
s967	1161	44	1530	31.8	48	1349	16.1	115	1306	12.5	92	1206	3.8	5284	1206	3.8	1702
s991	4034	175	2956	26.7	29	3854	4.5	44	3758	6.8	47	NA			NA		
			mx= 229.4, **av**= 20.6			**mx**= 16.1, **av**= 6.2			**mx**= 13.0, **av**= 5.9			**mx**= 13.0, **av**= 5.2			**mx**= 13.0, **av**= 5.7		

TABLE III

SWITCHING ACTIVITY ERRORS.

Circ.	$l = 0$		$l = 1$		$l = 2$		$l = 3$		$l = \infty$	
Name	Max	Avg	Max	Avg	Max	Avg	Max	Avg	Max	Avg
s1196	0.31	0.03	0.36	0.02	0.32	0.02	0.17	0.01	NA	
s1238	0.38	0.02	0.4	0.02	0.31	0.02	0.1	0.01	NA	
s3330	0.41	0.04	0.42	0.03	0.43	0.03	0.38	0.02	NA	
s382	0.2	0.03	0.19	0.02	0.18	0.02	0.18	0.02	0.18	0.02
s386	0.3	0.03	0.14	0.02	0.14	0.02	0.26	0.02	0.3	0.02
s400	0.2	0.03	0.19	0.02	0.26	0.02	0.26	0.02	0.26	0.02
s420	0.5	0.21	0.25	0.04	0.25	0.03	0.25	0.03	0.25	0.03
s444	0.2	0.03	0.19	0.02	0.26	0.02	0.26	0.02	0.26	0.02
s499	0.7	0.03	0.68	0.03	0.68	0.03	0.68	0.03	0.67	0.03
s510	0.5	0.13	0.22	0.04	0.2	0.03	0.14	0.03	0.19	0.03
s526	0.27	0.03	0.17	0.01	0.17	0.01	0.17	0.01	0.17	0.01
s635	0.1	0	0.11	0	0.1	0	0.1	0	0.1	0
s641	0.29	0.03	0.24	0.02	0.34	0.03	0.32	0.03	NA	
s713	0.26	0.03	0.23	0.03	0.27	0.02	0.27	0.02	NA	
s832	0.4	0.03	0.14	0.02	0.14	0.02	0.26	0.02	0.1	0.01
s938	0.5	0.22	0.25	0.02	0.25	0.01	0.25	0.01	NA	
s967	0.61	0.08	0.49	0.05	0.47	0.04	0.3	0.04	0.34	0.03
s991	1.31	0.24	0.8	0.1	0.7	0.11	NA		NA	
Max	1.31	0.24	0.8	0.1	0.7	0.11	0.68	0.04	0.67	0.03
Avg	0.41	0.07	0.3	0.03	0.3	0.03	0.26	0.02	0.26	0.02

for which there are results for all values of l. It can be observed that, with the exception of $l = \infty$, these values decrease with l. For $l = \infty$, the average value increses but that is due to only two circuits. For the rest of the circuits, the error for $l = 3$ is equal to $l = \infty$, which means that for these circuits there is no advantage in taking into account correlation for paths that reconverge after three or more logic levels. For $l = 0$, we can see that the confidence in the results is very low. But the error decreases significantly for $l = 1$. Note that for $l = 2$ results were obtained for all the circuits tested, with an average error below 6%. This average value could be lower if all the circuits were taken into account and not only those for which there are results for all values of l.

Table III presents the maximum and average error for the switching activity estimation over all the individual signals of each circuit. The average was computed by summing the absolut value of the switching probability error relative to the symbolic simulation method for all signals and dividing by the total number of signals. At the bottom of the table the average and maximum of the values for each column are indicated. It can be observed that the average and maximum switching activity error decreases with l. However, the switching activity error may present significant errors. Yet, since the average error is low, the number of nodes with high error is clearly small.

REFERENCES

[1] D. Cheng. *Power Estimation of Digital CMOS Circuits and the Application to Logic Synthesis for Low Power*. PhD thesis, University of California at Santa Barbara, December 1995.

[2] T-L. Chou and K. Roy. Statistical Estimation of Sequential Circuit Activity. In *Proceedings of the International Conference on Computer-Aided Design*, pages 34–37, November 1995.

[3] J. Costa, J. Monteiro, and S. Devadas. Switching Activity Estimation using Limited Depth Reconvergent Path Analysis. In *Proceedings of the International Symposium on Low Power Electronics and Design*, pages 184–189, August 1997.

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

[5] F. Najm. A Survey of Power Estimation Techniques in VLSI Circuits (*Invited Paper*). *IEEE Transactions on VLSI Systems*, 2(4):446–455, December 1994.

[6] F. Najm, S. Goel, and I. Hajj. Power Estimation in Sequential Circuits. In *Proceedings of the* 32^{nd} *Design Automation Conference*, pages 635–640, June 1995.

[7] K. Parker and E. McCluskey. Probabilistic Treatment of General Combinational Networks. *IEEE Transactions on Electronic Computers*, C-24(6):668–670, 1975.

[8] C-Y. Tsui, J. Monteiro, M. Pedram, S. Devadas, A. Despain, and B. Lin. Power Estimation Methods for Sequential Logic Circuits. *IEEE Transactions on VLSI Systems*, 3(3):404–416, September 1995.