Exact and Approximate Methods for Calculating Signal and Transition Probabilities in FSMs*

Chi-Ying Tsui, Massoud Pedram, Alvin M. Despain
Department of Electrical Engineering - Systems
University of Southern California, Los Angeles, CA 90089

Abstract-**In this paper, we consider the problem of calculating the signal and transition probabilities of the internal nodes of the combinational logic part of a finite state machine (FSM). Given the state transition graph (STG) of the FSM, we first calculate the state probabilities by iteratively solving the Chapman-Kolmogorov equations. Using these probabilities, we then calculate the exact signal and transition probabilities by an implicit state enumeration procedure. For large sequential machines where the STG cannot be explicitly built, we unroll the next state logic k times and estimate the signal probability of the state bits using an OBDD-based approach. The basic computation step consists of solving a system of nonlinear equations. We then use these estimates to approximately calculate signal and transition probabilities of the internal nodes. Our experimental results indicate that the average errors of transition probabilities and power estimation(compared to the exact method) are only 5% and 0.6% respectively when $k = 3$. This is an order of magnitude improvement in computation accuracy compared to the existing approaches.**

1 INTRODUCTION

One of the primary objectives in the design of portable systems - which are becoming widespread - is power reduction needed to minimize the size and weight allocated to batteries. Another driver of the progress in the area of low power design is the increasing need to reduce active and/or standby power consumption in all electronic systems. Essential elements of a low power design environment include means of analyzing the dissipation of a design and mechanisms for minimizing the power consumption when needed. This paper is concerned with the power estimation in finite state machines. Various approaches for power minimization at the sequential logic synthesis level [9] [6] can benefit from the techniques presented here.

In CMOS circuits, power is consumed during charging and discharging of the load capacitances. In order to estimate the power consumption, we have to calculate the signal and transition probabilities of the internal nodes of the circuit. These probabilities depend on the input patterns, the delay model and the circuit structure.

Several signal and transition probabilities estimation algorithms have been developed for combinational circuit. Burch et al. [2] introduced the concept of a probability waveform. Given such waveforms at the primary inputs and with some convenient partitioning of the circuit, they examined every sub-circuit and derive corresponding waveforms at the internal circuit nodes. Najm [7] described an efficient technique called probabilistic simulation to propagate the transition densities at the circuit primary inputs into the circuit to give transition densities at internal and output nodes. Both methods assume inputs to sub-circuits are independent and thus did not account for the reconvergent fanout and input correlations. Stamoulis et al. [10] improved the probabilistic simulation approach by calculating the statistics of the waveforms and delays more accurately and by considering signal correlation. Ghosh et al. [4] proposed symbolic simulation in order to produce a set of Boolean functions which represent conditions for switching at each gate in the circuit. Tsui et al. [11] described a tagged probabilistic simulation approach which employs a real delay model to account for glitchings and also handles reconvergent fanout. This approach requires much less memory and runs much faster than symbolic simulation, yet achieves a very high accuracy.

The above methods assume the primary inputs to the circuit are both spatially and temporally independent, i.e. the signal value x_i of a primary input i is independent of any other primary input, and x_i at time instance t is independent of x_i at time instance $t+1$. While this assumption holds for most combinational circuits, it does not hold for finite state machines where the present state bit inputs are spatially correlated by the state encoding and temporally correlated by the state transition behavior. Figure 1 shows the STG and the gate implementation of a 4-state finite state machine. If the state bits are assumed spatially independent, the signal probability of n ($P(n)$) is equal to 0.5*0.5*0.5 = 0.125. However, n will evaluate to 1 only if $s_1s_2 = 10$ and input i is 0, hence $P(n)$ = state probability of state-3 *0.5 = 0.0721. Furthermore, if the state input bits are assumed temporally independent, the transition probability of n ($P_{1->0}(n)$) is equal to 0.125*(1-0.125) = 0.1093. However, when the present state is state-3 and the input is 0, the next state is state-4 which always forces n to 0. Therefore, the actual transition probability of n ($P_{1->0}(n)$) is equal to 0.0721.

Let $f_t(n)$ denotes the Boolean value of n at time t. The output of n will switch exactly if

$$g(n) = f_0(n) \oplus f_t(n) \qquad (1)$$

*This research was supported in part by the NSF's Research Initiation Award under contract No. MIP92/11668 and by APRA under contract No. J-FBI-91-194.

Figure 2: Capture temporal correlation of s
by next state logic.

bilities by simple substitution. This procedure is
until each state probability reaches the desired
sion. Using these probabilities, we then calculate
signal and transition probabilities by an implicit s
meration procedure. We also demonstrate that
state bit signal probabilities can be obtained by
and cascading the next state logic of the FSM
note this as a *k-unrolled* network. Based on the *k*
network, we present an approximate transition pr
calculation procedure for large sequential machin
captures the spatial and temporal correlations a
state bits.

The rest of this paper is organized as follows. I
2 we describe an iterative procedure for calcula
exact state probabilities. Exact and approximate
for calculating the signal and transition probab
internal nodes of the combinational parts of the
described in Sections 3 and 4, respectively. Exp
results and conclusions are presented in Sections

2 THE EXACT METHOD

In this section, we show how to calculate the sta
bilities given the STG and then present an exac
for calculating the signal and transition probabil
ternal nodes of a FSM given its logic implementat
the state probabilities.

2.1 State Probability Calculation

A state transition graph is denoted by $G(V, E)$ w
tex $S_i \in V$ represents a state of the FSM and
$e_{i,j} \in E$ represents a transition from S_i to S_j.
denote the state probability, that is, the probabil

$$\sum_j P_{S_j} = 1$$

where M is the number of states.

The state probabilities can then be obtained by solving the Chapman-Kolmogorov equations [8] as follows:

$$\begin{array}{rl} P_{S_i} = & \sum_{j\in IN\_STATE(i)} p_{ji}P_{S_j} \quad i=1,2,\ldots,M-1 \\ 1 = & \sum_j P_{S_j} \end{array} \tag{3}$$

where $IN\_STATE(i)$ is the set of fanin states of i in the STG.

Alternatively, we could solve for S_i's as follows. Let $P_{S_i}(n)$ be the probability of S_i after n cycles. For a discrete-state discrete-transition Markov process, $P_{S_i}(n)$ and $P_{S_i}(n+1)$ are related by:

$$\begin{array}{rcl} P_{S_i}(n+1) & = & \displaystyle\sum_{j\in IN\_STATE(i)} p_{ji}P_{S_j}(n) \\ & & i=1,2,\ldots,M-1 \\ 1 & = & \displaystyle\sum_j P_{S_j}(n+1). \end{array} \tag{4}$$

Given a set of initial condition $P_{S_1}(0), P_{S_2}(0), \ldots, P_{S_M}(0)$, these equations can be solved iteratively for $n = 0, 1, 2, \ldots$ to determine the state probabilities as a function of n.

This process is continued until the state probabilities converge, that is, the difference between $P_{S_i}(n+1)$ and $P_{S_i}(n)$ for all states is within a user defined tolerance value.

2.2 Signal Probability Calculation

Let $S = S_1, S_2, \ldots, S_M$ be the set of states of the FSM, $s = s_1, s_2, \ldots, s_N$ ($N \geq \lceil log_2 M \rceil$) be the set of state bits, $PI = i_1, i_2, \ldots, i_K$ be the set of primary inputs and n be an internal node in the combinational circuit of the FSM.

Suppose f is a disjoint cover of the function computed by n, i.e.,

$$f = \sum_{m\in Disjoint\_Cover(n)} C_m \tag{5}$$

where C_m is a cube of the disjoint cover. C_m is a function of s and PI. We partition the inputs to C_m into two groups: the symbolic state support SS_m which includes all states S_i that have set the appropriate state bits, and the primary input support I_m which includes the PI inputs of C_m. Hence $\mathrm{C}_m = SS_mI_m$.

Given a disjoint cover of node n, the signal probability of n is given by:

$$P(n) = \sum_{m\in Disjoint\_Cover(n)} P(C_m). \tag{6}$$

Since the primary inputs are independent of the state that the machine is currently in and states of the FSM are distinct, we can write

$$\begin{array}{rcl} P(C_m) & = & P(I_m)P(SS_m) \\ & = & P(I_m)\displaystyle\sum_{S_i\in SS_m} P_{S_i}. \end{array} \tag{7}$$

From equations (6) and (7), we have:

$$P(n) = \sum_{m\in Disjoint\_Cover(n)} P(I_m) \sum_{S_i\in SS_m} P_{S_i}. \tag{8}$$

Let $SS(n)$ denote the union of the symbolic state supports of all the cubes in the disjoint cover of n, equation (8) can be written as:

$$P(n) = \sum_{S_i\in SS(n)} P(S_i)P(AUX\_I(n|S_i)) \tag{9}$$

where $AUX\_I(n|S_i)$ is a Boolean function (of the primary inputs) which forces n to 1 given that the present state of the machine is S_i. Equation (9) requires explicit enumeration of the states in $SS(n)$ and is very costly. In [12], a method which employs an implicit enumeration of states using OBDDs is described.

2.3 Transition Probability Calculation

The transition probability calculation can be reduced to a signal probability calculation based on equation (1) as shown in Figure 2. g is a function of PI_0,PI_t and PS_0. As these input vectors are assumed to be uncorrelated, equation (8) can be used to obtain the exact signal probability of g, thus, the exact transition probability of n.

3 THE APPROXIMATE METHOD

As the number of states is exponential in the number of flip flops, for sequential machines having large number of flip flops, we cannot explicitly build the STG, and thus the exact method cannot be applied. To calculate the signal and transition probabilities of the internal nodes, we have to use the signal and transition probabilities of the state bits. The state bits are correlated and hence their signal probabilities are not 0.5 (which is the probability of a random input). In the following, we describe an approximate method for calculating the signal and transition probabilities using finite network unrolling followed by iterative solution of a system of non-linear equations.

3.1 Signal Probability Calculation

Given the state probabilities and the state encoding, the signal probability of state bit s_i is given by:

$$P(s_i) = \sum_{S_m\in S\_EN(i)} P_{S_m} \tag{10}$$

where $S\_EN(i)$ is the set of states whose encodings have the i^{th} bit equal to 1.

The state bit signal probabilities can also be derived without explicitly calculating the state probabilities which is very costly for sequential machines with a large number of flip flops as described next.

The transition behavior of the STG is implicitly captured by the next state logic of the FSM. Assuming the present state bits are uncorrelated and their signal probabilities are given, the characteristics (i.e. signal probabilities and correlations) of the next state bits can be obtained from the Ordered Binary Decision Diagrams (OBDDs) [1] representation for the next state logic. If we unroll and cascade the next state logic k times to form a *k-unrolled* lingof the next state logic k times.

in Figure 3, we can then calculate the s of the state bits at the k^{th} cycle given abilities at the 0^{th} cycle.

ven that the state bits (s_i) at time in-related and have signal probability of 0.5, l probabilities obtained from the OBDD etwork, are identical to those calculated 0) where the state probabilities are ob-equation (4) k times, with initial con-

a specific state encoding, a set of state to a unique set of state bit signal prob-lations. Since the initial conditions for equivalent (uncorrelated state bits with s equal to 0.5 imply that each state has y of $\frac{1}{M}$), the state bit characteristics OBDDs of the k-unrolled network maps stateprobabilities obtained b y solving mes, and hence the signal probabilities methods are the same. ■

bit signal probabilities can be obtained xt state logic ∞ times (Figure 3a). This tical. W e thus approximate the signal rolling the next state logic k times where parameter. We then feed back the sig-ues of the state bits at the output to the put of the network and iterate until the s converge (Figure 3b). We denote this *nal probability feedback* method. Indeed, next state logic as many times as nec-onvergence, but after every k stages we the signal probability values and disre-ns of the state bits.

ationis kno wn as the Picard-Peano it-

$$\mathcal{N}_n^k \quad = \quad \mathcal{F}_n(PI, \mathcal{P}_1^0, \mathcal{P}_2^0, ..$$

where $\mathcal{N}_i^k$ and $\mathcal{P}_j^0$ denote the i^{th} next output and the j^{th} present state bit at th unrolled network, respectively and $\mathcal{F}_l$'s a functions. Alternatively,

$$\begin{aligned} ns_1^k &= f_1(pi, ps_1^0, ps_2^0, \ldots, \\ &\ldots \\ ns_n^k &= f_n(pi, ps_1^0, ps_2^0, \ldots, \end{aligned}$$

where ns_i^k and ps_j^0 denote the state bit pr i^{th} next state bit at the output and the bit at the input of the k-unrolled netw and f_l's are nonlinear algebraic function to find the steady state bit probabilities,

$$\begin{aligned} ps_1 &= f_1(pi, ps_1, ps_2, \ldots, \\ &\ldots \\ ps_n &= f_n(pi, ps_1, ps_2, \ldots, \end{aligned}$$

which is a system of non-linear equations

Theorem 3.3 *f_i is contractive on the d*

Proof As $f_i = ps_j P(\mathcal{F}_i(\mathcal{P}_j = 1)) + (1$ 0), we can write

$$\frac{\partial f_i}{\partial x_j} = P(\mathcal{F}_i(\mathcal{P}_j = 1)) - P(\mathcal{F}_i(\mathcal{P}$$

If $\mathcal{N}_i \neq \mathcal{P}_i$ for every i, then this part strictly less than one. ■

From theorems 3.2 and 3.3, we can see signal probability calculation for the k-un guaranteed to converge.

Note that the signal probabilities at t lines of the next state logic in Figure 2 result of signal probability calculation for work without any signal probability feedb to this method as *1-unrolled, 0-feedback* Figure 4a shows the method used to calcu bility of the internal nodes of the FSM usi network with signal probability feedback

3.2 Transition Probability C

We enhance the transition probability c dure of Figure 2 by using the concept o state logic network. Instead of connectin

Figure 4: Calculation of signal and transition probabilities using k-unrolled network.

are then exclusive-ored together (Figure 4b) [1]. By doing so, accurate state bit signal probabilities are used and the spatialand temporal correlations bet ween PS_0 and PS_t are captured.

4 EXPERIMENTAL RESULTS

To demonstrate the effectiveness of the signal and transition probability calculation procedures, we carried out several experiments using subsets of the MCNC-91 and ISCAS-89 sequential benchmark sets. The circuits were generated using the SIS mapper and an industrial gate library. A zero delay model was assumed. All experiments were carriedout on a Sparc 2 w orkstation with 64MB memory. Power consumption measurement was based on the following model:

$$P_{avg} = 0.5 \times C_{load} \times \frac{V_{dd}^2}{T_{cycle}} \times E(transitions). \qquad (14)$$

Table 1 shows the accuracy of state bit signal probability calculationb y unrolling the next state logic. Results are comparedto the exact signal probabilit y generated from equation (10) where the state probabilities were obtained by solving the Chapman-Kolmogorov equations. The values generated from the *1-unrolled* network without *signal probability feedback* (i.e. *1u_0f*) are also presented. For each FSM, the error is calculated by summing the abso- ute value of % error on all state bit signal probabilities dividedb y the number of state bits. Note that the % error values are obtained by comparing the larger of the estimated zero and one signal probabilities with the exact probability. As expected, error decreases when k increases and *signal probability feedback* is used. The

ure 2 are also presented. The error is calculated by sum ming the absolute value of % error in transition probability for each node divided by the total number of nodes in the network. It is seen that if inaccurate state bit signal probabilities are used and the present state bit correlations are not taken into account, the transition probability calculation can be very inaccurate. The network unrolling method produces more accurate results as k increases. From the experimental results, when k is equal to 3 the average error is only 5 % which confirms that a small value of k is sufficient to produce accurate results.

Table 3 contains run times for various approaches reported in Table 2. As expected, the computation time increases with k. To obtain the effect of transition probabilities on the power estimation, we also compare the power consumption values estimated by different methods. Table 4 summarizes these results. The % deviation from the exact power consumption falls to 0.6% for $k = 3$.

Finally, we applied our method to larger sequential machines from the ISCAS benchmark set. Power consumption estimates for these circuits using different estimation methods are summarized in Table 5.

5 CONCLUDING REMARKS

We have presented exact algorithms for estimating the signal and transition probabilities for FSM based on state probabilities. We introduced the notion of *k-unrolled* network to correctly estimate the signal probabilities of the state bits of a FSM given its logic implementation. An approximate algorithm for obtaining the transition probabilities based on the notion of *k-unrolled* network was also presented for large sequential machines of which STG cannot be explicitly built.

The signal and transition probabilities of a FSM depends on the state probabilities and the state encoding. It is still an open problem to find a state encoding such that the total transition probabilities of the network is minimized. Further research needs to be done to address this issue.

References

[1] R. Bryant. Graph-based algorithms for Boolean function manipulation. *IEEE Transactions on Computers*, C-35:677–691, Aug. 1986.

[2] A. R. Burch, F. Najm, P. Yang, and D. Hocevar. Pattern independent current estimation for reliability analysis of CMOS circuits. In *Proceedings of the 25th Design Automation Conference*, pages 294–299, June 1988.

[3] A. W. Drake. *Fundamentals of Applied Probability Theory.* McGraw-Hill.

[4] A. A. Ghosh, S. Devadas, K. Keutzer, and J. White. Estimation of average switching activity in combinational and sequential circuits. In *Proceedings of the 29th Design Automation Conference*, pages 253–259, June 1992.

[8] A. Papoulis. *Probability, Random Variables and Stochastic Processes.* McGraw-Hill, 1984.

[9] K. Roy and S. Prasad. SYCLOP: Synthesis of CMOS logic for low power application. In *Proceedings of the International Conference on Computer Design*, pages 464–467, October 1992.

[10] G. I. Stamoulis and I. N. Hajj. Improved techniques for probabilistic simulation including signal correction effects. In *Proceedings of the 30th Design Automation Conference*, pages 379–383, June 1993.

[11] C. Y. Tsui, M. Pedram, and A. Despain. Efficient estimation of dynamic power dissipation with an application. In *Proceedings of the IEEE International Conference on Computer Aided Design*, pages 224–228, Nov 1993.

[12] C. Y. Tsui, M. Pedram, and A. Despain. Exact and approximate methods for calculating signal and transition probabilities in FSMs. Technical Report CNEG 93-42, Electrical Engineering-System Department, University of Southern California, October 1993.

circuit	Method	*k-unrolled* with *k-feedback*		
	1u_0f	k=1	k=2	k=3
bbsse	20.81	4.93	1.11	0.24
beecount	6.01	2.23	0.93	0.28
cse	3.03	0.84	0.20	0.04
dk14	2.19	0.22	0.16	0.03
dk15	3.97	2.38	0.15	0.02
dk16	4.20	2.07	0.80	0.73
dk17	5.31	3.00	0.25	0.26
donfile	10.28	8.44	2.83	1.42
ex2	6.92	3.58	2.80	1.00
ex5	8.81	1.31	0.55	0.27
ex6	12.12	8.55	1.60	0.56
keyb	12.89	1.46	0.36	0.05
opus	20.13	15.35	5.05	3.20
planet	17.35	14.80	18.13	14.89
s1488	10.54	8.15	2.69	1.01
s386	24.20	1.81	0.44	0.04
s510	11.37	6.87	5.32	4.81
s820	13.78	2.07	0.89	0.47
scf	19.00	13.70	3.12	1.99
styr	18.70	4.16	1.32	0.72
tbk	5.17	3.87	1.45	0.93
average error (%)	*11.28*	*5.23*	*2.39*	*1.57*

Table 1: State bit signal probabilities.

circuit	Exact trans. prob.	Method	*k-unrolled* with *k-feedback*		
		1u_0f	k=1	k=2	k=3
bbsse	20.67	71.23	15.18	5.27	1.13
beecount	12.04	29.25	7.39	2.42	0.64
cse	16.09	98.55	8.10	2.31	0.53
dk14	23.20	19.88	2.88	0.30	0.05
dk15	19.61	11.47	5.99	1.04	0.04
dk16	38.43	23.18	9.52	3.79	1.45
dk17	16.42	19.66	8.28	2.24	1.16
donfile	23.56	18.63	8.83	5.10	2.90
ex2	29.65	41.74	34.88	11.43	4.96
ex5	13.61	51.20	9.17	3.97	1.25
ex6	21.66	25.64	11.33	2.99	1.59
keyb	26.53	50.56	6.11	1.55	0.43
opus	16.59	42.79	40.23	24.11	5.11
planet	69.98	42.93	40.76	28.10	26.80
s1488	69.58	241.14	119.05	34.79	14.60
s386	25.57	377.13	12.20	5.80	1.33
s510	42.82	39.31	14.56	1.48	9.04
s820	43.55	201.45	14.22	7.80	3.88
scf	59.22	102.57	79.31	38.51	18.64
styr	35.47	166.25	43.16	14.83	7.62
tbk	79.46	39.86	17.98	3.74	2.58
average error (%)		*81.64*	*24.24*	*9.60*	*5.03*

Table 2: Transition probabilities.

circuit	CPU time in seconds				
	Exact Method	Method *1u_0f*	*k-unrolled* with *k-feedback*		
			k=1	k=2	k=3
bbsse	7.39	3.76	3.28	5.16	8.86
beecount	2.40	1.43	1.05	1.45	2.08
cse	11.86	6.29	5.31	10.71	19.80
dk14	6.30	3.62	2.71	4.98	8.13
dk15	3.86	2.31	1.63	2.36	3.20
dk16	18.20	9.73	8.43	17.68	30.65
dk17	3.60	2.00	1.55	2.24	3.28
donfile	26.30	6.83	5.40	11.20	21.13
ex2	12.41	4.75	4.22	8.00	15.40
ex5	3.28	1.82	1.45	2.11	3.18
ex6	5.46	3.12	2.51	4.30	6.82
keyb	34.48	11.34	8.98	20.24	36.04
opus	6.48	2.48	2.26	3.90	5.96
planet	144.54	18.00	15.51	22.68	31.80
s1488	89.89	32.54	35.23	66.44	109.48
s386	26.30	5.80	5.16	10.00	16.23
s510	127.94	9.08	7.95	12.55	18.33
s820	33.51	22.15	16.06	42.50	84.62
scf	294.50	31.30	26.48	40.09	56.40
styr	38.78	23.03	19.19	43.23	84.60
tbk	127.14	74.56	60.34	209.60	905.00

Table 3: CPU times for transition probability calculation.

circuit	Power consumption				
	Exact Method	Method *1u_0f*	*k-unrolled* with *k-feedback*		
			k=1	k=2	k=3
bbsse	5072	6326	5035	5037	5061
beecount	2204	2422	2179	2172	2197
cse	4144	5842	4141	4144	4147
dk14	4913	5186	4939	4916	4913
dk15	3755	3793	3751	3748	3755
dk16	11947	12487	11928	11962	11972
dk17	3934	4151	4008	3945	3936
donfile	7787	7872	7773	7806	7812
ex2	8695	7963	7816	8377	8539
ex5	3051	3500	3037	3012	3001
ex6	4518	4866	4655	4567	4529
keyb	7712	9176	7638	7704	7728
opus	4080	4277	4238	4049	4095
planet	19497	19905	19778	19483	19489
s1488	22250	52481	31707	26735	23679
s386	7058	14749	6763	7037	7076
s510	8709	10816	9261)	8785	8745
s820	18078	31557	18001	17878	18116
scf	14309	18323	15434	14351	14342
styr	13020	16106	13367	12953	13045
tbk	74528	87964	73328	73337	74505
average error (%)		*26.83*	*4.30*	*1.51*	*0.63*

Table 4: Power estimates.

circuit	No. of latches	Power consumption			
		Method	k-unrolled network method		
		1u_0f	k=1	k=2	k=3
s344	15	7256	5791	5750	5707
s382	21	10099	3980	3992	3976
s400	21	10640	4063	4075	4060
s444	21	10174	3890	3897	3881
s526	21	14202	5778	5798	5762
s641	19	9500	8149	8111	8098
s713	19	9848	8566	8539	8522
s838	32	5934	5934	5934	5934
s953	29	16440	12381	11856	12367
s1196	18	28935	27889	27890	27890
s1238	18	31946	30911	30912	30912

Table 5: Power estimates for machines with large number of flip flops (the estimation accuracy increases as k increases).