Power Estimation of Sequential circuits using Hierarchical Colored Hardware Petri Net Modeling

Ashok K. Murugavel and N. Ranganathan
Dept. of Computer Science and Engineering
University of South Florida
Tampa, Florida 33620

ABSTRACT

A Hierarchical Colored Hardware Petri net (HCHPN) based model was proposed in [8] for estimating switching activity in combinational circuits. In this paper, we model sequential circuits as HCHPNs incorporating real delays for both gates and interconnects. Thus, the given sequential circuit is first modeled as a HCHPN and simulated for switching activity estimation in the petri net domain which leads to better accuracy and faster simulation. Experimental results for ISCAS'89 benchmark circuits show that the proposed HCHPN model yields accuracy on an average within 4.4% of that of PowerMill. The per-pattern simulation time for HCHPNs is about 2.4 times lesser than that of PowerMill.

Categories and Subject Descriptors

B.7 [**Hardware**]: Integrated Circuits; B.7.2 [**Integrated Circuits**]: Design Aids—*simulation*

1. INTRODUCTION

In this work we propose a new simulative approach to model sequential circuits for accurate power estimation at gate level using real delays for both gates and interconnnects. By transforming the circuits into their corresponding Petri net models, it is possible to capture spatial and temporal correlations. Simulating the circuits in the Petri net domain is much faster due the availability of efficient algorithms for Petri net simulation. Numerous techniques exist in the literature to estimate the average power of a circuit at the gate level under a zero delay model [9]. The zero delay models ignore the glitches, and can lead to under-estimation of power. Recently, techniques to estimate average power under a real delay model for gate delays,have been proposed [2, 3]. In deep sub-micron technologies the interconnects introduce critical problems on the signal integrity and interconnect delays dominate over the gate delays and so cannot be ignored. It has been observed that the existing power estimation methods are fast and accurate, but, do not consider real delay models that include interconnect effects [1].

In [10], a RT-Level methodology is proposed to estimate average power based on the total capacitance estimates. The total capacitance estimate is the product of the area complexity and the average capacitance. A fast and accurate glitch modeling technique based on logic simulation is given in [11]. The average power is estimated based on a toggle count mechanism. The toggle information obtained is not exact for circuits with high depth. The methodology has been shown to be faster than Hspice but with the loss of accuracy. The first RT-Level power estimation technique to include interconnect effects was presented in [1]. The model for switching activity estimation is based on the relationship between switching activity and entropy. The RT-level technique has an estimation error of about 25.8%. Sequential circuits are highly pattern dependent and the estimated average power can vary widely for different vector sets. In [6], a statistical power estimation technique considering blocks of consecutive vectors are used to obtain a realistic vector set to estimate the power of sequential circuits.

The technique proposed in this paper considers a complete delay model (considering both gate and interconnect delays) for switching activity estimation. In order to model switching activity, we describe a new type of Petri net called the Hierarchical Colored Hardware Petri net (HCHPN) [8] which incorporates accurately the spatial and temporal correlations with real delays. The logic circuit is first modeled as a Gate Signal Graph (GSG), which is then transformed into its corresponding Hierarchical Colored Hardware Petri net (HCHPN) and simulated as a Petri net to get power values. While providing the similar accuracy levels compared to existing simulators, the proposed strategy requires significantly less per-pattern simulation time.

2. HCHPN BASICS

A Petri net model consists of a set of *places* (P) and a set of *transitions* (T), in which the places correspond to the states of the model and the transitions represent the actions related to the states of the model [4]. The places and transitions are connected by a set of directed *arcs* (A). An arc can only connect a transition with a place or vice-versa. An arc directed from a place to a transition is called an input-arc and an arc from a transition to a place is called an output-arc. A transition is said to be *enabled* if there are enough tokens in each of the input places as specified by the arcs connecting the input places to the transition. An enabled transition can *fire* if the other conditions associated with the transition are satisfied. The initial state of the Petri net is called its *initial-marking*. The current state of a Petri net is the distribution of tokens to the places in the Petri net. For a theoretical treatment on CPNs, the reader is referred to [8].

Definition [4]: A Time set (TS) is defined as the set of all non-negative real numbers, $TS = x \in R$, where x represents the time instant and R is the set of all real numbers.

The concepts of Hardware Petri net (HPN) [12] and Hierarchical Colored Petri net (HCPN) [8, 4] can be modified to define a new

ISLPED'02, August 12-14, 2002, Monterey, California, USA.

kind of Petri net which we call as the Hierarchical Colored Hardware Petri net (HCHPN). In order to model real delays we need to add the factor of time into the Petri net definition. The features of HCHPNs are: (i) certain restricted places can store only one token at any time and a newly arriving token always overwrites the existing token, (ii) when a transition fires, the associated gate function is evaluated, (iii) besides the weights on the arcs, functions can be defined on the arcs as additional firing conditions; delays for the gates and the interconnects can be specified on the transitions as firing conditions, (iv) each token has a time stamp and is called a timed token, which is available only after the clock time of the simulator is greater than or equal to the time stamp on the token.

The HCHPN has a set of *substitution transitions*. The substitution transition allows the embedding of a small or sub-Petri net within a larger Petri net for higher level modeling. When the substitution transition needs to be fired, the underlying Petri net needs to be evaluated in order to determine the outcome of the substitution transition. We relate the individual CPN to a substitution transition, the individual CPN in a HCHPN is called a *page*. The input and output places of a page are called *port nodes*. The substitution transitions are called *super nodes* and the pages that contain the super nodes are called *super pages*. In a super page, the places that are connected to the super nodes (substitution transitions) are called *socket nodes*. The socket nodes of a super node (substitution transition) are related to the port nodes of the page through the port assignment function. The HCHPN has in addition to the substitution transitions, a set of transitions called *gate transitions*. The gate transitions are present only in the sub-nets of the HCHPN. An important function added to the HCHPNs is time. With the introduction of time to each token we can model highly complex real time systems. A token in a HCHPN is called a colored token. The colored token is a 3-tuple (p,d,i) where p is the current place of the token $p \in P$, d is the color set of the token and i is the time of the firing of the token. The HCHPN contains a set of places called *restricted places*, than can hold only a finite number of tokens. To impart additional conditions during the firing of a transition, functions can be added to the transitions, which are called *code segment functions*. The HCHPN can be defined mathematically as follows.
Definition: A HCHPN is defined as a 10-tuple,

$$HCHPN = (PG, ST, GT, RP, PO, PA, PR, TI, TO, TC) \quad (1)$$

where, (i) PG is a finite set of pages such that (a) each page $pg \in PG$ is a non-hierarchical CPN and, (b) none of the pages have any net element in common,
(ii) $ST \subseteq T$ where ST is a set of substitution transitions (super nodes),
(iii) ($GT \subset T$ and $GT \cap ST = \emptyset$), where GT is the set of gate transitions,
(iv) $RP \subset P$ is a set of restricted places, which can hold only a finite number of tokens,
(v) $PO \subset P$ is the set of port nodes, the port nodes are the input and output places of a page,
(vi) PA is a port assignment function, which defines the relation between the socket nodes of the substitution transition and the port nodes of the page,
(vii) $PR \in PG_{MS}$ is a multi-set of prime pages; it determines the number of instances each individual page has,
(viii) TI is the time set defined in **Definition 2**,
(ix) TO is the set of all possible colored tokens, i.e., all 3-tuples (p,d,i), where $p \in P$, d is the value of the token and $i \in TI$,
(x) TC is the code segment function. It is defined from T into the set of boolean functions,

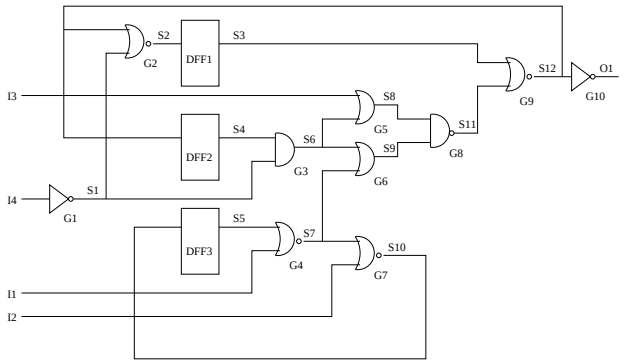

Figure 1: s27: ISCAS '89 sequential circuit

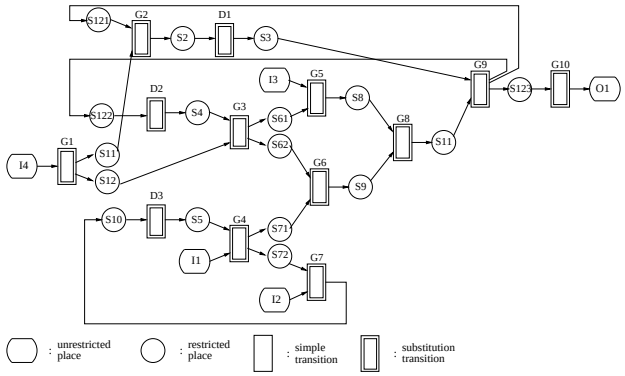

Figure 2: HCHPN for the s27 ISCAS '89 sequential circuit

3. MODELING USING PETRI NETS

We first, represent the circuit as a Flow Graph where each gate/flipflop in the circuit is represented as a node and each signal in the circuit as an arc. Figure 1 shows an ISCAS '89 sequential circuit **s27**. The Flow Graph thus obtained is transformed into a Gate Signal Graph (GSG), by introducing additional nodes corresponding to the signals, while maintaining connectivity by introducing arcs. To represent multiple fan-outs, we introduce a node for each output fanning out from the gate. The GSG thus obtained can be represented as a Petri net by replacing the gate nodes in the GSG by transitions and the signal nodes by places. This Petri net is a simple mapping of the circuit into a PN.

The Petri net is transformed into a HCHPN by adding type, color, hierarchy and delay information. A place in a HCHPN has a type function specifying the type of tokens the place is associated with. While in a Petri net, all places are of the same type which can store any number of tokens, in a HCHPN there are two kinds of places: (i) unrestricted places and (ii) restricted places. The places in the Petri net representing signals are replaced by restricted places in the HCHPN to model the interconnects. The input and output places of the Petri net are not replaced by restricted places. Fanning out outputs are replaced by an output place and a transition, to get the required number of tokens. Each transition in the Petri net is replaced with a substitution transition to perform the action corresponding to the gate that is modeled by the transition. The interconnect delay values are added to the arcs of the HCHPN.

In Figure 2, the substitution transitions *G1,..., G10*, represents the logic gates in the s27 seqential circuit. The substitution transitions *D1,..., D3*, represents the 3 D-flip flops, in the circuit. To accurately model a gate, we need to (i) capture any glitches in the

inputs, (ii) model the working of the gate, (iii) capture the switching activity of the gate and (iv) handle temporal and spatial correlations. The modeling of a 2-input NAND gate is shown in Figure 3. Similarly other types of gates can be modeled.

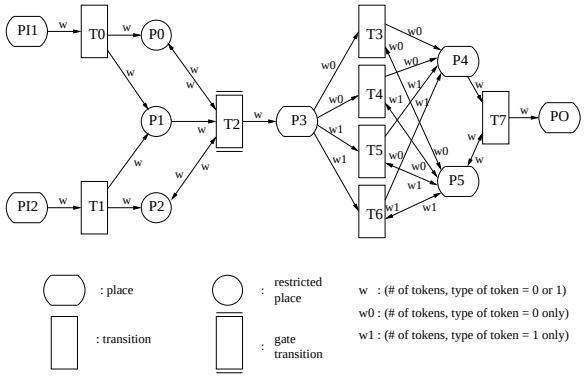

Figure 3: HCHPN structure of the NAND gate

The gates in a circuit can switch multiple times before settling to the correct output value, which leads to more switching activity. The main contributor of this unnecessary switching is the signal delays due to the different circuit paths. In order to consider the glitches in the circuit, the delay of each signal has to be taken into account while modeling switching activity. To obtain a delay model for deep sub-micron (DSM) circuits, both the gates and the interconnect delays have to be taken into consideration, as the interconnect delays dominate over the gate delays in DSM circuits.

In deep sub-micron circuits, the interconnect delay is important. The total delay is calculated as the sum of (i) intrinsic gate delay (T_{GD}) and (ii) gate load delay (T_{RD}), as $t_d = T_{GD} + T_{RD}$ [5]. The *intrinsic gate delay* is the delay due to the physical devices like transistors that constitute the gate. The intrinsic gate delay is calculated using Hspice. The *gate load delay* is the delay due to the RC network connected to the output of a gate. In this work, we use a Thevenin equivalent circuit to model the interconnect and thus, the signal delay. The capacitance and the resistance of the Thevenin equivalent are known from the layout. In [5], a one segment Π model has been proposed to model the gate output while still considering the resistance shielding effects. The model approximates the load by considering only the first three moments of the driving point admittance that approximates the entire interconnect load.

4. MODELING SEQUENTIAL CIRCUITS

To calculate the switching activity of the gate, we need to calculate the various instances at which the gate switches from either 0 to 1 or from 1 to 0. The transitions {T3, T4, T5, T6} in Figure 3 fire only when an event occurs, i.e., the gate switches. The total switching activity of the circuit is not the simple sum of the switching activity of the gates in the circuit, the temporal correlations in the circuit also need to be taken into account. In the modeling of the circuit/gate as a HCHPN, the temporal correlations are also taken into consideration, and so the switching activity of the circuit is given as the sum of switching activity of the gates of the circuit which is indeed the firing of the transitions in the HCHPN. Thus, we have a direct relationship between the transition firing in a HCHPN and the switching activity of the circuit. Sequential circuits are essentially combinational circuits with a delayed feedback. The sequential circuit also contains flipflops, which have a feedback internally.

Figure 4 gives the block diagram for the power estimation flow using HCHPNs for sequential circuit. The input to the tool is the gate-level net-list, and the RC estimates (layout-level) at each node, the *intrinsic gate delay* estimates for each of the gates and the input vectors to be simulated. For simulating the HCHPN, we use a software tool called the Design/CPN [7] (called from now on as the simulator).

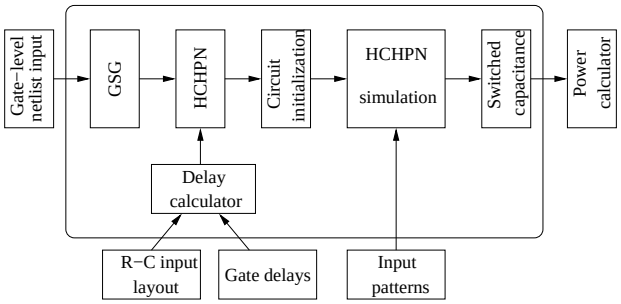

Figure 4: Block diagram of power estimation tool using HCHPNs

A major concern with sequential circuits is the identification of the initial state of the circuit. The identification of the initial state in a combinational circuit is not important and can be ignored. A simple approach is to assign a known initial state to the circuit, but for large circuits knowing this initial state is difficult. Hence, to over come this problem, there are different methods by which the initial state of the machine can be assigned, (i) run the circuits through a sequence of vectors, this initialization run will take the circuits from the unknown state to a known correct state, (ii) generate a random circuit state. Since knowing all the observable states of a circuit is difficult and could be exponential, the random generation of the circuit state can actually put the circuit in a state that may not be reachable. In this work, we use the first technique. The circuit is run through an initialization sequence or warmup sequence. The power consumed during this initialization period is ignored. The length of the initialization sequence is much to debate. The procedure outlined in 5, helps in the better understanding of the HCHPN simulator. At the start of the initializing run all the outputs of the gates in the circuit are initialized to zero, we call this state as the ”0” state. The length of the initializing sequence is a much debated problem. In this work we have assumed that the circuit will return to a known state within 10 000 vectors, which is 10% of the actual simulation vector sequence. This is a valid assumption since the vector sequence is long enough to bring the circuit into a known state. After the circuit is in the known state, the simulation vectors for the estimation of power are given to the simulator and the power is estimated at each node based on the switched capacitance estimates given by the simulator. The power consumed during this initialization sequence is ignored, since this is power does not reflect on the circuits activity and is only an initialization sequence.

5. EXPERIMENTAL RESULTS

The experimental results for the ISCAS ’89 sequential benchmark circuits are given. The proposed approach has been implemented and tested for benchmark circuits. The ISCAS circuits consist of the basic gates like AND, OR, NAND, NOR, NOT and D-flipflop. The circuits were synthesized using TSMC 0.4μ standard cell CMOS technology. The results for all the circuits has not be listed instead only a selected few has been listed for want of space.

Average power estimation is highly pattern dependent, and can

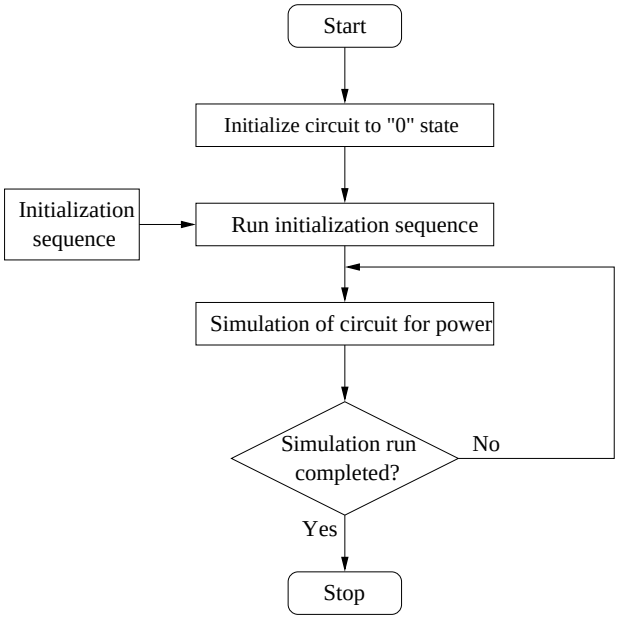

Figure 5: HCHPN sequential circuit simulator - Flow chart

Table 1: Accuracy for sequential circuits

Circuit	Power Values				% error $\frac{\|(1)-(2)\|}{(1)}$
	PowerMill (1)		HCHPN (2)		
	Avg	Max	Avg	Max	
s298	410.2	410.8	450.1	450.6	9.7
s349	291.4	311.0	311.6	329.4	6.9
s386	591.5	612.5	578.5	586.9	2.1
s444	476.4	476.7	489.3	490.1	2.7
s526	524.6	563.9	544.5	564.2	3.7
s641	889.2	1084.8	867.4	927.4	2.4
s820	674.2	888.9	661.7	856.7	1.8
s1196	2220.4	2414.6	2044.9	2240.0	7.9
s1488	864.5	981.0	887.5	994.7	2.6
Average					4.4

vary widely for different vector sets. This is not a very significant problem for combinational circuits but for sequential circuits this is a very critical problem. If we choose a random vector set, the sequences can put the circuit in states that do not normally occur in the working of the circuit [6]. Since, there are no standard input vector sets available for these benchmark circuits. We have tried to generate a set of correlated sequence consisting of 100 000 vectors based on a technique proposed in [6]. We have generated a set of 100 correlated sequences such that we can average out any extreme simulation results. The initialization sequence is a random vector sequence generated under a uniform random probability. The rise and the fall times of the ramp input given to PowerMill was 0.125*ns*. The intrinsic gate delays for the HCHPNs were calculated using Hspice. The HCHPNs for large circuits need to be segmented, such that the number of arcs between the segments is kept to a minimum. The simulations were performed on a Sun Ultra10 with 128 M RAM. Design/CPN [7], a Colored Petri net simulator is used to simulate HCHPNs.

Tables 1 and 2 compare the accuracy and speedup for modeling both intrinsic gate and interconnect delays. The % error of the HCHPN and PowerMill simulations over that of Hspice are calculated as absolute difference between the power estimate from Hspice and the power estimate from HCHPN/PowerMill by the power estimate from Hspice. On an average the power estimates

Table 2: Simulation times for sequential circuits

Circuit	Simulation times				Speedup $\frac{(1)}{(2)}$
	PowerMill (1)		HCHPN (2)		
	Avg	Max	Avg	Max	
s298	160.3	162.6	32.1	33.7	4.9
s349	105.2	105.8	48.6	49.1	2.1
s386	111.4	112.0	79.8	80.7	1.3
s444	193.8	196.8	80.7	80.9	2.4
s526	206.1	208.3	92.4	96.5	2.2
s641	120.6	136.6	36.6	38.2	3.2
s820	99.9	140.7	40.2	56.9	2.4
s1196	294.2	357.5	140.0	180.4	2.1
s1488	372.1	511.4	213.5	270.8	1.4
Average					2.4

were within 4.4% of that of PowerMill. The HCHPN simulations have an average speed-up of 2.4 times over that of PowerMill. The average and maximum simulation times, % error have been reported. Thus we have an accurate modeling of the gates/circuit using HCHPNs that is faster than the current existing power estimation tools.

6. REFERENCES

[1] Buyuksahin K. M. and Najm F. N. High-Level Power Estimation with Interconnect Effects. In *Proc. of the Intl. Symp. on Low Power Electronic Devices*, pages 197–202, 2000.

[2] Chou T.-L., Roy K. and Prasad S. Estimation of Circuit Activity for Static and Domino CMOS circuits Considering Signal Correlations and Simultaneous Switching. *IEEE Trans. on CAD*, 15(10):1257–1265, Oct. 1996.

[3] Ding C-S., Tsui C-Y., and Pedram M. Gate-level Power Estimation using Tagged Probabilistic Simulation. In *Proc. of the Intl. Conf. on CAD*, pages 1099-1107, 1998.

[4] Jensen K. *Colored Petri Nets: Basic Concepts*, volume 1. Springer, second edition, 1996.

[5] Kahng A. B. and Muddu S. Gate Load Delay Computation Using Analytical Models. In *Proc. of Asia-Pacific Conference on Circuits and Systems*, pages 433–436, 1996.

[6] Kozhaya J. N. and Najm F. N. Power Estimation for Large Sequential Circuits. *IEEE Trans. on VLSI Systems*, 9(2):400–406, Apr 2001.

[7] Meta Software Corporation. *Design/CPN Ref. Manual*, 1993.

[8] Murugavel A. K. and Ranganathan N. A Real Delay Switching Activity Simulator based on Petri net Modeling. In *Proc. of Intl Conf. on VLSI Design*, pages 181–186, 2001.

[9] Najm F. N. A Survey of Power Estimation Techniques in VLSI Circuits. *IEEE Trans. on VLSI Systems*, 2(4):644–649, Dec. 1994.

[10] Nemani M. and Najm F. N. High-level Area and Power Estimation for VLSI Circuits. *IEEE Trans. on Computer-Aided Design of Integrated Circuits*, 18(6):697–713, Jun. 1999.

[11] Rabe D., Jochens G., Kruse L., Nebel W. Power-Simulation of Cell based ASIC's: Accuracy-and Performance Trade-Offs. In *Proc. of the European Design Automation and Test Conf.*, pages 356 –36, 1998.

[12] Rokyta P., Fengler W., and Hummel T. Electronic System Design Automation Using High Level Petri Nets. In *Hardware Design and Petri Nets, ed. A. Yakovlev, L. Gomes and L. Lavagno*, pages 193– 204. Kluwer Academic Publishers, 2000.