Studies of Timing Structural Properties for Early Evaluation of Circuit Design *

Andrew B. Kahng
CSE and ECE Departments, UCSD
La Jolla, CA, USA 92093-0114
abk@ucsd.edu

Ryan Kastner, Stefanus Mantik, Majid Sarrafzadeh, Xiaojian Yang
Computer Science Department, UCLA
Los Angeles, CA 90095-1596
{kastner,stefanus,majid,xjyang}@cs.ucla.edu

ABSTRACT

The Rent parameter has been widely used to characterize interconnect complexity of designs. The Rent power-law relationship is often used for *a priori* wire estimation, which is an enabling component of timing closure methodologies. However, the Rent parameter does not explicitly address timing information. In this paper, we propose studies of circuit timing structure that can potentially help identify useful characterizations of the timing behavior of the design. We believe that such characterizations can allow us early identification of designs for which timing closure will prove difficult. One characterization of the timing structure is analogous to the Rent parameter: we propose a *temporal Rent* characterization based on analyze of the number of *active signals* intersecting a specified timing boundary within the clock period.

1. INTRODUCTION

With advancing semiconductor technology, IC designs experience rapid increase in gate count and design complexity. In addition, interconnect delays increasingly dominate total delays. These trends make timing closure harder to achieve. Furthermore, timing properties like crosstalk, wire sizing, etc. that were once ignored at the early stages of circuit design (e.g. logic synthesis), must now be considered to reach the timing closure.

Traditional timing analysis and optimization tools track actual arrival time (or data-ready time) and required arrival time – on a per-pin or per-edge basis – as the salient timing properties of circuits. Due to the recent increased importance of logical-physical synthesis integrations [3, 2] as well as convergent, predictable RTL-down implementation, we feel that additional analyses of *timing structural properties* may be necessary to help achieve reliable timing closure.

In the logic synthesis regime, where cell placements and net routings do not yet exist, delays can be estimated using fanout- and blocksize-based table lookups or *wireload models*. Such delay estimations have been characterized as inaccurate due to the lack of detailed physical embedding information [4, 5, 1]. At the same time, available *a priori* estimation methods can accurately estimate such layout parameters as total wirelength. Some of these estimators are based at least in part on *Rent's rule*, a measure of interconnect (topological) complexity that is typically used to estimate wirelength distributions. In certain physical implementation methodologies, the a priori estimation reduces design cycle time by preventing optimization runs that will likely to fail (e.g., due to global over-congestion). Motivated by the usage of Rent's rule in generating estimators and feasibility criteria for *wiring*, we seek analogous estimates and early feasibility criteria for *timing*.

In the remainder of this paper, we propose several novel analyses of circuit timing structure, and apply these analyses to three sample industrial circuits. Section 2 develops preliminaries (e.g., traditional timing parameters of the design) and the new concept of an *active interval* for a timing edge. Section 3 proposes several timing structure analyses, along with intuitions for how these analyses might be used to assess netlists for crosstalk, peak power, or other circuit properties. Section 4 applies the new analyses to a few tractable special-case topologies, and Section 5 concludes with a brief review of our ongoing research efforts.

2. PRELIMINARIES

Gate and wire delays can be modeled using a *pin graph* (Figure 2), a weighted directed acyclic graph (DAG) with each vertex representing a pin of a gate and each edge corresponding to a direct connection between two pins. Each edge is either a *gate edge* or a *net edge*. A gate edge (*internal edge*) connects an input pin and an output pin of the same gate; its weight (delay) is the intrinsic propagation delay through that particular path within the gate. A net edge (*external edge*) connects two pins from different gates; its weight is the propagation delay between the corresponding source-sink pin pair. Since exact gate and interconnect delays are not known until detailed routing has been completed, approximations such as wireload models or Elmore delay approximations are used at earlier stages of the design cycle. Note also that the hyperedges corresponding to signal nets are represented as directed stars in the pin graph, i.e., a gate output that fans out to k other gate inputs will be represented by k directed edges.

The traditional timing parameters are *actual arrival time* and *required arrival time*. The actual arrival time is the time at which the signal settles. Arrival times at primary inputs of a combinational circuit are part of the block's boundary timing conditions (in the experimental analyses described below, we set all arrival times at primary

*This work is supported by the MARCO Gigascale Silicon Research Center and by NSF grant MIP-9527389.

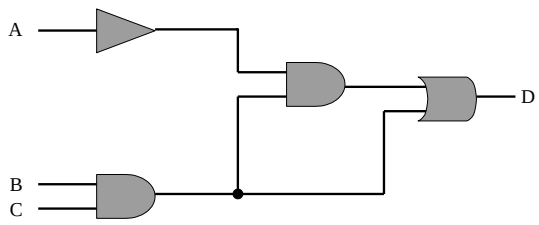


Figure 1: A small combinational circuit example.

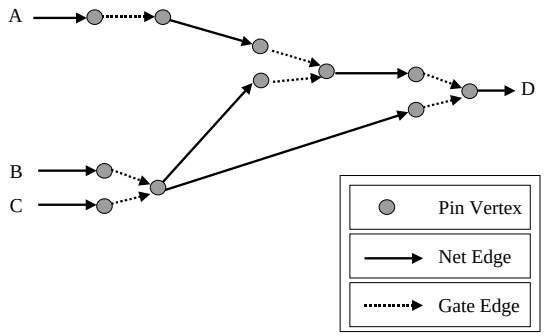


Figure 2: The pin graph corresponding to Figure 1.

inputs to zero). By traversing the circuit in a forward topological manner, we can compute the arrival times for each pin. Let aat_i denote the actual arrival time for vertex v_i, and let d_{ij} denote the propagation time for edge e_{ij}. Then:

$$aat_j = \max_{i:e_{ij} \in E} aat_i + d_{ij}$$

Required arrival times are computed similarly, considering the vertices in reverse topological order. The required arrival time rat_i at vertex v_i is:

$$rat_i = \min_{j:e_{ij} \in E} rat_j - d_{ij}$$

The actual arrival times and required arrival times at all vertices can be computed in $O(|V| + |E|)$ time. The *slack* of a pin is given by the pin's required arrival time minus its actual arrival time.

A central concept in our work is that of an edge being *active*. We say that an edge e_{ij} is *active* at time t if and only if $start_{ij} \leq t \leq end_{ij}$, where

$$start_{ij} = \min_{k:e_{ki} \in E} t_k + d_{ki}$$
$$end_{ij} = \max_{k:e_{kj} \in E} t_k + d_{kj}$$

Intuitively, $start_{ij}$ is the earliest arrival time possible at the head of the edge, v_i, given the actual arrival times at all the predecessors of v_i. By contrast, end_{ij} is the latest possible arrival time (which is to say, the "actual" arrival time according to the definition above) at the tail of the edge, v_j. We say that an edge e_{ij} is active during its *active interval* $[start_{ij}, end_{ij}]$. The key difference from traditional AAT/RAT analysis is that the earliest possible switching time is used as the start time of an edge.[1]

3. TIMING STRUCTURE ANALYSES

Now that we have defined the active interval of an edge, we wish to determine some possible uses for this definition. Specifically, we develop some useful models/graphs to ascertain timing properties of the design. This is helpful at the early stages of the circuit design as the effect of incremental changes may be estimated. For instance, the effect of a logic synthesis transformation, say gate resizing, could be determined by looking at timing analysis before and after the transformation. Instead of having the circuit designer analyze the timing using an ad hoc, visual inspection, we could characterize properties of the models and do this analysis without human interaction. This is one of the primary goals of the project.

For analysis, three industrial circuit designs are used. The properties of the designs are given in Table 1. The active intervals are calculated during three stages of the design flow: (i) the *pre-placement* stage on which delays are calculated using wireload model, (ii) the *post-placement* stage on which net delays are estimated based on the half-perimeter bounding box of the net, and (iii) the *post-routing* stage. We use Cadence Design Systems' QPlace and WarpRoute tool for placement and routing. The timing data are obtained with Pearl static timing analyzer. Of course, any place and route tool could be used. In fact, it would be interesting to compare the designs across a variety of physical design tools. This may yield intrinsic timing properties of the circuit that are not due to the physical design tool's optimization.

Name	# Cells	# Nets	Clock Period
Design 1	42352	44490	16.6 ns
Design 2	19832	22974	80 ns
Design 3	21103	21230	9.09 ns

Table 1: Design statistics for three industrial circuits.

3.1 Active Interval Distribution Graph

First, we propose the active interval distribution graph. This graph plots the size of the active interval on the x-axis and the number of edges on the y-axis where y is the number of active intervals less than equal to x. As you can see, it is a cumulative function i.e. it is monotone increasing function. The largest x value is the length of the clock cycle and by that time, the y value will be $|E|$ for a properly functioning circuit. The active interval distribution graphs for the three designs are shown in Figure 3.

We believe that a timing optimized design should have a large initial slope as in Design 1. This means that the uncertainty in a large number of edges is low. Ideally, we would want every edge to have a low switching uncertainty to help minimize crosstalk. A common first step in crosstalk constraint generation is to partition the nets into groups according to their switching windows. Nets with disjoint

[1] This will count an edge as "active" even if it is only glitching, but we believe that this is a more relevant criterion when we consider power and signal integrity analyses. In general, our proposed methods will ignore the logic function of the circuit, including sensitization.

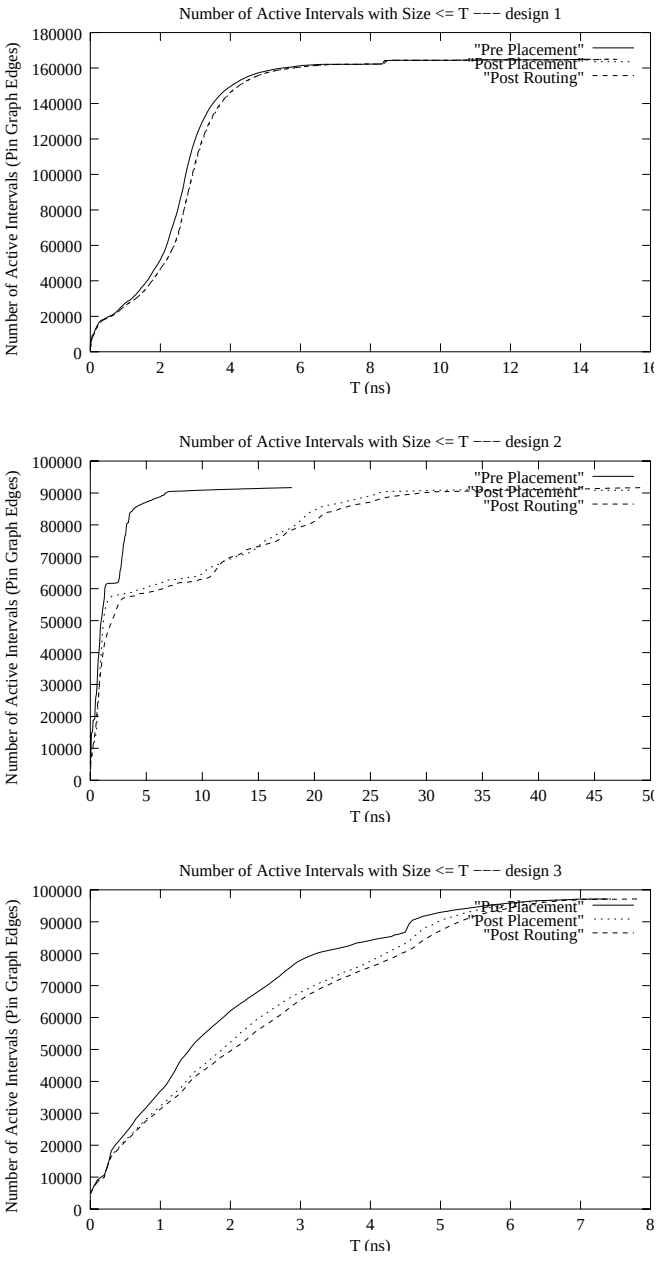


Figure 3: Distribution of the active interval sizes for three different designs (from top to bottom: design1, design2, design3). Each figure has three lines corresponding to the pre-placement, post-placement and post-routing distributions.

switching windows are partitioned into separate groups. Nets that do not switch at the same time cannot cause crosstalk on each other, hence they can be routed adjacent to another without negative effect. Also, crosstalk noise may cause circuit glitches that lead to incorrect circuit behavior even with disjoint switching windows. One must consider this case separately. Even in this case, wire uncertainty is important because the duration of the switching time factors into causing a circuit glitch. In order to be conservative, a large uncertainty would cause an overestimation of the switching time which could possibly be the difference between causing a glitch and safe behavior. Therefore, uncertainty may cause unnecessary crosstalk constraints for routing of the nets.

The real uncertainty would be the active interval minus the delay of the edge. If the edge has large delay this could be mistaken as a large uncertainty which is not the case. Essentially, we are trying to make the signals to a gate arrive at the same time. If they all arrive simultaneously (with no uncertainty), then signals will propagate through the gate (modulo difference in pin to pin gate delay), hence the output of the gate would have little to no uncertainty.

3.2 Clock Cycle Activity Graph

Our next graph plots the number of active intervals switching at a given time over the clock cycle. Here, the x-axis is the clock cycle offset which varies from zero – the start of the clock cycle – to the length of the clock cycle. The y-axis represents the number of edges that are currently active at the current time. Figure 4 gives the plots for the three considered designs.

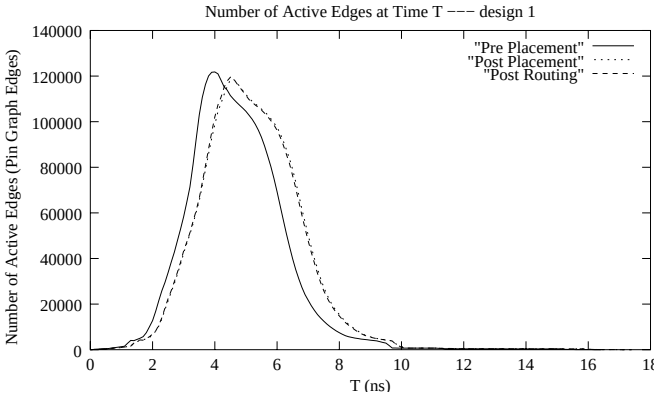
In this case, we feel that a "flatter" plot, i.e. a plot with small slope and large integral value, corresponds to a better design. A "flat" plot means that the switching is well distributed across the clock cycle. First, we want a significant number of switching edges near the end of the cycle. Consider a design with minimal switching at the end of the clock cycle. This means that there are a small number of critical paths. Therefore, it is likely possible to perform timing optimizations on these small number of paths to achieve a design with larger frequency. We believe that a well-optimized circuit would have a decent number of critical paths. Hence, it would be unlikely that further optimization would reduce the clock cycle. Additionally, a design that switches over the entire clock cycle would not have "hot spots" i.e. a time when a large amount of switching occurs. Hot spots have negative implications towards circuit reliability and power dissipation.

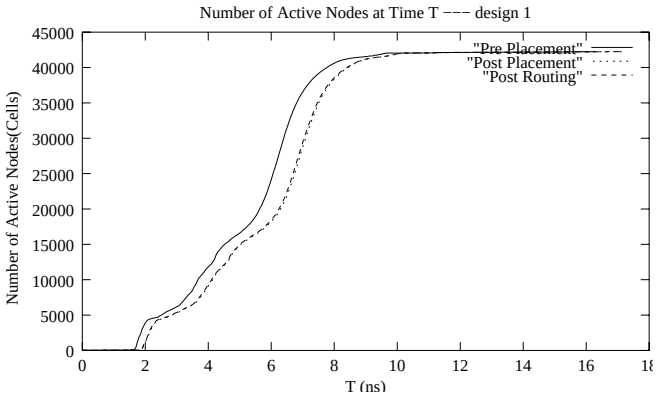
Figure 5 shows a similar analysis that can be done for node activity up to time x. A node is considered active if an edge that is connected to its output pin is active. Let the x-axis be the time from start of cycle, let the y-axis corresponds to the number of nodes that have their last arrival time less than or equal to x. In this case, the "goodness" of the plot is more easily measured. A 45 degree line would be an ideal circuit. Therefore, we could do a comparison of the actual plot to the 45 degree line to determine the quality of the design. For this reason, these graphs are preferable to those in Figure 4.

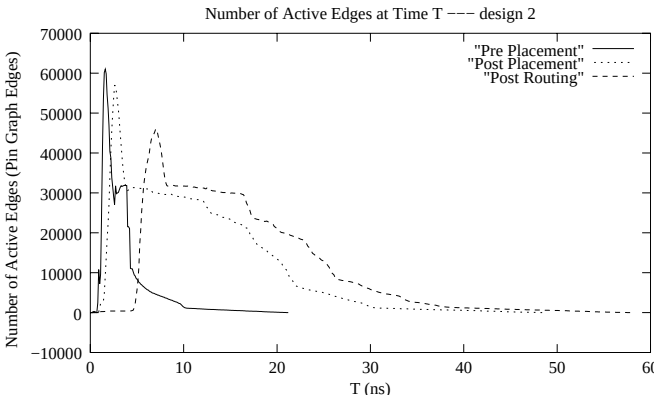
3.3 Temporal Analogue to Rent's Rule

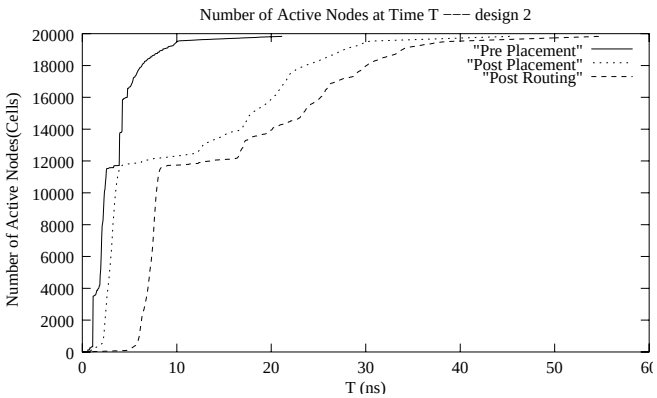
Our final graphs plot the edges analogous to Rent's rule, on which they are applied in the timing domain. Instead of finding relationships between pins and nets, we are looking at relationships between pins and communicating nets. Table 2 compares the Rent components to the proposed temporal components.

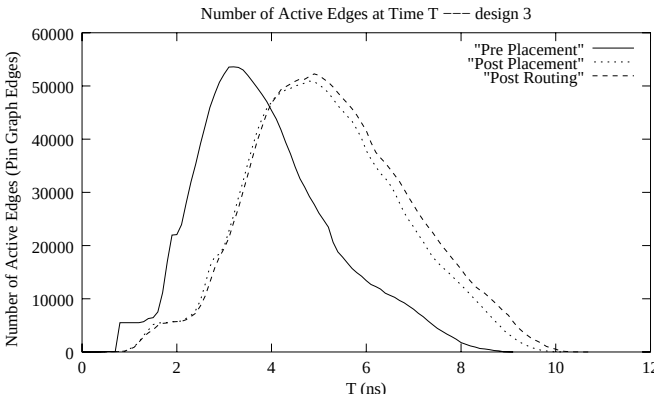
Rent's rule states that there is a relationship between the number











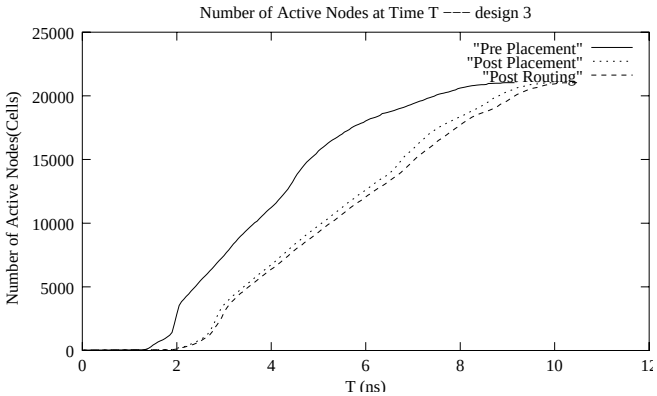


Figure 4: Clock cycle activity graphs for three different designs (from top to bottom: design1, design2, design3). The curves represent the number of edges that are active at time t. Each figure has three lines corresponding to the pre-placement, post-placement and post-routing distributions.

Figure 5: Growth of activity nodes for three different designs (from top to bottom: design1, design2, design3). The curves represent the number of active nodes that have switched by time t. Each figure has three lines corresponding to the pre-placement, post-placement and post-routing distributions.

	Rent's parameter	Temporal analogue
Basic counting element (x-axis value)	Cell	Time slice
Basic variable element (y-axis value)	External net	Signal
Counting domain	Chip area	Time axis
Largest possible domain	Entire chip	Entire clock cycle
Condition to "count" variable element	Connection from cells inside area to cells outside area	Data sent from inside (outside) time slice to outside (inside) time slice

Table 2: Comparison of basic elements in Rent's Rule to the temporal domain.

of cells inside an area/partition and their connections to cells outside that area/partition. The equivalent in the time domain would be somewhat similar. The relationship would still be between pins inside and outside an "area" except, the area would be temporal, i.e., a time slot. The "connections" would be both physical and temporal which mean that a net must physically connect a set of pins, but the net may or may not be actively communicating during a certain time period. We call a net as a *signal* if it is actively communicating data between pins. So, the equivalent Rent's rule area (or partition) would be a time slot. While the largest area in the traditional Rent's rule is the entire chip, the largest time slot (or area) in this temporal Rent domain is a clock cycle. The traditional net is replaced by a signal (a communicating net).

Given a time slot delta, we want to measure the amount of external communications done between that time slot and the "outside" or remaining time in the cycle. A communication could be defined as a signal starting inside the time slot and ending outside of the slot and/or starting outside the time slot and ending inside the time slot. A signal that starts and ends inside a time slot would not be considered. An initial experiment is done using the active interval as the communication window. By varying the size of the time slots, we can effectively get similar measurements in the time domain as Rent's rule does in the space domain.

The exact formulation is as follows:

1. Let M be the duration of time interval, $0 \leq M \leq CP$, where CP is the clock period.
2. Let $B(M)$ be the "boundary size of M", i.e., expected (average) number of active intervals that have exactly one endpoint inside a time interval of duration M.
3. A time interval of duration M is of form $[a, b]$ where $0 \leq a$, $b = a + M \leq CP$. if $M \to 0$, then $B(M) \to 0$; if $M \to CP$, then $B(M) \to 0$

Figure 6 shows the graphs for the three designs.

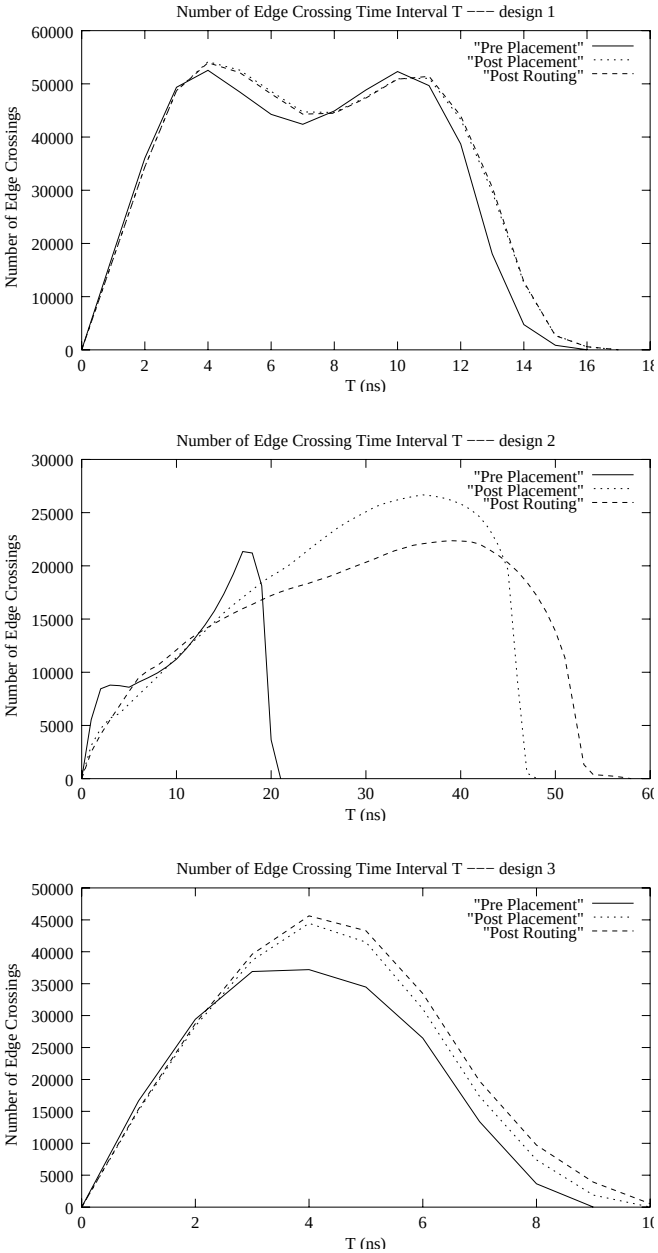


Figure 6: "Temporal Rent's Rule" analysis for three different designs (from top to bottom: design1, design2, design3). The curves show average number of boundary size for time interval T. The x-axis represents the size of the time interval. The y-axis represents the boundary size for a given time interval. Each figure has three lines corresponding to the pre-placement, post-placement and post-routing distributions.

4. ANALYSES OF SPECIAL-CASE CIRCUITS

In this section, we will apply the same methodology on some special circuits. A special circuit is an artificial circuit with special topology. It could be a *ring*, a *binary tree*, a *mesh*, a *clique*, or a *Rentian* circuit.[2] Analysis of these special circuits will help us to understand the new metrics defined in the previous section.

To analyze these special circuits, we make some assumption to simplify the circuits. First, we assume that all the gate delays in the special circuits are zero. Thus we can focus on the wire delay in the analysis. Second, we assume that all the wires have unit delay for all special circuits. The only exception is the clique circuit. We will discuss clique circuit under two assumptions of wire delay: unit wire delay, and wire delay proportional to wire length. Finally, we specify the signal directions in special circuits to obtain physically correct designs.

For each type of special circuit, let n be the number of cells, the clock cycle activity graph for this circuit will be drawn. The other important figure, active interval graph, is straightforward since all the wires have unit delay. Only in the clique circuit case, when wires have different delays, we draw both clock cycle activity graph and active interval graph.

4.1 Simple Circuits: Rings, Binary Trees, Meshes and Cliques

Ring

A ring circuit consists of a set of cells with one input and one output. The cells are connected in serial and the output pin of last cell is connected to the input pin of the first cell. The circuit and its corresponding active-edge curve is shown in Figure 7. In a ring circuit, the number of active edges at a given time will always be a constant because there is no delay uncertainty and when an edge becomes active, the previous active edge becomes inactive at the same time.

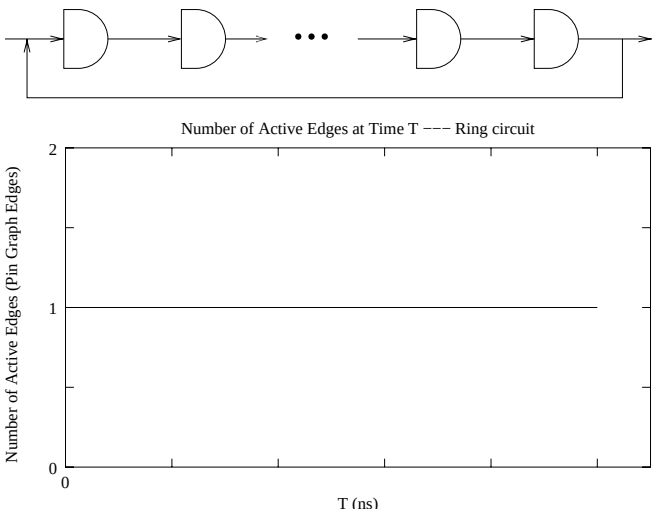


Figure 7: A ring circuit and its clock cycle activity graph.

Binary Tree

A binary tree circuit consists of a set of cells with one input and one output. However, each output pin of a cell is connected to the input pin of two other cells, i.e., all internal nets have 2 fanouts. The circuit and its corresponding clock cycle activity graph is shown in Figure 8. As time progresses, the number of active edges doubled every time a signal is propagated out from a cell.

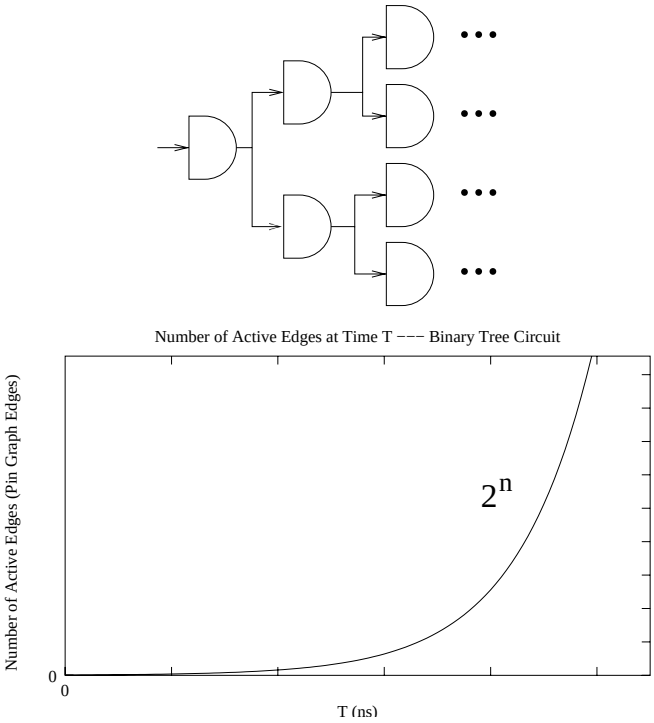


Figure 8: A binary tree circuit and its clock cycle activity graph.

Mesh

A mesh circuit consists of a set of cells with two inputs and one output. Each output pin of a cell is connected to one of the input pins of two other cells (a 2 fanout net). The circuit and its corresponding clock cycle activity graph is shown in Figure 9. Similar to the binary tree circuit where as time progresses, the number of active edges increases. However, for mesh circuits, the increase of the number of active edges is linear, instead of exponential in binary tree circuits.

Clique

There are two variations for clique circuits. In the first case, we assume that wires have different delays, and the delay of a wire is proportional to its length. For example, the wire delay between the first cell (the source cell that has $n-1$ external connections) and the second cell (the cell that has $n-2$ external connections) is one unit. However, the wire delay between the first cell and the last cell (the sink cell that has $n-1$ inputs and no output) is $n-1$ unit. In other words, the wire delay between cell c_i and cell c_j is the maximum number of hops from c_i to c_j, regardless of the existence of a direct connection between c_i to c_j. Figure 10 presents the clique circuit and its clock cycle activity graph under this assumption. The growth of active edges for this variation of clique circuits follows an inverse quadratic curve where it reaches its peak at the middle of the time period.

If we assume that all wires have the same unit delay, the net edges generated from wires have delay uncertainty. Thus the active-interval

[2] A Rentian circuit is a circuit which obeys Rent's rule.

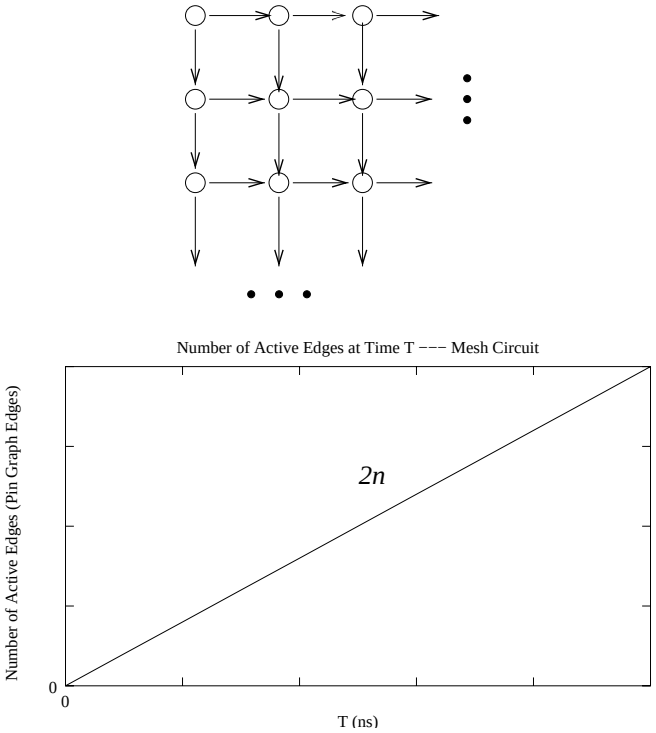


Figure 9: A mesh circuit and its clock cycle activity graph.

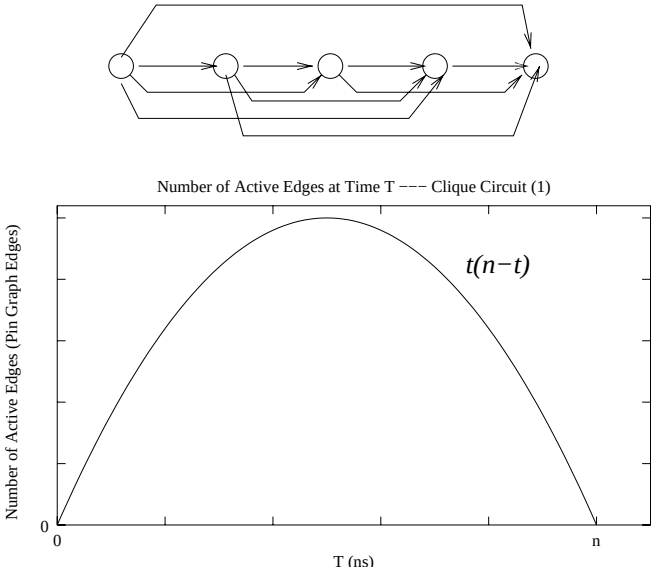


Figure 10: A clique circuit and its clock cycle activity graph.

curve is not trivial. We draw both active-interval and active-edge curve in this case. Figure 11 shows these two curves.

4.2 Rentian Circuits

A Rentian circuit is a circuit that obeys Rent's Rule. Rent's rule is an empirical observation first described by Landman and Russo [6]. It states the relationship between the number of cells B in a subcircuit of a partitioned design, and the number of external connections P of the subcircuit. Specifically,

$$T = AB^p$$

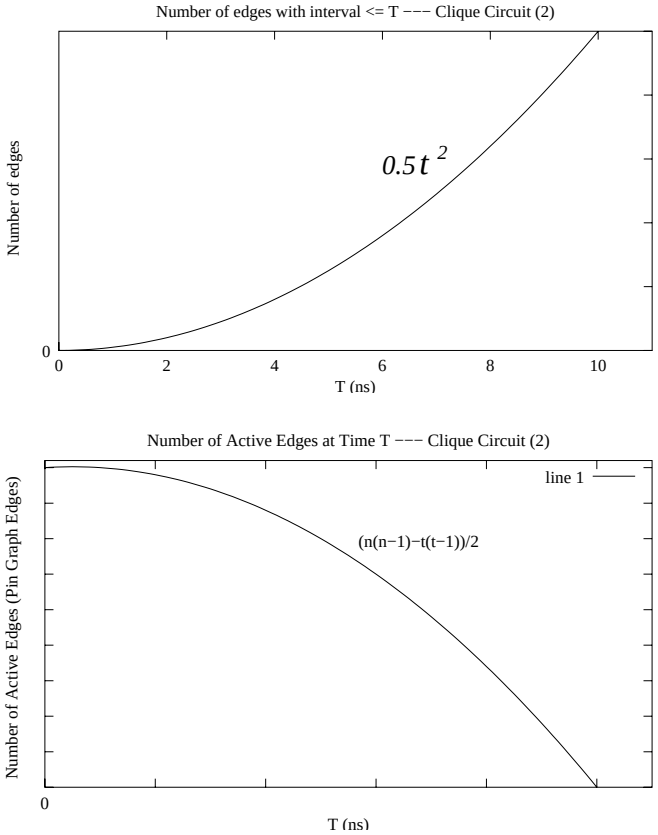


Figure 11: A clique circuit's active interval distribution (top) and clock cycle activity graph (bottom).

where A is the average number of nets per cell and p is the Rent exponent.

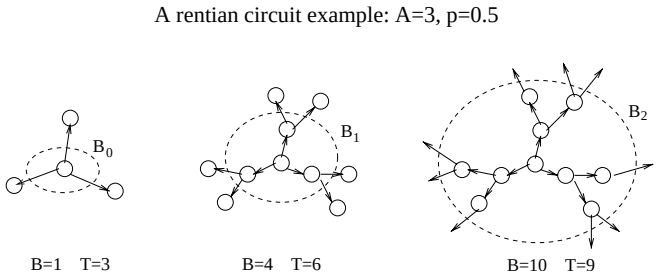


Figure 12: Use breadth-first search to define edge direction for "Rentian" circuit.

To analyze the timing structural properties for an ideal Rentian design, we specify the wire directions as the following. For the sake of simplicity, we assume that all nets are two-terminal nets and there is only one source node in the design.[3] From the source node we do breadth-first search (BFS). The direction of each edge created in BFS is defined as from the lower level (early visited) node to higher level (late visited) node. Figure 12 illustrates the edge directions for a Rentian design.

Also, Figure 12 shows an example of how the number of active edges varies as the time passes. Assume that all wires have one unit delay.

[3] This analysis can be extended for more complicated cases.

At the beginning (time interval [0,1]), nets that are connected to the source node are active.[4] Thus there are $A = 3$ active edges. At the next step, three newly included nodes and the source node form the subcircuit B_1. According to Rent's rule, the number of external connections of this subcircuit is $T = A|B_1|^p = 6$. Thus the number of active edges at time interval [1,2] is 6. Similarly, the size of subcircuit B_2 is 10 and the number of edges at time interval [2,3] is 9.5.

To summarize, the relation between the number of active edges $T(t)$ and time t can be described by the following recurrence:

$$\begin{aligned} T(0) &= 1; \\ T(t+1) &= A\Big[\sum_{i=0}^{t} T(i)\Big]^p \end{aligned}$$

The clock cycle activity graph corresponding to the above recurrence is shown in Figure 13. It should be noted that this is only a partial curve since there is only one synchronizing cell in the circuit. A more reasonable Rentian circuit is shown in Figure 14. There are two sequential cells at opposite ends of the "diameter" of the netlist. The circuit is symmetrical: if we reverse the signal directions for the right half of the circuit and do BFS from the right sequential cell, the same clock cycle activity graph will be derived. Therefore the clock cycle activity graph for the entire circuit can be obtained by combining the curve in Figure 13 and its mirrored curve (as shown in Figure 15).

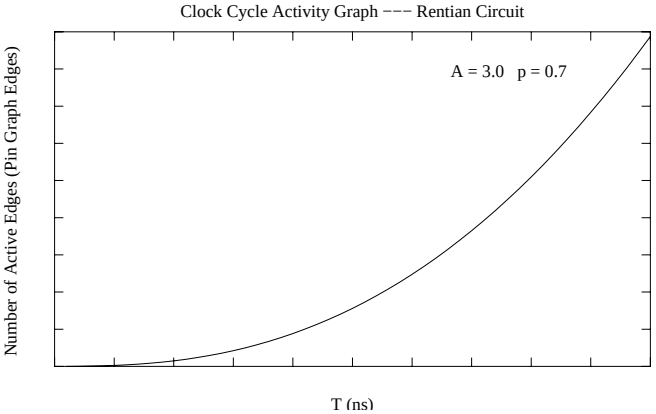


Figure 13: Clock cycle activity graph for a Rentian circuit.

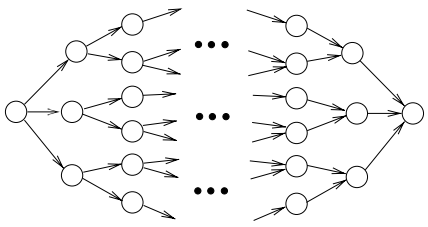


Figure 14: A symmetrical Rentian circuit which has two sequential cells at opposite ends of the netlist.

The similar curves can be obtained if we assume that multiple sequential cells appear at both ends of the netlist. For example, if there are two source cells and two sink cells in the circuit, the corresponding curve will have a higher beginning point, while the entire shape of the clock cycle activity graph remains unchanged.

[4] We ignore Region III phenomenon [7] in this analysis.

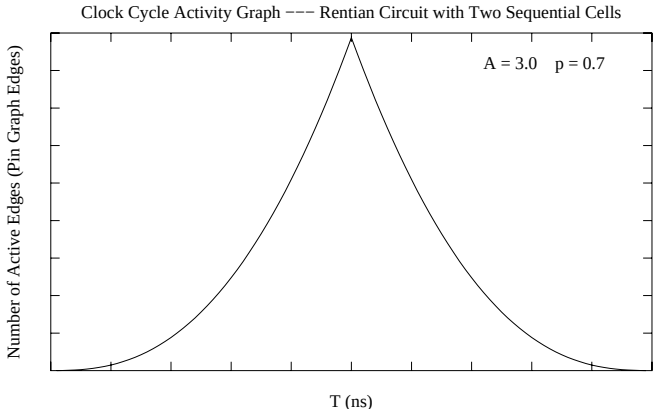


Figure 15: Clock cycle activity graph for Rentian circuit with two sequential cells at opposite ends of the netlist.

Not surprisingly, the clock cycle activity plot shown in Figure 15 has the similar shape with the curves of design1 and design3 in Figure 4. This is another observation of Rent's rule in designs with timing information.[5] For a specific design, the Rent parameter, the topological structure and the clock cycle will determine not only the shape of the curve, but also the details of the curve, i.e., sharpness, peak value, symmetrical or not, etc. Studies of these issues will improve designer's ability to predict a design's performance at early design stages.

5. CONCLUSION

We have proposed studies of circuit timing structure that can potentially help identify useful characterizations of the timing behavior of the design. We presented several design characterization models that characterize the timing behavior of the design. The new models allow us to identify designs that maybe hard to meet timing closure. Using similar analogy to the Rent parameter, the new characterization models, the temporal Rent, analyze the number of active signals intersecting a specified timing boundary within the clock period.

6. REFERENCES

[1] K. D. Boese, A. B. Kahng and S. Mantik, "On the Relevance of Wire Load Models", *ACM Intl. Workshop on System-Level Interconnect Prediction*, April 2001, pp. 91-98.

[2] W.-J. Dai, "Hierarchical Physical Design Methodology for Multi-Million Gate Chips", *Proc. Intl. Symposium on Physical Design*, April 2001.

[3] P. Gopalakrishnan, A. Odabasioglu, L. Pileggi and S. Raje, "Overcoming Wireload Uncertainty During Physical Design", *Proc. Intl. Symposium on Physical Design*, April 2001.

[4] A. Lu, H. Eisenmann, G. Stenz and F. M. Johannes, "Combining Technology Mapping with Post-Placement Resynthesis for Performance Optimization", *Proc. Intl. Conf. on Computer Design*, October 1998, pp. 616-621.

[5] L. Scheffer and E. Nequist, "Why Interconnect Prediction Doesn't Work", *ACM Intl. Workshop on System-Level Interconnect Prediction*, April 2000, pp. 139-144.

[6] B. Landman and R. Russo, " On a Pin Versus Block Relationship for Partitions of Logic Graphs", *IEEE Transactions on Computers*, c-20:1469–1479, 1971.

[7] D. Stroobandt, "On an Efficient Method for Estimating the Interconnection Complexity of Designs and on the Existence of Region III in Rent's Rule", *Proc. of the Ninth Great Lakes Symposium on VLSI*, pages 330–331. IEEE, March 1999.

[5] Previously Rent's rule is only observed by analyzing the topology of the designs.