ESTIMATION TECHNIQUES TO GUIDE LOW-POWER RESYNTHESIS ALGORITHMS FOR COMBINATIONAL RANDOM CMOS LOGIC

by

Christopher Kevin Lennard

Memorandum No. UCB/ERL M95/75

22 September 1995

ESTIMATION TECHNIQUES TO GUIDE LOW-POWER RESYNTHESIS ALGORITHMS FOR COMBINATIONAL RANDOM CMOS LOGIC

by

Christopher Kevin Lennard

Memorandum No. UCB/ERL M95/75

22 September 1995

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Abstract

Estimation Techniques to Guide Low-Power Resynthesis Algorithms for Combinational Random CMOS Logic

by

Christopher Kevin Lennard

Doctor of Philosophy in Engineering-Electrical Engineering and Computer Sciences

University of California at Berkeley

Professor A. Richard Newton, Chair

Existing resynthesis techniques for the minimization of power consumption in random combinational CMOS logic are limited by their inadequate prediction of expected change in global network power. An abstraction hierarchy for the estimation of global power change is developed in this dissertation. Within this hierarchy the maximum accuracy of estimation can be defined for decisions made at any point in the synthesis design flow. If the effect of change in a specific network property is not predictable at one level of the hierarchy, then synthesis choices have to be made insensitive to that property or restricted in such a way as to make estimation more accurate.

Three properties are used to define the form of a resynthesis step: change in function, change in spurious dynamic activity and change in delay. The prediction techniques developed herein for change in global power due to change in functionality and change in spurious dynamic activity are accurate at high levels of abstraction. While prediction of the effective change in power due to change in delay is not as accurate, the magnitude of effect of delay on power is the least critical of the three properties.

Professor A. Richard Newton
Dissertation Committee Chair

Contents

Find your lead,

Toffa,

let's go for a walk somewhere new.

List of Figures

List of Tables

This is the beginning of the end. In a voyage from an exam in a dinky little office at the University of Melbourne, Australia to the final page of this dissertation, the desire of youth is fulfilled. To that examiner and the advisor who has helped me identify and pursue valuable research opportunities thoughout the past six years, Prof. A. Richard Newton, I thank you. The financial support of the SRC (95-DC-324) during my academic stay was tremendous, as has been the computing environment provided by Digital Equipment Corporation. I would like to credit Motorola, in particular Noel Strader, for providing crucial industrial insight and I gratefully acknowledge the careful reading of this dissertation by Prof. Jan Rabaey and Prof. Henry Helson.

As the plane was towed away from the Tullamarine terminal and my furiously waving family, the chance to participate in study at one of the world's most recognised universities dominanted my thoughts. It came somewhat as a surprise to me just how quickly my experience here became defined by social interaction; not by the institution itself, not by the nation in which this school is entrenched. Mark, one of the Queen's subjects like myself, soon became a friend who I now seem to have known for longer than I can remember. Henry, a guy who appreciates that a video game can be a worthwhile cause for shelling out $10.-, has been at the other end of over-the-cubical-wall conversations for more than four years. This friendship is a tribute to our mutual interest in the finest science-fiction series ever produced, Dr. Who. Then there's Desmond, "Mr. Personality" and house mate of a over a year. If we were directors, all movies would be perfect. And, of course, Eric completes the lunch crowd. He adds vigour to many a debate held over a plate of beef, spinach and peanut sauce and has been a great cube-mate. Gone south but not forgotten, ex-lunch crowd member and walking encyclopaedia, Ken, who is a great addition to any trip to the symphony. There's also the other Mark (non-Canadian version) who joins Des and me in critiquing movies over beer. If it wasn't for him, I don't think I could ever have been able to ride to the top of the Berkeley Hills. Conversations with Don, a great friend, good listener and self-proclaimed philosopher have been a great help in alleviating many of those things that tear at the soul. Good Xmas turkey, too, so many thanks are owed to his family. Flora, with her sparkling personality and her Jo-Jos elevates the office experience far beyond the functional. Cheers also to Gary (Mr. BBQ), Jess (Ms. Paleo), Soren, Nicola & brand new family, Dot, Jaijeet, Tara, Tim, Michelle, Chirag, Kia, Adnan, Rajeev, Edoardo and Gitanjali.

In my six years here, one person in particular has become very significant to me.

Heather, you show me happiness I had given up for lost an aeon ago. Your support in my search for self over these past two years has been invaluable.

I have fallen badly out of touch with my Australian roots, but thanks are owed to many people there. I would like to extend my gratitude to Dr. Badcock my undergraduate communications professor, Mrs. Beecroft for the grueling Latin classes and the general encouragement of my pursuits, Mr. and Mrs. Van and Mr. Mills for the rigorous grounding in maths and science. In particular, I would like to thank my guitar teacher and good friend, Peter Lynch. Through him, the importance of music in my life was realised.

But it is my family to whom the greatest thanks are due. I remember with great fondness Grandfather Salt ("Poppa") who inspired my interest in the way machines worked. My desire to do engineering is in no small part due to his intricate wooden sand-pit toys and trips down to the paddle-steamers. My Uncle Richard showed great faith in my desire to pursue further studies, and his outlook on life is always refreshing. Kevin, my Dad, stimulated to my interest in mathematics and logical thinking; Jennifer, my Mum, has extended my perception of the human angle and has been tremendously supportive of intellectual achievement. Adrian, my younger brother and best friend, stops me from aging too fast. If it weren't for him, his off-the-wall sense of humour and riffs in Jimmy's key down 9000 miles of telephone line I would find the world a much more serious place. The love I have always felt from all these people from so far down under is an eternal source of comfort.

Finally, I wish to thank the unconditional affection afforded to me by the most wonderful dog on this planet, Toffa. Despite his surprising intellect, I don't think he ever did grasp why his master was absent for 42 of his dog years. To him, I extend my sincerest apologies.

So I reach the end of my student life. To quote the eternal wisdom of Monty Python:

"And now for something completely different ...."

Chapter 1

Introduction

Low-power design is fast becoming an issue as critical to digital chip design as the optimization for area and speed have been for the last decade. The search for optimality in this domain is motivated by the proliferation of portable electronics and the ever increasing device density on silicon.

Portable electronics must have self contained energy sources, the primary choice for this purpose being batteries. Batteries are bulky items, sometimes contributing more than 50% of the entire mass of hand-held devices such as cellular phones. Continuous operation of many portable electronic devices will draw sufficient current to deplete the energy source in less than the length of a standard working day. However, there is a demand for increased sophistication of portable applications and this is creating a need for faster and more computationally complex operations. As all the companies which compete in the development of portable electronics have access to the same battery technology, market share will primarily be increased by improving power efficiency of the products. Design specifically for the reduction of power consumption has consequently become an issue of vital importance at all levels of the design process, from system architecture and software optimization to combinational and sequential structure and device-level technology choices.

1.1 Structure of the Low-Power Design Problem

Techniques for reduced power consumption must be applied at every level of design abstraction. A simplified representation of the various levels of design abstraction for digital systems is depicted in Fig. 1.1. Note that a discussion of synchronous (clocked) vs.

asynchronous (not clocked) design is not presented here. Theory for synchronous design is currently more advanced than corresponding theory for asynchronous networks. Synchronous design strategies are likely to remain prominent for many years to come.

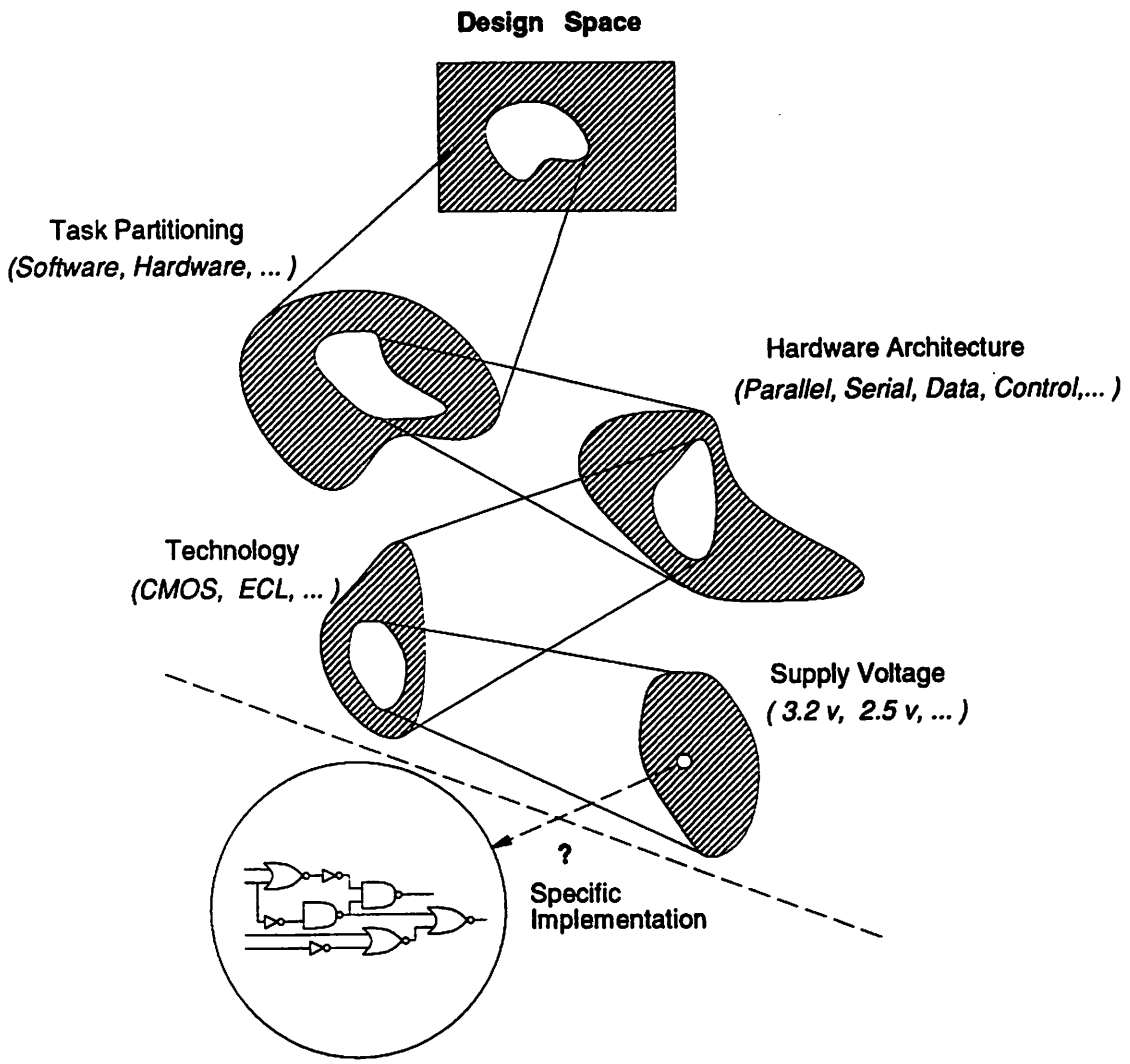

Figure 1.1: The High-Level Design Flow

The goal of a designer using the representations shown in Fig. 1.1 is to find an optimal implementation from all of the possible designs which correctly implement the input/output relations for the desired rate of throughput, while also minimizing average power consumption. Each section *"cut"* from the design space at a higher level is the set of designs within that space which satisfy specific decisions which are made. For the work presented in this dissertation, it is assumed that the architecture (for example, parallel or

serial processing, degree of data path pipelining, etc.), technology and power supply have been chosen (e.g. [12]). In particular, fully-static Complementary MOS (CMOS) logic is the technology assumed. This logic is the most common technology used in micro-processor and prototype ASIC designs. The properties which make this logic so prevalent are its high immunity to noise, robustness and ease of design and small dc power consumption. The material in this dissertation addresses the final stage of the design process - the extraction of a logic description which best implements the desired functionality.

The final optimization step which extracts logic gates from a description of functionality has two parts:

1. *Optimization of Structured Logic.* Structured logic has a recognizable operation on which a designer can utilize specific optimization techniques. Structured logic includes adders, multipliers, parity checkers, etc.

2. *Optimization of Random Logic.* Random logic has an input/output relation which cannot be described abstractly by anything other than its boolean functionality. Random logic is often associated with chip control functions. These functions vary significantly between applications.

There is a large body of work completed in the area of optimization of structured logic for reduced power consumption. These techniques reduce the number of power-hungry operations by optimizing the representation of arithmetic expressions. This is summarized extensively in [6].

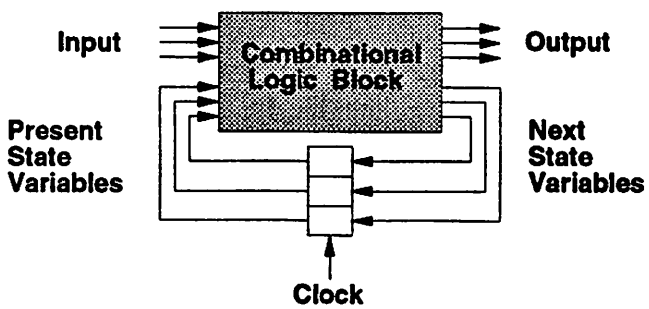

Figure 1.2: Synchronous Mealy State Machine

The optimization of synchronous random logic has two major phases: state encoding and implementation of the combinational logic block. The *state* of the synchronous logic is implemented by clocked latches, the *combinational logic block* is an interconnected set of

logic gates which determine the functionality associating the next state with the previous state and input variables (Refer to Fig. 1.2). The problem of finding an optimal encoding is tightly linked with the ability to predict the power consumption of combinational logic block. Although it is desirable to choose an encoding which minimizes the total number of state bits changing when moving from one high probability state to another (thereby minimizing the transition power in the latches), the savings gained can be offset by an increase in power consumption within the combinational logic. The empirical evidence presented in [11] supports this observation. This motivates the necessity to develop an understanding of considerations important to optimization of the combinational logic block of a sequential network.

The automated design of a combinational logic block for a controller which does not take into account the correlation between present and next input states is referred to as *Combinational Random Logic Synthesis* (CRLS). Optimization without sequential correlation is the precursor to *Sequential Random Logic Synthesis* (SRLS) which is needed to evaluate the power optimality of a state encoding. Study of combinational random logic synthesis for low-power is a relatively recent pursuit (the earliest papers addressing the problem were published in 1992, such as [20]). Any decision which is made to improve optimality depends upon accurate estimation of the influence of a synthesis decision upon network power. It is in this area that the material of this dissertation contributes.

1.2 Estimating Power Dissipation

CMOS logic implies logic gates for which the function is implemented both in nMOS and pMOS transistors. When the output is 1 (0), at least one path through the pMOS (nMOS) logic has all transistors in an ON state implying gate inputs 0 (1), and all paths in the nMOS (pMOS) logic contain at least one transitor OFF with gate input 0 (1). This situation is portrayed for a 2-input NAND gate in Fig. 1.3

Power is consumed in CMOS logic as a consequence of three factors:

- *Leakage Current.* The path from the voltage supply rail to ground has a finite resistance, even when every path contains at least one OFF transitor.

- *Short-Circuit Current.* Input transitions to a gate take a finite time. During the transition, there may exist a period where a path from supply to ground has a small

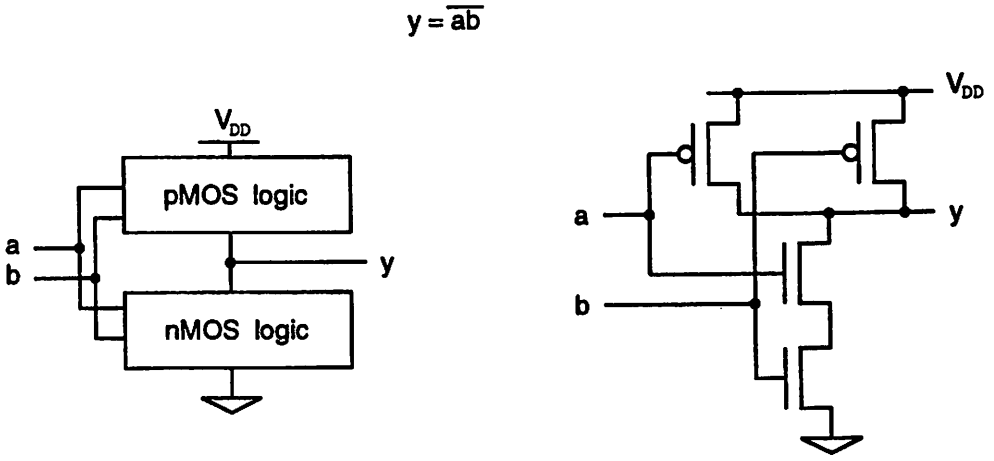

Figure 1.3: Complementary Logic Example

resistance. The current which flows directly from supply to ground as a consequence of this effect is known as Short-Circuit Current.

- *Capacitive Charging Current.* During an input transition, some transitor capacitances will be charged up, others discharged. The current consequently drawn from the supply is the Capacitive Charging Current.

The capacitive charging current is generally the most dominant effect in modern CMOS circuits. Power consumption due to leakage makes an almost negligible contribution to total power when a CMOS circuit is actively switching at typical clocking rates. Short-circuit current is only significant at a gate if the input transition time is very slow relative to the output transition time. Slow transition times arise as a consequence of excessive fanout loading of particular gates in the network. This is undesirable in a design as it makes accurate prediction of circuit delay difficult by requiring detailed analog analysis of digital circuit elements. It may therefore be assumed that such slow transitions are unlikely in well-designed networks. This eliminates the need to compute the power contribution from short-circuit current.

There are five major parasitic capacitances which must be modeled to describe accurately the dynamic behavior of a MOS transitor; the capacitance from gate to drain C_{gd}, gate to source C_{gs}, gate to substrate C_{gb}, drain to substrate C_{db} and source to substrate C_{sb}. All of these capacitance vary with voltage, but to a first order approximation the input capacitance, $C_g = 2 * C_{gd} + C_{gs} + C_{gb}$, is constant. This constant capacitance assumption

is not as valid for the output which has contributions from C_{db} and C_{sb} but, in general, the intrinsic output capacitance of a gate is dominated by external loading effects. This implies that, to first order, the dynamic behavior of a CMOS gate can be described by a *lumped* capacitance model [9]. The transitors in a CMOS gate are then modeled as ideal voltage controlled current sources driven by a capacitively loaded input signals.

Even for this simpler model, the complexity of computing the average power consumption is extremely high. An estimate of average power requires that a network be simulated with a large number of input vector pairs or that expressions describing the *entire* functionality be established (Refer to Sec. 1.2.1). Handling the highly non-linear output response of a gate in this context is very difficult. A linearization of the output response is a further simplification which can be made. In this case, the current drive of a transitor is assumed constant when it is switched on, thereby generating ramp outputs. However, the general model for gate behavior in a digital network can be reduced even further. This model, known as the *timing model*, describes the output behavior after an input change as constant until after a specific gate delay at which point the output may transition instantaneously. That is, the gate is modeled as an ideal switch with delay d. The delay, d, corresponds to the time taken for the analog gate output to cross the half-way point between the ground and voltage rail. The behavior description for these three simple models relative to an inverter is illustrated in Fig. 1.4.

The timing model is simple, but has also demonstrated itself to be sufficiently accurate for synthesis which optimizes network speed [21]. The accuracy is a property of the fact that a gate can be viewed as an open-loop amplifier with extremely high gain. When an input transition passes the input sensitivity point, the gate quickly limits to maximum current drive which will charge or discharge the output capacitance. The sensitivity point for a gate is usually designed to be close to half way between the source voltage and ground. This reduces the dependence of network delay on the strongly non-linear behavior of the actual output waveform near the voltage source and ground, as well as improving noise immunity. Consequently, information concerning *only* the half-way crossing can be used to estimate timing quite well. This timing model is used in a large number of power estimators, the simulators described in [5] [7] [15] [23] being four highly regarded examples.

The limited behavioral information contained in a timing model of a gate can result in an error in predicting the total transition count and transition timing at a node in a network when input transition arrival times are similar but not simultaneous. For

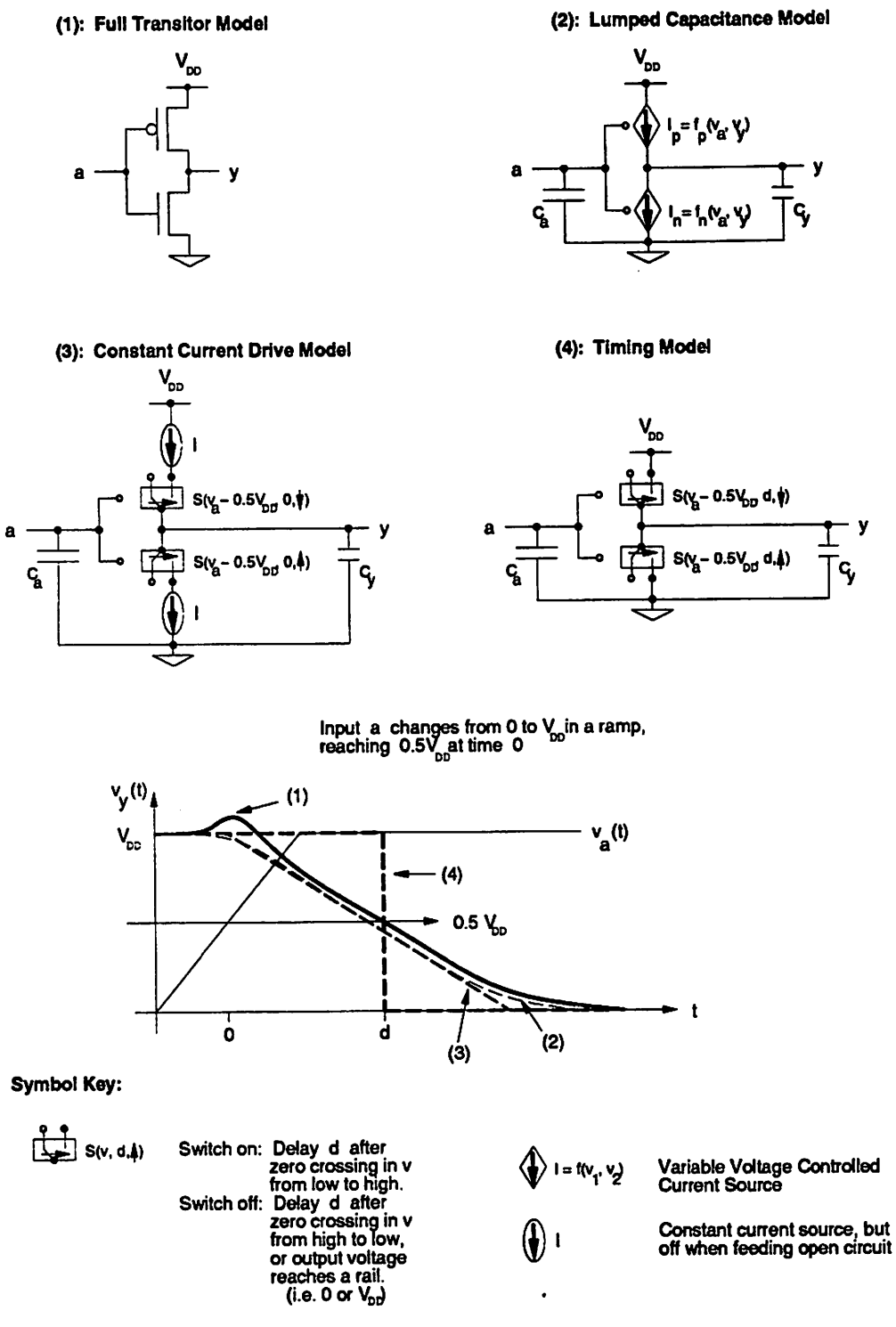

Figure 1.4: The Timing Model for a CMOS Inverter

example, consider the analog output waveforms $x(t), y(t)$ for the circuit shown in Fig. 1.5 for the full transistor model of the gates. The transitions on input b occur soon after a transition on input a. The delay between the input transition arrival times is similar to the timing delay, d_{NAND}, of the NAND gate. The output of the gate does not complete full transitions between the V_{DD} and ground in response to each input change. It effectively reacts faster to the transitions on input b than it would if that input alone changed. This is a consequence of the output state not being at a voltage rail when the input transition arrived. However, the simple timing model for the gate is constructed from the gate response assuming simultaneous input transitions so it assumes that the gate has settled before an input change occurs. This is an invalid assumption when the delay between the input transition arrival times is similar to the timing delay of the NAND gate. The simple timing model therefore has an inherent error for these complex arrival configurations. The error can propagate through transitive fanout gates, such as the inverter in this example.

There is a technique in timing simulation for reducing the over-estimate of transition activity. It uses a principle know as *inertial delay modeling* which is a post-processing (non-causal) filtering of transition activity. If the output transitions using the timing model are closer together than the delay of the gate, then the transitions are removed. However, as shown in Fig. 1.5, the best this technique can do is bound the possible signal delay. Either the earlier or later two output transitions of the NAND gate could be assumed non-plausible and so be filtered out. An empirical study performed in [11] for a vector-driven power simulator shows that varying the filtering approach in this way can alter the power estimate by up to 10%. In general, estimation of average power using the timing model described previously does not perform inertial delay filtering. Although this may result in an over estimate of the transition activity, it identifies points within the network which are highly sensitive to network delays. That is, nodes for which the functionality alone is not sufficient to filter out spurious dynamic activity. This is useful information to guide synthesis tools and most synthesis routines are tested using simulators based upon this simple timing model (e.g. [20], [10], [2], [22], [24]). This timing model for gate behavior is used throughout this dissertation.

In summary, the assumptions in the simplified CMOS model are: (1) all capacitance is lumped at the output node of a gate; and (2) current flows only from the supply rail to the load capacitor, or current flows from the load capacitor to the ground rail;
(3) all voltage changes are full swings from the supply rail to the ground rail voltage,

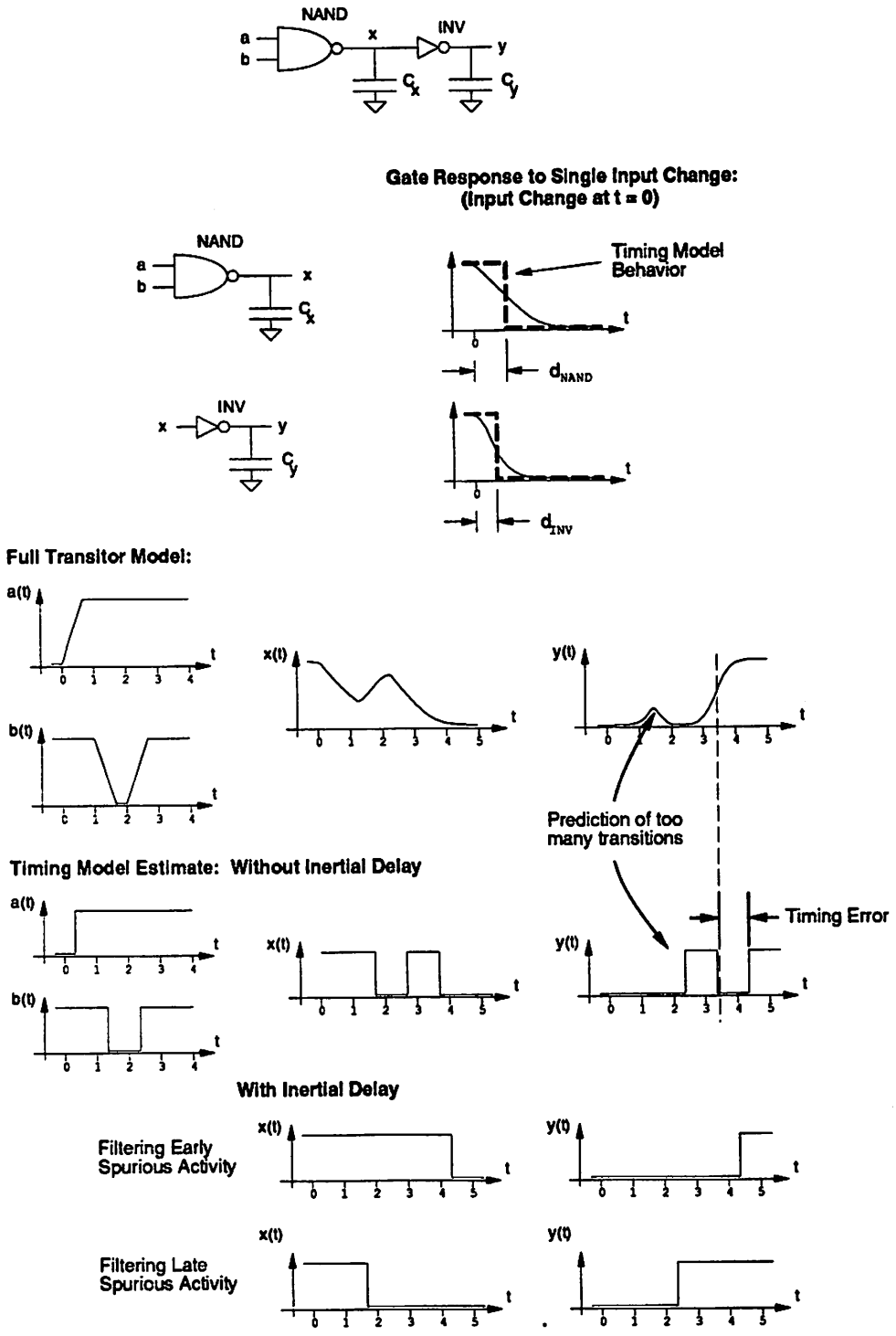

Figure 1.5: Inertial Delay and the Limitations of Timing Model Accuracy

or vice-versa; and (4) input/output transition time constants for a gate are the same order of magnitude. The energy consumption of a CMOS circuit is directly related to the switching activity when this simplified model of energy consumption is used. For a well-designed gate, the above assumptions are reasonable [8].

For a digital system with input latches clocked at a fixed frequency f, the power drawn from the supply by transitions at gate g_i is given by:

$$P_i = \frac{1}{2} \cdot C_i \cdot V_{DD}^2 \cdot f \cdot T_i \tag{1.1}$$

where P_i denotes the average power dissipated by gate g_i, C_i is the load capacitance at the output of gate g_i, V_{DD} is the supply voltage, and T_i is the average number of gate output transitions per clock cycle. Given a technology-mapped circuit or a circuit layout, all of the parameters in Eqn. 1.1 can be determined, except for T_i, which depends on both the logic function being performed and the statistical properties of the primary input signals.

1.2.1 Previous Work in Power Estimation

There are several techniques which have been developed for estimating average switching activity in combinational networks. They can essentially be divided into three categories:

- *Symbolic Analysis.* [7]
- *Probabilistic Analysis.* [15] [23]
- *Statistical Analysis.* [5] [11]

Symbolic analysis is a technique which obtains an exact result for transition activity given the simple timing model for gate behavior. All gates have a delay which is an element of the rationals, so there exists a non-zero largest common divisor, d_{GCD}, for these delays. This can be used to discretize time as it guaranteed that any transition *must* occur on an integer multiple of d_{GCD}. The functionality for the network for each time interval in the discretization can then be computed. A transition occurs if the functionality of a node in one time interval is not the same as that in an adjacent interval. To reduce the computational complexity, an implementation of the algorithm uses event-driven construction of equations. For example, with a gate of delay d, if no input transition occurs at time t then there can be no output transition at time $(t + d)$. This estimation strategy was

first proposed in [7]. The timed functionality is obtained by constructing a network with an output representing the functionality each node for every timing interval. The outputs representing adjacent time intervals for the same node are then XOR'ed together. The output of the XOR gates is therefore the functionality of any transition activity between the intervals which its inputs represent. From the construction of output functions for this network, the transition probabilities are computed. This analysis uses ordered Binary Decision Diagrams (BDDs) to represent the functionality. The BDD is an implicit graph-based functional representation [1] which is provably canonical when an ordering is applied to the variables [4]. This representation has been found empirically to be efficient for network synthesis. However, the complexity of full symbolic analysis for power estimation is still too high for use on large industrial networks.

Probabilistic analysis describes the set of techniques which use probability rather than functionality to describe average transition activity. For example, consider a two-input AND gate of delay d with inputs $\{a, b\}$ where any transition on a arrives at least delay d later than any transition on b. If input a has a probability 0.5 of being 1 at a time t when input b has a probability of 0.5 of a *transition*, then the probability of an output transition at time $(t + d)$ could be estimated as: $0.5 \times 0.5 = 0.25$. Because this is an arithmetic rather than logical manipulation, these algorithms generally have much smaller computational complexity than full symbolic simulation. However, the accuracy and complexity of the algorithm strongly depends upon the ability to account for correlation between inputs. Suppose for the example stated that the probability of a being 1 *given that* input b transitions is only 0.25. This implies that the actual output transition probability at time $(t+d)$ is $0.25 \times 0.5 = 0.125$, the error in the previous estimate being a property of an input independence assumption. There have been several estimation techniques proposed which account for input correlation with varying degrees of accuracy. The work of [15] models waveforms as independent strict sense stationary ergodic processes. They suggest a partitioning of a network into tightly interconnected modules. Inside these modules, their waveform model is invalid so more complex estimation strategies must be used. They found that most networks easily partition into small modules which are very well suited to the independence assumption. The estimation strategy proposed in [23] can be applied within networks with high correlation between internal nodes. In this case, the steady-state functional correlation between nodes in the network is computed. This correlation is assumed over all time during dynamic transition activity. From the results in this paper

it is claimed that an error of less than 4% in the estimation of average network power and speedup of about a factor of 10 can be obtained relative to the exact symbolic simulation tool of [7]. There are no computationally inexpensive probabilistic analysis techniques which can account for input correlation exactly. Exact correlation requires accurate information regarding network functionality over all time. If such data were available, then this technique would be comparable to symbolic analysis in its complexity.

Statistical analysis techniques are based upon fast timing simulators. Random input vectors are selected and applied to the input of the network. The power computed in the simulation is then added to that of previous tests and averaged. The routine stops when the average network power converges to within a certain variance. The Monte-Carlo simulation method of [5] has demonstrated quite rapid empirical convergence properties. The advantage of these techniques is their simplicity and their ability to easily be modified to handle more complex gate models. However, the average network power is the sum of the averages for every node in the network. The convergence within suitable error for total power is therefore much more rapid than for similar error tolerances on the estimate for any *specific* node.

Exact symbolic simulation for power estimation is the technique used to establish the results of this dissertation. Although the size of networks which can be analyzed in this way is limited, this simulator is used because of its precision in being able to define the accuracy of the estimation strategies presented here. Probabilistic analysis estimators are not used as they already make heuristic correlation approximations. Even though the error for the overall network power may be small, the error for any specific node may be significantly higher. Statistical analysis estimators also do not have the accuracy at every network node required for the testing of the theory proposed in this dissertation. Furthermore, neither estimation approach provides exact functionality of dynamic activity, a property which is required to test the validity of estimation abstractions. (Refer to Chap. 5).

1.3 Low-Power Synthesis for Combinational Networks

The design flow for synthesis of combinational networks is shown in Fig. 1.6. The problem may be expressed as the mapping of a set of two-level Boolean functions (logical sum of products or product of sums) onto a restricted set of atomic logic functions, these

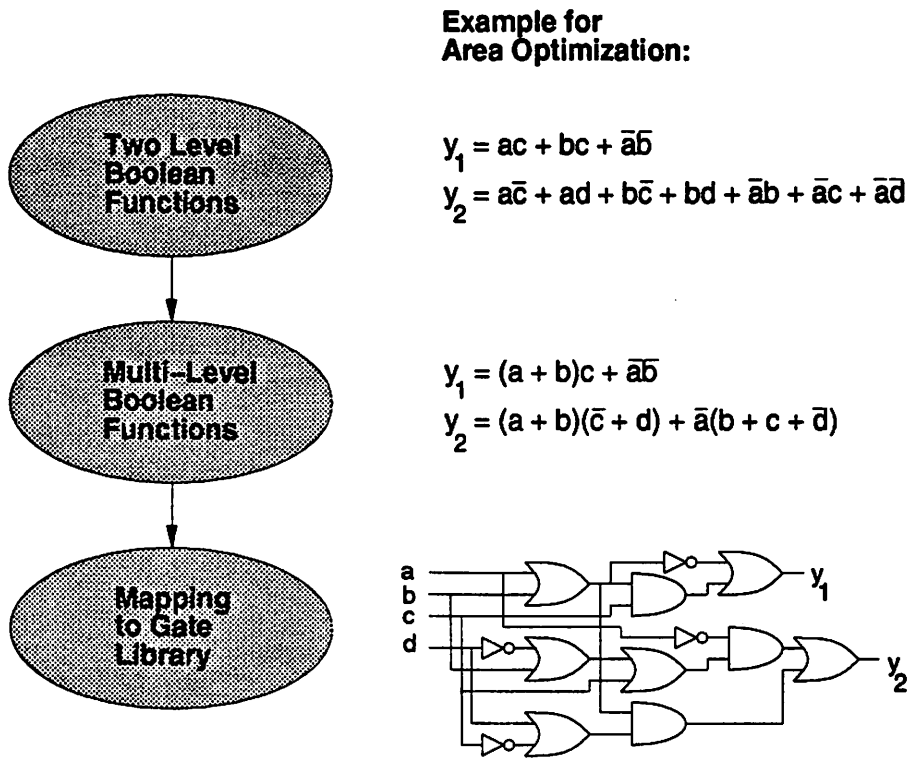

Figure 1.6: The Combination Synthesis Design Flow

functions represented by the gates within the technology library. The first step in the procedure is the extraction of a set of common subexpressions which minimizes a specific cost function. For example, the optimization for area attempts to minimize the literal count by maximizing logic sharing [3], the optimization for speed attempts to reduce logic depth and gate loading on the longest paths [21]. This step is followed by mapping onto the specific gate library. An example of this procedure for area optimization of a simple function mapped to a library of two-input gates is also presented in Fig. 1.6.

An important part of the synthesis process at all points in the design flow is the ability to simplify logic expressions. For example, the expression: $f = \bar{a}b\bar{c} + cd + bc\bar{d}$ may also be written: $f = \bar{a}b + bc + cd$. The simplification process can further be enhanced through use of Don't Care (DC) sets. Don't Care sets consist of parts of the functional space which can be manipulated without affecting circuit operation. There are two types used in combinational synthesis:

- *Satisfiability DC* (SDC). The set of input, or intermediate variable, combinations

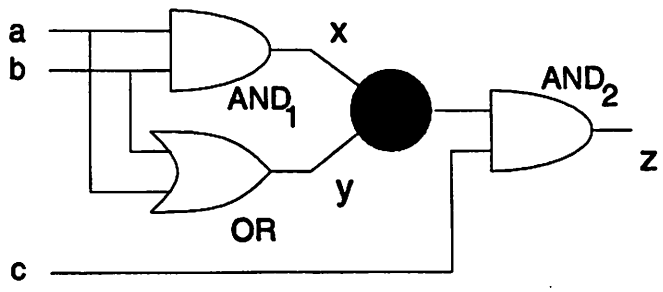

Figure 1.7: The Internal SDC and ODC Sets

which cannot functionally occur.

- *Observability DC* (ODC). The set of input, or intermediate variable, combinations which do no affect the circuit output functionality.

Both forms of DC sets may have components from internal and external contributions. The *external* DC set is formed from knowledge about the environment in which the circuit is embedded. At the input it is the set of all combinations of input logic states which will not occur, and at the output it is the combination of functions which are equivalent with respect to the environment. The *internal* DC set is a consequence of the specific circuit configuration. To explain the internal DC set further, consider the simple example of Fig. 1.7. The local inputs to the solid black node have functionality $x = ab$, $y = a + b$ implying that the condition $x\bar{y}$ cannot occur. This is the Satisfiability DC function. The circuit output z can only be influenced by the output of the solid node iff input c is high. This implies that the ODC set for this node is $\bar{c}$.

The non-satisfiability of certain local functionality implies that simplification of an expression which uses this does *not* affect the probability that the overall expression evaluates high. This permits modification of a region of a network which affects only the transition activity and capacitance (therefore power) *within* that region (modulo changes in delay). The ODC set, on the other hand, can also be used to improve the optimality local to a region of a network but this *can* affect onset probabilities throughout the transitive fanout of the region so an improvement in power locally does not guarantee an improvement overall. However, the ODC sets in most networks are typically very large and have been shown to provide significant synthesis freedom for improved optimality during synthesis for area [17] and speed [21]. It is therefore desirable to be able to make use of this in synthesis

for low-power.

The research results presented in this dissertation relate to the prediction of when a *locally* optimum decision translates to a *globally* optimum solution in synthesis for low-power. In particular, the problem of modifying functionality within the ODC set is addressed, as is the ability to predict the influence on network power associated with changes in delay. The work is presented in the framework of resynthesis, which is incremental synthesis applied to networks previously optimized for another constraint such as speed or area. It is restricted to this class of synthesis as significant information is available regarding the transitive fanin and fanout logic structure following the original synthesis pass. Network capacitances can then account for layout parasitics providing an accurate model for delay.

1.3.1 Previous Work in Synthesis for Low-Power

It is clear from Eqn. 1.1 that with a fixed power supply and clock frequency, a reduction of power is achieved through the reduction of network capacitance or switching activity. There are two forms of transition activity which can be affected during synthesis: *functional* and *spurious dynamic* activity. Functional activity is that which occurs even in an ideal circuit with zero delay, spurious dynamic activity is any other activity which is solely the product of gate delays in a real network. The work published to date has a focus on the reduction of capacitance or functional activity. No work specifically accounts for the influence synthesis techniques have upon spurious dynamic activity, as addressed in this dissertation.

The most straight forward approach to the reduction of power are those which do not perform functional manipulation. These include the technology mapping (e.g. [22], [24], [14]) and gate resizing [2] techniques. All of the technology mapping routines are dynamic programming optimization strategies for tree networks. At each branch in the tree, the optimal solution can be found using the optimal solutions for the leaves. This technique is generalized to circuits of arbitrary topology by heuristic partitioning of the network into trees. The optimality of these techniques depends upon the zero-delay assumption for the gates. The gate resizing algorithm finds fast paths in the network and down-sizes gates on these paths to reduce capacitance. As it is only on the paths which do not contribute to the latest output signal, this approach guarantees that network speed is not compromised. Similar to the mapping techniques, however, proposed gate resizing algorithms have not

taken into account the effect that a change in path delays can have upon transition activity throughout the network. In private communication with Iris Bahar, author of [2], this effect was claimed to account for several network optimizations which were less fruitful than expected.

There have been several attempts at logic restructuring for low-power. In general these algorithms are based upon very simple heuristics for estimating the power consumption within a logic block before that block is technology mapped. The reported reduction in power has been small in all cases, around 10 to 20%. In the work reported in [16], subexpression extraction is performed using the simple heuristic of maximizing logic sharing while minimizing the input loading for inputs with high switching activity. In [20], subexpressions are extracted according to their probability of evaluating high. The functional transition activity is given by: $2(1-p)p$ for a signal with probability p of evaluating high, so better candidate subexpressions have p far from 0.5. The ODC set is used to expand (contract) the logical function if it evaluates 1 (0) with a probability greater than 0.5. There is also a heuristic for the minimization of spurious dynamic activity based on collapsing sections of the network with short paths. These two-level functions when technology-mapped generally evaluate slower than multi-level representations. This tends to balance paths throughout the network. A recent publication [10] extracts subexpressions according to output activity of a subexpression but also according to the primary input variable support required to implement the function. The Don't Care set is used to find a support which has minimal switching activity. For example, consider the function $f = ab$ with ODC $a \oplus b$ where a has a higher switching activity than b. The function f may be expressed $f = a$ or $f = b$, so it is more optimal to choose $f = b$. They also address the problem of a local change in functionality affecting the functionality throughout the transitive fanout by restricting the ODC set construction. However, the ODC set is restricted in such a way that any local synthesis step is guaranteed to not increase the power at *any* point throughout the transitive fanout (Refer to Sec. 3.1). In fact, it is a general statement regarding all synthesis procedures presented in the literature that local optimization of functionality is made without considering the *amount* of influence upon power consumption throughout the transitive fanout circuitry. These logic restructuring techniques may therefore be assumed random changes within the ODC set relative to the structure of the transitive fanout. The theory and statistical evidence presented in this dissertation verifies that there exist simple estimation strategies which can account for this effect. This provides a technique for

improving the global optimality of existing synthesis routines.

1.4 Dissertation Outline

The concept of prediction of change in power expected in a resynthesis step is outlined in detail in Chap. 2. In particular, a estimation abstraction hierarchy is described. This is related to the form of information available for the most accurate possible prediction of global effects on power external to a resynthesis region. Chap. 3 contains an examination of estimating the effect of resynthesis on transitive fanout power in a zero-delay network. It is shown that the expected change in onset probability at the output of the resynthesis region is sufficient for construction of a highly accurate estimate of this effect. In Chap. 4, this material is generalized to networks with arbitrary delay elements. Here it is shown that it is not possible to use the bounds on transition arrival time to predict the sensitivity of transitive fanout power to changes in delay. However, the delay insensitive estimator which is developed has an accuracy similar to that of the estimation technique proposed for zero delay networks. The problem of estimating delay sensitivity is further examined in Chap. 5. It is shown that for combinational networks, delay sensitivity is generally a minor effect. Furthermore, it cannot be estimated without full simulation of the possible delay conditions. In Chap. 6 a summary of the dissertation is presented and an outline of the areas which appear to be the most profitable extensions for the formulation of a low-power synthesis strategy are described.

1.5 Definitions

Consider node n embedded in a digital circuit. The circuit has a set of primary inputs $I = \{i_1, i_2, ..., i_3\}$. Let $V(I)$ be the set of all possible pairs of input vectors. It is assumed that the circuit is allowed to settle completely after application of any input vector. Time $t = 0$ may now be set as the application time of the second vector in any input vector pair, v. Let P_v be the probability that vector pair $v \in V(I)$ occurs. Over the set, $V(I)$, the *earliest* arriving transition at node n occurs at time t_n^{min}, the latest at time t_n^{max}.

$T_n(t)$ denotes the functionality (subset of $V(I)$) for a transition to occur at the output of node n at time t. Let T_n^T be the function relating the number of transitions (**Total Transition Activity**) at the output of node n to space $V(I)$. For every $v \in V(I)$,

$T_n^T(v)$ at node n is the number of all $0 \rightarrow 1$, or $1 \rightarrow 0$, transitions in the interval $[t_n^{min}, t_n^{max}]$. The average total transition activity at n, $|T_n^T|$, is given by:

$$|T_n^T| = \sum_{v \in V(a)} P_v . T_n^T(v)$$

Definition 1.5.1 *The* **Functional Activity** *of node n is*

$$|T_n^F| = \sum_{v \in V(a)} P_v . |F_n^{0^-}(v) - F_n^{t_n^{max}+}(v)| \tag{1.2}$$

where: $F_n^t(v)$ is the logic value of node n at time t under input vector pair v.

This is equivalent to the activity at node n under the zero-delay model for all nodes.

Definition 1.5.2 *The* **Spurious Dynamic Activity** *of node n is*

$$|T_n^D| = |T_n^T| - |T_n^F| \tag{1.3}$$

The following naming conventions are used throughout the dissertation:

Definition 1.5.3 *$f_n(Z)$ is the static logic function at the output of node n in terms of variables in set Z.*

Definition 1.5.4 *$f_n(Z)|_z$ is the boolean co-factor of logic function $f_n(Z)$ with respect to $z \in B^{|Z|}$*

Definition 1.5.5 *$\Omega_Z(f)$ is the set of minterms of the function f contained within the space defined by the variables Z. A minterm is product term containing a literal from every input. A literal is the input variable (e.g. i_j) or its complement ($\bar{i_j}$).*

There are two forms of probability notation in common use here. Consider set $A \subset B$ where $|A|, |B|$ are the cardinalities of the respective sets. The following relative probabilities are defined:

Definition 1.5.6 $Pr(A|B) = \frac{Pr(A \cap B)}{Pr(B)}$

Definition 1.5.7 $CntPr(A|B) = \frac{Pr(|A \cap B|)}{Pr(|B|)}$

If all elements in set B have the same probability of occurrence, these are equivalent.

Chapter 2

Motivation

The ability to estimate power dissipation in a CMOS network depends upon the accuracy of modeling the functionality of the inputs to each node over all time. The complexity of the estimation increases with the accuracy of this information. There have been many techniques developed to explore this tradeoff including exact symbolic simulation [7], probability approximation heuristics ([15], [23]) and statistical Monte Carlo approaches [5]. Comparison of the accuracy/complexity of the methods suggests that significant reduction in error can be achieved over simplistic methods (such as assuming fully independent transition activity) with a small increase in complexity, but absolute accuracy comes at a high price as illustrated conceptually in Fig. 2.1. However, none of the techniques developed so

Figure 2.1: Accuracy/Complexity for Full Estimation of Activity

far provide a way of predicting circuit power when aspects of the activity are bounded but not known exactly. This is the problem associated with the prediction of overall change in network power which will result from a resynthesis operation. This global prediction is needed in the formulation of resynthesis decisions.

The minimization of error in the estimation techniques presented in the papers listed above has shown that simultaneous transition arrival and node input correlations are very important for accurate power estimation. During a resynthesis procedure, however, only minor changes are made to the original network. In many cases the correlations and simultaneous arrival properties of a node inputs are unlikely to change significantly. This fact can be exploited to allow information gained from an initial simulation to be used to improve the accuracy of simple estimation strategies applied during the resynthesis process. In fact, it is possible that when certain variables are not known exactly, prediction of a change in activity based upon a simple estimation technique may be *more* accurate than a complex strategy. This is a consequence of the addition of errors resulting from the uncorrelated approximation of many terms within a complex expression. On the other hand, too simplistic an estimation scheme will be poor for even small changes in activity, due to error inherent in the method. This suggests that under the conditions where certain information cannot be fully specified, the accuracy/complexity trade-off curve looks more like that in Fig. 2.2.

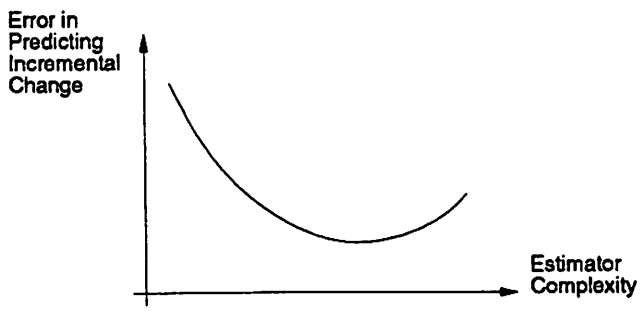

Figure 2.2: Accuracy/Complexity for Estimating Change in Activity

2.1 Activity: The Levels of Abstraction

The construction of estimation strategies based upon restricted knowledge about the change in the network requires that the levels of abstraction involved in estimation be defined. The complexity of an estimation strategy decreases with increased abstraction, as does the ability to achieve better than a certain accuracy of estimation. This provides a framework for assessing whether there exists an estimation strategy which will be able to predict global effects without first preforming a synthesis step.

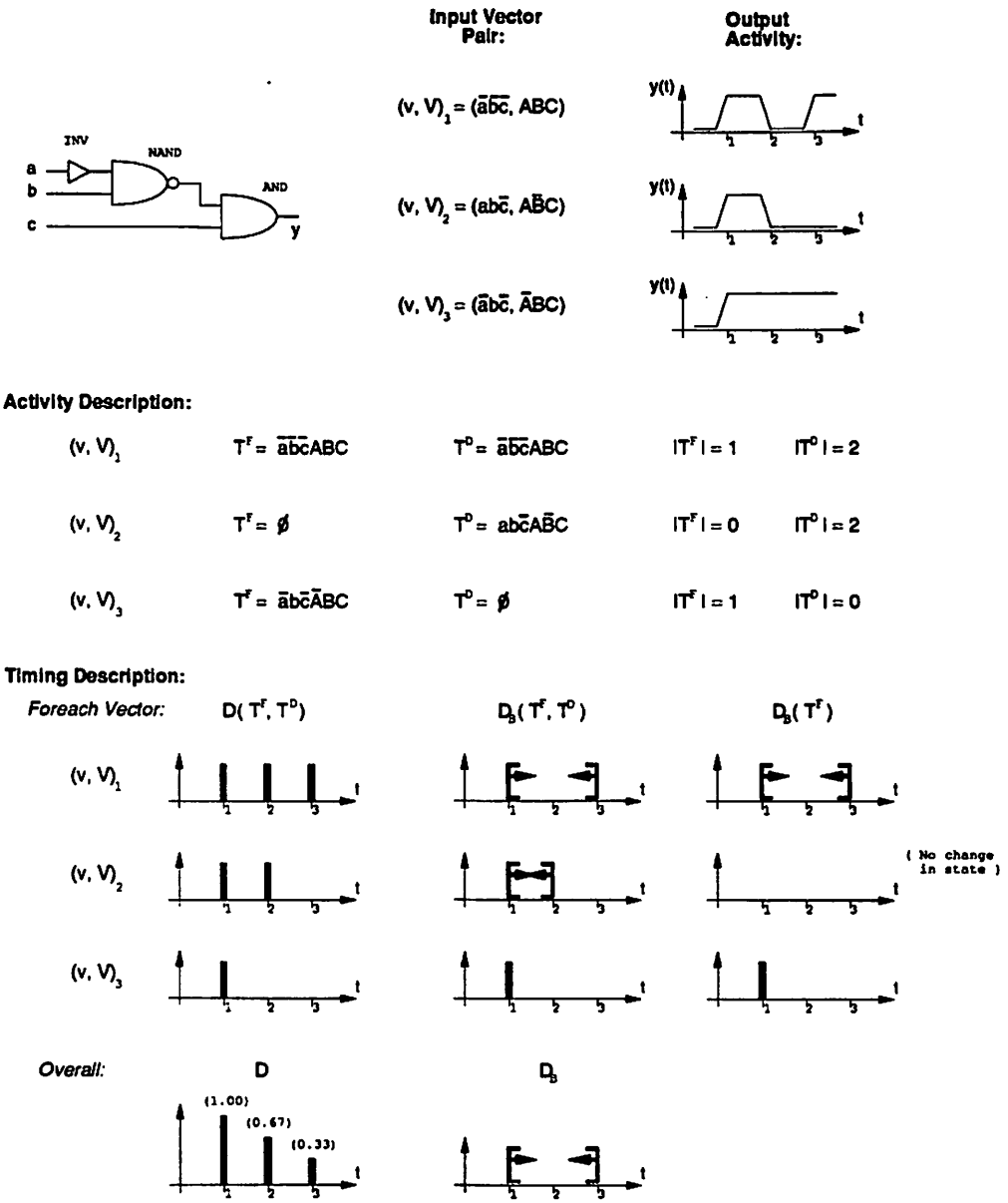

Figure 2.3: Abstracting Timing and Activity Information

An estimation strategy for activity is defined by the way in which it handles functionality of the transition activity, and the accuracy of the timing information associated with this. Both the functionality and timing abstractions have two variables requiring approximation. The functional abstraction is specified by the description of activity as the functionality of transitions T^F (T^D) or just the amount $|T^F|$ ($|T^D|$). Tim-

ing is specified by a distribution of transition timing for each input vector pair, one of $\{D(T^F,T^D), D_B(T^F,T^D), D_B(T^F)\}$, and overall switching distributions, one of $\{D, D_B\}$. The notation here is defined: $D(x)$ the switching distribution for each vector pair in which condition x occurs; D the overall switching distribution describing the probability of transition activity at any time over all input vector pairs; $D_B(x)$ the timing bounds on transition activity for each vector pair in which condition x occurs; and D_B the bounds on the overall switching distribution. So, $D_B(T^F,T^D)$ is a timing abstraction which corresponds to being able to bound activity under any input vector pair application in which spurious dynamic or functional activity occurs. If this bound is to be tight enough to be useful, it implies knowledge of the functionality of the spurious activity. $D_B(T^F)$, on the other hand, only implies knowledge of the earliest and latest transition times corresponding to a change in state. An example of these concepts is shown in Fig. 2.3 for describing the activity at the output of the AND gate for the three example input vector pairs listed in the figure. For the purposes of illustration, the overall distribution, D, shown in the example is only over the three vector pairs described.

Fig. 2.4 depicts the general abstraction levels for activity estimation. An increase in the functional abstraction reduces knowledge of the association between activity and input vector pairs. As level of abstraction is increased for timing, less information is maintained for each vector pair. Moving from τ_{1_A} to τ_{4_B}, loss of timing information is: $(\tau_{1_A} \rightarrow \tau_{1_B})$ exact timing for each input vector pair; $(\tau_{1_B} \rightarrow \tau_2)$ bounds on timing for any input vector pair; $(\tau_2 \rightarrow \tau_3)$ bounds on timing for any input vector pair resulting in a functional change; $(\tau_3 \rightarrow \tau_{4_A})$ specific times for any transition activity; and $(\tau_{4_A} \rightarrow \tau_{4_B})$ bounds on any transition activity. The functionality and timing information abstraction describe any form of activity estimation scheme. For example, an estimation scheme based purely upon the *amount* of activity at the inputs to a node (therefore neglecting correlation) and an approximation of all transition times is at activity abstraction level A_3, timing abstraction level τ_3 (e.g. [15]). Estimation schemes which associate time and functionality at every time point for any input vector are at level A_1, τ_{1_A} (e.g. [7]).

Notice that there are four shaded regions in Fig. 2.4. These regions dictate what are viable combinations of activity and timing abstractions. Any timing abstraction can be combined with an activity abstraction from the same region or below (lower abstraction). Any activity abstraction can be combined with a timing abstraction from the same region or above. The converse situations do not hold. For example, A_3 cannot be combined

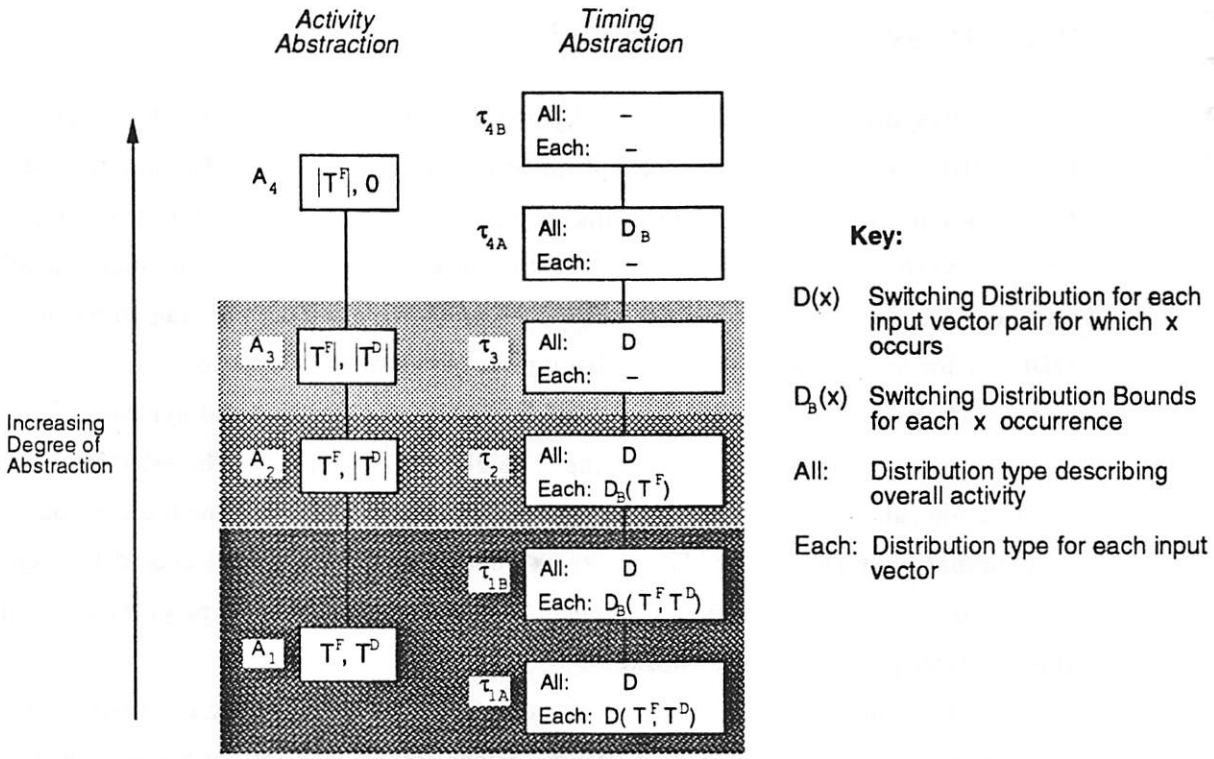

Figure 2.4: Abstracting the Concept of Activity Estimation

with timing abstraction τ_2 as having the distribution bound estimates for each functional transition implies that the functional transition equations must be known.

Although accuracy of an overall estimation scheme will increase with decreasing abstraction, this may not be true for estimation schemes involving small changes relative to an initial network simulation as depicted in Fig. 2.2. To determine the applicability of a certain level of abstraction for estimating global effects in resynthesis, it is necessary to test several abstraction levels and determine a vector of improvement. When the error increases with change in abstraction level, no further estimation strategies need to be examined. (This assumes that the second derivative of the error/complexity tradeoff curve has at most one zero crossing.) The following section outlines how the levels of abstraction are related to the prediction of the global effects of resynthesis on power dissipation, and how the testing was made complete for the work in this dissertation.

2.2 Resynthesis for Low-Power

Resynthesis procedures generally use compatible subsets of the Observability Don't Care (ODC) set [18] during global optimization. The ODC allows the function of a node to be altered without the change being observable at the output, while the compatibility permits synthesis of separate sections of the network without one modification affecting another. Any section of the network can then be changed without having to recompute the ODC set for any nodes in the transitive fanout of the modified section.

In a resynthesis procedure, the goal is to provide maximal synthesis freedom to a restricted set of nodes which are judged 'highly non-optimal'. The selection of a set of target nodes allows the definition of a node input ordering for optimizing the construction of a compatible ODC subset. For the target nodes, this ODC subset should be larger than that used in the original global synthesis pass, thereby increasing resynthesis freedom at these critical points in the network.

In general, resynthesis of a network region is preceded by collapsing the output nodes of the region back into a two-level functional representation for a subset of the transitive fanin nodes. A multi-level synthesis routine is then applied to this two-level subnetwork. Although there may be multiple outputs to a region, this just a generalization of the single output case for which the resynthesis region is a *'cone'* of fanin logic with a single fanout point at its apex. This restricted case is directly analyzed in this dissertation. However, all the approximation techniques are extended to handle simultaneous changes in activity to multiple inputs of a node. This provides a technique for the generalization of the single output theory.

The selection of an appropriate region for resynthesis must consider four effects. These effects are listed below, and pictorially represented in terms of their regions of influence in Fig. 2.5.

1. *Cone Power Change:* This is the difference in power between the new resynthesized region of network, and the nodes which it replaces.

2. *Functional Transitive Fanout (TFO) Effect:* Functional activity is the activity which occurs even in a zero-delay network model. A change in functionality of a node in the network alters the onset probability, thereby changing the probability of switching and power consumption. If the Observability DC set is used in resynthesis, the func-

tionality of the nodes in the transitive fanout of the cone fanout may change. This change may increase circuit power even if the Cone Power Change is negative.

3. *Spurious Dynamic Activity TFO Effect:* Spurious Dynamic Activity is any activity which is not Functional Activity. Any change in activity or delay at the cone fanout may change the spurious dynamic activity throughout the cone's TFO. The change in node sensitivities due to the Functional TFO Effect also influences the Spurious Dynamic Activity TFO Effect.

4. *Delay Affected Spurious Dynamic Activity:* Resynthesis may alter the capacitance at the inputs to the cone. This change in loading of the nodes driving the cone inputs affects network delays. This can alter spurious dynamic activity in circuitry other than just the TFO of the cone.

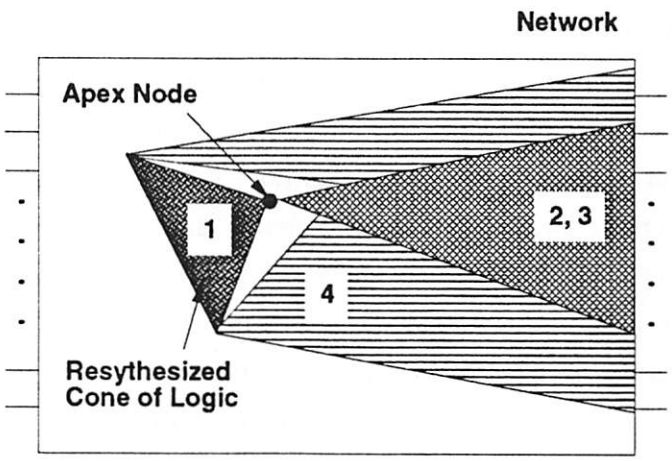

Figure 2.5: Resynthesis Effects

The resynthesis of a region within a network is constrained by the ODC set and the maximum delay allowed for the network. In more general terms, it can be stated that any resynthesis algorithm must have control over the possible change in functionality, the path lengths within the resynthesized region and the loading effect of the region on its inputs. None of these properties are fully predictable prior to application of resynthesis as any such optimization routine is a heuristic approach to an NP-complete problem. This makes it infeasible to find the *optimum* change in functionality, path lengths or loading effects. Improved optimality in resynthesis is dependent upon the assignment of costs to design choices which it can control. If these costs can be assigned without restricting

the design choices to a set for which the global effect can be estimated, then this global information is very likely to improve the optimality of resynthesis. On the other hand, a restriction implied by global constraints can significantly reduce local resynthesis options. The difficulty in predicting the severity of this effect implies that the global restrictions may worsen the final result rather than improve it.

The material in this dissertation does not address specific optimization of the resynthesis region, the Cone Power Change. Consequently, complete resynthesis results for the optimization of power are not presented. The work completed is a thorough analysis of the ability to predict the other three effects of local resynthesis upon global circuit power, regardless of the resynthesis algorithm used. This answers the question:

What information is required to estimate certain global power properties?

The final change in functionality at the output of the resynthesis region can be influenced by any design choice during the resynthesis process. Computation of the exact Functional TFO Effect for every possible design choice at any point in this procedure is not a useful approach. Beyond issues of computational complexity, optimum decisions made at every point of the resynthesis process does not even guarantee an improved solution due to the heuristic nature of the cost function used for optimality detection. The measure of the Functional TFO Effect which can be used to guide the resynthesis process needs to use the *magnitude* of a change of functionality. To make the technique more specific, the magnitude of the Functional TFO Effect within a class of functions which partition the space can be computed. For example, the expected Functional TFO Effect can be computed for conditions of a functionality change within the space i_k, or space $\overline{i_k}$, where i_k a network input. An estimation scheme based upon the magnitude of the change in functionality corresponds to the activity abstraction A_4, τ_{4_B}, and this is the material of Chap. 3. The results of the estimation scheme outlined there correlate to measured changes in power extremely well, thereby demonstrating that the Functional TFO Effect can be used to optimize resynthesis decisions without functional restriction.

Prediction of the effect of resynthesis on spurious dynamic activity at the output of the resynthesis region is more complex than prediction of the change in functionality. Estimation of spurious dynamic activity requires knowledge of both timing and functionality at every node in the resynthesis region. The lowest (most accurate) level of functional abstraction which can be used to estimate the expected Spurious Dynamic Activity TFO

Effect without restricting the design choices is therefore that which uses magnitudes of functional and spurious dynamic activity, A_3. The formulation of estimation strategies which are maximally accurate for this level of functional abstraction, as depicted in Fig. 2.6 by the heavily outlined boxes, is the material of Chap. 4. It is shown that there is an *increase* in the error with decreasing timing abstraction due to the complexity of the estimation required. This is shown by the construction and testing of estimators for timing abstractions τ_{4_A} and τ_{4_B}.

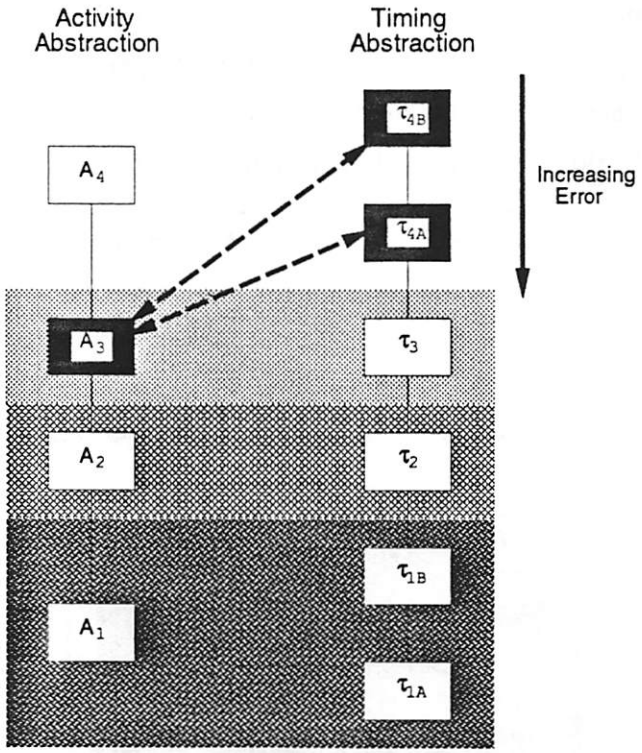

Figure 2.6: Spurious Dynamic Activity TFO Effect Abstractions

The contents of Chap. 5 concerns prediction of the Delay Affected Spurious Dynamic Activity. As changes in network activity outside the resynthesis regions themselves will be small, it can be assumed that the activity at the input to any region can be estimated with a high degree of accuracy. That is, the functionality of the transitions over all time should be predictable. The estimation of Delay Affected Spurious Dynamic Activity is consequently at the functional abstraction level A_1. The local effect of the loading changes at the resynthesis region inputs is the problem of estimating spurious dynamic activity at the output of a node when input delays are changed. An optimal input loading

for a resynthesis region is one which affects delays in such a way as to maintain, or reduce, power in the parts of the network which are influenced. It is shown that even when the the functionality of the dynamic activity is known exactly, the optimality attainable cannot be adequately predicted for timing abstractions beyond τ_3, and that any timing abstraction below τ_2 requires a computational complexity equivalent to full simulation of all possible delay configurations (Fig. 2.7). A statistical examination of the significance of delay dependence shows that delay sensitivity will not be a dominant effect during resynthesis. Furthermore, a better delay-dependent estimator for the Spurious Dynamic Activity TFO Effect than that of Chap. 4 requires restricting synthesis to permit estimation at levels of functional abstraction better than A_3. Sensitivity to delay of the output of a region is therefore not important in the formulation of an adequate cost function for low-power resynthesis.

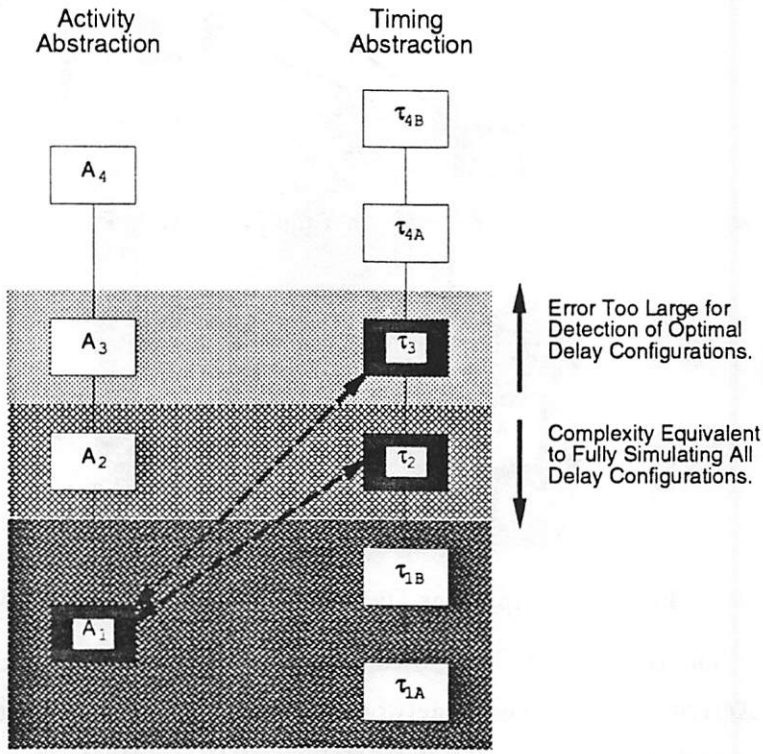

Figure 2.7: Delay Affected Spurious Dynamic Activity Abstractions

The estimation techniques presented in this dissertation can be used directly to further optimize any existing resynthesis routine as they do not restrict the design space. Alternatively, they can be used to guide the development of resynthesis algorithms based upon properties shown to be statistically significant and predictable.

Chapter 3

Functional Activity Estimation: The Zero Delay Model

Resynthesis of a region to reduce power locally may influence the output functionality of that region. Such a change in functionality may alter the power dissipation throughout the transitive fanout. The work described in this chapter involves analysis of this specific issue.

The development and empirical test of a simple probabilistic model which can be used to predict expected change in TFO power are presented. The necessity for such an estimator is motivated in Sec. 3.1, and Sec. 3.2 contains an the theoretical development and justification of the basic estimation strategy. The applicability of a probabilistic averaging technique for this problem is the basis of Sec. 3.3, and 3.4 are the results of empirical testing. For the basic estimation scheme, using simple approximations for the input probabilities and ODC set, a data correlation of 0.99 between theory and experiment illustrates the usefulness of the method. In the latter part of this chapter, Sec. 3.5 and Sec. 3.6, the two simplifying approximations are removed from the theory and a tradeoff between method accuracy and complexity presented. This extends the work first presented by this author in [13]. A technique for utilizing this method to optimize ODC construction is also covered here.

3.1 The Effect of Functionality Change on TFO Power

To motivate the need for a study into the effect of functionality change on TFO power, it is first necessarily to illustrate the mechanism by which a decision based upon local optimality considerations may be bad in a global sense.

The cost function relating power to onset probabilities is not linear. In the case of a circuit with zero-delay elements, it follows from Eqn. 1.1 that the power consumption at a node a is proportional to $Pr(a).Pr(\bar{a}) = Pr(a).(1 - Pr(a))$ where $Pr(a)$ is the probability of node a being 1. The trivial example of Fig. 3.1 demonstrates a case in which a local improvement results in an overall increase in power for the circuit.

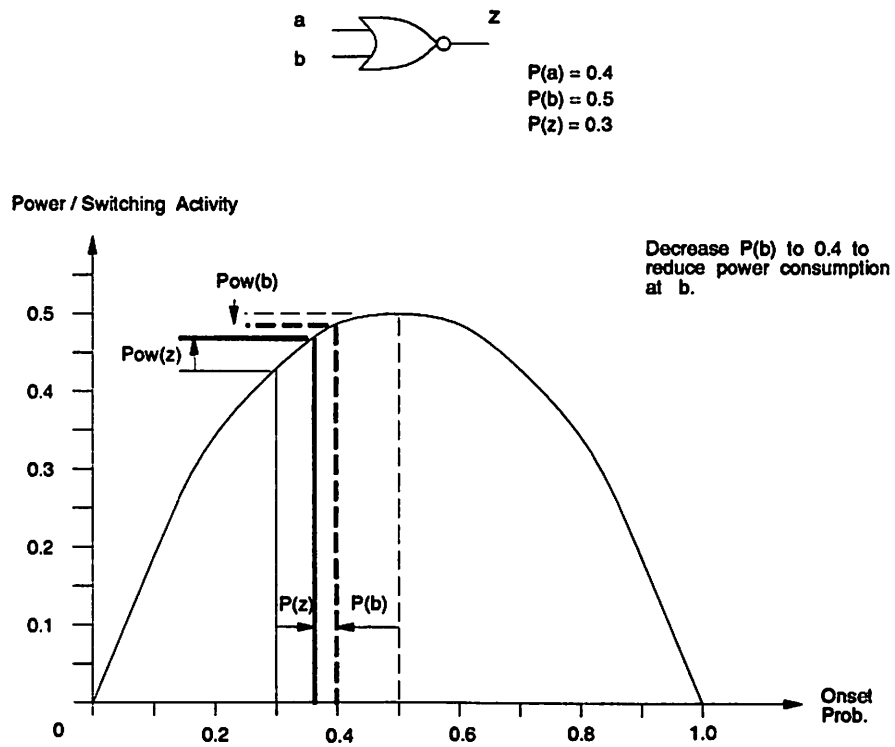

Figure 3.1: A Decrease In Input Power Offset By An Increase In TFO Power

In this example, the NOR gate inputs are independent and the capacitance at each node is assumed equivalent. The power consumed at input b is the highest power consumption at any of the three nodes. Consider the case in which this element is embedded in a larger circuit. Assume that the ODC set associated with node b allows the probability of b being one to be reduced to 0.4. This reduces the power consumption local to b.

However, as a consequence the probability of z being one increases from 0.3 to 0.36. The corresponding increase in power dissipation at the output z more than offsets the reduction in power dissipation at input b.

A small change in the on-set probability of a node with high functional activity is more likely to result in a change in the transitive fanout power which offsets the local optimization. A node with smaller functional switching activity is less likely to have as much DC set freedom for a locally beneficial resynthesis step. However, any change in functional power which can be made has a lower probability of being overwhelmed by detrimental effects in the transitive fanout. This introduces a key insight, essential to the generation of a resynthesis routine targeting low-power. *Power consumption at nodes with the highest functional switching probability is the least sensitive to small changes in onset size.*

In this context, choosing a node for resynthesis becomes far less obvious than it might initially appear. In some cases, nodes with low switching probability may be more favorable choices in a resynthesis algorithm than those of high probability. This realization motivates a need for a technique to estimate the change in power of the transitive fanout of a node due to local changes in node function.

It is possible to restrict the construction of the ODC in such a way as to guarantee that a local resynthesis step does not detrimentally influence the TFO [10]. However, this may not not desirable. For example, consider the situation depicted in Fig. 3.2. The node m has been selected for resynthesis. This node is the input to nodes p and q within a network. Nodes p and q are both AND gates so that an addition to the the onset of input m in resynthesis will increase, or not change, the onset probability of these gate outputs. From fanout loading effects, node q effectively drives a much larger capacitance than node p. The critical functions in this analysis are shown by the shaded regions within the rectangles representing the Boolean space, Ω_I, in Fig. 3.2.

The functional switching probability of p is very low as the onset probability $P_p \ll 0.5$. Nodes m and q have high switching probability as P_m and P_q are marginally greater than 0.5. Assume that the ODC set for node m is entirely contained within the offset of the original function. To reduce the switching at node m, the onset probability of node m must be increased. A change in functionality of node q as a consequence of this will also result in further power reduction, so this node does not restrict good functional choices at m. On the other hand, an addition to f_m will *increase* power at node p except in the case where it does not affect the functionality at p. To ensure that *no* TFO node is

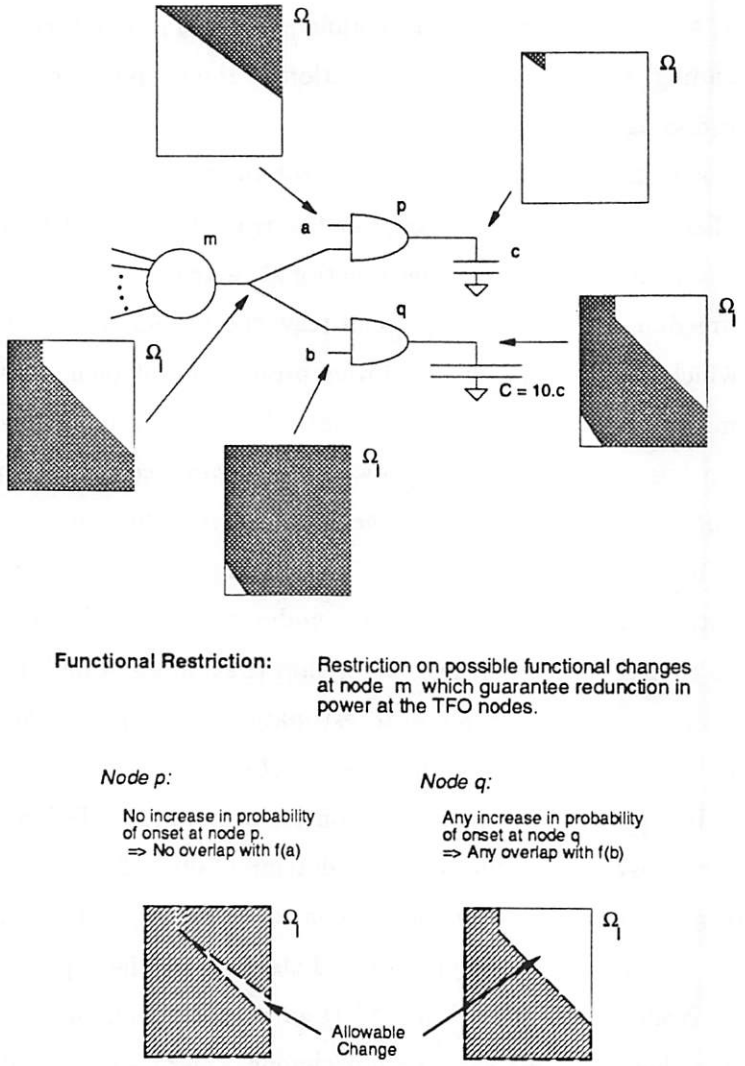

Figure 3.2: Severe Over-Restriction of the DC Space

influenced such that the power local to that node increases, the set of viable functions has to be the intersection of those viable for all TFO nodes. The possible change in functionality is limited by the most severely restricted node p even though the reduction in power at the high-load node q would dominate this effect.

More generally, it may be stated that the problem with a functional restriction technique such as that of Iman et. al. [10] is that it does not take into account the relative changes in power throughout the TFO, so cannot predict the *likelihood* that a local change in functionality is globally beneficial, even if that change does not influence *all* TFO nodes in a beneficial way. Even though the example of Fig. 3.2 is somewhat contrived, the results

of this chapter show that there are a limited number of conditions under which a functional restriction is necessary. The estimation technique developed here is useful in identifying these occurrences.

3.2 Estimating the Expected Change in TFO Activity

A node n is a good candidate for resynthesis if a local change in activity, plus the change in activity throughout the transitive fanout of the node, reduces the overall power consumption. A functional activity change is achieved by altering the function of n locally within the bounds of the observability DC set.

It is not possible to pre-determine how a node will be resynthesized as local delay and area considerations influence this step. Furthermore, the functionality of the TFO nodes does *not* influence a local decision. It may be assumed that any change within the bounds of the ODC is random relative to the TFO. A node is defined to be a good candidate in a *statistical* sense if there is a high expectation of a significant decrease in total power resulting from a local change in functional activity.

A method for determining the expected change in activity throughout the transitive fanout of n is outlined in the following section. This technique will be shown to predict an exact average for circuits without reconvergent fanout. For mathematical simplicity, all inputs are assumed to have $Pr(1) = 0.5$ which implies that all minterms in the input space have the same probability. Furthermore, the analysis presented in this section assumes the entire Boolean input space, Ω_I, as a bound on the ODC set at a node. Extension of the theory beyond these assumptions is the subject of Sec. 3.5 and Sec. 3.6.

3.2.1 The Single Fanin Change

Consider the circuit element of Fig. 3.3. The diagonally shaded regions indicate the original function onsets. The set of nodes is embedded in a circuit with a tree graph structure. Assume that node n_1 is changed in a resynthesis step such that a set of minterms, A_{n_1}, is added to the original onset of the node. A_{n_1} is disjoint from the original onset of n_1. Assume that the size of A_{n_1} is known, but that the exact elements which compose the set are not.

Define the set of inputs to n, $\Lambda_n = \{n_1, n_2, ..., n_j\}$.

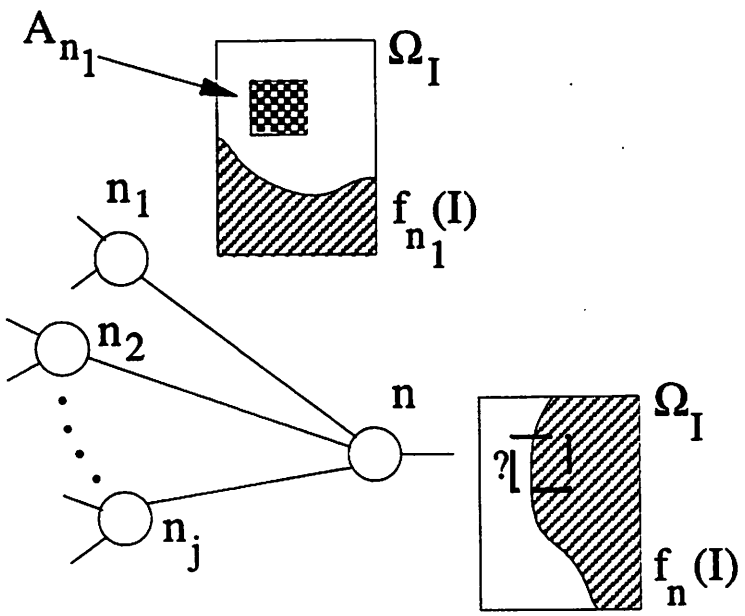

Figure 3.3: Single Input Change

Elements within the set of A_{n_1} may be added to, removed from, or not propagated through to the onset $f_n(I)$ of the node n.

Lemma 3.2.1 *The set of elements added to $f_n(I)$ due to the addition of a set of elements A_{n_1} to $f_{n_1}(I)$ is determined by the relation:*

$$\begin{aligned} A_n(A_{n_1}) &= \Omega_I(S_n^p(n_1)) \cap A_{n_1} \\ &\equiv \Omega_I(f_n(\Lambda_n)|_{n_1} \cdot \overline{f_n(\Lambda_n)|_{\overline{n_1}}}) \cap A_{n_1} \end{aligned}$$

$S_n^p(n_1)$ is the positive-phase sensitivity at n with respect to n_1.

Proof: For simplicity, consider a function f with local input a. We may express f as the logical sum of its algebraic co-factors with respect to a; $f_a, f_{\overline{a}}$ and remainder f_a^R.

$$f = f_a.a + f_{\overline{a}}.\overline{a} + f_a^R$$

An minterm x added to the onset of a is added to f if $x \in \Omega_I(\overline{f_a^R})$, $x \in \Omega_I(f_a)$ and $x \in \Omega_I(\overline{f_{\overline{a}}})$

$$\begin{aligned} \text{i.e. } x &\in \Omega_I(f_a.\overline{f_{\overline{a}}}.\overline{f_a^R}) \\ &\equiv \Omega_I(f|_a.\overline{f|_{\overline{a}}}) \end{aligned}$$

Hence, for a set of minterms A_a added to a, the set added to the onset of f is $\Omega_I(f|_a.\overline{f|_{\overline{a}}}) \cap A_a$. □

Similarly:

Lemma 3.2.2 *The set of elements removed from $f_n(I)$ due to the addition of a set of elements A_{n_1} to $f_{n_1}(I)$ is determined by the relation:*

$$\begin{aligned} R_n(A_{n_1}) &= \Omega_I(S_n^n(n_1)) \cap A_{n_1} \\ &\equiv \Omega_I(\overline{f_n(\Lambda_n)|_{n_1}}.f_n(\Lambda_n)|_{\overline{n_1}}) \cap A_{n_1} \end{aligned}$$

$S_n^n(n_1)$ is the negative-phase sensitivity at n with respect to n_1.

$S_n^p(n_1)$ and $S_n^n(n_1)$ are disjoint. The standard definition of node sensitivity is derived from the union of these functions:

$$S_n(n_1) = S_n^n(n_1) + S_n^p(n_1)$$

To address the problem of determining the size of an *expected* change in $f_n(I)$, the following proposition is required:

Proposition 3.2.1 *Consider a set of points C which contains a subset B. Consider the random selection of $|A|$ points from C. If any point in C is chosen with equal probability, the average number of chosen points which are also elements of B is $|A|.|B|/|C|$*

The proof of this proposition follows from basic probability theory.

The expected change in $f_n(I)$ can be computed by considering A_{n_1} as the random choice $|A_{n_1}|$ points within the set $\Omega_I(\overline{f_{n_1}})$. By definition: $\Omega_I(S_n^p(n_1).\overline{f_{n_1}}) \cap \Omega_I(f_n) = \emptyset$. Fig. 3.4 depicts the situation, where functions $f_n(I)$ and $f_{n_1}(I)$ are the same as that shown in Fig. 3.3. The dark solid line depicts the set within which the probabilities are computed. The expected size of the solid black set A_n is computed from Prop. 3.2.1 and Lemma. 3.2.1 as:

$$\begin{aligned} E(|A_n(A_{n_1})|) &= CntPr(S_n^p(n_1) \,|\, \overline{f_{n_1}}).|A_{n_1}| \\ &\equiv |A_{n_1}|.\frac{|\Omega_I(S_n^p(n_1).\overline{f_{n_1}})|}{|\Omega_I(\overline{f_{n_1}})|} \end{aligned}$$

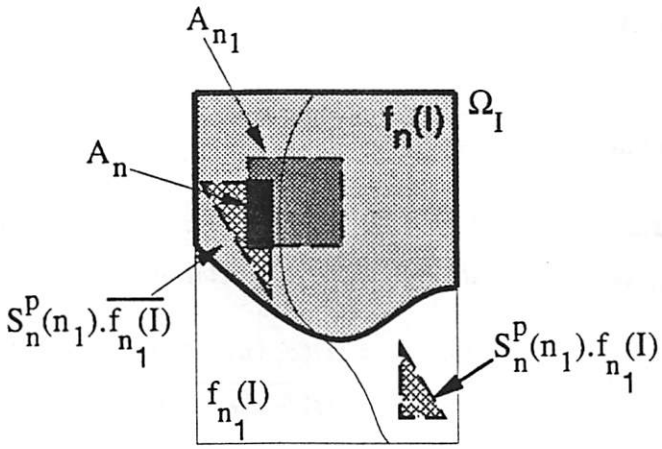

Figure 3.4: Overlap with the Sensitivity Set

Similarly:

$$E(|R_n(A_{n_1})|) = CntPr(S_n^n(n_1) \mid \overline{f_{n_1}}).|A_{n_1}| \tag{3.1}$$

The effect of a decrease in the onset size of node n_1 may also be computed. The relevant lemmas are:

Lemma 3.2.3 *The set of elements added to $f_n(I)$ due to the removal of of a set of elements R_{n_1} from $f_{n_1}(I)$ is determined by the relation:*

$$A_n(R_{n_1}) = \Omega_I(S_n^n(n_1)) \cap R_{n_1} \tag{3.2}$$

Lemma 3.2.4 *The set of elements removed from $f_n(I)$ due to the removal of a set of elements R_{n_1} from $f_{n_1}(I)$ is determined by the relation:*

$$R_n(R_{n_1}) = \Omega_I(S_n^p(n_1)) \cap R_{n_1} \tag{3.3}$$

It follows that:

$$E(|R_n(R_{n_1})|) = CntPr(S_n^p(n_1) \mid f_{n_1}).|R_{n_1}| \tag{3.4}$$

$$E(|A_n(R_{n_1})|) = CntPr(S_n^n(n_1) \mid f_{n_1}).|R_{n_1}| \tag{3.5}$$

Sets A_{n_1} and R_{n_1} are disjoint by definition. Consequently, for the case in which there is both a set of points added to and removed from f_n, the expectations may be summed.

$$E(|A_n|) = CntPr(S_n^p(n_1) \mid \overline{f_{n_1}}).|A_{n_1}| + CntPr(S_n^n(n_1) \mid f_{n_1}).|R_{n_1}| \tag{3.6}$$

$$E(|R_n|) = CntPr(S_n^n(n_1) \mid \overline{f_{n_1}}).|A_{n_1}| + CntPr(S_n^p(n_1) \mid f_{n_1}).|R_{n_1}| \tag{3.7}$$

These relations may be applied recursively throughout the transitive fanout of the original changed node, n_1.

Claim 3.2.1 *Consider a circuit with a tree graph structure. Assume that an arbitrary set of minterms X (Y) is added to (removed from) node a, and that the size of X (Y) is fixed. The technique outlined above correctly predicts the expected change in the onset size throughout the transitive fanout of a.*

Proof: By definition of the structure of a tree graph, no node in the circuit has a in more than one transitive fanin path. The claim then follows from the fact that the sum of the averages is equivalent to the average of the sum. □

To translate this result to an expected change in power, the following assumption is made:

Assumption 3.2.1 *Consider a node a with a transitive fanout node b. Let m be the number of possible ways of adding(removing) a set of X (Y) minterms relative to the original onset of a. Let* $\Delta = \{\delta_1(b), \delta_2(b), ..., \delta_m(b)\}$ *be the set of changes in the size of the onset of node b for the m possible changes at a. Assume that the standard deviation of the elements in* Δ *is sufficiently small such that the following approximation holds:*

$$E((Pr(f_b) + Pr(\delta_b))^2) \approx (Pr(f_b) + E(Pr(\delta_b)))^2 \tag{3.8}$$

The functional activity power consumption of node n obeys the proportionality relation:

$$P(n) \propto Pr(f_n).(1 - Pr(f_n))$$

The assumption allows the following approximation to be made:

$$\Delta(P(n)) \propto (E(Pr(\delta_n)) - 2.Pr(f_n).E(Pr(\delta_n)) - (E(Pr(\delta_n)))^2) \tag{3.9}$$

These local changes in power may be summed over the transitive fanout to determine the total expected change in power. A *functional power sensitivity* can be formulated by linearizing the expression prior to summation.

3.2.2 The Multiple Fanin Change

In a general circuit structure, it is necessary to examine the effect of changes to multiple inputs to a single node. Even in the case of resynthesizing a single node in the network, this situation may arise as a consequence of reconvergent fanout. Furthermore, this is a necessary extension of the theory to account for resynthesis of a region with multiple outputs.

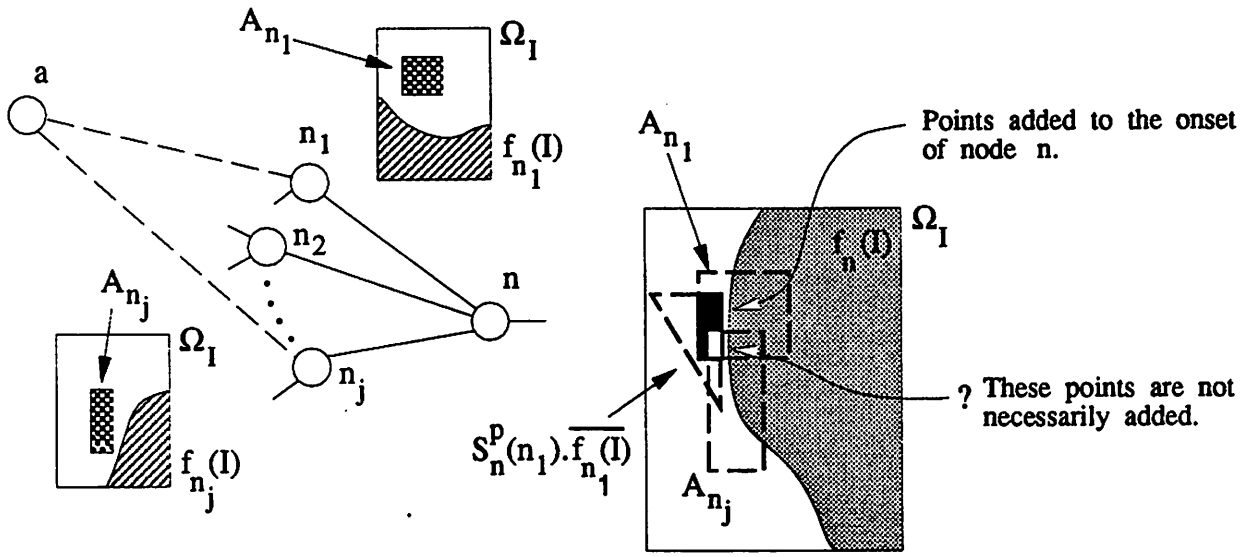

Figure 3.5: Multiple Input Change

Consider the example depicted in Fig. 3.5. Node n is in the transitive fanout of a. Node a is altered in a resynthesis step. In this case, inputs 1 and j are changed, their onsets being increased by sets A_{n_1} and A_{n_j} respectively. Consider just the elements in A_{n_1} which will also be added to $\Omega_I(f_n)$. $x \in A_{n_1} \cap \overline{A_{n_j}} \cap \Omega_I(S_n^p(n_1))$ will be added, but

$x \in A_{n_1} \cap A_{n_j} \cap \Omega_I(S_n^p(n_1))$ may not be. Sensitivities for multiple input changes can be calculated, but these computations are exponential in the number of fanins to the node. This motivates the use of an approximation scheme, such as the following:

Approximation 3.2.1 *Half the points within the intersection of set A_{n_j}, A_{n_1} and $\Omega_I(S_n^p(n_1)) \cap \overline{\Omega_I(f_n)}$) are assumed to be added to the onset of f_n.*

This is extended to handle more than two input changes in a straight forward manner. For example, if two inputs, n_i, n_j, along with n_1 are altered, the points added to the sensitivity set are half those in the union of those A_{n_i} and A_{n_j} which also lie in the set $\Omega_I(S_n^p(n_1)) \cap \overline{\Omega_I(f_n)}$). So the increase in the probability of the sensitivity to input n_i due to A_{n_i}, A_{n_j} is estimated as:

$$(1 - CntPr(S_n^p(n_i) \mid \overline{f_{n_1}})).(CntPr(A_{n_i}) + CntPr(A_{n_j}) - \frac{1}{2}.CntPr(A_{n_i}).CntPr(A_{n_j}))$$

To obtain the general expressions, the change in the sensitivity set due to *reductions* in the onset also needs to be considered. The application of the same assumption for R_{n_j} as for A_{n_j} implies that the two changes can just be added and handled as a *total* change. For example, the expression above for increase in the sensitivity set when inputs n_i, n_j are arbitrarily effected becomes:

$$\begin{aligned}(1 - CntPr(S_n^p(n_i) \mid \overline{f_{n_1}})).\quad & (CntPr(A_{n_i}) + CntPr(R_{n_i}) + CntPr(A_{n_j}) + CntPr(R_{n_j}) \\ & -\tfrac{1}{2}.(CntPr(A_{n_i}) + CntPr(R_{n_i})).(CntPr(A_{n_j}) + CntPr(R_{n_j})))\end{aligned}$$

Similarly, for those inputs changing, the reduction in the sensitivity set is:

$$\begin{aligned}CntPr(S_n^p(n_i) \mid \overline{f_{n_1}}).\quad & (CntPr(A_{n_i}) + CntPr(R_{n_i}) + CntPr(A_{n_j}) + CntPr(R_{n_j}) \\ & -\tfrac{1}{2}.(CntPr(A_{n_i}) + CntPr(R_{n_i})).(CntPr(A_{n_j}) + CntPr(R_{n_j})))\end{aligned}$$

These two terms can be added to determine the total change in probability of the sensitivity set. To formulate the arbitrary case, let $B(i)$ be the binary representation of $i \in J^+$, $B_j(i)$ the j^{th} digit. Let $\beta^j(i)$ be $B(i)$ with 0 placed in the j^{th} digit, all original binary values in positions $\geq j$ in $B(i)$ right shifted by one. eg. If $B(i) = 101101$, then $\beta^3(i) = 1010101$. Let $|B^1(i)|$ be the number of 1's in $B(i)$.

$$E(|A_n(A_{n_i})|) = |A_{n_i}|.\ (CntPr(S_n^p(n_i)\,|\,\overline{f_{n_1}}) \tag{3.10}$$
$$+\quad (1 - 2.CntPr(S_n^p(n_i)\,|\,\overline{f_{n_1}}))$$
$$.(\sum_{j=1}^{2^{|\Lambda_n|-1}-1} (\frac{-1}{2})^{|B^1(j)|}.\prod_{n_k \in \Lambda_n \setminus n_i} (CntPr(A_{n_k}) + CntPr(R_{n_k}))^{\beta_k^i(i)}))$$

Similar equations can be formulated for $E(|A_n(R_{n_i})|)$, $E(|R_n(A_{n_i})|)$ and $E(|R_n(R_{n_i})|)$. These equations could be made more precise by considering the cases under which certain added and removed sets cannot overlap. For example, if the onset of one input is contained in the offset of another, there cannot be overlap in the added onsets of these two inputs. The complexity involved in performing such estimation increases exponentially with number of inputs as it requires that the overlap between all input onsets/offsets be computed. It is demonstrated through the accuracy of the empirical results presented later in this chapter that there would be little to gain by using this more complex approach.

3.3 Predicting the Average as an Estimation Technique

Sec. 3.2 contains an estimation scheme based upon predicting the *expected* change in global power when a region of a network is resynthesized. However, the standard deviation of the estimator results has not been considered. If the standard deviation is large relative to predicting the effect of a *specific* resynthesis step then the method is not useful. It would imply that there has to be a many resynthesis steps performed on a circuit to guarantee an overall reduction in circuit power with a reasonable degree of confidence. This is undesirable as each resynthesis step is itself an event with a statistical likelihood of success with respect to local power reduction. The complexity of the issues involved in *a priori* estimating the possible reduction in power local to the resynthesis region implies that there will be only a small percentage of the entire network predicted suitable for resynthesis. Improvement in the results of resynthesis obtained by formulating a cost function from an estimator of large standard deviation but correct expectation could only be observed by averaging the algorithm performance over a large number of networks.

To make the discussion of this concept somewhat simpler, consider a reduction of the problem to the overlap of two sets A and B in a space C. It was outlined in the previous section how this problem is equivalent to the determination of the expected

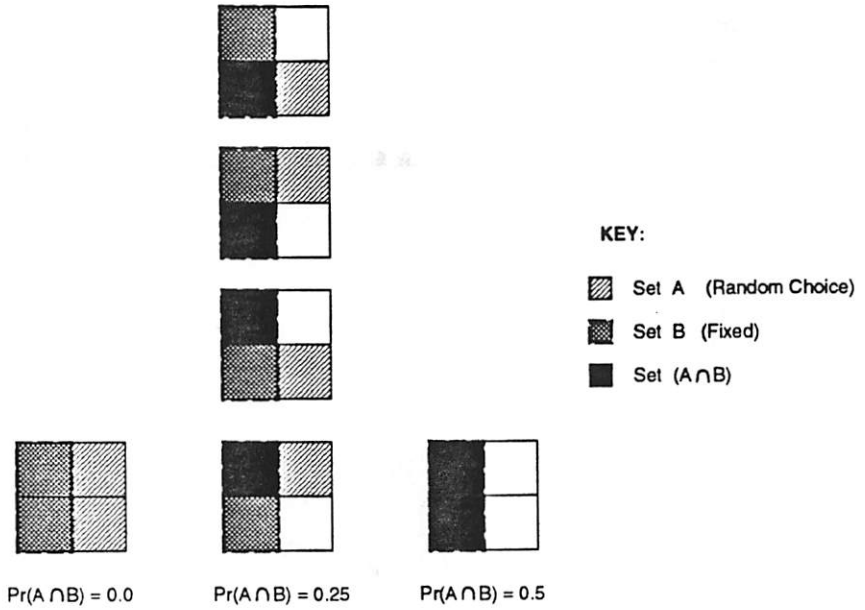

Figure 3.6: Overlap Variation for Two Sets A, B where Pr(A) = Pr(B) = 0.5

overlap between a change in functionality at a node input and the sensitivity set for that input within the ODC set. A is chosen at random by the synthesis algorithm, B is a *fixed* set representing the sensitivity. In Fig. 3.6 the possible variation in overlap for the case where $Pr(A) = Pr(B) = 0.5$ is illustrated. The entire space C consists of four elements of equal probability. Note that the maximum error in assuming the average is actually equal to the average, but there are four cases which generate the average only one case each for the two possible extremes.

A *change* in functionality (corresponding to a subset of the ODC) which occurs during local optimization of a network region is chosen independently of the sensitivity sets throughout the TFO. Consequently, the probability that a change in functionality chosen during resynthesis has a certain overlap with the sensitivity set is equivalent to the proportion of total possible function choices which have that overlap. For the example of Fig. 3.6, the probability of an overlap of probability of 0.25 is $\frac{2}{3}$, the probability of zero overlap or overlap of 0.5 are $\frac{1}{6}$ each.

The size of the set C determines the *peaking* of the overlap distribution. The larger the set, the more pronounced the peaking, and the smaller the resultant standard deviation relative to the expected change in TFO power. Fig. 3.7 depicts the probability of set overlap for $Pr(A) = Pr(B) = 0.5$, $|C| \in \{20, 60, 100\}$ points respectively. These three curves show

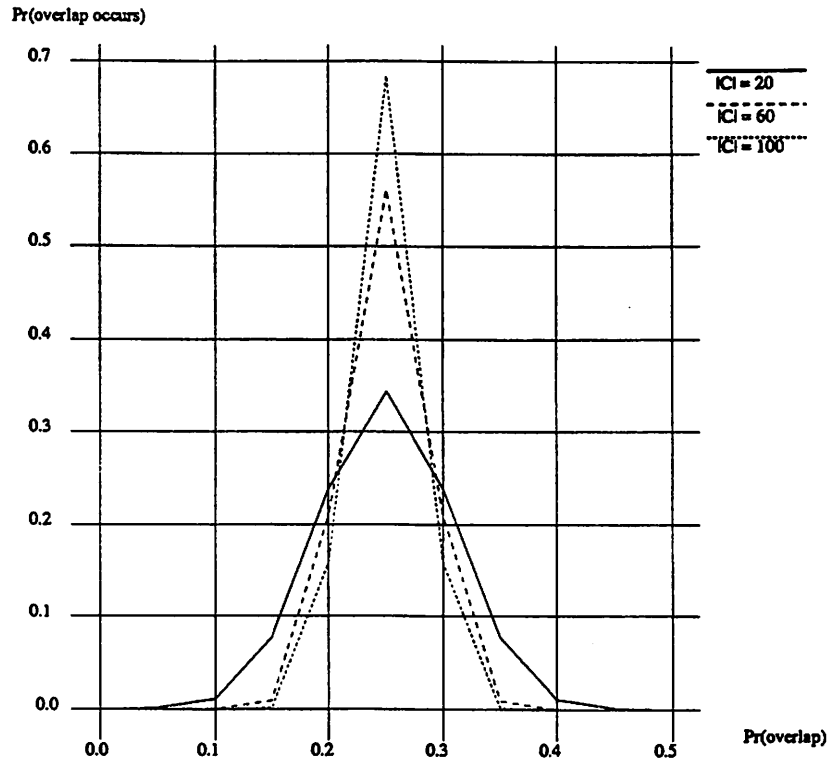

Figure 3.7: Overlap Variation as a Function of Set Size

the increase in peaking with the increase in the size of the set C. The standard deviation for the three curves is respectively: 0.057, 0.033 and 0.025. For $|C| = 100$, the standard deviation has already been reduced to 10% of the average.

In Fig. 3.8, and 3.9 graphs for the generalization of the concept discussed for $Pr(A) = Pr(B) = 0.5$ are presented. These two figures represent data for $|C| \in \{20, 100\}$ respectively. They are graphs of the ratio of the standard deviation to the actual average. The cases tested correspond to $Pr(A) \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$, $Pr(B) \in \{0.0, 0.1, 0.2, ..., 1.0\}$. The two graphs show how the standard deviation relative to the average set overlap decreases with increasing $|C|$.

There is a linear relationship between the natural log of the maximum ratio of standard deviation to average for a specific $\frac{|A \cap B|}{|C|}$ (denoted SD_{max}) and $ln(|C|)$. Fig. 3.10 depicts this for different $\frac{|A \cap B|}{|C|}$. From this, it is concluded that for $\frac{|A \cap B|}{|C|}$ fixed: $ln(SD_{max}) \propto (-0.5).ln(|C|)$. i.e. $SD_{max} \propto \frac{1}{|C|^{0.5}}$. Consequently, the error in estimating a specific change in power using an averaging technique reduces exponentially with the number of circuit inputs. A standard deviation of less that 2% of the average for $\frac{|A \cap B|}{|C|} \in [0.05, 0.95]$

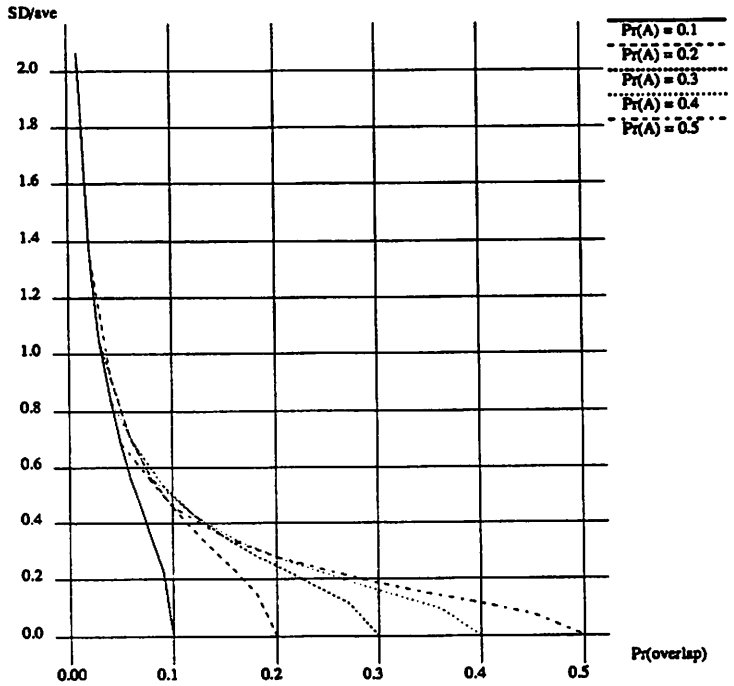

Figure 3.8: (Standard Deviation)/(Average) for $|C| = 20$

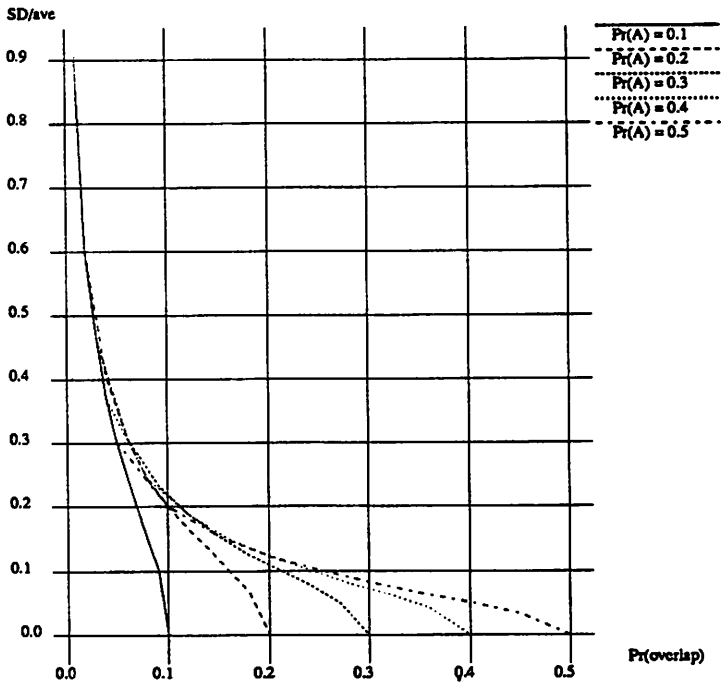

Figure 3.9: (Standard Deviation)/(Average) for $|C| = 100$

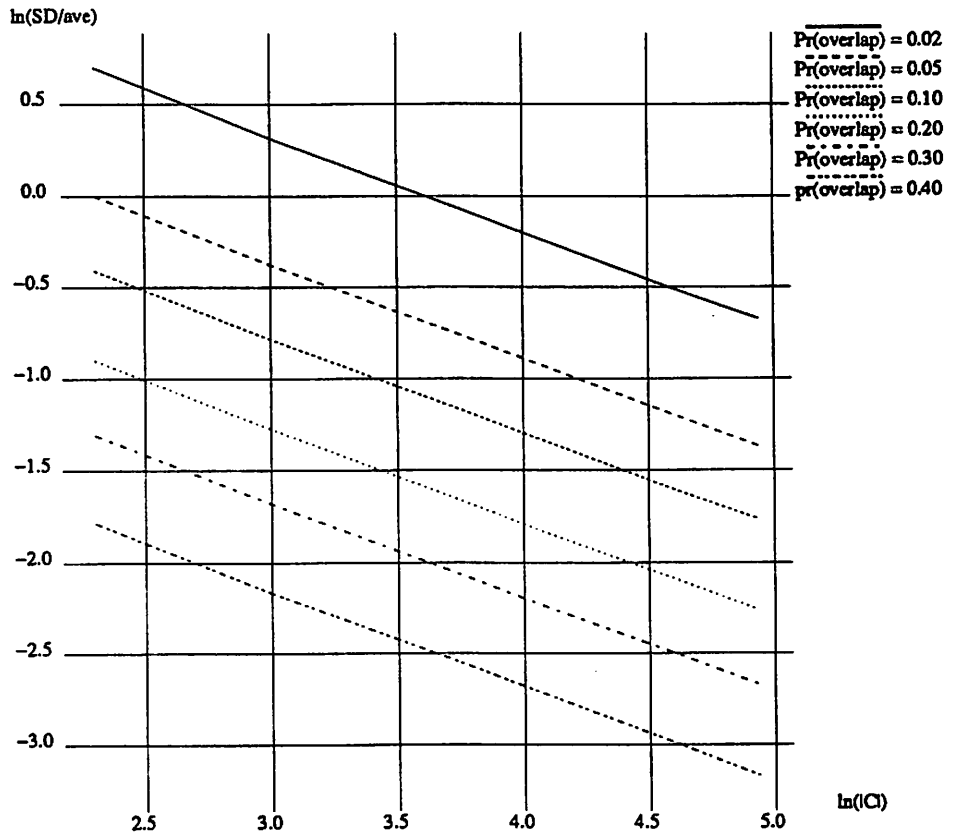

Figure 3.10: $\ln(\frac{SD_{max}}{ave})$ vs. $\ln(|C|)$ for six fixed $\frac{|A \cap B|}{|C|} \in \{0.02, 0.05, 0.10, 0.20, 0.30, 0.40\}$

is achieved with a network of 15 or more inputs. This justifies the use of an averaging technique and also validates the assumption of Eqn. 3.8. The small standard deviation is observed in the results of the following section.

3.4 The Functional Activity Change Estimator Accuracy

A program was written to empirically verify the theory of Sec. 3.2. The program randomly selects a set of nodes from a network. Every node in this set is resynthesized several times, each time with a random expansion or contraction of the onset of the node which guarantees a local change in functional activity. To imitate the behavior of a synthesis algorithm which intentionally changes the functionality (not just uses the flexibility of functionality within the ODC as resynthesis for area or timing will do), the change in function was implemented by the functional inclusion of a random set cubes from the BDD

Power Change		Occurrence		Estimation	
LOCAL	GLOBAL	Actual	Est.	Detected	Incorrect
Decrease	Increase	0.21	0.21	0.91	0.07
Decrease	$\leq 0.5x$	0.27	0.29	0.99	0.07
Decrease	$\geq 2x$	0.17	0.17	0.91	0.06
Decrease	$\geq 5x$	0.06	0.05	0.77	0.10
Increase	Decrease	0.09	0.08	0.84	0.05

Table 3.1: Functional Activity Estimation

representing the *local* onset or offset of a network node. The *local* onset/offset BDD is a representation of the node functionality relative to intermediate network nodes, not necessarily the primary inputs. These intermediate nodes are chosen by iteratively stepping back in levels from the test node until the onset/offset BDD representations explicitly contain 10 or more cubes. These BDD's are fairly small and are optimally ordered for compactness using the standard BDD variable ordering techniques offered in SIS[19]. A change in functionality using random selections of BDD cubes mimics the form of functional change likely in resynthesis in so far as eliminating functional changes which require major network reconfiguration.

Full symbolic simulation of the circuit using a technique based on the principles outlined in [7] is performed before and after each modification. This allows a direct comparison between actual change in power and that which the estimator predicts. Accuracy of the estimator relative to the actual change in power demonstrates the suitability of the theory for use in computing sensitivity of the TFO to node activity changes.

Fifteen circuits from the ISCAS '89 [25] benchmark set were examined in this experiment. They varied in size from 187 to 1005 literals. The circuits were initially mapped into the `msu` gate library and optimized using `script.rugged`[19] within SIS.

In Tab. 3.1 the results for functional activity change estimation are presented. The first two columns are the form of the statistic. The first column indicates whether the change in power local to the resynthesized node was decreased or increased. The second column indicates the actual global effect on power. A multiplier in this column indicates a bound on the global change in power relative to the local change. For example, $\leq 0.5x$ in Row 2 of the table implies that the beneficial change in the local power is reduced by more

than a factor of 0.5 by the corresponding increase in transitive fanout power. Column 3 is the probability of actual occurrence of the event indicated by the first two columns; and Column 4 is the estimated probability of this occurrence. Columns 5 and 6 indicate the accuracy of the estimator. **Detected** is the percentage of actual occurrences correctly detected; **Incorrect** is the percentage of the estimated points for this occurrence which are incorrect estimates (eg. 0.10 in this column for a particular event implies that in 10% of the predicted occurrences, a different event actually occurred).

Each row of the table presents a condition which would influence the suitability of node for resynthesis. Rows 1 and 2 are conditions under which a node becomes less suitable, while Rows 3, 4 and 5 are the converse. In particular, the nodes counted in Rows 3 and 4 are excellent candidates for resynthesis due to the strongly beneficial influence the local improvements have upon power throughout the transitive fanout. The estimator demonstrates an ability to predict better than 90% of the cases in which resynthesis becomes less suitable, and is only 7% overly conservative. More than 77% of the increased optimality conditions were detected with less than a 10% possibility of error.

Fig. 3.11 depicts the the strong correlation of the functional estimator to simu-

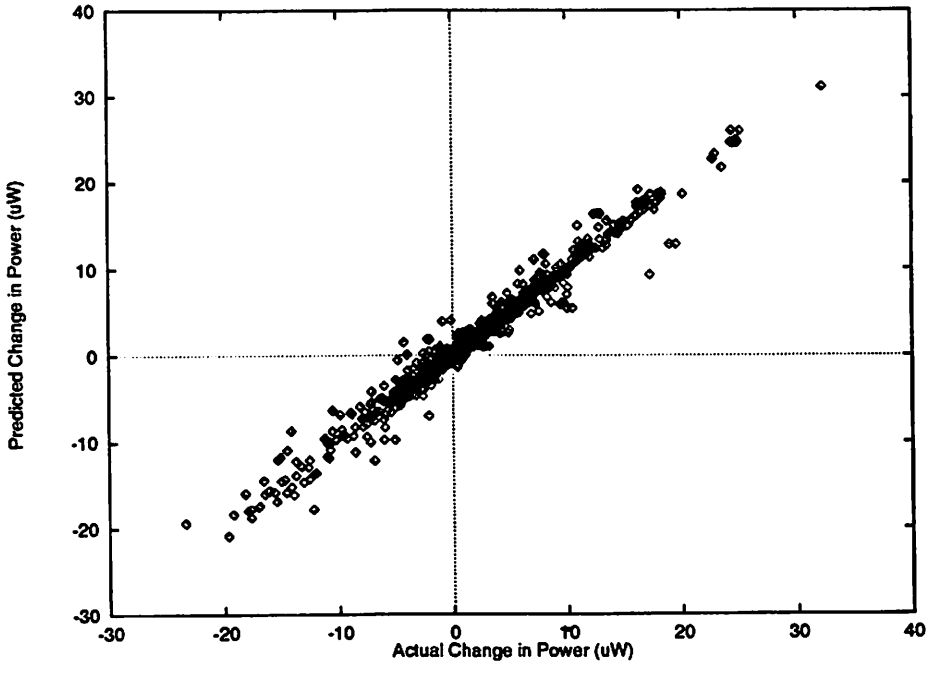

Figure 3.11: Change in Transitive Fanout Functional Power

Ckt.	Area	PI/PO/latch	Correl.	Ckt.	Area	PI/PO/latch	Correl.
s344	213	9 / 11 / 15	0.98	s820	458	18 / 19 / 5	1.00
s386	187	7 / 7 / 6	1.00	s832	453	18 / 19 / 5	0.99
s400	259	3 / 6 / 21	0.97	s838	449	34 / 1 / 32	1.00
s444	254	3 / 6 / 21	0.96	s1196	851	14 / 14 / 18	0.98
s510	378	19 / 7 / 6	0.95	s1238	838	14 / 14 / 18	1.00
s526	290	3 / 6 / 21	0.98	s1488	1004	8 / 19 / 6	1.00
s641	249	35 / 23 / 17	1.00	s1494	1005	8 / 19 / 6	0.99
s713	253	35 / 23 / 17	1.00				

Table 3.2: Estimate Correlation for Change in Functional TFO Power

lated changes in power. The *y-axis* is the estimated functional power change, the *x-axis* the actual functional power change. Each point corresponds to the result of a different possible resynthesis step. The power has been scaled relative to a 20MHz input vector arrival frequency and supply voltage of 5V. A unit on the graph corresponds to $1\mu W$. The correlation coefficient for the functional estimator is 0.99.

The results shown in Tab. 3.1 and Fig. 3.11 are aggregate statistics for all the circuits. The overall statistics are very representative of those obtained for each individual circuit. This is evident from the individual correlation coefficients listed in Tab. 3.2. In this table, **Area** refers to the literal count for each network which is a good high-level estimate of area, **PI/PO/latch** are the primary input/output and latch counts. In this experiment for combinational networks, the latches are treated as primary inputs.

3.5 Modifying the Theory for Arbitrary Input Probabilities

The theory of Sec. 3.2, Sec. 3.3 and the empirical results of Sec. 3.4 justify the accuracy of the basic estimation technique. It demonstrates the ability to predict the change in the size of the onsets throughout the TFO in terms of the number of minterms contained in each onset. A consequence of the assumption that all inputs have a probability 0.5 of being one is that all minterms in the space are of equal probability so a change in the number of minterms in an onset relates directly to the functional power consumption at that node. However, this statement does not generalize to the case where input probabilities are spread. Consider a simple Boolean space described by two variables $\{x, y\}$ where $Pr(x) = 0.2$ and

$Pr(y) = 0.4$. There are four minterms which describe this space: $\{\bar{x}\bar{y}, x\bar{y}, \bar{x}y, xy\}$ which have respective probabilities: {0.48, 0.32, 0.12, 0.08}. For this example there are no two minterms with the same probability of occurrence. The theory outlined previously would still correctly predict the expected change in power as the *average* minterm probability is always $\frac{1}{2^{|I|}}$ for a Boolean space described by $|I|$ variables. However, the standard deviation will be unacceptably large.

The material of this section is an outline of how the distribution of minterm probabilities throughout the Boolean space can be used to generalize the theory. The space is fully partitioned into a set of mutually exclusive classes φ, each class containing minterms of similar probability. Within each class, the estimation technique can then be applied to obtain $A_n^{C_i}(R_n^{C_i})$ for each $C_i \in \varphi$. The total probability added to (removed from) the onset of node n is then given by the sum over the different class contributions:

$$E(Pr(A_n)) = \sum_{C_i \in \varphi} E(|A_n^{C_i}|).\frac{Pr(C_i)}{|C_i|}$$

$$E(Pr(R_n)) = \sum_{C_i \in \varphi} E(|R_n^{C_i}|).\frac{Pr(C_i)}{|C_i|}$$

The accuracy/complexity trade-off depends upon the number and functional complexity of the classes. The worst accuracy, smallest complexity case places all the minterms in a single class and assumes a single minterm average, but this has no control over the error. The smallest error, highest complexity case splits the space into classes containing only minterms of *exactly* the same probability which in the worst case could generate a class for every single minterm. This also does not provide information useful to the selection of a suitable region for resynthesis as it is not possible to predict which specific minterms will define to the change in functionality occurring in resynthesis. The classes need to be sufficiently large to provide useful information on expected trends.

The form of the information useful to a resynthesis algorithm is that which separates the probability space into a *limited* number of classes, each of which contains a significant proportion of the total probability space and has an acceptable error for estimation. In the four minterm example outlined above, a possible partioning would be: $\{C_1 = \{\bar{x}\bar{y}.x\bar{y}\}, C_2 = \{\bar{x}y, xy\}\}$. Class C_1 encompasses 80% of the probability space and has an average minterm size 0.40; Class C_2 encompasses 20% with an average 0.10. C_1 has a maximum error of 0.08 relative to the average minterm size for that class, while C_2 has

a maximum error of 0.02, giving the total maximum error of 0.10 if these two classes are analyzed separately by the techniques of Sec. 3.2. This reduces the error from an original possible maximum of 0.30, considering the entire space with just a single average. (i.e. A minterm average with respect to the entire space is 0.25, and so $Pr(\bar{x}\bar{y}) + Pr(x\bar{y}) = 0.80$ would be estimated as 0.50.)

To generalize this concept, it is necessary to define a methodical technique for the generation of suitable minterm classes. There are six steps involved in the defining of these classes. Each of these are addressed separately:

- **Restricted Discretization of the Input Probabilities.** Inputs are grouped by similar probability. This minterm class construction is exponential in the number of separate groups.

- **Minimizing the Classes within an Error Bound**

  1. *Construction of Error Curves.* For each discrete input probability, an error curve is constructed for the different possible classes.
  2. *Construction of Probability Independent Classes.* The minimum set of classes is chosen for the desired maximum error by fixing the same error in each class.
  3. *Reducing Error through Probability Dependent Classes.* The probability of the classes is used to improve the overall error by reducing the error in the large probability classes, increasing it in the small classes.

- **Minimizing the Classes with Error Penalty**

  1. *Reducing Number of Small Probability Classes.* Following the previous step, it is likely that the error is smaller than necessary. This allows a tradeoff curve to be established for the number of classes against minimal error.
  2. *Combining Classes of Similar Average.* Any classes with the same average minterm size can be combined without affecting error. Classes with similar average can be combined by paying an error penalty. This is the last step to reduce the total number of classes to within a fixed bound.

3.5.1 Restricted Discretization of the Input Probabilities.

In terms of the minterm probabilities in the space, it is only necessary to consider the smaller of the onset/offset probabilities of an input. That is, an input x with $Pr(x) = 0.2$ or $Pr(x) = 0.8$ results in the same statistical minterm probability spread. Without loss of generality it may be assumed that a each network input has $Pr(1) \leq 0.5$.

Consider the case in which all input probabilities constitute a set of discrete probabilities P in the interval [0, 0.5]. Let the subset of inputs which have onset probability $p \in P$ be I_p. It will be shown empirically that a there is a simple tight bound on the error of predicting onset probability when the set of minterm classes are the product of classes chosen within every space defined by an $I_p, p \in P$. The simplicity of this bound behooves the approximation of the continuous probability space with a discrete set. As $|I|$ is finite, such a set always exists by default, but the number of classes is possibly exponential in the number of discrete probabilities. For example, if $|I| = 10$ but all the input probabilities are the same, the maximum set of useful classes is 11. However, if all input probabilities are different, there are 2^{10} possible classes. (This will be explained further in the error curve construction of Sec. 3.5.2.)

In general, as the number of discrete probabilities chosen to represent the entire set of input probabilities is reduced, so too is the accuracy of the power estimation. The restricted set of discrete probabilities needs to be chosen to minimize the RMS error of estimating the entire set of input probabilities. This loss of accuracy can be estimated by simulating the network with the restricted set of discrete input probabilities and comparing with the true simulated power. A restricted discretization is sufficient if the power estimate error is much smaller that the reduction in power regarded as 'significant' for resynthesis. For example, if a 10% or greater reduction in simulated power is found to translate reliably to a reduction in the power dissipation of the production silicon, then an acceptable error in the estimation is 1% or less.

3.5.2 Minimizing the Classes within an Error Bound

In general, it is not possible to test every set of input classes to determine those which are the most optimal. This makes an exact error/complexity tradeoff curve difficult to establish. Usually, only a limited error in the estimation is tolerable. A more practical approach to finding a set of classes is based upon the determination of a minimal set of

classes for a specific error, then examining the error/complexity tradeoff around that point.

Construction of Error Curves.

The error in a class C_i is defined as: $\max_{A \subset C_i} |(Pr(A) - |A| . \frac{Pr(C_i)}{|C_i|})|$. The maximum error is used as opposed to standard deviation as the distribution of minterms within a class cannot be approximated by a Gaussian distribution. Consider the set of inputs $I_p, p \in P$. The minimal set of classes will be generated from error curves for each of these input sets. Every minterm in the space Ω_{I_p} is one of $(|I_p|+1)$ possible sizes, those sizes being described by the set: $S_{I_p} = \{(1-p)^i . p^{(|I_p|-i)} | i \in \{0, .., |I_p|\}\} = \{s^i_{I_p}\}$. The minimum set of classes which fully partition the space Ω_{I_p} such that each class has zero error, $\varphi^0_{I_p}$, is the set defined by the minterm classifications of S_{I_p}. Let $\varphi_{I_p} = \{C_1, C_2, ...\}$ be a set of classes chosen from the union of classes in $\varphi^0_{I_p} = \{C^0_1, C^0_2, ...\}$. The maximum error will be minimized if the subset of $\varphi^0_{I_p}$ corresponding to any $C_i \in \varphi_{I_p}$ is contiguous with respect to indicies of $\varphi^0_{I_p}$.

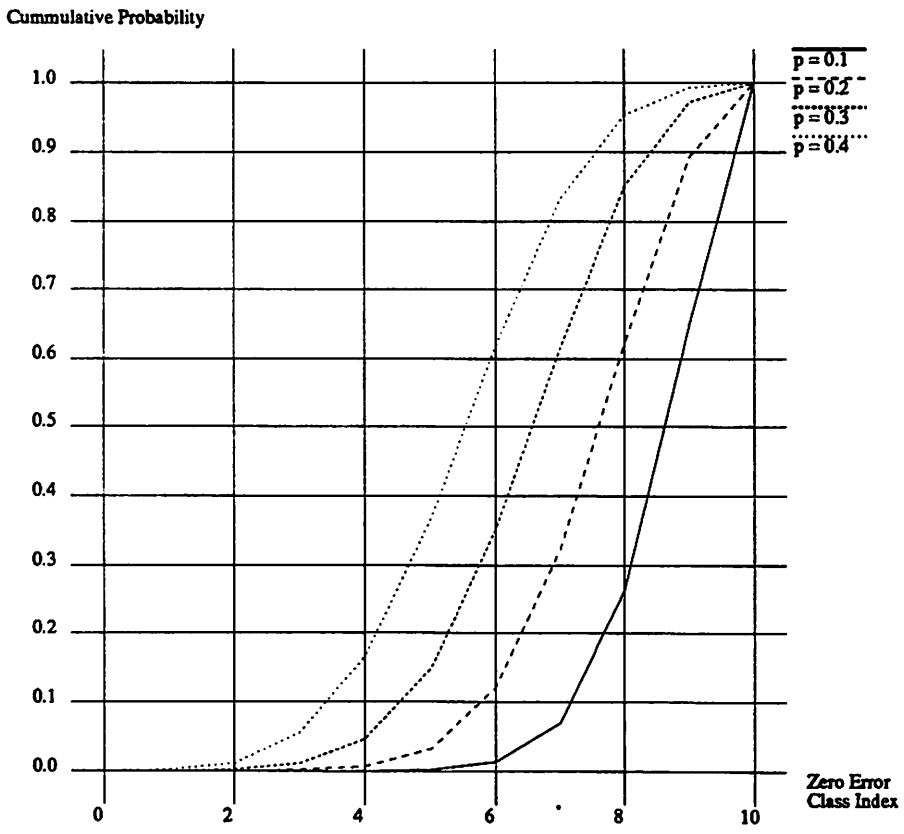

Figure 3.12: Probability Contained in Zero Error Classes: $|I_p| = 10$

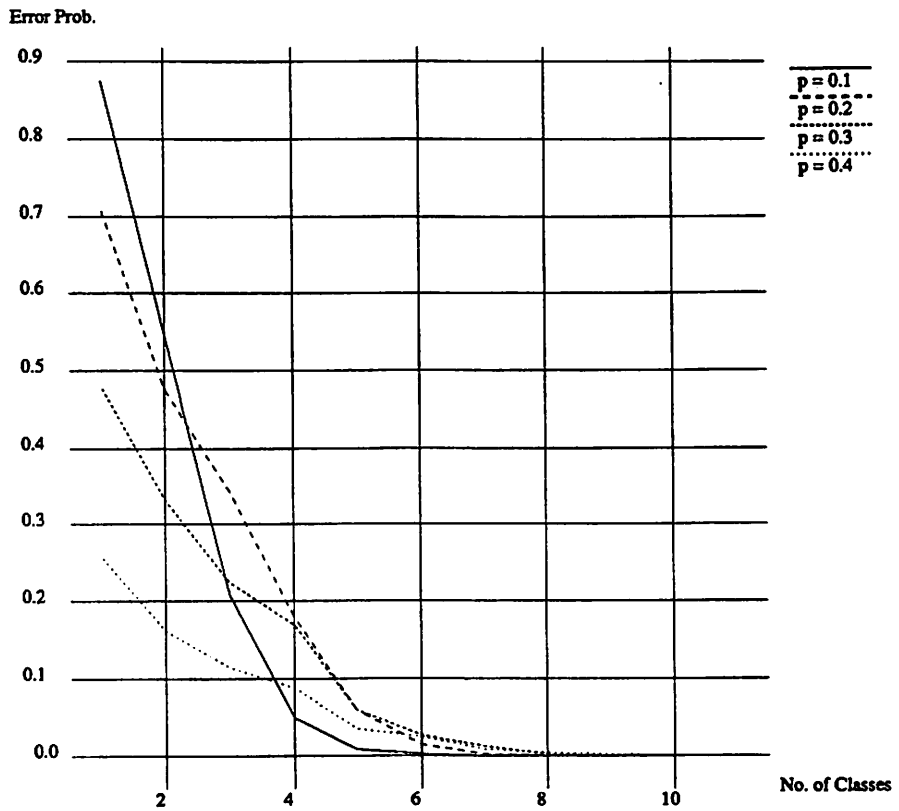

Figure 3.13: Error vs. Number of Classes: $|I_p| = 10$

eg. $C_1 = C_1^0 \cup C_2^0, C_2 = C_3^0 \cup C_4^0$ will have smaller error than $C_1 = C_1^0 \cup C_3^0, C_2 = C_2^0 \cup C_4^0$. Generation of φ_{I_p} for minimal error then becomes equivalent to the optimal selection of a set of contiguous, mutually exclusive subsets of $\varphi_{I_p}^0$. A heuristic technique to do this is the selection of the subsets of $\varphi_{I_p}^0$ in order to maximize the uniformity of the total probability contained in each of the final classes.

A large proportion of Ω_{I_p} is contained within a small subset of the classes in $\varphi_{I_p}^0$. Fig. 3.12 depicts the cumulative probability contained in the sets $\cup_{i<j} C_i^0$ against j, $p \in \{0.1, 0.2, 0.3, 0.4\}$. For increasing p, the curves are ordered right to left. The sharp gradient over a limited range in this graph implies that the final classes cannot actually be very uniform in total probability unless the total number of final classes is much smaller than $|I_p|$. For example, if one class in $\varphi_{I_p}^0$ has probability 0.3, it is not useful to break up this zero-error class to ensure that if there are 8 classes in φ_{I_p} all have a probability around 0.125. An approach which sensibly exploits the zero-error classes while still maximizing uniformity of the final classes and ensuring that these classes are contiguous with respect to

$\varphi^0_{I_p}$ is a greedy algorithm based upon grouping together the classes of maximum probability. Classes from $\varphi^0_{I_p}$ are added together until the probability of the new class exceeds the total probability remaining divided by the number of classes yet to be assigned. A plot of the total error (which is the sum of the error for each class) against the total number of classes for this class assignment strategy is shown in Fig. 3.13. For each curve, $|I_p| = 10$.

Construction of Probability Independent Classes.

The technique above generates a class/error tradeoff curve for a single p. However, in general, there are multiple probability discretizations, P. It is therefore desirable to find a minimal set of classes for the *entire* input space through utilizing the easily generated class/error tradeoff curves for each $p \in P$.

Consider the set of classes described by the Boolean product of classes chosen for each input subset I_p. The total number of classes for the overall space is then: $\prod_{p \in P} |\varphi_{I_p}|$. As each set of classes φ_{I_p} covers the entire space with mutually exclusive sets, the overall set of classes formed from the product maintains this necessary property. Let ϵ_p be the total error for the set of classes φ_{I_p}. An upper bound on the error for the set of classes on I defined by the product is given by:

$$\epsilon = 1 - \prod_{p \in P} (1 - \epsilon_p)$$

The tightness of this bound is illustrated in Fig. 3.14. The data on this graph is generated from a series of statistical tests for input probabilities $p \in \{0.1, 0.2, 0.3, 0.4\}$ with $|I_p|$ chosen randomly from $[2, 10]$ and $|\varphi_{I_p}|$ from $[0, |I_p|]$. The bounding technique is clearly sufficiently accurate for use when the total error is small.

It is necessary that the global error be small, so it may be written:

$$\epsilon \approx \sum_{p \in P} \epsilon_p$$

The global error can then be partitioned amongst the I_p and a minimal set of classes chosen using the class/error tradeoff curve. A positive (negative) *error sensitivity*, $\delta^+_p(\epsilon)(\delta^-_p(\epsilon))$ is defined for each I_p by computing the increase (decrease) in error corresponding to decreasing (increasing) $|\varphi_{I_p}|$ by 1. Similarly, each has a positive (negative) *class count sensitivity* with respect to the increase (decrease) in the total number of classes in the product, $\delta^+_p(|\varphi_I|)(\delta^-_p(|\varphi_I|))$. The total number of classes can further be minimized by

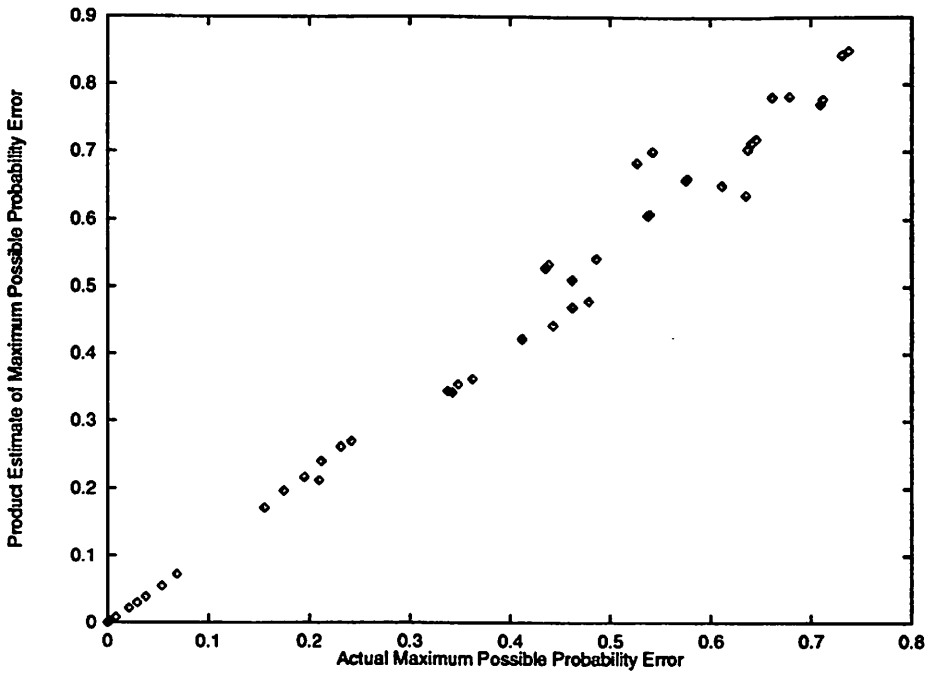

Figure 3.14: Actual Global Error vs. Product of Errors Estimate

first reducing error by increasing $|\varphi_{I_p}|$ for I_p with the greatest $|\frac{\delta_p^-(\epsilon)}{\delta_p^+(|\varphi_I|)}|$, then exploiting this error slack by decreasing $|\varphi_{I_p}|$ for I_p with the smallest $|\frac{\delta_p^+(\epsilon)}{\delta_p^-(|\varphi_I|)}|$. The procedure continues until the increase in classes needed for sufficient error slack to decrease any $|\varphi_{I_p}|$ exceeds $|\delta_p^-(|\varphi_I|)|$.

As an example, consider a network with $P = \{0.1, 0.2, 0.3, 0, 4\}$ and $|I_p| = 10$ for all $p \in P$. For zero error, the total number of classes required is 10^4. However, with an allowable 10% error $|\varphi_{I_p}|$ can be chosen respectively as: { 5, 6, 5, 6 } (From Fig. 3.13). There are 900 classes in this product, a reduction of over a factor of 10.

Reducing Error through Probability Dependent Classes.

The probability independent classes constructed from a product of the φ_{I_p} can be minimized by the techniques described above. However, this does not take into account the probability contained in each class. It is likely that for the same error bound, many of the small classes can effectively be combined if even a single large class is partitioned further. Consider the example of a circuit with 11 inputs, input a with $p = 0.1$, the others inputs with $p = 0.4$. For less than 10% error, $|\varphi_{I_{0.1}}| = 2$, $|\varphi_{I_{0.4}}| = 4$ so $|\varphi_I| = 8$. However, the set of all classes for which $\bar{a}$ is true consume 90% of the probability space. Suppose that for $\bar{a}$,

$|\varphi_{I_{0.4}}|$ is increased to 5. This reduces the error within this set of classes from 8.5% to 3.5%, the total error by $(0.9 \times 5.0) = 4.5\%$. The set of all classes for which a is true are 10% of the total probability space. The 4.5% error slack allows an increase in the error within these classes by 45% without increasing the total error above that of the original product. This allows $|\varphi_{I_{0.4}}|$ for these classes to be reduced to 1. This configuration has then maintained the error of the original configuration, but reduced the total number of classed from 8 to 6.

The construction of a global optimization routine based upon this strategy requires a estimate for the error within each class in terms of the errors for each I_p. Let $F(C)$ be the boolean function which describes class C. Let $P = \{p_1, p_2, .., p_m\}$, $\varphi_{I_{p_l}} = \{C_1(p_l), C_2(p_l), ...\}$, and the error in class $C_i(p_k)$ be $\epsilon_{C_i(p_k)}$. A class C is the product of the classes in the set $C_{\{P\}} = \{C_i(p_1), C_j(p_2), ..., C_n(p_m)\}$. i.e. $F(C) = F(C_i(p_1)).F(C_j(p_2))...F(C_n(p_m))$. An upper bound on the error of the class is:

$$\epsilon_C = 1 - (1 - \epsilon_{C_i(p_1)}).(1 - \epsilon_{C_j(p_2)})...(1 - \epsilon_{C_n(p_m)})$$

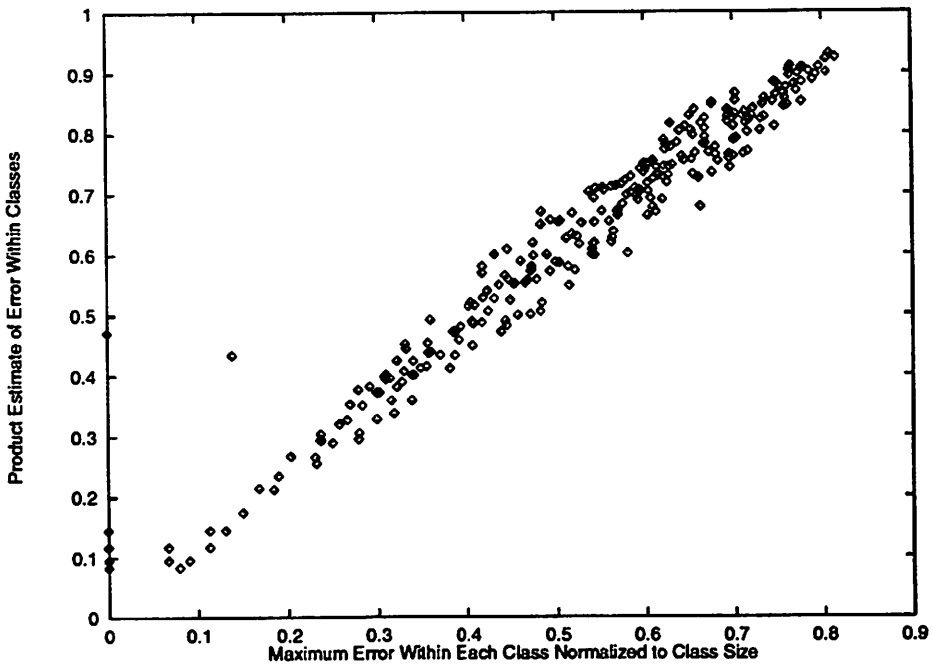

Figure 3.15: Actual Class Error vs. Product of Errors Estimate

The accuracy of this technique as a bound for the error within specific classes is shown in Fig. 3.15. The data is generated from the same test as that for Fig. 3.14, but the errors shown are scaled by the probability of each specific class. The data points shown are

only those for the non-trivial cases when there is more than one class in $C_{\{P\}}$ of non-zero error. With few exceptions, the bound on the error within a class is within 20% of the true error. This implies that the error curves for the independent I_p can be used to optimize the error for any specific product class.

The optimization of the total number of classes requires that there be a sensitivity established with respect to $|\varphi_{I_p}|$ for each class in $|\varphi_I|$. Define the set of classes $\varphi_{I\setminus I_p}$ on I with I_p excluded. For each $C' \in \varphi_{I\setminus I_p}$, $\varphi_{I_p}^{C'}$ is the subset of φ_{I_p} defined by all $C(I_p)$ which occur in the subset of φ_I which maps into to C'. This is a uniquely defined set of classes. For example, suppose $\varphi_I = \{C_a, C_b, C_c\}$ where $C_a = \{C_0(p_1), C_0(p_2)\}$,$C_b = \{C_0(p_1), C_1(p_2)\}$ and $C_c = \{C_1(p_1), \Omega_{I_{p_2}}\}$. Then $\varphi_{I\setminus I_{p_2}} = \{\{C_0(p_1)\}, \{C_1(p_1)\}\}$ and let $C'_m = \{C_0(p_1)\}$, $C'_n = \{C_1(p_1)\}$. Then $\varphi_{I_{p_2}}^{C'_m} = \{C_0(p_2), C_1(p_2)\}$ and $\varphi_{I_{p_2}}^{C'_n} = \{\Omega_{I_{p_2}}\}$.

A positive (negative) error sensitivity can be defined with respect to the set of classes in $\varphi_{I_p}^{C'}$. A similar method to that described for optimization of the probability independent class construct can then be applied to minimize the class count while remaining inside the specified bound on maximum total error.

3.5.3 Minimizing the Classes with Error Penalty

The techniques presented in Sec. 3.5.2 may not be sufficient to reduce the number of classes to within acceptable limits. The following techniques allow a class/error tradeoff curve to be established for the entire probability space. Resynthesis can then be attempted for various numbers of classes until the reduction in power being observed does not justify further time expenditure.

Reducing Number of Small Probability Classes.

The technique for optimizing error through the use of probability dependent classes can be extended to the formulation of the overall class/error tradeoff curve. Each point on this curve is found by computing the global error after reducing the number of classes described by the minimum ratio of positive error to negative class count sensitivity. This procedure is repeated until the error is too large to make the method usable, or the total number of classes is reduced below an acceptable number.

Combining Classes of Similar Average.

At each point of the class/error tradeoff curve, the total number of classes can be

reduced trivially without increased error by combining those classes with identical average minterms. This technique can also be extended to combine classes with different but similar averages. This incurs an error penalty bound by the change in the size of average minterm for each combined class multiplied by the total number of minterms in each class. This technique is useful if there is a fixed bound on the total number of classes desired. For each point which exceeds this bound on the class/error tradeoff curve, classes of similar average can be combined until the number of classes is acceptable. The configuration which generates the minimum error can then be chosen.

3.6 Ordering Inputs to Maximize ODC Set Flexibility

Prior to resynthesis, it is necessary to establish the ODC sets for each class and at every node in the network. For simplicity, the discussion which follows assumes just a single class for the entire space Ω_I. In general, the technique outlined can be applied within the subset of the Boolean space described by each class. This generalization is presented at the end of this section.

The construction of a compatible set of ODC subsets requires that a node input ordering be established [18]. A node input ordering which maximizes the ODC subsets for the most highly non-optimal nodes maximizes resynthesis flexibility. However, the non-optimality of a network region is a combination of both the resynthesis flexibility, *and* the expected gain obtained if that region of the network is resynthesized. Consequently, the choice of non-optimal nodes within a network has to precede the construction of a node input ordering. This motivates the definition of node non-optimality in terms of a *bound* on the possible size of the compatible ODC subsets.

An increase in the bound of the compatible ODC set at a node increases the expected flexibility. However, not all functional changes are useful options for the reduction of power. The determination of which options are likely to be beneficial requires theory or empirical evidence which directly relates the operation of a specific resynthesis algorithm to the network and ODC functionality. Without making an assumption about the form of the resynthesis algorithm the size of the ODC is the only possible measure of resynthesis flexibility.

The maximum compatible ODC subset which may be used for resynthesizing the input to a node is derived assuming that all other inputs to that node are held fixed. This

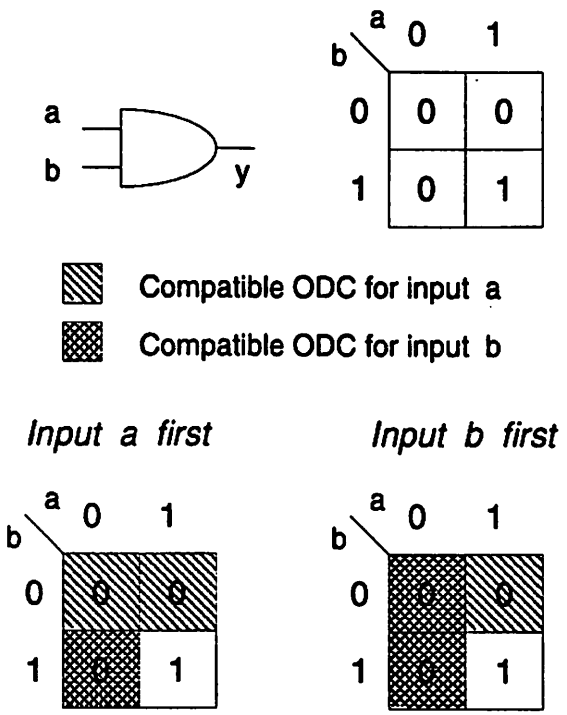

Figure 3.16: Increase in Compatible ODC for First Input of Input Order

corresponds to being the first input selected in the node input ordering, an example of which is depicted in Fig. 3.16. For this two input AND gate, suppose inputs a, b might be changed to a', b' but the output functionality is not to change. The compatible ODC for input a assuming b is fixed is: $\bar{b}$. These are the conditions under which the output is not sensitive to input a. However, this allows a new input $a' = a + \bar{a}\bar{b}$ so the only possible change in the other input can be to: $b' = ab$. The compatible ODC for b when input a has first order is therefore: $\bar{a}b$. Conversely, if input b is first in the ordering, the compatible ODC sets for inputs a and b are $a\bar{b}, \bar{a}$ respectively. Rigorously, if the ODC at the output of n is $ODC(n)$, then a bound on the ODC at input node n_i is:

$$ODC_n^{max}(n_i) = ODC(n) + \overline{S_n(n_i)} \tag{3.11}$$

In the computation of *expected* non-optimality, it is not statistically relevant whether $ODC_n^{max}(n_i)$ contains, or is contained within, the correct ODC for a node. This allows the the effect of reconvergent fanout to be neglected resulting in the very simple concatenation, expressed in Eqn. 3.12.

$$ODC_n^{max}(n_i) = ODC^{max}(n_i) + \overline{S_n(n_i)} \tag{3.12}$$

If an input n_i fans into a set of nodes $\{n\}$, the $ODC^{max}(n_i)$ is given by:

$$ODC^{max}(n_i) = \prod_{m \in \{n\}} ODC_m^{max}(n_i) \tag{3.13}$$

$ODC^{max}(n)$ is an estimate of the maximum function space which can be exploited in the resynthesis of node n. In Sec. 3.2, the entire function space Ω_I was assumed available for resynthesis. The modification of the theory presented there is the computation of probabilities within $ODC^{max}(n_i)$ rather than Ω_I. For example, assuming a single class for the entire input probability space, Eqn. 3.1 becomes:

$$E(|R_n(A_{n_1})|) = CntPr(S_n^n(n_1) \,|\, (\overline{f_{n_1}}.ODC^{max}(n_1))).|A_{n_1}| \tag{3.14}$$

It is necessary to show that this probability of propagation of a change is correct with respect to the ODC at the output of node n so that if there is an overlap between the functionality change at n_i and the sensitivity set within $ODC(n_i)$, this is the *only* possible change in functionality at n and, furthermore, this change is within $ODC(n)$. The former condition is straightforward as the change in functionality at the output requires sensitivity to input n_i. To prove the latter, consider the converse. This implies that there exists a subset of the $ODC(n_i)$ which overlaps the sensitivity set and is outside $ODC(n)$. However, if the node n is sensitive to input n_i outside $ODC(n)$, this is by definition in the care set of n_i, hence in $\overline{ODC(n_i)}$. This is a contradiction, proving the original assertion.

3.6.1 Finding the Highly Non-Optimal Nodes

For the establishment of an *expected benefit* of resynthesizing a node, an approximation has to be made correlating the size of the sets $ODC^{max}(n_i)$ to the size of the expected change in functional activity after resynthesis. This approximation can be constructed from an empirical study of the specific resynthesis algorithm.

To illustrate the concept of expected benefit, it might be assumed that the probability of the final resynthesized onset of a node varies as a linear function of distance from the pre-resynthesized probability of the node function being 1, and the distance to the probability of $f_n.\overline{ODC^{max}(n)}$ and $\overline{f_n}.\overline{ODC^{max}(n)}$. This is represented in Fig. 3.17. The change

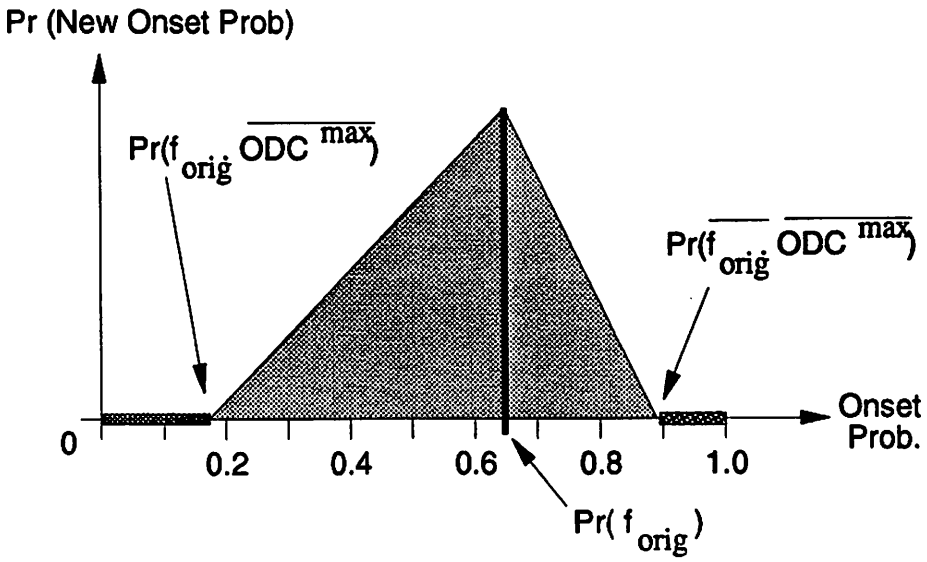

Figure 3.17: Resynthesis Probabilities

of transitive fanout power corresponding to various changes in onset size can be computed and scaled by the probability of the event occurring. The summation of these probability scaled power changes with the expected local resynthesis cost is a measure of the expected benefit, or *non-optimality*, $C(n)$, of the node.

Although this operation must be performed recursively throughout the transitive fanout of each node in the network, it is not expensive. Once the node sensitivities and ODC subset bounds have been calculated, all other operations are algebraic. Furthermore, a study of the results from the experiment outlined in Sec. 3.4 have shown that 90% of the power change throughout the TFO occurs within just five levels of logic depth.

The expected benefit computation for each node assumes no change in other nodes within the circuit. To allow resynthesis of multiple regions in one pass, an independence graph for the network needs to be built. This construction is established directly from the expected benefit computation as that analysis predicts the expected influence of one node upon another. The set of nodes best for resynthesis is the Maximal Independent Set extracted from the independence graph. An overall network resynthesis step would be repeated several times, each time generating a new set of expected benefits and an independence graph. The iterations would continue until no further improvement occurs.

3.6.2 Ordering the Compatible DC Set Construction

Subsets of the ODC subsets used to resynthesize the nodes identified as non-optimal need to be maximized and compatible. This implies an ODC construction similar to Savoj et al. [18]. The input ordering presented in that work is a partial order implied by parsing the network structure in reverse topological order. This ensures that the ODC is calculated at a node only after the ODC has been computed at all its TFO nodes, thereby ensuring a *maximal* construction. However, this input ordering is not unique to this processing order.

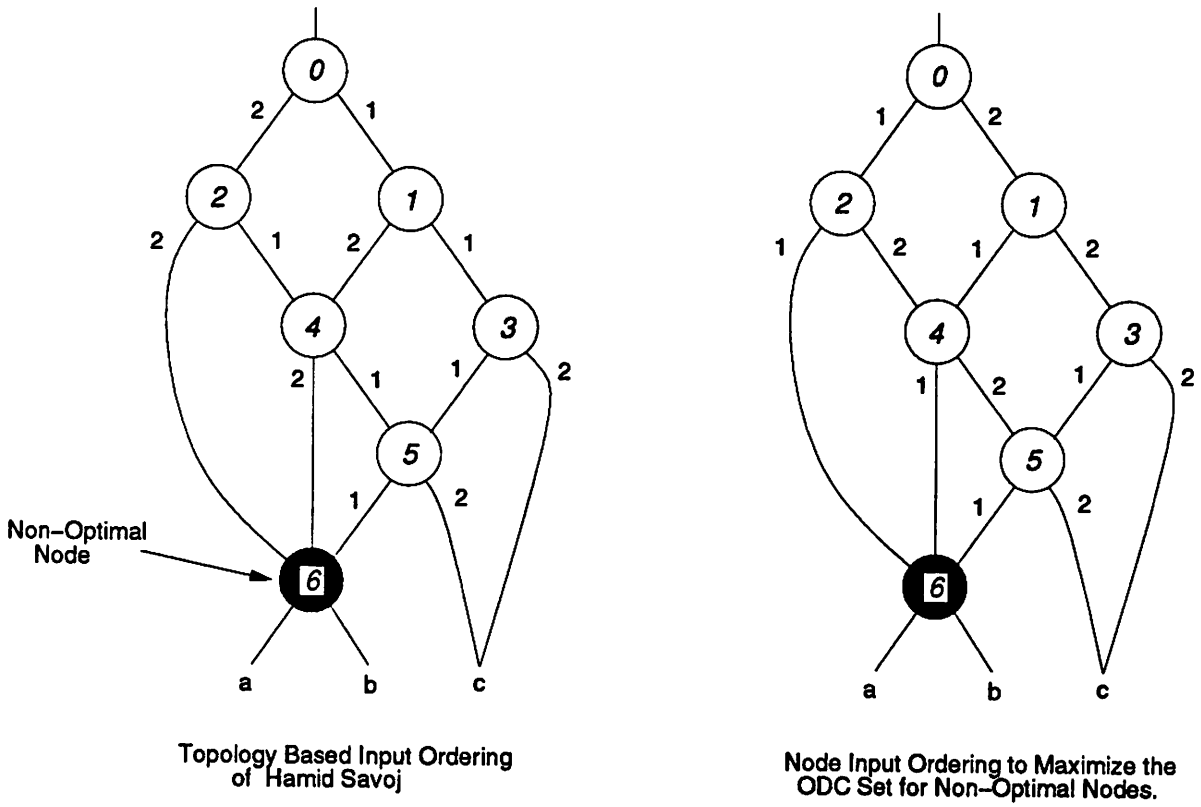

Figure 3.18: Node Input Ordering for ODC Construction

Consider the example of Fig. 3.18. The nodes are labeled in reverse topological order of the ODC construction, the edges numbered with respect to the input ordering applied to each node. In the left-hand figure, the node processing order is used to determine the input ordering as proposed in [18]. When the ODC is computed at a node, it 'calls' the construction of the ODC contribution from each output then takes the logical product of these functions. In this example, node 6 has been identified as being the most highly

non-optimal, therefore most desirable for resynthesis. Note how the inputs on the fanout of 6 do not all have the optimal order for maximal ODC freedom. The ODC at node 6 is restricted by an increase in the ODC set at nodes 4 and 5 where flexibility is not required for resynthesis. A more desirable ordering is that indicated in right hand network of Fig. 3.18. This input order can be maintained in the framework of the reverse topological processing by computing input ODC contributions rather than the total ODC at each node from scratch. For example, consider a node n. The ODC contribution of each input n_i of n to its fanin node can be computed given an arbitrary input ordering. When the node connecting to n_i is finally scheduled for processing using the reverse topological order, all of its fanout nodes have been processed so the contributions from each has been computed. The ODC at this node is then the logical product of all these functions. The ODC contributions from each fanin will then be *maximal* in the same way as defined in [18].

To optimize the resynthesis flexibility the node input ordering needs to be based upon:

- *Non-Optimality.* The greater the expected gain for resynthesizing a node, the higher priority it should be assigned in the node input ordering.
- *Proximity.* When close to a highly non-optimal node, this node should dominate the ordering strategy.

The node input ordering is established by assigning a weight w_{n_i} to the inputs in the following fashion. Let T_i be the set of nodes selected for resynthesis which are in the transitive fanin of n. For $x \in T_i$, d_x is the number of levels (excluding buffers, inverters and XOR's) between x and n.

$$w_{n_i} = \sum_{x \in T_i} \frac{C(x)}{d_x} \tag{3.15}$$

The input order is in reverse order of the weights, the higher weighted input assigned the highest priority, and so on.

If the space is separated into a set of classes φ_I, the non-optimality of each node may be different for each class $C_i \in \varphi_I$. For each class, a set of weights $w_{n_i}^{C_i}$ can be established and the subset of the ODC within each class computed according to this ordering. This ensures that if two regions do not have the same classes within which the bulk of their non-optimality (i.e. their best resynthesis choice) lies, then the ODC for each region will be optimally emphasized within the respective classes.

3.7 Summary

In this chapter the concept of using averaging as a suitable estimation technique for predicting the change in functional TFO power is presented. An averaging technique is important as there is only restricted information available prior to resynthesis of a region of a network. Even though the estimation strategy effectively neglects correlation between the original network node functionality and the specific expected change, it is shown that the standard deviation of such a method is small. This follows from the fact that the vast majority of functional changes of a fixed size at a node have similar overlap with sensitivity sets throughout the TFO of that node. Even though the extreme possible error condition relative to the predicted average may be significant, the statistical likelihood of such an occurrence is extremely small. This theoretical result is well supported by statistical evidence obtained though testing of the theory against actual changes in an extensive benchmark set. The spread of the predicted change in power relative to the actual simulated changes is sufficiently small to justify the use of this estimation strategy in improving the resynthesis choices for low-power resynthesis algorithms.

The estimation theory has been extended to formulate a technique for guiding of a resynthesis algorithm. The framework consists of three parts:

- *Class Definition.* A set of functions (classes) are defined which partition the space into regions of similar minterm probability. This minimizes the error of the estimation technique.
- *Specific Resynthesis Statistics.* The statistical relationship between the size of the ODC and useful local resynthesis options is used to determine the non-optimality of a network region.
- *Node Input Ordering.* Within each class, the node inputs are ordered in such a way as to maximize the resynthesis freedom for the highly non-optimal nodes.

The second of these three steps was presented in conceptual form only as it is a property of specific resynthesis approaches. The heuristic approach to Class Definition described in this chapter provides a technique for isolating a minimal number of classes for a specific error bound and a technique for the establishment of a class count/error tradeoff curve. The presentation concerning Node Input Ordering contains the outline of a definition for node non-optimality without having to first establish fixed compatible ODC sets. This definition is then used to establish a cost function for the compatible ODC node input ordering.

Chapter 4

Dynamic Activity Estimation: Bounded Delays and Expected Functionality Change

Resynthesis of a network node affects not only the functionality of the apex node as addressed in Chap. 3, but also affects the spurious dynamic activity in the network. This change in spurious dynamic activity results from changes in both delay and functionality within the resynthesis cone.

The significance of changes in spurious dynamic activity throughout the TFO of the apex node is critical in deciding suitability of a region for resynthesis. High sensitivity of the spurious dynamic activity in the TFO of a node to changes in delay, function or dynamic activity at the node restrict the resynthesis options which are globally beneficial. On the other hand, a node which does not have good resynthesis options in a local sense may become suitable for resynthesis if it can be shown that local changes in dynamic activity or delay can produce global reductions in dynamic power.

This defines the main issue which is addressed in this chapter. In particular, a theoretical and empirical study is made of the problem of estimating the sensitivity of the spurious dynamic activity in the TFO of a node to resynthesis of that node. The three aspects of this problem are presented separately:

- *Delay Sensitivity* (Sec. 4.2)

- *Dynamic Activity Sensitivity* (Sec. 4.3)
- *Functional Spurious Sensitivity* (Sec. 4.4)

The sensitivity to delay determines whether it is more beneficial to delay or speed up the output of the resynthesis region. A high sensitivity to dynamic activity implies that in structuring the resynthesized region, it is desirable to minimize path imbalance. Functional Spurious Sensitivity (as opposed to the topic of Chap. 3 which is *Functional Sensitivity*) determines whether a change in function is likely to affect the sensitivities of nodes in the TFO in such a way as to increase or decrease power.

In this chapter, it will be shown that a sensitivity with respect to bounds on delay does not provide useful information to guide resynthesis. To estimate the effect of delay on power, accurate estimates of the functionality of spurious dynamic activity and its precise arrival times are required. The inability to estimate these effects implies that a sensitivity to delay cannot be established for the output of a resynthesis region. Furthermore, it is shown that a change in the *amount* of spurious activity at the output of the resynthesized region. in combination with the Functional Spurious Sensitivity, are the dominate factors in the determination of a change in the spurious dynamic activity of the TFO. The estimation techniques of Chap. 3 are shown to be very accurate at estimating the Functional Spurious Sensitivity as well.

4.1 Effects Upon Dynamic Activity

The three aspects which affect spurious dynamic activity are illustrated here by example. In the estimation of dynamic sensitivity of the TFO of a node, all three aspects have to be accounted for simultaneously. However, they are presented here separately.

The effect of delay upon spurious dynamic activity relates back to the very definition of spurious dynamic activity as presented in Chap. 1.5. In Fig. 4.1, under the input conditions shown and delays $d_1 > d_2$ a spurious pulse is generated at the output. If $d_1 < d_2$ a similar pulse is produced for the opposite input vector pair, but if $d_1 = d_2$, no such pulse is observed under any input conditions. Change in the delays alone can influence the functionality which cause spurious transitions. The effect of producing spurious dynamic activity from non-spurious input functions is denoted *Spurious Activity Generation.*

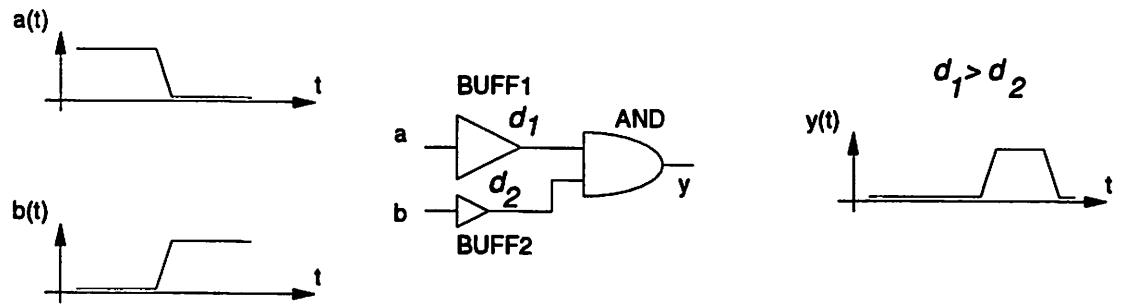

Figure 4.1: Spurious Output from Non-Spurious Inputs

A further example of the influence of delay upon spurious activity is shown in Fig. 4.2. In this circuit configuration, the only spurious activity at the output of AND1 occurs when input pair (a, b) changes from $(1, 0)$ to $(0, 1)$. For simplicity assume that the delay of all gates except BUFF2 is unity. If $d_2 \leq 1$, when the spurious activity at the output of gate AND1 occurs, there is a 1 present at the other input to AND2. This guarantees that this undesirable activity passes through to the output. If $d_2 \geq 1$, it is logically guaranteed that the spurious activity does not propagate. This effect of propagation or non-propagation of spurious activity from the input of a gate the the output is denoted *Spurious Activity Transmission.* The total spurious dynamic activity at a node is the sum of the Spurious Activity Transmission and Spurious Activity Generation contributions from each input.

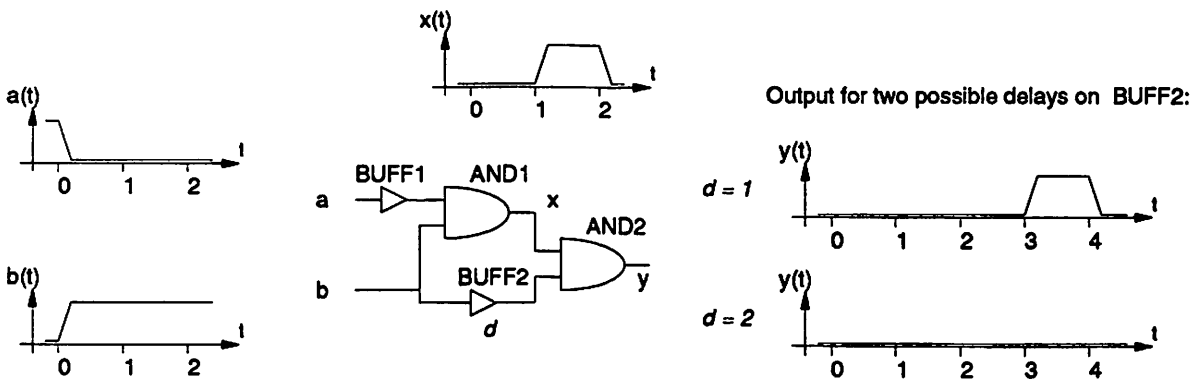

Figure 4.2: Spurious Output from Spurious Input

To distinguish between Dynamic Activity Sensitivity and Functional Spurious Sensitivity, consider an AND gate with inputs a and b. Assume that there is spurious dynamic activity on input a, but none on input b. Input b always arrives after input a, even following the changes described below.

As an example of Dynamic Activity Sensitivity, consider a change in the functional or spurious activity of input a while input b remains unchanged. Any increase or decrease in the spurious dynamic activity of a when $b = 1$ will be reflected in the output of the AND gate.

As an example of Functional Spurious Sensitivity, consider changes in the *functionality* of input b without introducing spurious activity at this input. This change in function alters the conditions under which a is observable from the output of the AND gate, consequently affecting the conditions for transmission of spurious activity. Furthermore, an increase in the correlation between a and $\bar{b}$ will increase the probability of generation of spurious activity at the output of the AND gate.

4.1.1 Defining Generation and Transmission

The definitions of *Spurious Activity Transmission* (Spurious Transmission) and *Spurious Activity Generation* (Generation) must be made rigorous. In particular, what constitutes spurious dynamic activity should be defined and separated from functional activity.

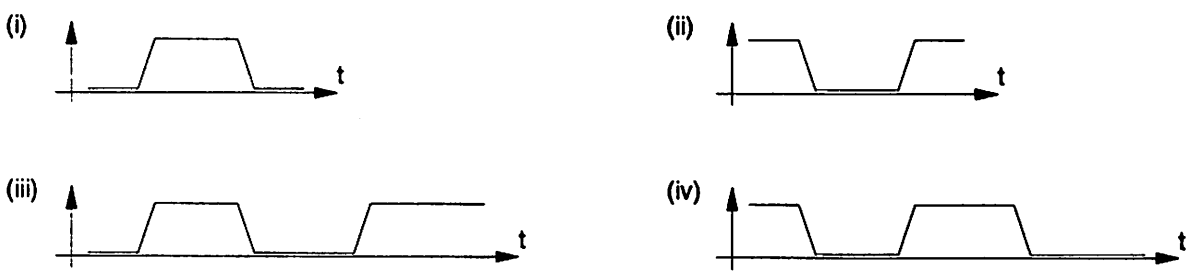

Figure 4.3: Types of Spurious Dynamic Activity

Consider Fig. 4.3. In cases *(i)* and *(ii)*, both transitions are clearly spurious as there is no final change in state. In cases *(iii)* and *(iv)*, either transitions 1 and 2, or transitions 2 and 3 may be regarded as spurious. In general, although the *amount* of spurious activity can be computed, a specific *set* of spurious transitions cannot be uniquely identified. In fact, this ambiguity brings into question the motivation for defining the concepts of Spurious Transmission and Generation, which are themselves based upon a definition of spurious and functional transitions.

The definition of Spurious Transmission and Generation becomes important in

the context of the design of an estimator which can be used to guide resynthesis with the limited information predictable prior to an optimization. The estimation strategy outlined in Chap. 3 is able to predict the *expected* change in functional power to a very high degree of accuracy, even in a circuit with arbitrary node input correlations. The work of [15] and [23] has demonstrated the need for estimation to consider such correlations so it is desirable to have a strategy for estimating changes in the amount of spurious activity *alone.* This result can then be concatenated with the functional activity estimator.

Consider the form of the activity at the output of the resynthesized section of the network. Before a resynthesis step is taken, an estimate of the dynamic activity which incorporates both function and timing information is considerably more difficult than estimating the final functional activity alone. This estimate would require a prediction of the structure of the resynthesized region which even incorporates loading effects. The determination of the applicability of a node for resynthesis must therefore be founded in variables which can be predicted and controlled. These variables are:

- *Delay Bounds* (B_L, B_U). The resynthesis step can be controlled in such a way as to bound the earliest and latest arriving output signals. However, the exact form of arrival time distribution is unable to be estimated prior to resynthesis. $[B_L, B_U]$ is denoted the *Activity Interval.*

- *Amount of Spurious Activity* $(|T^D|)$. By balancing the paths, the amount of spurious activity at the output of the resynthesized region is likely to be reduced. However, the functionality of the spurious dynamic activity cannot be predicted *a priori.*

- *Expected Change in Functionality* $(|T^F|)$. As outlined in the previous chapter, it may be possible to predict the ability for a low-power resynthesis technique to change the probability of the resynthesis region output being one. However, it is not possible to predict the exact change in functionality which is the optimal choice for resynthesis.

An example of how these variables relate to actual activity is shown in Fig. 4.4 for a single waveform. For example, B_L is the earliest time for any transition to be seen at the node, B_U is the latest. In general, these variables summarize the behavior of the node for all input vector pairs. Within this restricted set of describing variables, $\{(B_L, B_U), |T^D|, |T^F|\}$, the concepts of Spurious Transmission and Generation are well defined.

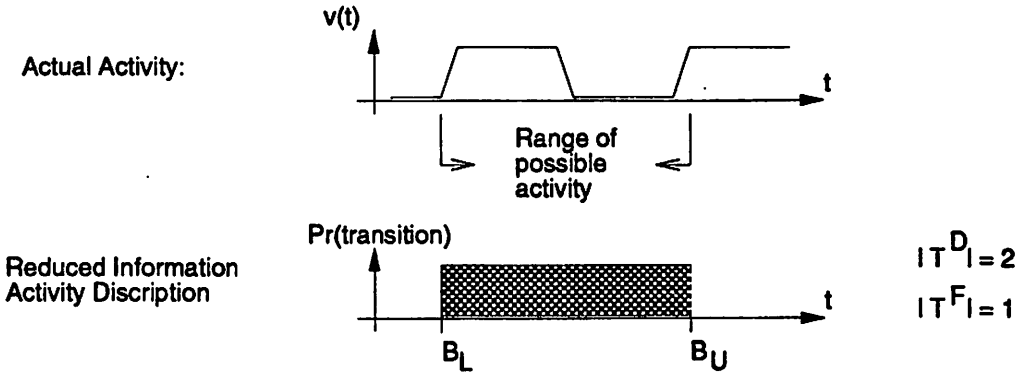

Figure 4.4: Dynamic Activity Variables to Guide Resynthesis

Generation is defined in terms of the *probability* that one input n_j to node n has resolved to its final state after application of a input vector pair before, after or during a change in state of the other inputs $\{n_j\}$. If the change in state of n_j is propagated through the node and changes the output to a state different to the known final output state, this is Generated spurious activity at the output. The concept is illustrated for two cases of a three input AND gate in Fig. 4.5. In this example, the contribution to generated activity at the output of the AND gate due to input a is under consideration. In the first case presented, input a arrives after any transition on input b and before any transition on input c. The original state of each input is indicated by lower case, the final by upper case. Clearly, if $Bc = 1$, any positive transition on a is propagated through the gate. However, if $C = 0$, then the final state of the output is 0. This implies that the activity propagated from input a under these conditions is not necessary, consequently spurious. This condition generates two spurious output transitions, one transition which incorrectly changes the output phase, and one which corrects it. In the second case, a and b may arrive simultaneously. The conditions for which a generates spurious activity are the same as that for the first example. However, for the generation conditions under which both a and b has positive transitions, the contribution from a is halved. This allows independent summation of Generated Activity contributions for each gate input.

The alternative to defining Generated Activity contributions from each input would be to associate the total output activity to a unique input arrival order. This is of the same complexity as the Generated Activity construction, but does not have a straightforward simplification to a pairwise correlation approximation.

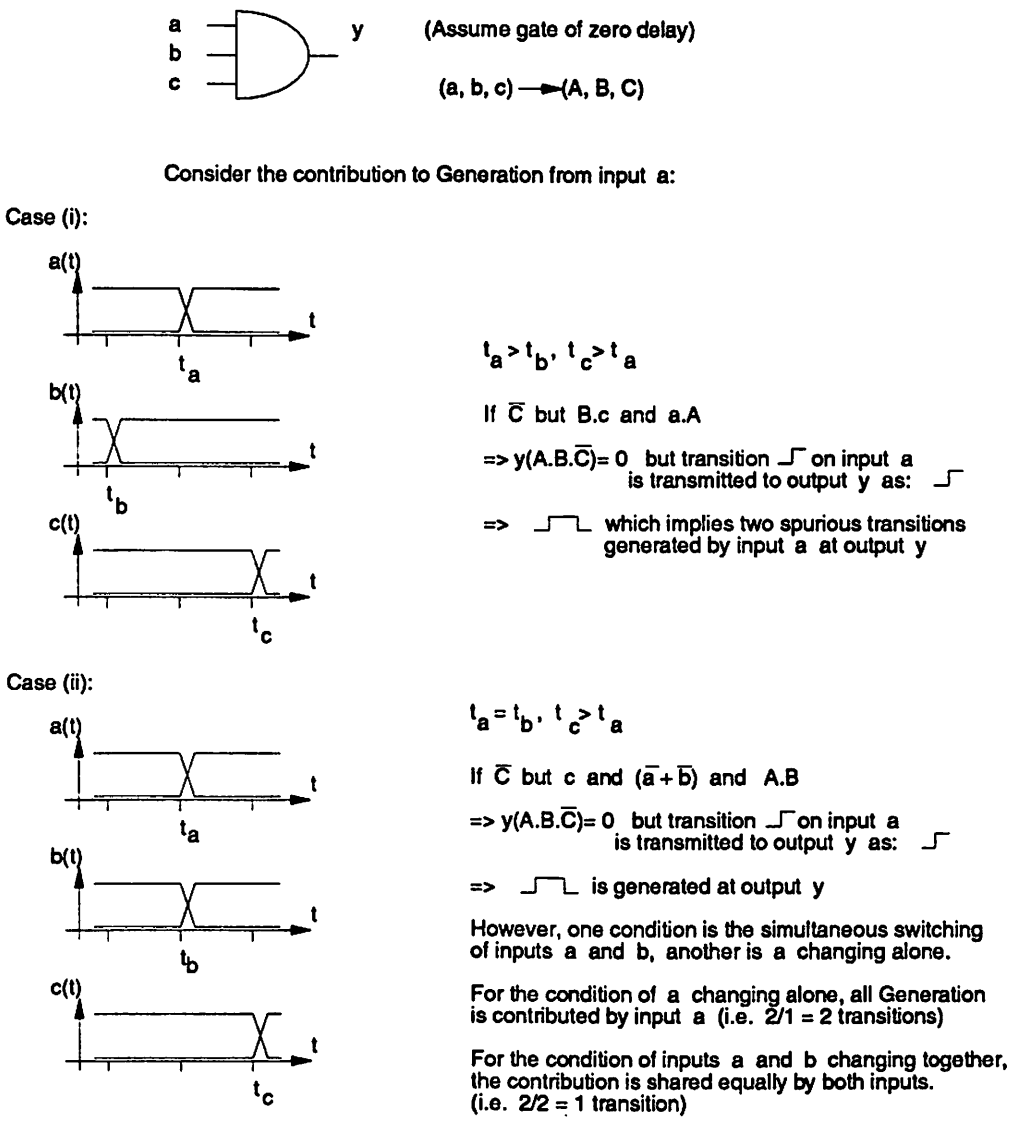

Figure 4.5: Example of Different Generation States

Spurious Transmission is defined in the context of the following two assumptions:

Assumption 4.1.1 *Spurious dynamic activity at an input to a node is independent of the spurious dynamic activity at the other inputs.*

Assumption 4.1.2 *A spurious transition on one input does not arrive simultaneously with a spurious transition on any other input.*

These assumptions are a direct consequence of the formulation of the set of three

variables $\{(B_L, B_U), |T^D|, |T^F|\}$ to be used in the estimation. The first assumption does not increase the error of the estimator beyond the information already lost in the formulation of the three activity describing variables. This assumption follows from the fact that prior to resynthesis it is not possible to estimate how spurious dynamic activity will be related to an input function. The second assumption uses the fact that if the spurious dynamic activity is assumed independent, the probability of simultaneous spurious transitions on multiple inputs is the product of each occurring separately. This is a low probability event, so it can be assumed to not occur without increasing the first order error of the estimate.

The Spurious Transmission at the output of a node contributed by a specific input is given by multiplying the amount of spurious activity at the input by the probability that the node is sensitive to that input. The total Spurious Transmission and Generation at a node is formed by summing the contributions from the individual node inputs. Spurious Transmission and Generation produce *exact* estimates in the case where no input to a node has spurious dynamic activity, or in the case where the spurious dynamic activity is independent on all the inputs and these input activity intervals do not overlap.

4.2 Sensitivity to Delay

In this section, a mathematical formulation of the concepts of generation and transmission is presented. There are two levels of approximation. The first assumes that functional input transitions pass through a gate with a probability independent of that function so the sensitivity of the node to an input is independent of that input. The second level of approximation is an exact formulation for a two input gate network. In both cases, the theory presented is generalized to multiple input gates using a pairwise correlation assumption.

It is shown that the concept of transmission and generation is only useful under the conditions where time is very coarsely discretized. When time is discretized as recommended in [7] for sufficient accuracy of power estimation, the transmission/generation estimation technique is unable to detect optimality. This implies that delay bounds are insufficient measures for computing delay sensitivity. It is therefore shown that optimality obtainable through delay manipulation requires estimates of the functionality and time distribution of the spurious dynamic activity. This is the basis of the material in Chap. 5.

4.2.1 The Functionally-Independent Delay-Sensitive (FIDS) Model

The pairwise correlation formulation is derived from theory based on the assumption that only two inputs actually change.

Theory for Gates with Two Inputs Changing

Consider node n with inputs $\Lambda_n = \{n_1, n_2, ..., n_m\}$. Assume that the two inputs which change are inputs n_j and n_k. The total amount of Spurious Transmission at the output of n, T_n^{DT}, is given by:

$$|T_n^{DT}| = \sum_{n_i \in \{n_j, n_k\}} Pr(S_n(n_i)).|T_{n_i}^D| \tag{4.1}$$

For Generation, the functional input transitions are considered. A functional input transition is passed though the node n as a *spurious* transition if it changes the node output to the incorrect final state. The probability of Generation is computed separately with respect to the two possible input transitions - high to low, and low to high.

Assume that input n_j has a functional transition from high to low. This transition is propagated as a spurious transition if it is transmitted with negative phase and the final output of n is 0, or it is transmitted with positive phase an the final output is 1. This is summarized in the two terms of the following equation:

$$P_n^{g\downarrow}(n_j) = Pr(S_n^n(n_j)).Pr(\overline{f_{n_j}}.\overline{f_n} \mid \overline{f_{n_j}}) + Pr(S_n^p(n_j)).Pr(\overline{f_{n_j}}.f_n \mid \overline{f_{n_j}}) \tag{4.2}$$

Similarly, for an input transition from low to high:

$$P_n^{g\uparrow}(n_j) = Pr(S_n^n(n_j)).Pr(f_{n_j}.f_n \mid f_{n_j}) + Pr(S_n^p(n_j)).Pr(f_{n_j}.\overline{f_n} \mid f_{n_j}) \tag{4.3}$$

Overall, these may be summed to give a *total* probability of a functional input change being propagated as a spurious transition:

$$P_n^g(n_j) = P_n^{g\downarrow}(n_j) + P_n^{g\uparrow}(n_j) \tag{4.4}$$

The amount of Generated activity at the output of node n contributed by input n_j is then obtained by multiplying the result of Eqn. 4.4 with the probability that there

is a functional transition on n_j, and that this input arrives early. Summing over the two inputs which change, the total generated activity is obtained:

$$|T_n^{D_G}| = \frac{1}{2} \sum_{n_i \in \{n_j, n_k\}} P_n^g(n_i).P_{early}(n_i).|T_{n_i}^F| \tag{4.5}$$

The probability that a transition arrives early is computed assuming that the functional transitions in the interval $[B_L(n_i), B_U(n_i)]$ are uniformly distributed amongst the discrete time points. The uniform distribution assumption is a consequence of the lack of more accurate information prior to a resynthesis step. $P_{early}(n_i)$ actually consists of two terms: the probability that a functional transition occurs on the other changing input; and the probability that the functional transition on n_i precedes the other given that both inputs transition. This is outlined following the generalization of the two input theory.

Generalizing to Gates with Arbitrary Input Changes

To generalize the theory for two-input gates multiple input changes, it is assumed that the probability that functional transitions occur on more than two gate inputs is much less significant than the probability of a single input pair changing. For example, consider a gate with three independent inputs, each with a probability of changing of 0.5. The probability that a *pair* of inputs change is 0.475, and the probability that *three* inputs change is 0.125. Hence, 75% of the multiple input change conditions is associated with just input pairs. The applicability of this approximation depends strongly upon the input correlation, but for the pairwise approximation it is taken as a general assumption.

The formulation of Spurious Transmission is the same for an arbitrary gate as for a gate with only two inputs changing, except the sum is now across all inputs.

$$|T_n^{D_T}| = \sum_{n_i \in \Lambda_n} Pr(S_n(n_i)).|T_{n_i}^D| \tag{4.6}$$

For Generation the probability that an input n_j arrives earlier than another gate input n_k is approximated using pairwise correlation. The probability that a transition on input n_j arrives earlier than some other input is equivalent to the probability that it doesn't arrive later or at the same time as other gate inputs. i.e. $P_{early} = 1 - P_{late} - P_{same} = 1 - P_{late,same}$. To notate this, two further symbols are defined: σ_i is the condition under which input n_i has a functional transition, $P_{early}^{n_i}(n_j)$ is the probability that a functional

transition on input n_j arrives earlier than a functional transition on n_i given that both inputs transition.

$$P_{late,same}(n_j) = \prod_{n_i \in \Lambda_n \setminus n_j} \left(Pr(\overline{\sigma_i} \mid \sigma_j) + (Pr(\sigma_i \mid \sigma_j).P^{n_j}_{early}(n_i)) \right)$$

This is then used to generate a scaling factor $K_{early}(n_j)$ for the sum of the individual $P^{n_i}_{early}(n_j)$.

$$K_{early}(n_j) = \frac{1 - P_{late,same}(n_j)}{\sum_{n_i \in \Lambda_n \setminus n_j} Pr(\sigma_j \mid \sigma_i).P^{n_i}_{early}(n_j)}$$

Eqn. 4.5 is then modified by summing the contribution to Generation from each input relative to every other input. $P^{g(n_j)}_n(n_i)$ is the probability that a functional change on input n_i is propagated as a spurious transition assuming all inputs other than n_j remain fixed.

$$|T^{D_G}_n| = \frac{1}{2} \sum_{n_i \in \Lambda_n} \left(\sum_{n_j \in \Lambda_n \setminus n_i} P^{g(n_j)}_n(n_i).K_{early}(n_i).Pr(\sigma_i \mid \sigma_j).P^{n_j}_{early}(n_i) \right).|T^F_{n_i}| \qquad (4.7)$$

This results in what is refered to hereafter as the Functionally-Independent Delay-Sensitive (FIDS) model.

4.2.2 Testing the Functionally-Independent Delay-Sensitive (FIDS) Model

The FIDS model has been tested on networks with the unit delay gate model. The results show that the technique can be used to drive an algorithm which inserts delays to balance network paths thereby minimizing the amount of spurious dynamic activity generation. Although the approach of inserting power consuming delay elements would be a bad technique in practice, it is a theoretical demonstration of the accuracy of the estimation routine. The test demonstrates the ability of the estimator to predict whether an expected change in transitive fanout power more than offsets the power consumed in the buffers required in a balance step.

The concept of path balancing is shown in the example of Fig. 4.6. Assume that this is a small circuit embedded in a larger network. Let the set of inputs $\{a, b, c\}$ resolve to $\{A, B, C\}$. For the purpose of illustration, consider the balancing of AND2. In the original circuit, a glitch is generated at the output of AND2 under the logical conditions:

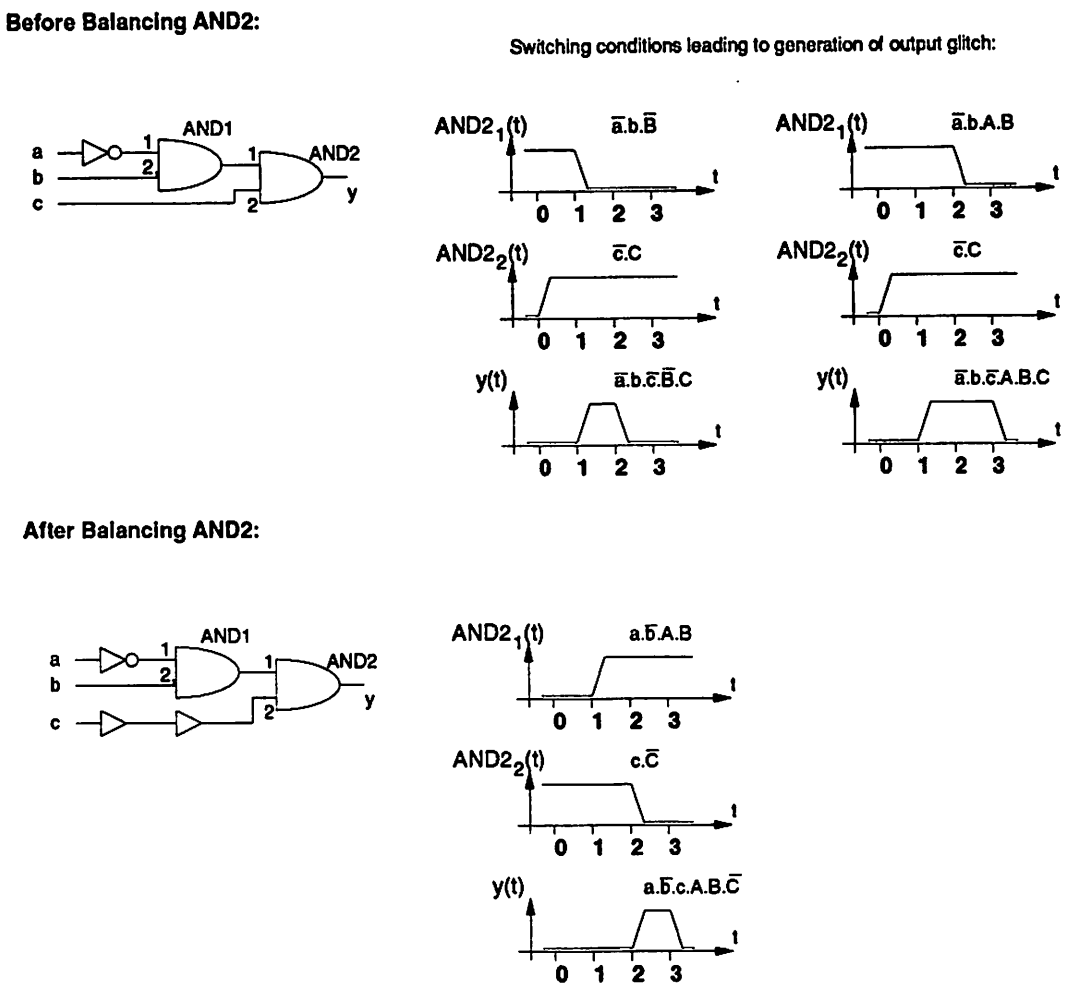

Figure 4.6: The Concept of Balancing Path to Reduce Activity

$\bar{c}C\bar{a}b(AB + \bar{B})$. By inserting two unit delay buffers on input AND2$_2$, the arrival of the transition $\bar{c}C$ at this input is simultaneous with the signal transition $\bar{a}bAB$ at input AND2$_1$. Consequently, this transition does not generate a spurious output transition. The conditions for a Generation at AND2 are now: $a\bar{b}Bc\bar{C}$. If this reduction in spurious output activity when propagated throughout the TFO of AND2 reduces network power consumption by more than the power dissipation of the inserted buffers, the balance is a valid optimization. The estimation of the change in power both local to a gate and throughout the TFO is to be performed using the FIDS estimation technique. In this test, the paths are only balanced to the *latest* arrival time of the node inputs. (Note that when the delay intervals are finite, the concept of 'balancing' two paths is not well defined.)

The change in delays caused by balancing paths makes it necessary to estimate the change in activity intervals throughout the TFO. Upper and lower time bounds are handled in a similar manner, so the heuristic used is described here only for the upper bound. In Fig. 4.7, a gate with delay d has three inputs with delay bounds illustrated: $B_U(n_3) < B_U(n_1) < B_U(n_2)$. Let the delay of input n_2 be adjusted either positively or negatively, and let $(B_U(n) - d)$ be in the interval $(B_U(n_1), B_U(n_2))$ such that $(B_U(n) - d) = (B_U(n_2) - \delta(n_2))$. Let $B_U(n_2)$ be changed to $B'_U(n_2)$. The new upper bound on the output activity interval $B'_U(n)$ is then estimated: $B'_U(n) = \max\{(B'_U(n_2) - \delta(n_2)), B_U(n_1)\} + d$. In general, $B'_U(n) = \max_{n_i \in \Lambda_n}\{(B'_U(n_i) - \delta(n_i))\} + d$, where $\delta(n_i) = 0$ if $(B_U(n) - d) \geq B_U(n_i)$.

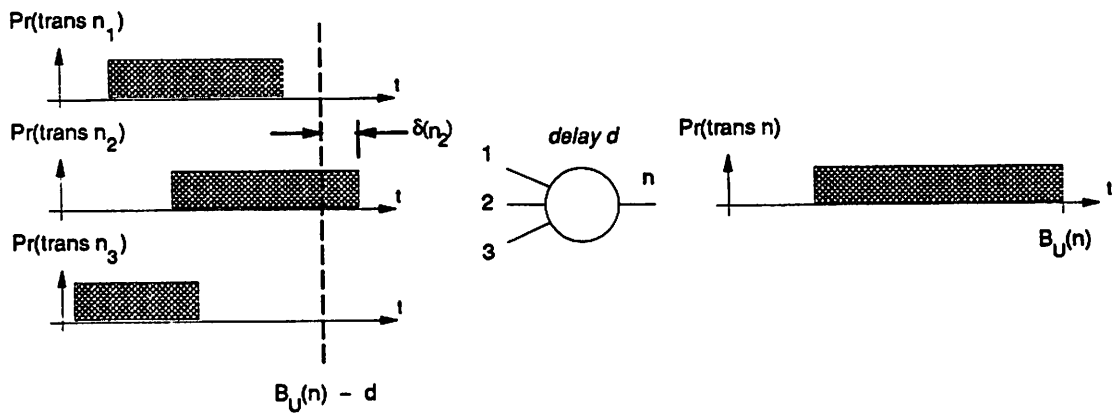

Figure 4.7: Estimating the New Delay Bounds

The balancing algorithm works forward from the primary inputs through to the primary outputs. For each node output, strings of unit delay buffers are constructed which balance this signal for each fanout gate. This is illustrated in the example of Fig. 4.8. The gate interconnection on the left hand side of the figure is presented against a balancing time scale for four different cases of buffer strings balancing zero, one, two or all three fanouts. Each possible buffer string is analyzed for its contribution to overall network power consumption. The optimum length buffer string is then chosen. The power consumption of each buffer is assumed equivalent to an inverter with minimum sized transistors.

The insertion of a buffer string is made *before* it is known whether further buffers will be inserted in the TFO. This, however, does not add error to the testing of this estimation technique. The buffers in the TFO will only be added if their insertion further reduces overall network power consumption. Their cost is therefore accounted for at their time of

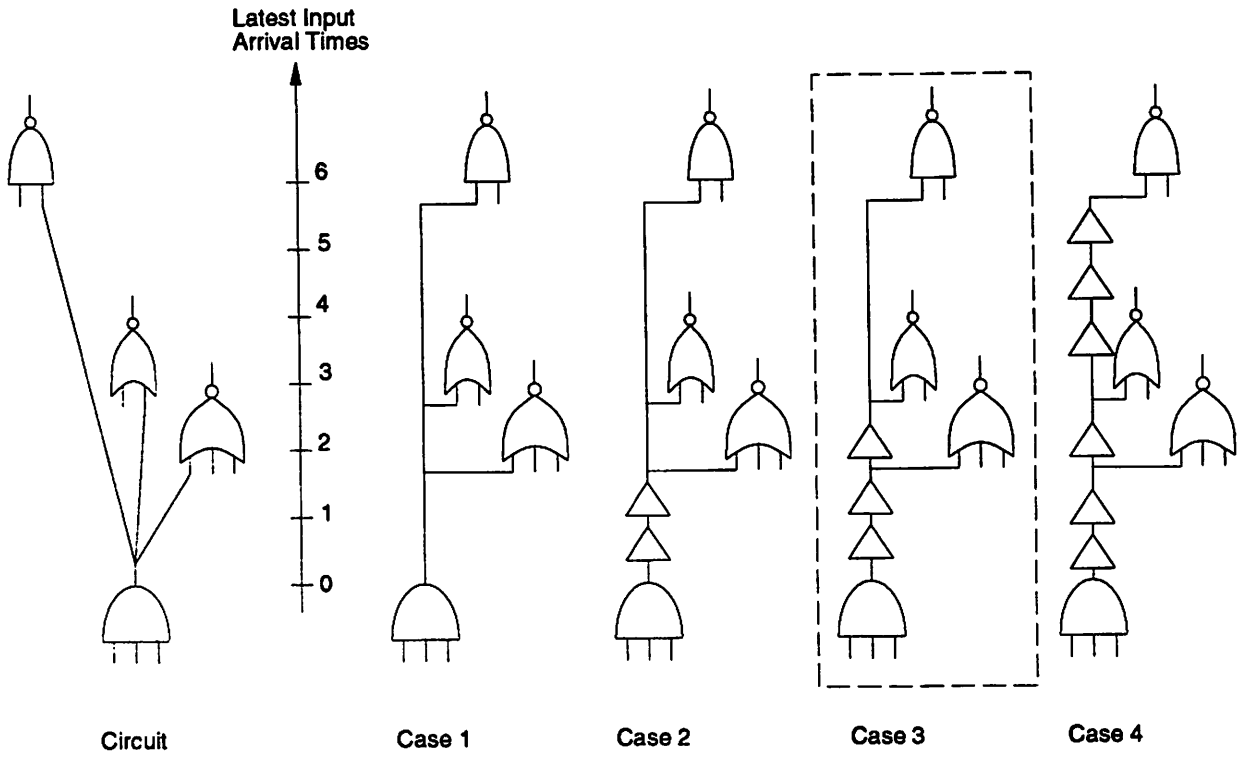

Figure 4.8: Finding the Optimal Buffer String

insertion and this does not influence the accuracy of earlier decisions.

The program based upon this algorithm was run on several networks from the ISCAS '89 benchmark set [25]. The networks were first synthesized for low-power using a technique similar to that presented by A. Shen et al. [20]. The buffer insertion experiment was then run on these low-power synthesized circuits, and these results are presented in Tab. 4.1.

The four experiments presented in Tab. 4.1 are: **Area Opt.** for an area optimized circuit, **Power Opt.** for a network synthesized for low-power, **Delay Insert.** for the buffer insertion to reduce spurious activity, and **Delay/Area** for buffer insertion under area constraints. In this final case, the area constraint is that the percentage increase in circuit area as a consequence of buffer insertion has to be less than four times the percentage decrease in power. The circuit area is measured in terms of the literal count, the power is in μW for a supply of 5V and clocking frequency of 20MHz. The power and area results of the **Power Opt.** experiment is scaled relative to the area optimized circuit, the results of the **Delay Insert.** and **Delay/Area** experiments are scaled relative to the power optimized circuit.

	Area Opt.		Power Opt.		Delay Insert.		Delay / Area	
Ckt.	Area	Pow.	Area	Pow.	Area	Pow.	Area	Pow.
s27	18	54	1	0.88	1.11	0.99	1	1
s344	203	800	1.01	0.89	1.27	0.89	1.20	0.89
s382	216	855	0.99	0.89	1.32	0.96	1	1
s420	247	922	1.03	0.76	1.28	0.98	1.02	0.97
s444	210	1061	0.99	0.71	1.30	0.95	1	1
s510	377	1606	1.01	0.80	1.42	0.95	1.19	0.95
s526	322	1571	1.03	0.75	1.28	0.97	1.11	0.98
s641	243	1056	1.07	0.89	1.30	0.90	1	1
s713	240	1130	1.03	0.80	1.27	0.91	1.19	0.93
s820	465	1895	0.99	0.73	1.41	0.92	1.20	0.94

Table 4.1: Optimization with Unit Delay Model Power Estimation

The results show that the FIDS estimation model is able to predict the expected change in power relative to changes in delay. This is demonstrated by the ability of the buffer insertion routine to select balancing strategies which ensure that the global reductions in power more than offset the local increases. It should be noted that the insertion of buffers to fully balance all paths increased network power between 10 to 20% in all cases. The operation of the algorithm within the constraints of a power/area tradeoff further illustrates that the method is able to make accurate predictions for unit delay model power estimation. The large reduction in area for the same power gain in the majority of cases shows that a significant percentage of balance choices made without an area constraint have global reductions in power almost identical to the power consumption of the inserted buffers. In the four cases where no improvement in power can be made under the area constraint, it is clear that each original balance condition only contributed a marginally power saving. Consequently, the area constraint makes all possible balance conditions nonviable.

The program was modified to operate on circuits with arbitrary delay. However, no balancing occurred for any benchmark example tested. In Sec. 4.2.3 this is explained in a consideration of the accuracy of the Functionally Correlated Delay Sensitive estimation technique. The balancing effect observed for unit delay networks was strongly dependent upon two properties not present under the arbitrary delay model assumption. These properties are:

1. *Coarse Time Discretization.* This increases the probability that transitions are estimated to occur simultaneously so exaggerates the reductions possible in generated spurious dynamic activity.

2. *No Fanout Loading.* This increases the probability that the balancing of the input to one gate also balances an input to another. For the arbitrary delay model, the loading effect can vary gate delays significantly making this multiple-balancing effect less probable.

4.2.3 The Functionally-Correlated Delay-Sensitive (FCDS) Model

The problem with the estimation strategy of the previous subsection is that is does not take into account the correlation between a node input n_i and the sensitivity of the node to that input. Consider the example of Fig. 4.9.

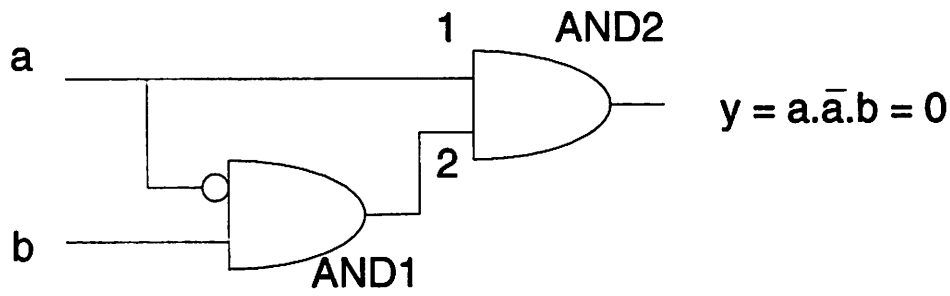

Figure 4.9: The Correlation of an Input to its Sensitivity

Assume that all gates have unit delay, and that a and b are primary inputs on which transitions arrive at time $t = 0$. A glitch will be generated at the output of AND2 under the conditions where there is a transition from 0 to 1 on input a which is propagated through the gate. The logical function of the output is $ba\bar{a} = 0$, so any propagation of a transition from input a is spurious. The *probability* that input $AND2_2$ is 1 is $\frac{1}{4}$. Therefore, $Pr(S_{AND2}(AND2_1)) = \frac{1}{4}$. However, under the conditions where $a = 0$, the output of AND1 is 1 iff $b = 1$. Hence, under the conditions of a positive transition on input a, the probability of erroneous propagation is actually $\frac{1}{2}$. The estimation scheme of this subsection takes into account this correlation.

Theory for Gates with Two Inputs Changing

In order to construct a formulation for the probability of generated activity which

has correct conditional probabilities, it is necessary to itemize the elements which define Generation. Assume that the two inputs to node n which change are n_j and n_k. Input n_j generates a spurious output transition if:

1. There is a functional transition on n_j.
2. There is a functional transition on n_k.
3. The functional transition on n_j arrives earlier than that on n_k.
4. The transition on n_j is transmitted through the node.
5. The result of the transmission of the transition on n_j opposes the final state of the output.

The estimation scheme of the previous subsection correctly took into account effects (1), (2) and (3), but assumed independence in the estimation of (4). To account for the conditional probabilities, the estimate is broken down into the four different cases:

$$\begin{aligned} f_{n_j}.f_{n_k} &\rightarrow \overline{f_{n_j}}.\overline{f_{n_k}} \\ f_{n_j}.\overline{f_{n_k}} &\rightarrow \overline{f_{n_j}}.f_{n_k} \\ \overline{f_{n_j}}.\overline{f_{n_k}} &\rightarrow f_{n_j}.f_{n_k} \\ \overline{f_{n_j}}.f_{n_k} &\rightarrow f_{n_j}.\overline{f_{n_k}} \end{aligned}$$

The first of these two cases give rise to the following formulation:

$$\begin{aligned} P_n^{g\downarrow}(n_j) = \quad & Pr(f_{n_k} \mid f_{n_j}). \quad \{Pr(S_n^p(n_j) \mid f_{n_j}.f_{n_k}).Pr(f_n \mid S_n^p(n_j)|_{n_k}.\overline{f_{n_j}}.\overline{f_{n_k}}) \\ & \quad +Pr(S_n^n(n_j) \mid f_{n_j}.f_{n_k}).Pr(\overline{f_n} \mid S_n^n(n_j)|_{n_k}.\overline{f_{n_j}}.\overline{f_{n_k}})\} \\ & +Pr(\overline{f_{n_k}} \mid f_{n_j}). \quad \{Pr(S_n^p(n_j) \mid f_{n_j}.\overline{f_{n_k}}).Pr(f_n \mid S_n^p(n_j)|_{\overline{n_k}}.\overline{f_{n_j}}.f_{n_k}) \\ & \quad +Pr(S_n^n(n_j) \mid f_{n_j}.\overline{f_{n_k}}).Pr(\overline{f_n} \mid S_n^n(n_j)|_{\overline{n_k}}.\overline{f_{n_j}}.f_{n_k})\} \end{aligned} \tag{4.8}$$

From the latter two cases, the following is derived:

$$\begin{aligned} P_n^{g\uparrow}(n_j) = \quad & Pr(f_{n_k} \mid \overline{f_{n_j}}). \quad \{Pr(S_n^p(n_j) \mid \overline{f_{n_j}}.f_{n_k}).Pr(\overline{f_n} \mid S_n^p(n_j)|_{n_k}.f_{n_j}.\overline{f_{n_k}}) \\ & \quad +Pr(S_n^n(n_j) \mid \overline{f_{n_j}}.f_{n_k}).Pr(f_n \mid S_n^n(n_j)|_{n_k}.f_{n_j}.\overline{f_{n_k}})\} \\ & +Pr(\overline{f_{n_k}} \mid \overline{f_{n_j}}). \quad \{Pr(S_n^p(n_j) \mid \overline{f_{n_j}}.\overline{f_{n_k}}).Pr(\overline{f_n} \mid S_n^p(n_j)|_{\overline{n_k}}.f_{n_j}.f_{n_k}) \\ & \quad +Pr(S_n^n(n_j) \mid \overline{f_{n_j}}.\overline{f_{n_k}}).Pr(f_n \mid S_n^n(n_j)|_{\overline{n_k}}.f_{n_j}.f_{n_k})\} \end{aligned} \tag{4.9}$$

To explain this expression, consider an example term, in particular:

$$Pr(f_{n_k} \mid f_{n_j}).Pr(S_n^p(n_j) \mid f_{n_j}.f_{n_k}).Pr(f_n \mid S_n^p(n_j)|_{n_k}.\overline{f_{n_j}}.\overline{f_{n_k}})$$

$Pr(f_{n_k} \mid f_{n_j})$ is the probability that input n_k is one given that n_j is 1; $Pr(S_n^p(n_j) \mid f_{n_j}.f_{n_k})$ is the probability that the output is sensitive to input n_j given that n_k, n_j are originally one; and $Pr(f_n \mid S_n^p(n_j)|_{n_k}.\overline{f_{n_j}}.\overline{f_{n_k}})$ is a term contributed from effect (5) in the list enumerated above. For this particular example, this term is the probability that the node output settles to logic 1 given that n_j and n_k settle to logic 0, and that the functional transition on input n_j was propagated through the gate as a negative output transition. These definitions for $P_n^{g\uparrow}(n_j)$ and $P_n^{g\downarrow}(n_j)$ can be substituted directly back into Eqn. 4.4 and the subsequent formulae.

Generalizing to Gates with Arbitrary Input Changes

The technique for the generalization of the two-input gate theory to multiple input gates is identical to that presented in Sec. 4.2.1 with $P_n^{g\uparrow}(n_j)$ and $P_n^{g\downarrow}(n_j)$ replaced by the more accurate formulations presented in this section.

4.2.4 Accuracy of the Functionally-Correlated Delay-Sensitive Model

The FCDS model was tested for absolute accuracy and also for its ability to determine sensitivity to changes in network delays. The test of absolute accuracy was performed by comparing the results of exact simulation of a network to the predictions of the estimator. For each node in the network, the node output activity was estimated given the node input activities and their delay bounds. The sensitivity to delay is tested by balancing the node input arrival times to the latest input arrival time. After balancing, the network is simulated again and the total change in power is compared to that estimated by the FCDS method. Both tests were performed on networks from the ISCAS '89 benchmark set, first optimized with `script.rugged`[19] and mapped to the `msu.genlib` gate library. The arbitrary gate delay model is used for the power simulation.

The absolute accuracy of the FCDS model is presented in Fig. 4.10. This graph is a plot of estimated spurious dynamic activity against the actual spurious dynamic activity for every node in the tested networks. The estimator is well correlated to simulation with a correlation coefficient of 0.98. It is worth noticing that the correlation is better when the amount of spurious dynamic activity is small. The increase in error is a result of the

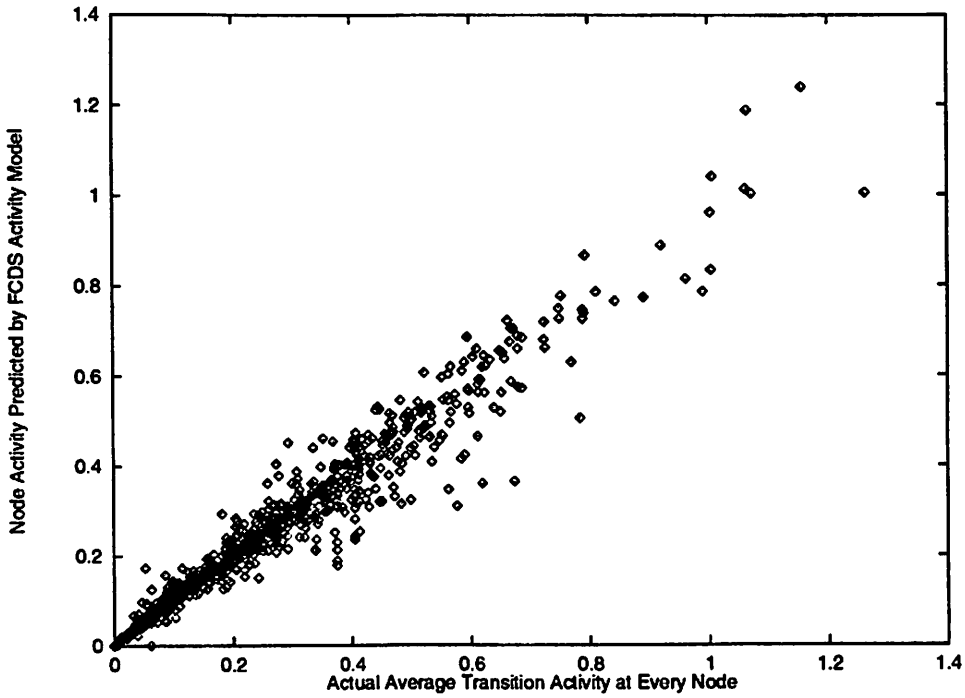

Figure 4.10: The Accuracy of the FCDS Method: Estimate vs. Actual Activity

estimate becoming dominated by transmitted activity when spurious dynamic activity is large. The Spurious Transmission is not as accurate an estimate as Generation due to the fact that it assumes independence between the spurious input activity and other node inputs.

In Fig. 4.11, the proportion of the spurious dynamic activity at a node which is Generation is plotted against the total amount of spurious dynamic activity. Fig. 4.12 is a running average of the same data computed with a minimum step size of 0.01, and minimum data count per average of five points. In the case where spurious dynamic activity exceeds 0.25, less than half of this activity is accounted for by Generation effects.

The results of the balancing test are depicted in Fig. 4.13 . The same results are presented in Fig. 4.14 with all the trivial balancing cases removed. (A trivial balancing case is one in which the input activity intervals are of zero width so that node balancing reduces the node output spurious dynamic activity to zero.) In both cases, the estimator predicted change in activity is plotted against the actual change in spurious dynamic activity. From these graphs it is clear that an estimator based upon the concepts of Spurious Transmission and Generation does not have sufficient accuracy to predict changes in activity with changes

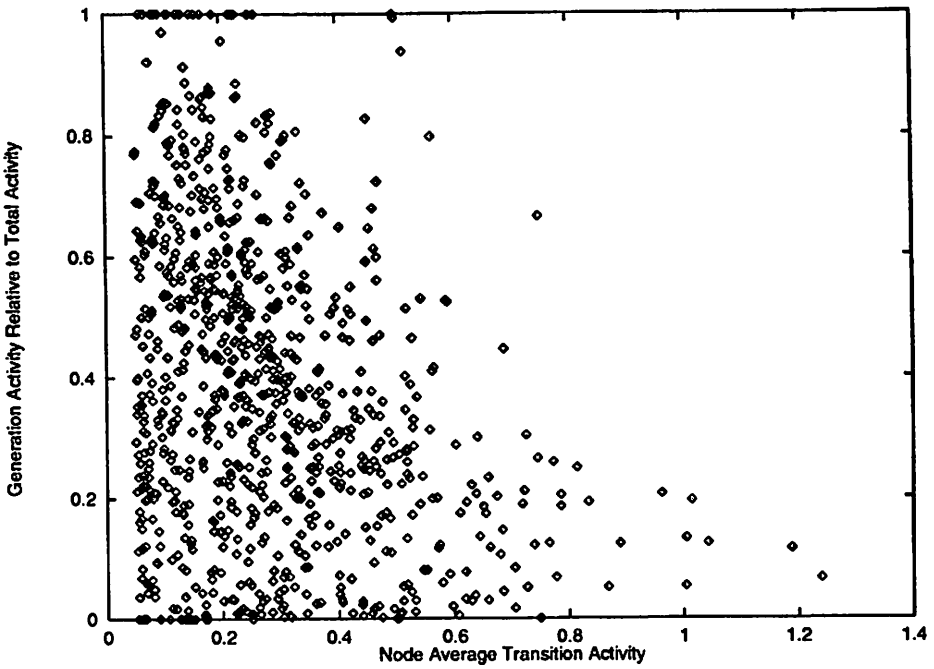

Figure 4.11: Generation Relative to Total Spurious Activity

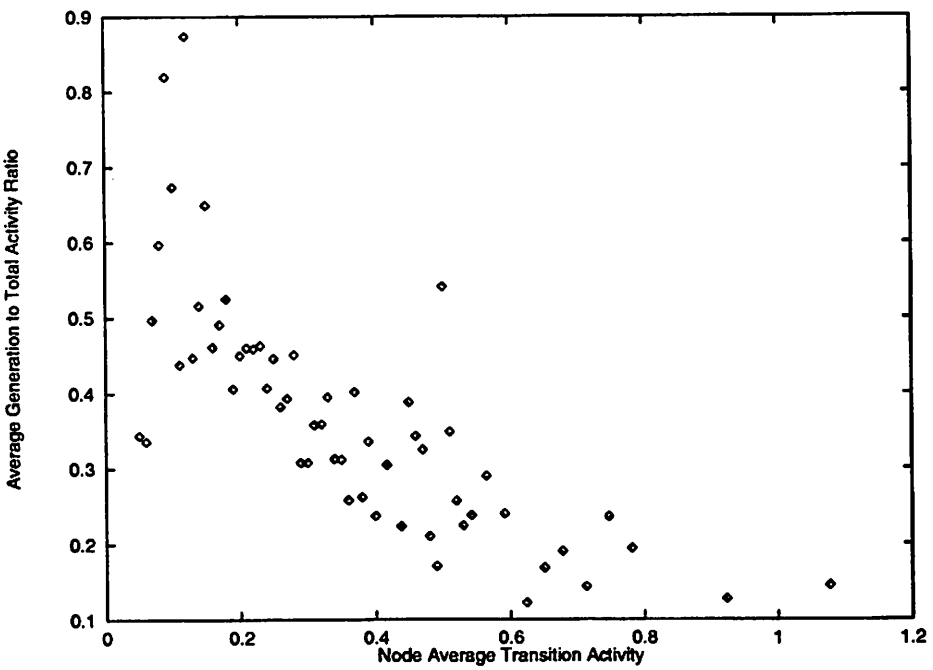

Figure 4.12: Average Generation Relative to Total Spurious Activity

in delay despite the overall estimator accuracy depicted in Fig. 4.10.

In Fig. 4.14, there are significant clusters of points on or near both the x- and y-axes. The cluster of points along the x-axis are conditions which the estimator fails to predict. The independence assumption associated with the estimate of Spurious Transmission accounts for this lack of identification of optimality conditions. To improve the ability to estimate these changes, it is necessary to have information about the *functionality* of the spurious dynamic activity. The points clustered on the y-axis are conditions which the estimator predicts but which don't actually occur. Generation is the only part of the FCDS estimate affected by delay. Hence, the over prediction of changes in activity is a consequence of error in the Generation estimate, and this error therefore implies that the uniform distribution model within activity bounds is insufficient.

The results of these tests show that sensitivity to delay cannot be predicted in the absence of:

- An accurate estimate of the activity distribution
- An estimate of the functionality of the spurious dynamic activity

These properties cannot be predicted effectively prior to the resynthesis of a region. It is not possible therefore to determine the sensitivity of global network power to the output delay of a network region being resynthesized. The sensitivity to delay under the conditions where an estimate of the switching activity distribution and functionality can be obtained is the focus of Chap. 5.

4.3 Sensitivity to Dynamic Activity

Sensitivity to dynamic activity is the sensitivity of the network to changes in total activity at a node, neglecting delay effects. The estimation technique of the previous section incorporates terms for both a change in functional and spurious dynamic activity at a node inputs. However, the majority of the complexity of the estimate is associated with the computation of spurious Generation. The empirical results have shown that the concept of Generation does not adequately account for delay effects in the absence of accurate switching distribution estimates. Consequently, this term can be removed from the estimation technique without affecting accuracy with respect to modeling Dynamic Sensitivity. During resynthesis, changes in the network are incremental so statistical properties

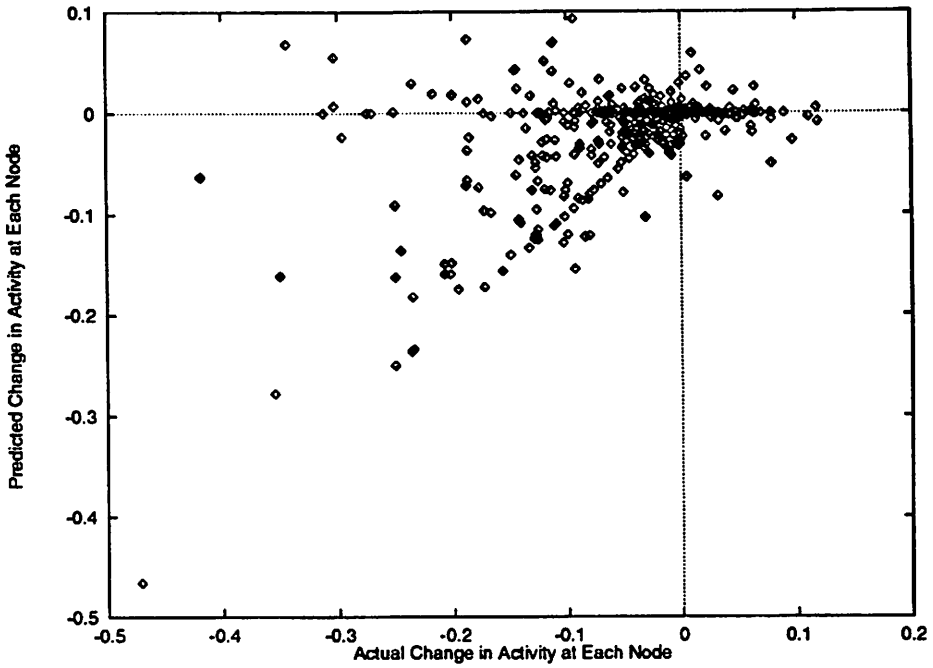

Figure 4.13: Estimating the Change in Spurious Activity after Balancing

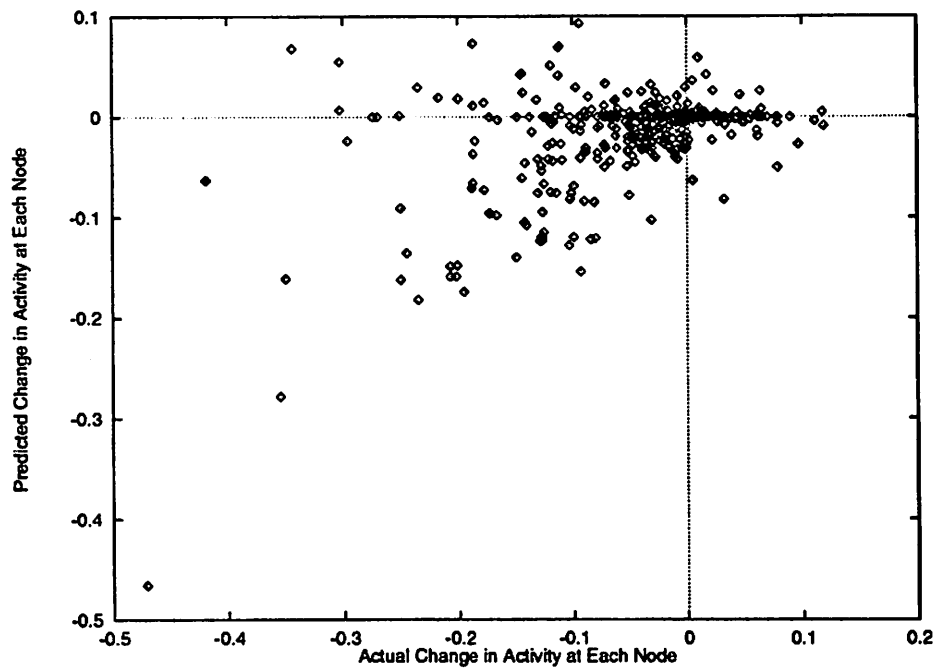

Figure 4.14: Estimating Balancing Effects - Trivial Cases Removed

relating the input activity to the output spurious dynamic activity at any specific node are unlikely to change significantly. The original state of the network can then be used to establish how a change in spurious dynamic activity will globally affect network power dissipation.

4.3.1 The Delay-Insensitive (DI) Model

An example of a very elementary estimator for activity at the output of node n with input activities $\{T_{n_i}^T\}$ is:

$$|T_n^T(est)| = \sum_{n_i \in \Lambda_n} Pr(S_n(n_i)).|T_{n_i}^T| \tag{4.10}$$

The general problem with this approximation is that it neglects correlations between the inputs. However, from an initial network simulation a ratio can be computed which relates the actual spurious dynamic output activity to the input activities. A change in spurious dynamic activity at an input to a node is assumed independent of the other inputs. Hence, the sensitivities, $S_n(n_i)$, may be viewed as a measure of how significant a change in the dynamic activity of input n_i is to a change in the spurious dynamic output activity.

For the estimation strategy of Eqn. 4.10, the ratio of original output spurious activity to input dynamic activity is:

$$R_n^D = \frac{|T_n^D(orig)|}{(\sum_{n_i \in \Lambda_n} Pr(S_n(n_i)).|T_{n_i}^T(orig)|} \tag{4.11}$$

Establishing the ratio of the *spurious* output activity to the estimate of Eqn. 4.10 forms a distinction between functional and spurious dynamic output activity estimation. This allows the result of the estimation strategy of Chap. 3 to be coupled with the spurious dynamic estimate. In this way, a good approximation of functional correlation is maintained. To estimate a change in activity at the output of a node given a change in the spurious dynamic activity at the inputs, the ratio R_n^D is assumed constant giving:

$$\delta(|T_n^D|) = R_n^D.(\sum_{n_i \in \Lambda_n} Pr(S_n(n_i)).\delta(|T_{n_i}^T|)) \tag{4.12}$$

4.3.2 Accuracy of the Delay-Insensitive (DI) Model

The accuracy of the DI model was examined for the same circuits as the test for the FCDS model. This test is actually performed on the estimation technique for overall activity presented in Eqn. 4.10. Note that the the DI model uses scaling, so its true accuracy is only defined with its ability to predict *changes* in activity, not absolute values (See Sec. 4.5). However, the accuracy of Eqn. 4.10 in predicting total spurious activity is still interesting. This test demonstrates the validity of the assumption that node sensitivity can be used as measure of the statistical significance.

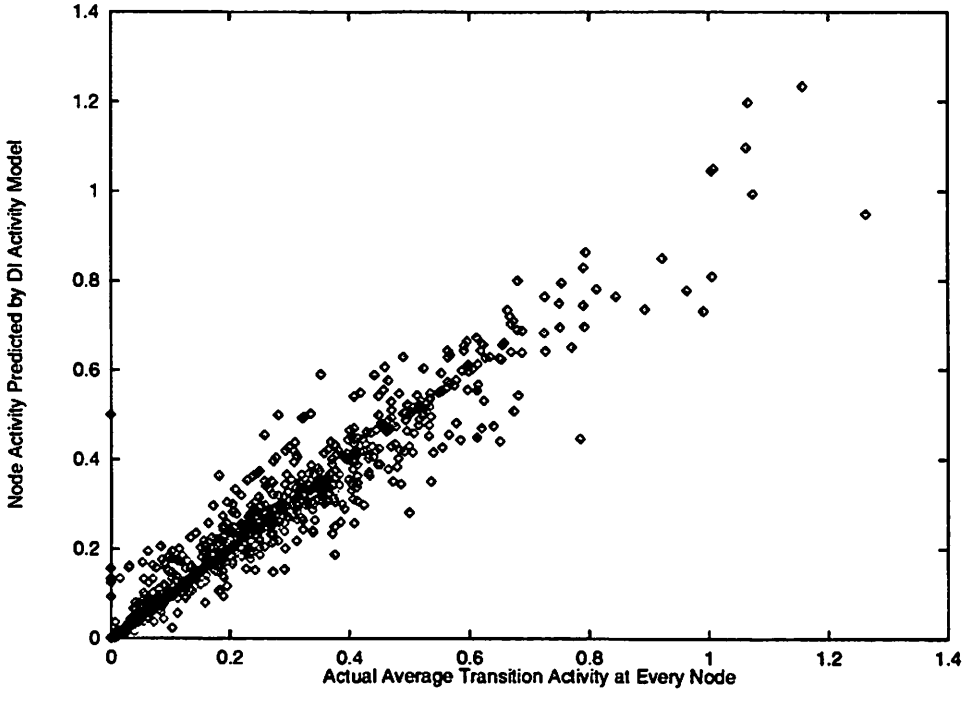

Figure 4.15: The Accuracy of the DI Method

The results of this test are shown in Fig. 4.15. Compared with Fig. 4.10, it is observed that for large spurious dynamic activity the FCDS and DI techniques have similar accuracy. This arises from the fact that the two approximation methods become almost equivalent when Spurious Transmission dominates. This occurs for spurious dynamic activities greater than 0.2 (Fig. 4.12).The results of this test suggest that the accuracy of the DI method in predicting changes in spurious dynamic activity will be limited when spurious dynamic activity is small, but it will improve as the spurious dynamic activity increases.

4.4 Sensitivity to Functional Spurious Activity

Sensitivity to functional spurious activity involves estimating how node sensitivities are affected by expected change in input functionality. All of the terms in the approximation schemes of Sec. 4.2.3 and Sec. 4.3.1 can be estimated using the techniques of Chap. 3. The function describing each probability in these expressions has the same inputs as the original node. The sensitivities of these functions with respect to each input may be established, allowing the expected change in function to a change in input onset probability to be computed.

The problem with such an approach is that each term is averaged independently. The *correct* expected change in output activity is derived from the average of the product, not the product of the averages. Taking into account the correlation between the individual terms of the expressions to compute the average of the product is extremely difficult. Consequently, the product of the averages is used as an approximation. The empirical results show that this simplifying approximation makes the FCDS model less accurate at estimating Functional Spurious Sensitivity and Dynamic Sensitivity than the DI model. Increasing the complexity of analysis used in the FCDS model could only improve accuracy for the small measures of spurious dynamic activity for which the FCDS technique is more accurate than the DI estimate. There is little to be gained by such a pursuit.

The DI model has a very simple first-order approximation to the combination of Functional Spurious and Dynamic Sensitivity expressed in the following equation:

$$\delta(|T_n^D|) = R_n^D.(\sum_{n_i \in \Lambda_n} Pr(S_n(n_i)).\delta(|T_{n_i}^T|) + \delta(Pr(S_n(n_i))).|T_{n_i}^T(orig)|\,) \qquad (4.13)$$

4.5 Overall Sensitivity Results

The results of this section demonstrate the accuracy of estimating changes in power consumption using Functional Spurious and Dynamic Activity Sensitivities. The program to generate test cases is the same as that outlined in Chap. 3 and the experiment is run on the same circuits from the ISCAS '89 benchmark set. Following the selection of a random change in function of a network node, this node is resynthesized locally using `script.rugged` and mapped into the `msu.genlib` library. This resynthesis and remapping changes delay, function and topology of the affected region. This provides a suitable platform for testing of

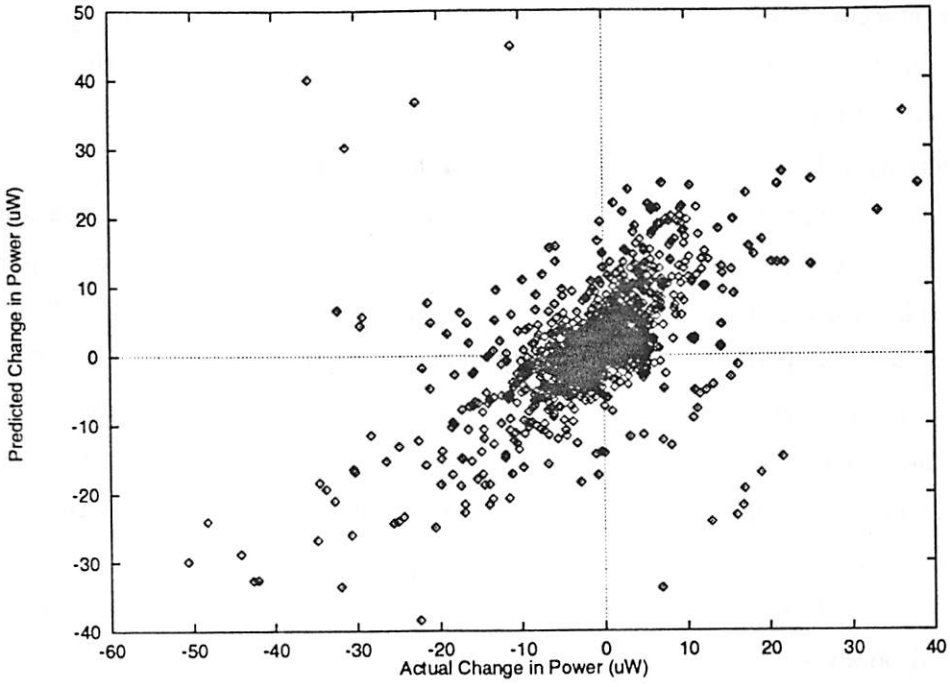

Figure 4.16: Spurious Dynamic Activity Estimation without Functional Spurious Sensitivity

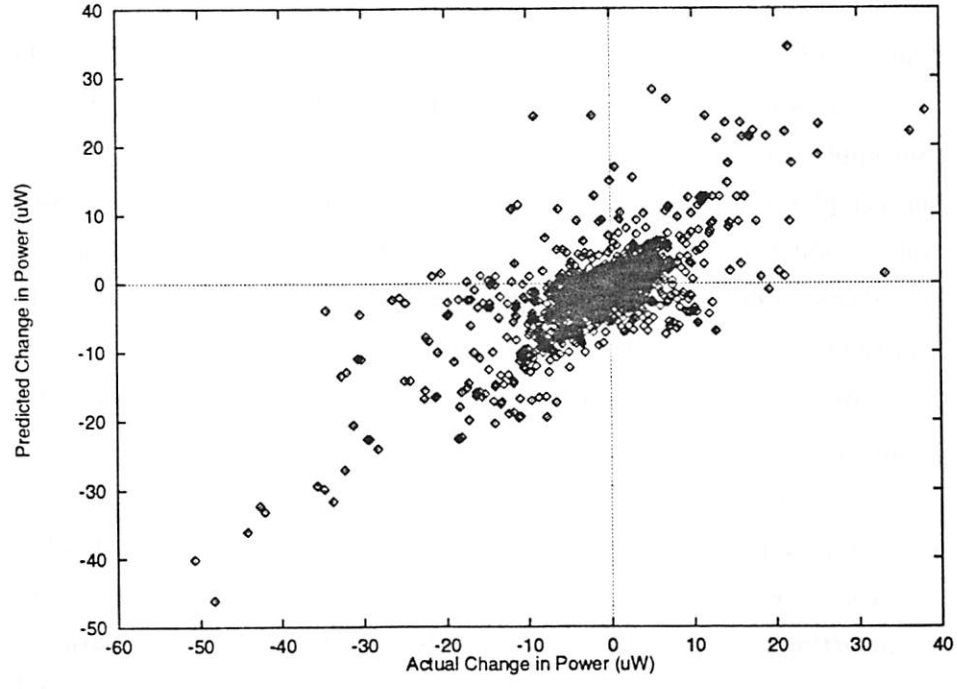

Figure 4.17: Spurious Dynamic Activity Estimation without Dynamic Activity Sensitivity

the accuracy of the estimation techniques of this chapter. The changes in power measured from full simulation are compared against those predicted by the estimation techniques. The scale of the graphs presented here is normalized to a 20MHz input vector arrival frequency, supply voltage of 5V. A unit on the graph corresponds to $1\mu W$.

Fig. 4.16 is a graph of the results of the expected change spurious dynamic TFO power against actual change, but without accounting for Functional Spurious Sensitivity. (This is using the DI model without updating the $Pr(S_n(n_i))$ terms.) The results show that without estimation of the change in functionality throughout the TFO, Dynamic Sensitivity alone is a very poor estimation strategy. The correlation coefficient is only 0.56. In Fig. 4.17 the results of the DI estimator are presented for Functional Spurious Sensitivity alone, Dynamic Sensitivity neglected. Again the correlation is poor, this time with a correlation coefficient of 0.73.

In Fig. 4.18 and Fig. 4.19 the results of the FCDS and DI models respectively with full analysis of all sensitivities modeled by each scheme is shown. (Fig. 4.18 shows results from an earlier test which ran fewer iterations than the final testing of the more accurate DI model.) The problem with neglecting the correlation between the sensitivities of the more complex FCDS model is immediately obvious. The correlation coefficient for this model is only 0.67, whereas the DI scheme exhibits a correlation of 0.93. This is a clear example of how application of simple approximation techniques to a complex model can actually make the complex model less accurate for incremental estimation. In this case, the error of the simple model in predicting change is suitably accurate for use in a synthesis routine. As sensitivity to delay has been shown to depend upon both an accurate timing distribution, and the functionality of the spurious dynamic activity, a spurious activity estimator for use in guiding synthesis routines cannot be expected to improve significantly beyond the error obtained with the DI model.

The contents of Tab. 4.2 are the results of estimator correlation for the DI model on each individual circuit examined. The column headed **Old Sens. Correl.** gives the estimator correlation when Functional Spurious Sensitivity is neglected, **Est. Sens. Correl.** is the estimator correlation with this effect approximated. Estimation without Functional Spurious Sensitivity can produce correlations of less than 0.4 for a number of circuits, but full estimation always produces a correlation of better than 0.9 . The accuracy of the full estimation strategy is not strongly influenced by either the accuracy of the estimation without Functional Spurious Sensitivity accounted for, or the size of the circuit.

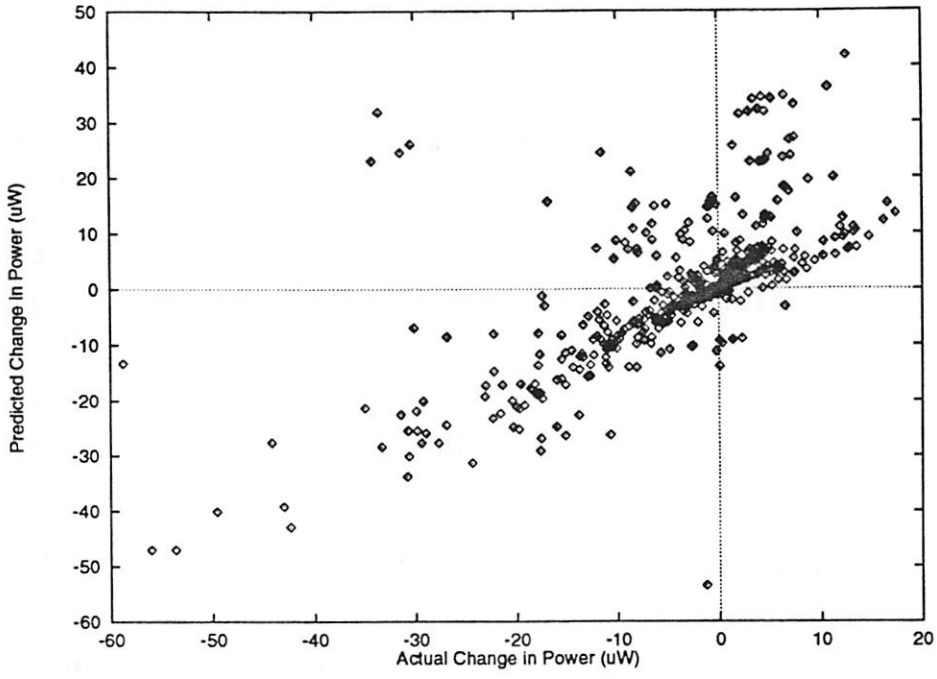

Figure 4.18: Spurious Dynamic Activity Estimation with the Full FCDS Model

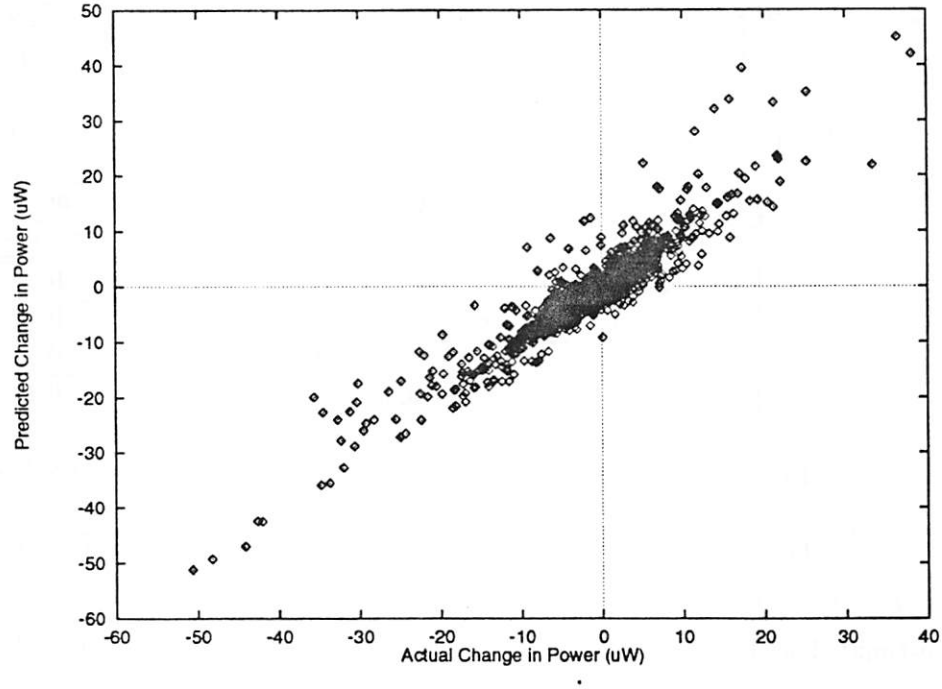

Figure 4.19: Spurious Dynamic Activity Estimation with the Full DI Model

Ckt.	Old Sens. Correl.	Est. Sens. Correl.	Ckt.	Old Sens. Correl.	Est. Sens. Correl.
s344	0.72	0.91	s820	0.91	0.97
s386	0.79	0.92	s832	0.20	0.98
s400	0.81	0.93	s838	0.83	0.90
s444	0.31	0.97	s1196	0.63	0.90
s510	0.39	0.92	s1238	0.52	0.98
s526	0.61	0.98	s1488	0.68	0.91
s641	0.39	0.94	s1494	0.70	0.92
s713	0.65	0.95			

Table 4.2: Estimate Correlation for Change in Sp. Dynamic TFO Power

Power Change		Occurrence		Estimation	
LOCAL	GLOBAL	Actual	Est.	Detected	Incorrect
Decrease	Increase	0.13	0.11	0.57	0.32
Decrease	$\leq -0.5x$	0.09	0.07	0.58	0.30
Increase	Decrease	0.11	0.043	0.37	0.02
Increase	$\leq -0.5x$	0.08	0.02	0.24	0.04

Table 4.3: Total Activity Estimation without Node Sensitivities Prediction

Power Change		Occurrence		Estimation	
LOCAL	GLOBAL	Actual	Est.	Detected	Incorrect
Decrease	Increase	0.13	0.13	0.79	0.20
Decrease	$\leq -0.5x$	0.09	0.08	0.78	0.17
Increase	Decrease	0.11	0.12	0.86	0.15
Increase	$\leq -0.5x$	0.08	0.08	0.89	0.11

Table 4.4: Total Activity Estimation with Node Sensitivities Prediction

The results of Chap. 3 are combined with those of the DI model in Fig. 4.20 and Fig. 4.21. The first of these two figures shows the total change in power consumption estimated with Functional Spurious Sensitivity ignored, and the second includes this sensitivity. Estimation of Functional Spurious Sensitivity improves the estimate correlation from 0.82 to 0.97. The ability of the estimator to detect overall optimality conditions is presented in Tab. 4.3 and Tab. 4.4. The improvement in the results of Tab. 4.4 over Tab. 4.3

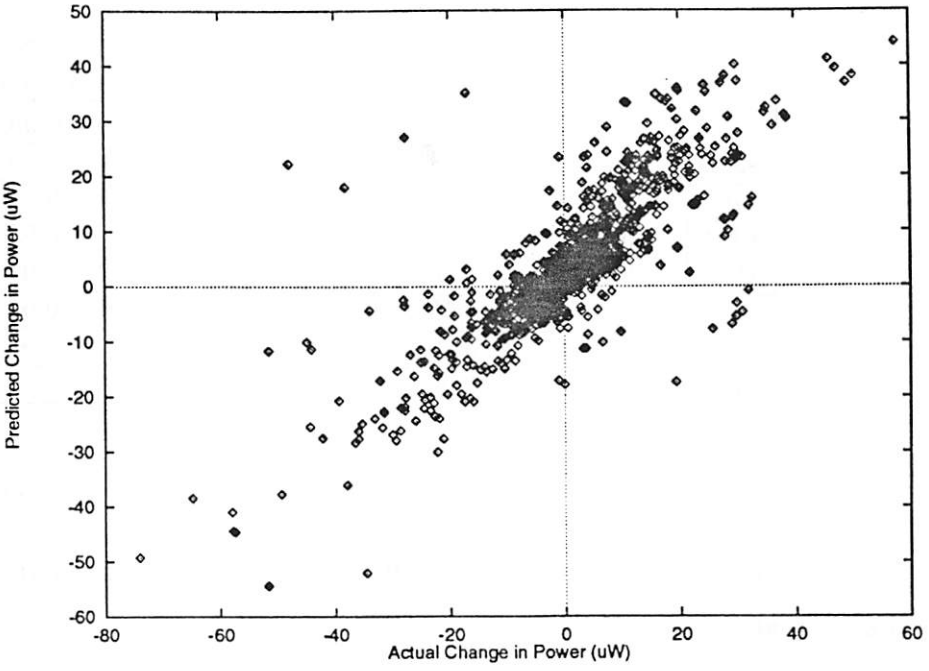

Figure 4.20: Change in Overall Power without Sensitivity Prediction

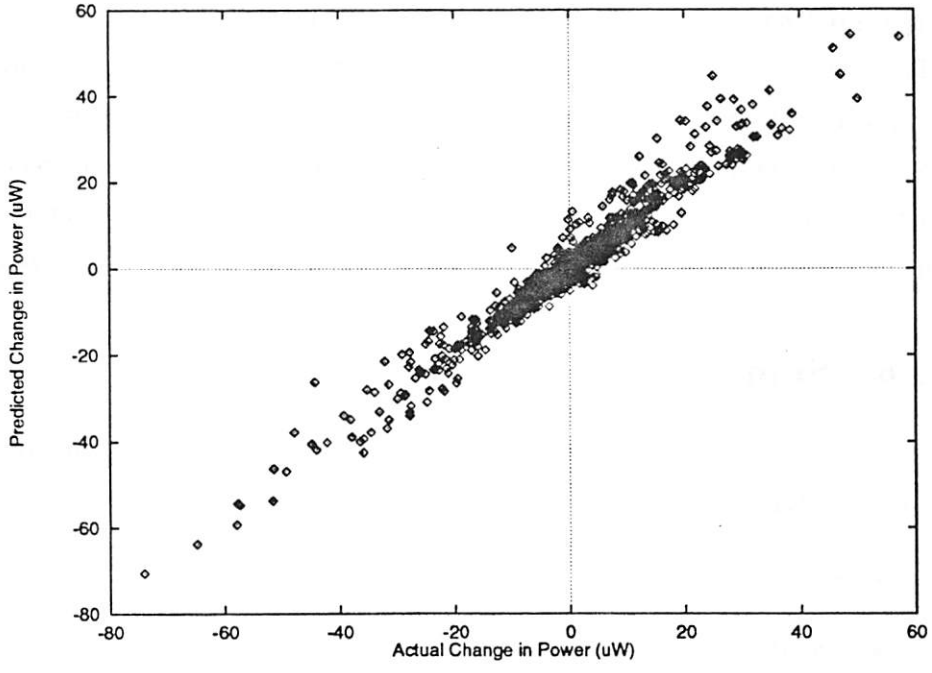

Figure 4.21: Change in Overall Power with Sensitivity Prediction

Ckt.	Old Sens. Correl.	Est. Sens. Correl.	Ckt.	Old Sens. Correl.	Est. Sens. Correl.
s344	0.84	0.96	s820	0.94	0.99
s386	0.93	0.98	s832	0.73	0.99
s400	0.91	0.96	s838	0.93	0.97
s444	0.68	0.98	s1196	0.87	0.93
s510	0.62	0.96	s1238	0.85	0.99
s526	0.70	0.99	s1488	0.92	0.97
s641	0.87	0.99	s1494	0.91	0.98
s713	0.96	0.99			

Table 4.5: Estimate Correlation for Change in Total TFO Power

is a result of Functional Spurious Sensitivity estimation. Tab. 4.5 contains results of estimator correlation for each circuit tested.

The results of Tab. 4.4 show that the optimality predictions of the overall activity estimator are only about 10% less accurate than those of the functional activity estimator results of Chap. 3. This is a good result given extremely complex phenomena being modeled. The importance of modeling the overall activity change is made clear by a comparison of Fig. 3.11 with Fig. 4.21 which indicates that the spurious dynamic activity can alter the overall change in power consumption by a factor of three or more relative to functional activity change alone. From Tab. 4.5 it is observed that the accuracy of the estimator including sensitivity prediction does not appear to be related to the size of the circuit.

4.6 Summary

In this chapter, three concepts important to the influence of resynthesis upon spurious dynamic network power have been isolated. These three concepts are:

- Sensitivity to Delay
- Sensitivity to Dynamic Activity
- Sensitivity to Functionality (Functional Spurious Sensitivity)

Prior to synthesis, the only estimate of these quantities which can be made is bounds on delay, the amount of spurious dynamic activity (as opposed to knowing its precise function-

Chapter 5

Dynamic Activity Estimation: Complete Activity Distribution and Functionality

In this chapter, the ability to predict changes in activity due to changes in delay is addressed in detail. Chap. 4 includes an analysis of this property in the context of simple delay bounds which might be used to guide resynthesis. The object of the material in this chapter is the development of theory and empirical evidence which relates the ability to determine delay sensitivity to the necessity for accurate estimation of transition distributions and spurious transition functionality.

The theory presented in this chapter applies to routines which perform changes in delay with minor or no changes in network functionality. This is the case during transistor resizing for reduced power consumption, and also in the prediction of the effects of change in fanin loading of a resynthesis region. The theory addressing this concept could also be used to guide resynthesis of a region if the resynthesis options are sufficiently limited to allow accurate prediction of output transition distributions and spurious functionality. It is shown in the results presented here that delay sensitivity is not a very accurate premise in the absence of exhaustive enumeration of the possible delay combinations. Without simulating every delay combination, the accuracy in predicting a change in activity cannot be improved beyond a correlation of 0.8.

This chapter has three sections. In Sec. 5.1 several different activity types are

ality), and the expected change in the probability of node onsets (from Chap. 3).

Two estimation strategies have been proposed. The first uses a measure of sensitivity to delay, the second is independent of this quantity. The estimation scheme which is sensitive to delay incorporates the concept of simultaneous arrival for state transitions and their correlations. Consequently, this model is as accurate as possible under the constraints on the information indicated above. Empirical verification of this model has shown that sensitivity to delay cannot be formulated without accurate estimates of both the switching distribution and the functionality of the spurious dynamic activity.

The spurious dynamic activity model which is delay insensitive is based upon the fact that during resynthesis, network correlations are unlikely to change significantly. Consequently, properties from an initial simulation of the network are used to determine the expected statistical significance of incremental changes. Empirical tests show that changes predicted using this estimator correlate very well with simulation results. The examination of results for the Dynamic and Functional Spurious Sensitivity estimates alone demonstrate that both effects are almost of equal importance. Furthermore, the final accuracy of this model shows that sensitivity to delay is by far the least significant effect.

The combination of the delay insensitive estimation scheme with the results of the functional activity estimator is able to predict overall changes in network power very well. The correlation of overall estimate to experiment is 0.97.

isolated, and estimation techniques for those which can be predicted without full simulation are presented. This section also contains an outline of the form of the empirical testing strategy used for establishing the results in this chapter. In Sec. 5.2 and Sec. 5.3, the two effects which can be estimated without full simulation are addressed. The results for the different estimators are combined in Sec. 5.4, and summarized in Sec. 5.5.

5.1 Isolating Activity Types

There are three effects which determine the amount of spurious activity at the output of a node. These effects are:

- *Simultaneous Reduction.* If one input transition to a gate arrives simultaneously with another input transition, the number of output transitions is less than or equal to the number of transitions which occur if the transitions are not simultaneous.
- *Functional Transmission.* If a transition occurs on an input n_i to gate n at a time when all the other inputs have not transitioned or have settled to their final state, the transmission of the signal on input n_i can be computed from the previous and next state input variables.
- *Non-Functional Transmission.* If a transition on an input n_i to gate n occurs at a time when one or more other inputs have transitioned but have not yet settled, the transmission of the signal on input n_i may not be able to be computed from a combination of previous and next state input variables.

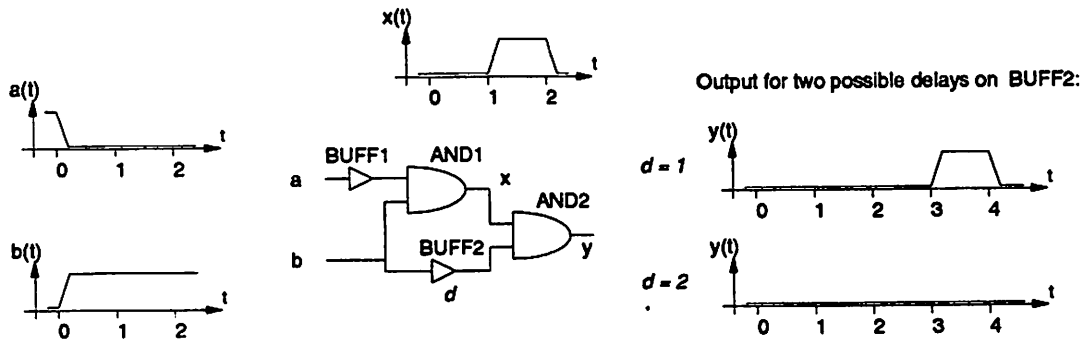

Figure 5.1: Example of Functional Transmission

The first of these effects is obvious and has been discussed in Chap. 4. The concept of Functional Transmission is illustrated by an example introduced in Chap. 4, duplicated here in Fig. 5.1. The generation of spurious dynamic activity on x uniquely defines input conditions on (a, b) as an initial state $(1, 0)$ and final state $(0, 1)$. This implies that if the other input to AND2 is correlated to the previous state of b when the glitch is produced on x, it cannot propagate. On the other hand, correlation to the next state of b guarantees that the glitch *will* propagate. Because the propagation of this activity depends solely upon the previous and next state functions of b, this is denoted *Functional Transmission.*

An example of Non-Functional Transmission is shown in Fig. 5.2 . In Case (i), spurious activity occurs on input a while the other input is also under the influence of unnecessary transitions. Consequently the output transitions even though the next and final state of both inputs is 0. In Cases (ii) and (iii), input b does actually have a state change. The propagation of the glitch on input a is correctly predicted *iff* the result is correlated against the correct state (next or previous) of input b. However, determining the correct state depends upon knowledge of the topology (timing, not just the function) of the fanin logic. Consequently, this is defined *Non-Functional Transmission* as it requires knowledge of the specific gate mapping of the network.

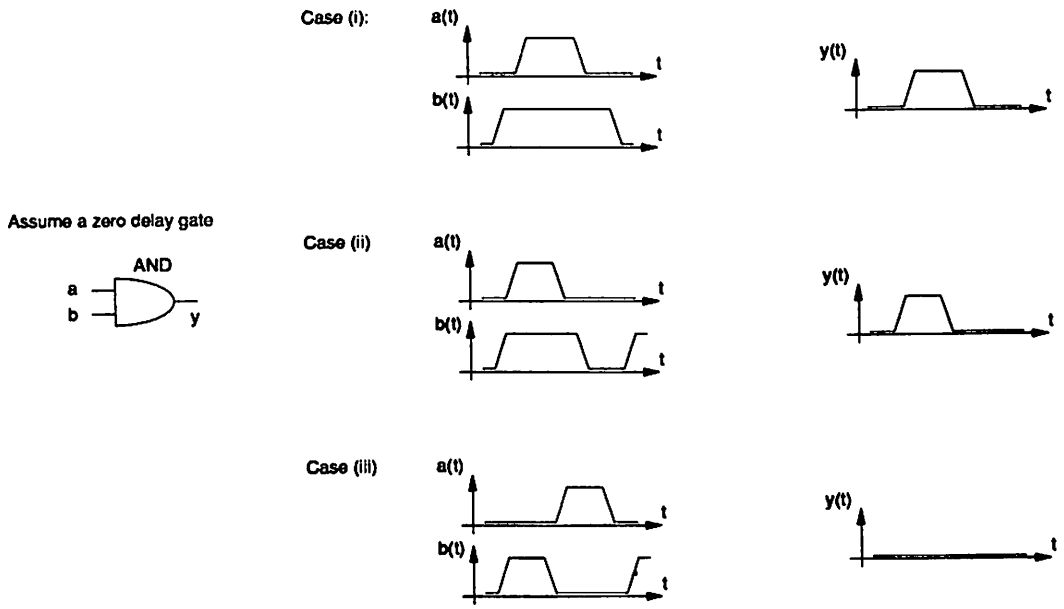

Figure 5.2: Example of Non-Functional Transmission

The reduction in activity due to Simultaneous Reduction must be separated from the Functional Transmission and Non-Functional Transmission. This can be achieved by defining a measure of activity *without* simultaneous arrival and then the difference between this measure and the actual activity is the Simultaneous Reduction (SR).

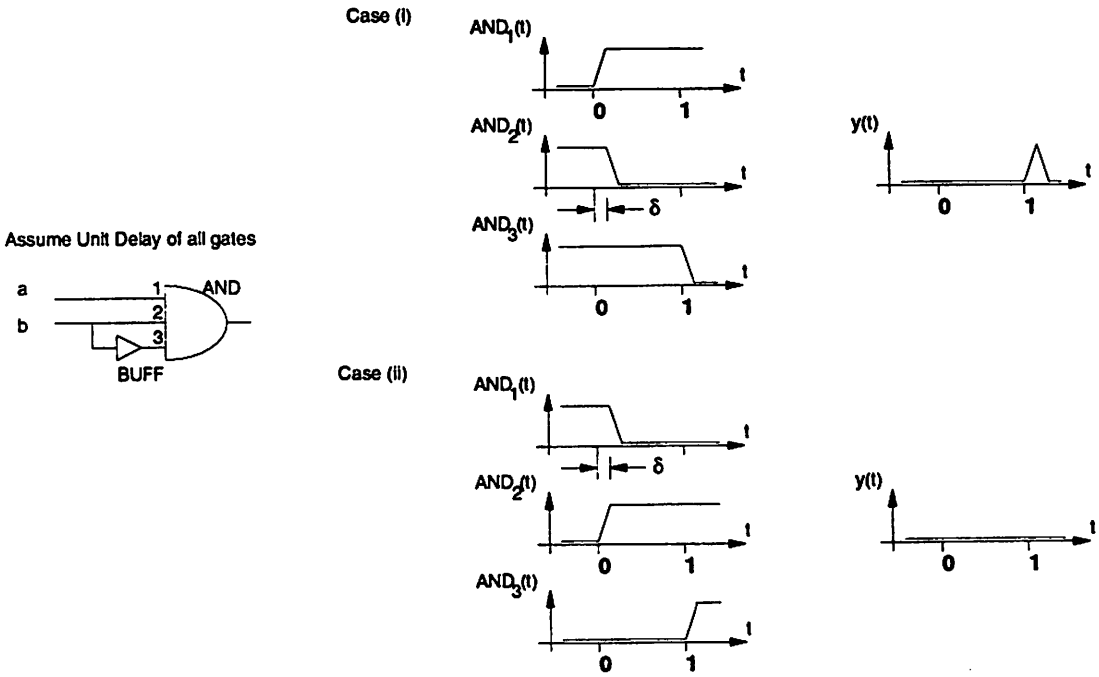

Figure 5.3: Defining Non-Simultaneous Activity and Simultaneous Reduction

Activity without simultaneous arrival (*Non-Simultaneous Activity* or NSA) can be defined through the application of an arbitrary arrival order to gate input signals which arrive at the same time. Consider the example of Fig. 5.3. Let a and b be independent inputs with $Pr(1) = 0.5$. From full simulation, the AND gate output has a functional transition with probability $2.\frac{1}{4}.\frac{3}{4} = \frac{3}{8}$, and has no spurious dynamic activity. Inputs AND_1 and AND_2 arrive simultaneously at time $t = 0$, input AND_3 arrives at time $t = 1$ (assuming unit delay). To estimate activity without Simultaneous Reduction, let the arrival times for AND gate inputs (1) and (2) be t_1 and t_2 respectively. If $t_1 < t_2$, when (a, b) changes state from $(0, 1)$ to $(1, 0)$, $AND_3 = 1$ so the glitch is seen at the output. The amount of amount of Non-Simultaneous Activity for this ordering is therefore: $\frac{3}{8}+2.\frac{1}{8} = \frac{5}{8}$, giving a Simultaneous Reduction of $\frac{1}{4}$. If, however, $t_1 > t_2$ then no spurious activity is generated and the Non-Simultaneous Activity is $\frac{3}{8}$ with Simultaneous Reduction of 0. It is shown through this

example that if an arbitrary arrival order for the simultaneous input transitions is assumed, Non-Simultaneous Activity is a function of this order. To form a input order *independent* measure of Non-Simultaneous Activity, it is defined with respect to the average of *all* possible orderings of simultaneous input arrivals. For this example, there are two possible orders producing a measure of Non-Simultaneous Activity as $\frac{4}{8} = \frac{1}{2}$ and Simultaneous Reduction is $\frac{1}{8}$.

Consider an arbitrary gate n with inputs $\Lambda_n = \{n_1, n_2, ..., n_m\}$ and the set of all possible input orderings being $Perm(\Lambda_n)$. The set of all input transition times is $\tau = [t_1, t_2, ..., t_p]$, and let $|T_n^{O_j}(t_i)|$ be the amount of output activity at n caused by input transitions at time t_i if they are made to arrive in order O_j. Non-Simultaneous Activity is then defined:

$$|T_n^{NSA}| = \frac{1}{|Perm(\Lambda_n)|}\left(\sum_{t_i \in \tau}\left(\sum_{O_j \in Perm(\Lambda_n)} |T_n^{O_j}(t_i)|\right)\right) \tag{5.1}$$

The Simultaneous Reduction (which is always less than or equal to zero) is defined:

$$|T_n^{SR}| = |T_n^T| - |T_n^{NSA}| \tag{5.2}$$

When there is no simultaneous input transition arrival, the output transition activity is exactly the sum the amount of Functional Transmission and Non-Functional Transmission. The Non-Simultaneous Activity construct allows optimality to be defined with respect to the three forms of activity separately. The material of this chapter is focused on the detection of optimality attainable through estimation of the changes in Functional Transmission and Simultaneous Reduction. No estimator is presented for approximation of Non-Functional Transmission effects alone. This is because prediction of Non-Functional Transmission requires estimation of signal functionality over all time. With estimation of signal functionality over all time, the problem of finding optimality becomes equivalent to full simulation of all possible input delay conditions.

A statistical testing strategy is required to empirically test the relative importance of optimality obtained through a different activity types. The results presented in this chapter are extracted from the simulation of a gate before and after input delay permutations. Delay permutations are randomly selected from the set of all possible input delay permutations which impose different input switching orders. In the example of Fig. 5.4 which illustrates this concept, only the delay of input b is changed as it is the *relative* delay

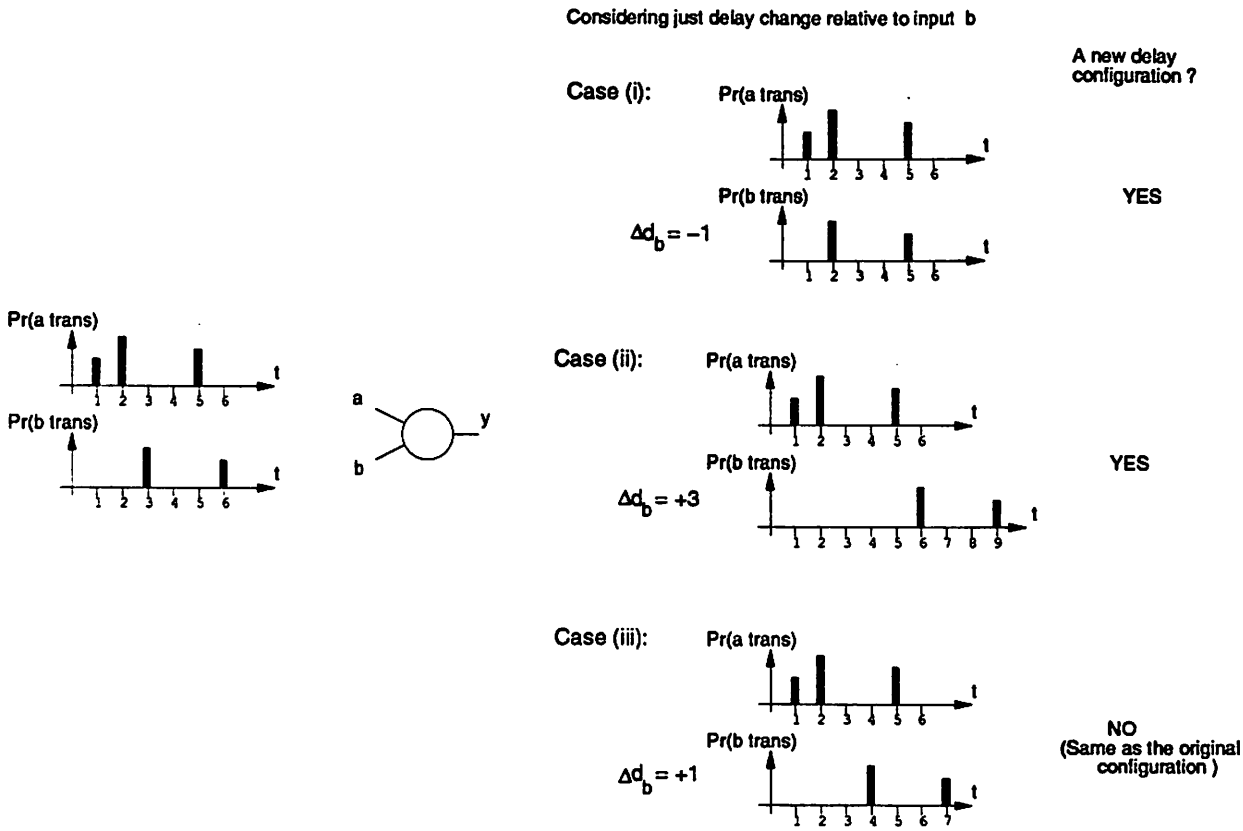

Figure 5.4: Defining Suitable Test Delay Permutations

conditions which are critical, not the absolutes. The first two cases shown represent valid delay permutations for a test. In Case (i), there is now simultaneous arrival at time 2, and in Case (ii) the order of the switching distributions has changed. In Case (iii), however, a delay of 1 does not change the input switching order relative to the original. Consequently, this is the same test permutation as the original arrival configuration. It may be noted here that under some conditions Case (i) or Case (ii) will not change the input arrival time order. Taking Case (ii) as an example, a transition at time 5 on input *a* may never occur when a transition at time 3 on the original input *b* has occurred. However, being able to compute this requires explicit knowledge of the association between transition functionality and time. As the object is to test the estimator validity when this information is *not* considered, testing an estimate for that change in the input switching *distribution* order is actually very important. For every gate tested, the number of permutations chosen for testing is the lower of the total number of possible permutations and 10.

Power Change Factor	Occurrence		Estimation	
	Actual	Est.	Detected	Incorrect
0.50x	0.004	0.002	0.54	0.00
0.80x	0.024	0.014	0.58	0.00
0.90x	0.108	0.060	0.56	0.00
0.95x	0.209	0.139	0.66	0.00
1.05x	0.101	0.113	1.00	0.11
1.10x	0.046	0.051	1.00	0.10
1.20x	0.014	0.015	1.00	0.07
1.50x	0.001	0.001	1.00	0.00

Table 5.1: Detection of Optimality using Non-Simult. Activity

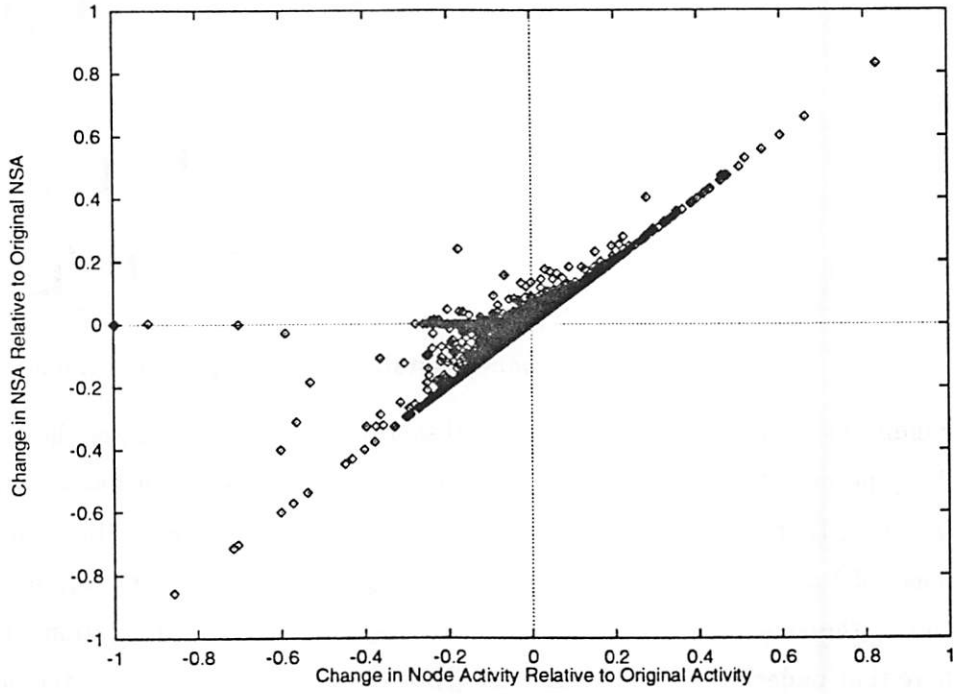

Figure 5.5: Accuracy of Optimality Detection using Only Non-Simultaneous Activity

In this empirical study, delay permutations are only tested for non-trivial cases in which at least one of the inputs to the gate has a non-zero activity interval. In the other cases, the association between transition functionality and time is trivial as input distributions are single time points so Functional Transmission and Simultaneous Reduction describes all the behavior. The results in this chapter are proportional changes relative to original activity.

The results of empirically testing 13 circuits from the ISCAS '89 benchmark set for the effect input delay permutation on gate output activity is plotted in Fig. 5.5. This is a graph of the change in activity predicted using only the change in Non-Simultaneous Activity against the actual simulated change in activity. The correlation is 0.85. (Note that the results always lie on or above a 1-to-1 correspondence line due to the fact that Simultaneous Reduction is always zero or negative.) The ability to predict optimality using just a measure of the Non-Simultaneous Activity is summarized in Tab. 5.1. Columns 2 and 3 show the proportion of total tests for which a change in power given by the factor in Column 1 occurs or is predicted to occur. Column 4 is the proportion of cases correctly predicted, Column 5 is the proportion of predictions of the occurrence which are incorrect. The results of this table are influenced by the skew evident in Fig. 5.5. An increase in activity is over-estimated by Non-Simultaneous Activity, a decrease under-estimated. In fact, Non-Simultaneous Activity alone only accounts for between 50 and 70% of the total optimality attainable. This implies that the Simultaneous Reduction effect is almost equally important.

In Fig. 5.6, the change in Simultaneous Reduction is plotted as a proportion of the change in Non-Simultaneous Activity and Fig. 5.7 is a running average of this data. The running average seems to show that contributions from Simultaneous Reduction would be expected to influence the result of the Non-Simultaneous Activity analysis by 10% or less. However, this is not the case as the error in prediction of optimality conditions using Non-Simultaneous Activity alone is affected by the *form* of distribution of the number of cases relative to the absolute change predicted. This relationship is illustrated in Fig. 5.8.

Consider the case of being able to predict the conditions for which the the total change in transition activity relative to the original Simulated Activity is x. Let $Pr^x(y)$ be the probability that the magnitude of the change in Simultaneous Reduction is greater than $(x - y)$ where y is the amount of change in the Non-Simultaneous Activity. Further, let $N_{NS}(y)$ be the number of conditions detected for Non-Simultaneous Activity change y. The number of points for which the Simulated Activity is scaled by a factor of less than $(1 + x)$ relative to the original activity is given by $N(< x)$:

$$N(< x) = \int_x^{\infty} Pr^x(y).N_{NS}(y).dy + \int_{-\infty}^{x} N_{NS}(y).dy \qquad (5.3)$$

Consider $x < 0$ (i.e. A reduction in activity). The first term of the above

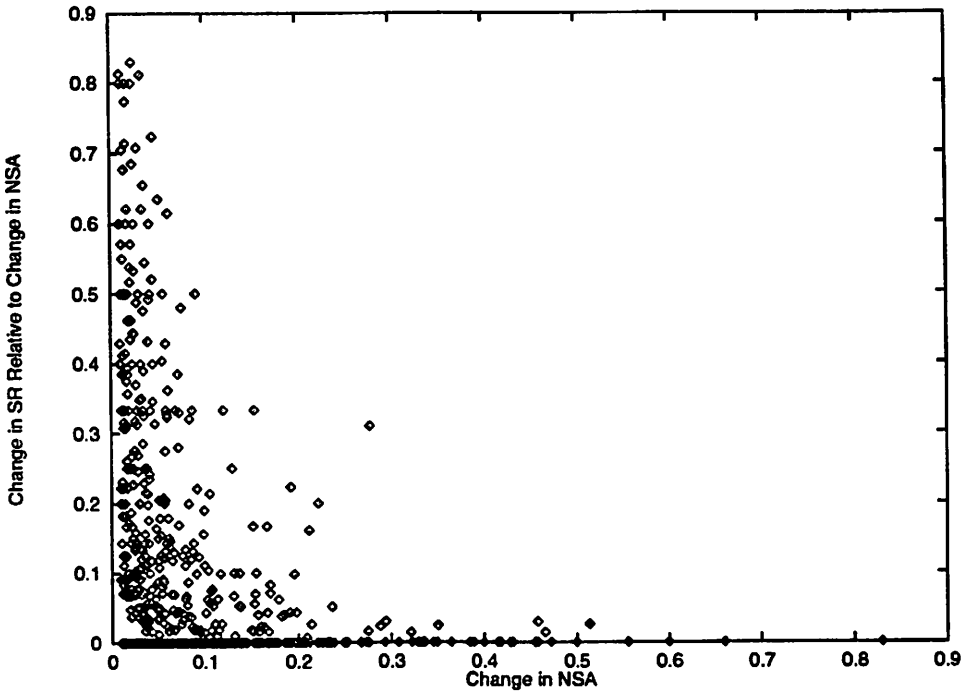

Figure 5.6: Change in SR relative to Change in NSA

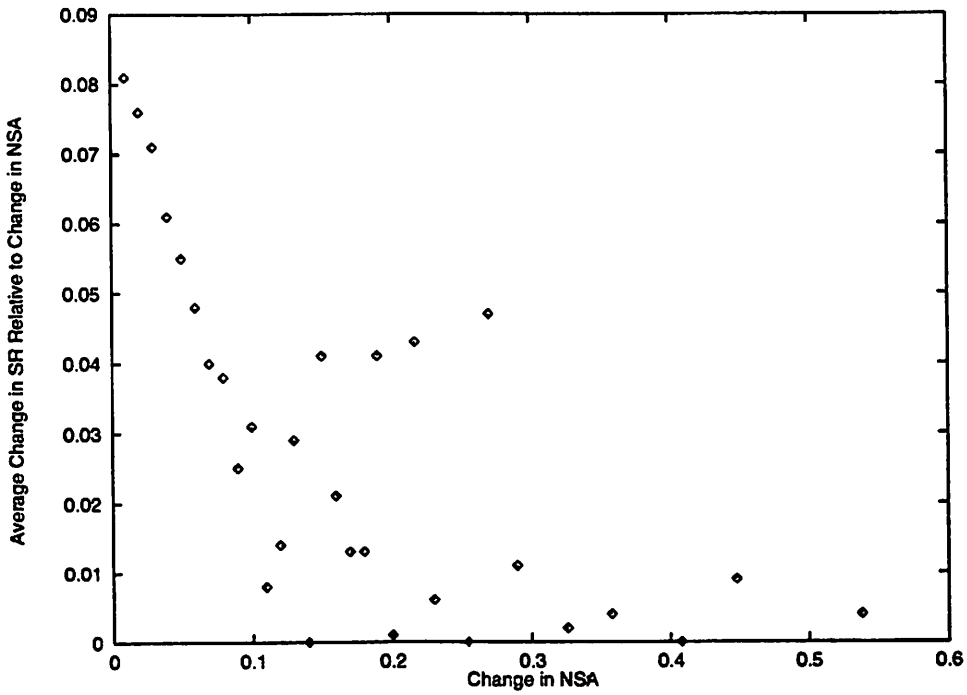

Figure 5.7: Average Change in SR relative to Change in NSA

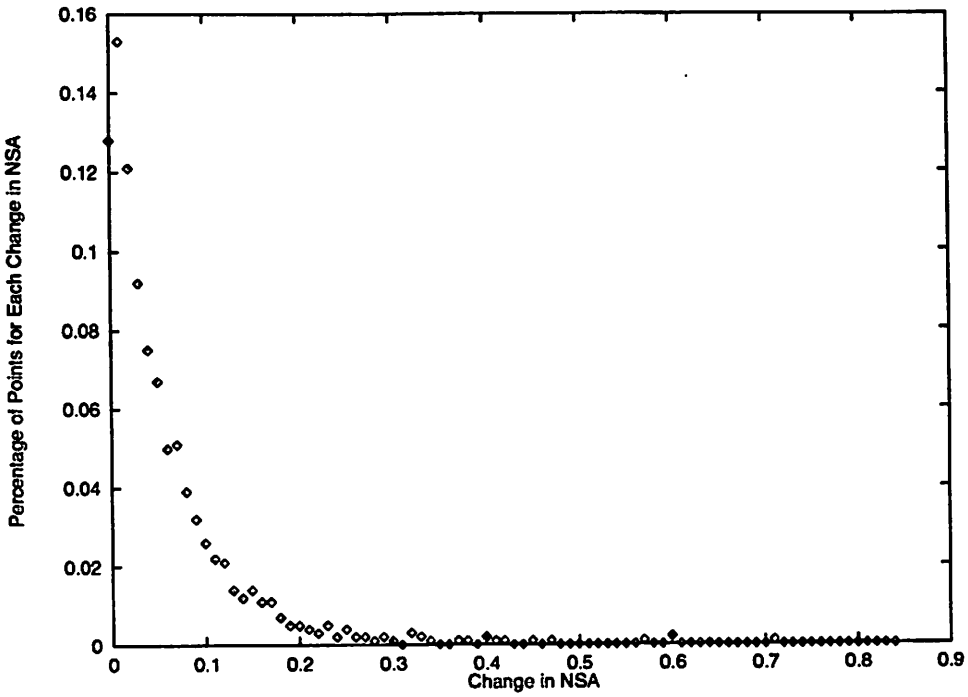

Figure 5.8: Proportion of NSA Points vs. Change in NSA

expression gives the number of points for which Non-Simultaneous Activity does not correctly estimate that the actual change in activity is larger than $|x|$. The sharp gradient depicted in Fig. 5.8 in the region of $(0.01 < |x| < 0.15)$ illustrates that $\int_x^{\infty} N_{NS}(y).dy \gg \int_{-\infty}^{x} N_{NS}(y)$ for $x < -0.05$. Although Fig. 5.7 suggests that the $Pr^x(y)$ term is generally quite small there is still a measurable number of points for which the change in Simultaneous Reduction is a significant proportion of the total activity change (Fig. 5.6). The strong inequality in the total number of points before and after $|x|$ for $|x| > 0.05$ implies that the first term of Eqn. 5.3 is significant relative to the second. This accounts for the size of the error noted in Tab. 5.1.

5.2 Non-Simultaneous Activity

The formulation of an estimation strategy based upon the concept of Functional Transmission is covered in this section. The overall goal is to test the importance of the association between functionality and time. That is, an examination of whether a reduction in the amount of Non-Simultaneous Activity can be predicted *without* explicit knowledge of the

time/function correlation. Inability to predict changes in activity without this knowledge implies that explicit simulation of input arrival times is required to detect optimality.

The estimation strategy is formulated by testing the accuracy for increasing levels of assumption. As the level of assumption increases, the ability to predict optimality decreases *but* the formulation becomes simpler. The simpler the formulation, the easier it is to find an optimal condition predicted by the estimator. The levels of assumption are:

- *The Early/Late Functional Transmission Construct.* Drops knowledge about exact functionality at any time by considering only those transitions which lead to a final change in state. (Equivalent to τ_2 in Fig. 2.4)
- *Time Independence.* Dissociates the functionality from the transition distribution completely. (Equivalent to τ_3 in Fig. 2.4)
- *Single Input Derivative.* Establishes a gradient for the expected change in activity when one input arrives earlier than another, thereby approximating input correlation.

The consideration of the validity of an approximation method needs to take into account a measure of complexity. Complexity is measured by the number of BDD computations required relative to exact symbolic simulation. Consider full simulation of every possible input delay configuration of node n with inputs $\Lambda_n = \{n_1, n_2, n_3, ..., n_m\}$. Suppose each input n_i has TT_{n_i} transition times. A BDD operation is required for every input switching time so for Non-Simultaneous Activity the complexity is $O((\sum_{n_i \in \Lambda_n} TT_{n_i})!/(\prod_{n_i \in \Lambda_n} TT_{n_i}!))$. This corresponds to every possible transition time ordering excluding those which re-order transition times for any particular input.

5.2.1 The Early/Late Functional Transmission Construct

The Early/Late Functional Transmission (ELFT) construct uses information about the earliest and latest functional transitions on each gate input. It therefore tests the ability to estimate Non-Simultaneous Activity using just previous and next state conditions of input signals. How this concept models a signal is illustrated in Fig. 5.9 for four different cases exhibiting spurious dynamic activity. In Case (i) (Case (ii)) there is no change in state, so the signal is assumed to be logic 0 (1) for the entire activity interval. In Case (iii) (Case (iv)) there is a change in state from $1 \rightarrow 0$ ($0 \rightarrow 1$). The change in state may be assumed to occur before or after the activity interval, so during the activity interval $t \in [0, 3]$ the

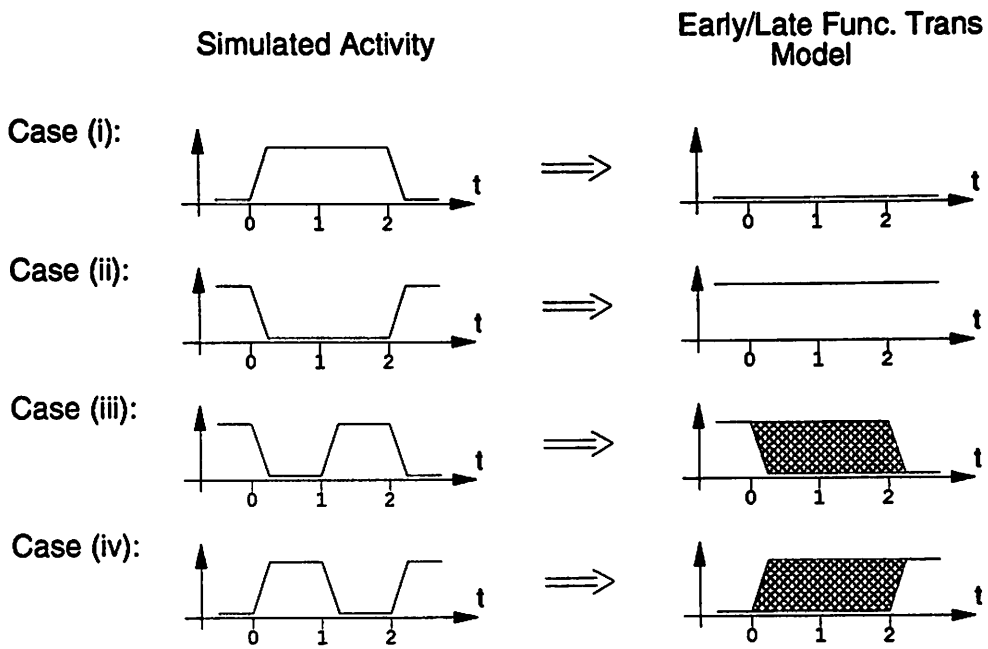

Figure 5.9: Approximating a Signal Using Early/Late Functionality

function is approximated by considering *both* the previous and next state. That is, the signal is assumed equally likely to be a logic 1 or logic 0 during the activity interval.

The Early/Late Function approximation of a signal $x(t)$ produces a relation $R_{E/L}(x(t))$ expressing conditions for logic 0 (1) at time t. Let $I_F(x) = \{(i_1, I_1), (i_2, I_2), ..., (i_n, I_n)\}$ be set of primary input pairs for which signal x changes state, $I_{NF}(x)$ all other primary input pairs. Let $t_E(i_j, I_j)$ ($t_L(i_j, I_j)$) be the earliest (latest) time for a signal change on x under application of input vector pair $(i_j, I_j) \in I_F(x)$. For static input I, let the function of x be expressed as $f(I)$. The Early/Late Function signal approximation is then given by:

$$\begin{aligned} R_{E/L}(x(t)) &= f(i_j) \quad for\,(i_j, I_j) \in I_{NF}(x) \\ &= f(i_j) \quad for\,\{((i_j, I_j) \in I_F(x))\,and\,(t_E(i_j, I_j) > t))\} \\ &= f(I_j) \quad for\,\{((i_j, I_j) \in I_F(x))\,and\,(t_L(i_j, I_j) < t))\} \\ &= \{1, 0\} \quad otherwise \end{aligned}$$

From this relation, two functions may be extracted to describe the Early/Late Function signal approximation. These are denoted $x^1_{E/L}(t)$ for the mapping to 1 and $x^0_{E/L}(t)$ for the mapping to 0 within the activity region. To illustrate how these functions are used to determine an activity approximation for a gate, consider a gate n with inputs

$\Lambda_n = \{n_1, n_2, ..., n_m\}$. Consider just the contribution to activity from a input n_j at time t. $S_n^{N(t)}(n_j)$ is the sensitivity of the output to n_j at time t given inputs with functions $N(t)$. Let $B(i)$ be the binary representation of number i, $B_j(i)$ is the j^{th} digit. The contribution to activity is then approximated by averaging over all possible input conditions described by the Early/Late Functional input relations:

$$|T_n(t)| = \frac{1}{2^{|\Lambda_n|-1}} \sum_{i=0}^{2^{|\Lambda_n|-1}} |S_n^{\{x_{1\,E/L}^{B_0(i)}(t), x_{2\,E/L}^{B_1(i)}(t), ..., x_{(|\Lambda_n|-1)\,E/L}^{B_{|\Lambda_n|-1}(i)}(t)\}}(n_j).T_{n_j}(t)| \qquad (5.4)$$

This equation embodies the operation of averaging between all possible input conditions described by the Early/Late Functional input relations. This form of approximation correctly predicts all the optimality obtainable through Functional Transmission considerations. However, the estimate also includes a contribution from an *approximation* of Non-Functional Transmission. The accuracy of this part of the estimate tests the validity of the activity interval approximation scheme used in this estimation construct. The complexity of analysis of this estimate is equivalent to that of full simulation due to the association of time and functionality.

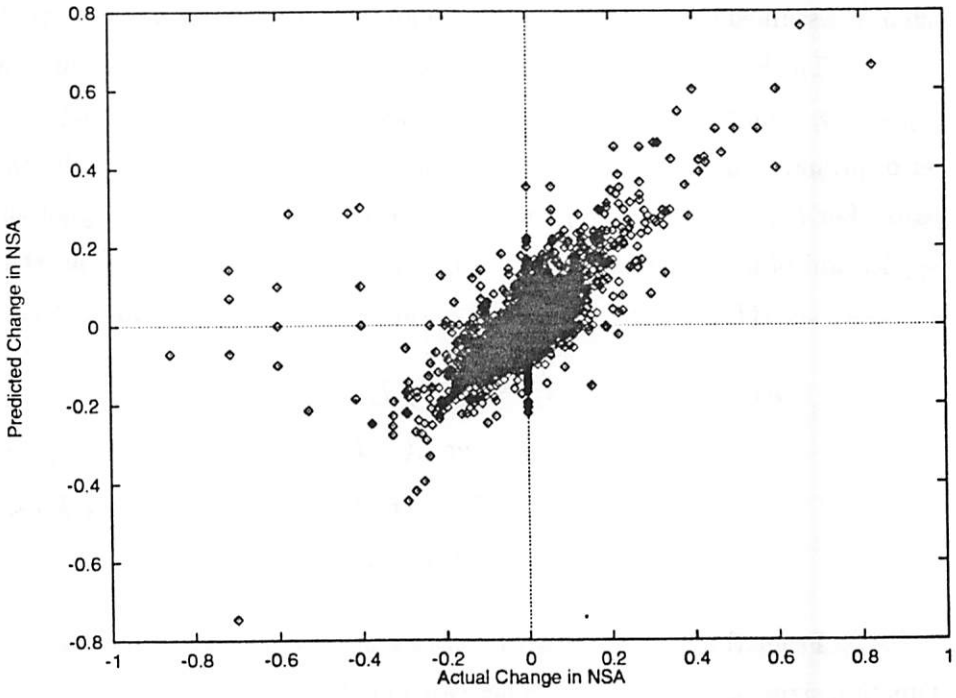

Figure 5.10: ELFT Approximation vs. Non-Simultaneous Activity

Power Change Factor	Occurrence		Estimation	
	Actual	Est.	Detected	Incorrect
0.50x	0.002	0.000	0.08	0.00
0.80x	0.013	0.007	0.42	0.29
0.90x	0.058	0.066	0.59	0.47
0.95x	0.134	0.192	0.71	0.50
1.05x	0.112	0.131	0.63	0.46
1.10x	0.050	0.063	0.68	0.46
1.20x	0.014	0.022	0.77	0.52
1.50x	0.001	0.002	0.83	0.44

Table 5.2: Detection of Non-Simult. Optimality using Early/Late Func. Trans.

This construct has been tested on the ISCAS '89 benchmark set. The aggregate results are plotted in Fig. 5.10 for the ELFT estimator against the Non-Simultaneous Activity. A correlation coefficient of 0.66 is obtained, and the results of using this to predict optimality is summarized in Tab. 5.2. It is interesting to note that there are points clustered along the y-axis corresponding to a change in activity being estimated when there is actually no change. However, there is no such cluster on the x-axis. The error mechanisms for points which lie on the x- or y-axis exclusively are described in Fig. 5.11. The first case will

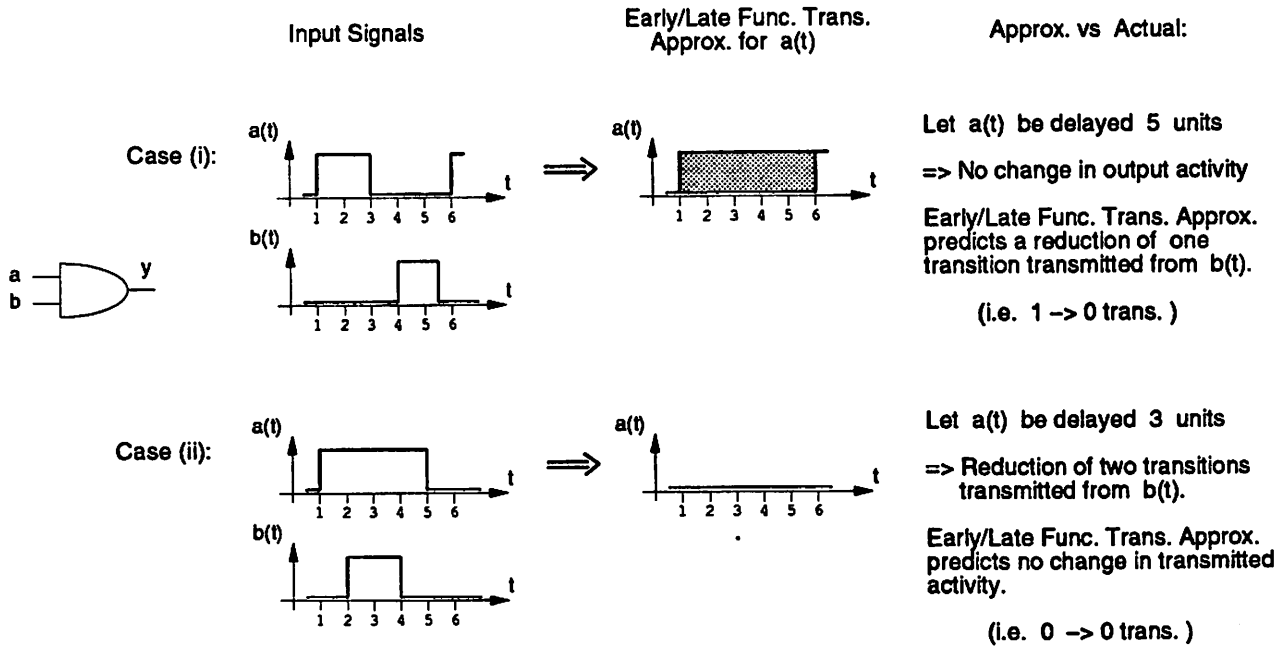

Figure 5.11: Errors with the ELFT construct

generate a point on the y-axis. The approximation of the activity interval using either the previous or next state implies that there will be a change predicted by the ELFT construct when there is no change in Non-Simultaneous Activity. The second case is the converse of this with no output activity change predicted but a change in actuality. The fact that there is *not* a strong clustering of points along the x-axis similar to the y-axis implies that the error incurred by neglecting totally spurious activity (activity when there is no change in state) is not as significant as that incurred by assuming that either the previous or the final state is equally likely during the activity interval.

The results show that the accuracy of the ELFT construct is not sufficient for use in detection of Non-Simultaneous Activity optimality. However, it is not expected that an estimator based upon an independence of switching distribution and functionality will be able to predict optimality significantly better than this construct can (Sec. 5.2.2).

5.2.2 The Time Independence Assumption

When time and function are dissociated, the ability to estimate the correlation between inputs over all time is severely compromised. Fig. 5.12 is an illustration of this point. Two transition probability distributions are presented for activity on an input $x(t)$, one distribution for the positive transitions and one for the negative. For the same previous and next state correlation, there are seven different activity types which these distributions could describe. Clearly, if two inputs to a gate were actually the same, this correlation of 1.0 over all time is no longer detectable. This motivates the need to describe activity using more than just absolute transition distributions.

An estimate of the correlation between lines can be improved by extracting distributions which describe the time of a change in state given that a change in state occurs. The distribution gives the probability that a state change has occurred prior to some time t. For example, suppose Case (i) in Fig. 5.12 is the correct activity type. A distribution of the *earliest* possible change in state has a probability of 1.0 at time $t = 1$ for both a positive and negative change in state. The only cases which satisfy this are Case (i), Case (v), Case (vi) and Case (vii). Consider now a distribution of the *latest* possible change in state. The positive and negative distributions have a probability of 1.0 at time $t = 3$. Now the only applicable activity type which satisfies both these is Case (i). These two distributions are defined as the Early Functional Distribution (*Early FD*), and the Late Functional

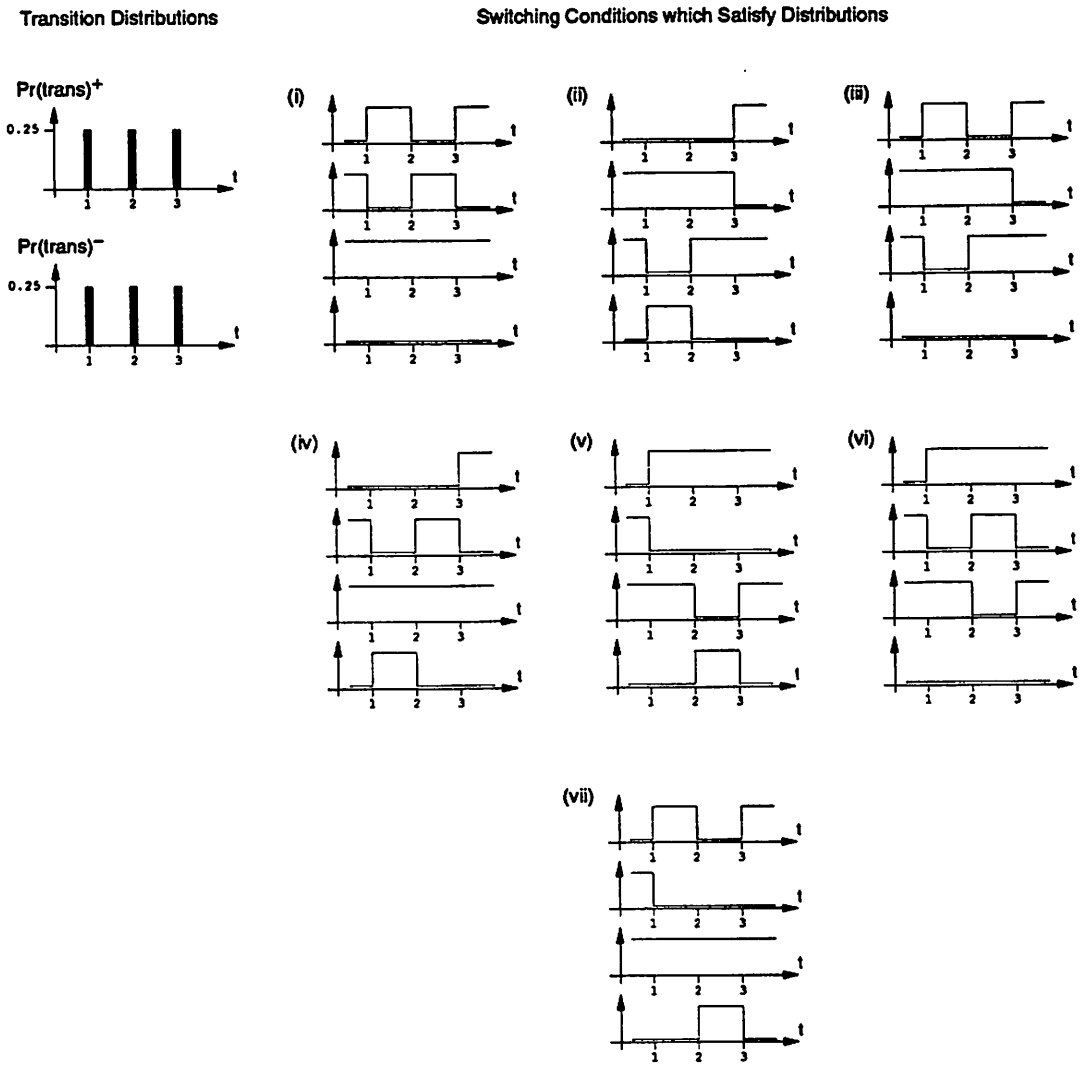

Figure 5.12: The Loss of Correlation with Time/Function Dissociation

Distribution (*Late FD*).

In general, the Early FD, Late FD and Transition Distribution together do not uniquely specify a satisfying activity type. Fig. 5.13 illustrates what is actually predicted through the use of these distributions. Consider the cases for which the earliest positive state transition occurs at time $t = 3$, an event of probability 0.3 given that a positive state transition does occur. The total probability of the latest functional transition occurring *after* this time is 0.7. The late functional distribution thereby encompasses three possible events with a activity intervals of 0, 3 and 5 respectively. The probability of each event is the probability that the early functional transition occurs at $t = 3$ multiplied by the

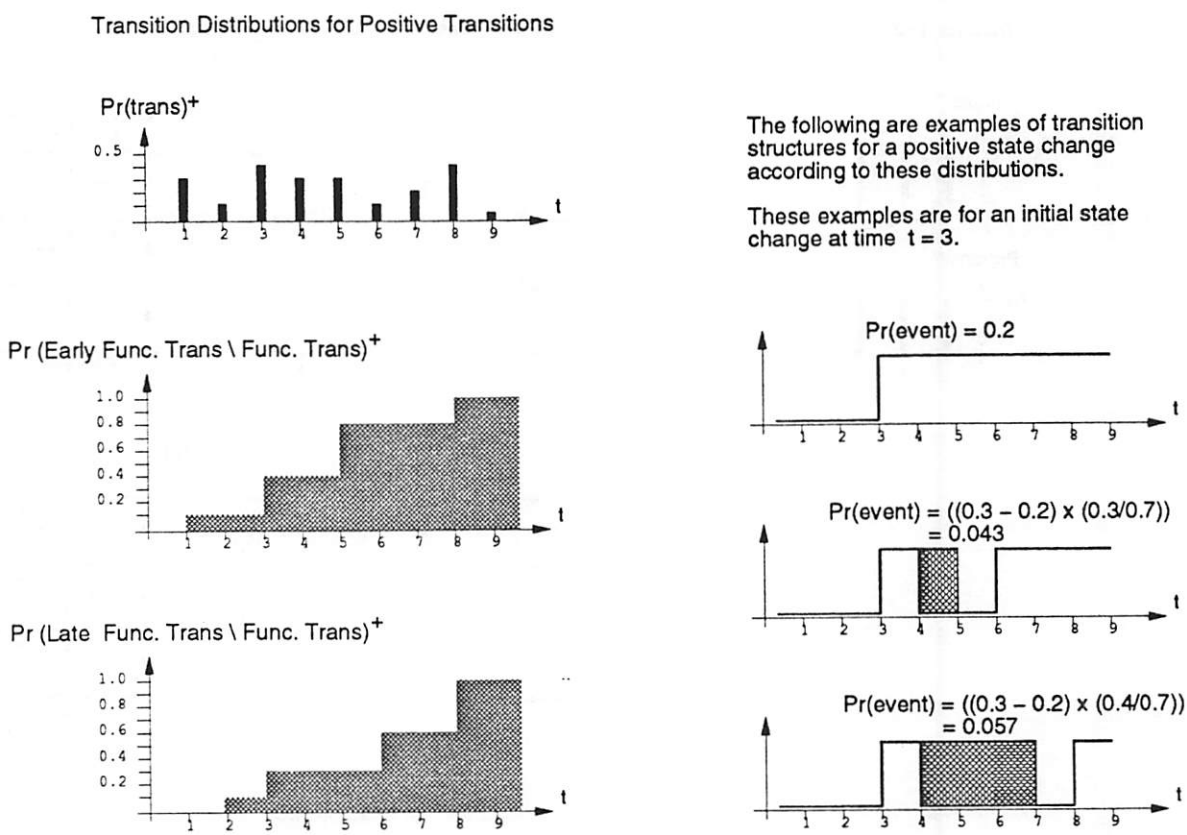

Figure 5.13: Specifying and Activity Type using Early/Late Function Distributions

probability that a late functional transition then occurs at times $t = 3, 6$ or 8 respectively.

The dissociation of functionality from the Early/Late Distributions implies that the probability computed for a certain width of activity interval is not exact. However, the independence allows the distributions to be taken separately. At time t, the probability that a line changes state earlier (later) than t is given by the Early FD (Late FD) at t. The transmission results obtained from the average of considering each distribution separately is then *exactly* the ELFT construct with time and function dissociated, denoted here the Time Independent Early/Late Functional Transmission (TI-ELFT) approximation. The ELFT construct has already been shown to be inadequate. This suggests that perhaps more complex probability descriptions need to be made. However, it will be shown from the results of this section that the dissociation of time and function alone is expected to incur an error penalty which is of the same order as the ELFT construct, regardless of the complexity of the probabilistic estimation strategy. This is a conclusion derived from an

examination of how well the TI-ELFT approximation estimates the ELFT construct. As these activity models are identical except for the fact that one uses the dissociation of time and function, this is a test of the validity of that particular concept.

The distributions described establish the probability that state transitions occur on the inputs to a gate in a certain order. To establish an estimate of the output activity of a gate n due to a transition on an input n_i at time t, it is necessary predict the probability of transmission given the input switching order probabilities. It was established in Chap. 4 that such an estimate needs to depend upon the functionality of the spurious activity. This is approximated by first computing the transmitted activity from each input under every possible input ordering assuming separate activity intervals. The concept is depicted in Fig. 5.14. For each input order, the amount of activity transmitted from each input can be computed exactly. From this, a probability of transmission can be established for each input given a state transition ordering on the other gate inputs. The complexity of the analysis required to compute the separate activity interval configurations is $0(|\Lambda_n|!.\sum_{n_i \in \Lambda_n} TT_{n_i})$.

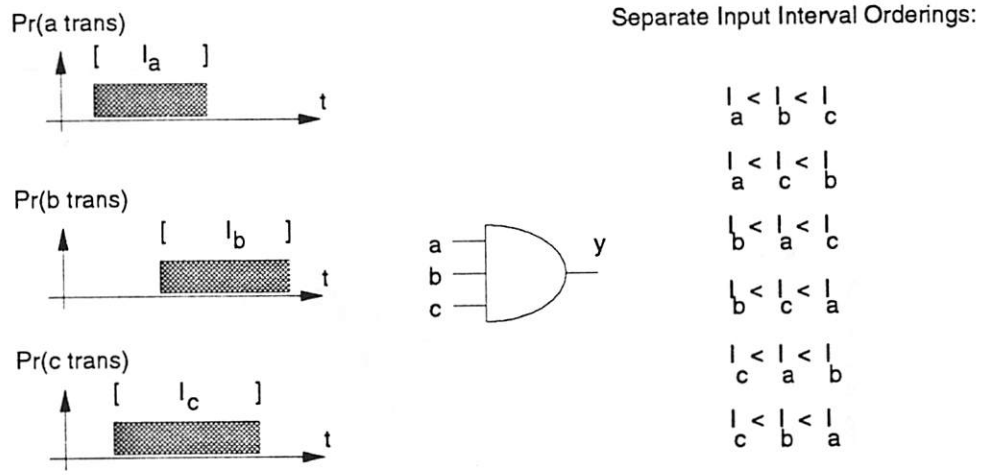

Figure 5.14: Testing All Separate Activity Interval Configurations

The activity estimate computation is summarized in the following equation. For input $n_i \in \Lambda_n$ at time t, the arrival of a state transition on another input is indicated by a 0 for earlier, 1 for later. $P^0_{n_j}(t)|_E$ ($P^1_{n_j}(t)|_E$) is the probability that a state transition will occur earlier (later) than t on input n_j using the Early FD. $P^0_{n_j}(t)|_L$ ($P^1_{n_j}(t)|_L$) is similarly defined for the Late FD. $T_{n_i}(t)$ is the total amount of activity on n_i at time t, $Pr^{B(k)}_{trans}(n_i)$ be the probability of transmission from n_i under input state transition order $B(k)$ relative

to n_i. Assume time discretization $t \in \{t_0, t_1, ...\}$.

$$|T_n| = \sum_{i=0}^{\infty}(\sum_{n_j \in \Lambda_n} |T_{n_j}(t_i)|.\frac{1}{2}(\sum_{k=0}^{2^{|\Lambda_n|-1}} Pr_{trans}^{B(k)}(n_j).(\prod_{n_l \in \Lambda_n \setminus n_j} P_{n_j}^{B_l(k)}(t_i)|_E + \prod_{n_l \in \Lambda_n \setminus n_j} P_{n_j}^{B_l(k)}(t_i)|_L))) \tag{5.5}$$

Eqn. 5.5 involves only numerical computation. Consequently, the complexity of the entire estimation method in terms of BDD manipulations is determined by the separate activity interval configuration analysis and construction of the Early and Late FD (second term of the following complexity expression). This gives a total complexity of $0((|\Lambda_n|!.\sum_{n_i \in \Lambda_n} TT_{n_i}) + 2.\sum_{n_i \in \Lambda_n} TT_{n_i})$. The number of numerical operations required to find the optimum delay configuration is the same as the BDD operation complexity for full simulation, namely $O((\sum_{n_i \in \Lambda_n} TT_{n_i})!/(\prod_{n_i \in \Lambda_n} TT_{n_i}!))$.

The accuracy of this approximation has been tested on the ISCAS '89 benchmark set. The aggregate results are plotted in Fig. 5.15 for the TI-ELFT approximation against the Non-Simultaneous Activity. The correlation coefficient for the data is 0.72. The prediction of optimality is summarized in Tab. 5.3.

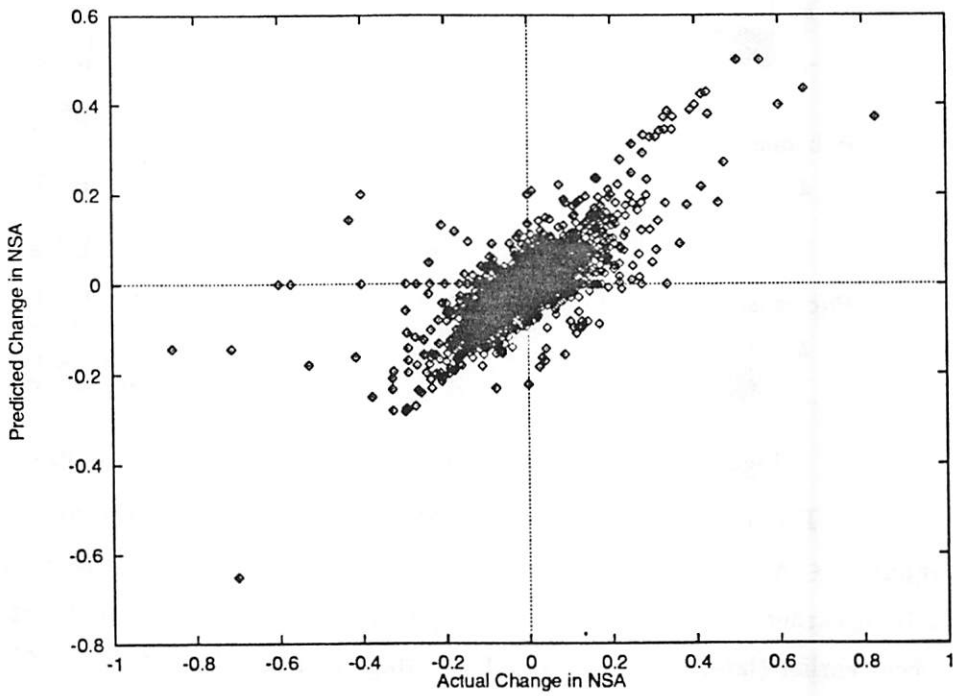

Figure 5.15: TI-ELFT vs. Non-Simultaneous Activity

Power Change Factor	Occurrence		Estimation	
	Actual	Est.	Detected	Incorrect
0.50x	0.002	0.000	0.08	0.00
0.80x	0.013	0.004	0.26	0.14
0.90x	0.058	0.037	0.51	0.20
0.95x	0.134	0.094	0.59	0.16
1.05x	0.112	0.069	0.47	0.24
1.10x	0.050	0.024	0.37	0.22
1.20x	0.014	0.007	0.40	0.24
1.50x	0.001	0.000	0.33	0.00

Table 5.3: Detection of Non-Simult. Optimality using Time Indep. Approx.

The results of this estimator are actually slightly *better* at predicting the optimality of Non-Simultaneous Activity than the ELFT construct which this technique approximates. This is a consequence of the probability estimation effectively 'smoothing out' the error conditions indicated in Fig. 5.11. Specifically, the ELFT construct still associates functionality with time which can generate particular functional associations which are quite erroneous.

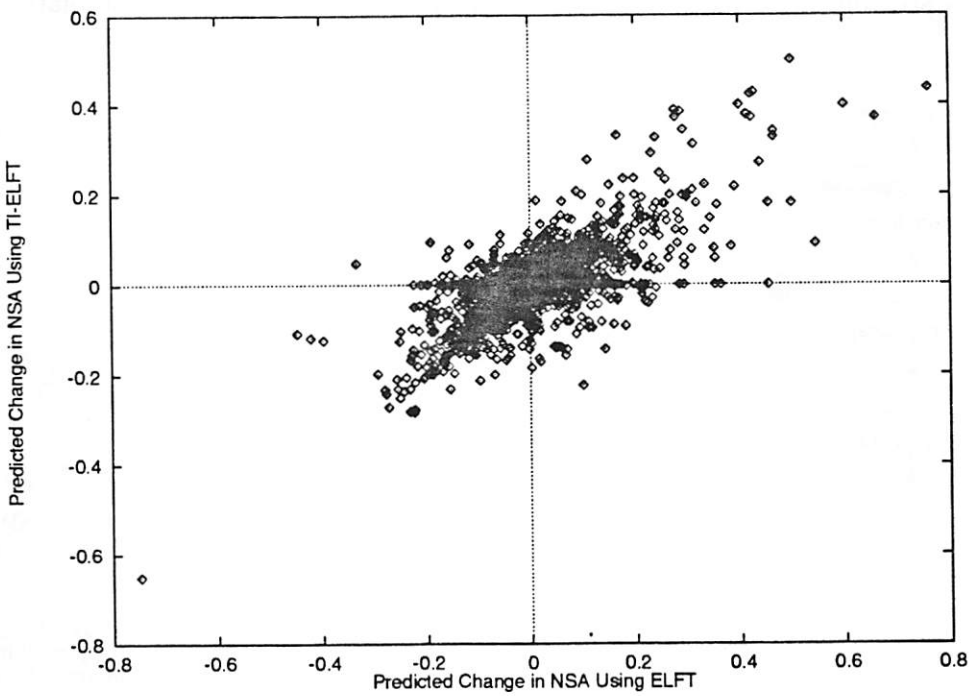

Figure 5.16: ELFT vs. TI-ELFT

The dissociation of time and function eliminates construction of such particular error conditions, thereby suppressing them.

It needs to be emphasized, however, that the accuracy of the TI-ELFT approximation is *not* limited by the fact that it essentially models the ELFT construct. Fig. 5.16 is a plot of the ELFT construct against the TI-ELFT. This then shows how well a time/functionality dependent construct can be explicitly approximated when time and functionality are dissociated. The data correlation is only 0.68. The implication here is that the dissociation of time and functionality absolutely restricts the accuracy of an estimate of change in activity due to change in delay to an error of this significance.

5.2.3 The Single Input Dependent Derivative Approximation

The estimation strategies outlined in the previous two subsections require testing of all possible delay configurations in the determination of optimality. The best framework for detection of optimality without having to test so many cases is one which makes possible a statement about whether increasing or decreasing delay with respect to any other single input is beneficial or detrimental. The TI-ELFT approximation associates the detection of optimality with a specific delay ordering applied over all inputs, not relative timing between

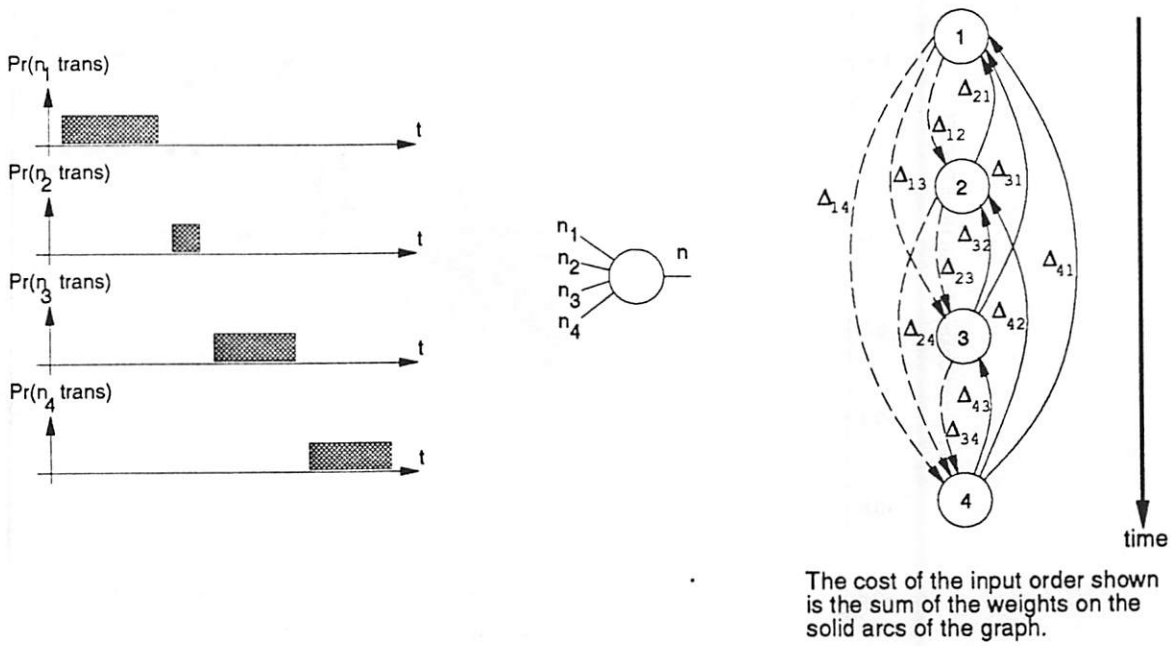

Figure 5.17: Single Input Dependent Derivative with Separate Input Activity Intervals

input pairs. The desired framework for detection of delay optimality is depicted in Fig. 5.17. This example assumes separate input activity intervals.

The graph of Fig. 5.17 represents the activity change with change in the input arrival order. Each input is associated with a node which represents the total activity transmitted from that input when it arrives earliest. From each graph node there is an edge to every other node. A directed edge $< e_{n_i}, e_{n_j} >$ represents a *change* in transmitted activity from input n_i when it arrives later than functional transitions on node n_j. In this example, the solid arcs represent the cost $C(order)$ of the input ordering, namely $C(n_1, n_2, n_3, n_4) = \Delta_{21} + \Delta_{31} + \Delta_{32} + \Delta_{43} + \Delta_{42} + \Delta_{41}$. Finding an optimal transition arrival order is then equivalent to finding a node ordering on the graph of minimum cost, the cost being the sum of all edges from higher to lower order nodes.

For a general case with overlapping input activity intervals, the cost function is the sum of edge weights scaled by the probability that one input transitions earlier than another. The goal then is to find a relative timing which maximizes the scaling factor on edges with the greatest negative weight. This forms the basis of a simple heuristic to find optimality without having to exhaustively test all possible input delay configurations.

The accuracy of the Single Input Dependent Derivative Approximation (*SID-TI-ELFT*) has been investigated. To establish the approximation, an $0(2.|\Lambda_n|^2.\sum_{n_i \in \Lambda_n} TT_{n_i})$ method was used, as compared to the $0(|\Lambda_n|!.\sum_{n_i \in \Lambda_n} TT_{n_i})$ complexity of the separate input activity interval configuration computation of Sec. 5.2.2. The number of cases to be analyzed is reduced by only considering the following separate input activity interval configurations relative to each input n_i:

1. n_i is later (earlier) than all other inputs

2. n_i is later (earlier) than all inputs except $n_j, j \neq i$

From these test cases, an estimate of the average change in activity can be established for delay conditions which swap the arrival order of any two inputs n_i, n_j.

Fig. 5.18 depicts the correlation between the SID-TI-ELFT approximation and the TI-ELFT approximation. The correlation coefficient is 0.83. Tab. 5.4 summarizes the overall use of the method in the detection of Non-Simultaneous Activity optimality. It should be noted from the table that the *accuracy* of the detection of optimality conditions is similar for this method and the TI-ELFT approximation, however the *number* of conditions

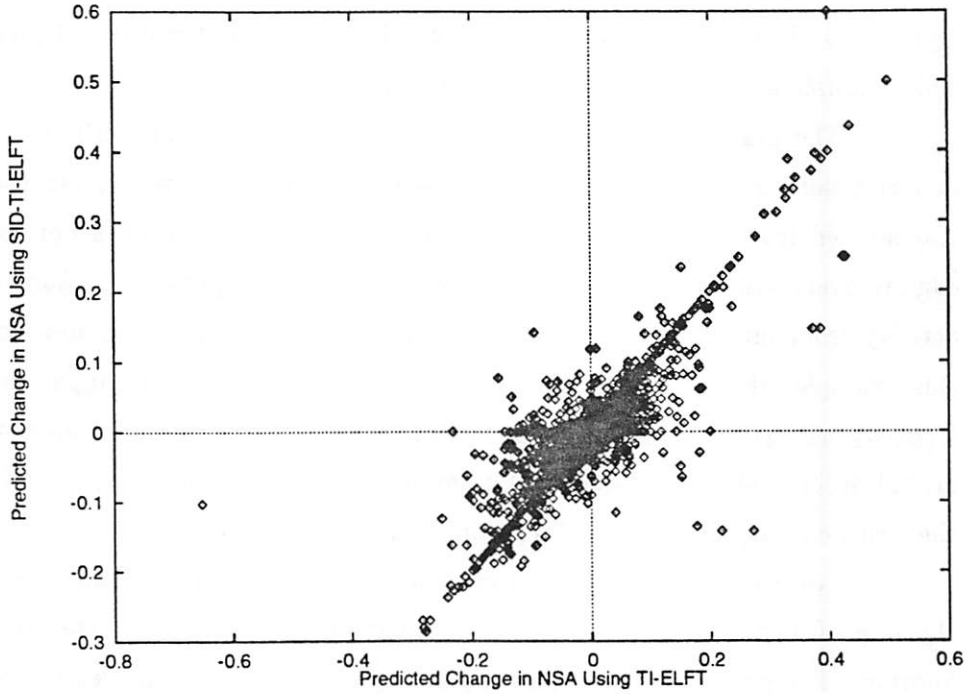

Figure 5.18: SID-TI-ELFT approximation vs. TI-ELFT approximation

Power Change Factor	Occurrence		Estimation	
	Actual	Est.	Detected	Incorrect
0.50x	0.002	0.000	0.08	0.00
0.80x	0.013	0.002	0.15	0.08
0.90x	0.058	0.025	0.29	0.32
0.95x	0.134	0.069	0.38	0.26
1.05x	0.112	0.046	0.29	0.29
1.10x	0.050	0.016	0.27	0.17
1.20x	0.014	0.005	0.33	0.13
1.50x	0.001	0.001	0.50	0.00

Table 5.4: Detection of Non-Simult. Optimality using Single Input Depend. Deriv. Approx.

detected is significantly reduced. This property can be seen in the distribution of points in Fig. 5.18 which consists of points clustered along the line $x = y$, and other points clustered along the x-axis which corresponds to zero sensitivity. In combination with the limited accuracy predicted for *any* time/function independent approximation (Sec. 5.2.2), it cannot be expected a method which does not account for input correlation exactly will be sufficient

for detection of delay dependent optimality in combinational networks. A discussion of the heuristics which may be applied to the SID-TI-ELFT approximation has therefore been omitted.

5.3 Simultaneous Reduction

The analysis of Sec. 5.2 dealt with approximating activity assuming no simultaneous switching. This section outlines an approach to the estimation of Simultaneous Reduction when time and functionality are dissociated. A description of the problem in the form of a probability relation is first needed. To aid in the construction of this formulation, consider the case of separating the arrival times for simultaneous transition times on the inputs to a gate. This concept first described in Sec. 5.1 is illustrated by example for one possible ordering in Fig. 5.19.

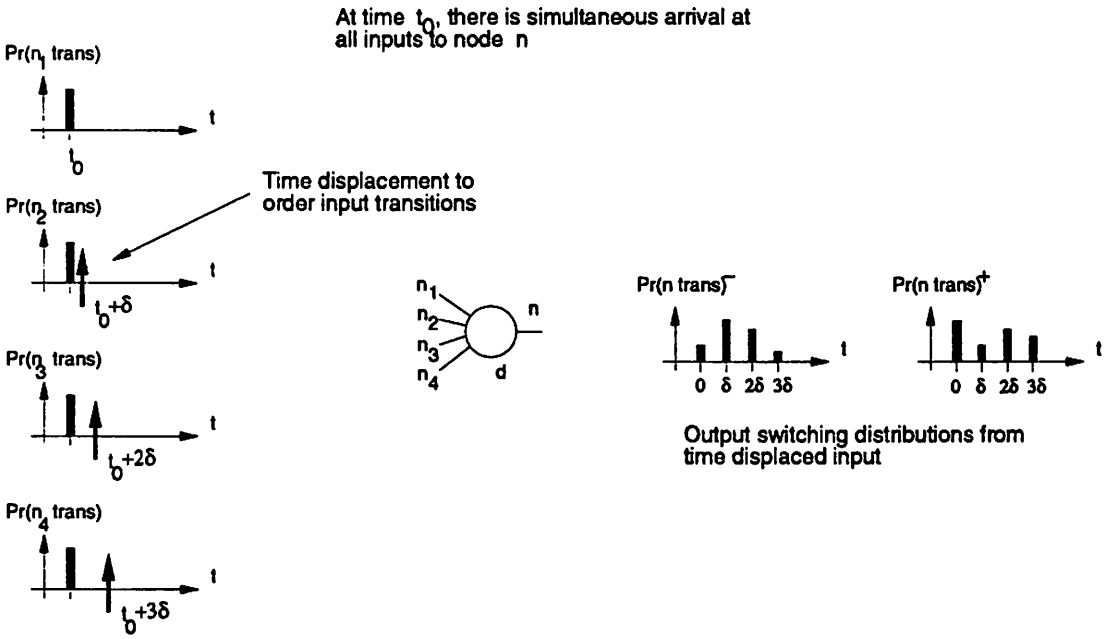

Figure 5.19: Formulation of Simultaneous Reduction

For a given transition ordering, distributions for the positive and negative output transitions can then be computed from complete simulation or the TI-ELFT approximation. To formulate of the Simultaneous Reduction (T_n^{SR}) let $Pr_i(\uparrow_j)$ ($Pr_i(\downarrow_j)$) be the probability of a positive (negative) output transition at time $t_i + j.\delta$, $t_i \in \{t_0, t_1, t_2, ...\}$ the actual

set of possible output transition times allowing simultaneous input arrival and $j.\delta$ added to the timing to make arrivals non-simultaneous. $Pr_i(\uparrow_{<j})$ ($Pr_i(\downarrow_{<j})$) is the probability of a positive (negative) transition at time t_i after output transition times $[t_i, t_i+\delta, ..., t_i+(j-1).\delta]$ are collapsed back into one another.

$$Pr_i(\uparrow_{<j}) = Pr_i(\uparrow_{<(j-1)}).\overline{Pr_i(\downarrow_j \mid \uparrow_{<(j-1)})} + Pr_i(\uparrow_j)$$

$$Pr_i(\downarrow_{<j}) = Pr_i(\downarrow_{<(j-1)}).\overline{Pr_i(\uparrow_j \mid \downarrow_{<(j-1)})} + Pr_i(\downarrow_j)$$

$$|T_n^{SR}| = \sum_{i=0}^{\infty} \sum_{j=0}^{|\Lambda_n|-1} (Pr_i(\uparrow_j \mid \downarrow_{<j}).Pr_i(\downarrow_{<j}) + Pr_i(\downarrow_j \mid \uparrow_{<j}).Pr_i(\uparrow_{<j})) \quad (5.6)$$

Eqn. 5.6 requires that the correlation between transitions at different times be known. The dissociation of these properties reduces the above equations to the following:

$$Pr_i(\uparrow_{<j}) = Pr_i(\uparrow_{<(j-1)}).\overline{Pr_i(\downarrow_j)} + Pr_i(\uparrow_j)$$

$$Pr_i(\downarrow_{<j}) = Pr_i(\downarrow_{<(j-1)}).\overline{Pr_i(\uparrow_j)} + Pr_i(\downarrow_j)$$

$$|T_n^{SR}| = \sum_{i=0}^{\infty} \sum_{j=0}^{|\Lambda_n|-1} (Pr_i(\uparrow_j).Pr_i(\downarrow_{<j}) + Pr_i(\downarrow_j).Pr_i(\uparrow_{<j})) \quad (5.7)$$

The results of two tests of Simultaneous Reduction estimation are presented here. The first test uses the results of the TI-ELFT approximation to estimate the relevant switching distributions. The second test assumes that the sign of the output transition has not been approximated, so the probability of a positive or negative transition is half the total transition probability at that time. This demonstrates how the accuracy of the Simultaneous Reduction estimate is severely penalized by the lack of a good estimate of output transition sign. This fact detrimentally affects the usefulness of averaging techniques such as the SID-TI-ELFT approximation which are much less accuracy at predicting sign than the TI-ELFT approximation. Fig. 5.20 and Tab. 5.5 refer to the first test, Fig. 5.21 and Tab. 5.6 to the second.

The estimate which uses transition sign information has a correlation of 0.96, as compared with the 0.73 for the estimator which does not have good transition sign prediction. This test shows that the TI-ELFT approximation is clearly sufficient for estimation of the Simultaneous Reduction even though it has demonstrated limited accuracy in the determination of optimality relative to Non-Simultaneous Activity. The improved accuracy here is a consequence of the fact that Simultaneous Reduction depends upon the *absolute*

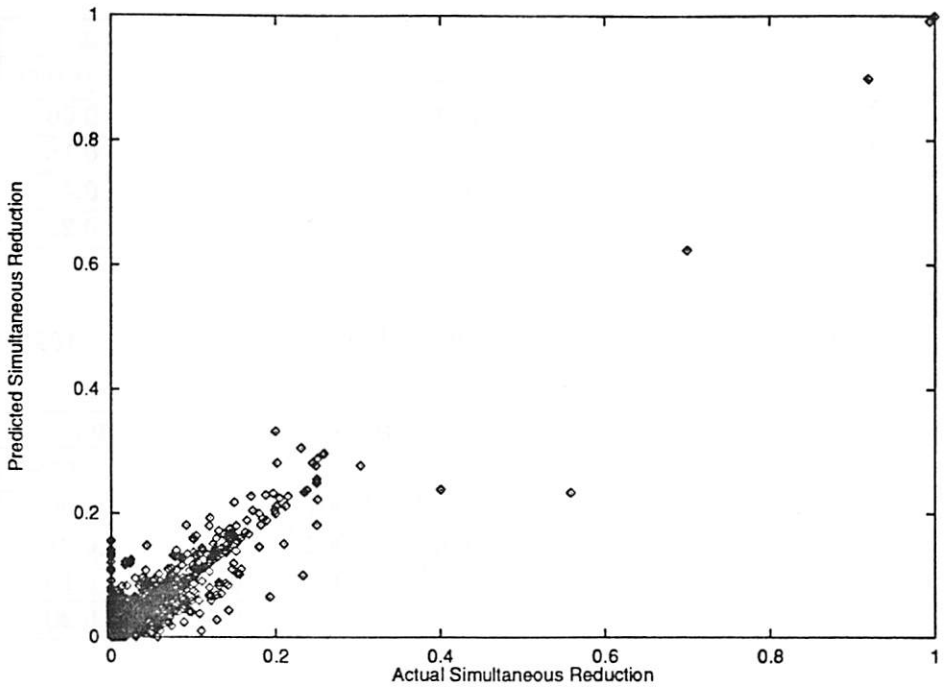

Figure 5.20: Simultaneous Reduction: TI-ELFT and Full Trans. Sign Info.

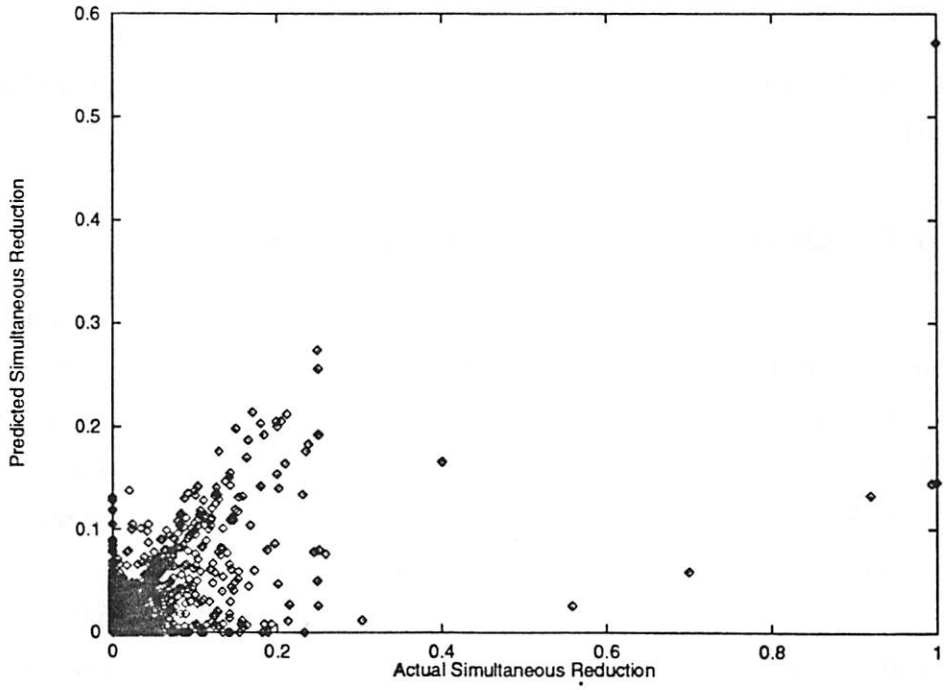

Figure 5.21: Simultaneous Reduction: TI-ELFT and No Trans. Sign Info.

Power Change Factor	Occurrence		Estimation	
	Actual	Est.	Detected	Incorrect
-0.50x	0.001	0.001	0.88	0.00
-0.20x	0.007	0.008	0.92	0.16
-0.10x	0.044	0.047	0.92	0.15
-0.05x	0.080	0.089	0.88	0.22

Table 5.5: Detection of Simult. Reduction using Time Indep. Approx. with Sign

Power Change Factor	Occurrence		Estimation	
	Actual	Est.	Detected	Incorrect
-0.50x	0.001	0.001	0.38	0.00
-0.20x	0.007	0.002	0.21	0.27
-0.10x	0.044	0.024	0.44	0.19
-0.05x	0.080	0.052	0.52	0.20

Table 5.6: Detection of Simult. Reduction using Time Indep. Approx. with Sign

amount of Non-Simultaneous Transition activity, whereas the results of the previous section reflect the ability to detect incremental changes in this quantity.

5.4 Overall Optimality Prediction

The results of the previous two sections are combined here to demonstrate the accuracy of the time/functionality independence assumption. The Non-Simultaneous Activity is estimated using the TI-ELFT approximation of Sec. 5.2.2, and the Simultaneous Reduction using the output sign sensitive approximation of Sec. 5.3. Overall, a correlation of 0.85 between estimate and empirical data was found. The aggregate data is presented graphically in Fig. 5.22 and is summarized in Tab. 5.8.

The results shown in Tab. 5.8 demonstrate that the usefulness of such an approximation scheme is limited. Although the optimality conditions predicted by this estimator are reasonably accurate (better than 75% percent of points selected for any particular optimality condition are correct), the sensitivity of the method is not sufficient. Less that 50% of the delay conditions which correspond to an increase of activity are correctly categorized.

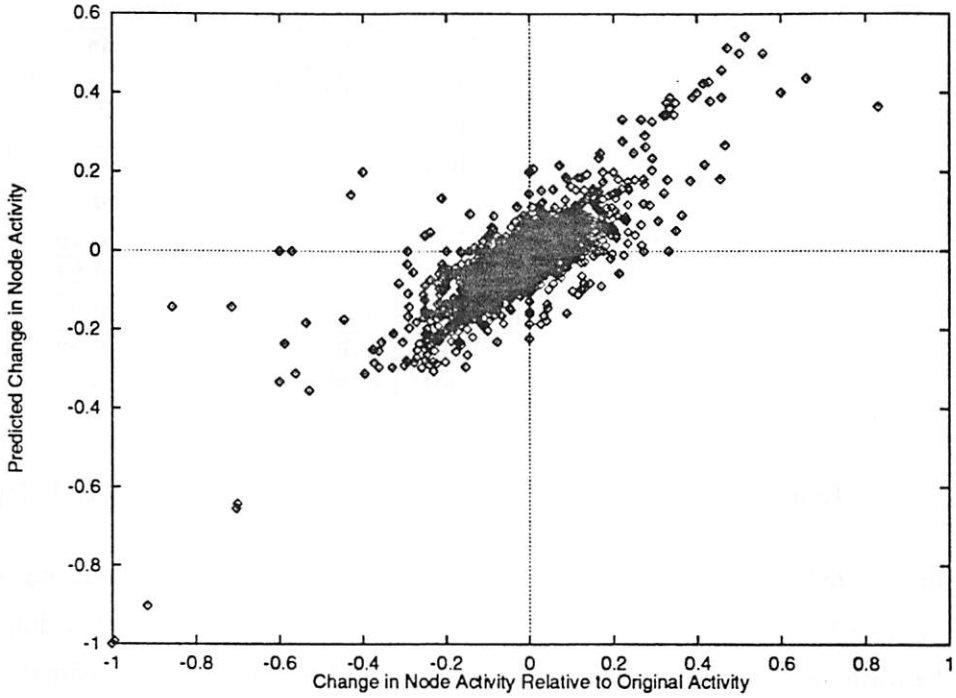

Figure 5.22: Overall Optimality Prediction with Time/Functionality Independence

Ckt.	Non-Simult. Correl.	Simult. Red. Correl.	Overall Correl.		Non-Simult. Correl	Simult. Red. Correl.	Overall Correl.
s344	0.81	0.80	0.80	s713	0.47	0.97	0.54
s386	0.96	0.98	0.97	s820	0.76	0.94	0.78
s400	0.81	0.82	0.80	s832	0.78	0.95	0.81
s444	0.90	0.89	0.86	s838	0.68	0.99	0.97
s510	0.85	0.90	0.87	s1196	0.80	0.94	0.83
s526	0.78	0.93	0.83	s1238	0.75	0.95	0.81
s641	0.43	0.98	0.55				

Table 5.7: Correlation for Time Independent Model

This is also true for reduction in activity of greater than 10%.

In Tab. 5.7 the significant dependence of estimator accuracy upon the specific circuit is shown, in particular with respect to the estimation of Non-Simultaneous Activity changes. This is quite different to the stable statistical properties exhibited for the estimation techniques presented in Chap. 3 and Chap. 4. The statistical variability is a consequence of the strong dependence of the optimality sought upon the functionality *and*

Power Change Factor	Occurrence		Estimation	
	Actual	Est.	Detected	Incorrect
0.50x	0.004	0.001	0.33	0.00
0.80x	0.024	0.014	0.48	0.19
0.90x	0.108	0.092	0.70	0.18
0.95x	0.209	0.192	0.75	0.19
1.05x	0.101	0.066	0.46	0.24
1.10x	0.046	0.021	0.36	0.24
1.20x	0.014	0.008	0.44	0.22
1.50x	0.001	0.001	0.43	0.25

Table 5.8: Detection of Optimality using Time/Functionality Independence

precise delays in the network. The importance of correctly estimating signal properties such as those depicted in Fig. 5.11 can vary in significance between different circuit configurations. This influences the applicability of an activity approximation scheme which dissociates time and functionality.

5.5 Summary

The material described in this chapter completes a thorough study of the ability to dissociate time and functionality while still being able to estimate optimality. The presentation describes the three effects which contribute to delay dependent changes in gate activity. These effects are:

- Simultaneous Activity
- Functional Transmission
- Non-Functional Transmission

These effects are isolated by first splitting the total activity into the sum of two activity types: Non-Simultaneous Activity and Simultaneous Reduction. Estimation techniques for both activity types have been outlined.

Optimality detection for Non-Simultaneous Activity requires estimation of both Functional and Non-Functional Transmission. Even if Functional Transmission is estimated

well, it has been demonstrated empirically that the lack of a good Non-Functional Transmission estimate restricts accuracy to a correlation of less than 0.70. However, a good estimate of Non-Functional Transmission requires information about the association between functionality and time. This implies that any Non-Simultaneous Activity change estimator based upon a decoupling of time and functionality has severely limited accuracy.

Changes in Simultaneous Reduction are generally less significant than the optimality attainable from changes in Non-Simultaneous Activity. However, it is more suited to prediction by a model of activity which dissociates time and functionality. A model of this form has been proposed and tested. The accuracy of this model in predicting Simultaneous Reduction is high with a correlation of 0.96. It has also been shown that lack of good transition sign estimation severely degrades the usefulness of this estimate. This limits the simplicity of suitable models.

The best delay-sensitive estimation technique proposed has a total computational complexity of $0(|\Lambda_n|!.\sum_{n_i \in \Lambda_n} TT_{n_i})$ where $|\Lambda_n|$ is the total number of inputs to node n, TT_{n_i} the number of possible transition times on input n_i. This is compared to the total number of BDD operation required for full simulation of all possible delay configurations which is $O((\sum_{n_i \in \Lambda_n} TT_{n_i})!/(\prod_{n_i \in \Lambda_n} TT_{n_i}!))$. The coefficient of correlation for this model averaged over all the benchmark circuits is 0.85. However, it has been shown that the applicability of the model can vary significantly between specific networks and that sensitivity for detection of optimality is insufficient for practical use.

In conclusion, through the construction and test of these estimation strategies it has been verified that the decoupling of time and functionality reduces the ability to detect optimal delay conditions beyond acceptable limits. Furthermore, the amount of optimality which can be obtained, and the complexity of simulating to find optimal delay conditions eliminates the value of further research in this area. It is therefore justified to ignore sensitivity to delay at the input cutset and output nodes of a resynthesis region.

Chapter 6

Conclusions

In this dissertation the concept of estimating transition activity effects in a restricted information environment is presented. The concepts outlined are specifically applicable for guiding resynthesis algorithms which target reduced power consumption for CMOS combinational networks. The accuracy of the estimation strategies has statistically validated the dominance of changes in different circuit parameters during restructuring.

In the process of resynthesis, it is necessary to select network regions for restructuring based upon global circuit properties. Local restructuring can influence the zero delay (functional) and delay dependent (spurious dynamic) activity. The use of the Observability Don't Care (ODC) set during resynthesis can, in turn, influence the functional activity throughout the transitive fanout (TFO) of the resynthesis region, as well as the spurious dynamic activity due to changes in node sensitivities. The TFO activity is affected also by changes in the circuit structure within the resynthesis region which affect the amount and functionality of the spurious dynamic activity at the region outputs. There may also be changes in signal arrival times which are a consequence of changes in the timing of paths internal to the resynthesis region, or changes in the capacitive load of the region inputs on the network. The spurious dynamic activity at any dependent parts of the network could change as a result. The material presented in this dissertation is a detailed theoretical and empirical study of all these effects.

The determination of whether a network region is an good choice for restructuring is made before resynthesis can be performed. The definition of the most non-optimal regions allows construction of the compatible ODC to be directed for maximizing resynthesis freedom. However, without actually performing resynthesis the optimal change in function-

ality, or internal structure, cannot be predicted exactly. A cost function for region selection should therefore incorporate statistical properties extracted from observing a specific resynthesis algorithm operating upon a large collection of networks. Global sensitivities to delay, functionality and the amount of spurious dynamic activity are measures of how constrained a resynthesis operation must be.

The sensitivity to changes in functional activity requires that an estimation strategy be based upon the expected *size* of the functional change at a region outputs as exact change in function which will result from resynthesis is not known *a priori*. A probabilistic estimation scheme based upon predicting the global change in functional TFO activity given only information about the expected size of the change in functionality is the material of Chap. 3. This estimation technique correlates extremely well with simulation with a correlation coefficient of 0.99 for circuits with input switching probabilities of 0.5. The technique assumes independence between the functionality of the change, and the functionality of the TFO nodes (except in so far as that the change is made within the ODC set). This is a generally valid assumption for resynthesis operations. The small standard deviation observed is a consequence of the probability distribution corresponding to the overlap of a set describing the random change in functionality relative to the fixed sensitivity sets throughout the TFO. This distribution is very strongly peaked about the average implying that a prediction of expected change in activity correlates extremely well with most specific changes. The results presented in Chap. 3 for zero-delay power show that TFO effects can offset a decrease in power at the output of a resynthesis region in 21% of total cases. This reduces viable resynthesis options. On the other hand, it is also shown that a local increase in power can be offset by a global decrease (9%), and that a local improvement can be enhanced by a factor of 5 when considering TFO effects (6%). All of these effects are accurately predicted using the simple estimation strategy proposed. The estimation technique has been generalized for networks with arbitrary input switching probabilities. This consists of a technique for partitioning the space into regions of similar minterm probabilities. Within each region the standard deviation of the estimator results will compare to that shown for networks for which the uniform 0.5 input switching probabilities assumption holds. The ODC can then be optimized in such a way as to emphasize the flexibility within each partition for which a beneficial TFO influence is predicted. The change in function at the output of a resynthesis region also affects sensitivities, and this change can be predicted in an analogous fashion to the change in TFO functionality.

The sensitivity to changes in the spurious dynamic activity determines the form of path imbalance within the resynthesis region which is tolerable. If there is a high sensitivity, then the resynthesis step has to attempt to minimize this variable either through balancing of path timing, or functional elimination of spurious activity. Chap. 4 contains the results of an analysis of this problem. As the results of resynthesis are unknown it is necessary to establish a sensitivity based upon the *amount* of spurious dynamic activity at the outputs, not its functionality. The estimation strategy proposed is based upon a very simple transition estimation technique which assumes input functional and temporal independence. Although these assumptions are generally not sufficient for accurate power estimation, they are appropriate in estimating activity changes during resynthesis. Resynthesis incrementally adjusts the network so the results of a single accurate activity simulation completed prior to resynthesis can be used to scale probabilities for the simple activity model. This then provides an estimation technique which is delay independent, and embodies terms for both the sensitivity to spurious dynamic activity, and the sensitivity to changes in sensitivities throughout the TFO (resulting from a functional change). Again, this technique correlates well with simulation producing an overall correlation coefficient of 0.93. The worst correlation on a specific network for this estimator is 0.90. Similar to the functional sensitivity estimation, there does not appear to be a strong influence of circuit size upon estimator accuracy. The accuracy of this delay independent estimation supports the conclusion that sensitivity to changes in function and the amount of spurious dynamic activity at the output of a resynthesis region dominate delay effects when predicting global changes in network activity. Combined with the functional activity estimation strategy, the prediction of the total change in power throughout the TFO with a delay insensitive estimator is 0.97.

The sensitivity to changes in delay defines how critical it is for path timing within the resynthesis region to meet specified bounds. Furthermore, it can be used to define the effect of loading at the region inputs. Not only is this the least dominant of the effects, it is also the most complex to handle. In Chap. 4 this problem is addressed by assuming knowledge about the amount of spurious dynamic activity at the outputs, and bounds on the signal arrival times corresponding to changes in state. Ensuring that more detailed information is known would be very constraining upon the form of resynthesis. It is shown that this additional complexity actually results in an estimator which performs *worse* than the delay insensitive estimate. This is a consequence of having to estimate the changes in

many correlated terms of the describing expression. In this case, it is better to construct an estimate by scaling the results of a simpler approximation technique rather than suffering the error accumulation from the more complex approximation. Given that bounds on the delay are insufficient to describe delay sensitivity, the question then arises as to what knowledge of the timing and functionality is required to estimate delay sensitivity accurately, the subject of Chap. 5. There are two aspects to be considered relative to delay sensitivity: Functional Transmission and Simultaneous Switching. Functional transmission is the effect whereby the initial or final *state* of an input to a node ensures that spurious activity is guaranteed to, or not to, pass. As optimality related to this effect depends upon signal arrival time inequalities, this could be a feasible source of optimality. Simultaneous switching, on the other hand, is a property of exact equality in the delay of convergent paths. This is a constraint which is too difficult to satisfy with general layout heuristics. In Chap. 5, it is shown that even just the decoupling of time and functionality is sufficient to lose significant estimator accuracy for functional transmission (correlation coefficient of around 0.75). Consequently, sensitivity of the output of the resynthesis region to delay (where change in functionality and exact timing cannot be predicted) is not a feasible computation during resynthesis. However, it is *also* shown that the influence of delay insertion at the inputs of a node upon its output activity (excluding simultaneous switching) affects the activity by less than 10% in eighty-five percent of possible cases, less that 20% in ninety-four percent of possible cases. The overall probabilistic estimation strategy proposed which combines functional transmission and simultaneous switching estimates achieves a correlation coefficient of better than 0.80 on most circuits. However, optimality can only be detected by testing all possible delay conditions. The magnitude of the effect and the complexity of the estimation required eliminates the need to formulate delay sensitivity in the detection of highly non-optimal regions for resynthesis.

6.1 Future Work

The material of this dissertation can be summarized as the development and test of three estimation strategies useful for resynthesis for low-power. Considering relative importances of the effects analyzed and the complexity of the estimation required, two of these should prove to be useful strategies for construction of global cost functions.

The functional activity estimator can be used to determine the expected influ-

ence on zero-delay power consumption throughout the TFO when a node is resynthesized. Combined with statistical properties relating the local functionality and bounds on the compatible ODC set, the non-optimality of a node can be defined. Clusters of these nodes can be collapsed into viable resynthesis regions, and the construction of the ODC directed in such a way as to maximize flexibility. This concept is outlined in detail in Chap. 3.6. The construction of the ODC can also be directed according to the optimality within different partitions of the Boolean space, each partition corresponding to minterms of similar probability. The results of this analysis will statistically establish the complexity/accuracy tradeoff corresponding to the definition of the partitions. (Definition of these partitions to minimize error is the subject of Chap. 3.5).

During each step of a resynthesis procedure, choices which are identical relative to the zero-delay activity cost function can be separated by their estimated impact on spurious dynamic activity by using the delay-insensitive dynamic activity estimation strategy. The spurious dynamic sensitivity should not be primary in the decision making process, however, due to the inherent limitations of the timing model. The functional activity and delay-independent dynamic sensitivity estimation strategies can be modified to encompass the properties of specific resynthesis approaches. If the resynthesis is known to perform only limited changes relative to the original function, then the probability space describing the range of possible changes can be restricted to improve accuracy.

The probability formulations presented here need to be extended to the domain of sequential networks. In particular, the problem of delay sensitivity should be examined further to determine whether the relationship between states makes the functional transmission effect more dominant. If not, then it can generally be stated that sensitivity to delay is a minor effect in the formulation of a global cost function for low-power resynthesis of CMOS logic.

Bibliography

[1] S. B. Akers. Binary Decision Diagrams. In *IEEE Transactions on Computers*, Vol. C-27, pages 509–516, June 1978.

[2] R. I. Bahar, G. D. Hactel, E. Macii, F. Somenzi. A Symbolic Method to Reduce Power Consumption of Circuits Containing False Paths. In *Proceedings of the Int'l Conference on Computer-Aided Design*, pages 368–371, November 1994.

[3] R. K. Brayton, R. Rudell, A. Sangiovanni-Vincentelli, A. R. Wang. MIS: A Multiple-Level Optimization System. In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. CAD-6, No. 6, pages 1062–1081, November 1987.

[4] R. E. Bryant. Graph-Based Algorithms for Boolean Function Manipulation. In *IEEE Transactions on Computers*, Vol. C-35, No. 8, pages 677–691, August 1986.

[5] R. Burch, F. Najm, P. Yang, and T. Trick. McPOWER: A Monte Carlo Approach to Power Estimation. In *Proceedings of the Int'l Conference on Computer-Aided Design*, pages 90–97, November 1992.

[6] A. P. Chandrakasan, M. Potkonjak, R. Mehra, J. Rabaey, R. W. Broderson. Optimizing Power Using Transformations. In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 14, No. 1, pages 1–20, January 1995.

[7] A. Ghosh, S. Devadas, K. Keutzer, and J. White. Estimation of Average Switching Activity in Combinational and Sequential Circuits. In *Proceedings of the* 29^{th} *Design Automation Conference*, pages 253–259, June 1992.

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

[9] D. Hodges and H. G. Jackson. *Analysis and Design of Digital Integrated Circuits.* McGraw Hill, 1988.

[10] S. Iman, M. Pedram. Multi-Level Network Optimization for Low Power. In *Proceedings of the Int'l Conference on Computer-Aided Design*, pages 372–377, November 1994.

[11] M. W. Kallu. *ENPOWER: A Vector-Driven Power Estimating Tool for CMOS Logic Circuits.* Masters Thesis, U.C. Berkeley, December 1993.

[12] P. Landman. *Low-Power Architectural Design Methodologies.* Ph.D. Dissertation, ERL Memo. No. M94/62, U.C. Berkeley, August 1994.

[13] C. K. Lennard, A. R. Newton. An Estimation Technique to Guide Low Power Resynthesis Algorithms. In *Proceedings of the Int'l Symposium on Low Power Design*, pages 227–232, April 1995.

[14] B. Lin and H. De Man. Low-Power Driven Technology Mapping under Timing Constraints. In *Proceedings of the Int'l Workshop on Logic Synthesis (IWLS-93)*, May 1993.

[15] F. Najm. Transition Density, A Stochastic Measure of Activity in Digital Circuits. In *Proceedings of the* 28^{th} *Design Automation Conference*, pages 644–649, June 1991.

[16] K. Roy, S. C. Prasad. Circuit Activity Based Logic Synthesis for Low-Power Reliable Operations. In *IEEE Transactions on VLSI Systems*, Vol. 1, No. 4, pages 503–513, December 1993.

[17] H. Savoj. *Don't Cares in Multi-Level Network Optimization.* Ph.D. Dissertation, ERL Memo. No. M92/122, U.C. Berkeley, October 1992.

[18] H. Savoj, R. K. Brayton. The Use of Observability and External Don't Cares for the Simplification of Multi-Level Networks. In *Proceedings of the* 28^{th} *Design Automation Conference*, pages 297–301, June 1990.

[19] E. M. Sentovich, K. J. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. R. Stephan, R. K. Brayton, A. Sangiovanni-Vincentelli. *SIS: A System for Sequential Circuit Synthesis.* ERL Memo. No. M92/41, U.C. Berkeley, May 1992.

[20] A. Shen, S. Devadas, A. Ghosh, and K. Keutzer. On Average Power Dissipation and Random Pattern Testability of Combinational Logic Circuits. In *Proceedings of the Int'l Conference on Computer-Aided Design*, pages 402–407, November 1992.

[21] K. J. Singh, A. Wang, R. K. Brayton, A. Sangiovanni-Vincentelli. Timing Optimization of Combinational Circuits. In *Proceedings of the Int'l Conference on Computer-Aided Design*, pages 282–285, November 1988.

[22] V. Tiwari, P. Ashar, and S. Malik. Technology Mapping for Low Power. In *Proceedings of the* 30^{th} *Design Automation Conference*, pages 74–79, June 1993.

[23] C-Y. Tsui, M. Pedram, A. M. Despain. Efficient Estimation of Dynamic Power Consumption under a Real Delay Model. In *Proceedings of the Int'l Conference on Computer-Aided Design*, pages 224–228, November 1993.

[24] C-Y. Tsui, M. Pedram, and A. M. Despain. Technology Decomposition and Mapping Targeting Low Power Dissipation. In *Proceedings of the* 30^{th} *Design Automation Conference*, pages 68–73, June 1993.

[25] S. Yang. *Logic Synthesis and Optimization Benchmarks User Guide Version 3.0.* MCNC Technical Report, Research Triangle Park, NC, January 1991.