3. SIMULATION WITH BOUNDS

There is a well known law of nature, attributed to someone named Murphy, that is often cited in jest to explain unfortunate events. By this law, the most pessimistic predictions of the future always come to pass. Although this law may appear to have some validity at times, due to human psychology, fortunately our world is not exclusively governed by it. When an attempt is made to generate rigorous bounds on anything, one enters a domain in which Murphy's law reigns supreme. The most pessimistic prediction is, by definition, the desired result. The worst case error is always used. The effects of errors never cancel, but instead they always accumulate. Simplifications always produce the worst possible effect. As a result, to be useful bounding must be undertaken with great care.

There is reason to believe, however, that bounding techniques are feasible in digital VLSI circuits. Rough bounds have been used successfully in digital circuits since the appearance of the first integrated logic gates. Data sheets have always listed minimum and maximum times required to produce "valid" outputs. Logic designers have routinely used these to generate meaningful bounds on the behavior of their circuits. While care must still be taken in useful bound generation for more complex circuit models, there is reason for optimism in this pessimistic domain, at least when considering digital circuits.

There is no fundamental reason why bounds can not be made arbitrarily tight in the absence of uncertainty. For a problem such as digital MOS circuit

2The completion of this book serves as a counterexample to Murphy's law [38].
simulation, where the output is not highly sensitive to input uncertainty, small simplifications can theoretically produce very tight bounds. In practice, though, efficient bounding algorithms are often difficult to generate. A simple example can illustrate why some calculations are more difficult to bound efficiently than others [39]. Consider first the calculation \( Y = X - X \), with an output \( Y \) that is very insensitive to uncertainty in the input \( X \). If the variable \( X \) is known to lie in the interval \([0,1]\), a straightforward bound on the subtraction operator will conclude only that \( Y \) must lie in the interval \([-1,1]\). By ignoring any "correlations" between the two operands of the subtraction, bounding is made feasible but information is lost. Here the practice of ignoring correlations amplifies uncertainty. A calculation that does not exhibit a correlation problem is \( Y = X + X \). Knowledge that the two operands must be identical does not change the conclusion that \( Y \) must lie in the interval \([0,2]\) if \( X \) lies in \([0,1]\).

When bounding is used at the level of each arithmetic operation in a general algorithm, the results are often disappointing due to the effect of correlations. When digital circuits are considered at the level of logic signal waveforms, the correlations that are ignored in an efficient analysis do not generally cause major problems. The effect of correlations is considered in detail throughout this chapter. For the most part, the adder example serves as a better analogy for digital MOS simulation than the subtractor. Hence, efficient bounding algorithms are possible. While the correlation problem is manageable for digital circuits, bounding algorithms should be designed with the important correlation effects in mind.

This chapter serves as an overview of the use of bounds in digital MOS simulation, including the specific approach to bounding taken in this book. Both theoretical and algorithmic details are postponed until later. In the first section useful bounds are defined for circuit behaviors, circuit models, and circuit inputs. The second section examines the bounding strategy proposed in the book. The third section considers the performance of the bounding strategy on various types of MOS subcircuits. The fourth section presents applications for a bounding simulator such as uncertainty management and worst case analysis, and demonstrates that a bounding approach is becoming essential for low-cost, reliable simulation.

### 3.1 Bound Definition

In an abstract sense, a bound defines a subset of a larger set in which an element must lie. Consider the set of all positive real numbers, representing