Chapter 18

The Synthesis of Stochastic Circuits for Nanoscale Computation

Weikang Qian
University of Minnesota, USA

John Backes
University of Minnesota, USA

Marc D. Riedel
University of Minnesota, USA

ABSTRACT

Emerging technologies for nanoscale computation such as self-assembled nanowire arrays present specific challenges for logic synthesis. On the one hand, they provide an unprecedented density of bits with a high degree of parallelism. On the other hand, they are characterized by high defect rates. Also they often exhibit inherent randomness in the interconnects due to the stochastic nature of self-assembly. We describe a general method for synthesizing logic that exploits both the parallelism and the random effects. Our approach is based on stochastic computation with parallel bit streams. Circuits are synthesized through functional decomposition with symbolic data structures called multiplicative binary moment diagrams. Synthesis produces designs with randomized parallel components—and operations and multiplexing—that are readily implemented in nanowire crossbar arrays. Synthesis results for benchmarks circuits show that our technique maps circuit designs onto nanowire arrays effectively.

1 INTRODUCTION

As the semiconductor industry contemplates the end of Moore’s Law, there has been considerable interest in novel materials and devices (IRTS, 2006). Technologies such as molecular switches and carbon nanowire arrays offer a path to scaling beyond the limits of conventional CMOS (FENA, 2006). Most such technologies are in the exploratory phases, still years or decades from the point when they will be actualized. Accordingly, the development of software tools and techniques for logic synthesis remains speculative.

DOI: 10.4018/978-1-60960-186-7.ch018
And yet, for some types of new technologies, we can identify broad traits that will likely impinge upon synthesis. For instance, nanowire arrays are stochastically self-assembled in tightly-pitched bundles. Accordingly, they exhibit the following (DeHon, 2005):

1. A high degree of parallelism.
2. Minimal control during assembly.
3. Inherent randomness in the interconnect schemes.
4. High defect rates.

Existing strategies for synthesizing logic for nanowire arrays are based on routing schemes similar to those used for field-programmable gate arrays (FGPAs) (DeHon, 2005). These rely on probing the circuit and programming interconnects after fabrication.

We describe a general method for synthesizing logic that exploits both the parallelism and the random effects of the self-assembly, obviating the need for such post-fabrication configuration. Our approach is based on stochastic computation with parallel bit streams. Circuits are synthesized through functional decomposition with symbolic data structures called multiplicative binary moment diagrams. Synthesis produces designs with randomized parallel components -- AND operations and multiplexing -- operating on the stochastic bit streams. These components are readily implemented in nanowire crossbar arrays. We present synthesis results for benchmarks circuits illustrating the method. The results show that our technique is effective in implementing designs with nanowire arrays, with a measured tradeoff between the degree of redundancy and the accuracy of the computation.

2 CIRCUIT MODEL

Our discussion of synthesis is framed in terms of a conceptual model for nanowire arrays. (In the later part of the paper, we justify this model with implementation details.) A nanowire crossbar is illustrated in Figure 1. The connections between horizontal and vertical wires are random. However, we assume that these connections are nearly one-to-one, that is to say, nearly every horizontal wire connects to exactly one vertical wire, and vice-versa. This is a specific attribute of types of nanowire arrays, controlled during self-assembly (DeHon, 2005).

2.1 Parallel Stochastic Bit Streams

Our synthesis method implements digital computation in the form of parallel stochastic bit streams. We refer to a collection of parallel nanowires as a bundle. The width of a bundle is the number wires. Its current weight is the number of logical 1’s on its wires. The signal that it carries is a real value between zero and one corresponding to the fractional weight: for a bundle of \( N \) wires, if \( k \) of the wires are 1, then the signal is \( k / N \). Let \( P(X = 1) \) denote the probability that any given wire in bundle \( X \) carries a 1.

2.2 Shuffling Devices

We implements computation with two basic nanowire constructs: shuffled ANDs and Bundle-plexers. We describe these only in conceptual terms.