A Systematic Design and Explanation of the Atrubin Multiplier

Shimon Even and Ami Litman

Computer Science Dept., Technion, Haifa, Israel 32000
and
Bellcore, 445 South St., Morristown, NJ 07960-1910

ABSTRACT

The Atrubin systolic array, for multiplying two serially supplied integers in real-time, was invented in 1962, but to date, no simple explanation of its operation, or proof of its validity, has been published.

We present a methodical design of the array which yields a simple proof of its validity. First, we use a broadcast facility, and then we show how it can be removed by retiming which avoids the introduction of either slow-down or duplication.

A similar retiming technique can be used to remove instantaneous accumulation. These retiming methods are applicable to arrays of any dimension.

1. Introduction

In 1962, Allan J. Atrubin invented a synchronous system for real-time multiplication of integers. (It was published in 1965, [A].) The host (user) feeds the system two binary encoded multiplicands, \(x\) and \(y\), serially, least significant bits first, and the system outputs the product \(x \cdot y\), in binary, serially, least significant bit first. Clearly, the time it takes to multiply two \(n\)-bit multiplicands is \(2n\).

Informally, a (finite, or infinite) synchronous system, serving a host, is called systolic, if it has the following characteristics. The system consists of segments, connected to each other and to the host by communication lines. Each segment consists of a modest amount of hardware, which realizes a Moore finite state automaton; i.e. its current output signals, which appear at its output ports, depend only on its present state, and its next state depends on the present state and the present input signals, which appear presently at the input ports. Without loss of generality, we may assume that each output port of a segment is the output port of a (clocked) delay flip-flop. The lines go from one output port to one input port; there is no fan-in or fan-out in these connections.

Systolic systems are free of long paths on which a rippling effect can occur between clock ticks. In fact, if a clock rate is not too high for each single segment, then it is not too high for the whole system, no matter how large the system is.
The Atrubin system is systolic. Furthermore, its segments are all identical, and are arranged in a linear array. This simplifies the design and production of the system.

It is well known, [W], how to perform multiplication in \( O(\log n) \) time, but the equipment required is of size \( O(n^2) \). The asymptotically smaller circuit, [SS], of size \( O(n \cdot \log n \cdot \log \log n) \), which performs multiplication in \( O(\log n) \) time, is not a practical alternative. Thus, the Atrubin multiplier, which requires equipment of size \( O(n) \), remains competitive.

The general layout of the multiplier is depicted in the following diagram. In order to multiply \( n \)-bit numbers, the array must consist of at least \( \left\lceil \frac{n}{2} \right\rceil \) cells. All cells have the same structure. Each consists of a few hundred transistors, and realizes a finite Moore automaton.

Recently, Even showed how, with the addition of another systolic array, repeated modular multiplication can be computed almost as fast ([EI]). The combination of the two arrays may be useful in cryptographic applications.

The Atrubin Multiplication Array

In spite of its reputation, the structure of the Atrubin array has remained a mystery. It is the purpose of this paper to explain this mystery away, and prove the validity of the multiplier. This is done by breaking the design into stages.

First, the operation and validity of a multi-input serial adder is discussed. Next, a simplified version of the multiplier is studied, in which a broadcast facility is used. Finally, by using retiming, the systolic version is achieved.

Finally, we show that our design methodology is helpful in building other useful systolic arrays. This is true for systems in which it is natural to use a broadcast facility in the intermediate design stage, as well as for systems in which it is natural to use instant-accumulation. The technique can be used, not just for linear arrays, but for arrays of any dimension. It is interesting to note that contrary to the general case, [EL], no duplication of hardware is necessary for the removal of broadcast or instant-accumulation from arrays.