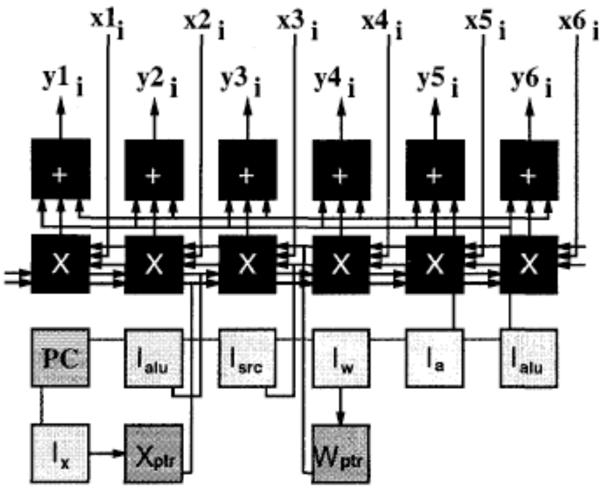
MATRIX: A Reconfigurable Computing Architecture with Configurable Instruction Distribution and Deployable Resources

Ethan Mirsky, André DeHon



Year of publication: *1996*
Area: *Architecture and Technology*

This paper describes a coarse-grain, reconfigurable computing array that has been highly cited by subsequent work using the coarse-grain approach.

The novelty of this work is that the fundamental array element can be configured as control logic, memory, or datapath logic, providing the flexibility to implement the functionalities where they are needed. To illustrate the flexibility of the array, it is shown how a simple convolution can be implemented using several architectures: systolic, microcoded, VLIW and a hybrid multi-SIMD/VLIW.

More significantly, the work is representative of an era when there was significant activity devoted to proposing new architectures for reconfigurable computing. VLSI technology was still simple enough that universities could contemplate and prototype significant designs with graduate student labor within the timeframe of a graduate degree. In this paper, the layout of one of the array elements is done using a 0.5µm technology, which then provides reliable information about the size of the block and a reasonable estimate of performance. This is important because it is information that can be presented to venture capitalists as validation that it is worthy to fund a new company. Many startups followed this model. Few survived. The work in this paper led to the founding of Silicon Spice, Inc., which may be the most successful reconfigurable computing company if measured by its peak value. Silicon Spice was acquired by Broadcom Corp. in August 2000 for $1.2 billion during the height of the *Dot Com* bubble at the turn of the century.

As an architecture paper, this paper lacks any quantitative comparisons to alternative architectures that would validate the benefits of this architecture. It only provides a qualitative evaluation using a simple application. In today's world, this paper would also need some discussion of the software needed to map to this architecture.

Paul Chow

DOI: http://dx.doi.org/10.1109/FPGA.1996.564808