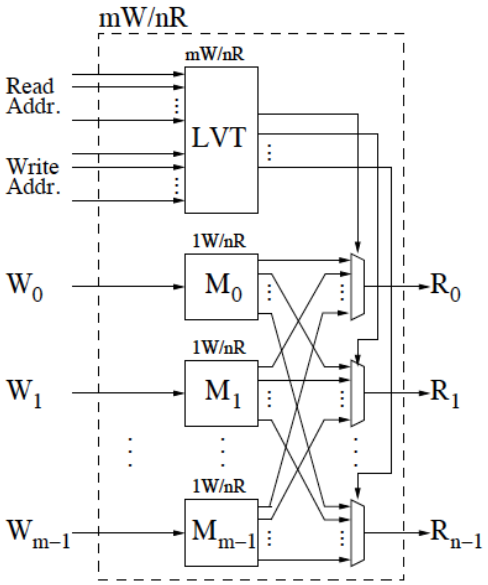
Efficient Multi-Ported Memories for FPGAs

Charles Eric LaForest,
J. Gregory Steffan

Year of publication: *2010*
Area: *Applications*



One of the key features of an FPGA is the ability to provide a large number of simple building blocks, which can be cascaded together to form ever larger and more powerful elements as needed. While the logic block in an FPGA may only support 4 to 6 inputs each, a tree of LUTs can support an arbitrary number of inputs for any desired computation. An individual flipflop can store a single bit of information, but can be ganged together to support 32-bit data, and these elements sequenced into a multi-stage FIFO.

As FPGAs evolved, they started adding fixed components for specific tasks, such as multipliers, memories, and even entire processor cores. These radically improved the implementation of specific common computations, but their relative inflexibility makes them harder to cascade into larger computations. If the hard block does not have the right set of features, we may be forced to abandon those blocks, and go back to cobbling together LUTs and flipflops. A good example of this problem is embedded hard memory blocks. If the memory unit has too little total storage capacity, or too few bits per address, we can easily group multiple blocks to provide the appropriate memory structure. However, the number of access ports to these memories is fixed – if the memory blocks allow two writes a clock cycle, yet our applications needs three, there is little that can be done. Extra read ports are easily supported by duplicating the memory, but for RAMs the number of independent write ports is seemingly a fixed bound.

This paper fundamentally changed this tradeoff. By realizing that the underlying hardware can often operate much faster than the user's actual design, memory ports can be time-multiplexed to increase the number of independent accesses. Also, by clever indirection mechanisms, we can gang together multiple memories to support increased read and write bandwidth, while maintaining control information to find the proper live value. Combined, this gives an effective mechanism for providing increased memory ports for many different applications.

I still recall seeing this work presented for the first time. "Damn, I wish I thought of that!" was my immediate reaction. This is a problem that many of us have faced, but it took LaForest and Steffan to recognize how to overcome this fundamental problem and significantly increase the capabilities of modern FPGAs via a simple memory generator structure.

Scott Hauck

DOI: http://doi.acm.org/10.1145/1723112.1723122