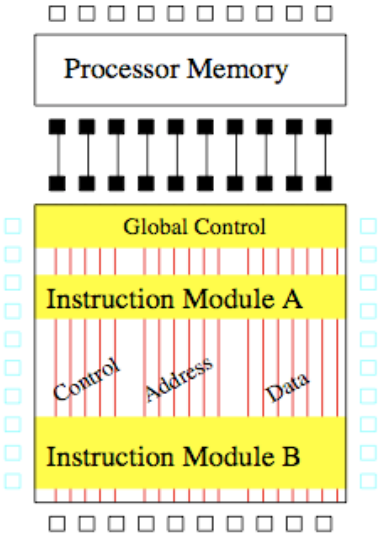
A Dynamic Instruction Set Computer

Michael J. Wirthlin, Brad L. Hutchings

Year of publication: *1995*
Area: *Run-Time Systems and Run-Time Reconfiguration*



The year 1995 was early in the history of reconfigurable computing research. Commercial FPGAs had existed for only a decade, and the idea of partial run-time reconfiguration was even newer. The DISC project, and this paper in particular, brought together a number of concepts that are pervasive throughout reconfigurable computing research, and did so using a fully-implemented system.

One of the key contributions of this paper is the idea of treating the two-dimensional FPGA area as a one-dimensional array of custom instructions, which the paper refers to as *linear hardware space*. Each custom instruction occupies the full width of the array but can occupy different heights to allow for instructions of varying complexity. Instructions also have a common interface that includes pass-thrus which connect to the previous and next instructions in the array. This communication interface greatly simplifies the problem of *relocating* a custom instruction to a free area, allowing other instructions to potentially remain in-place. The communication paths for an instruction are always at the same relative positions, and run-time CAD operations (and their corresponding overheads) are completely avoided. Thus, the linear hardware space reduces the overheads and simplifies the process of run-time reconfiguration of custom processor instructions.

Beyond the linear hardware model, the DISC system is notable because it takes the idea of custom processor instructions to the extreme: *all* instructions are custom instructions, swapped in and out of hardware by a global controller located within the processor. Even instructions that might be considered “standard”, such as add/subtract, are swapped in and out of the hardware as needed. Although this particular approach is not often taken by more modern research, it was motivated by a drive to explore the capabilities (and effects) of run-time reconfiguration.

As part of this exploration, the DISC paper presented one of the first studies of run-time reconfiguration overheads, concluding that the overhead was approximately 71%, but could be reduced to 16% with “maximized” configuration speeds. Although this demonstrates that the overhead of treating all instructions as custom reconfigurable instructions is perhaps too high, these results, coupled with the idea of the linear hardware model, provided a simple conceptual model and a powerful motivating force for future reconfigurable computing research.

Katherine Morrow

DOI: http://dx.doi.org/10.1109/FPGA.1995.477415