

Quasi-Static Scheduling for SCORE

Masters of Science Report

Yury Markovskiy

Abstract

Previous work¹ introduced a dynamic compute model aimed at eliminating existing barriers to the widespread efficient exploitation of reconfigurable devices. Among other achievements this model decoupled application design-time decisions from the run-time physical resource bindings. The compute model uses graphs of compute pages and memory blocks connected by stream links to capture the definition of a computation abstracted from the detailed hardware size. An automatic run-time scheduler is a required component in this compute model in that it selects the temporal sequencing of virtual resources onto the physical device, allocates hardware resources, and configures the device. Although such a scheduler could be computationally expensive, this work describes a quasi-static scheduling strategy that dramatically reduces run-time overhead without restricting the full semantic power of the dynamic dataflow graphs. This work describes the quasi-static scheduling system, analyzes the trade-offs involved in selecting a scheduler implementation, and highlights critical algorithms. It pays particular attention to the temporal partitioning of compute graphs and the management of live computation state.

The complete report is available at

www.cs.berkeley.edu/~yurym/masters/ms_report.pdf

¹Eylon Caspi, Michael Chu, Randy Huang, Nicholas Weaver, Joseph Yeh, John Wawrzynek, and André DeHon. Stream computations organized for reconfigurable execution (SCORE): Extended abstract. In *Conference on Field Programmable Logic and Applications (FPL '2000)*, pages 605–614. Springer-Verlag, August 28-30 2000.