Hardware Module Reuse and Runtime Assembly for Dynamic Management of Reconfigurable Resources

Abelardo Jara-Berrocal
Dr. Ann Gordon-Ross

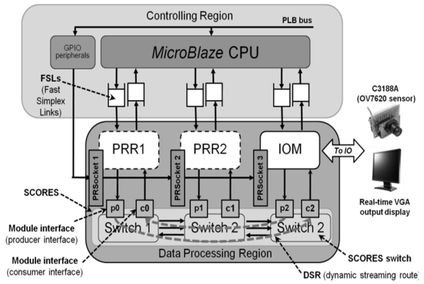
CHREC NSF Center for High-Performance Reconfigurable Computing

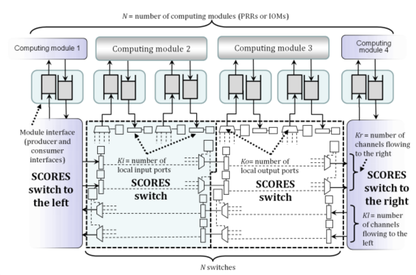
UF UNIVERSITY of FLORIDA

Introduction

- Goals
  - Efficient hardware resource management in partially reconfigurable (PR) system-on-chips (SoCs)
- Motivations
  - Enable multiple applications to concurrently execute on a PR SoC while meeting performance deadlines
- Approach
  - Leverage scalable and flexible base architecture
    - Virtual Architecture for Partially Reconfigurable Embedded Systems (VAPRES)
  - Implement dynamic resource manager (DRM) to orchestrate SoC hardware resource management
    - Improve hardware resource utilization and reduce reconfiguration time through run-time hardware adaptation

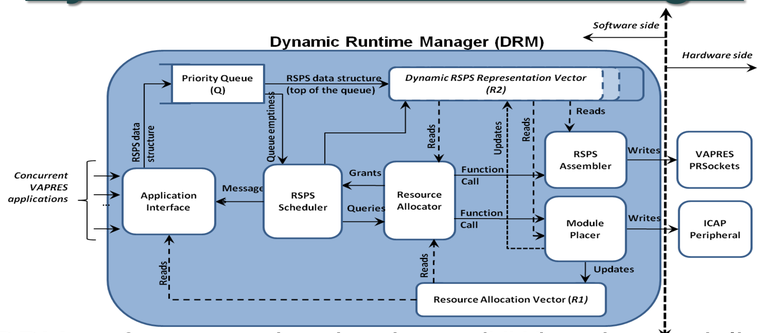
VAPRES Overview





- VAPRES is a parametric, scalable, and flexible architecture
  - Number of PR regions (PRRs), FIFO depths, PRR width & height, inter-PRR communication network
  - Inter-PRR communication using SCORES (a streaming-based dynamic inter-module communication architecture)
  - PR modules (PRMs) operate at different clock frequencies
  - Streaming communication with asynchronous FSLs

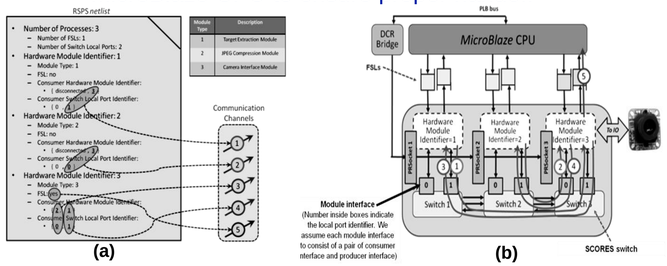
Dynamic Resource Manager



- DRM performs run-time hardware (HW) task scheduling and communication interface management
  - Caches HW modules inside PRRs
    - Enables HW reuse
    - Reduces wasted PR HW resources
  - Updates inter-PRR communication links for reconfigurable stream processing systems (RSPSs)
    - Eliminates HW relocation
    - Decreases overall reconfiguration time

RSPS Run-time Assembly

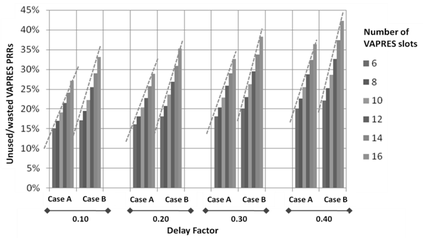
- RSPS netlists allow run-time assembly/ transformation of the inter-PRR data streams
  - Transformation done by dynamically updating SCORES communication channels
  - Allows PRMs to be seamlessly loaded/unloaded to PRRs without loss of processed data
    - DRM software modules execute inside the MicroBlaze CPU to ensure proper handoff

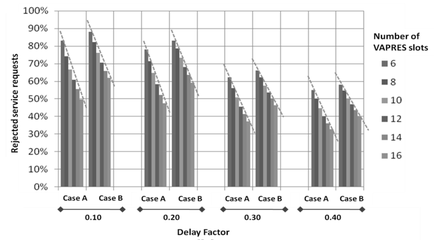


(a) Sample RSPS netlist and (b) mapping of hardware modules and communication channels to VAPRES

DRM Performance Evaluation

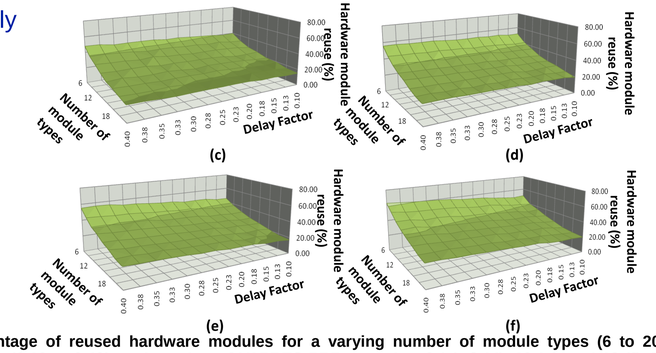
- DRM performance evaluated with respect to the percentage of rejected service requests and HW module reuse
  - Compared a DRM with HW module reuse and run-time assembly (case A) to a DRM without HW module reuse and runtime assembly (case B)
  - Lower percentage of rejected service requests translate to increased performance
    - More PRMs are loaded to PRRs over a certain amount period of time
  - Higher percentage of HW module reuse translates to reduced reconfiguration time
    - Less time reconfiguring the FPGA as PRMs can be reused by leveraging run-time assembly





Rejected service requests (a) and unused/wasted PRRs (b) for a varying number of VAPRES slots and delay factors for a DRM that leverages hardware module reuse and RSPS runtime assembly (case A) and for a DRM that does not leverage hardware modules reuse and RSPS runtime assembly (case B). The dotted lines correspond to the slope of the regression linear functions that approximate the result values.

- Average reduction times compared between case A and case B for varied number of module types
  - Reconfiguration time - 33%
  - Unused HW resources - 13%



Percentage of reused hardware modules for a varying number of module types (6 to 20), delay factors (0.10 to 0.40), and number of VAPRES PRRs equal to 6 (c), 8 (d), 10 (e), and 12 (f). For each figure, averages are calculated across all combinations of delay factor and number of module types.