A Novel O(n) Parallel Banker’s Algorithm for System-on-a-Chip

Jaehwan John Lee and Vincent John Mooney III
School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, U.S.A.
http://codesign.ece.gatech.edu/

Introduction

- Future SoC designs
  - Multiple heterogeneous processors (tens of processes)
  - Multiple on-chip hardware resources
    - DSP, FFT, MPEG, GPS, Shared-Memory, etc.
  - Examples
    - Xilinx Virtex-II Pro FPGA includes multiple PowerPC cores
    - Broadcom BCM1400 includes multiple MIPS64 cores
- Processes in such an SoC
  - Dynamically request and use resources
  - May end up in deadlock
- Current embedded systems or single processor systems
  - Today, typically ignore deadlock possibilities

Methodology

- Fully HW-oriented parallelized version of the Banker’s Algorithm
  - For multiple-instance resources
- Advantages
  - Guarantees deadlock avoidance
  - Support multiple instance resources
  - Provide O(n) run-time complexity
    - Reduced from O(n^2) in the best case
- Disadvantages
  - Require specialized hardware
  - Require maximum claim declaration
  - May under-utilize resources

Implementation

- Using Verilog HDL
  - Deadlock IP Generator
- Hardware Chip Area
  - Synopsys Design Compiler
  - TSMC .25µm technology library from Qualcore Logic
  - 0.05% of the total SoC area with five PEs and memory
  - All PBAUs able to handle up to 16 instances for each resource

Experimental Results

- Five processors
- Four resources
  - Q1: Multiple DSPs
  - Q2: Hardware semaphores
  - Q3: I/O buffers
  - Q4: A memory allocator
- PBAU 5x5
- A robotic application
  - Five processes
  - Requires multiple instances
  - 22 service requests to PBAU
    - Requests, releases and claim settings

- Performance improvement
  - 99% algorithm execution time reduction
  - 19% reduction in an application execution time
- Execution time comparison (PBAU vs. BA in software)

Integration into the δ hardware/software RTOS partitioning framework

Hardware/software RTOS/MPSoC configuration framework
- Enables automatic generation of different mixes of the HW/SW RTOS
- Can be generalized to instantiate additional HW or SW RTOS components
- Integrates parameterized IP generators such as DDU, DAU and PBAU generators
- Designed by V. Mooney, J. Lee and K. Ryu
- RTOS Components: designed by B. Akgul, P. Kuahrcoen, J. Lee, K. Ryu, M. Shalan and E. Shin

Conclusion

- Parallel Banker’s Algorithm Unit (PBAU)
  - Faster deadlock avoidance for multiple instance multiple resource systems (1600x)
  - O(n) run-time complexity with O(1) in the best case
  - Small area (less than 0.1% in our example SoC)
- Integration into the δ framework
  - With custom deadlock IP generator for a specific target

More Details