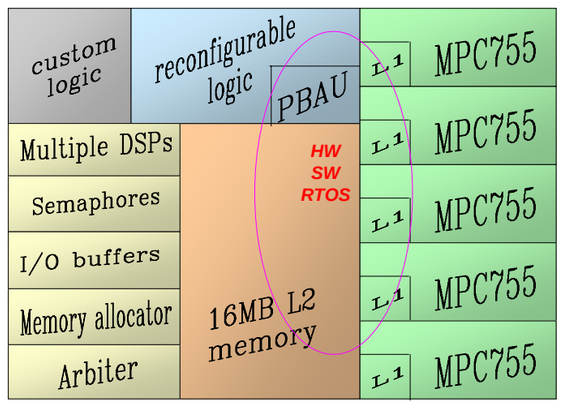
A Novel O(n) Parallel Banker's Algorithm for System-on-a-Chip

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

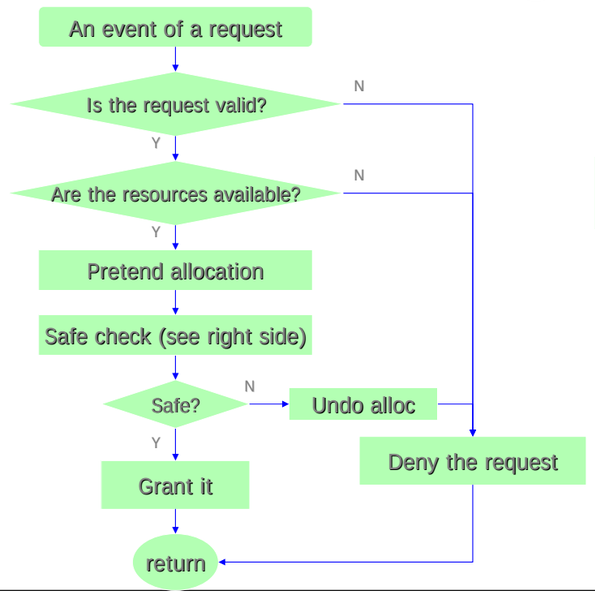
Multiprocessor system-on-a-chip (MPSoC)
A multiprocessor RTOS
An application on top of the system

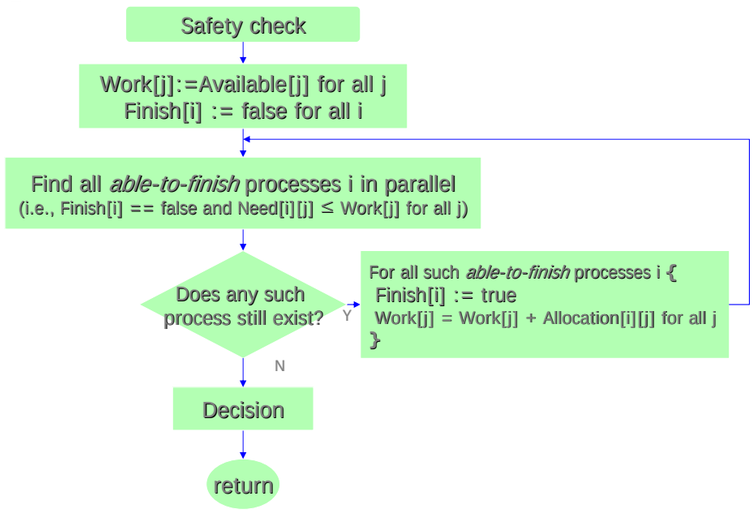


- Problem
  - How to deal with deadlock as the number of processors and resources approaches 50-100?
- Goal
  - Allow software to make requests in any order
  - Grant as many resources as possible while maintaining safety properties
  - Avoid deadlock correctly and quickly
- Solution
  - A hardware/software mechanism of deadlock avoidance, easily applicable to Real-Time Multiprocessor System-on-a-Chip (SoC) design

Methodology

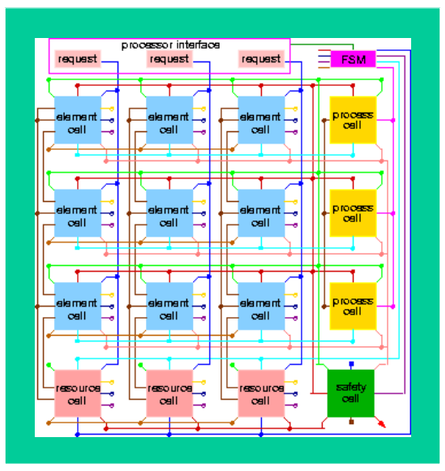
- Fully HW-oriented parallelized version of the Banker's Algorithm
  - For multiple-instance resources
- Advantages
  - Guarantee deadlock avoidance
  - Support multiple instance resources
  - Provide O(n) run-time complexity
    - Reduced from O(mn)
    - O(1) in the best case
- Disadvantages
  - Require specialized hardware
  - Require maximum claim declaration
  - May under-utilize resources



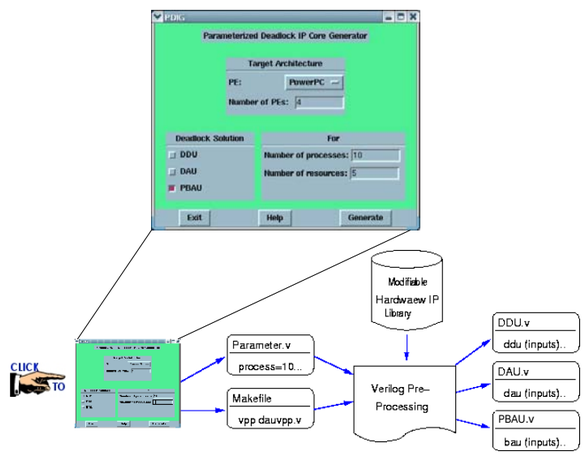


Implementation

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

Module Name	Total Area in terms of two-input NAND gates	Lines of Verilog HDL Code
PBAU 5x5	1303	600
8x8	3243	700
10x10	5030	770
15x15	11158	1000
20x20	19753	1350
MPSoC	42 Million	-

Clock period used: 4 ns

TSMC: Taiwan Semiconductor Manufacturing Company
PE: Processing Element

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

Method of Deadlock Avoidance	Algorithm Exe. Time (cycles)	Normalized Exe. Time	Application Exe. Time (cycles)
PBAU Hardware	3.32 (average)	1X	185716
BA in Software	5398 (average)	1625X	221259

- Execution time comparison (PBAU vs. BA in software)

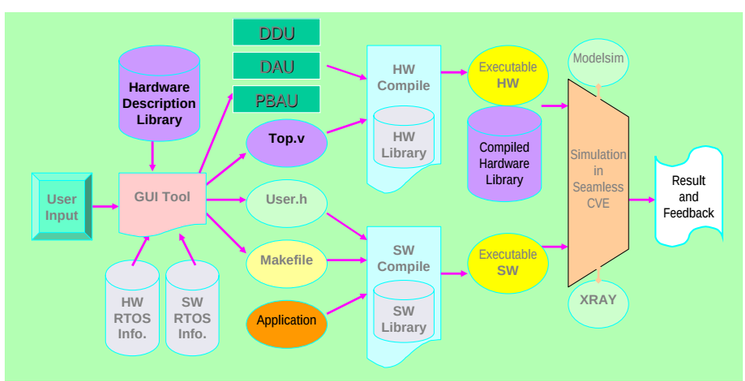
Method of Implementation	Set Available	Set Max. Claim	Request Command	Release Command	Wrong Command
# of Command	1	5	9	6	1
PBAU hardware	1	1	6.5	1	2
BA in Software	416	427	11337	2270	560

The time unit is a clock cycle, and values are averaged

Integration into the δ hardware/software RTOS partitioning framework

δ 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. Kuarchroen, 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

J. Lee, "Hardware/Software Deadlock Avoidance for Multiprocessor Multiresource System-on-a-Chip," Ph.D. thesis, Georgia Tech, Fall 2004.
http://etd.gatech.edu/theses/available/etd-11222004-083429/