Gigascale Systems Research Center

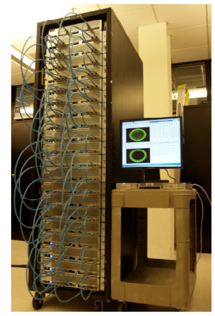
RAMP Blue: A Message-Passing Many-Core System in FPGAs

GSRC Fall Symposium 2007
September 20th, 2007

Alex Krasnov, Andrew Schultz, John Wawrzynek, Greg Gibeling, and Pierre-Yves Droz

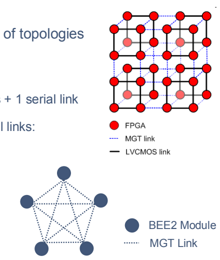
Version Highlights

- **V1: 256 cores total**
  - 8 BEE2 modules
  - 4 user FPGAs * 8 cores per FPGA
  - 100MHz Xilinx MicroBlaze soft cores running uCLinux.
  - Dec 06: 256 cores running benchmark suite of UPC NAS Parallel Benchmarks
- **V2: 768 cores total**
  - 16 BEE2 modules, 12 cores per FPGA
  - Cores running at 90 MHz
- **V3: 1008 cores total**
  - 21 BEE2 modules, 12 cores per FPGA
  - Summer 2007
- **V4: Upcoming release**
  - Written in RDL
  - Growing parameterization support
  - Waiting on external code bug fixes
- **Future versions**
  - Use newer BEE3 FPGA platform
  - Support for other processor cores.

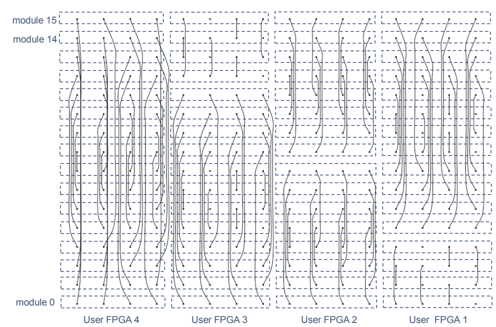


Physical Network Topology

- InterModule
  - 10Gb/s serial links permit a wide variety of topologies
  - High latency (10's of cycles)
  - All-To-All Topology
    - Used in RAMP Blue v1-v2
    - Each FPGA is at most 4 on-module links + 1 serial link connection away from any other
    - + Minimizes dependence on use of serial links:
    - - Scales only to 17 modules total
  - 3-D mesh
    - Used in newer topologies
- InterFPGA
  - High speed parallel I/O
  - Organized as a ring
  - Low latency (2-3 cycles)
- InterCore
  - All-to-all within an FPGA
  - Only 12 cores



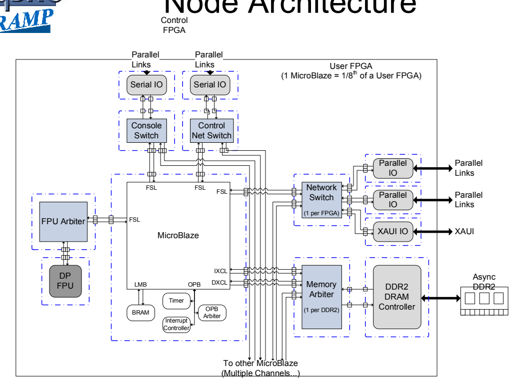
16 Module 3D Mesh



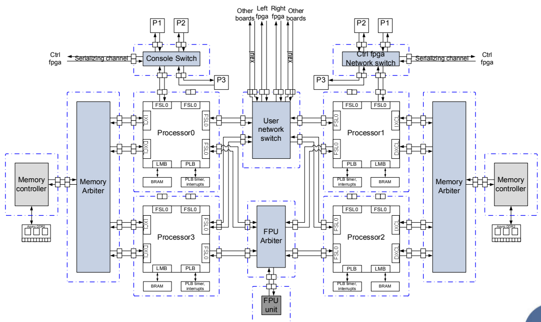
MicroBlaze v4

- 3-stage, RISC designed for FPGAs
  - Accounts for FPGA features & shortcomings
    - fast carry chains
    - lack of CAMs in cache
  - Short pipeline minimizes multiplexors in bypass logic
- Max clock rate of 100 MHz (~0.5 MIPS/MHz) on Virtex-II Pro
- Split I and D cache with configurable size, direct mapped
  - We use 2KB $I, 8KB $D)
- Optional single precision floating point unit
- Up to 8 independent fast simplex links (FSLs) with ISA support
- Configurable hardware debugging support (watch/breakpoints)
  - MDM (Microprocessor Debug Module)
- GCC tool chain support and ability to run uClinux

Node Architecture

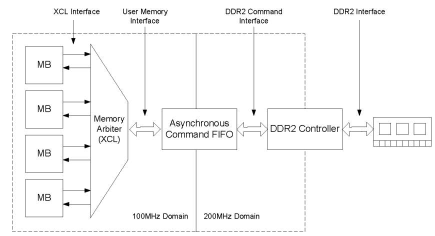


Board Architecture



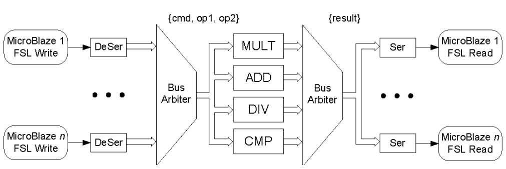
Memory System

- DIMMs are shared, Memory is not
  - More MicroBlaze cores than DIMMs
  - No coherence, DIMMs are partitioned
  - Bank management isolates cores



Double Precision FPU

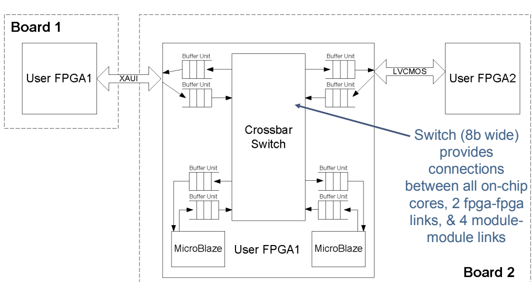
- Shared FPU
  - Necessary due to size constraints
  - Similar to "reservation stations"
- Implemented with Xilinx CoreGen library FP components



Network Characteristics

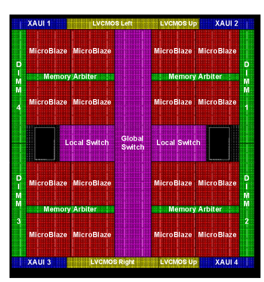
- Interface
  - Simple FSL channels
  - Currently **programmed I/O,** but could/should be DMA
  - Encapsulated Ethernet packets with source routes prepended
- *Source routing* between nodes
  - dimension-order style routing, non-adaptive ⇒ link failure intolerant
  - FPGA-FPGA and board-to-board link failure rare (1 error / 1M packets)
  - CRC checks for corruption, software (GASNet/UDP) uses acks/time-outs and retransmits for reliable transport
- Cut through flow control with virtual channels for deadlock avoidance
- Topology of interconnect
  - Full cross-bar on chip
  - Ring on module
  - Intermodule: all-to-all (v1-2), 3D mesh (v3-on)

Network Implementation



(In current versions, processor connection to network is programmed I/O limiting performance.)

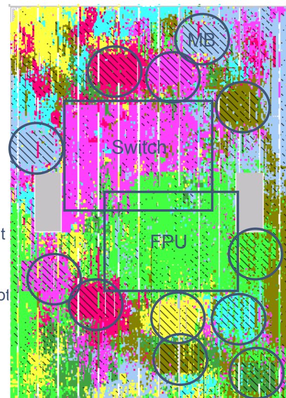
Early "Ideal" Floorplan



- 16 Xilinx MicroBlazes per XC2VP70 FPGA
- Each group of 4 clustered around a memory interface
- FP Integrated with each core

12 Core

- FPU size/efficiency
  - Shared block
- Scaling is resource constrained
  - 16 cores would fit without infrastructure (network, FPU)
- Floorplanned
  - FPU and switch placement
  - Uses roughly 93% of logic blocks, 55% BRAMs.
- Place and route to 100HMz not practical
  - many PAR builds
  - Currently running at 90MHz.



Software

- Development:
  - Early: Xilinx FPGA tools (EDK, ISE)
  - Final: RDL (RAMP Description Language)
    - Allows parameterization
    - First step in making RAMP Blue into a emulator
- System:
  - Each node boots its own copy of uClinux
  - Each node mounts an NFS file system
  - Unified Parallel C (UPC)
    - Shared memory abstraction over messages framework
  - FPU code generated by custom GCC SoftFPU backend

Applications

- Application:
  - Runs UPC (Unified Parallel C) version of a subset of NAS (NASA Advanced Scientific) Parallel Benchmarks (all class S, to date)

CG	Conjugate Gradient, IS Integer Sort	512 cores
EP	Embarassingly Parallel, MG Multi-Grid	512, 1008 cores
FT	FFT	<64 cores

RDL & Emulation

- The “RAMP Description Language” (RDL)
  - Hierarchical structural netlisting langauge
  - Describes message passing distributed event simulations
  - System level: contains no behavioral spec.
- Tradeoffs
  - Costs
    - Use of the RAMP target model
    - Area, time and power to implement this model
  - Benefits
    - Abstraction of locality & timing of communications
    - System debugging & power tools
    - Determinism, sharing and research
  - Goal: trade costs for benefits as needed

Implementation Issues

- Large Hardware/Software System with many bugs:
  - Reliable low-level physical SDRAM controller has been a major challenge
  - A few MicroBlaze bugs in both gateware and GCC tool-chain (race conditions, OS bugs, GCC backend bugs)
  - FPU: Compilation Problems & Bad Results
  - RAMP Blue pushed the use of BEE2 modules to new levels - previously most aggressive users were for Radio Astronomy
    - memory errors exposed memory controller calibration loop errors (tracked down to PAR problems)
    - DIMM socket mechanical integrity problems
- Long “recompile” times hindered debugging
  - FPGA place and route takes 3-30 hours

Future Work / Opportunities

- Processor/network interface currently very inefficient
  - DMA support should replace programmed I/O approach
- Many of the features for a ***RAMP*** *(emulator)* currently missing
  - Time dilation
    - Ex: change relative speed of network, processor, memory)
  - Extensive HW supported monitoring
  - Virtual memory, other CPU/ISA models
  - Other network topologies
  - RDL implementation is a start
- Collaboration
  - Good starting point for processor+HW-accelerator architectures
  - At least one other group at Berkeley is already using it
  - Released version available soon in our design repository at:
    http://repository.eecs.berkeley.edu

GSRC RAMP

Conclusions

- A First Step Towards
  - Developing a robust RAMP infrastructure for more complicated parallel systems
    - Required debugging/insight capabilities
    - Driver & source for general RAMP infrastructure
  - Reuseable Gateware
    - Much of the RAMP Blue gateware is directly applicable to future systems
  - Fixing bugs and reliability issues
    - Exposed & corrected bugs in BEE2 platform and gateware
    - Help in design of future RAMP hardware and gateware
- RAMP Blue represents the largest soft-core, FPGA based computing system ever built!