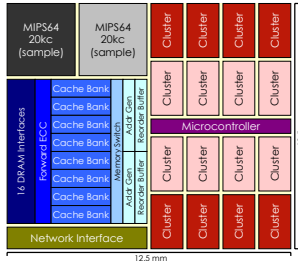
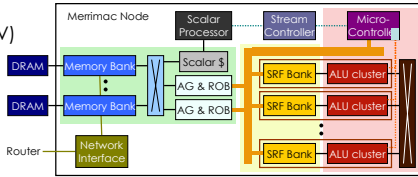


William J. Dally, Mattan Erez, Jung Ho Ahn, Nuwan Jayasena, Abhishek Das, Francois Labonte, Timothy Knight and Binu Mathew

Merrimac Node

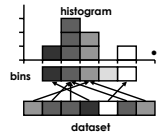
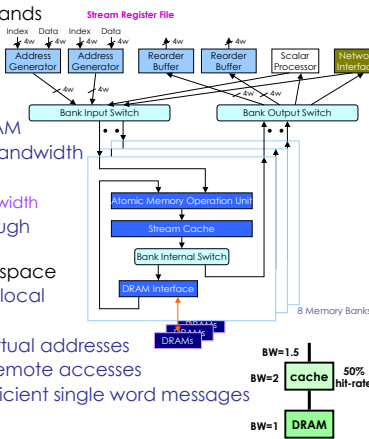
- 90nm CMOS process (1 V)
- ASIC technology
- 1 GHz (37 FO4)
- 128 GFLOPs



- Inter-cluster switch between clusters
- 156.25 mm² (small ~12.5 x 12.5)
 - Stanford Imagine is 16mm x 16mm
 - MIT Raw is 18mm x 18mm
- 25 Watts per processor (P4 = 75 W)
- 41 Watts total per node (with DRAM)

Memory System

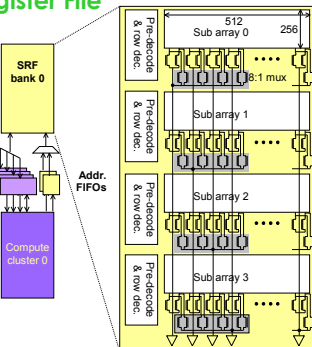
- Single instruction accesses thousands of multi-word records
 - Fill a very deep and wide memory pipeline
- High per-node bandwidth
 - 16 banks of 1 Gbit DDR2-SDRAM for 25.6 GB/s peak memory bandwidth
 - Memory access scheduling
 - Improves average DRAM bandwidth
 - Bandwidth amplification through stream cache
- High bandwidth global memory space
 - Flat address space to access local memory at any node
 - Segment registers translate virtual addresses
 - Network controller performs remote accesses
 - High radix routers allow for efficient single word messages



- Remote stream operations (scatter-add)
 - Increase data-parallel performance on "difficult" algorithms

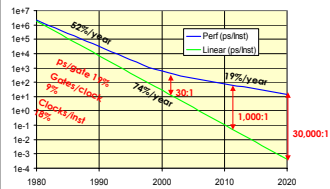
Stream Register File

- Single ported memory
 - Efficient wide access of 4 contiguous words
- Implemented using sub arrays
 - Reduced access time
 - Reduced power
- Stream-buffers match bandwidth to compute needs
 - Time multiplex the SRF port
- Indexed SRF at low extra cost
 - 8:1 MUX in sub-arrays
 - Row decoder per sub-array

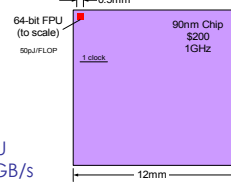


The Merrimac streaming supercomputer project aims to develop a scientific computer that offers an order of magnitude or more improvement in performance per unit cost compared to cluster-based scientific computers built from the same underlying semiconductor and packaging technology. We expect this efficiency to arise from two innovations: stream architecture and advanced interconnection networks. Organizing the computation into streams and exploiting the resulting locality using a register hierarchy enables a stream architecture to reduce the memory bandwidth required by representative computations by an order of magnitude or more.

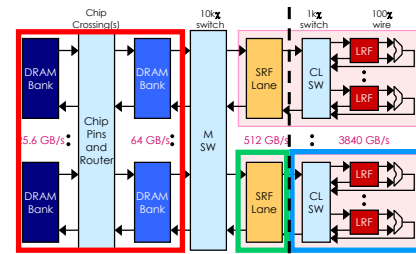
Requirements for Achieving High Performance on Modern Semiconductor Processes



- Parallelism
 - 100s FPU's per chip (millions per system)
- Latency Tolerance
 - 500 cycle remote memory access
- Locality
 - To match 20Tb/s ALU bandwidth to ~100GB/s chip bandwidth



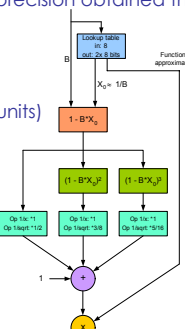
Bandwidth/Register Hierarchy



- Exploits locality
 - Producer-consumer within kernel in the LRF
 - Producer-consumer across kernels in the SRF
- Reduces the distance data travels
- Support large number of ALUs

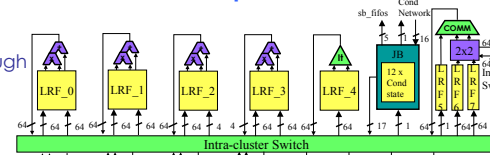
Iterative Unit

- Speeds up inverse and inverse sqrt
 - Generates 27 bits of precision
 - 54 bits of precision obtained through single Newton-Raphson Iteration (on MADD units)



- Fully pipelined
 - Each cluster can sustain 1 inverse or 1 inverse square root every cycle
 - Latency of 3 Ops (15 cycles) vs 12 Ops previously
- Reduces kernel run-time by 50% for kernels with lots of divide (StreamFLO)

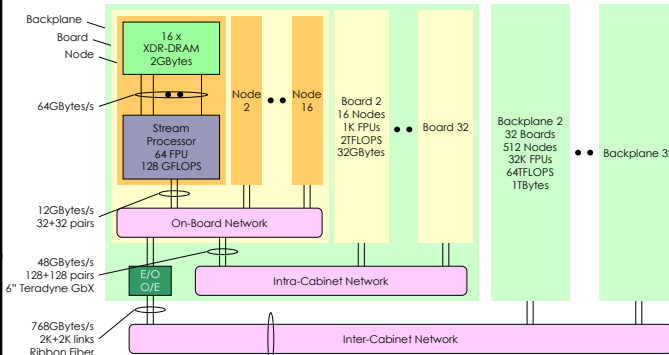
Compute Cluster



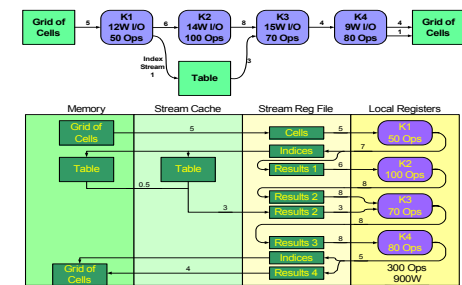
- 4 64-bit MADD units
 - Fully pipelined with 5 cycle latency
- Iterative operation support
- Communication unit
 - Inter-cluster communication
- Distributed local register files
 - Low area and power
 - Adding switches on read and write ports provides flexible connectivity
 - better scheduling and efficient use of available registers
 - saves area and power

Register File Organization	# Regs / MADD	Area (mm ²)
3 single-ported RF (64 entries / RF)	192	0.1008
1 5-ported RF (64 entries)	64	0.07995
3 single-ported RF (32 entries / RF) + switches	96	0.0567 + switch area

Merrimac System



Stream Applications



Application	Sustained GFLOPs	FP Ops / Mem Ref	SRF Refs	Mem Refs	
StreamFEM3D (Euler, quadratic)	31.6	17.1	153.0M (95.0%)	6.3M (3.9%)	1.8M (1.1%)
StreamFEM3D (MHD, constant)	39.2	13.8	186.5M (99.4%)	7.7M (0.4%)	2.8M (0.2%)
StreamMD (grid algorithm)	14.2	12.1	90.2M (97.5%)	1.6M (1.7%)	0.7M (0.8%)
GROMACS	38.8	9.7	108M (95.0%)	4.2M (2.9%)	1.5M (1.3%)
StreamFLO	12.9	7.4	234.3M (95.7%)	7.2M (2.9%)	3.4M (1.4%)

Simulated without iterative operation acceleration unit
* The low numbers are a result of many divide and square-root operations

Merrimac Implementation Plan

- Completed Merrimac processor architecture specification document
 - Architecture definition, Instruction set architecture, Exceptions
- Responded to NASA H&RT BAA with detailed prototype plan
- Chip design steps (in collaboration with LLNL):
 - Contact several chip design firms for accurate estimates and design proposals
 - Chip design with most competitive firm
 - Fabrication, testing and packaging
- System design steps (in collaboration with LLNL):
 - Merrimac processors designed to integrate with Cray YARC interconnection network
 - Merrimac boards designed to plug into Cray Rainier system
- Software system steps:
 - Continued compiler development collaboration with Reservoir Labs
 - Collaborate with LLNL on run-time system design
 - Collaborate with LBNL on UPC integration and multi-node