MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY
Concurrent VLSI Architecture Group

THE M-MACHINE MULTICOMPUTER

Marco Fillo, Stephen W. Keckler, William J. Dally,
Nicholas P. Carter, Andrew Chang, Yevgeny Gurevich, Whay S. Lee

November 30, 1995

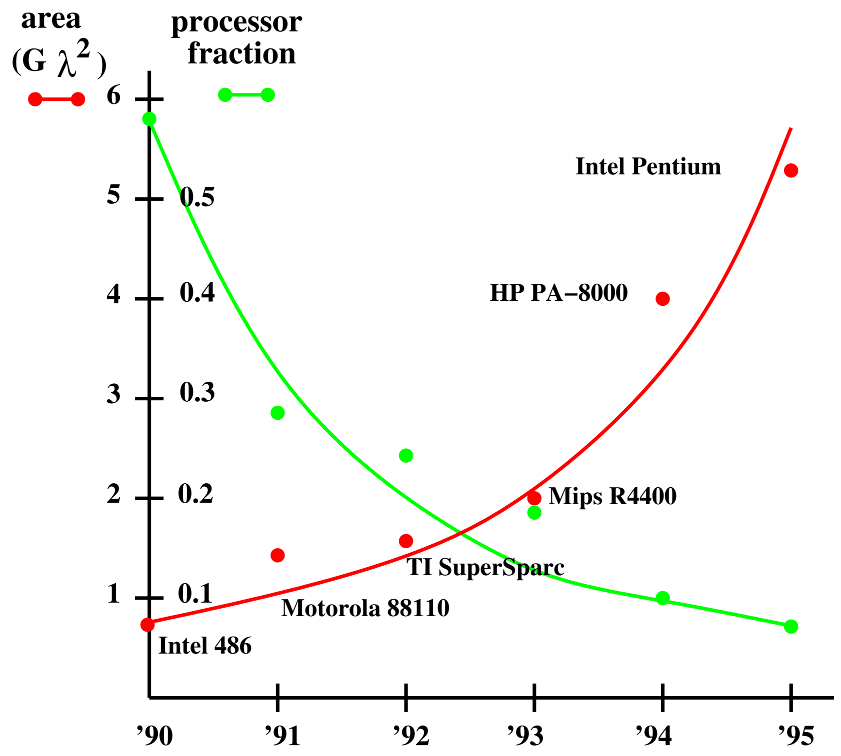
Outline

1. Motivation
2. M-Machine Mechanisms
   - Instruction Level Parallelism
   - Multithreading
   - Communication
3. Software
4. Summary

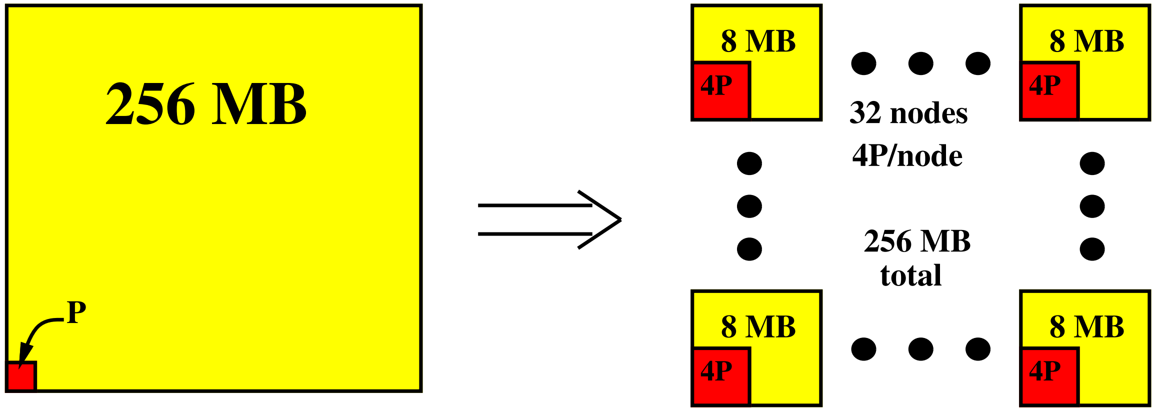
The Vanishing Processor

64 bit processor w/ pipelined FPU (R4600) = $400M\lambda^2$

- λ = 1/2 feature size, process independent
- 8% of today's die
- 0.13% of today's system (256MB)



What to Do?



- Increase processing per unit memory
  - 11% of 256MB system
  - 128 times peak performance
  - 1.5 times cost

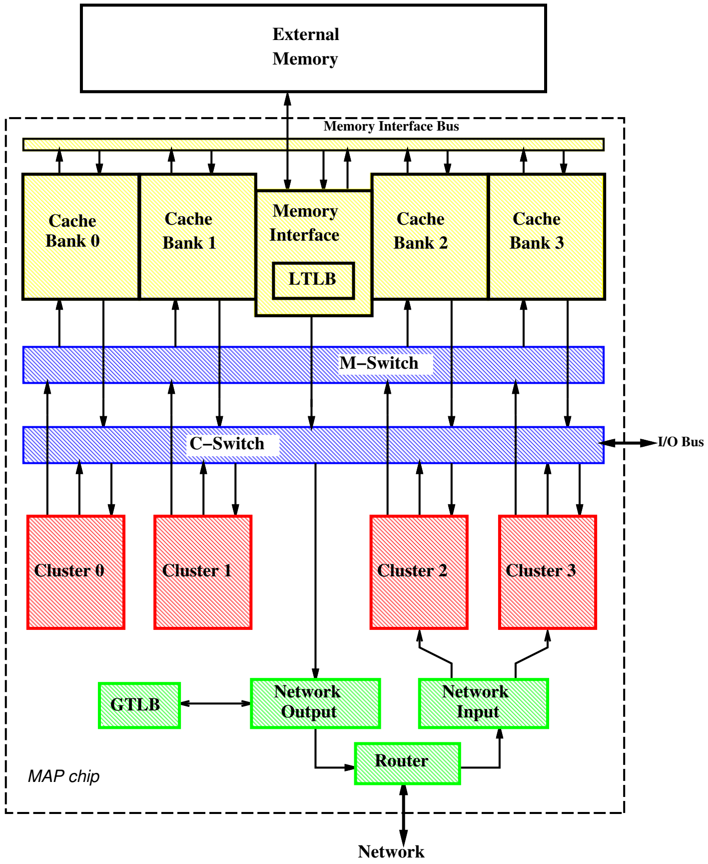
Requirements for System

- Exploit concurrency
  - Instruction level parallelism
  - Multiple processors
- Fast access to remote memory
  - Detect when remote access is needed
  - Translation
  - Communication

The Multi-ALU Node

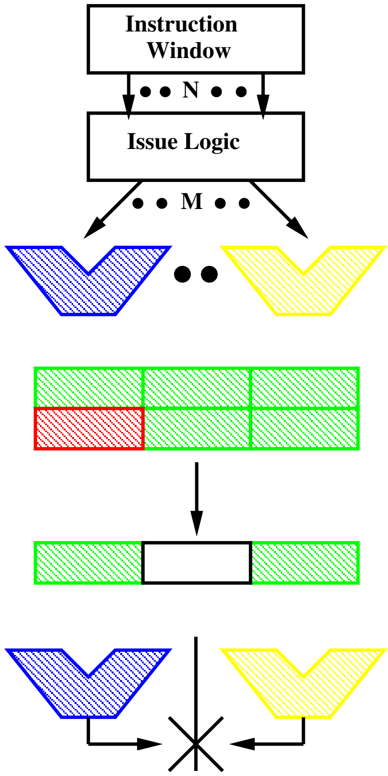
Highly integrated node (6 chips) containing:

- 8 MBytes of external memory
- A multi-ALU processor (MAP) I.C.
- 160KB cache (D+I)
- 8 register files
- 4 FPUs, 8 Integer units
- NI + Router



Multiple ALUs

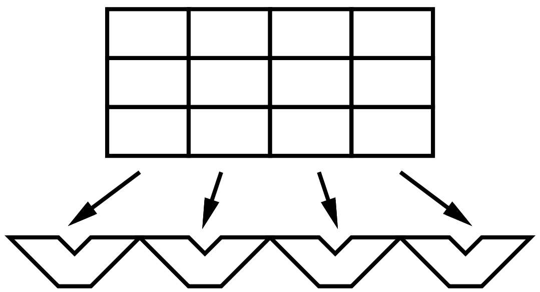
Problem: Current multi-ALU control is inadequate



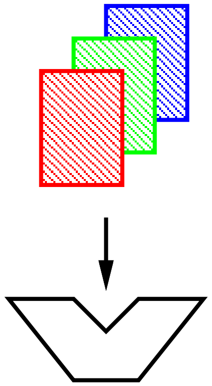
- Superscalar
  - Issue logic and RF at scaling limits
  - Does not exploit compiler
- VLIW
  - Variable latency
  - Empty issue slots
- Multiple independent processors
  - Long interaction latency

Processor Coupling

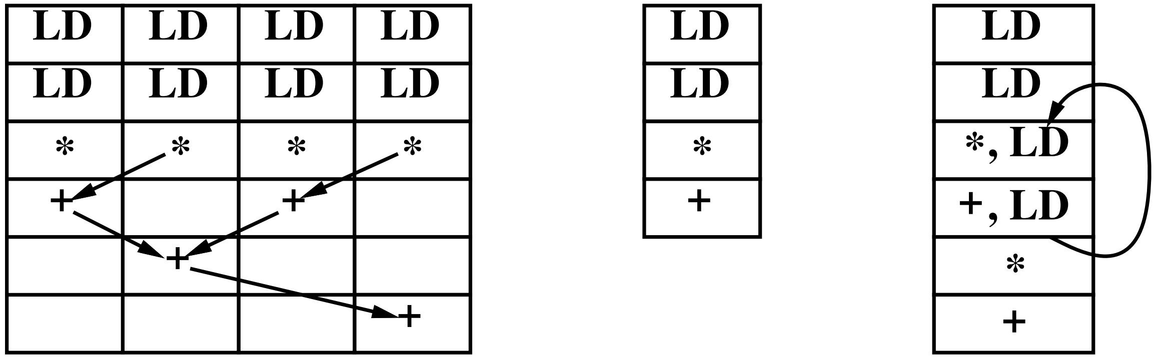
- Compile-time scheduling (H-Threads)
  - Instruction Level Parallelism
  - Independent cluster execution
  - Register based communication
  - Allows slip between different clusters



- Runtime multithreading (V-Threads)
  - Hides variable latencies
  - Exploits slip between different clusters



Dot Product on Multiple H-Threads



Schedule for 4 clusters

Compressed Schedule for one cluster

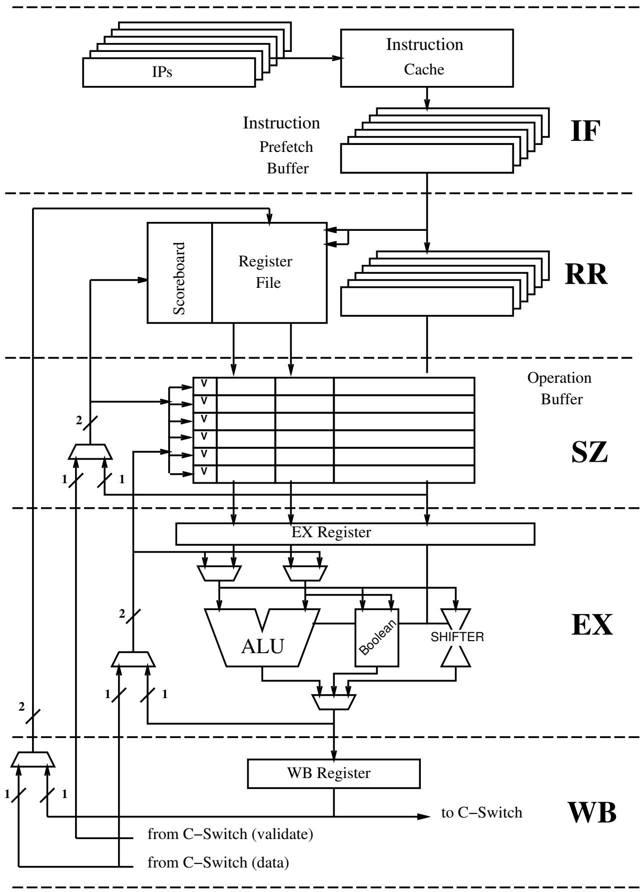
Pipelined Schedule for one cluster

- Local control only
- Register interlocking
- NO-OP compression

V-Thread Interleaving

V-Threads:

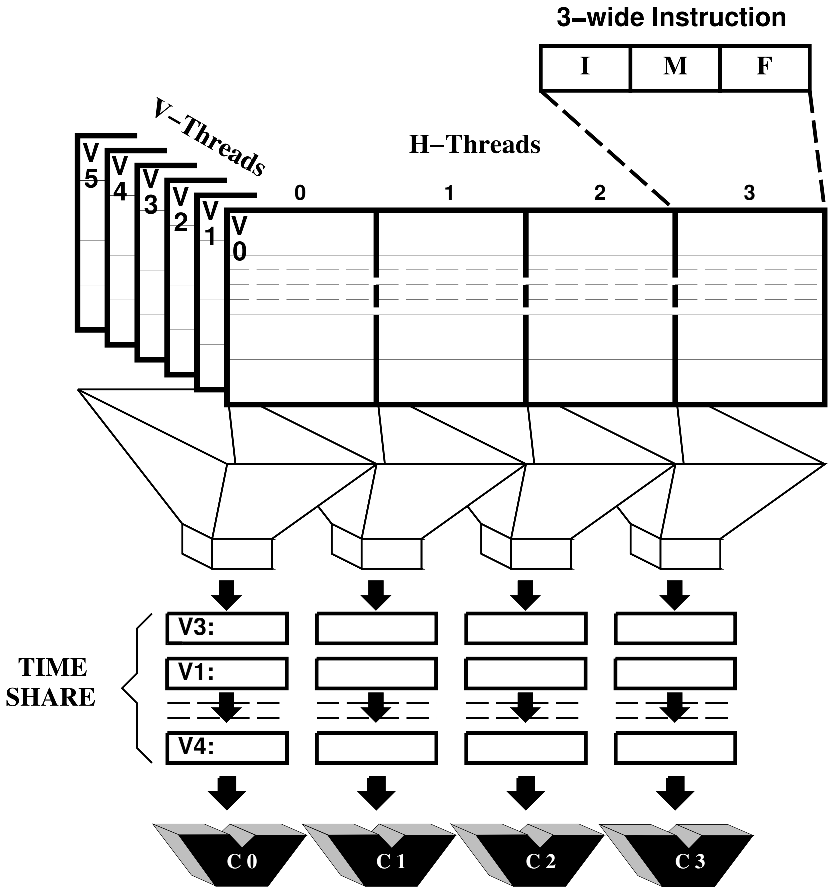
- Zero-overhead interleaving
- Independent register files
- Duplicate pipeline registers
- SZ pipeline stage
- Share memory system and ALU pipelines



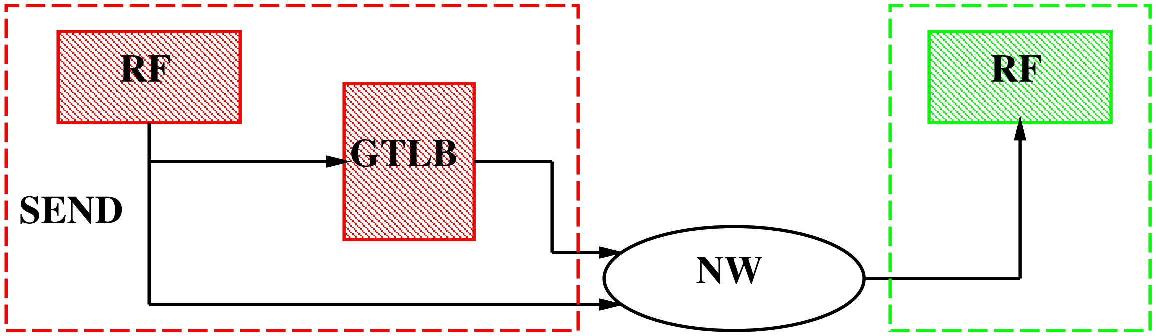
Dynamic Instruction Scheduling

V-Thread usage:

- V0 – V3: User
- V4: Event
- V5: Exception

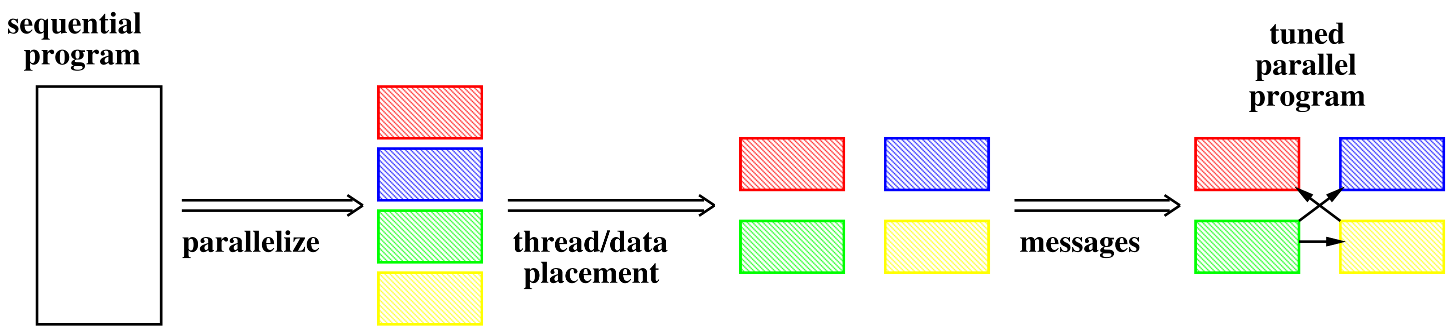


Communication Mechanisms



- Message Injection: SEND instruction
- Automatic Translation
  - Global TLB: VA $\rightarrow$ Node ID
- Fast message reception: dedicated H-Thread
- Automatic data retrieval
  - Coherence bits on cache line
  - Trap on load/store

Application Development



- Incremental approach
- Old code compiles and runs immediately
- Only need 1.5X speedup for cost/performance win

Status

- 13 million transistors, 100MHz target
- 70% of schematics, 30% of layout complete
- Tapeout scheduled 11/96
- Software
  - Multiflow Compiler port: Joint with Scalable Concurrent Programming Laboratory (Caltech)
  - Runtime System

Summary

- Balance silicon area costs
- Microarchitectural mechanisms
  - Processor Coupling
    - Register based synchronization and communication
    - Multithreaded pipeline
    - On-chip switches
  - Fast access to remote memory
    - Send instruction
    - Fast trap handling (dedicated thread)
    - Automatic node-ID translation
    - Coherence bits on cache lines