

**JPL**



*Presentation to the*

*High Performance Embedded Computing Conference 2002:*

---

## **From PIM to Petaflops Computing**

# **MIND: Scalable Embedded Computing through Advanced Processor in Memory**

Thomas Sterling

California Institute of Technology

and

NASA Jet Propulsion Laboratory

September 24, 2002

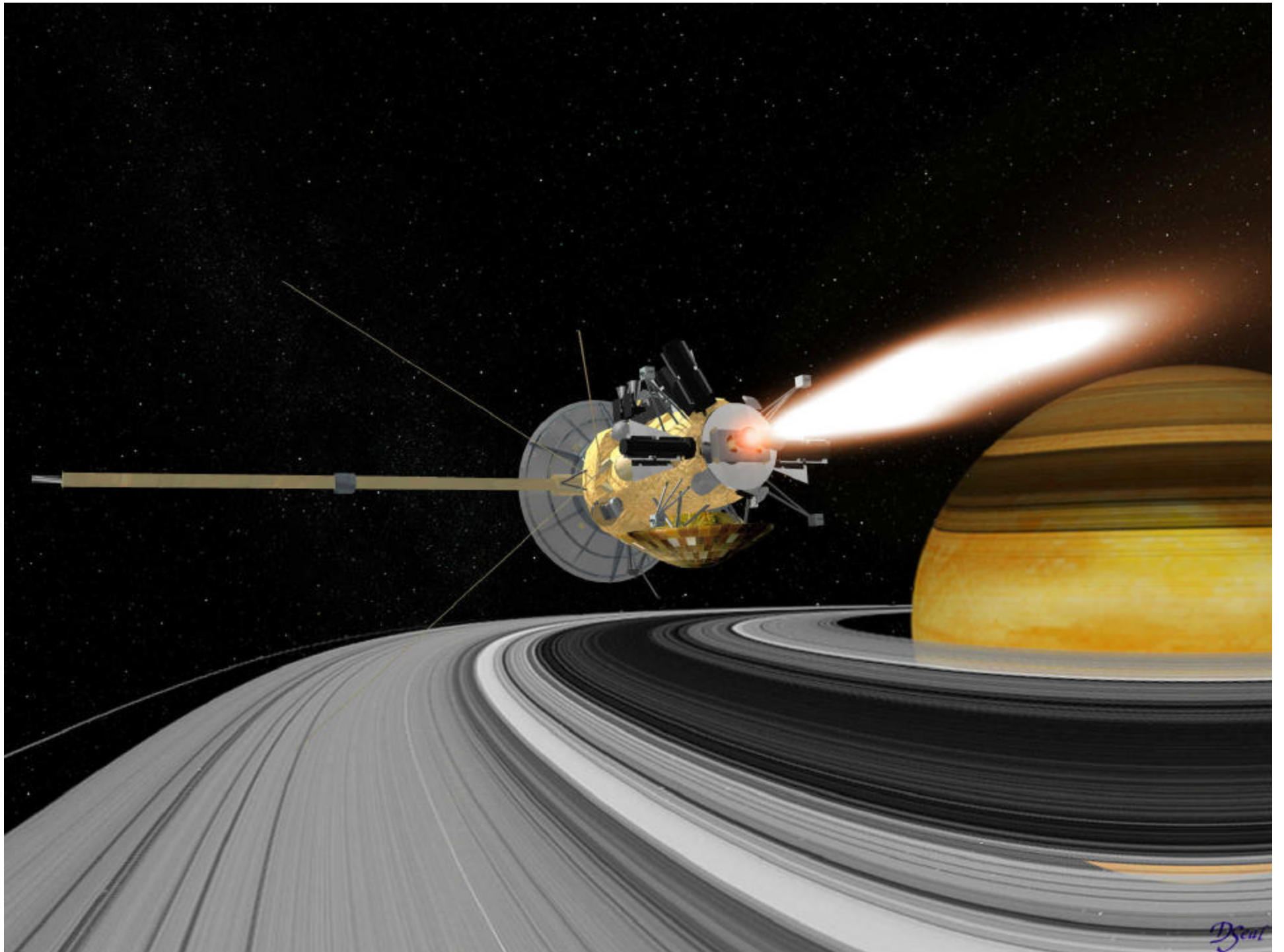
---

# Report Documentation Page

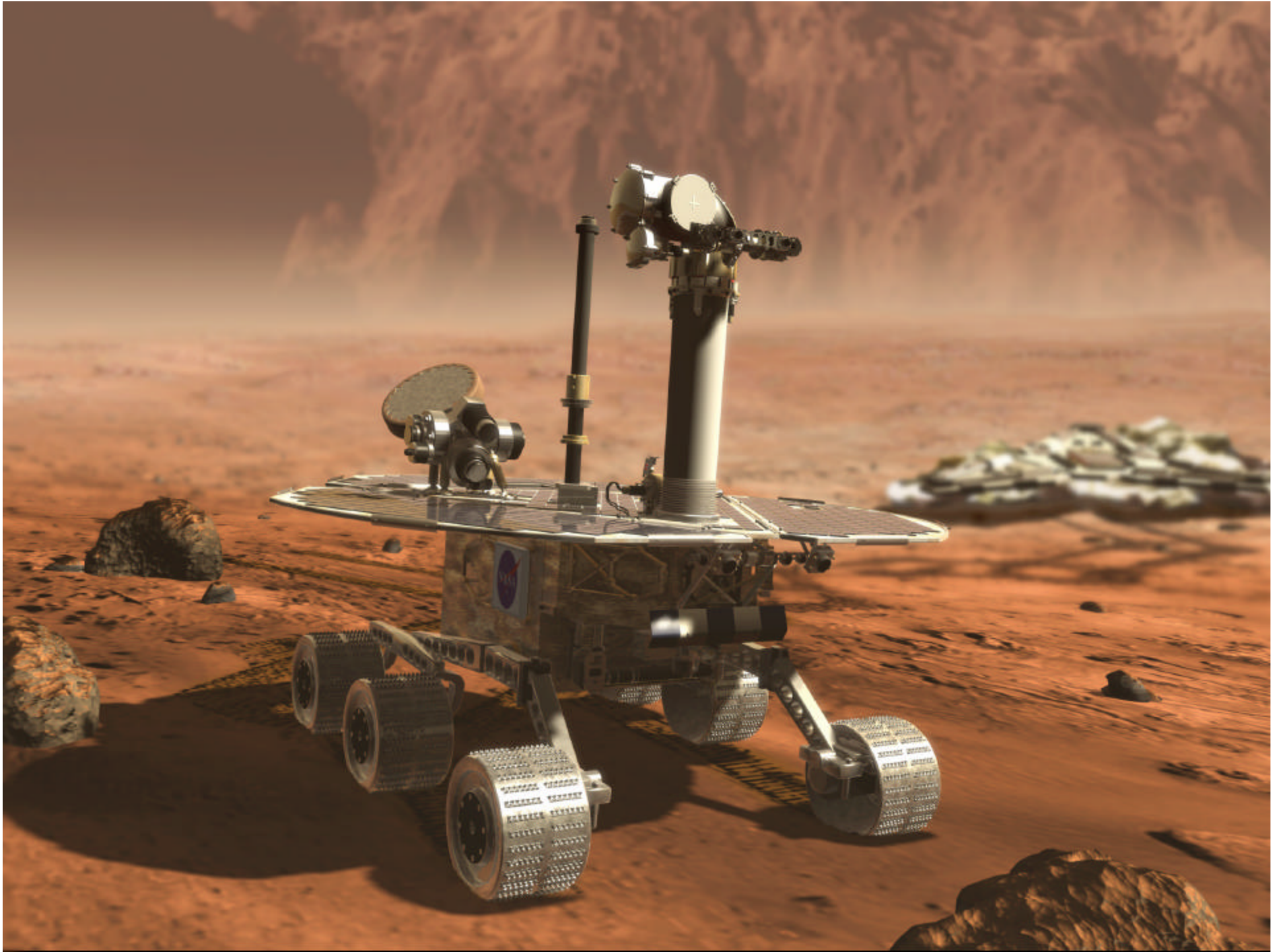
Form Approved  
OMB No. 0704-0188

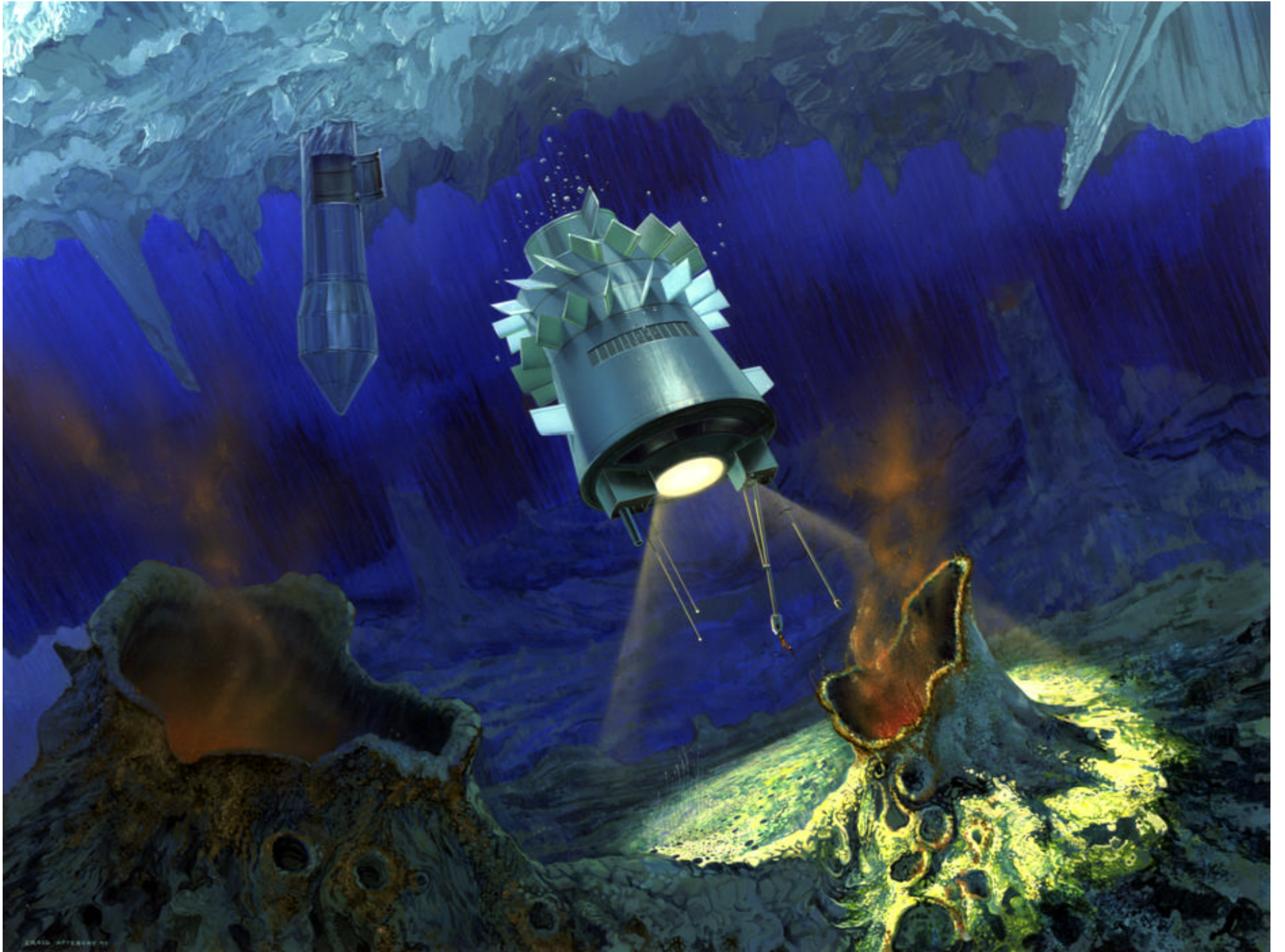
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE <b>21 MAY 2003</b>		2. REPORT TYPE <b>N/A</b>		3. DATES COVERED <b>-</b>	
4. TITLE AND SUBTITLE <b>From PM to Petaflops Computing, MIND: Scalable Embedded Computing through Advanced Processor in Memory</b>				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) <b>California Institute of Technology and NASA Jet Propulsion Laboratory</b>				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT <b>Approved for public release, distribution unlimited</b>					
13. SUPPLEMENTARY NOTES <b>The original document contains color images.</b>					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT <b>UU</b>	18. NUMBER OF PAGES <b>36</b>	19a. NAME OF RESPONSIBLE PERSON
a. REPORT <b>unclassified</b>	b. ABSTRACT <b>unclassified</b>	c. THIS PAGE <b>unclassified</b>			



*Deer*







# Summary of Mission Driver Factors

---

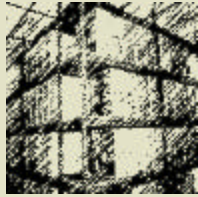
- ◆ Speed of light preclude real time manual control
- ◆ Mission duration and spacecraft lifetime up to 100 years
- ◆ Adaptivity to system and environmental uncertainty through reasoning
- ◆ Cost of ground based deep space tracking and high bandwidth downlink
- ◆ Weight and cost of space craft high bandwidth downlink
  - Antennas, Transmitter, Power supply
  - Raw power source
  - Maneuver rockets and/or inertial storage, Mid course main engine thrusters
  - Launch vehicle fuel and type
- ◆ On-board science computation
- ◆ On-board mission planning (long term and real time)
- ◆ On-board mission fault detection, diagnostic, and reconfiguration
- ◆ Obstructed mission profiles



# Goals for a New Generation of Spaceborne Supercomputer

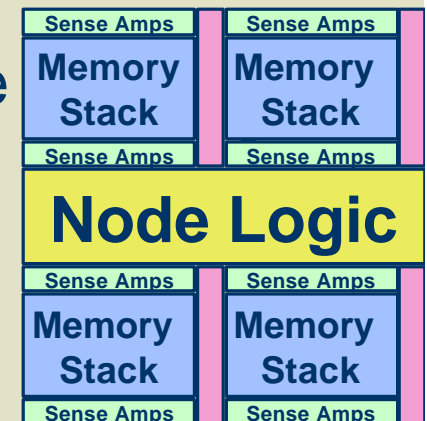
---

- ◆ Performance gain of 100 to 10,000
- ◆ Low power, high power efficiency.
- ◆ Wide range for active power management.
- ◆ Fault tolerance and graceful degradation.
- ◆ High scalability to meet widely varying mission profiles.
- ◆ Common ISA for software reuse and technology migration.
- ◆ Multitasking, real time response.
- ◆ Numeric, data oriented, and symbolic computation.



# Processor in Memory (PIM)

- ◆ PIM merges logic with memory
  - Wide ALUs next to the row buffer
  - Optimized for memory throughput, not ALU utilization
- ◆ PIM has the potential of riding Moore's law while
  - greatly increasing effective memory bandwidth,
  - providing many more concurrent execution threads,
  - reducing latency,
  - reducing power, and
  - increasing overall system efficiency
- ◆ It may also simplify programming and system design

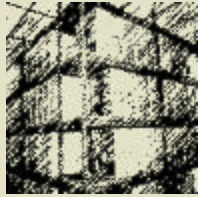




# Why is PIM Inevitable?

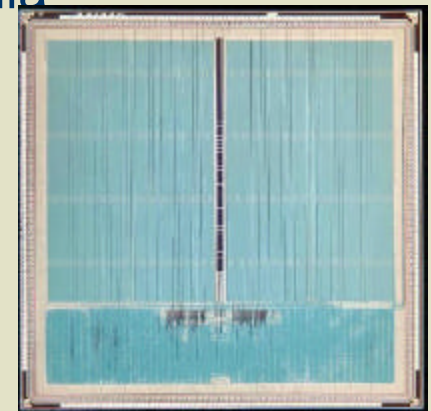
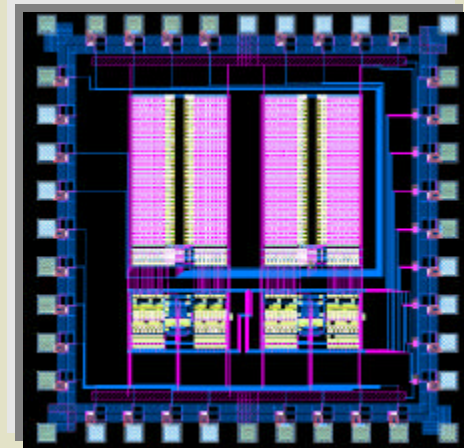
---

- ◆ Separation between memory and logic artificial
  - von Neumann bottleneck
  - Imposed by technology limitations
  - Not a desirable property of computer architecture
- ◆ Technology now brings down barrier
  - We didn't do it because we couldn't do it
  - We can do it so we will do it
- ◆ What to do with a billion transistors
  - Complexity can not be extended indefinitely
  - Synthesis of simple elements through replication
  - Means to fault tolerance, lower power
- ◆ Normalize memory touch time through scaled bandwidth with capacity
  - Without it, takes ever longer to look at each memory block
- ◆ Will be mass market commodity commercial market
  - Drivers outside of HPC thrust
  - Cousin to embedded computing



# Current PIM Projects

- ◆ IBM Blue Gene
  - Pflops computer for protein folding
- ◆ UC Berkeley IRAM
  - Attached to conventional servers for multi-media
- ◆ USC ISI DIVA
  - Irregular data structure manipulation
- ◆ U of Notre Dame PIM-lite
  - Multithreaded
- ◆ Caltech MIND
  - Virtual everything for scalable fault tolerant general purpose

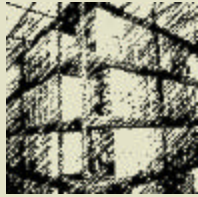




# Limitations of Current PIM Architectures

---

- ◆ No global address space
- ◆ No virtual to physical address translation
  - DIVA recognizes pointers for irregular data handling
- ◆ Do not exploit full potential memory bandwidth
  - Most use full row buffer
  - Blue Gene/Cyclops has 32 nodes
- ◆ No memory to memory process invocation
  - PIM-lite & DIVA use *parcels* for method driven computation
- ◆ No low overhead context switching
  - BG/C and PIM-lite have some support for multithreading



# MIND Architecture

---

- ◆ *Memory-Intelligence-and-Networking Devices*
- ◆ Target systems
  - Homogenous MIND arrays
  - Heterogeneous MIND layer with external high-speed processors
  - Scalable embedded
- ◆ Addresses challenges of:
  - global shared memory and virtual paged management
  - irregular data structure handling
  - dynamic adaptive on-chip resource management
  - inter-chip transactions
  - global system locality and latency management
  - power management and system configurability
  - fault tolerance



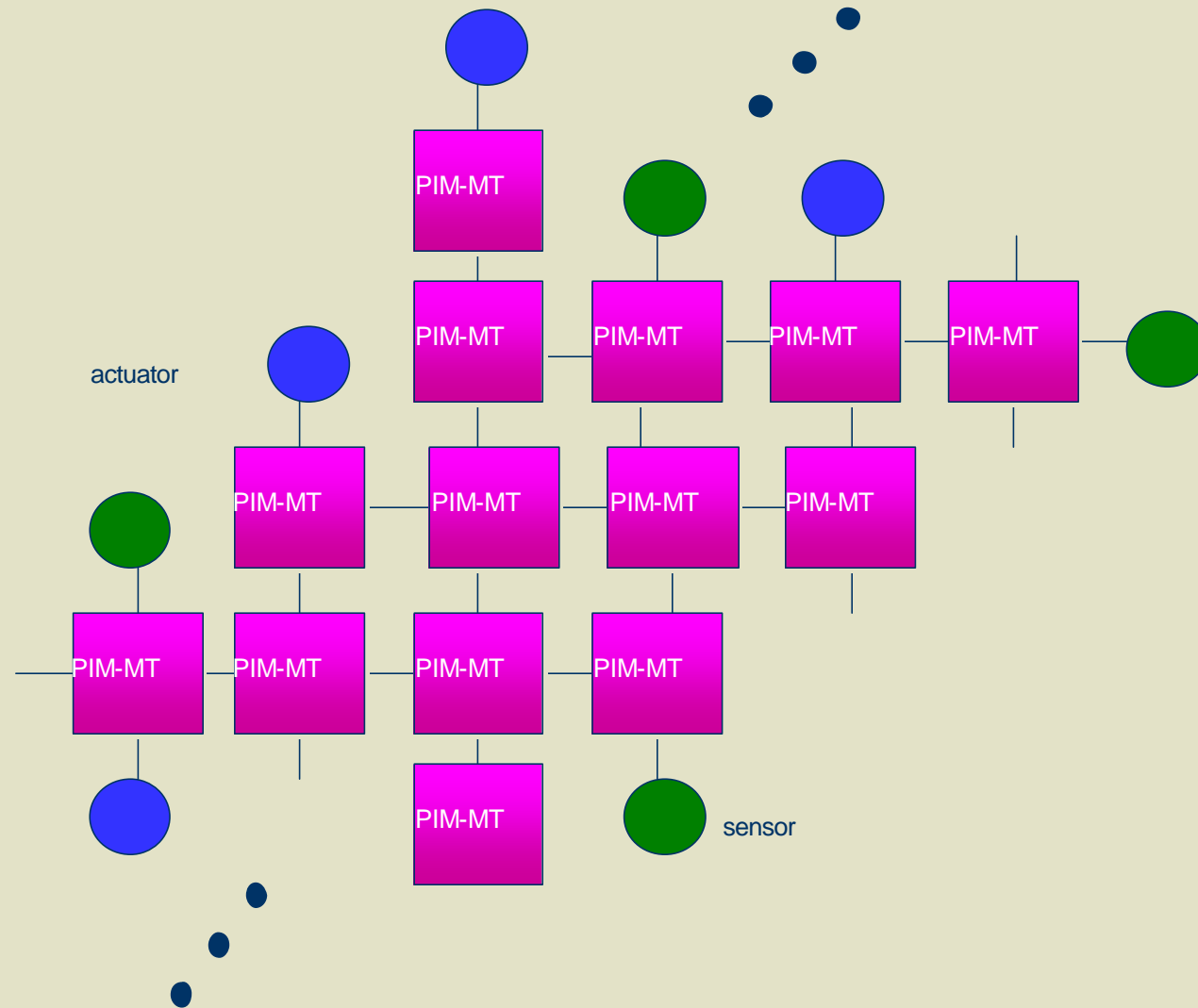
# Attributes of MIND Architecture

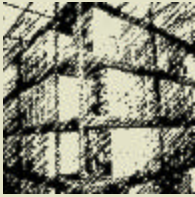
---

- ◆ Parcel active message driven computing
  - Decoupled split-transaction execution
  - System wide latency hiding
  - Move work to data instead of data to work
- ◆ Multithreaded control
  - Unified dynamic mechanism for resource management
  - Latency hiding
  - Real time response
- ◆ Virtual to physical address translation in memory
  - Global distributed shared memory thru distributed directory table
  - Dynamic page migration
  - Wide registers serve as context sensitive TLB
- ◆ Graceful degradation for Fault tolerance

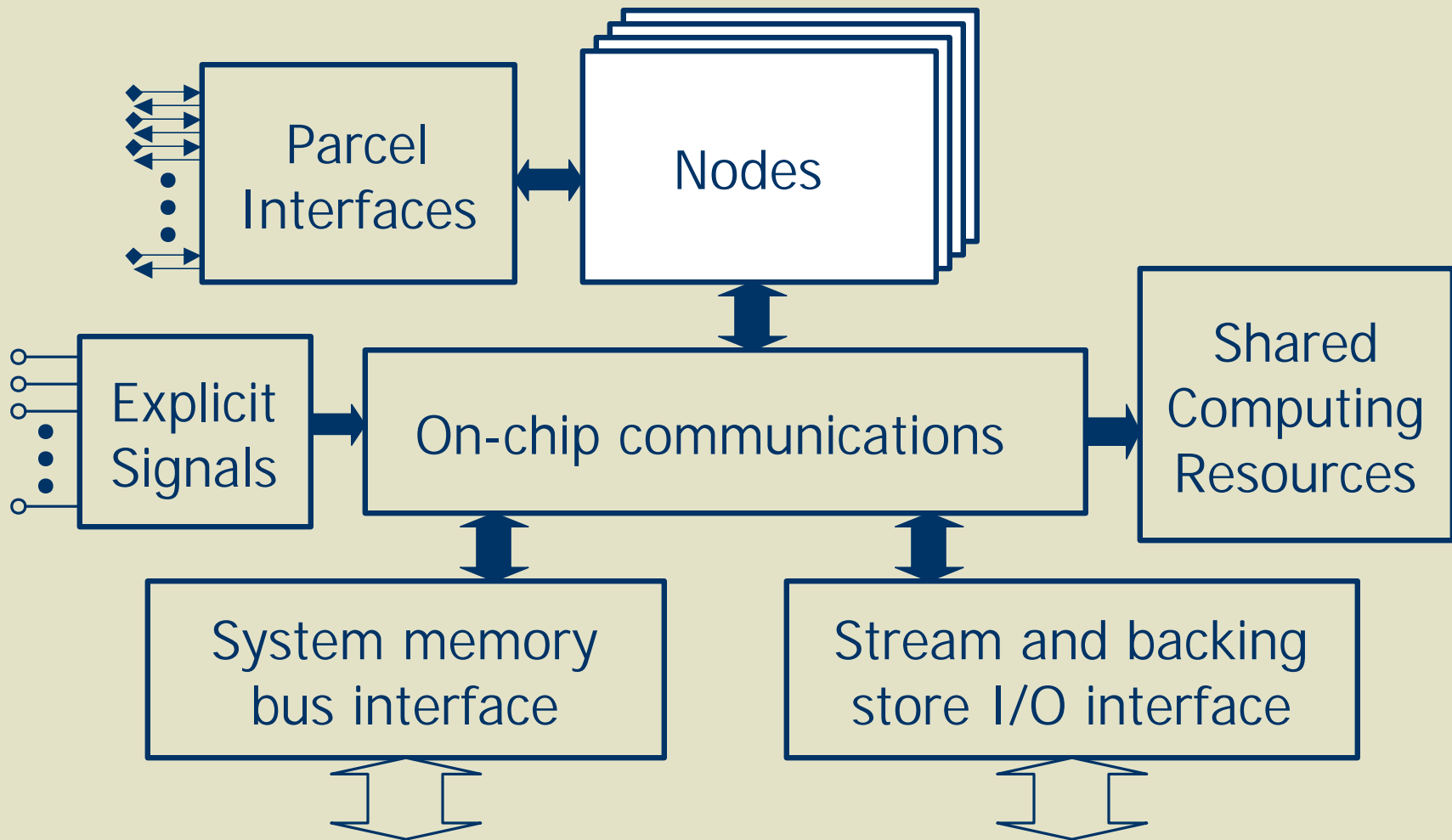


# MIND Mesh Array





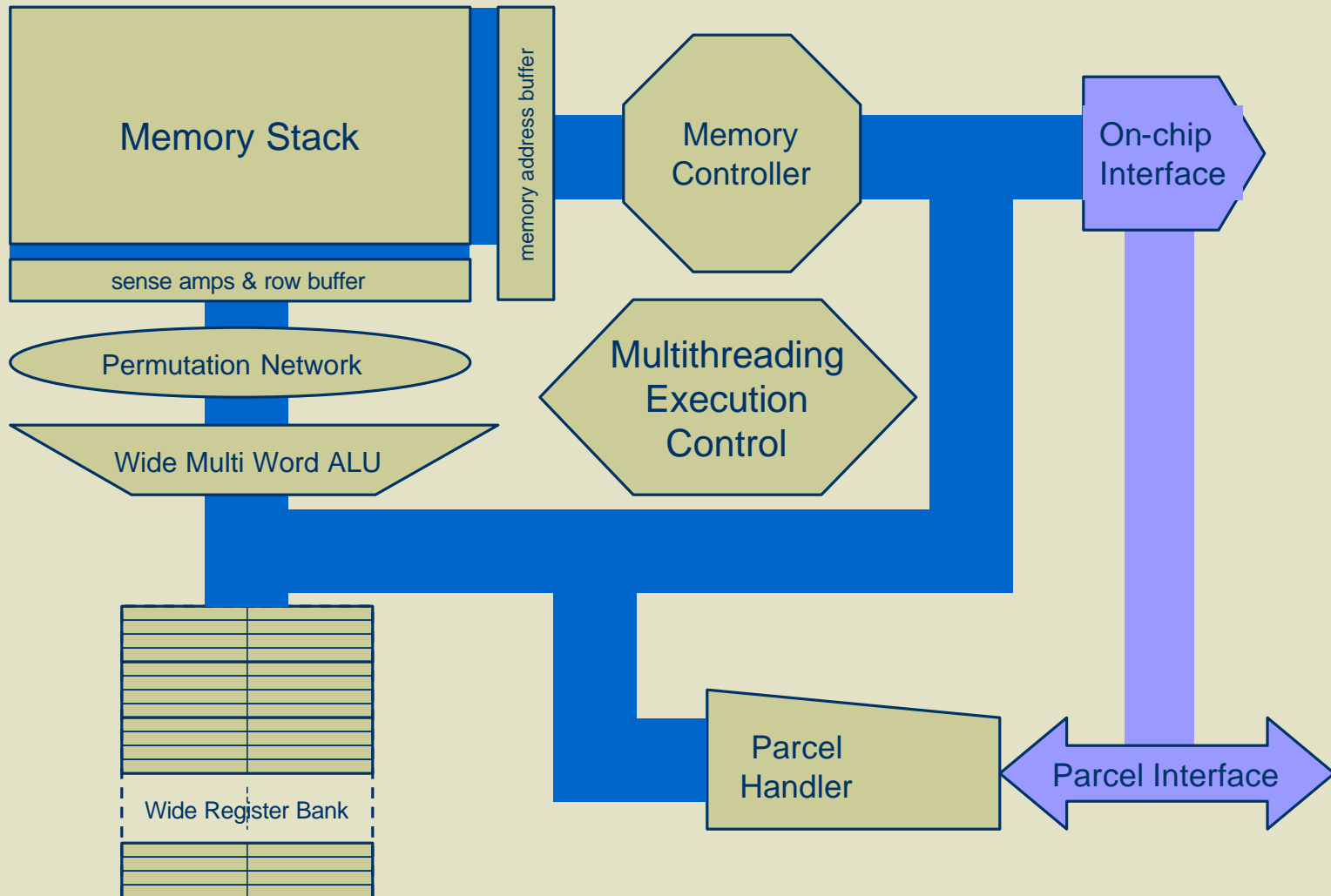
# Diagram - MIND Chip Architecture







# MIND Node





# Unified Register Set Supports a Diversity of Runtime Mechanisms

---

- ◆ Node status word
- ◆ Thread state
- ◆ Parcel decoding
- ◆ Parcel construction
- ◆ Vector register
- ◆ Translation Lookaside Buffer
- ◆ Instruction cache
- ◆ Data cache
- ◆ Irregular Data Structure Node (data, pointers, usw.)



# MIND Node Instruction Set

---

- ◆ Basic set of word operations
- ◆ Row wide field permutations for reordering and alignment
- ◆ Data parallel ops across row-wide register and delimited subfields
- ◆ Parallel dual ops with key field and data field for rapid associative searches
- ◆ Thread management and control
- ◆ Parcel explicit create, send, receive
- ◆ Virtual and physical word access; local, on-chip, remote
- ◆ Floating point
- ◆ Reconfiguration
- ◆ Protected supervisor



# Multithreading in PIMS

---

- ◆ MIND must respond asynchronously to service requests from multiple sources
- ◆ Parcel-driven computing requires rapid response to incident packets
- ◆ Hardware supports multitasking for multiple concurrent method instantiations
- ◆ High memory bandwidth utilization by overlapping computation with access ops
- ◆ Manages shared on-chip resources
- ◆ Provides fine-grain context switching
- ◆ Latency hiding

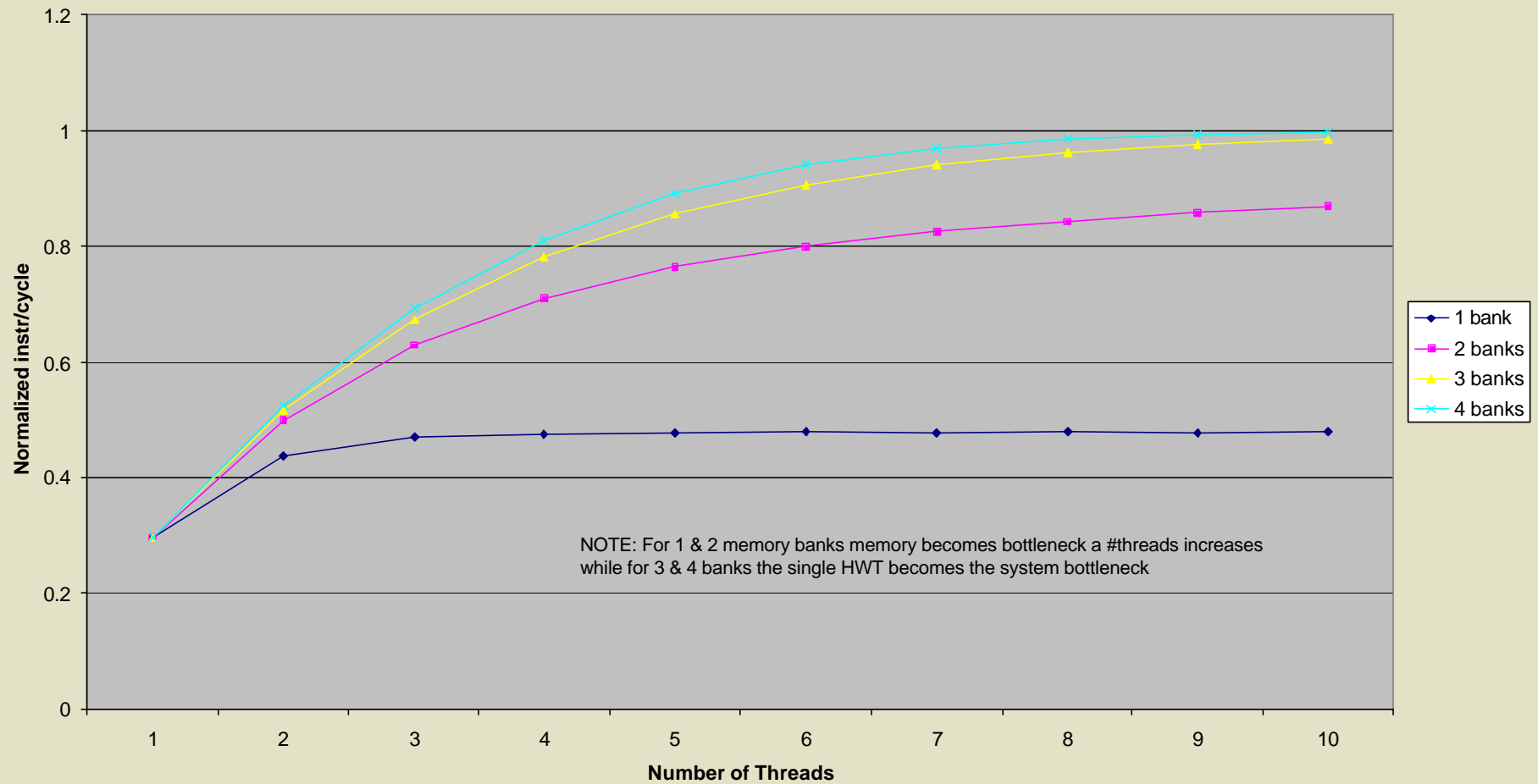


### Single HWT: Multiple Memory Banks: MultiThread

probability of reg-to-reg instr fixed at 0.7

probability of data cache hit fixed at 0.9

Memory access fixed at 70 cycles





# PIM Parcel Model

---

- ◆ **Parcel: logically complete grouping of info sent to a node on a PIM chip**
  - by SPELLs, other PIM nodes
- ◆ **At arrival, triggers local computation:**
  - Read from local memory
  - Perform some operation(s)
  - Write back locally (optional)
  - Return value to sender (optional)
  - Initiate additional parcel(s) (optional)





# Virtual Page Handling

---

- ◆ Pages preferentially distributed in local groups with associated page entry tables
- ◆ Directory table entries located by physical address
- ◆ Pages may be randomly distributed within MIND chip or group
- ◆ Pages may be randomly distributed requiring second hop from page table location
- ◆ Supervisor address space supports local node overhead and service tasks.
- ◆ Copying to physical pages, not to virtual
- ◆ Demand paging to/from backing store or other MIND chips
- ◆ Nodes directly address memory of others on same MIND chip



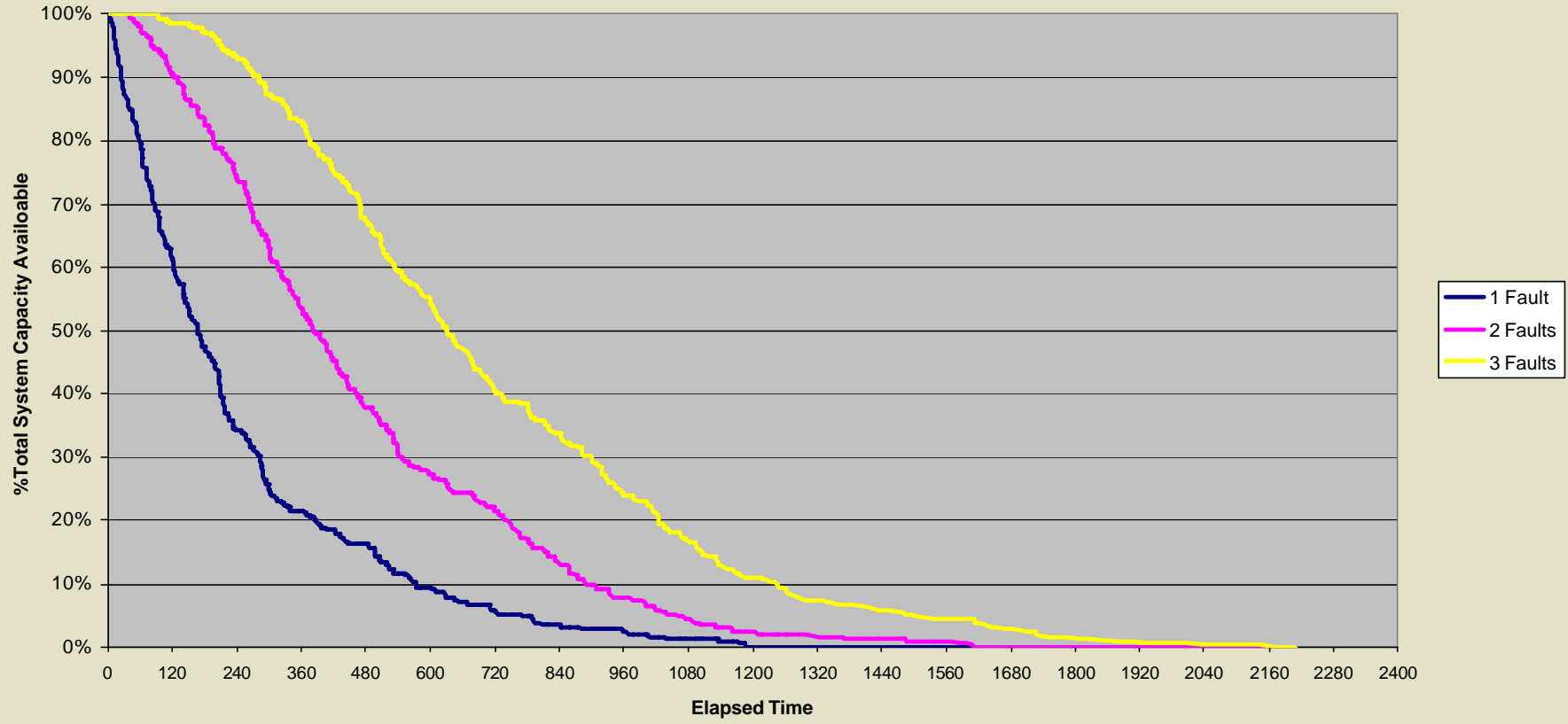
# Fault Tolerance

---

- ◆ Near fine-grain redundancy provides multiple alike resources to perform workload tasks.
- ◆ Even single-chip Gilgamesh (for rovers, sensor webs) will incorporate 4-way to 16-way redundancy and graceful degradation.
- ◆ Hardware architecture includes fault detection mechanisms.
- ◆ Software tags for bit-checking at hardware speeds; includes constant memory scrubbing.
- ◆ Monitor threads for background fault detection and diagnosis
- ◆ Virtual data and tasks permits rapid reconfiguration without software regeneration or explicit remapping.



**System Availability as Function Of Number of Faults Before Node Failure**  
**MTBF = 1 unit Exponential Arrival Rate of Faults**  
**64 Modules 4 Nodes/Module**

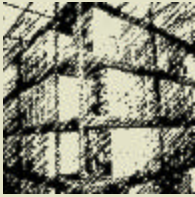




# Real Time Response

---

- ◆ Multiple nodes permits dedication of a single node to a single real time task
- ◆ Threads and pages can be *nailed down* for real time tasks
- ◆ Multithreading uses real time priority for guaranteed reaction time
- ◆ Preemptive memory access
- ◆ Virtual address translation can be buffered in registers as TLB
- ◆ Hardwired signal lines from sensors and to actuators



# Power Reduction Strategy

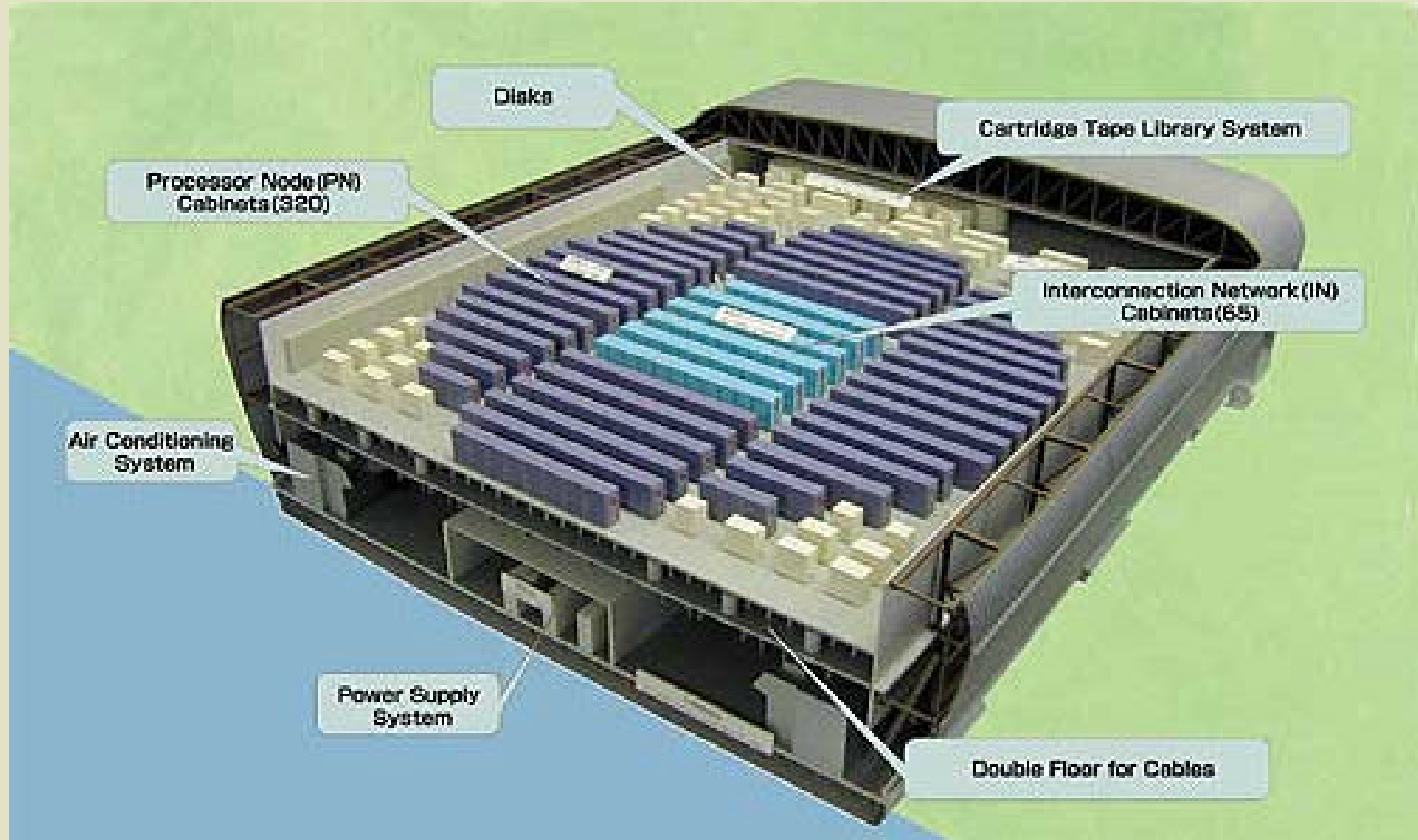
---

- ◆ **Objective**: achieve 10 to 100 reduction in power over conventional systems of comparable performance.
- ◆ On-chip data operations avoids external I/O drivers.
- ◆ Number of memory block row accesses reduced because all row bits available for processing.
- ◆ Simple processor with reduced logic. No branch prediction prediction, speculative execution, complex scoreboarding.
- ◆ No caches.
- ◆ Power management of separate processor/memory nodes.



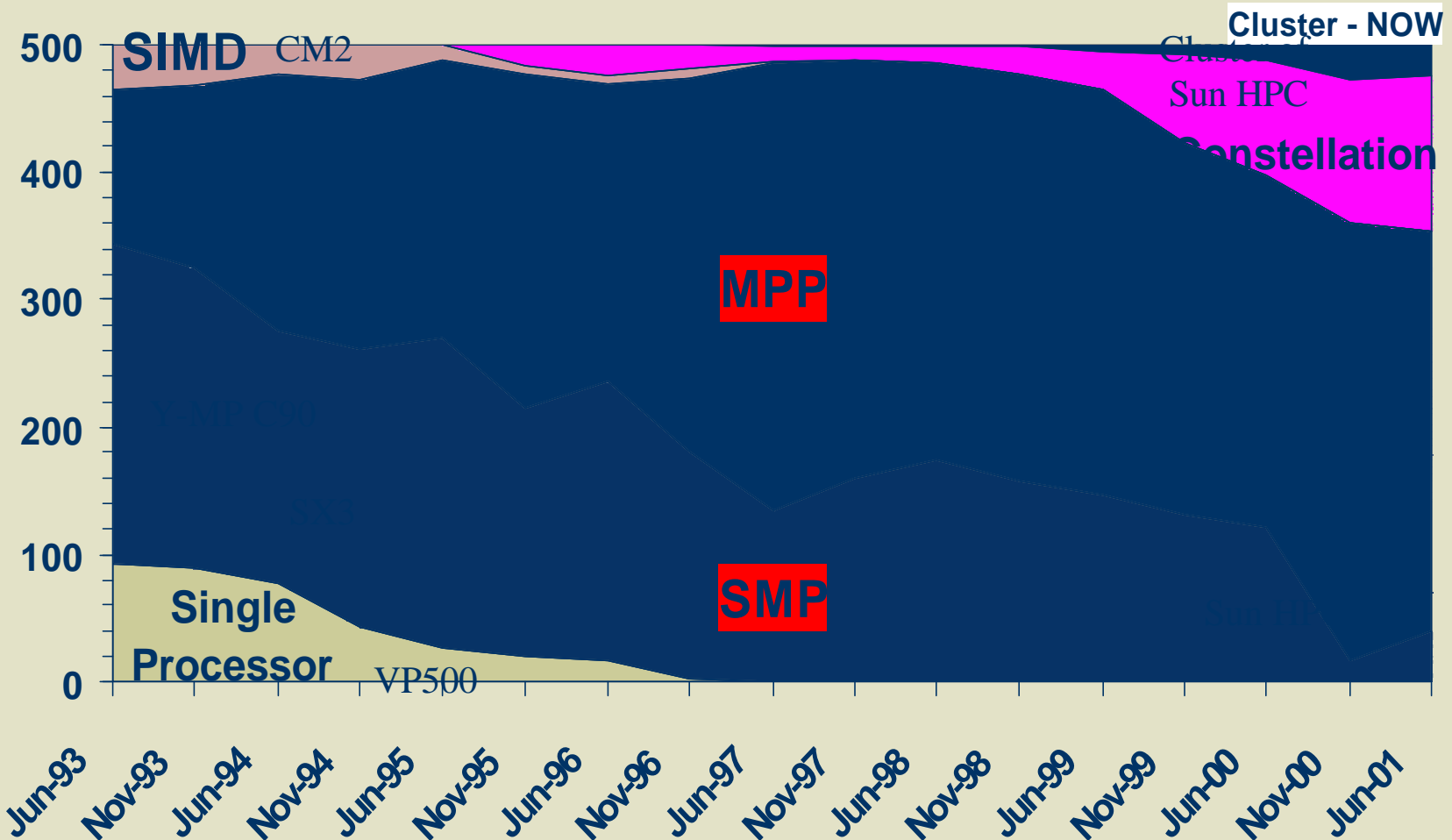


# Earth Simulator



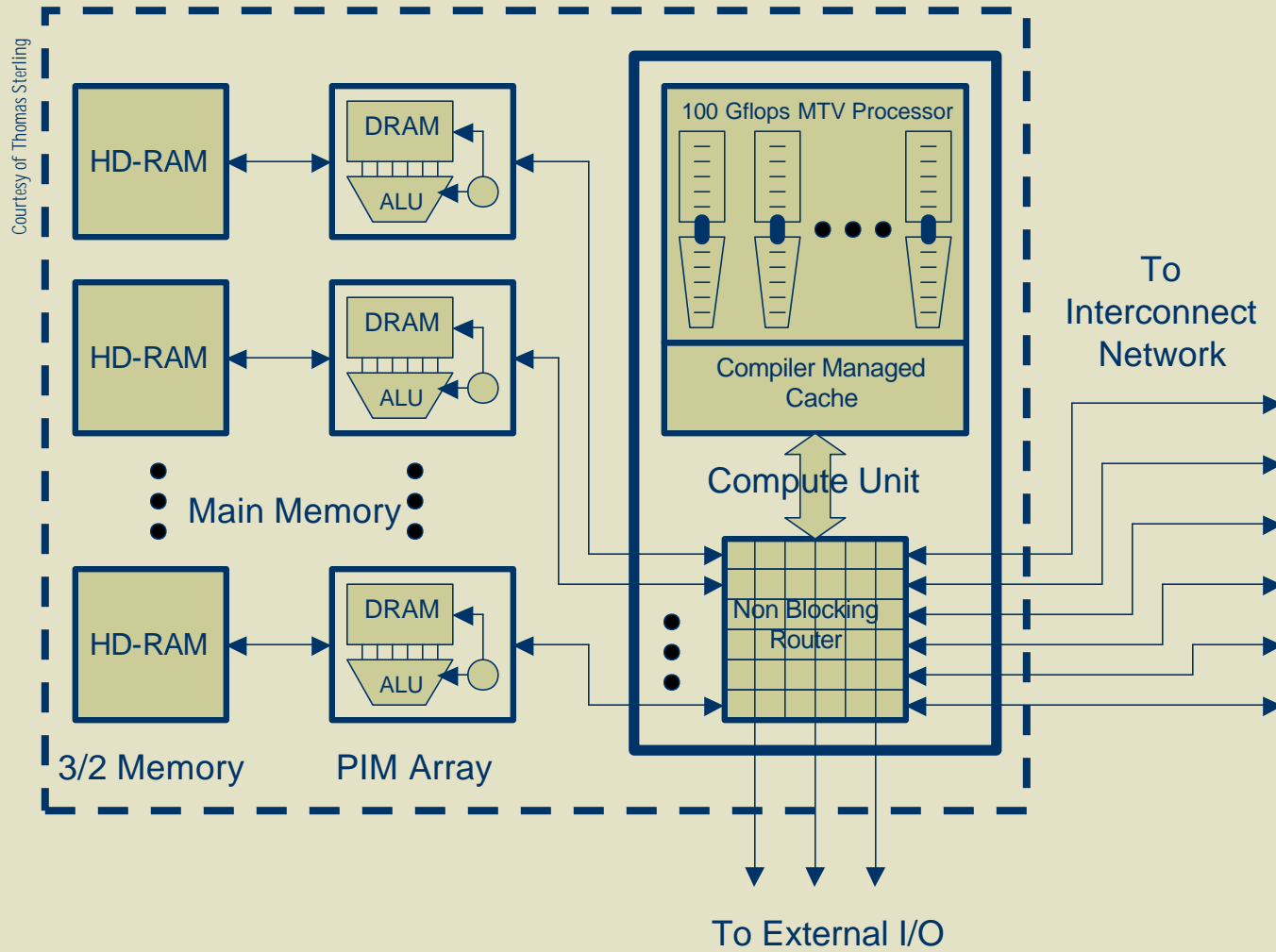


# Architectures





# Cascade Node





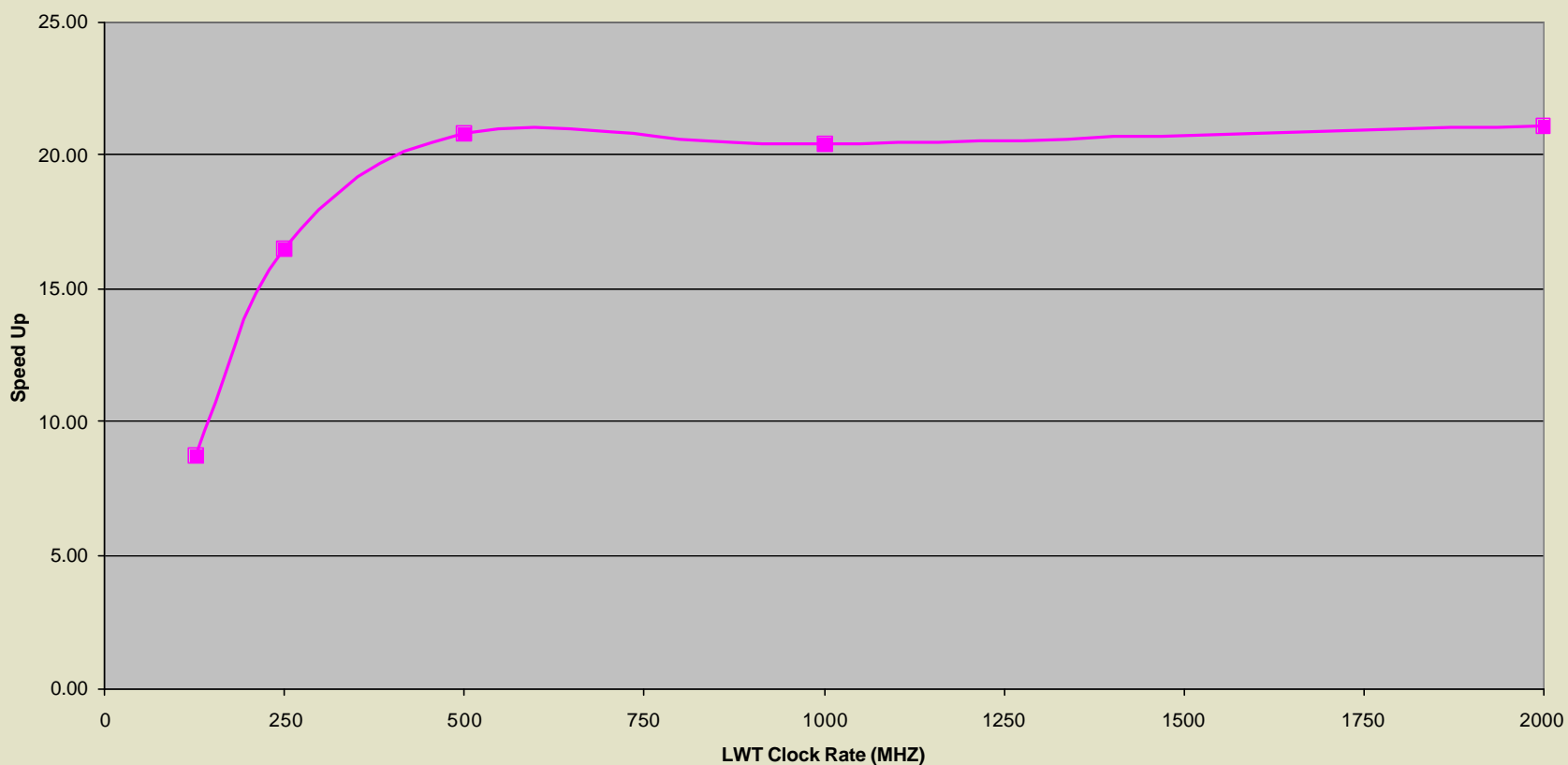
## Roles for PIM/MIND in *Cascade*

---

- ◆ Perform in-place operations on zero-reuse data
- ◆ Exploit high degree data parallelism
- ◆ Rapid updates on contiguous data blocks
- ◆ Rapid associative searches through contiguous data blocks
- ◆ Gather-scatters
- ◆ Tree/graph walking
- ◆ Enables efficient and concurrent array transpose
- ◆ Permits fine grain manipulation of sparse and irregular data structures
- ◆ Parallel prefix operations
- ◆ In-memory data movement
- ◆ Memory management overhead work
- ◆ Engage in prestaging of data for MTV/HWT processors
- ◆ Fault monitoring, detection, and cleanup
- ◆ Manage 3/2 memory layer



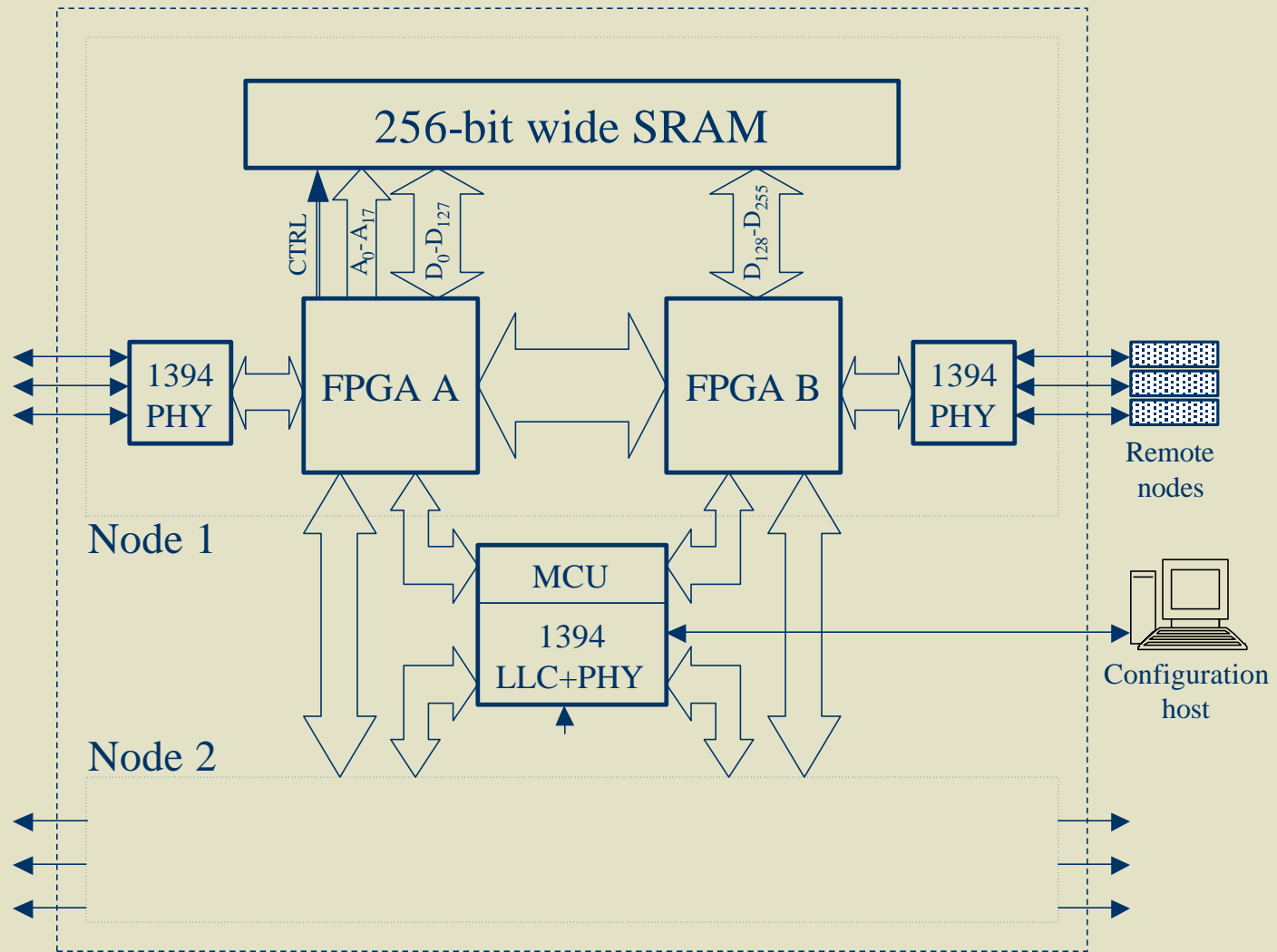
Speedup Smart Memory Over Dumb Memory for Various LWT Clock Rates  
64 Smart Memory Nodes





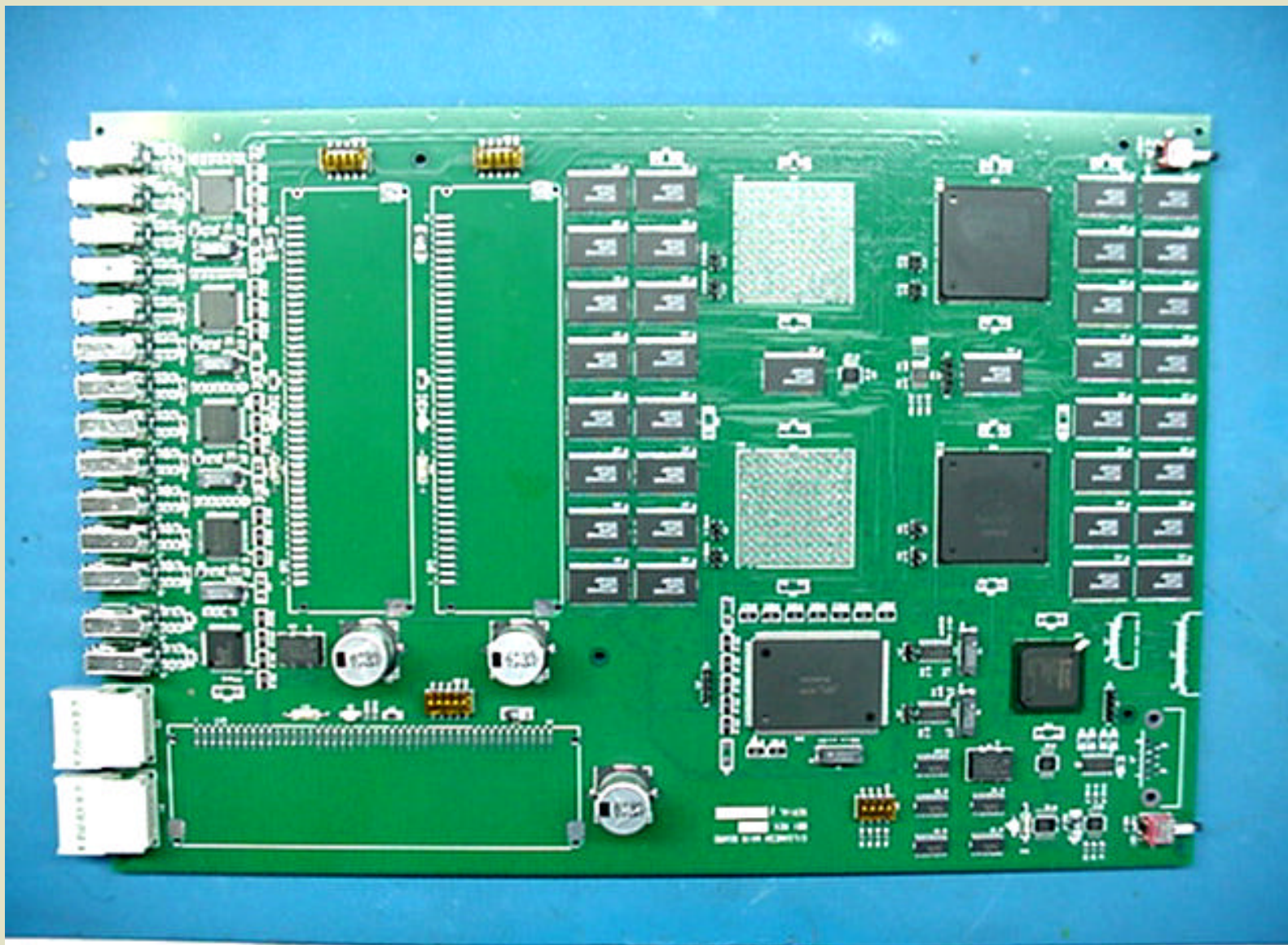
# FPGA-based Breadboard

- ◆ FPGA technology has reached million gate count
- ◆ Rapid prototyping enabled
- ◆ MIND breadboard
  - Dual node MIND module
  - Each node
    - 2 FPGAs
    - 8 Mbytes of SRAM
    - External serial interconnect for parcels
    - Interface to other on-board node
- ◆ Test Facility
  - Rack of four cages
  - Each cage with eight MIND modules
- ◆ Alpha boards near completion (4)
- ◆ Beta board design waiting next generation parts





# MIND Prototype



September 24,