PIMS IN SPACE

Processing-In-Memory (PIM): Capabilities and Opportunities for Deep Space Systems

Peter M. Kogge, kogge@cse.nd.edu
McCourtney Prof. of CS & Engr., IBM Fellow (retired)
Jay B. Brockman, jbb@cse.nd.edu
Assistant Prof., CS & Engr.
University of Notre Dame

PIMS IN SPACE

PIM Support

- DARPA, "A Hybrid Technology Multithreaded Computer Architecture for Petaflops Computing"
- NSF ACS96-12028, "Point Designs for 100TF Computers Using PIM Technologies"
- *JPL 960587, "Scalable Spaceborne Computers Using PIM Technologies"*
- *DARPA/SPPDG Group at Mayo Clinic, "PIM-Based Accelerator Technology Infrastructure Development"*
- NEC Research, "High Speed Image Retrieval Techniques Using PIM Technologies"
- NASA NAG 5-2998, "PIM-Based Architectures for Peta(fl)ops Potential"

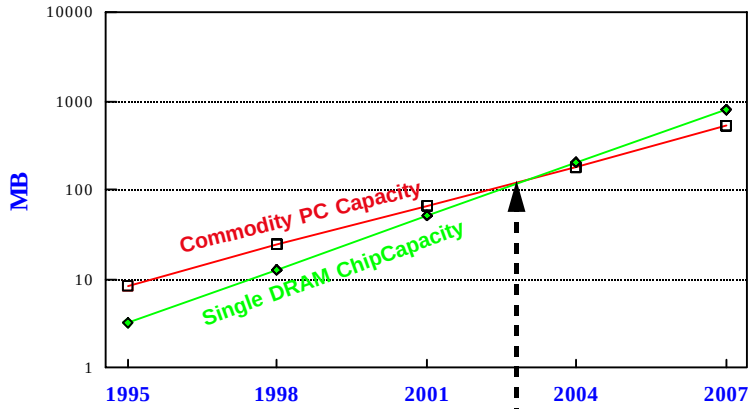
PIMS IN SPACE

Topics

- The Case for PIM
- Shamrock Tiled PIM Architecture
- REE Study Overview
- Followon Architectures
- Future Potential

PIMS IN SPACE

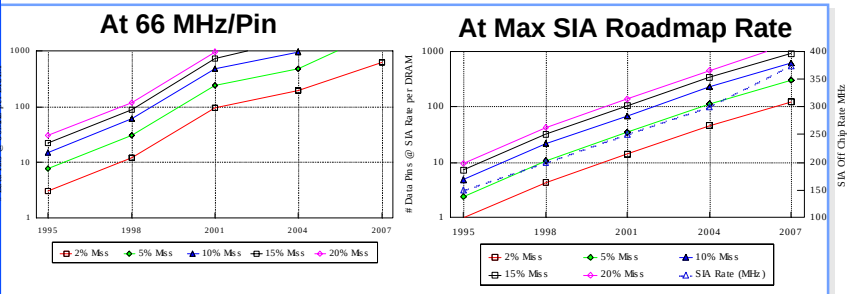
An Observation on Memory Density



ONE DRAM chip satisfies *ENTIRE* PC demand by 2003

PIMS IN SPACE

How Many Pins?



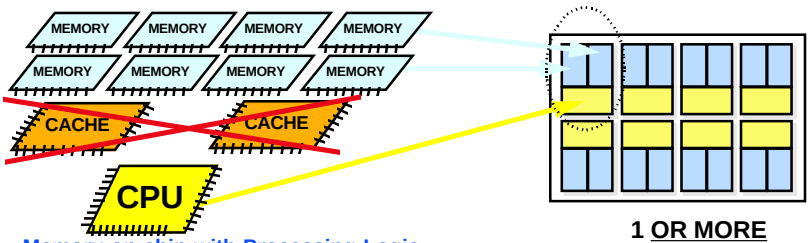
Bottom Line: Do we *really expect* multi hundred pin memory parts?

- ***Packaging*** **Complexity**
- ***Power*** **(chip contacts)**
- ***High power expensive cache*** **hierarchies**

PIMS IN SPACE

PIM=*Processing-In-Memory*

PIM = Mixing Significant *Logic* and *Memory* ON SAME CHIP



1 OR MORE Compute Nodes on ONE Chip

Memory on chip with Processing Logic:
=> More Bandwith/Less Latency
=> Less Cache/Simpler Design/Lower Power
=> New Architecture/Higher Per Chip Performance
=> No External Parts
=> Fewer Chip Contacts/Lower Power
=> Single Chip Type Designs/Highly Scalable

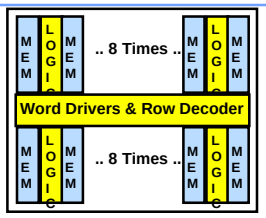
PIMS IN SPACE

Relevance to Space Systems:

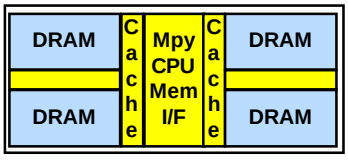
- Complete system on single chip => lower cost
  - At most 1 new space qual needed
  - Direct transition to next generation technology
- Better silicon usage => less chips, *power*, weight
  - Direct utilization of on chip memory bandwidth
  - Elimination of mem hierarchy off chip contacts
  - Permits simpler CPU architecture
- Natural Scalability, Fault Tolerance
- Elegant programming models: SIMD, MIMD, SPMD
  - "In the Memory" naturally parallel computing
  - Locally shared, globally distributed
  - Very high internode bandwidth

PIMS IN SPACE

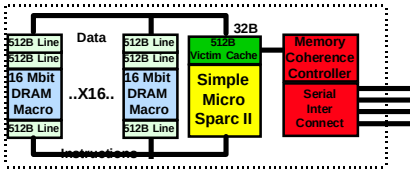
PIM Architectural Spectrum



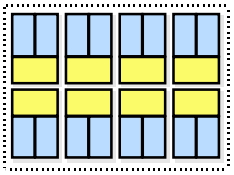
(a) SIMD: Linden DAAM



(b) Single Chip Computer: Mitsubishi M32R/D



(c) SMP Node: Proposed SUN part



(d) Tiled: EXECUBE

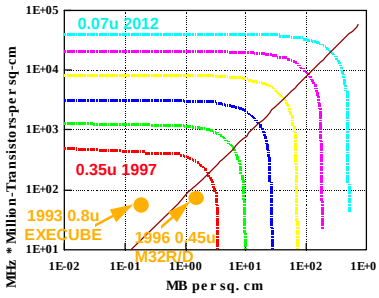
PIMS IN SPACE

Sample PIM Chips

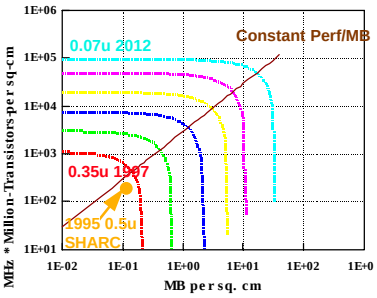
Chip	Organization	First Silicon	Peak Perf	Storage	Watts	MB /Perf	Perf /Watt
EXECUBE [Kogge, 94]	8 Way Hypercube	1993	50 Mips	0.5 MB	2.7	0.01 MB/Mip	19 Mips/Watt
Terasys [Gokhale, 95]	64 1-bit ALUs	1993	625 M bit ops	16 KB		0.000026 MB/bit op	
AD SHARC [AD, 93]	DSP + Memory	1994	120 MF	0.5 MB	1.5	0.005 MB/MF	80 MF/Watt
TI MVP [Guttag, 93; TI, 94]	1 CPU + 4 DSPs	1994	2000 Mops	0.05 MB	7.5	0.000025 MB/Mop	260 Mops/Watt
Linden DAAM	1024 8-bit ALUs	1996	862 Mops	0.5 MB	0.7	0.0006 MB/Mop	1200 Mops/Watt
MIT MAP [Dally, 93; Keckler, 92]	4 64-bit CPUs	1996	800 MF	0.128 MB		0.0016 MB/MF	
M32R/D [MITSU, 96]	32-bit RISC + Memory	1996	51 Mips	2 MB	0.27	0.04 MB/Mip	189 Mips/Watt

PIMS IN SPACE

PIM Design Space



DRAM PIM



SRAM PIM

PIM VLSI constraints (as of today):

- Approx. *2 year* lag between SIA announce & PIM parts
- DRAM Memory is "Array" segments, with <30 ns access
- DRAM logic supports only 3 levels of metal => Area penalty
- DRAM logic: 1/2 speed of conventional logic

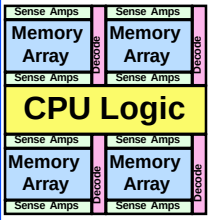
PIMS IN SPACE

ND PIM Projects

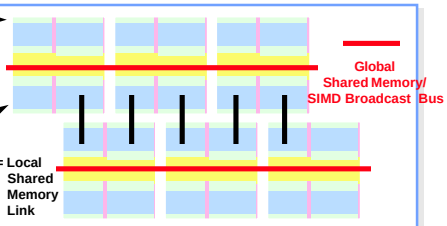
- **Completed**
  - DARPA: PIM Technology
  - NASA: PIMs for Petaflop; NSF: 100 TF Point Design
  - JPL REE Study: PIMs for Deep Space
  - NEC: PIMs in Image Database Search Engines
- **Ongoing**
  - HTMT Petaflop design study
  - *PIM FAST* design: early PIM for space
  - *ASAP:* Pitch matched "At the Sense Amps Processing"
  - 8 way "Smart Memory" PIM accelerator using M32R/Ds
  - New memory semantics & programming models

PIMS IN SPACE

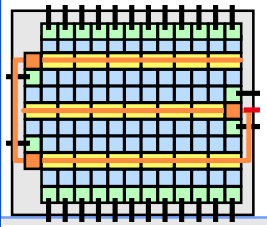
ND's Shamrock



Basic Silicon Macro



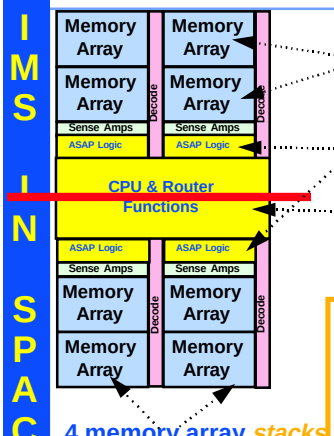
Tiled over chip into locally 4-way shared memory MPP



To Create:
- Single Chip MPP
- "Looks like" memory
- Hi Speed I/O

PIMS IN SPACE

Potential Shamrock Node



4 memory array *stacks* accessible to CPU

Concurrent accesses of up to 256 bits in each 128KB to 1 MB stacked array

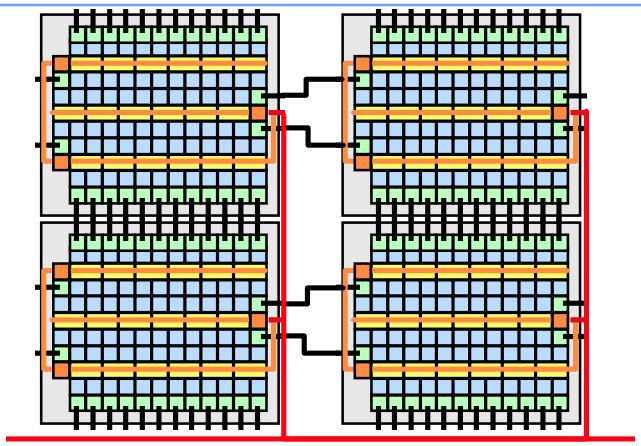
At the Sense Amp Processing (*ASAP*)
- SIMD processing on entire access
- Fast data routing between stacks

CPU: (Goal: Simplicity & *try* to avoid cache)
- Simple VLIW to utilize full memory width
- Multithreaded to hide remaining DRAM latency

Potential DRAM Design Characteristics:
- up to 500 MHz Clock (at end of SIA)
- <30 ns local access (<15 cycle latency)
- 4-6 accesses pending per memory stack
- Potential of 1.6+ GB/s bandwidth

PIMS IN SPACE

Multiple Chip Configuration

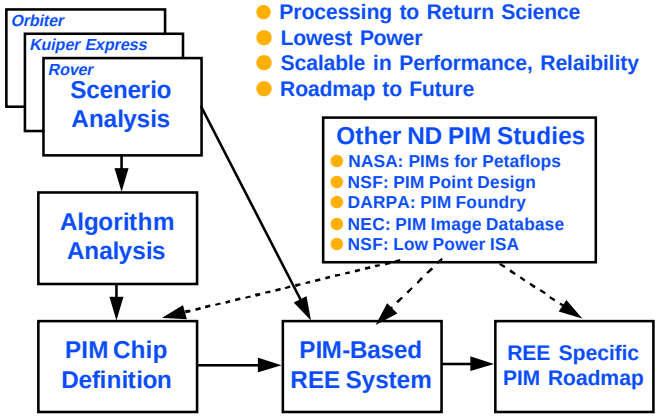


Multiple chips scale to *architecturally constant* topologies

PIMS IN SPACE

1996 ND REE PIM Study

Approach:



Objectives:

- Processing to Return Science
- Lowest Power
- Scalable in Performance, Relaibility
- Roadmap to Future

Other ND PIM Studies

- NASA: PIMs for Petaflops
- NSF: PIM Point Design
- DARPA: PIM Foundry
- NEC: PIM Image Database
- NSF: Low Power ISA

PIMS IN SPACE

Scenerio Analysis Summary

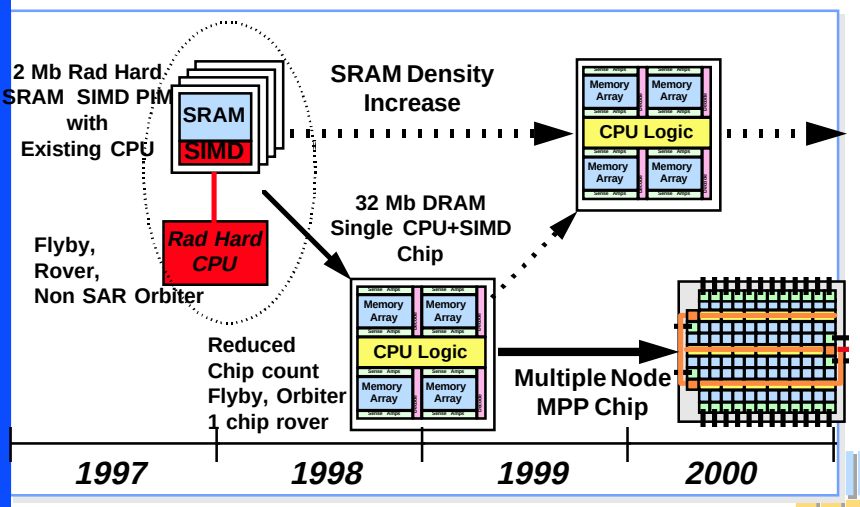
- Flyby: Autonomous Mapping, Satellite Search
- Nanorover: Autonomous Nav/Mapping, Feature Search
- Orbiter: Change Detection, Data Downlink Reduction

Key Science

Algorithm	Flyby	Nanorover	Orbiter
Histogram	✔		✔
(De)Compression	✔	✔	✔
Registration	✔	✔	✔
Thresholding	✔		✔
Image Combination	✔		✔
Classification	✔		✔
Filter Correlation	✔	✔	✔
Region Growing	✔		✔
2D Interpolations			✔
2D FFTs			✔
Path Planning	✔	✔	✔

PIMS IN SPACE

REE Study Suggested Roadmap



PIMS IN SPACE

REE Chip Projections

Basis for a *Perfect* Single 1/4 chip Nanorover

A *Perfect* Single chip Flyby

REE Chip Date	1998	2001	2004	2007	2010	2013
Feature Size	0.35	0.25	0.18	0.13	0.10	0.07
Vdd	3.3	2.5	1.8	1.5	1.2	0.9
Node mW	1700	952	300	141	77	35
Node Mops/W	641	1,144	3,631	7,748	14,094	31,472
Node (mm2)	116	51	25	13	6	3
Nodes/Chip	1	2	4	9	20	36
MB/Chip	2	4	8	18	40	72
Mflops/Chip	66	132	264	594	1,320	2,376
mW/Chip	1704	1909	1202	1267	1547	1246

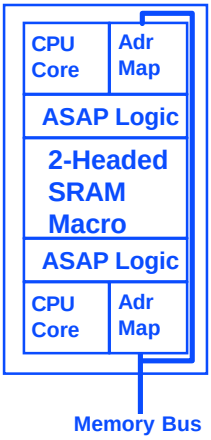
PIMS IN SPACE

PIM FAST

- Followon to REE Study
- Attempt to:
  - Bring more GP processing in earlier
  - Emphasize reduced power
  - Support multiple scaling architectures
  - Begin to explore Shamrock-like architectures
  - Allow near term fabrication

PIMS IN SPACE

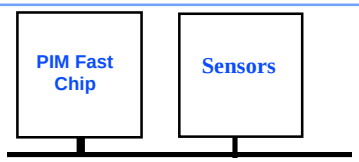
PIM *Fast* - the Chip



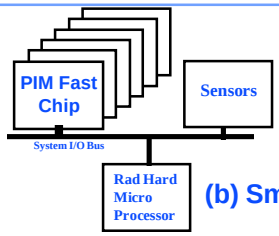
- Looks like memory to outside
- StandardCPU core for GP operations
  - Chosen for *optimal Mips & Watts*
  - Minimal/no CPU core modifications
- ASAP logic for fixed point vector ops
  - Controlled via STOREs from CPU
- Power Management logic to control clocks, etc
- Dynamically reprogrammable Mapping Logic
  - Memory mapped address translation tables
  - Permit fault tolerance/application virtualization

PIMS IN SPACE

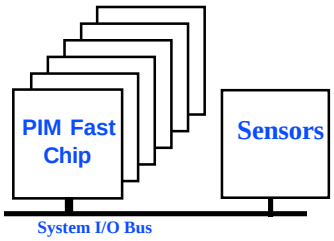
PIM *FAST* System Architectures



(a) Single Chip Systems



(b) Smart Memory

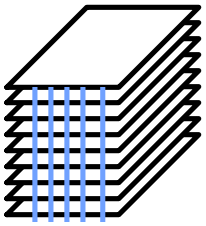


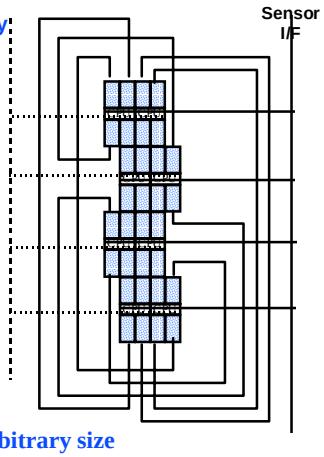
(c) Dedicated Task Mixed Shared/Local Memory

PIMS IN SPACE

A Stacked PIM Fast Configuration

A Single PIM Fast Chip Configured as Shamrock





A Stack of 9 PIM Fast Chips
- Looks like memory to outside
- Any chip - any place in memory space
- Programmable as wrapped toroidal mesh of arbitrary size
  - OR as 8 separate nodes on common bus

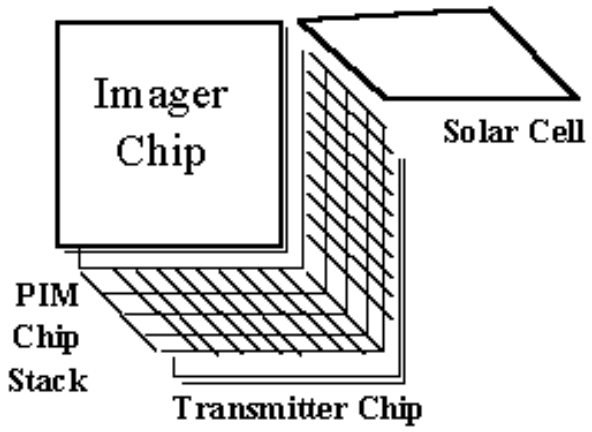
PIMS IN SPACE

Discriminators from Today's PIMs

- Naturally scalable PIM architecture
  - From single chip systems
  - To multi chip smart memory subsystems
- Begins to demonstrate inherent PIM features
  - especially as we add "at the sense amps" processing
- Architected for embedded environment
  - Natural followons: rad hard, low power, dense 3D stacking
  - Fault Recovery via mapping reconfiguration
- Simplified programming model
- Compatible with other existing components
- Clear path to following generations

PIMS IN SPACE

Ultimate PIM Configuration



PIMS IN SPACE

Conclusions

- PIM: Coupling of new VLSI opportunities with new architectures and simple execution models
- Resulting Space Opportunities:
  - Reduced power/weight
  - Enhanced scalability, fault tolerance
  - Significant growth potential
- A *Paradigm Shift*