

# The RM9000 Family of Integrated Multiprocessor Devices

Paul Cobb, PMC-Sierra Inc.  
Technical Advisor, MPD  
Hot Chips 14: August 2002

## Outline

- What is an integrated multiprocessor?
- Why take integration to this level?
- Application in networking systems
- Internals: caches, interconnect
- Application & performance examples
- RM9000x2 statistics
- Future directions
- Summary

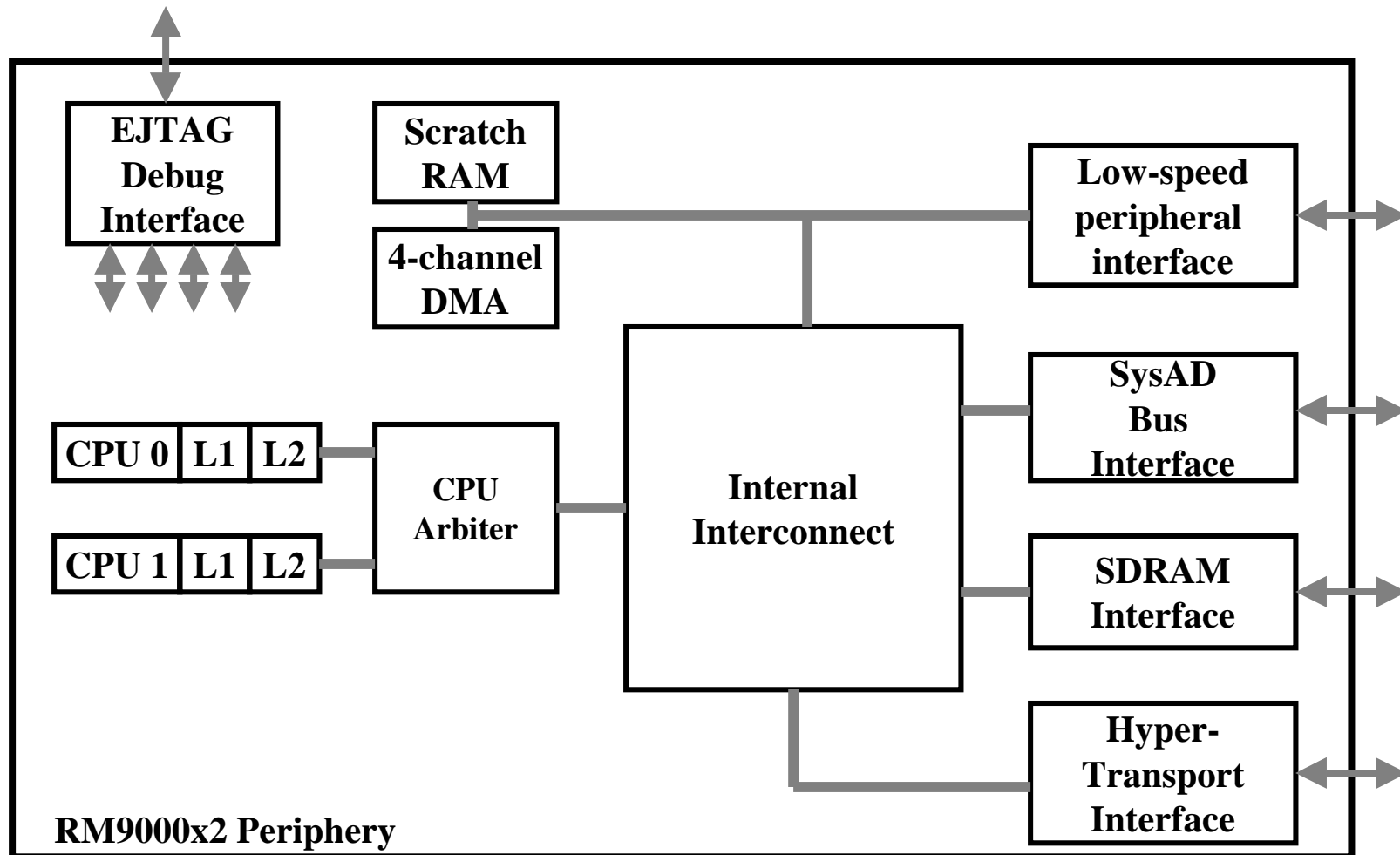
## RM9000 Device Resources

- **Multiple 64-bit MIPS CPUs**
  - Including private 2-level cache hierarchy.
- **Memory subsystems**
  - Control external devices, e.g. SDRAMs
  - Internal SRAM for fast local storage.
- **I/O controllers**
  - With dedicated and/or shared DMA engines.
- **Internal interconnect**
  - Move data efficiently between other resources.

## RM9000x2

- **Interface controllers:**
  - HyperTransport: 8 bits/dir, 500 MHz DDR
  - SysAD: 64 bits, 200 MHz
    - Compatibility with existing ASICs etc.
  - SDRAM: 64 bits, 200 MHz DDR
- **Processor complex:**
  - Dual 64-bit CPUs @ 1 GHz;
    - 2-way superscalar (appropriate for embedded)
    - 16 KB L1-I, L1-D; 256 KB L2; full MMU and FPU
- **Internal resources:**
  - 4-channel DMA controller; 8 KB Scratch RAM

# RM9000x2 Block Diagram



## Why take integration to this level?

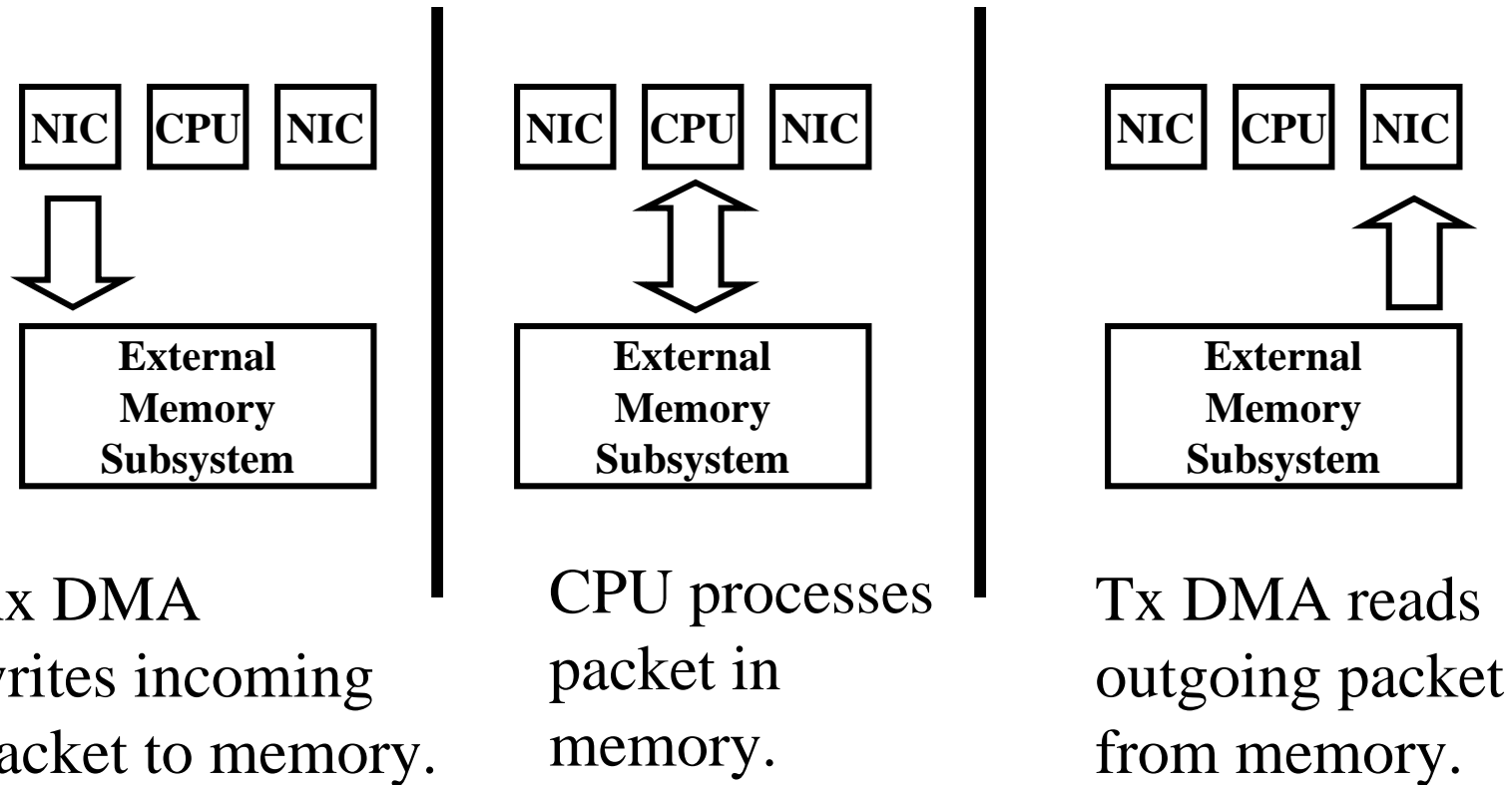
Networking subsystems must typically:

- Handle aggregate I/O bandwidths on the order of Gbits/sec.
- Make efficient use of CPU resources:
  - Run complex routing/topology algorithms.
  - Accumulate diagnostic and accounting info.
- Exploit the flexibility of software:
  - Reuse same hardware in multiple products.
  - Track evolving standards with field upgrades.

## Application in Networking Systems

- RM9000 devices can serve in a wide variety of processing roles:
  - Preserve and extend the networking industry's substantial investment in MIPS software.
- Fully general-purpose CPUs
  - Well suited to heavy computational demands of control-plane and management tasks.
  - Familiar programming model and software tools.
- Tightly integrated Memory and I/O
  - Provide appropriate latency and bandwidth for use directly in data-plane processing.

# Basic Packet Processing



## Limitations of basic implementation

- Memory subsystems sees combined demands of ALL traffic types:
  - Packet movement (store & forward)
  - CPU traffic (forwarding lookups, etc.)
  - Descriptor access by CPU & DMAs, for buffer memory management.
- CPU memory access not efficient
  - Commodity memory technologies are not well suited to typical access patterns.
  - Arbitration delays add to memory device latency.

## Cache Hierarchy

Need to bring incoming packet headers as close as possible to the CPU(s):

- Transparent redirection of incoming data to L2 cache of one or both CPUs:
  - Auto deposit: first N bytes (packet header).
  - Live deposit: entire DMA or HT transfer block.
- **Fast Packet Cache:**
  - Refill requested line to L1 Data cache, leaving L2 cache unmodified.
  - Preserves longer-lived info in L2.

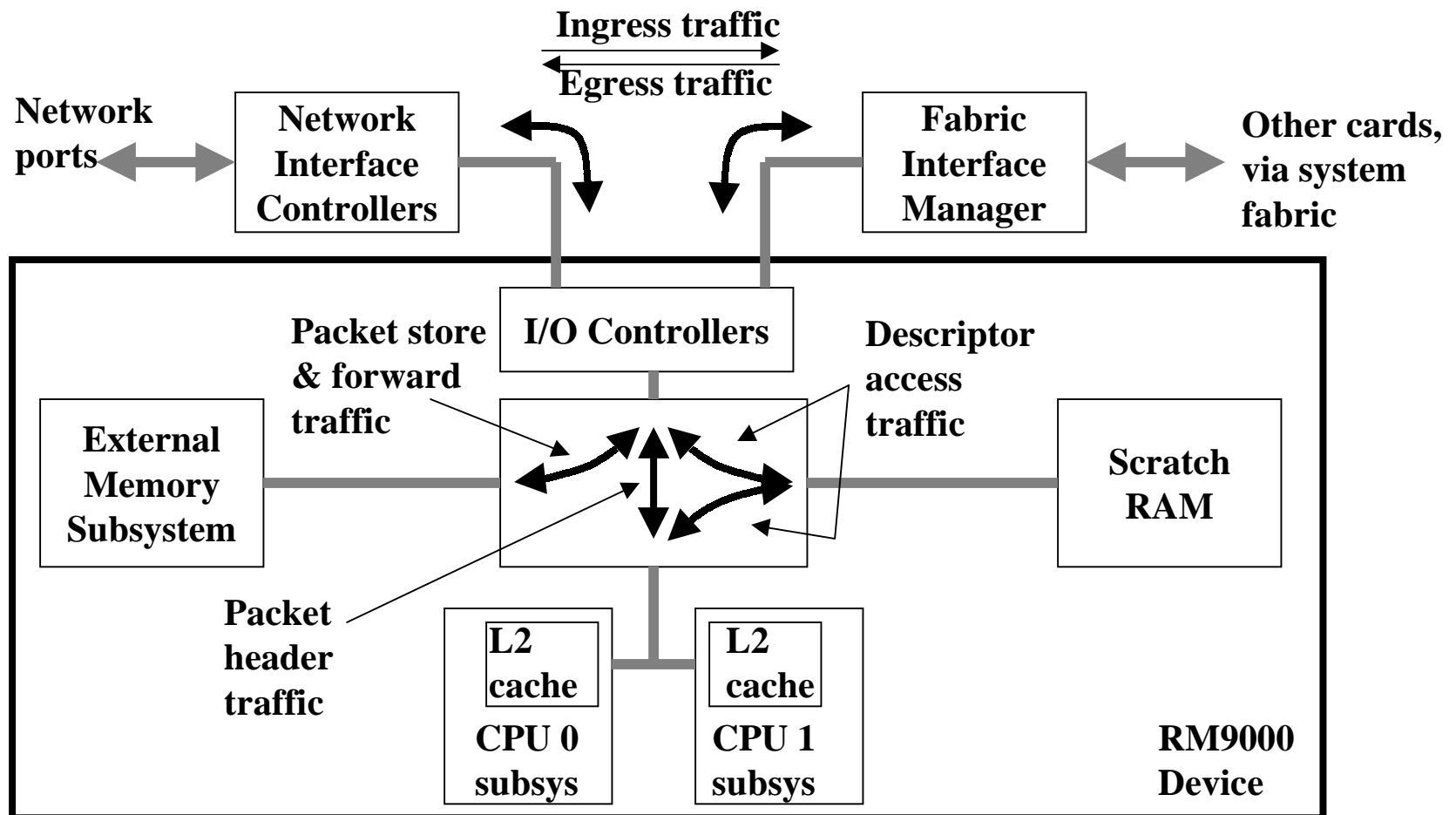
## Internal Data Movement

- Internal transfers pass through a centralized pool of shared buffers.
- Runs at up to half the CPU pipeline clock rate (currently 500 MHz).
- Multiple ports (currently 5).
- 64 bits per clock edge per port.
- All ports can transfer concurrently.
- Peak bandwidth: 40 Gbits/sec/port.

## Enhancing the Cache Coherency

- Maintained in hardware:
  - Across CPU subsystems
  - On I/O transfers
  - Selectively enabled under software control.
- Shadow tags at L2 caches minimize overhead at CPU pipelines.
- DMA and external agents move information directly to/from L2 caches
  - Minimizes CPU exposure to Memory and I/O latency.

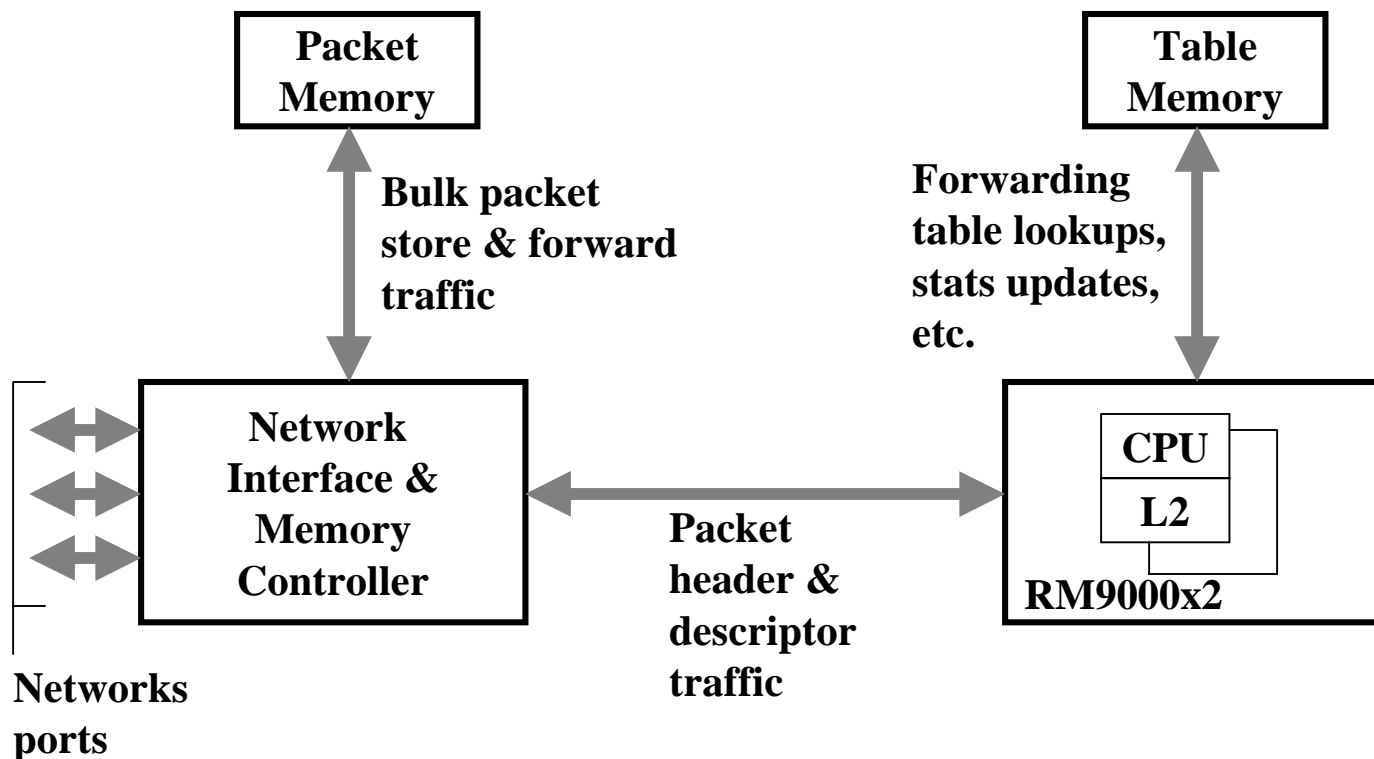
# Data Movement Example #1



# Data Movement Example #2

**Goal: maximize bandwidth,  
with strongly sequential  
reference patterns**

**Goal: minimize latency,  
with weak locality of  
reference**



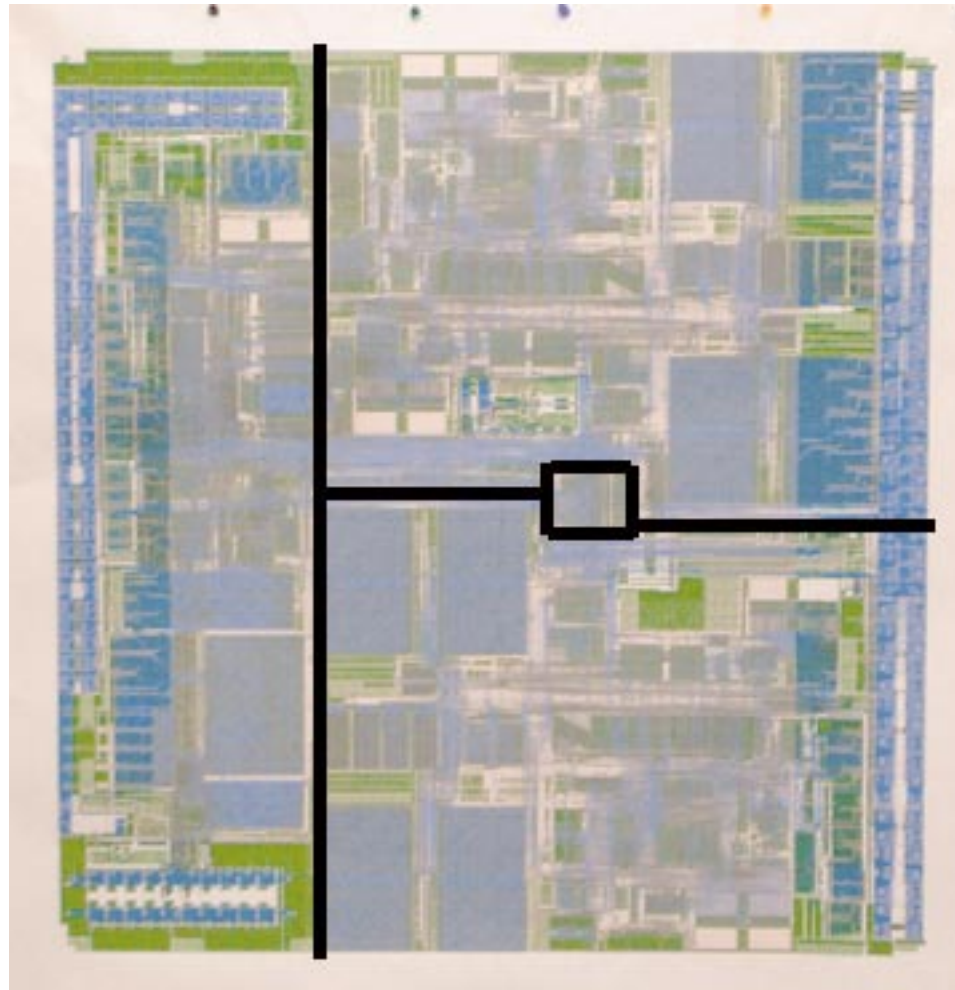
## Performance Example

- Basic packet processing benchmark
  - IPv4 forwarding per RFC1812
  - No IP options
- Maximize internal transfer efficiency:
  - HT auto-deposits headers to L2
  - DMA moves modified headers from L2
  - Hold descriptors in internal Scratch RAM
- Sustained forwarding rate: 3.3 Mpps
  - 64 Byte packets
  - Performance directly observed in RTL simulation.

## RM9000x2 Stats

- **Technology:**
  - 0.13 um (drawn)
  - 8 level metal (1 for bumps, 2 for power planes)
  - 672 pin FCBGA
- **Die usage:**
  - ~50M transistors total: ~20M per CPU subsys.
  - CPU subsystems, incl. caches: ~56% of die.
- **Status:**
  - First silicon June '02: now operational in lab.
  - Sampling: Aug '02; Production estimated 1Q03

# RM9000x2 Die Image



## Multi-Processing Examples

- **Ex #1: Simple software partitioning:**
  - One CPU handles all control and management operations.
  - Other CPU can be dedicated to running highly tuned low-level packet processing code.
  - Greatly simplifies scheduling issues.
- **Ex #2: Pooled CPUs for high-touch applications:**
  - Use hardware cache-coherency for efficient sharing of data structures among CPUs.
  - Distribute computationally expensive forms of processing among multiple CPUs.

## Future Directions

Extend the range of on-chip functions:

- I/O and Memory controllers:
  - Track evolving industry standards.
  - Match growing application demands.
- CPU subsystems
  - Vary number & attributes of CPU subsystems to serve other markets, e.g. digital imaging.
- Other peripherals
  - Enhanced DMA, local storage; application-specific accelerators.

## Summary

RM9000 device family demonstrates that integrated devices can:

- Open up new roles for general-purpose processors within networking systems.
- Help balance low latency against high bandwidth, to suit the system workload.
- Deliver unprecedented performance and flexibility, within an appropriate power budget.