

MOSAID Semiconductor

Fabr-IC

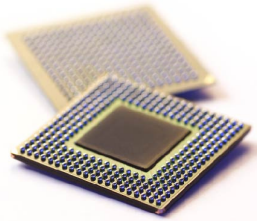
(A Single-Chip Gigabit Ethernet
Switch With Integrated Memory)

@Hot Chips

Dave Brown

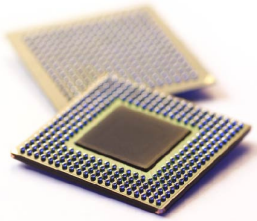
Chief Architect

July 4, 2001



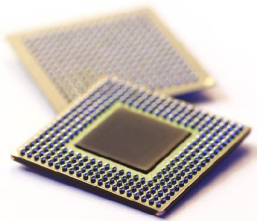
Fabr-IC Feature summary

- 2 Gig ports
- 1 gig port for stacking expansion (up to 2+96)
- 24 fast (10/100MHz)
- 256 VLANs
- 4 QoS
- link aggregation (trunking)
- mirroring
- SNMP, RMON statistics
- 16K address table on chip
- PCI for host interface and packet exception



Fabr-IC Feature summary

- 4Mbyte shared buffer
- L2/3/4 Policy engine
- L2 bridging
- L3 routing
- IP Multicast
- Flexible scheduling
 - strict priority
 - bandwidth limit
 - round robin



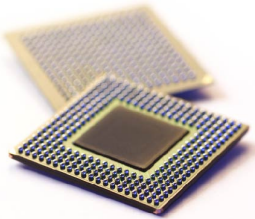
Innovations

- **Architectural**

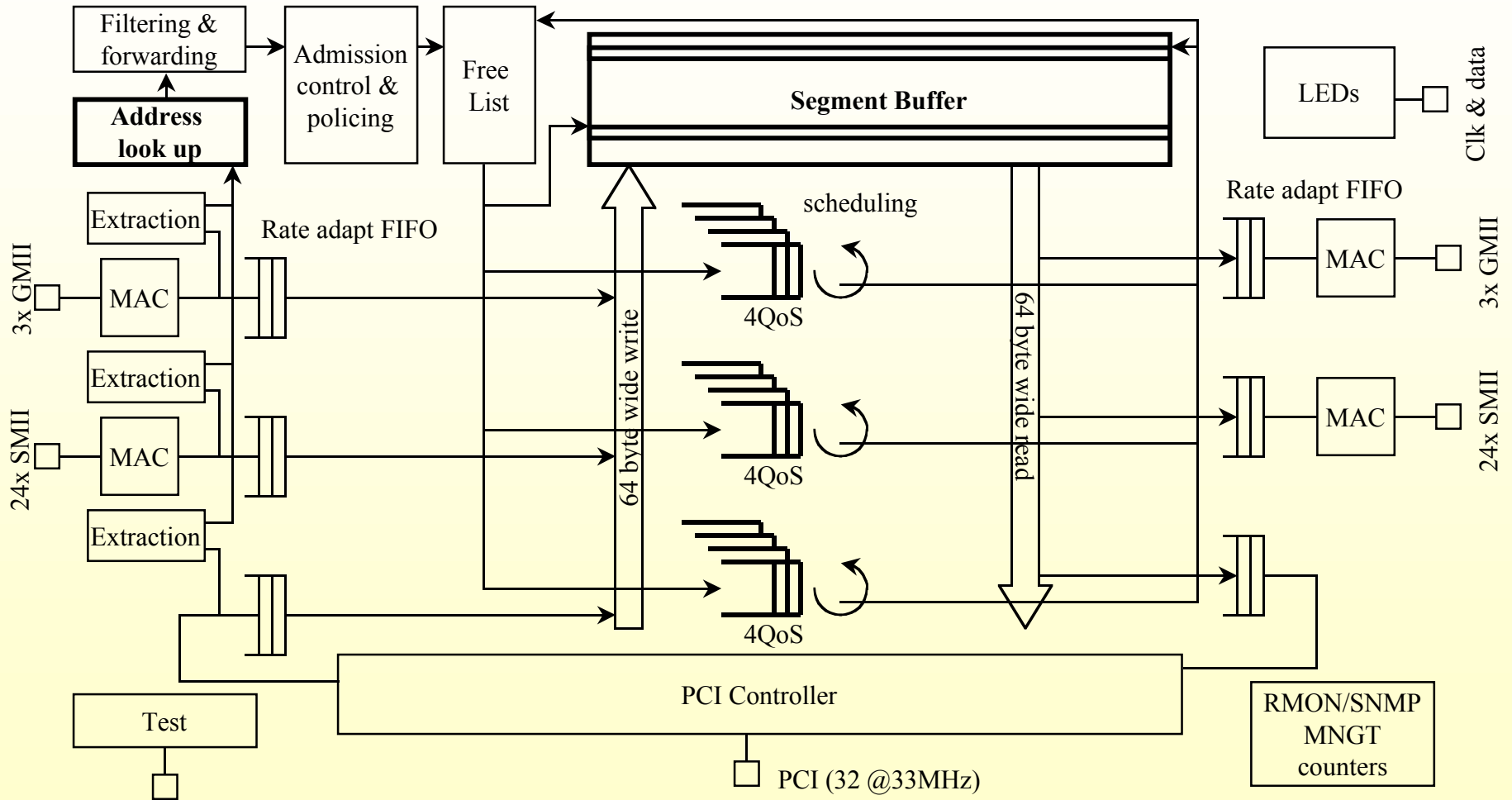
- 4 way interleaved buffer
- full “spatial” multicast
- quad hash table

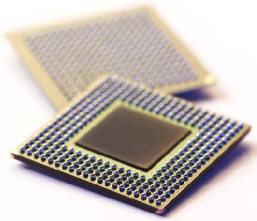
- **Technology**

- use of custom embedded DRAM, (eDRAM)
- 10:1 density over SRAM
- very wide I/O > 512 bits (high bandwidth)
- enables large buffer and address table



Architectural Overview

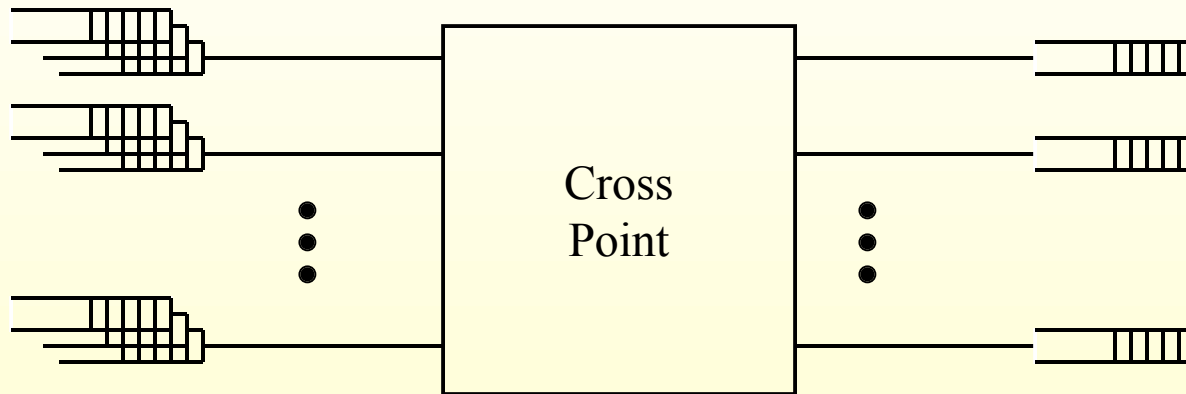




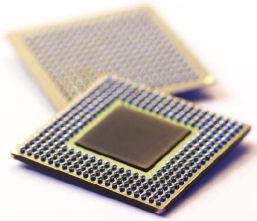
Cross Point

Input queues
(per output port)

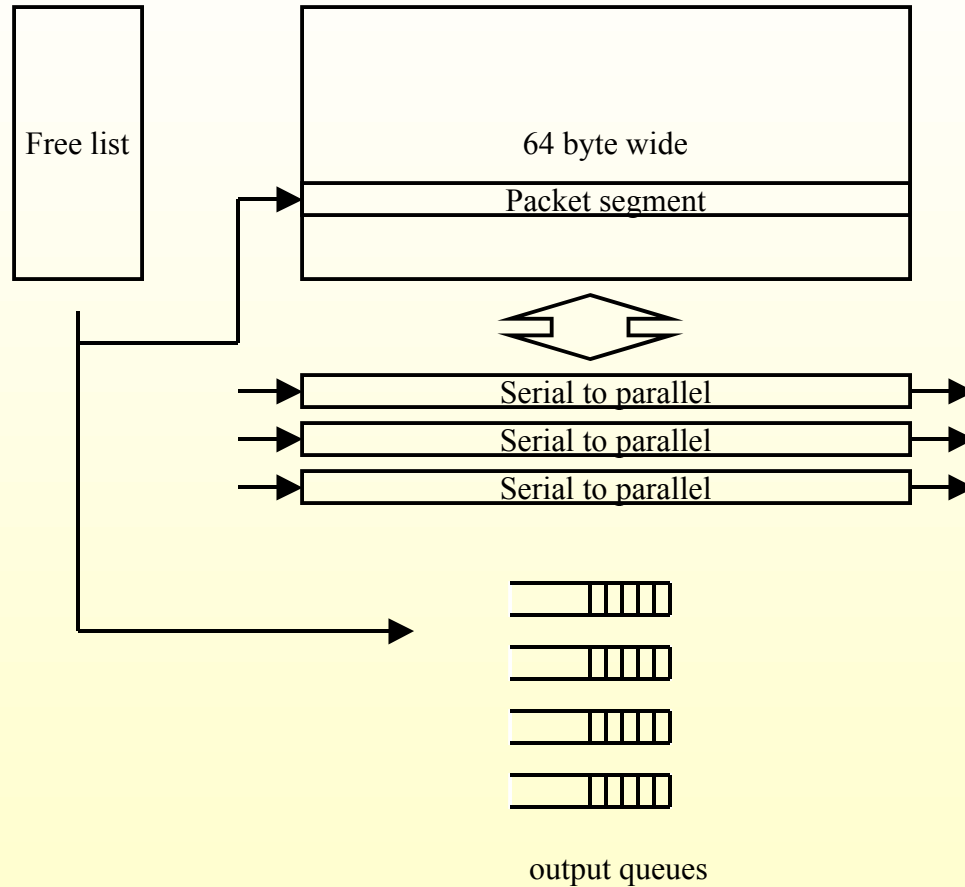
Output queues



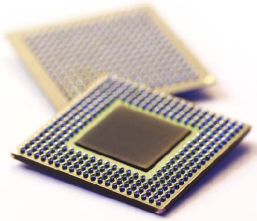
Single QoS shown



Common Memory



Pointer from free list indexes packet buffer, and is then stored in output queue(s)



2 way interleaved

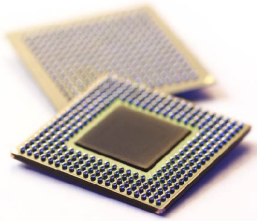
2X bandwidth achieved

Ingress writes to
opposite bank

Bank A

Egress pulls from bank
A or bank B

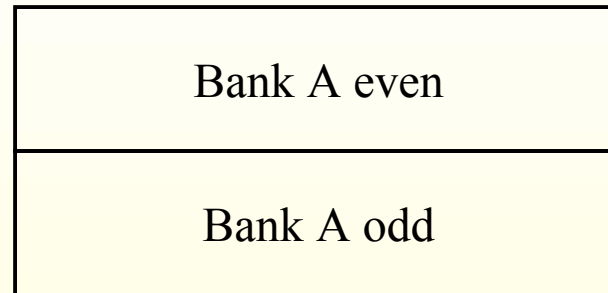
Bank B



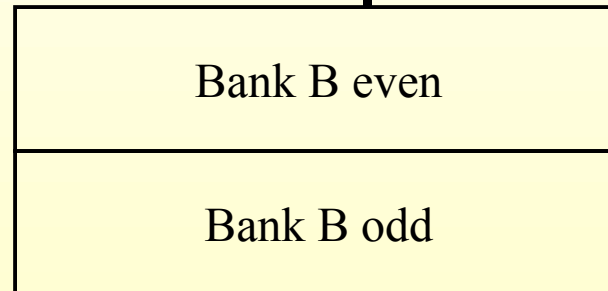
4 way interleaved

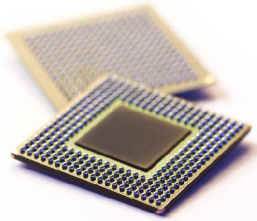
4X bandwidth achieved

Packet segmentation
is alternately even
and odd



Overcomes 65 byte
packet which would
require 2x bandwidth
to solve

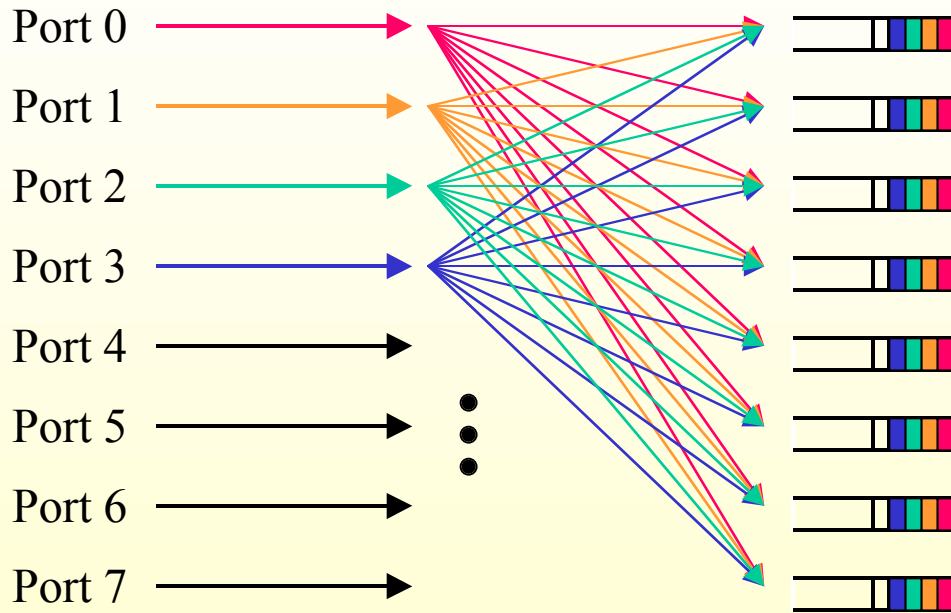




Multicast: N squared problem

Incoming packets

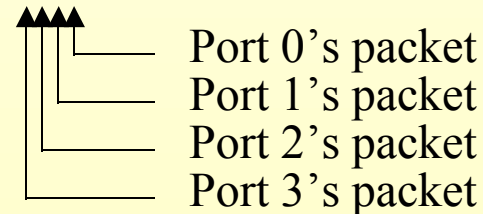
Output queues

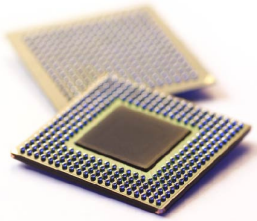


Output queues require N squared operations per system cycle, worst case

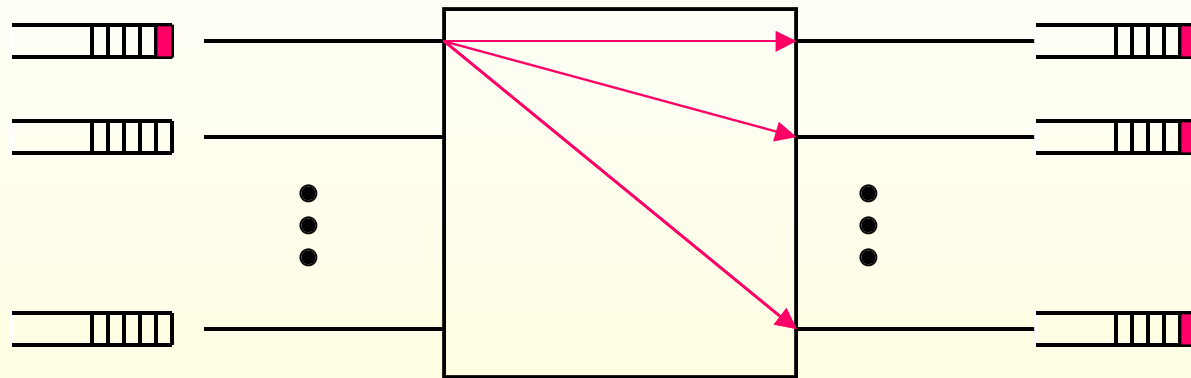
Could create N separate memories

Issue: this does not allow statistical sharing between queues





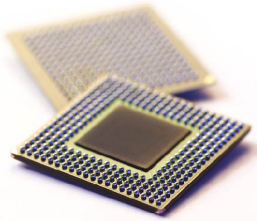
Multicast: cross point issues



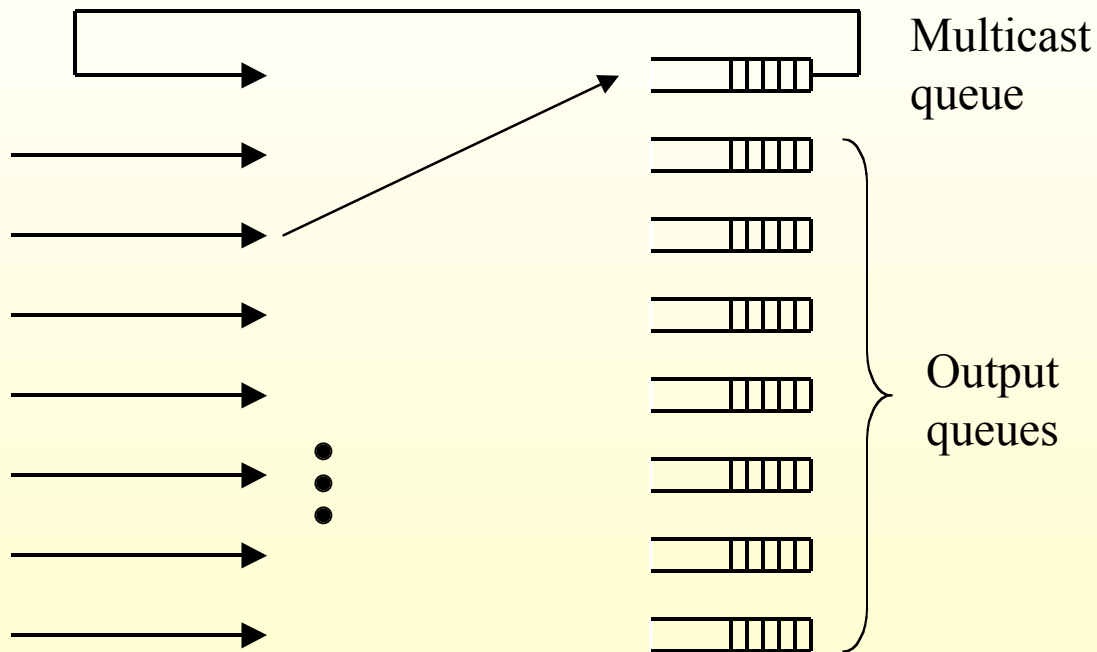
One port broadcasts all other output queues

Issue: Multiple copies of packet created

Issue: blocks other inputs

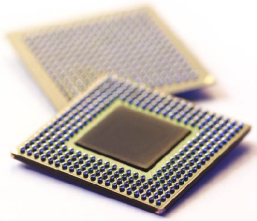


Multicast: job queue

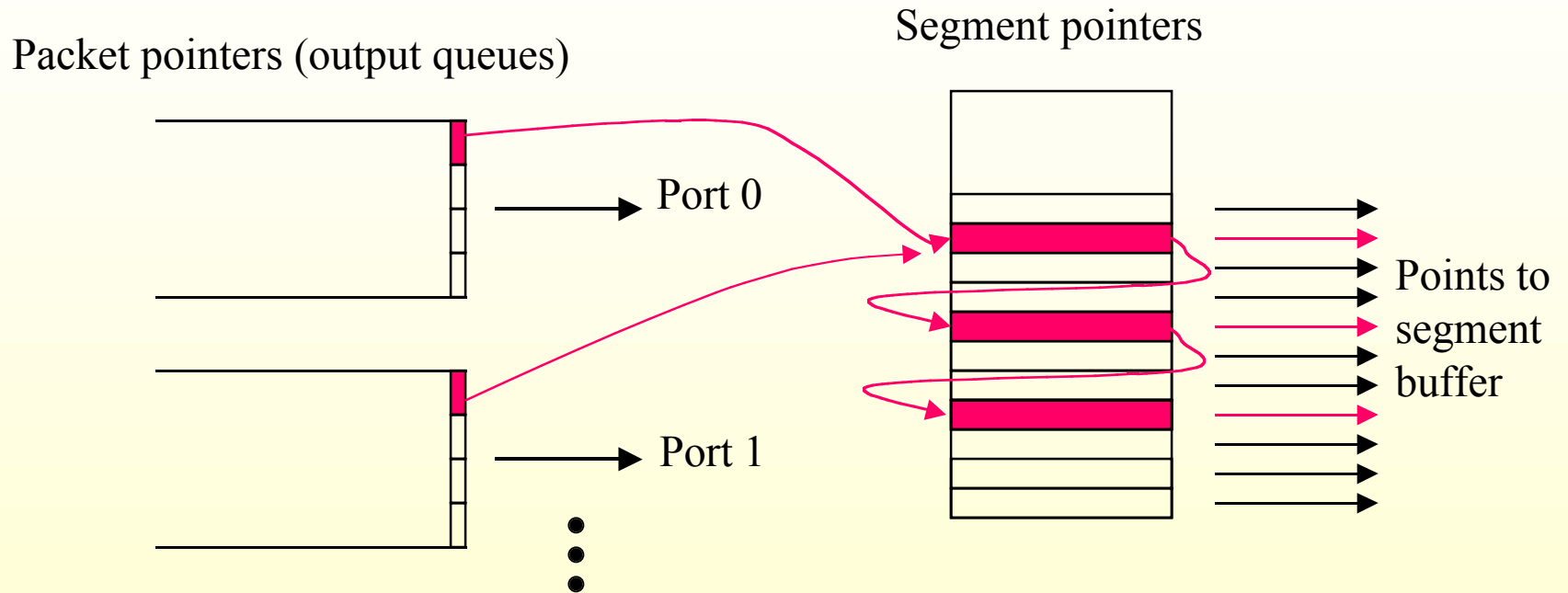


All multicast packets sent to one queue. Packets are then recirculated, a copy at a time, to the output queues

Issue: high latency on multicast, and limited BW for multicast



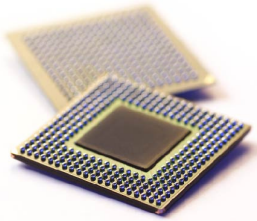
Multicast: what we did



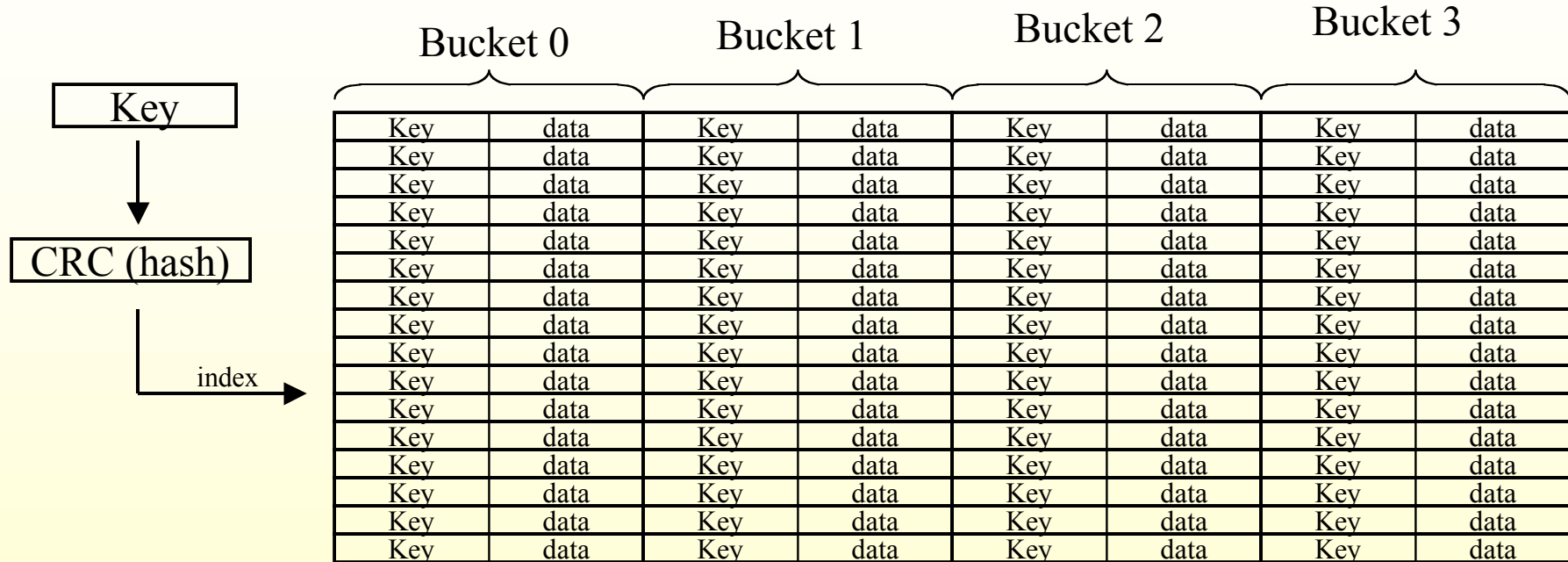
Queues are made wider to allow multiple pointers (from each ingress port) to be written in one cycle

Issue: required very wide memories

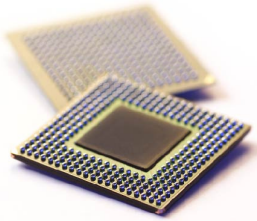
Solution: use wide eDRAM



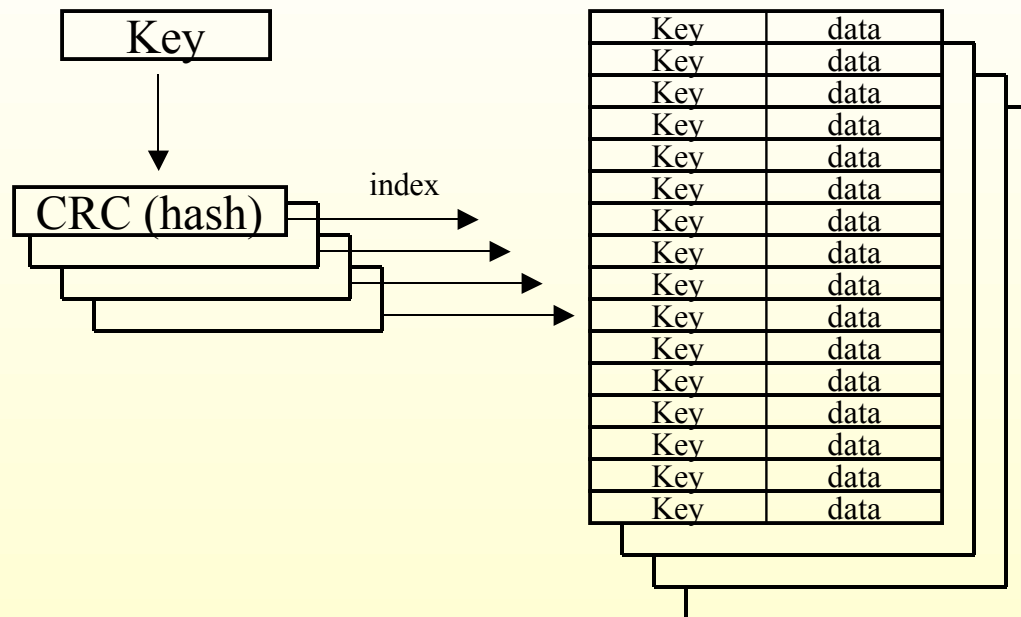
Typical multi bucket

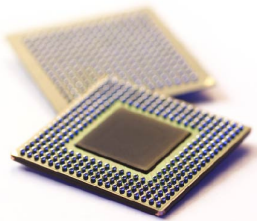


Key = MACDA+FID
data = forward vector, etc

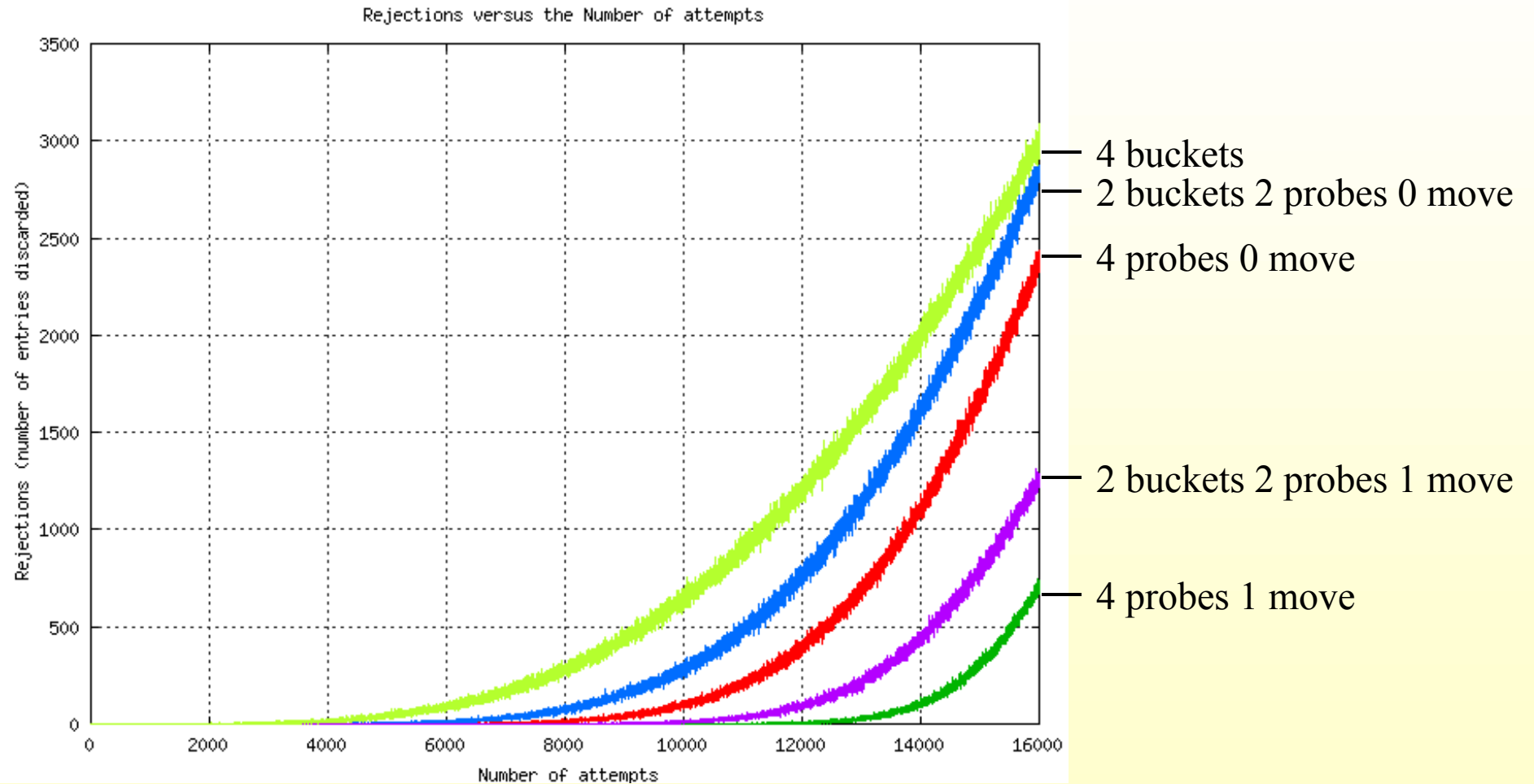


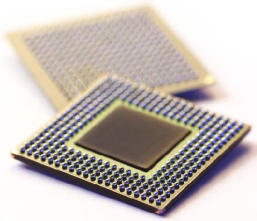
Multi probe





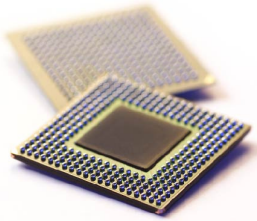
Simulation results





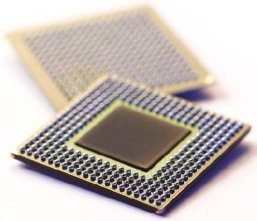
Implementation Statistics

- 2.1 Million gates
- 21 DRAM instances
- 5 DRAM macros
- 75 memory instances
- TSB TC260D (0.18/0.14)
- Merged DRAM & logic process
- COT flow
- 4.0 W
- 520 pins, SBGA, 40mm on a side, 1.27mm ball pitch



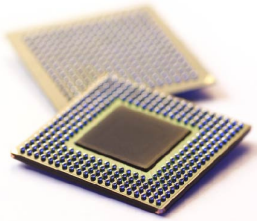
Implementation Tools

- Custom flow for DRAM
 - Composer (Cadence) schematic capture
 - Virtuoso-XL (Cadence) custom Layout
 - HSPICE (AVANT!)
 - Star Sim (AVANT!)
- ASIC/COT flow for Logic
 - Verilog RTL
 - VCS (Synopsys) logic simulation
 - Design Compiler (Synopsys) logic synthesis
 - Primetime (Synopsys) STA
 - Fire & ice (Simplex) parasitic extraction
 - Silicon Ensemble, CTGEN (Cadence) hierarchical P&R, clk tree synthesis
 - Fast SCAN (Mentor) for DFT, ATPG
 - Voltage Storm (Simplex) for IR drop analysis
 - Calibre (Mentor) DRC, LVS
 - Tuxedo (Verplex) formal verification



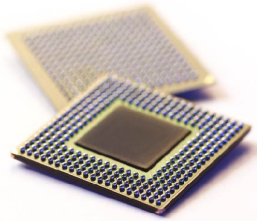
Die Photo

- This is 5 Mbytes file, I'll show during presentation only, too big to email



Die Plot

- This is 8 Mbytes file, I'll show during presentation only, too big to email



Status

- silicon in evaluation now
- metal spin tape out imminent