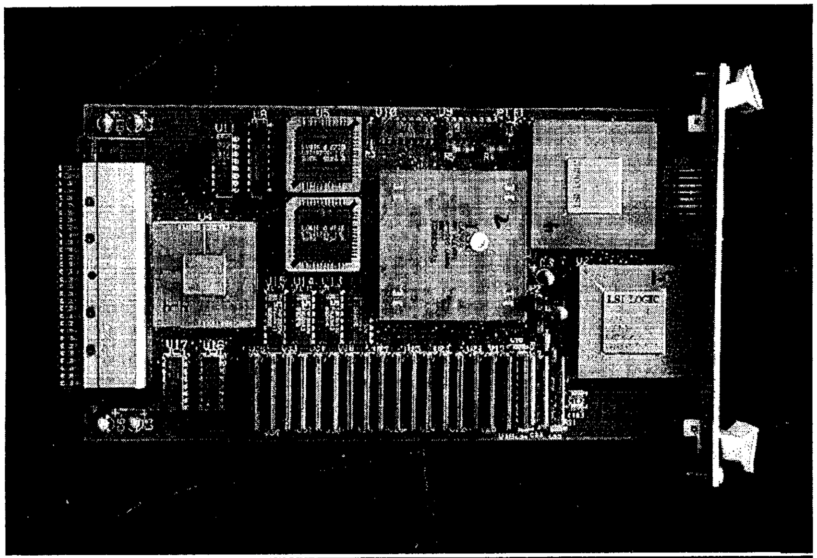
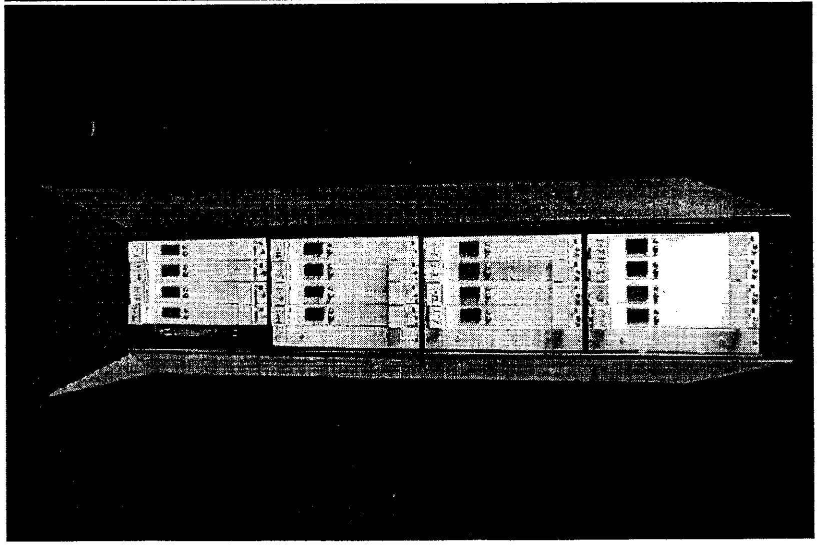
The Alewife-1000 CMMU:

Addressing the Multiprocessor Communications Gap.

It's not a fish

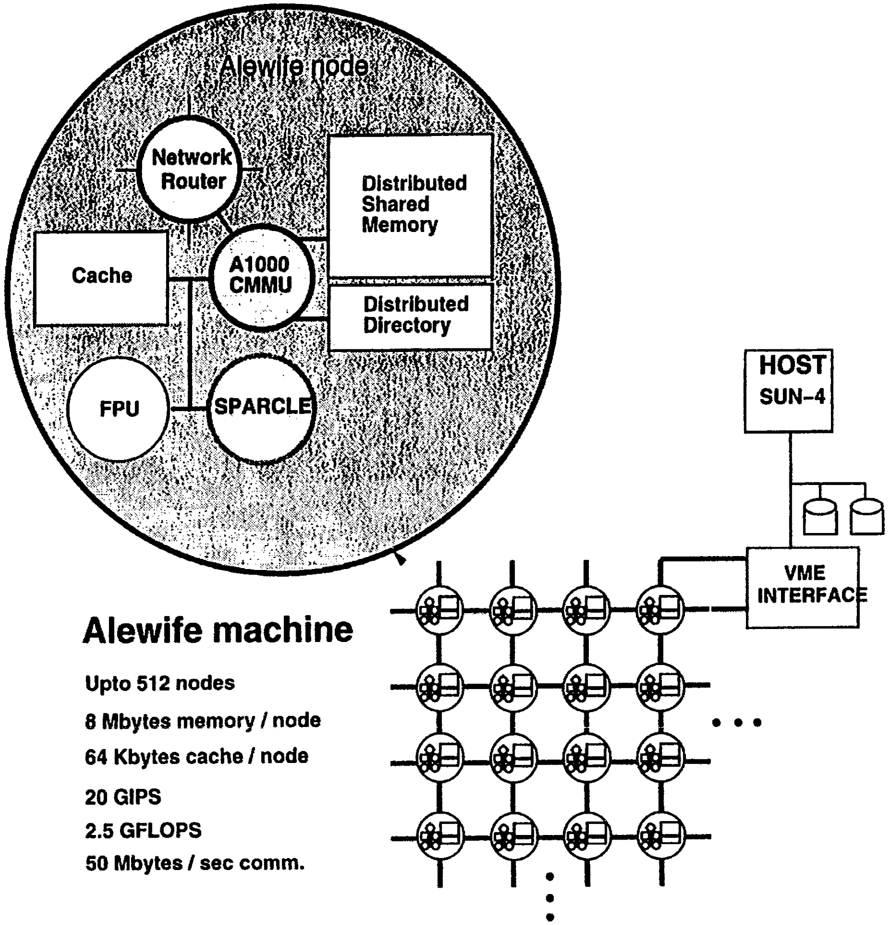
It's a T station





John Kubiatowicz
Massachusetts Institute of Technology
Laboratory for Computer Science
kubitron@lcs.mit.edu

Shared Memory Machine



Alewife machine

Upto 512 nodes

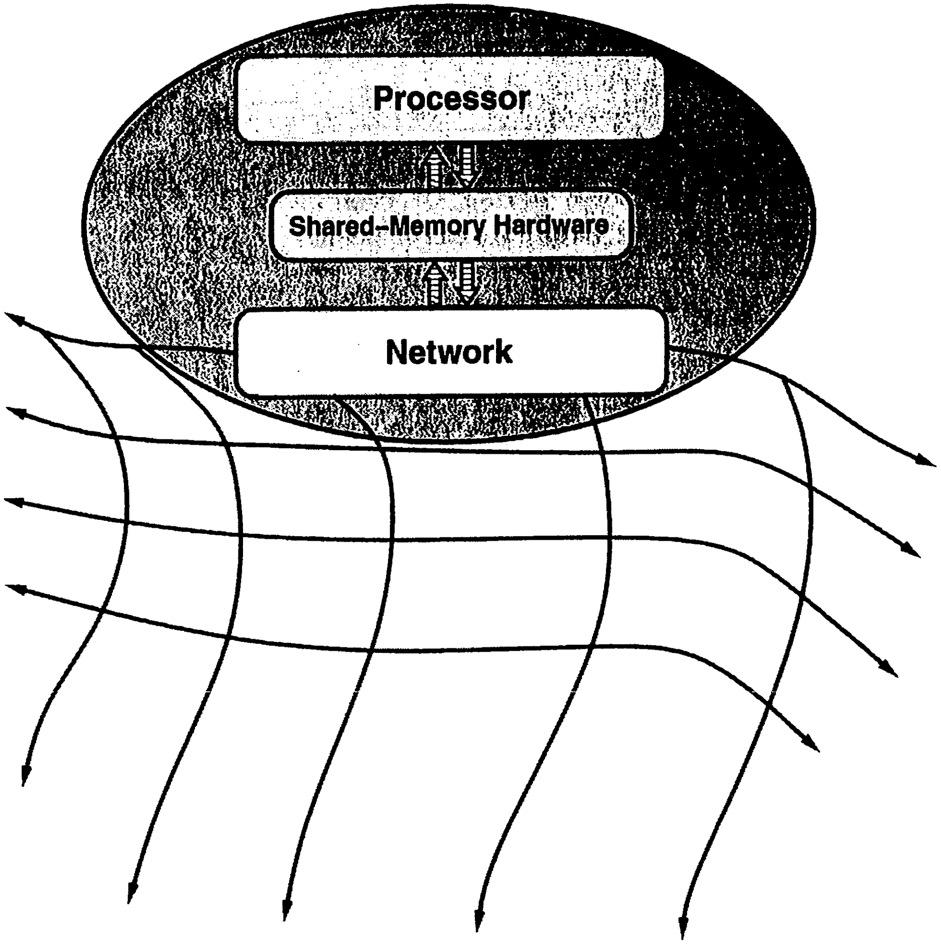
8 Mbytes memory / node

64 Kbytes cache / node

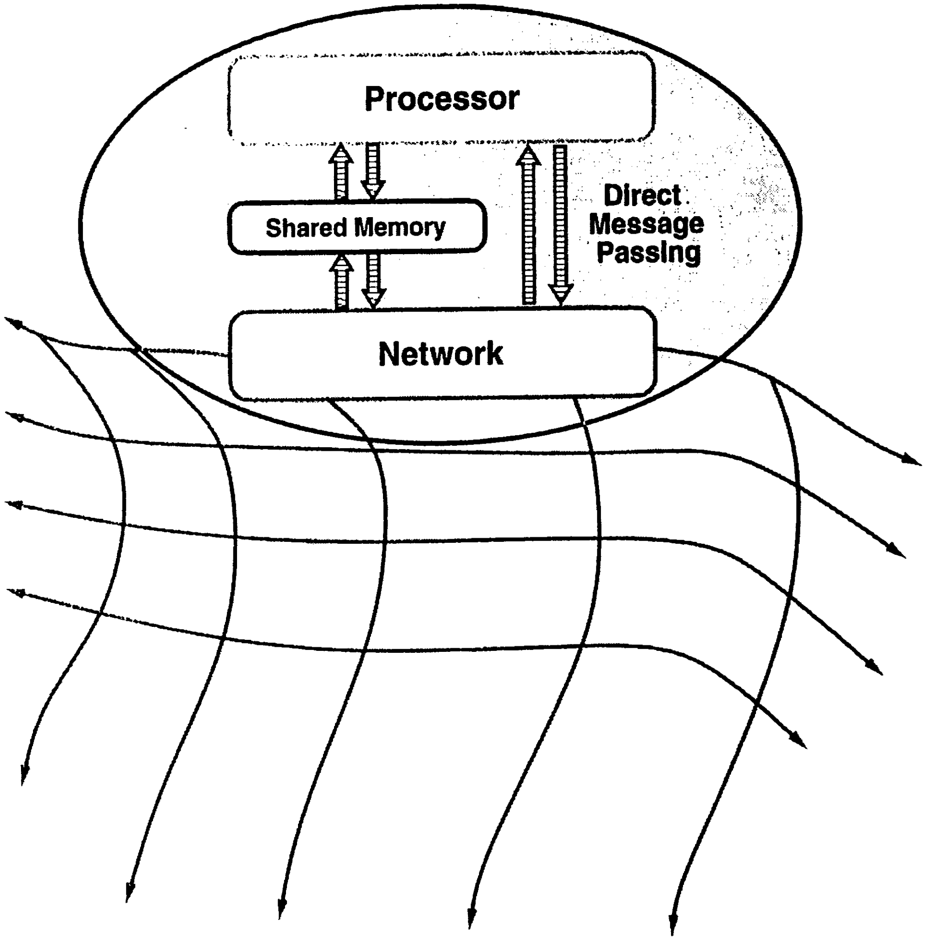
20 GIPS

2.5 GFLOPS

50 Mbytes / sec comm.



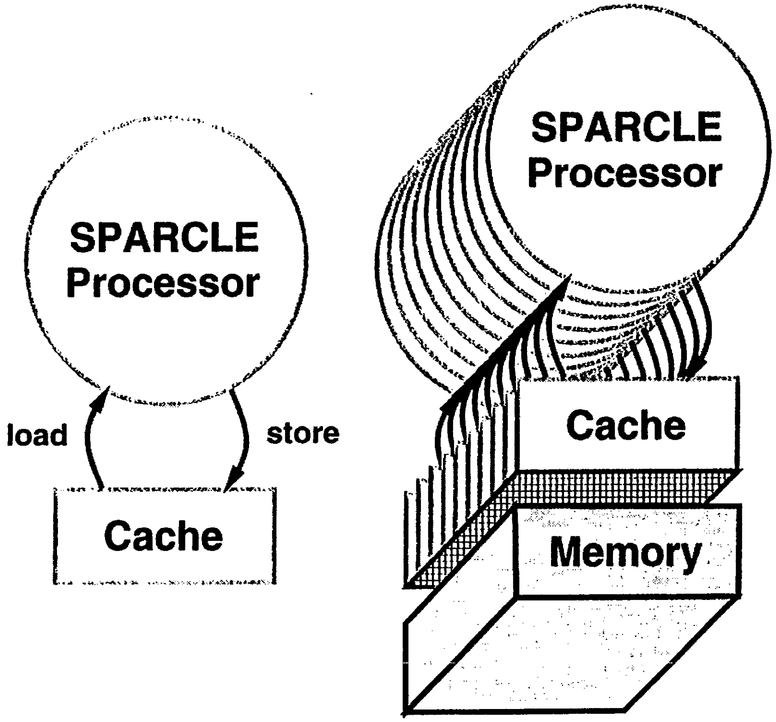
Best of BOTH worlds



Outline

- Motivation
- Mechanisms
- Performance Data
- Implementation
- Status and Conclusions

Shared Memory Interface



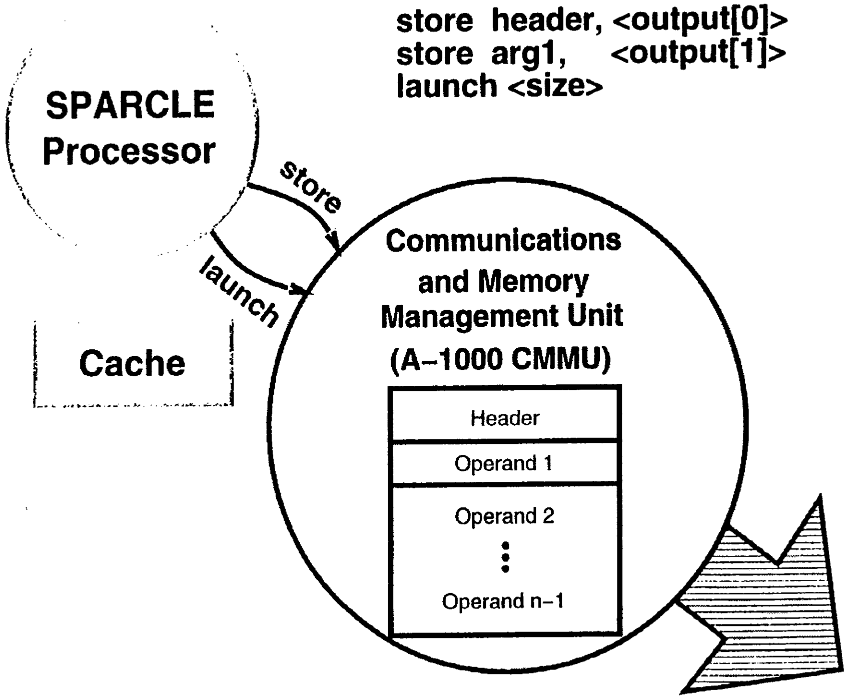
Single Coherent View Of Memory

Properties of the Shared Memory

- Data is physically distributed.
- Each node has 64 Kbytes of hardware-managed cache for shared and private data.
- Rapid context-switching and software prefetch permit latency tollerance.
- Cached data is kept *coherent* through a combination of hardware and software techniques.

Message Output Interface

```
store  header, <output[0]>
store  arg1,    <output[1]>
launch <size>
```



Launch instruction takes one cycle and is *atomic*.

Properties of the Message-Passing Interface

- Direct, user-level access to message interface.
- Efficient generation/consumption of short *and* long packets.
- Integral DMA for block transmission.
- Support for "typical" multi-packet I/O traffic.
- Deadlock-free control of the network.

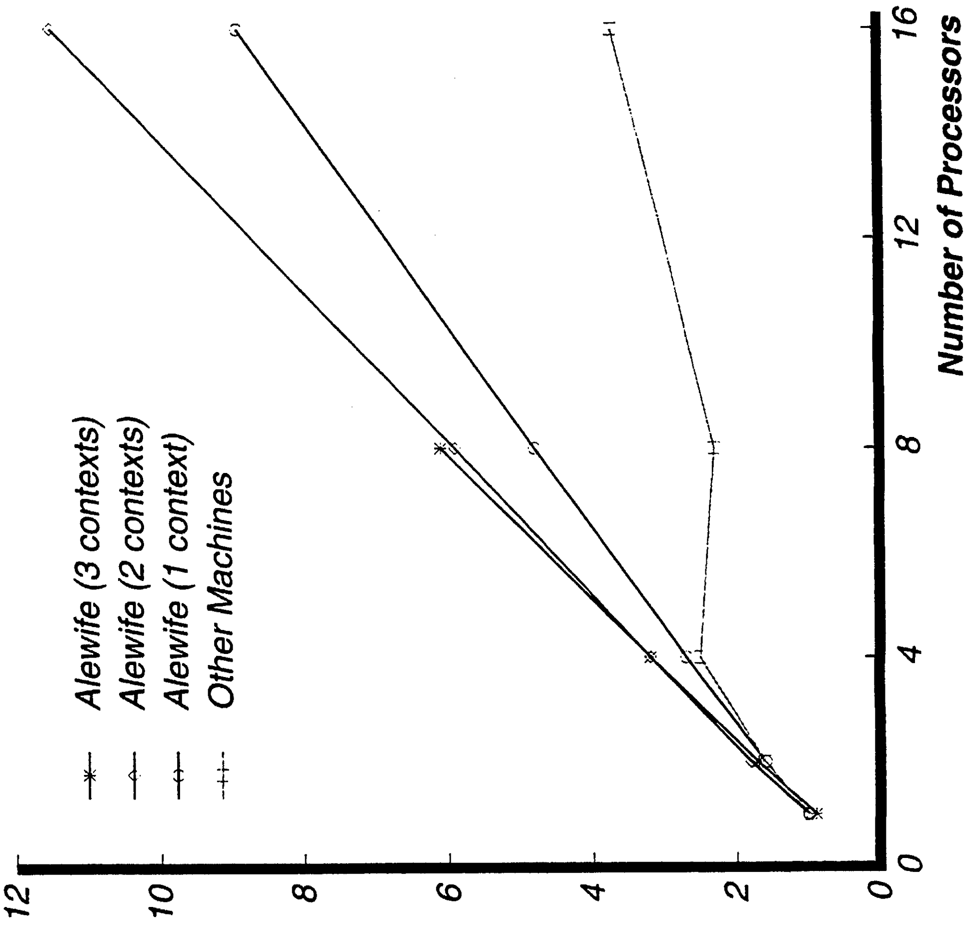
Cycle Counts

Action	Processor Cycles
Load Instruction:	2
Store Instruction:	3
Private cache-miss penalty:	9
Shared Local cache-miss penalty:	11
Context-switch time (data request):	14
Directory-read (cached):	5
Directory-write (cached):	6
Message-send (2 words at user-level):	8 (processor time) 11 (to network)
Message-receive (null active message at user-level):	35
Fast Task Dispatch:	88 + 1.1 × distance

Is fast messaging expensive? No!

Modern RISC pipelines can be extended with *interfaces* for seamless integration with the network and memory system.

- Multiple "flavors" of uncached load and store instructions.
- A pipeline extension mechanism.
- A spare register set.
- Fast, user-level vectored interrupts.



Remote Read Latecies at 30Mhz (Actual Measurements)

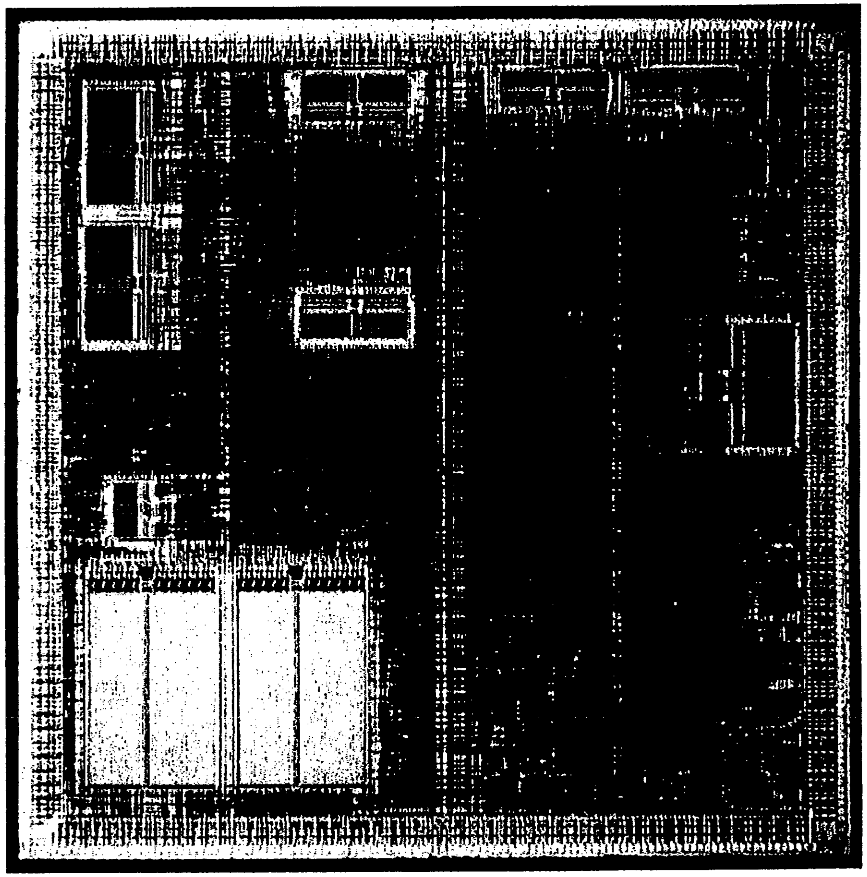
Data status	Total Latency in Processor Cycles	CMMU cost
Clean	46.6 + 2.3 × distance	16.4 cycles
Dirty in home	51.5 + 2.5 × distance	23.2 cycles
Dirty in third	78.0 + 2.3 × distance	30.5 cycles

Implementation of CMMU

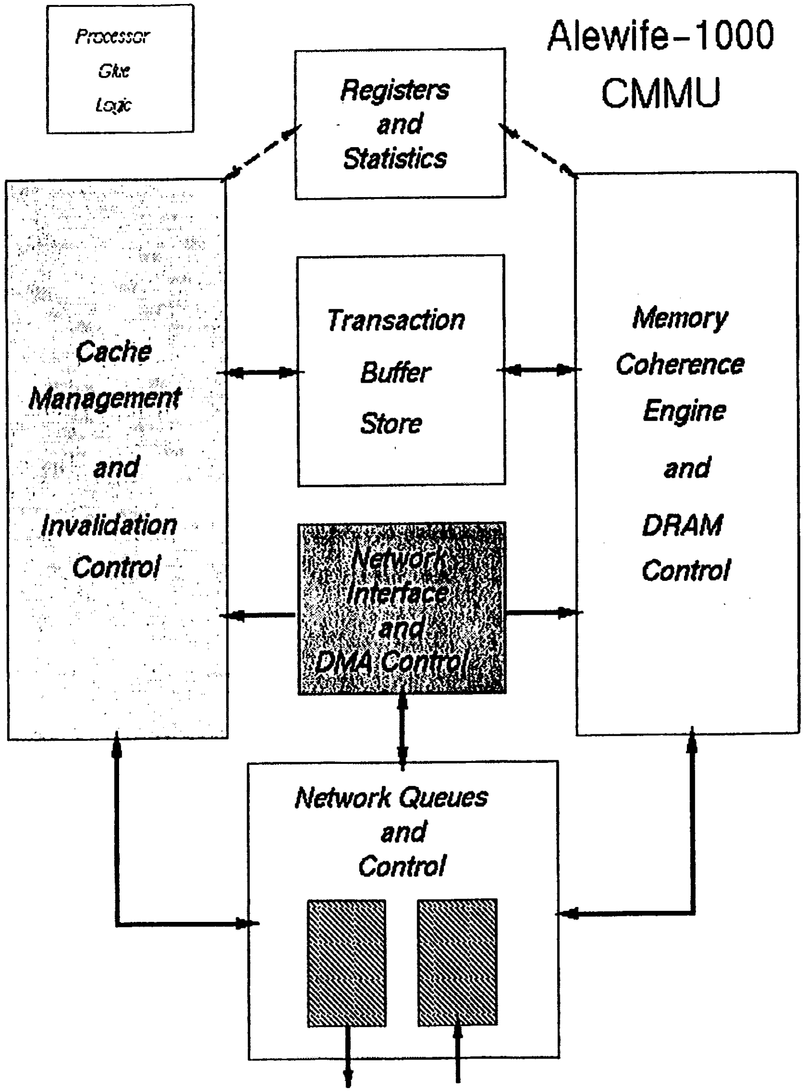
The A-1000 CMMU is implemented in the LEA300K hybrid gate-array process from LSI Logic.

- 95,000 gates and 100Kbits of SRAM.
- Memories were produced with LSI compiler.
- Core logic was synthesized from LES (by LSI Logic) and optimized with Berkeley SIS.
- Critical circuits were designed directly with schematic editor.
- Hybrid simulation for verification.

A-1000 Die photo



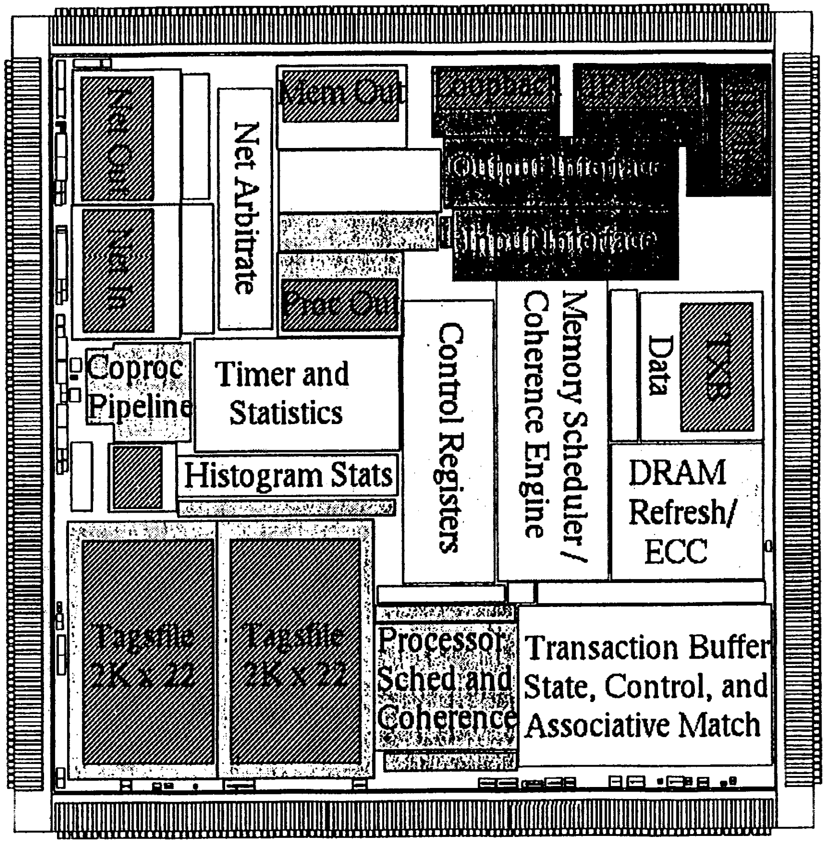
- Die size: 15mm × 15mm
- 95K random gates.
- 100K bits of SRAM (regular structures).



Alewife-1000 CMMU

Processor Glue Logic
Registers and Statistics
Cache Management and Invalidation Control
Transaction Buffer Store
Memory Coherence Engine and DRAM Control
Network Interface and DMA Control
Network Queues and Control

Alewife-1000 CMMU



○ **Processor and Cache Control**

○ **Transaction Buffer**

○ **Asynchronous Network**

○ **Memory and DRAM Control**

○ **Registers and Statistics**

● **IPI Message Interface**

Sizes in Gates:

Cache Control:	12688	13.7%
Coprocessor Pipeline:	2179	2.4%
Memory:	16085	17.4%
Dram Control:	7385	8.0%
Transaction Buffer (state):	14640	15.8%
Transaction Buffer (data):	1650	1.8%
Livelock Removal:	1886	2.0%
Message Interface (Input):	5527	6.0%
Message Interface (Output):	4804	5.2%
Network:	5647	6.1%
Statistics:	12464	13.5%
Other Registers:	7560	8.2%
Total:	92515	

Status of the Alewife Machine

[To be updated much closer to final deadline]

First run of silicon for A-1000 CMMU:

- 16-node machine operational since June 17.
- Runtime system and compilation environment supports C and Mul-T (a dialect of LISP).
- A number of large kernels and benchmarks have been run.
- Small number of bugs in first run of silicon.

Conclusion

- Efficient communication mechanisms are important in a multiprocessor.

- The Alewife-1000 CMMU integrates *both* message-passing and cache-coherent shared memory in a single hardware framework.

- Uniprocessor pipeline designers can provide simple "hooks" for efficient multiprocessor interfacing.