file

cc: Holloway, Rettberg, Barker
orig: me

To: Steve Levy
Dan Sullivan
Dave Walden
Alex McKenzie
From: Steve Blumenthal

A copy of slides we presented to DARPA and NSA

MONARCH

BBN Systems and Technologies

Steve Blumenthal
Phil Carvey
Gregg Bromley

January 23, 1990

Monarch

Agenda

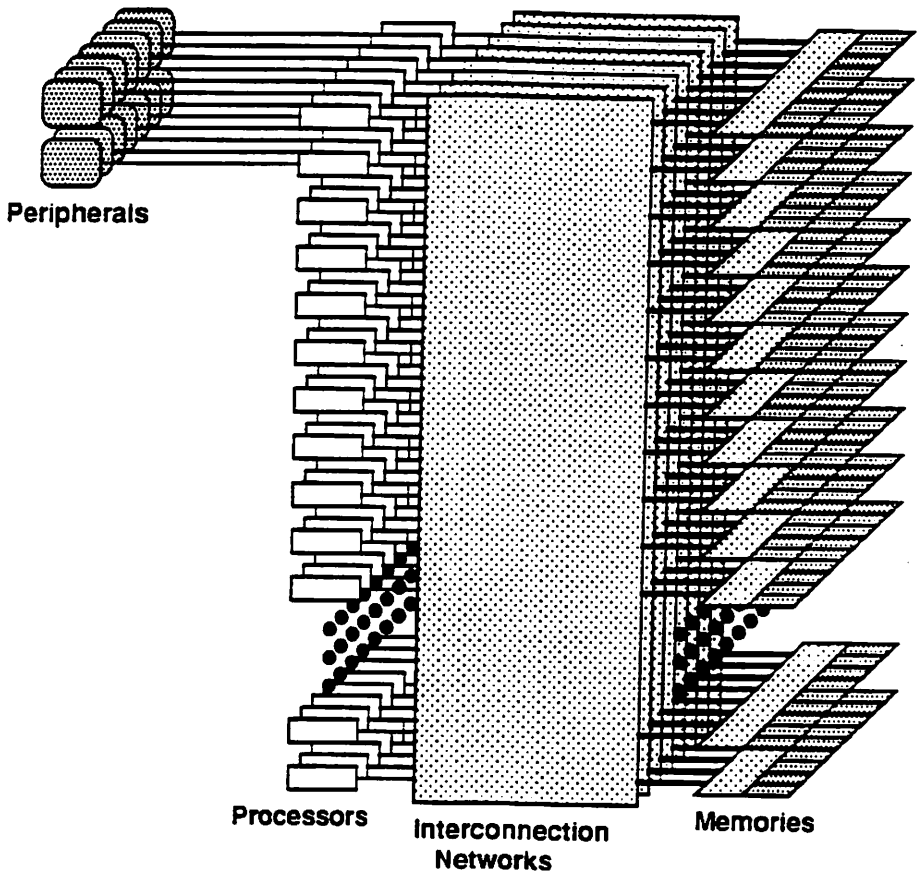
• **Monarch Introduction**	Blumenthal
• **Monarch Hardware**	Carvey
• **High Density Packaging**	
• **Programming Model and Software**	Bromley
• **Program Plan and Costs**	Blumenthal

Monarch

Overview

- **Uniform Access Shared Memory Machine**
- **Single Program/Multiple Thread**
  Fine Grain Parallel MIMD Programming Model
- **Scalable to Teraflop Machine**
- **Low Latency Interconnection Scheme**

Monarch System Design



The Switch is the Key

- Minimize Effective Latency
- Provides Very High Processor-Memory Bandwidth
- Economically Scalable System

Monarch

Processors

- **Commercial Processors can Meet Multiprocessor Requirements**
  - Cost/Performance
  - Lower Development Risk
  - Evolutionary Family Tracks Technology
  - Community of Uniprocessor Users and Software
- **with Latency Masking Techniques**
  - Parallel Instruction Execution & Memory Access
  - Operand Prefetch
  - Multiple Operand Fetch

Monarch

Scalability

Procs	GIPS	GFLOPS	Mbytes
65K	6,500	6,500	17,180
8K	800	800	2,150
1024	100	100	270
128	13	13	67

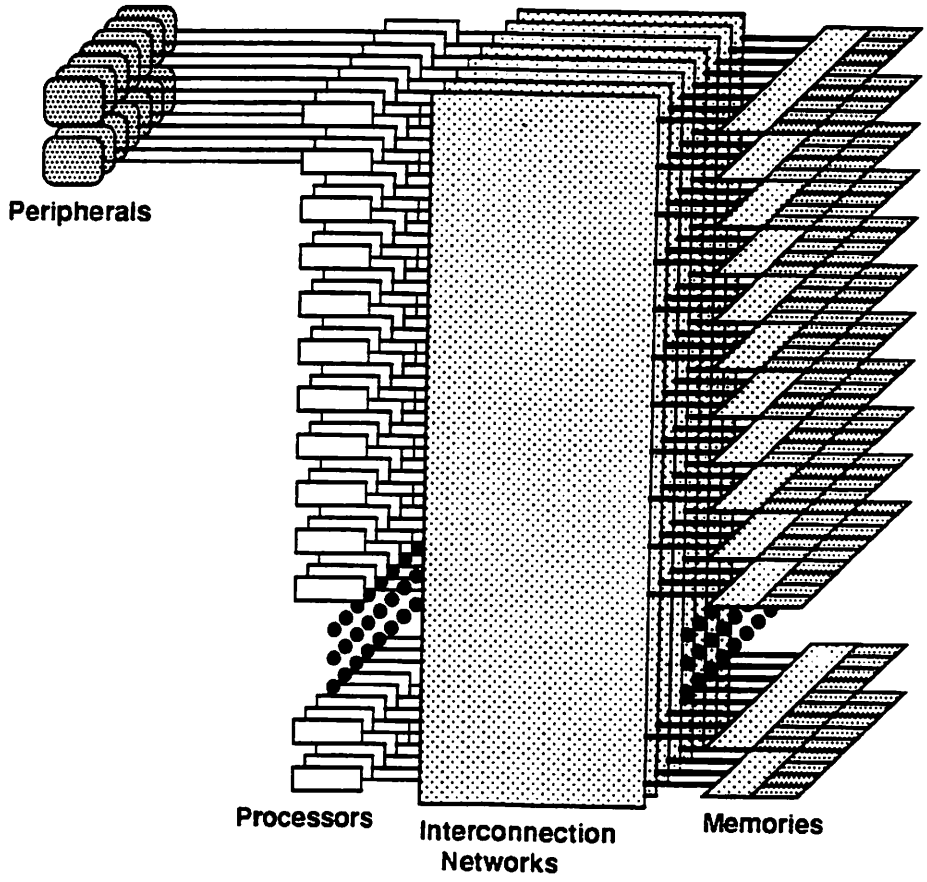
- **Cost Effective**
  - Target raw parts cost Cost $2000 per Processor
  - Target raw parts cost $20 per MIP
  - Target raw parts cost $20 per MFLOP

Monarch

Recent Developments

- **BBN ACI introduced the TC/2000 multiprocessor**
  - Software - Significant Development Investment
    - MACH Operating System
    - Programming Languages
    - Parallel Runtime & Synchronization Libraries
    - Debuggers and Tools
- **BBN STC has continued to investigate highly parallel VLSI multiprocessor systems**
    - Updated Switch Design
    - Use of Commercial Microprocessor
    - Latency Masking Research
    - New High Density Packaging Concepts
- **BBN STC IR & D study to adapt switch technology to multigigabit networks**

Monarch System Design



* Memory physically partitoned into equal elements accessed via spatial switching

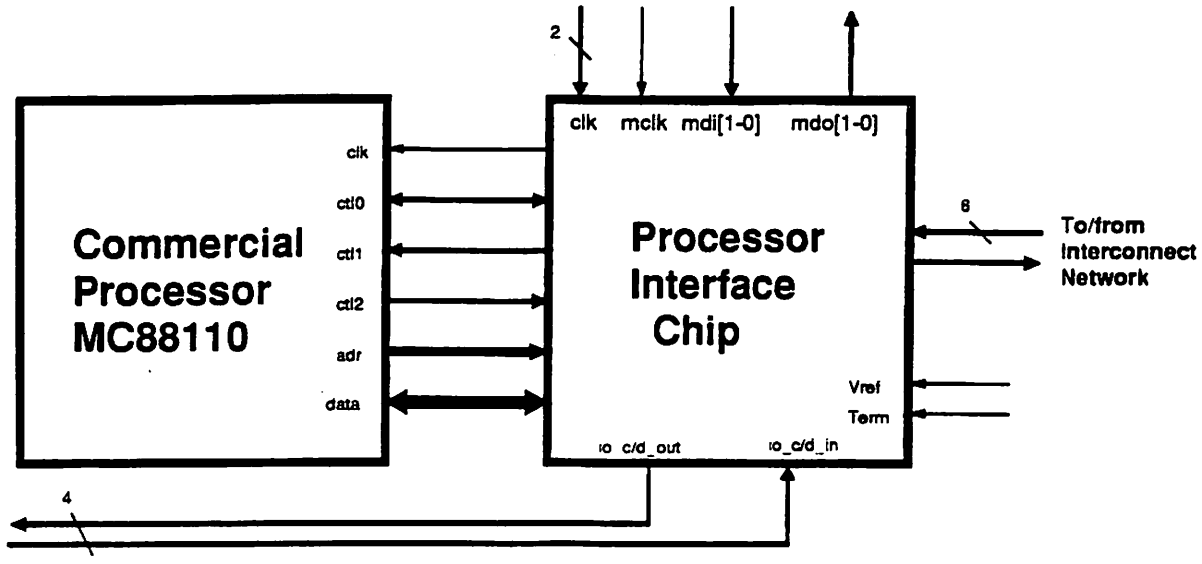
* When memory elements have sufficient bandwidth, two or more processor - interconnect sets can be supported per memory elements via time multiplexing

Monarch Characteristics

* Shared memory machine, all memory is equally accessible to all processors with constant access time

* System consists of:
  - Processor modules = Motorola MC88110 plus custom coprocessor interface to interconnection network
  - Interconnection network = multiple stages built from just two custom switch elements
  - Memory modules = two identical custom network interface chips and 34 SRAM chips
  - I/O subsystems, VME bus compatible
  - Utility modules

* Interconnection network supports
  - very low latency transactions
  - pipelined operations for high wire utilization
  - combining reads and writes
  - large plurality of paths from any processor to a memory module, paths selected via random number generator
  - M>N path redundancy for fault tolerance
  - continuous link monitoring
  - link testability during machine operation
  - process, temperature, & link delay compensation of all links during machine operation
  - extensible to very high bandwidths
  - memory module to processor communication path during each frame for very fast processor to processor interrupts
  - very low blocking probability



* Management of instruction fetches & operand loads and stores

* Small cache of recently fetched operand strips

* Address hashing

* Normal & low priority transactions

* Steal, write if not stolen, read & write combining

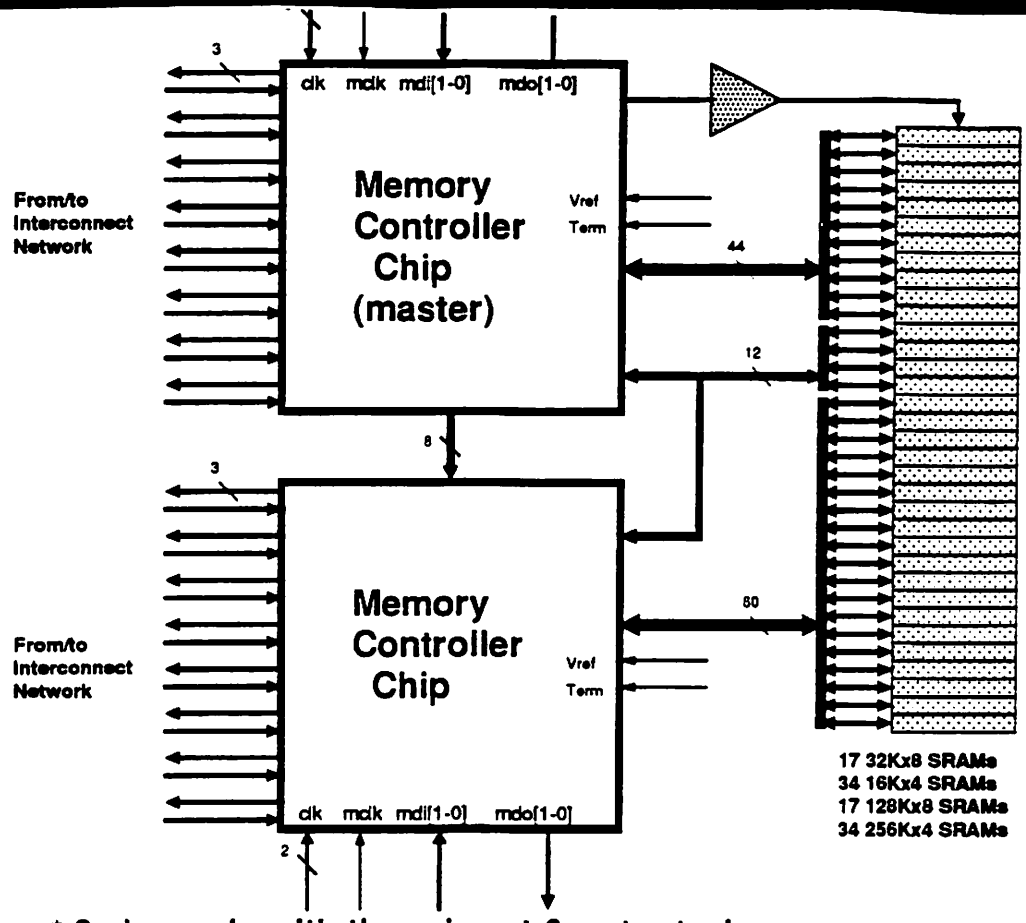
* Real time clock interrupts

* Processor to processor transactions

* Configuration dependent request formatting

* Random path assignment

* Reset, interrupt vector, bootstrap code



* 8 channels with three input & output wires

* MC supports two modes: master and slave

* Three reads & five writes per frame imply blocking probability under 6%

* Small number of write queues per channel

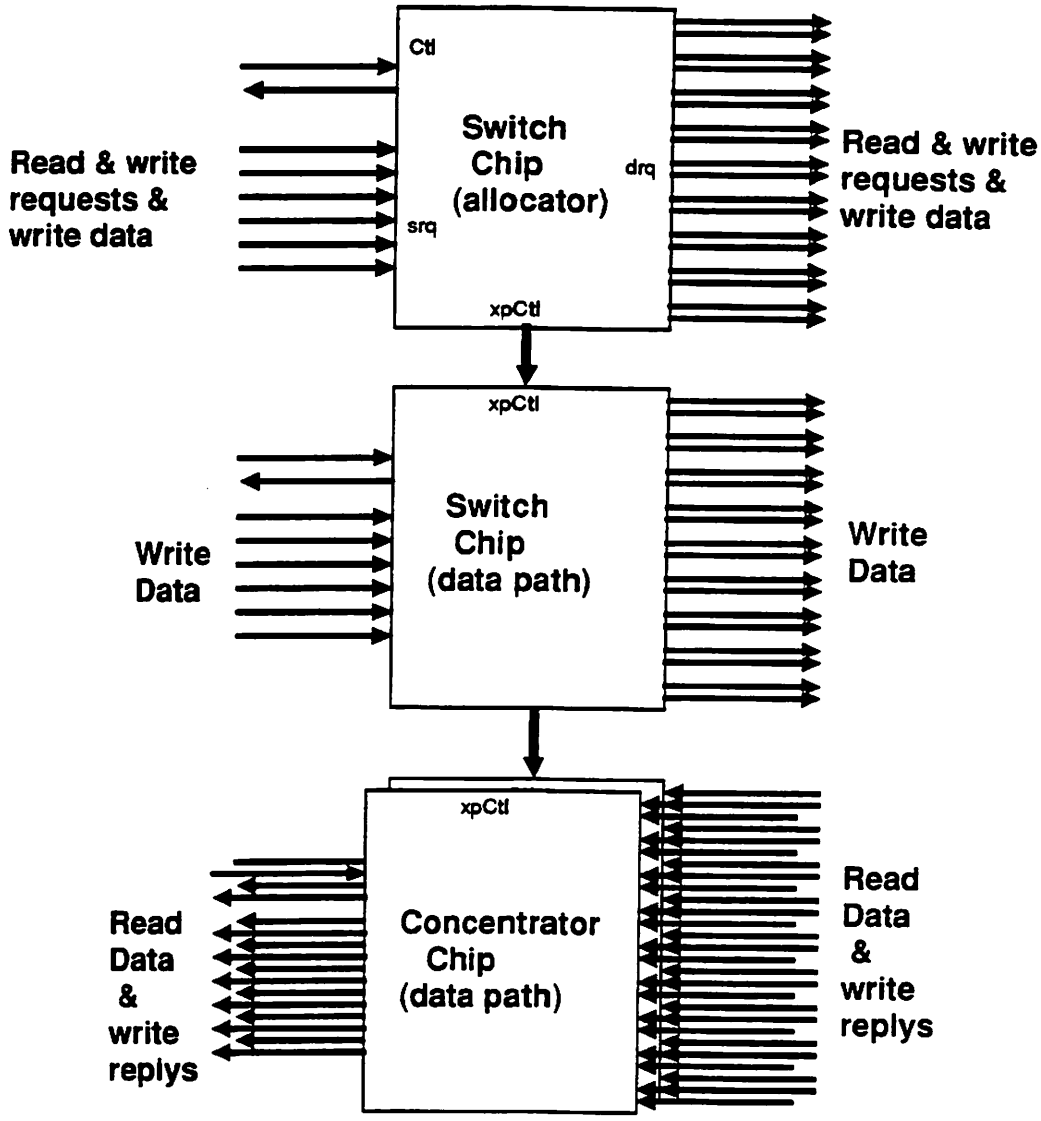
* In address mode, MC captures request addresses, arbitrates for three SRAM access slots, controls the SRAM, & broadcasts control info to companion chip

* In slave mode, MC inputs control info & transfers data from SRAM buffers to switch channel

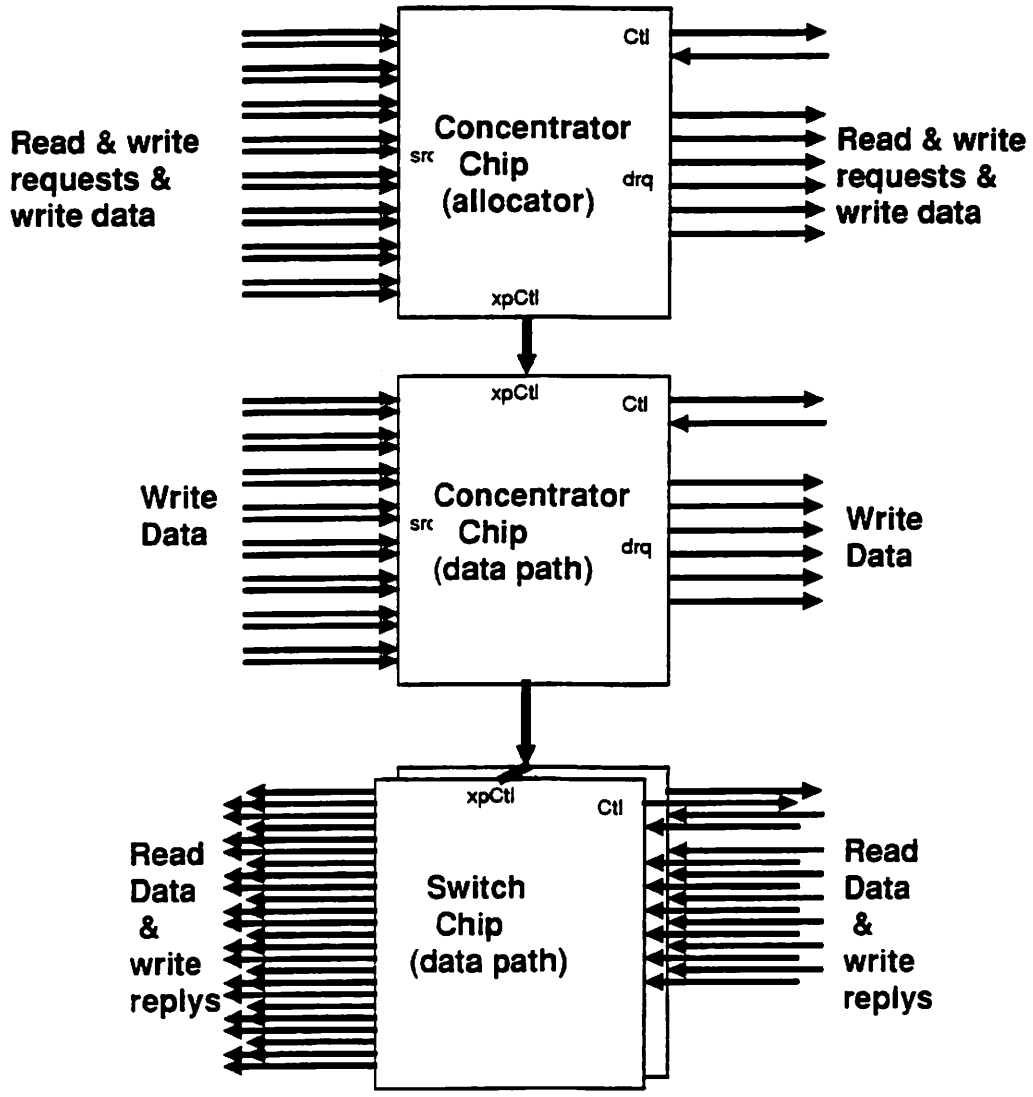
* MC supports read & write combining, write if not stolen, and steal operations

* Double error detection, single error correction

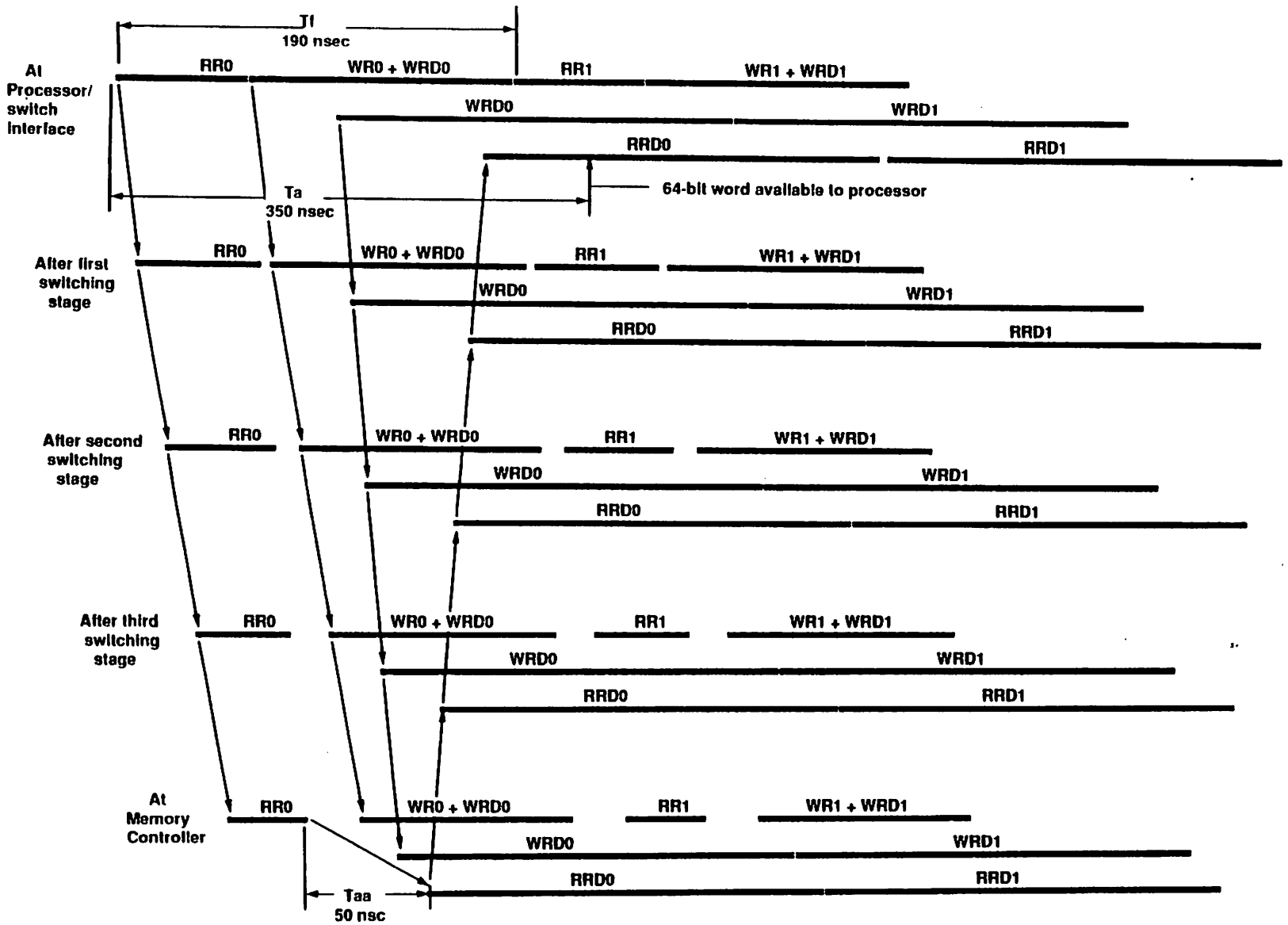
* Mailbox in each MC transmitted to corresponding processor each frame



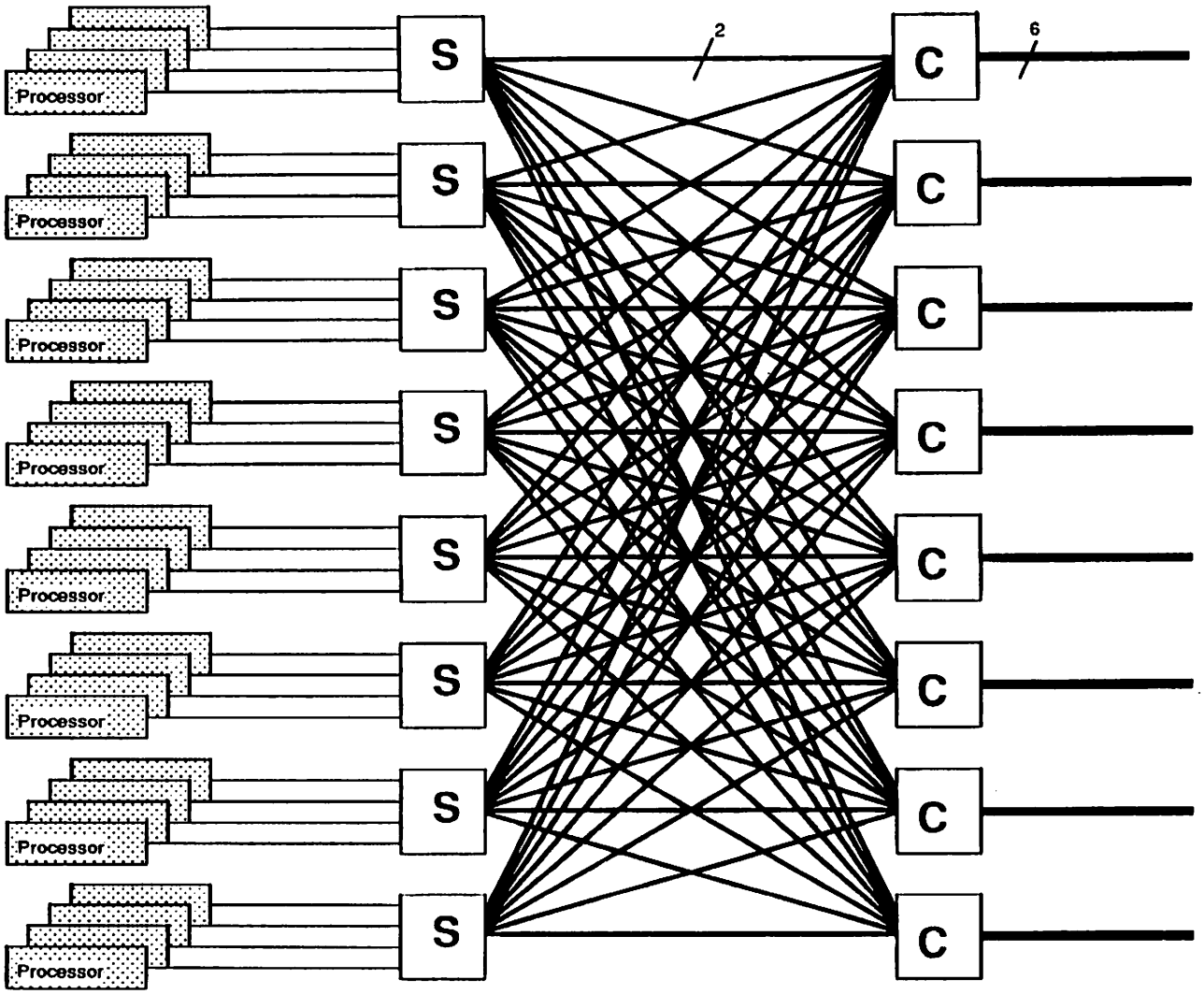
* 6 inputs, 8 pair of outputs, 8-way switching
* Redundant pair of outputs
* 12 wires per channel
* 256 bit reads and 256 bit writes
* Net aggregate bandwidth 337 Mbytes/sec
* Latency <20 nsec
* Frame period 190 nsec



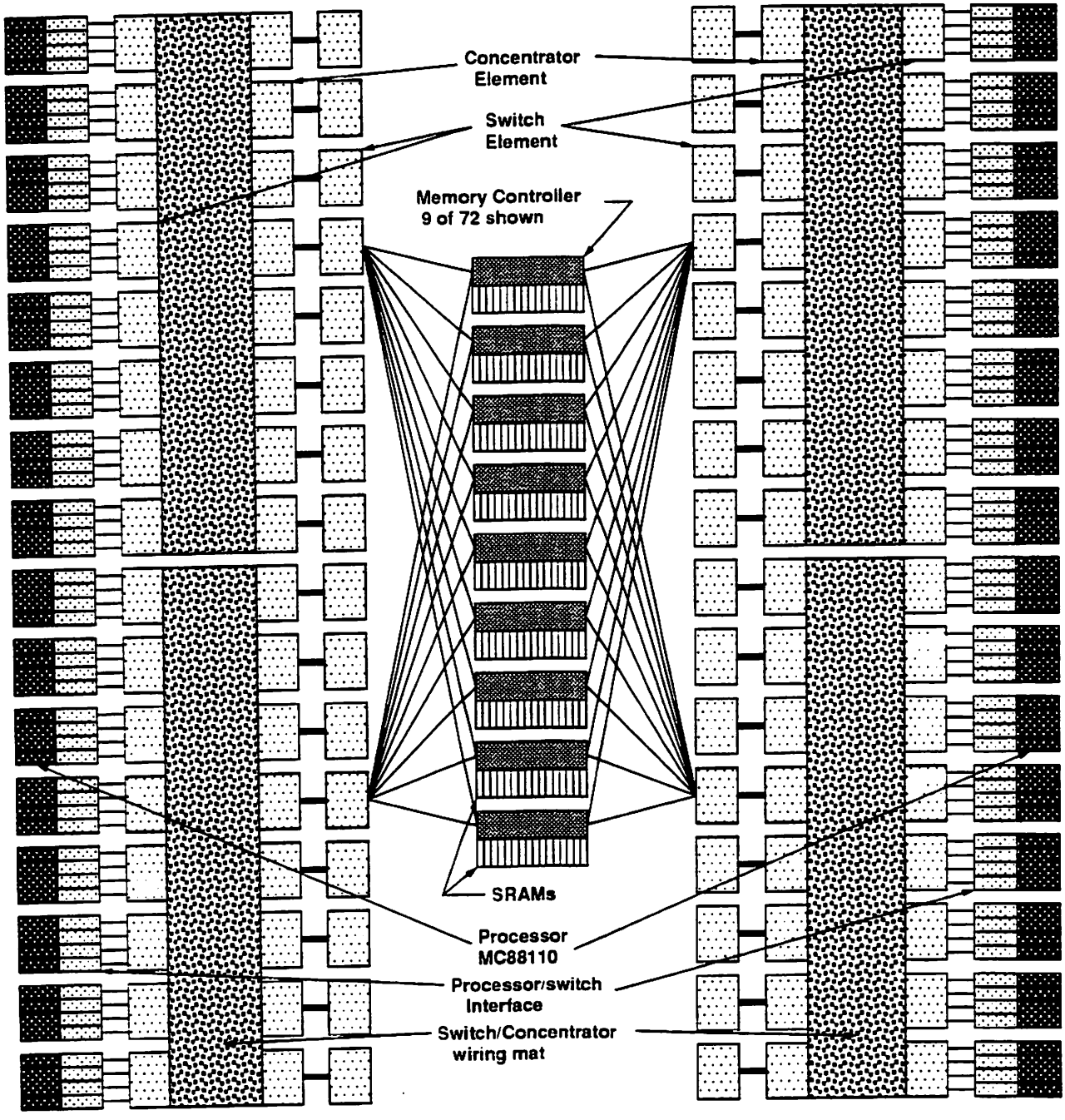
* 16 inputs, 6 outputs, one way switching
* 12 wires per channel
* 256 bit reads and 256 bit writes
* Net aggregate bandwidth 337 Mbytes/sec
* Latency < 20 nsec
* Frame period 190 nsec



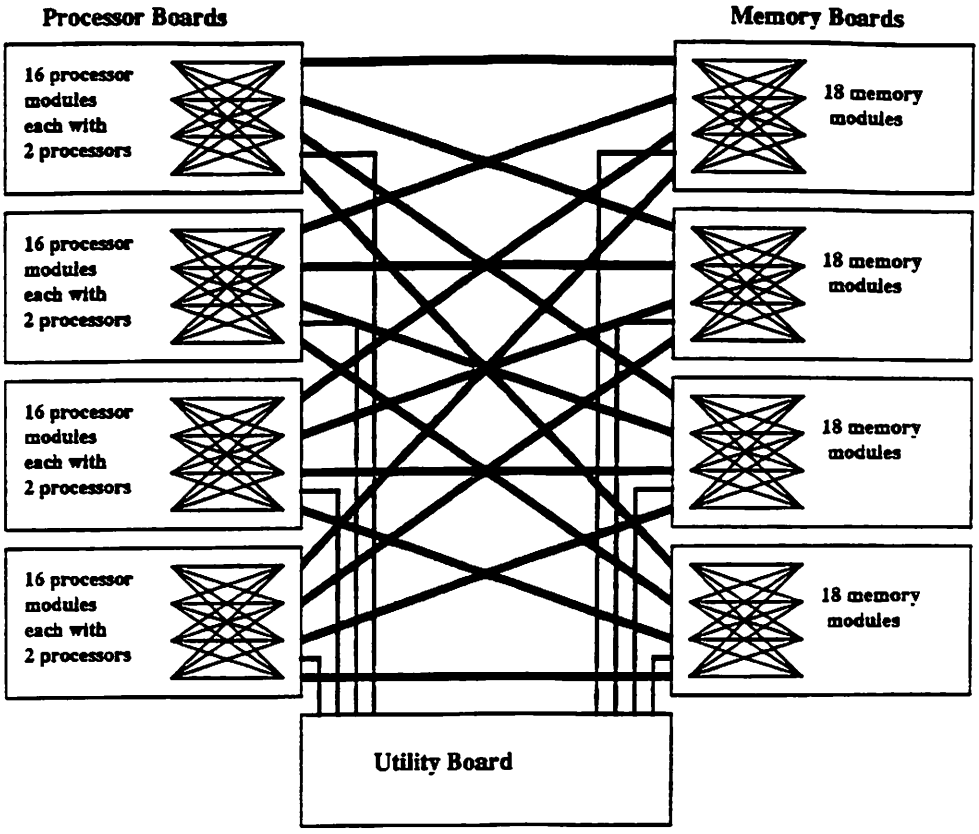
Request and reply timing for two consecutive reads and writes



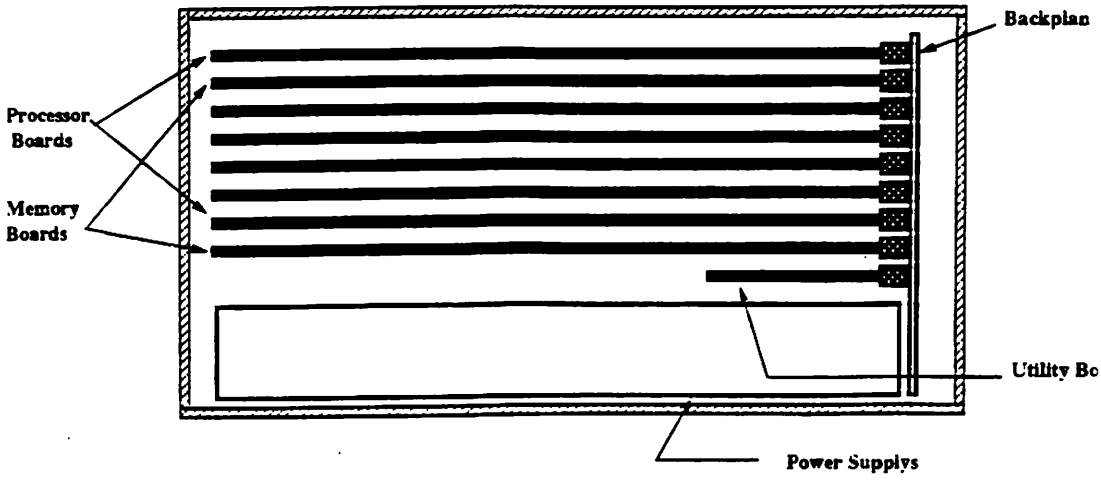
Monarch Processors and first stage switching



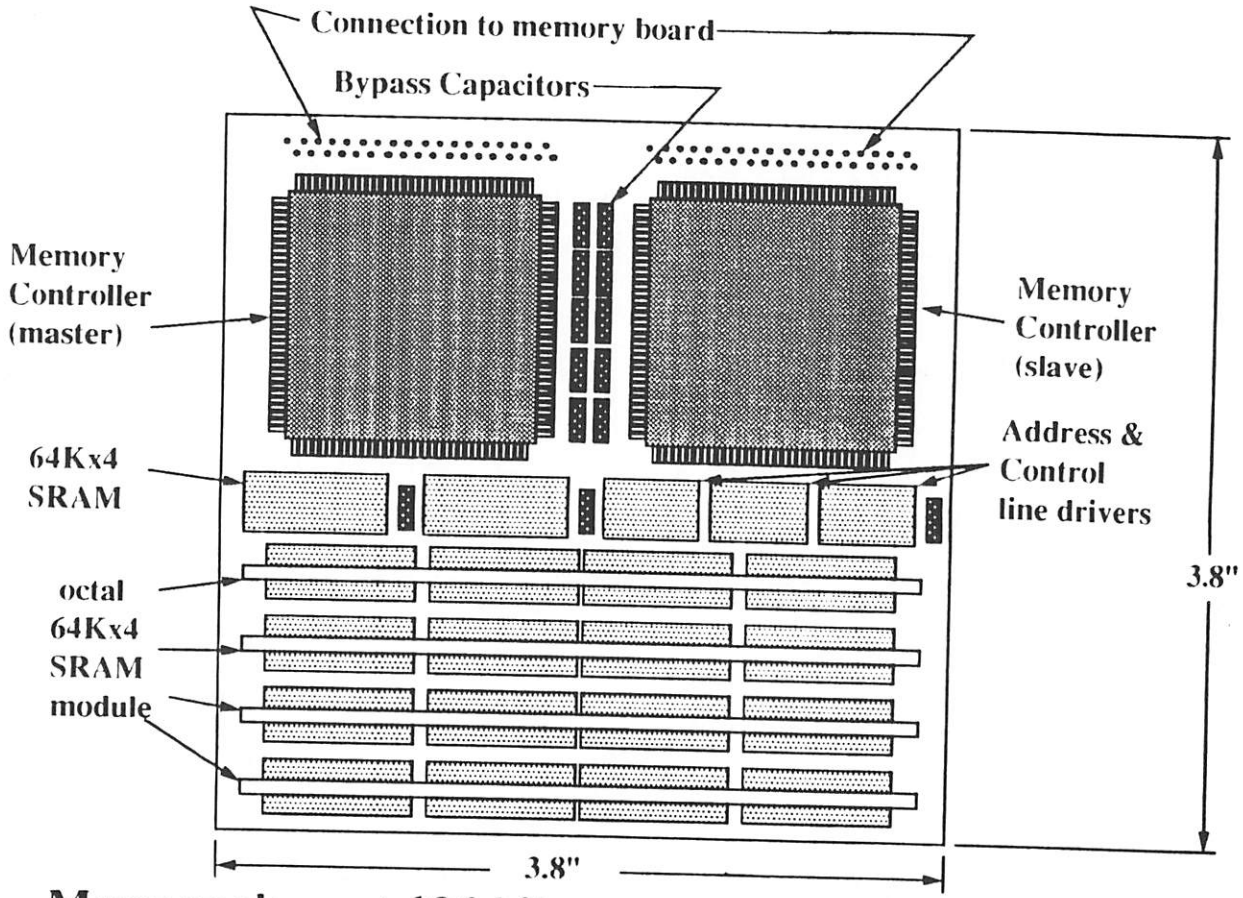
Monarch 128 Processor Block Diagram



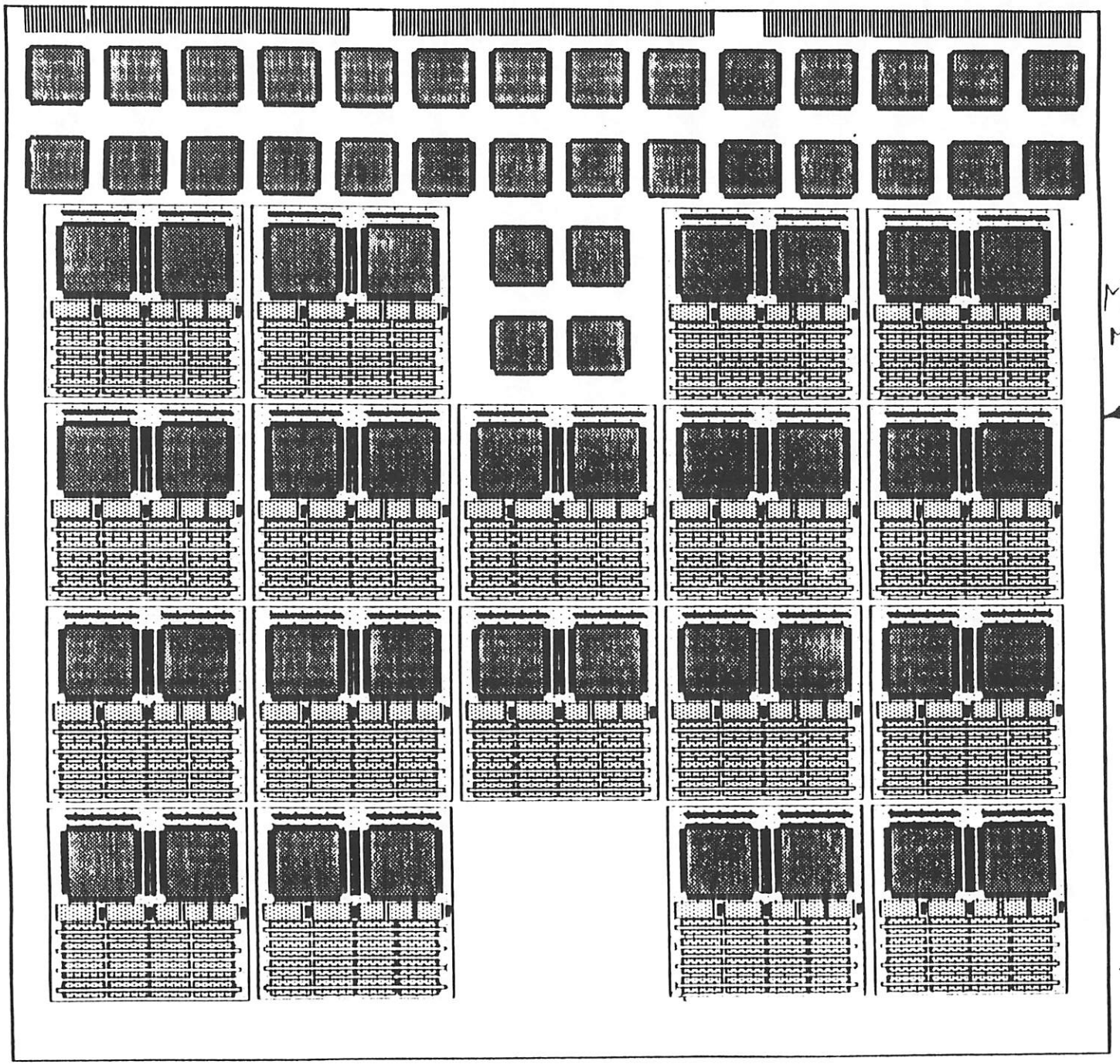
128 Processor Monarch as a set of interconnected boards

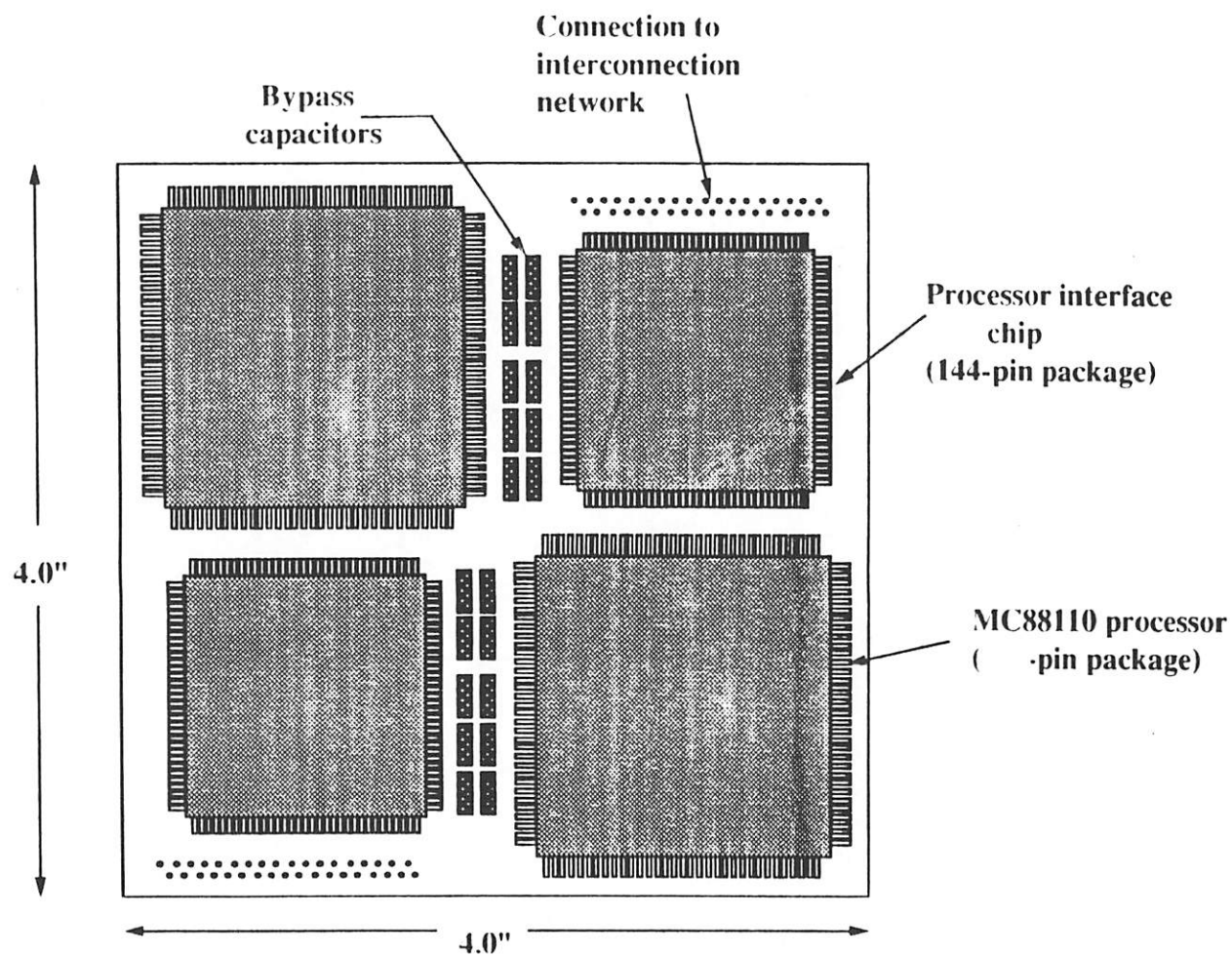


128 Processor Monarch card cage



Memory element 136-bit accesses, 28.6-nsec cycle


MONARCH
MEMORY
BOARD



Processor Module consisting of two MC88110s and two processor interface chips

High Density Monarch

* Standard PC board and backplane packaging technology for smaller machines, high density packaging for large machines

* High density packaging desired because of mechanical limitations, backplane-cable-connector constraints

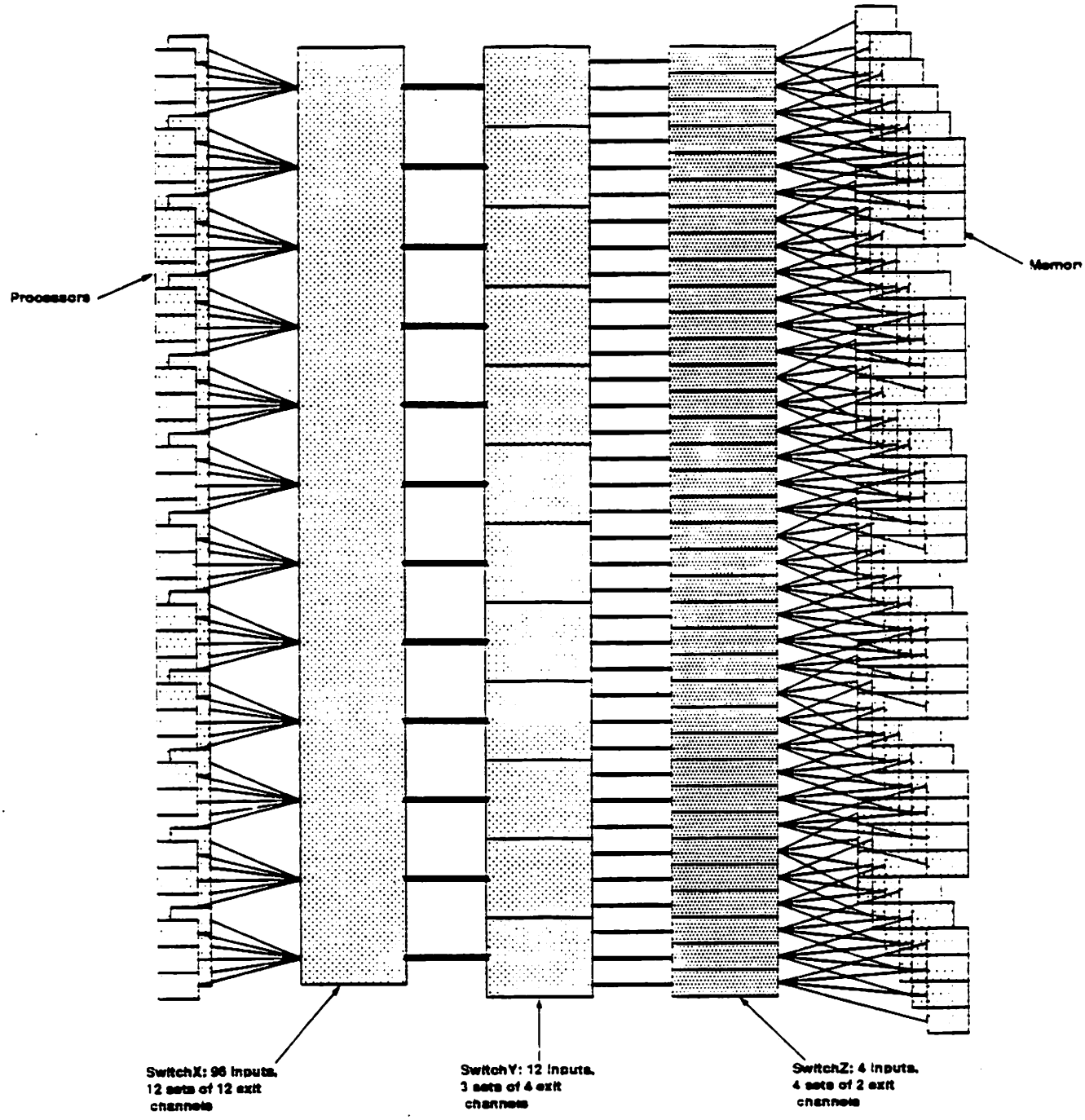
* Packaging very difficult problem for large machines

* Requirements:
  - small size to reduce speed of light latency and skin effect losses in transmission lines
  - low temperature rise to decrease failure rates
  - serviceability without major machine disassembly
  - control of intermodule signal moise coupling
  - fault tolerance of power distribution
  - fault detection without invasive probing
  - small number of different mechanical assemblies

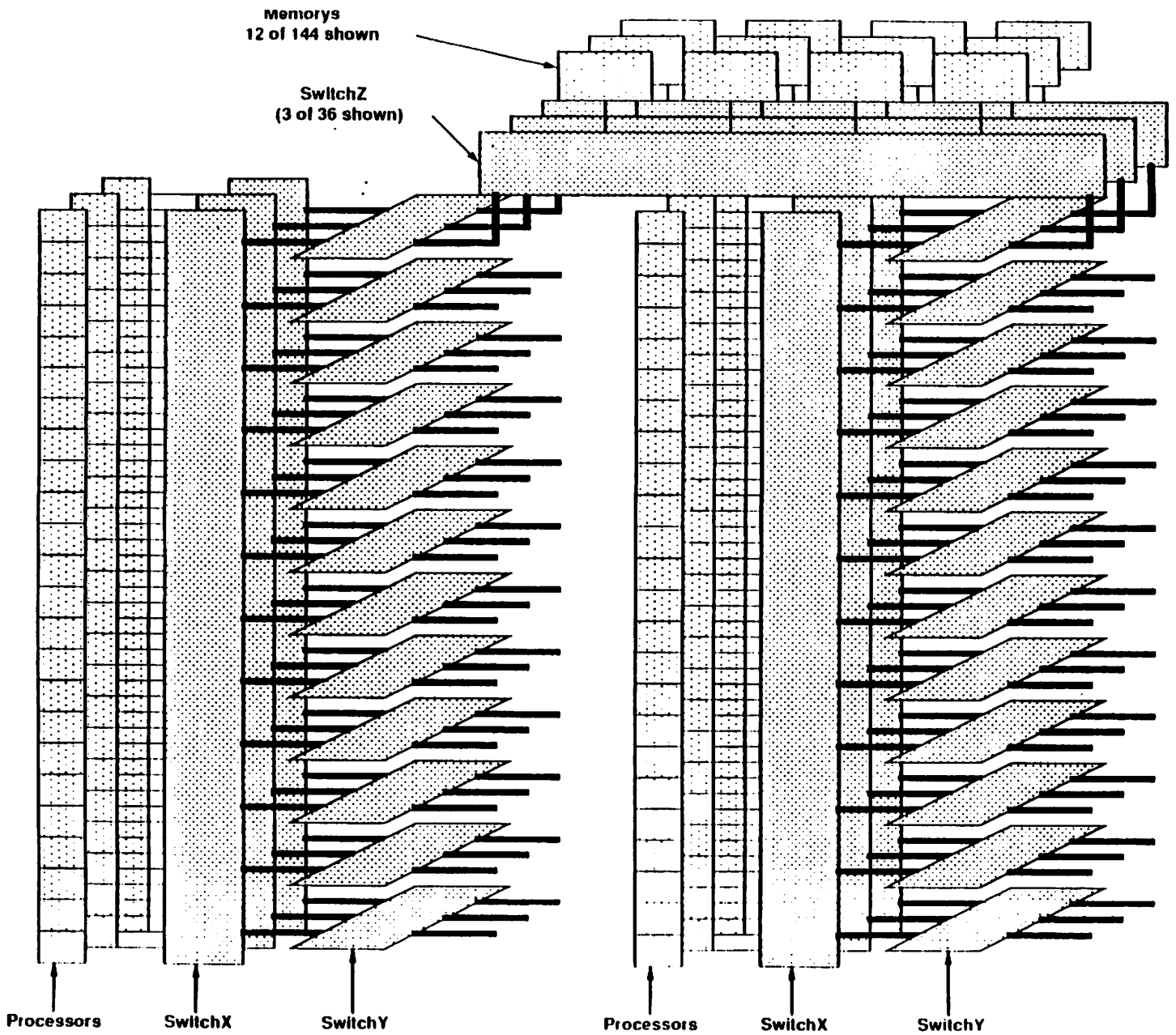
* Ideal machine would use no cables interconnecting assemblies - optical interconnect

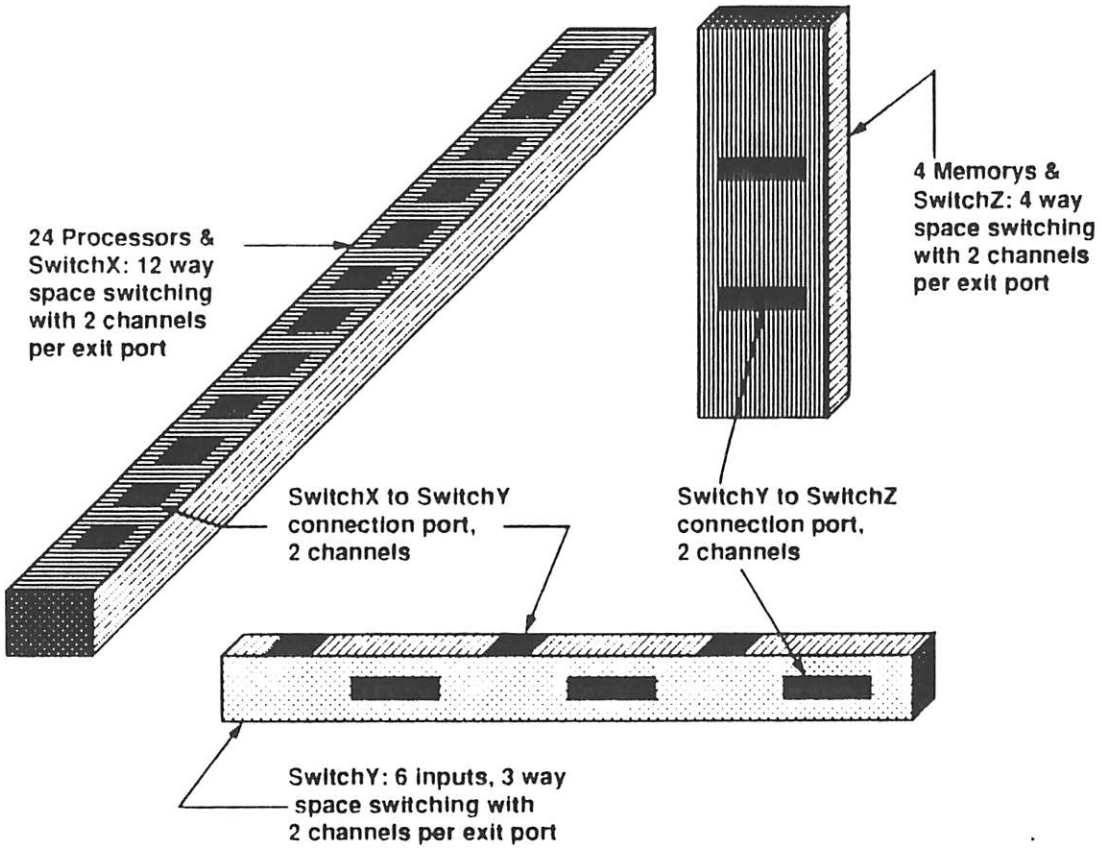
* Liquid cooling required above 2-3 KW per cubic foot, ideal machine should not use immersion cooling

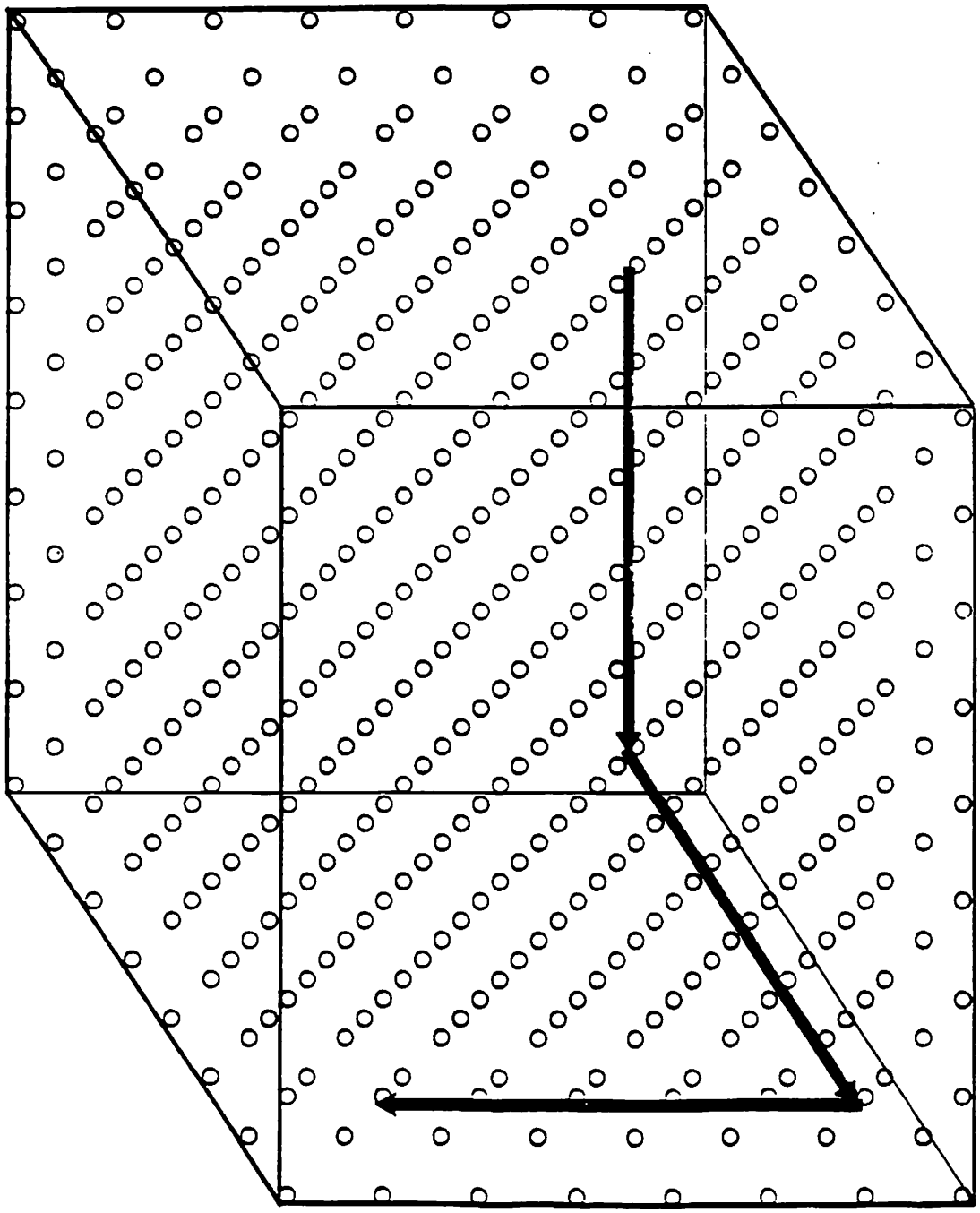
* Ideal machine would be volumetric homogeneous over some minimum sized volume

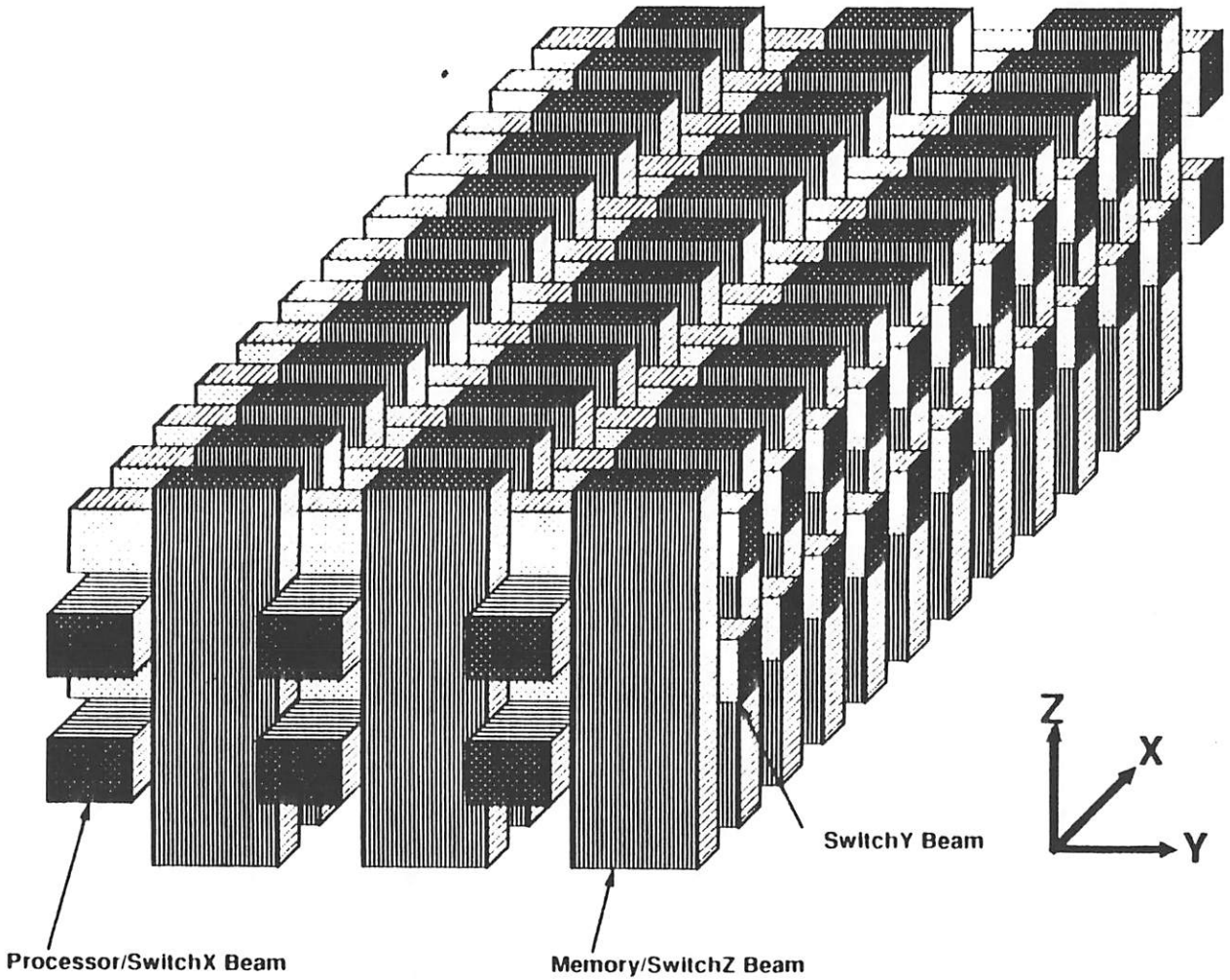


96 Processor/144 Memory system interconnected via a three stage indirect network S12:3:4

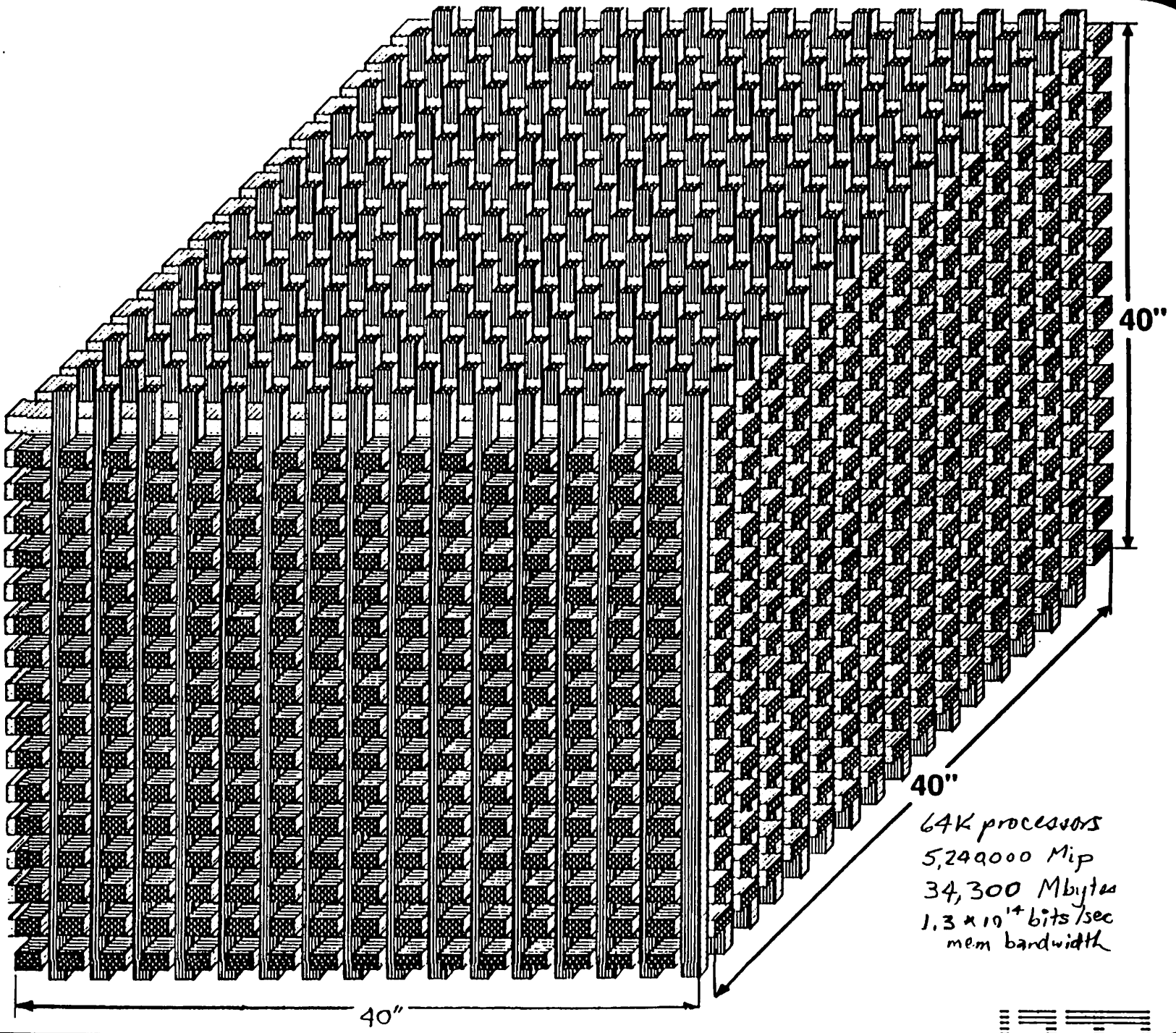

Memorys
12 of 144 shown
SwitchZ
(3 of 36 shown)
Processors
SwitchX
SwitchY
Processors
SwitchX
SwitchY


24 Processors &
SwitchX: 12 way
space switching
with 2 channels
per exit port
4 Memorys &
SwitchZ: 4 way
space switching
with 2 channels
per exit port
SwitchX to SwitchY
connection port,
2 channels
SwitchY to SwitchZ
connection port,
2 channels
SwitchY: 6 inputs, 3 way
space switching with
2 channels per exit port

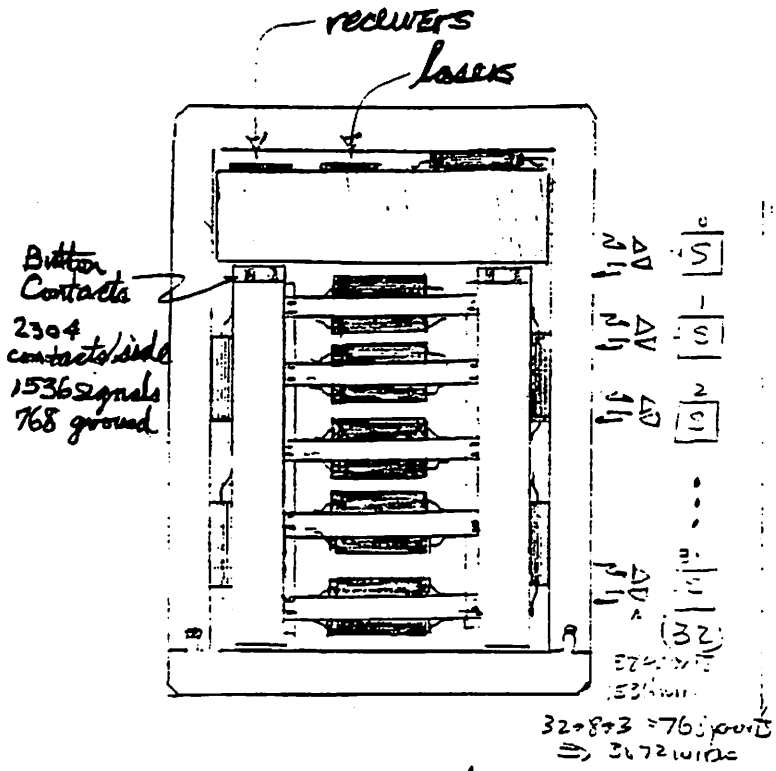




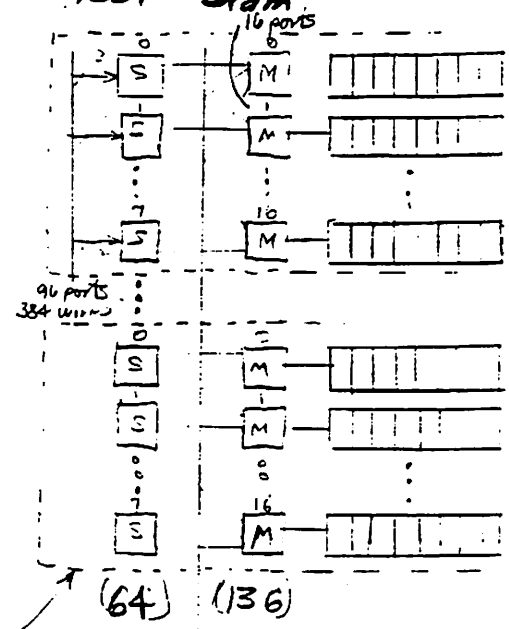
96 Processor/144 Memory system with 12:3:4 switching


40"
40"
40"
64K processors
5,249000 Mip
34,300 Mbytes
1.3 x 10^14 bits/sec
mem bandwidth

SSM Module Packaging



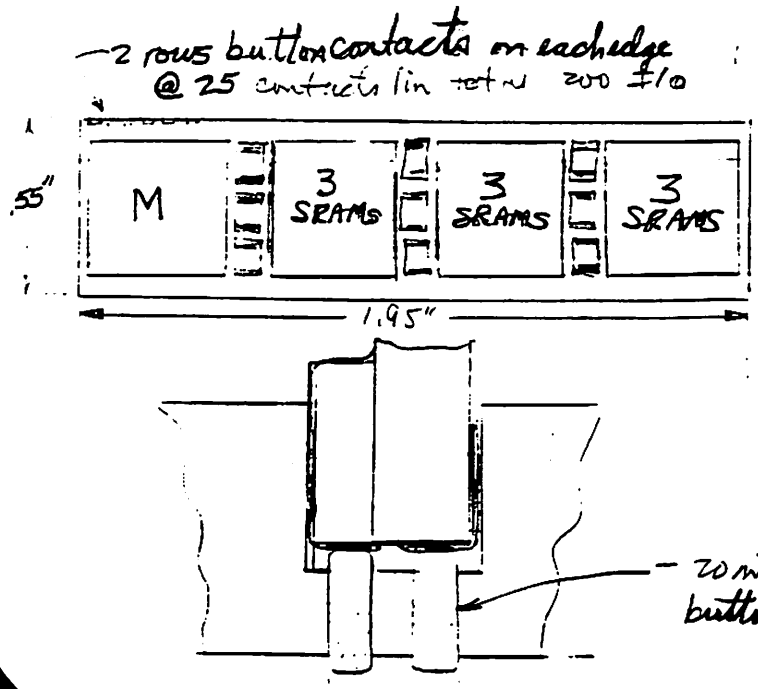
SSM Module
192 S
136 M
1224 Sram

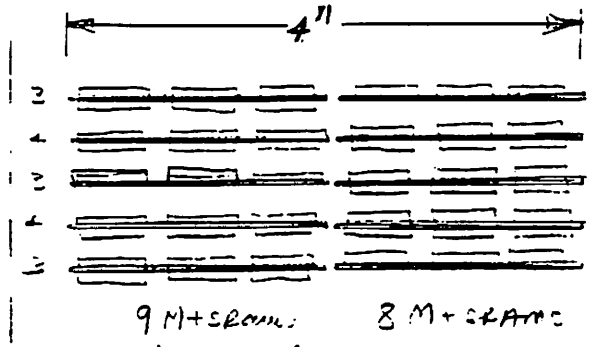


64+17×2 = 2176 ports ⇒ 8704 wires

Allow 4" per sub module containing
16 switch chips
17 memory controller
153 static rams
stacked 3 high

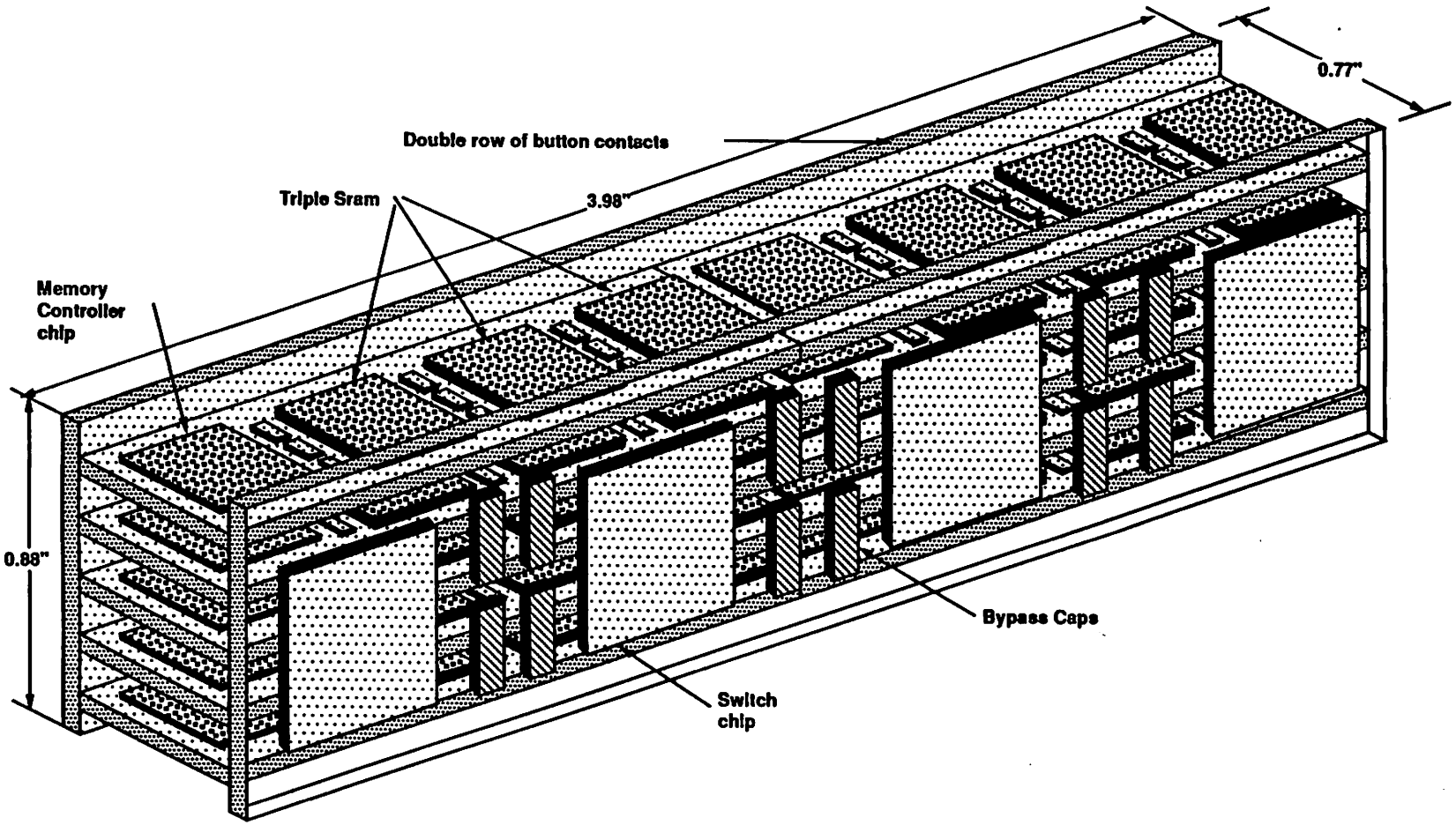
Note submodules may be removed from SSM module and disassembly down to dual M+9 Srams





Sub Module side view

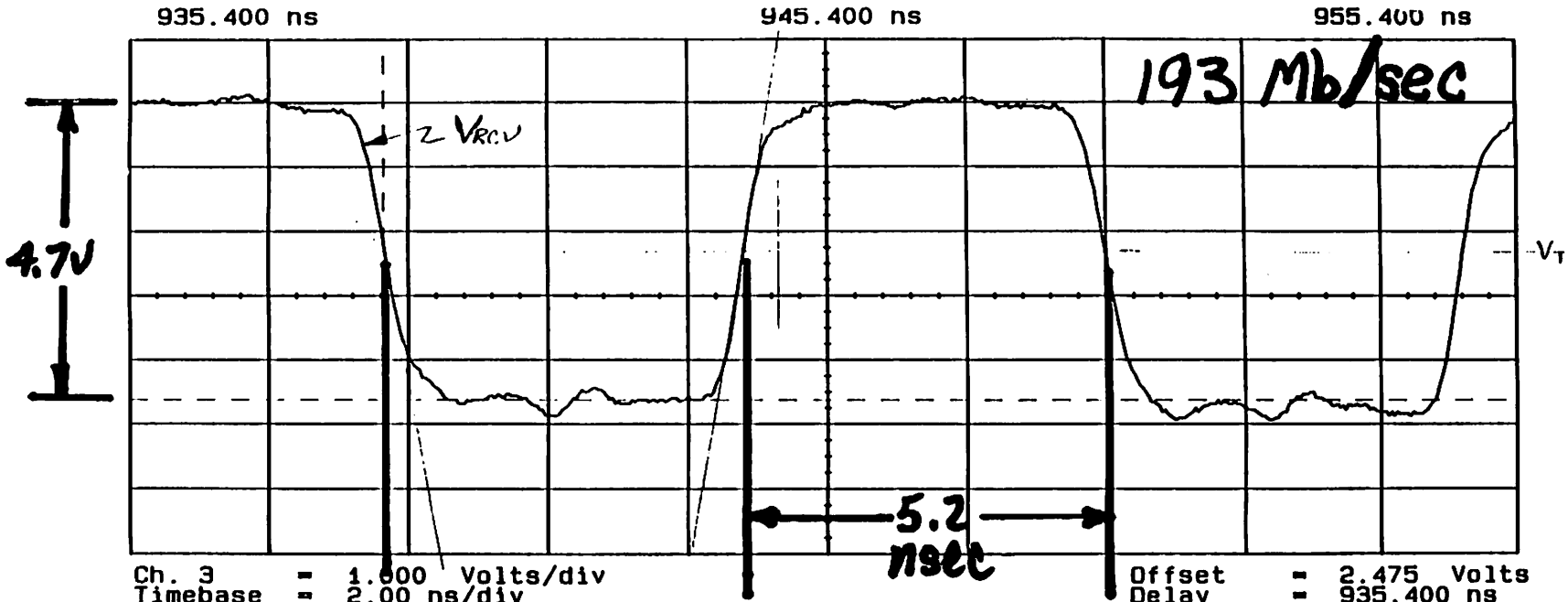
Integrate DC/DC converters every 5 inches as in PSC module


0.77"
Double row of button contacts
Triple Sram
3.98"
Memory
Controller
chip
0.88"
Bypass Caps
Switch
chip

Monarch Enabling Technology

* High speed signaling
* High density packaging
* High density DC/DC converters
* Integrated free space optical interconnect
* Liquid cooling

First Generation High Speed Signaling



Ch. 3	= 1.000 Volts/div			Offset	= 2.475 Volts
Timebase	= 2.00 ns/div			Delay	= 935.400 ns
Ch. 3 Parameters		Freq.	= 96.5400 MHz	Period	= 10.3584 ns
Rise Time	= 722.2 ps	+ Width	= 5.2080 ns	- Width	= 5.1504 ns
Fall Time	= 820.8 ps	Preshoot	= 2.702 %	Overshoot	= 7.432 %
P-P Volts	= 5.0937 Volts	RMS Volts	= 3.8149 Volts	Dutycycle	= 50.27 %

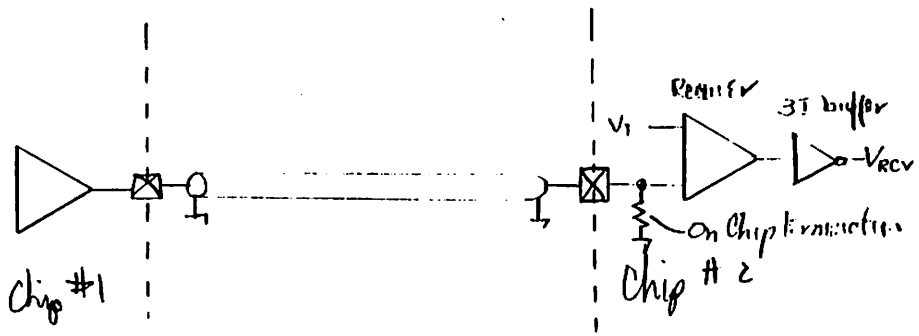
10.12.89

SCM Signalling

Data on chip (after ECL → CMOS level conv.)

$\overline{RXD}$ (after 3-inverter buffer)

1.6 μ CMOS



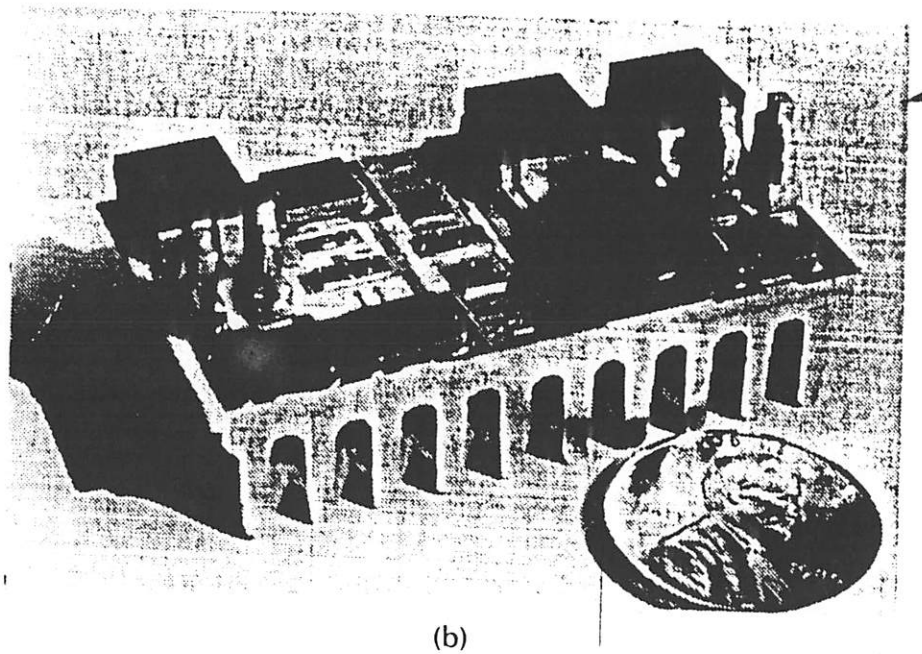
Second Generation High Speed Signaling

* Next generation chips will incorporate enhancements to improve noise sensitivity, power discipation, process variation tolerance, and latency

- PAD latchup protection
- duty cycle correction of clock and data receivers
- new chip synchronization scheme
- gated clock circuits to minimize power discipation
- process compensated line driver and receiver
- digital controlled delay line for bit level data recovery
- better testibility
- static logic for low speed circuitry
- low cost, high speed socketable chip package
- new scheme for transfering data inside chip interior to minimize latency

High Density Power Supplys

(a)



(b)

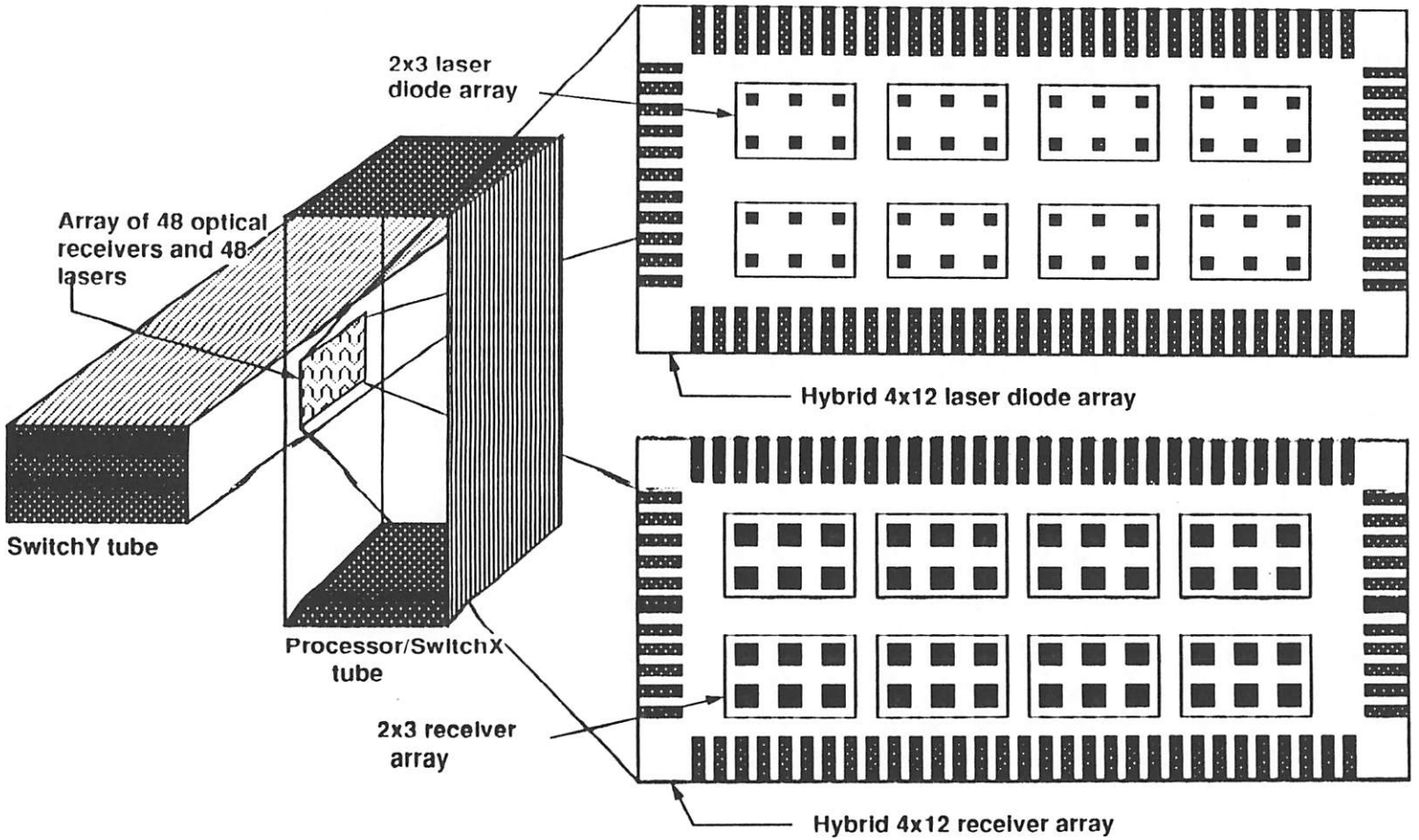
Fig. 1. (a) Block diagram illustrating the concept of a distributed power supply for a mainframe computer. (b) Physical prototype of the 10 MHz, 50 W, converter being developed at MIT for use in distributed power supplies.

2 scale 2 to 1

4.5cm × 1.75cm × 1.25cm
1.77" × .689" × 0.492"
0.603in³

penny

Integrated Free Space Optical Interconnect



Monarch

Software

- Programming Model
- Operating System
- Synchronization
- Parallel Debuggers and Tools
- Latency
- Machine Model

Monarch

Programming Model

- **MIMD Multiprocessor**

  Virtual Multiprocessor - a partition of the machine.

  - **P** Identical Processors - Space Shared

- **Uniform Shared Memory Paradigm**
  - Large uniformly accessible shared memory
  - Weak Coherence

- Message Passing Paradigm

  500 nano-second processor-processor interrupt

- Dataflow Paradigm

  I-structure references (tagged memory)

Monarch

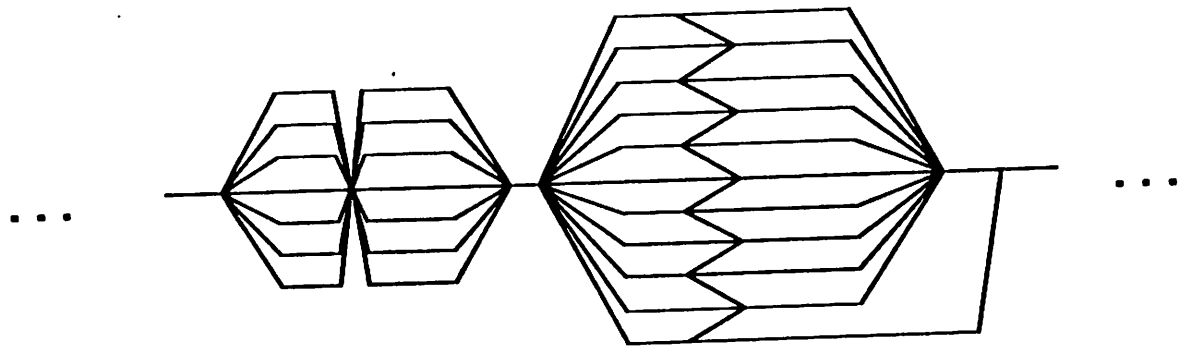
Granularity

- **Fine Grain Parallelism.**
  - **Single Program/Multiple Thread**
    - Procedure Level
    - Explicit Specification of Parallelism
    - Explicit Synchronization
  - **Coarser Grain Parallelism Readily achieved**

- **System Support - Fine Grain Shared Memory**
  - Operating System
  - Runtime Library
  - Programming Language(s)
  - Hardware

Monarch

Program Execution



Process

Sync

Process

Combine

Process

Locally Combine

Process

Monarch

Operating System - MACH

- **BBN ACI enhanced MACH**

  Multiprocessor Unix variant - 4.3 BSD compatible
  - DARPA supported research (via CMU)
  - ACI versions on 2 different multiprocessors
  - ACI parallelized I/O and other bottlenecks

- **Task/Thread**

  Fits Single Program/Multiple Thread

  Large Scale Parallelism

- **MACH Virtual Memory**

  Copy-on-Write

Monarch

MACH TASK

- **Execution Environment**
  - Basic Unit of Resource Allocation
  - Generally High Overhead
- **Virtual Address Space**
  - Ordered collection of mappings to memory objects
- **Access to System Resources**
  - Processors
  - Virtual Memory
  - Port Capabilities

Unix Process = 1 Task + 1 Thread

Monarch

MACH THREAD

- **FLOW OF CONTROL**
  - Independent Program Counter
  - Stack Pointer
  - All THREADS of TASK share the TASK resources
- **Basic Unit of CPU utilization**
  - Generally Low Overhead
  - "Lightweight Process"
  - "Parallel Coroutine"
- **Key to MACH support of parallelism**

  Unix Process = 1 Task + 1 Thread

Monarch

MACH VIRTUAL MEMORY

- **COPY-on-WRITE Sharing**
  Copying is delayed until a thread performs a write on COW shared page of memory. Only necessary copying.

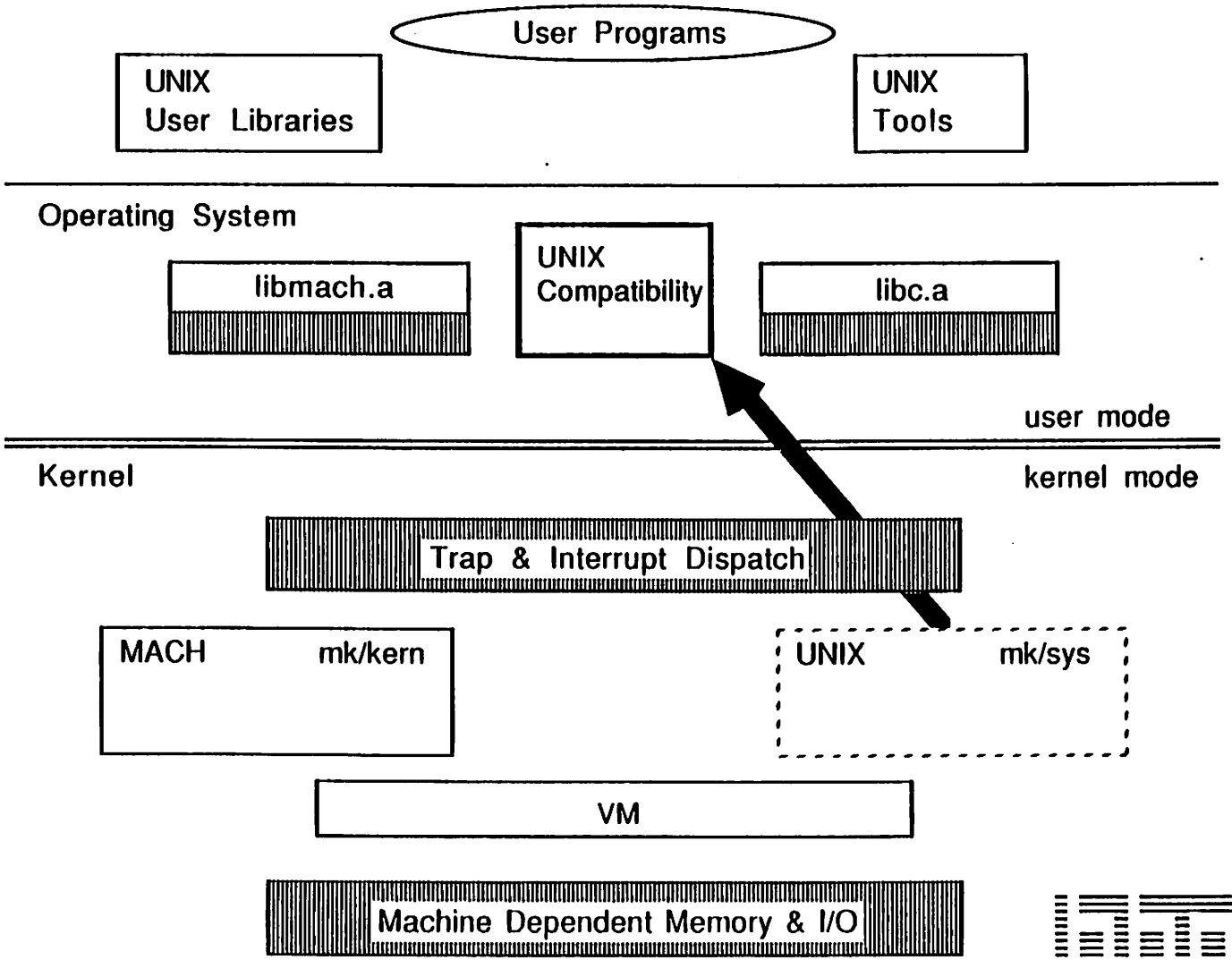
- **READ/WRITE Sharing**
  Memory object is mapped into multiple address spaces

- **NO Sharing**
  Memory object is private to a task.

INHERITANCE and PROTECTION
on a per page basis

Monarch

MACH Structure



Monarch

STEAL

- Every Memory Location is **tagged**
- **STEAL** operations sets "stolen" tag bit
- Store clears "stolen" tag bit.
- Load or Steal of stolen location hangs till tag bit is clear
  - Prefetch quietly fails
  - Override Steal No-retry
  - Other memory access variants

- **STEAL** used to construct synchronization mechanisms
  - LOCKS (spin-wait)
  - Write-once storage
  - **LOG P** Barrier
  - **LOG P** Combining and Prefix

Simple Locks

A location which is stolen and then written implements a lock directly.

Code to guarantee exclusive access to the variable SharedCounter using the lock variable Lock

```
Steal Lock
Update(SharedCounter)
Lock = 0      ;Storing into lock "unsteals" it
```

Monarch

Barrier Synchronization

Use a divide and conquer approach

Synchronize P / 2 pairs of processors using simple locks implemented with Steal

Synchronize P / 4 pairs of processors, 1 from each pair in the first stage

Repeat recursively

As processors drop out they can poll a location. The last processor can set this location, releasing all other processors simultaneously.

Monarch

Data Synchronization

Use Steal Primitive Directly

1. Presteal Shared Data to be Computed
2. Compute Data
3. Write Data Once Computed
4. Use Data whenever needed relying on Steal mechanism to guarantee integrity of value

Technique useful to avoid task synchronization and pipeline tasks together when a piece of data is computed by a small number of processor and then read by a large number or random set of processors

Monarch

Write-Once Storage

Write once storage can be prestolen and then used without synchronization. Honoring the stolen bit ensures that data is not used before it is produced.

```
for i from 1 to N Steal Y(i, myJ)
synchronize
for i from 1 to N
    Y(i, myJ)  =  f(Y(i-1))
```

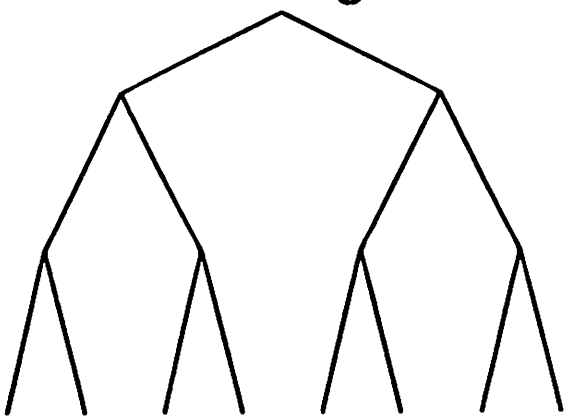
Monarch

Parallel Prefix and Result Combination

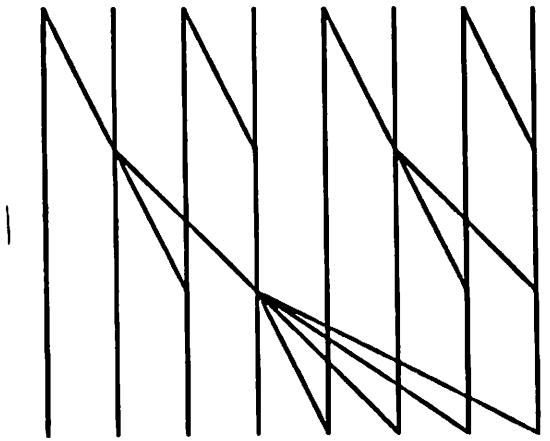
Use a divide and conquer approach as in barrier synchronization.

Combining read allows for simple algorithm

Combining Tree



Prefix Tree



Monarch

Parallel Debuggers and Tools

- **BBN ACI Experience in Parallel Programming**
  - Window Oriented Source Level Debug
  - **GIST** - Event Logger
    Visualizing dynamic behavior of Parallel Programs

- **Parallel Unix Utilities**
  - Make
  - Grep
  - Shell

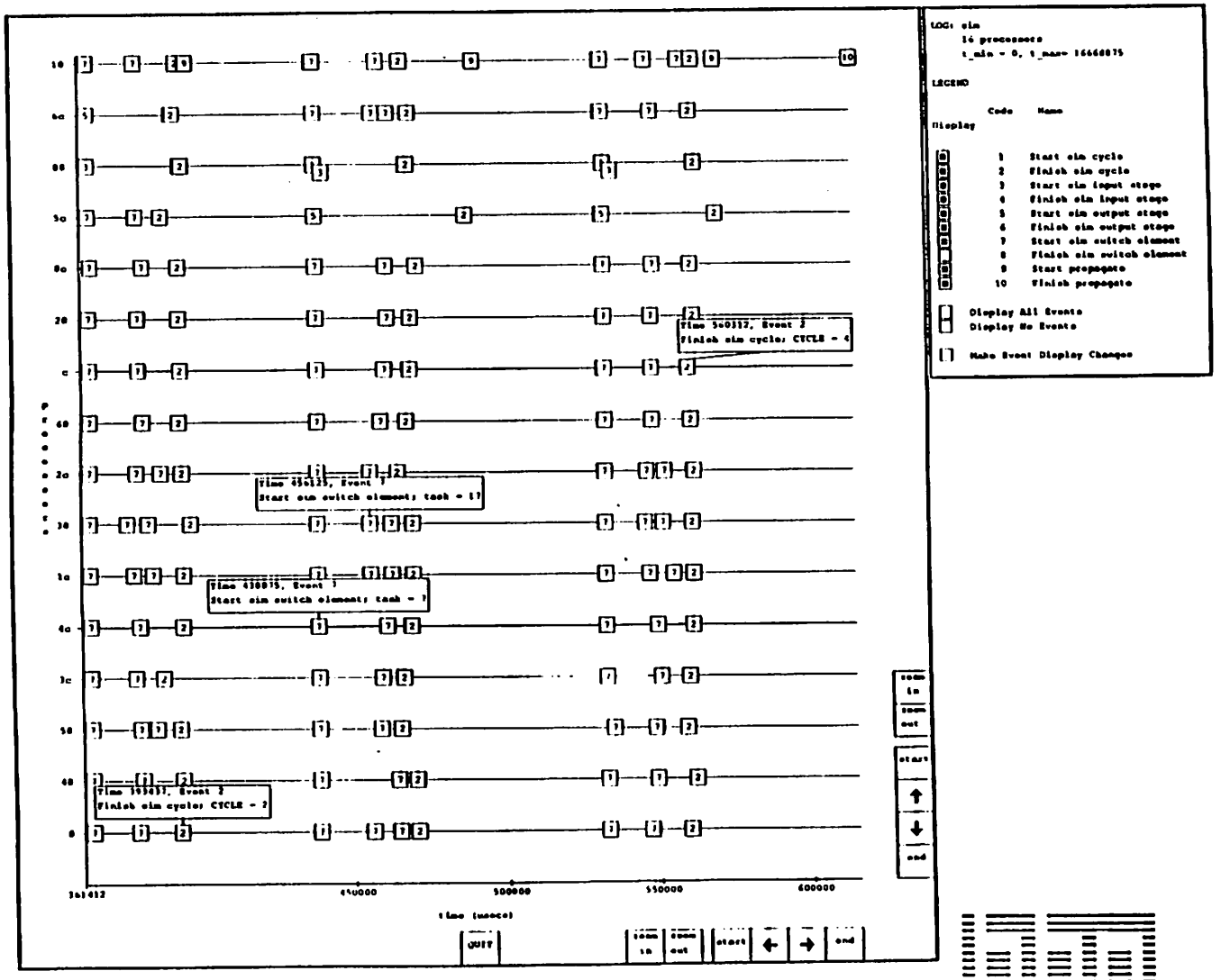
    Wildcards - `cp *.c /usr/darpa/newdir`

    Pipelines - `tbl | eqn | troff |psprint`

    Loop - `foreach f (*.c)`

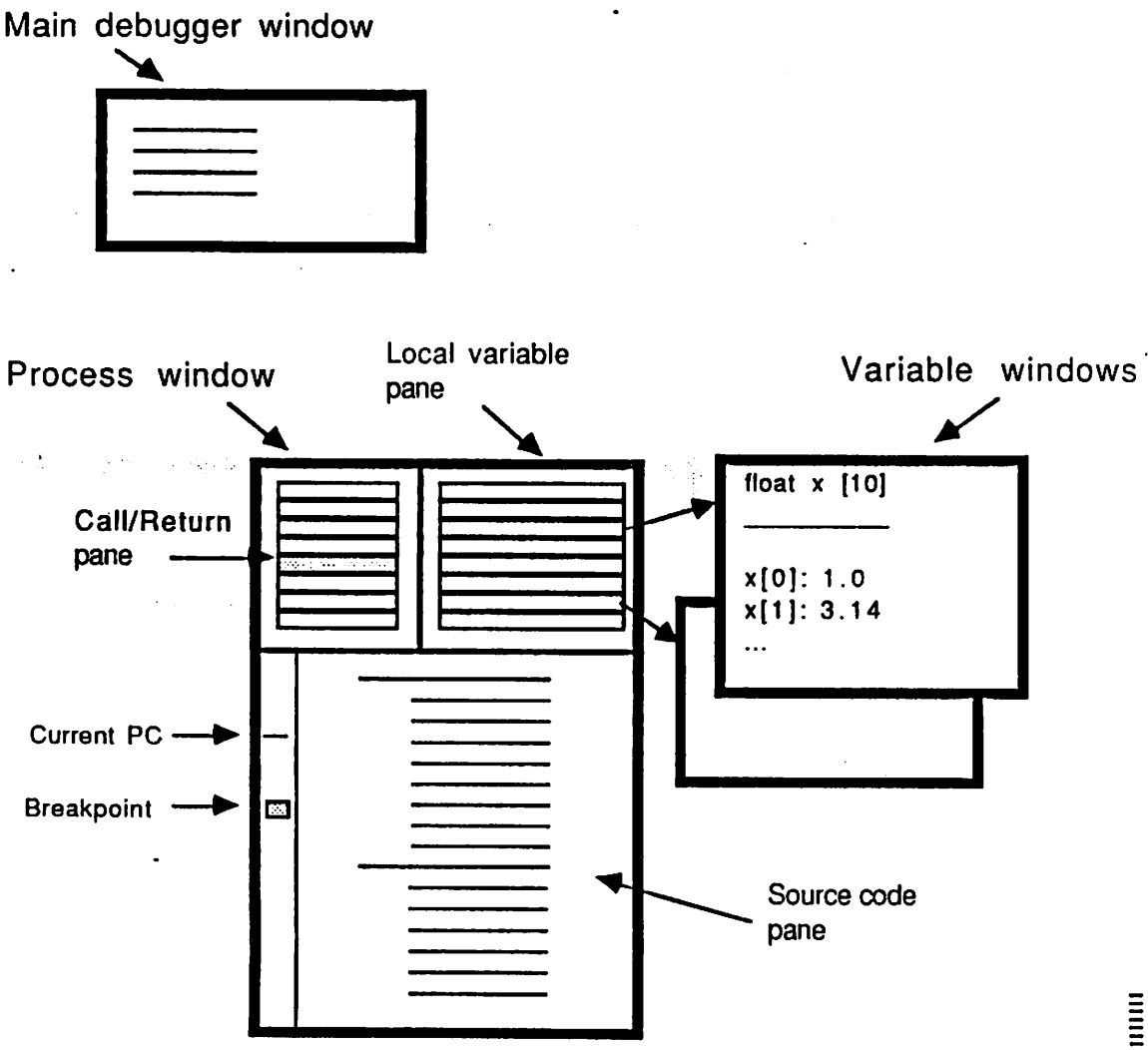
Monarch

Gist



Monarch

Window Oriented Debugger



Monarch

Latency

The roundtrip time to memory and back

Performance is limited by operand fetches

- **The Switch Network is Very Good.**
  C-MOS ~350ns GAs ~ 200ns

- **Commercial Processor**
  Designed for **ultra-low latency** memory systems
  Multiprocessor MIPS < Uniprocessor MIPS

- **Latency Masking**
  Mechanisms for tolerating long
  roundtrip times to memory and back

Monarch

Masking Latency

Use extra Bandwidth to regain processor performance.

- **Parallel Memory Access and Computation**
  Insufficient natural instruction-level parallelism

  - **Instruction stream blocked by memory access**
    Allow multiple outstanding memory accesses.

  - **Instruction stream blocked by data dependency**
    Increase instruction level parallelism by generating earlier memory requests - **PREFETCH**

Monarch

Latency

Simple Model	I0	I1	I2	**L3**	I4	I5	I6	....		
Latency	I0	I1	I2	**L3**	- - -	- - -	- - -	I4	I5	I6...
Overlap Inst/Mem	I0	I1	I2	**L3**	I4	I5	I6	I7	I8	I9 ...
Load	I0	I1	I2	**L3**	I4	**L5**	- - -	I6	I7	I8 ...
Multiple Loads	I0	I1	I2	**L3**	I4	**L5**	I6	I7	I8	I9 ...
Data Dependency	I0	I1	I2	**L3**	I4	**I5**	- - -	I6	I7	I8 ...
Ideal Prefetch	I0	**P**	I1	I2	I3	I4	**I5**	I6	I7	I8 ...
Monarch Prefetch	I0	**P**	I1	I2	I3	**L3**	I4	**I5**	I6	I7 ...

Monarch

Benchmark Simulation

Problem	Problem Name	Cray 1S Runtime (microseconds)	Monarch versus Cray runtime 1,024 processors	16,384 processors	65,536 processors
1	Small Matrix	2,270,000	14	220	764
2	Very Sparse Large Matrix	6,960,000,000	43	682	2531
3	Sparse Matrix	9,900,000	27	425	639
5	Bit Permutation	94,300,000	323	5153	20500
6	Modulo 2 Inverse	322,000	7	24	31
7.1	Sort (small	19,500	25	115	130
7.2	Big Sort)	390,000,000	1191	18987	74713
8	Binary Dynamic	36,600,000	55	107	220
10	Dynamic Programming	1,000,000	7	103	345
11	Pattern Recognition	966,000	55	743	2123
12	Pattern Search w/free bits	1,280,000	27	405	1384
13	Select and summation	466,000	24	370	1347
14	Every Eleventh Bit	28,000	113	903	1217
15	Decimate Binary Streams	65,000	31	419	1016
16	Right Justify	62,000	21	328	1216
17	Modulo 2^46 inverse	291,000	49	766	2798
18	Frequency Histogram	229,000	36	398	954
19	Find 100 Zeros	10,000	10	80	122
20	Transpose Blocks of Words	2,637,000	32	480	1884
21	Table Lookup and Summatior	12,500	56	278	313
22	Fast Fourier Transform	6,500,000	18	274	1066
23	Manhattan Distance	780,000,000	19	305	1209
24	Bit Manipulation	140	6	11	13
4.1	Frequency Search	11,400,000	735	8906	22800
4.2		13,230,000	306	4147	8969
4.3		500,000	407	1667	2000
4.4		1,320,000	329	2260	2694
9.1	Baskett	363,000	40	81	79
9.2		34,800,000	26	258	909
9.3		1,590,000,000	143	2298	9191
9.4		19,800,000,000	218	3474	13944

Monarch Development Plan

- **Three year R & D Plan:**
  - integrate commercial microprocessor
  - develop and test VLSI components
  - port Mach O/S and develop programming software
  - construct 128 processor demonstration machine
- **Additional Optional Design Study on High Density Packaging**

Monarch

Year 1 Hardware Tasks

Develop Pad Test Chip

- verify latchup protection pad layout
- verify programmable data delay circuit
- verify several process compensation circuits
- verify token recovery circuit
- verify power-up logic

Low cost, high speed package for Switch and Concentrator

Develop Test Board to exercise Pad Test Chip

Finish Development of Switch and Concentrator chips

Develop chip/commercial tester interface and test program

Develop Memory, Utility and Backplane PC boards

Develop Specs for Processor & Memory Controller Interfaces

Develop circuit level simulation

Monarch

Year 2 Hardware Tasks

Develop Processor Interface chip

Develop Memory Controller chip

Produce Memory Controller and Processor Carriers

Chassis, power supply, cooling for 128 Processor Monarch

Develop VME bus I/O adaptor board

Monarch

Year 3 Hardware Tasks

Complete Build of two 8 Processor Monarchs

Complete Build of one 128 Processor Monarch

Run the machines

Complete Documentation

- Features
- Hardware Design
- Construction
- Operation & Diagnostics

Monarch

Year 1 Software Tasks

Sun MDS interface

Switch/Concentrator Diagnostic-Debug

Board Level Testing Software

Architectural Level Simulator & Benchmarks

Assembler

Linker

Prefetching C Compiler

Implement Multiprocessing & Synchronization Primitives

Design Machine Dependent Part of Operating System

Monarch

Year 2 Software Tasks

Operating System MDS

Use Simulator to Debug Software

Downloader/Boot Software

Logic Analyzer Based Software Development Station

Machine Dependent Layer of Operating System

Operating System Kernel Port

Essential Operating System Libraries

Operating System I/O

Monarch

Year 3 Software Tasks

Parallel Debugger(s)

Port other Unix Libraries

Port Essential Unix Tools

Identify Operating System Bottlenecks - Redesign and Implement for Massively Parallel Machine

Parallel File System

Documentation

Optional Additional Tasks

- Machine Resource Allocation Strategies
- Checkpointing
- Port X-Windows
- Port NFS File System
- Parallelize and Improve selected tools

Monarch

Project Costs

Year	Hardware	Software	Mgmt	Total
1	$1,075K	$500K	$150K	$1,725K
2	$2,300K	$1,000K	$200K	$3,500K
3	$1,800K	$1,000K	$200K	$3,000K
Total	$5,175K	$2,500K	$550K	**$8,225K**