'folio-file

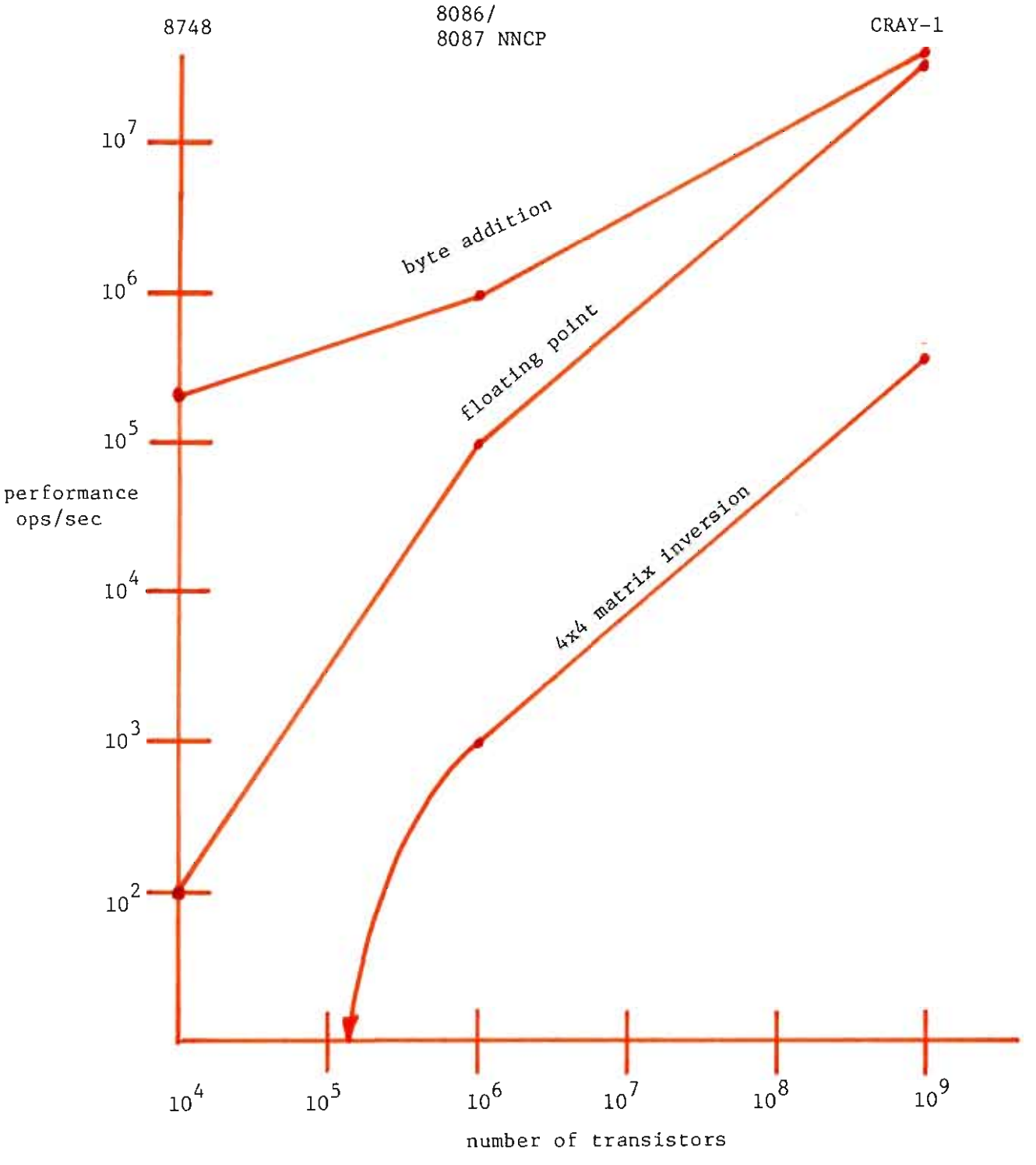
6 POCKETS

NATIONAL | 03-065

Made in U. S. A.

NATIONAL BLANK BOOK COMPANY, INC. HOLYOKE, MASS. 01040

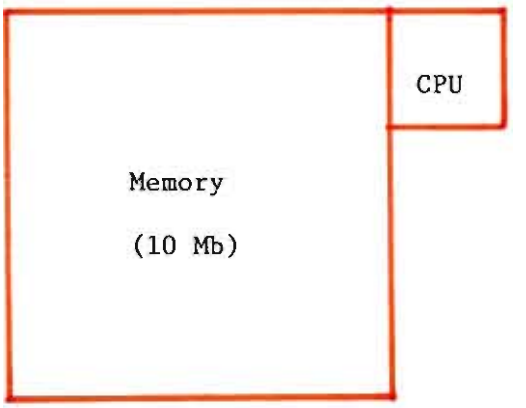
System Performance



Processor Characteristics

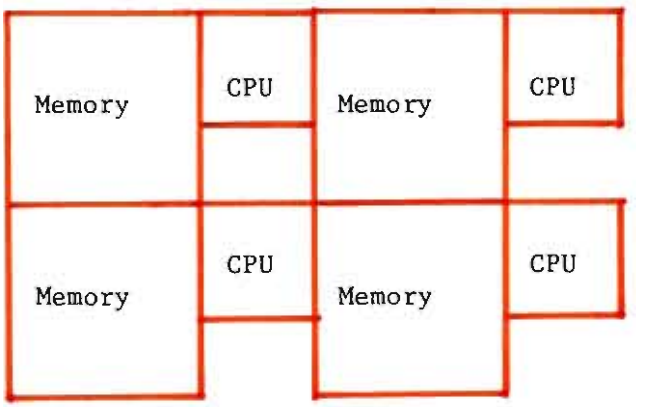
Mainframe

10^8 transistors



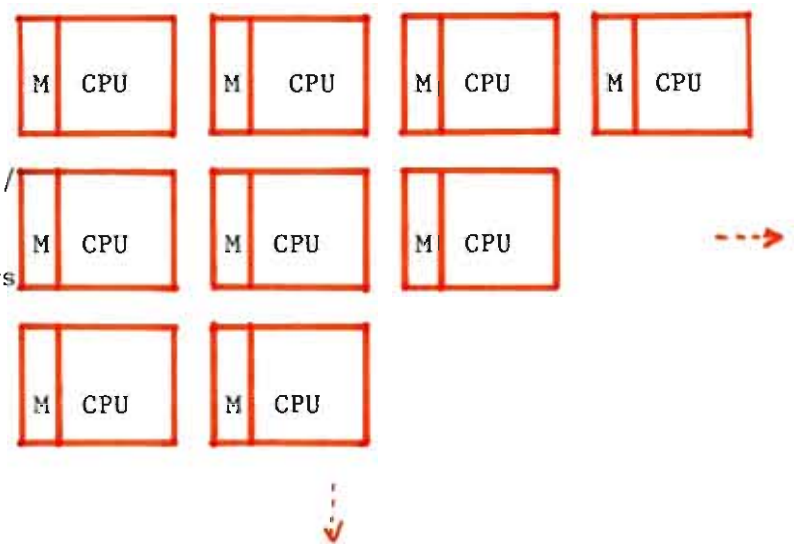
Intermediate

10^6 transistors



Systolic Array/
Macromodule

10^4 transistors

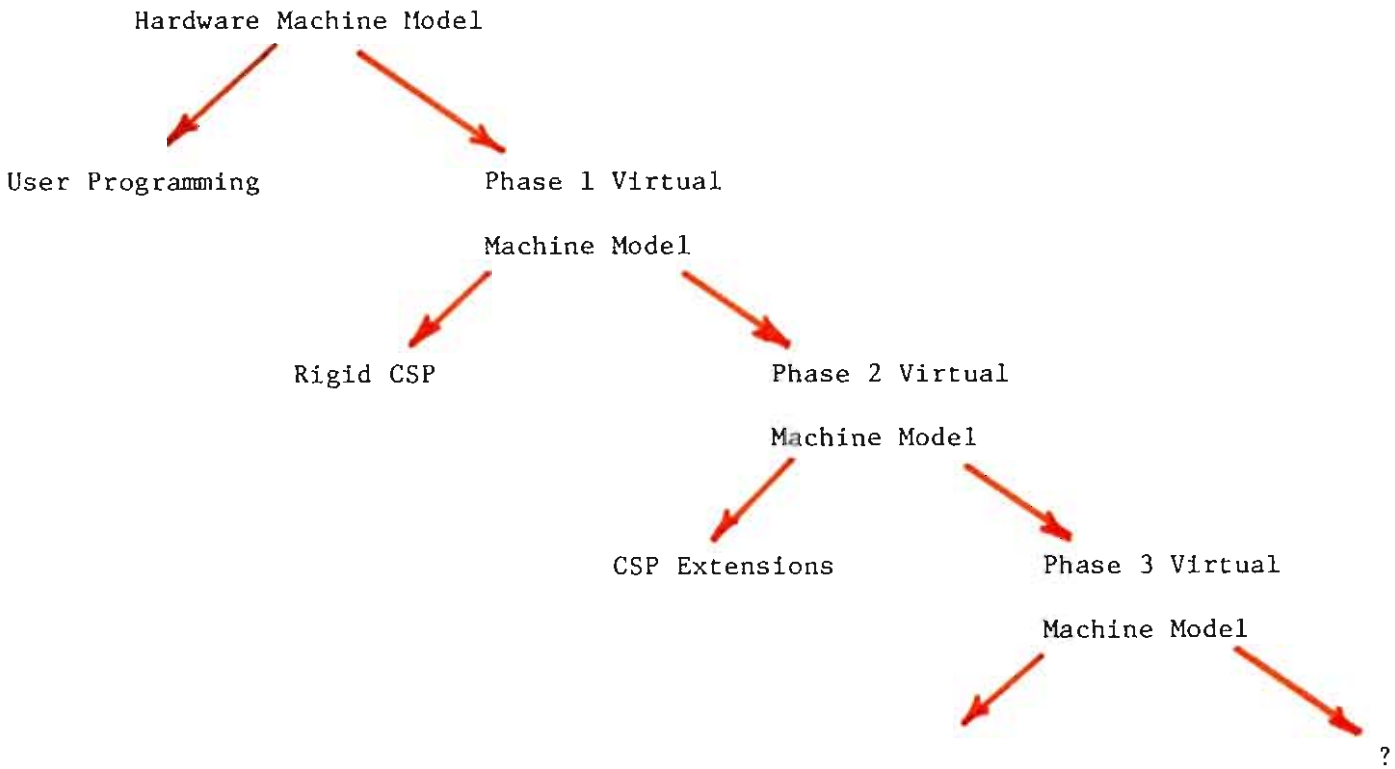


Comparison of Three Types of Processors

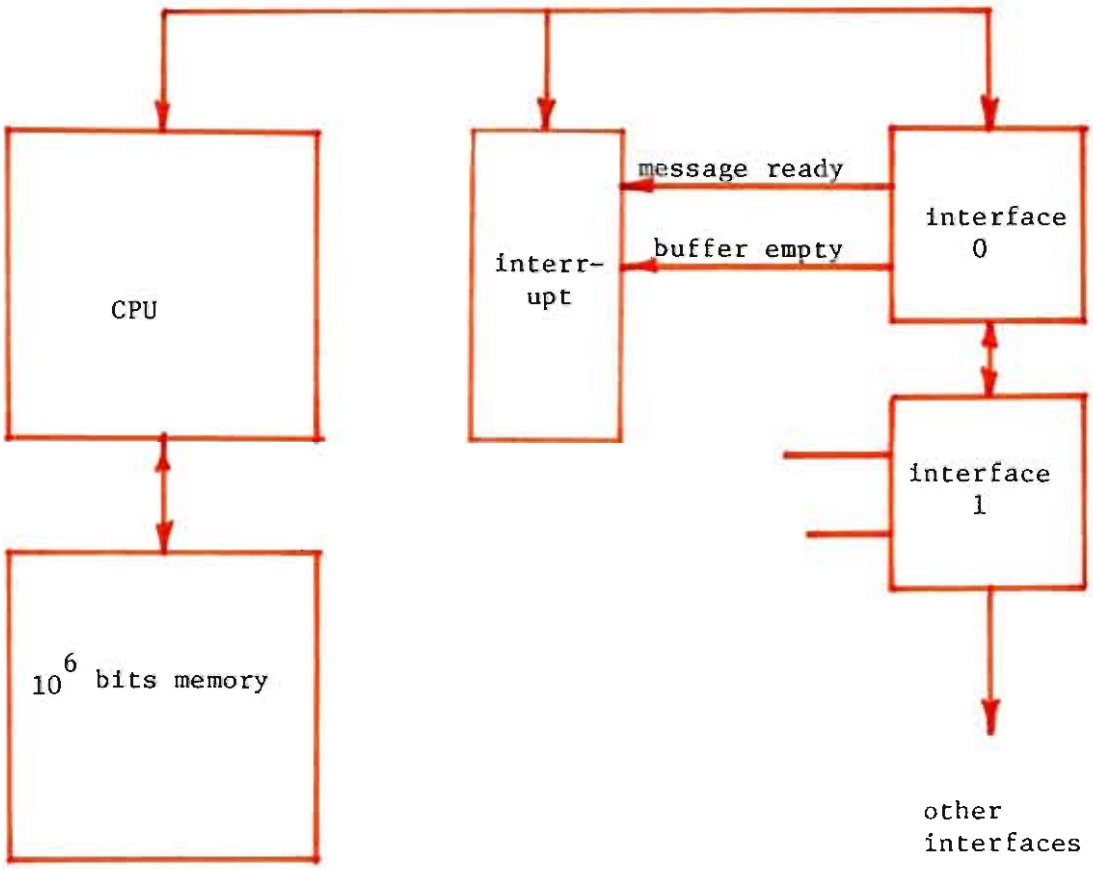
Problem: conversion of a 1000x1000 matrix to upper triangular (no pivoting).

processor	sequential steps	time/step	total time	size	#chips	cost	cost*time
mainframe	10^9	1 uS	10^3 sec	1 Mb		$200K	$2x10^8$
10x10 NNCP	10^7	10 uS	100 sec	100 bds		$200K	$2x10^7$
1000x1000 systolic	10^3	100 uS	.1 sec	10^6 chips/ 10^4 boards		$20M	$2x10^6$

Software Development Plan

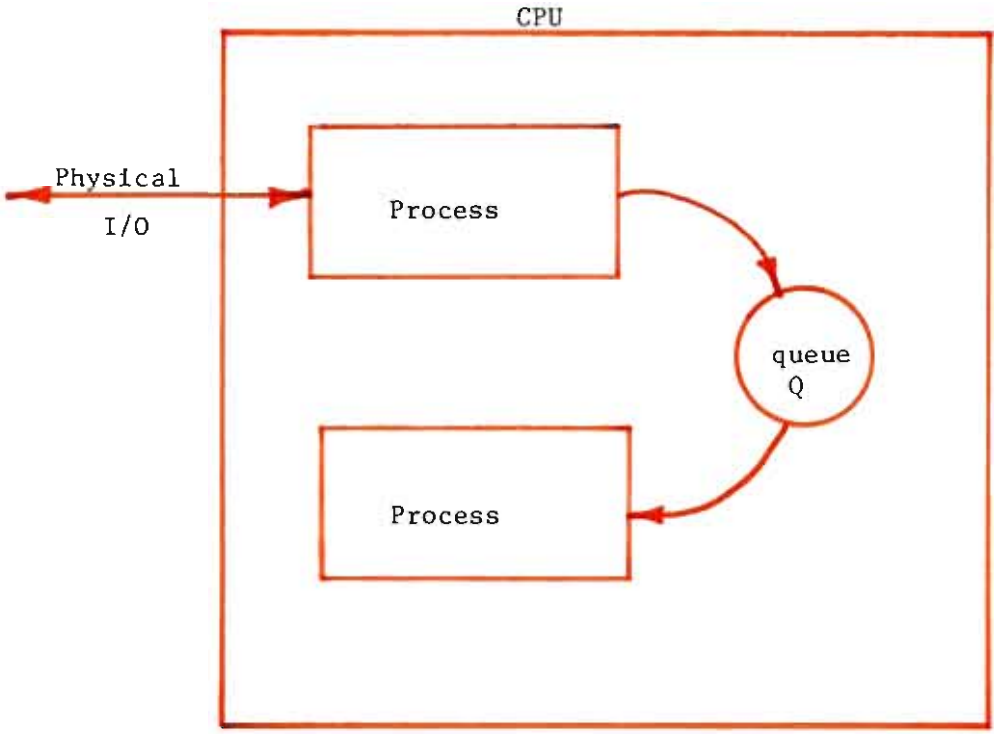


Hardware Machine Model



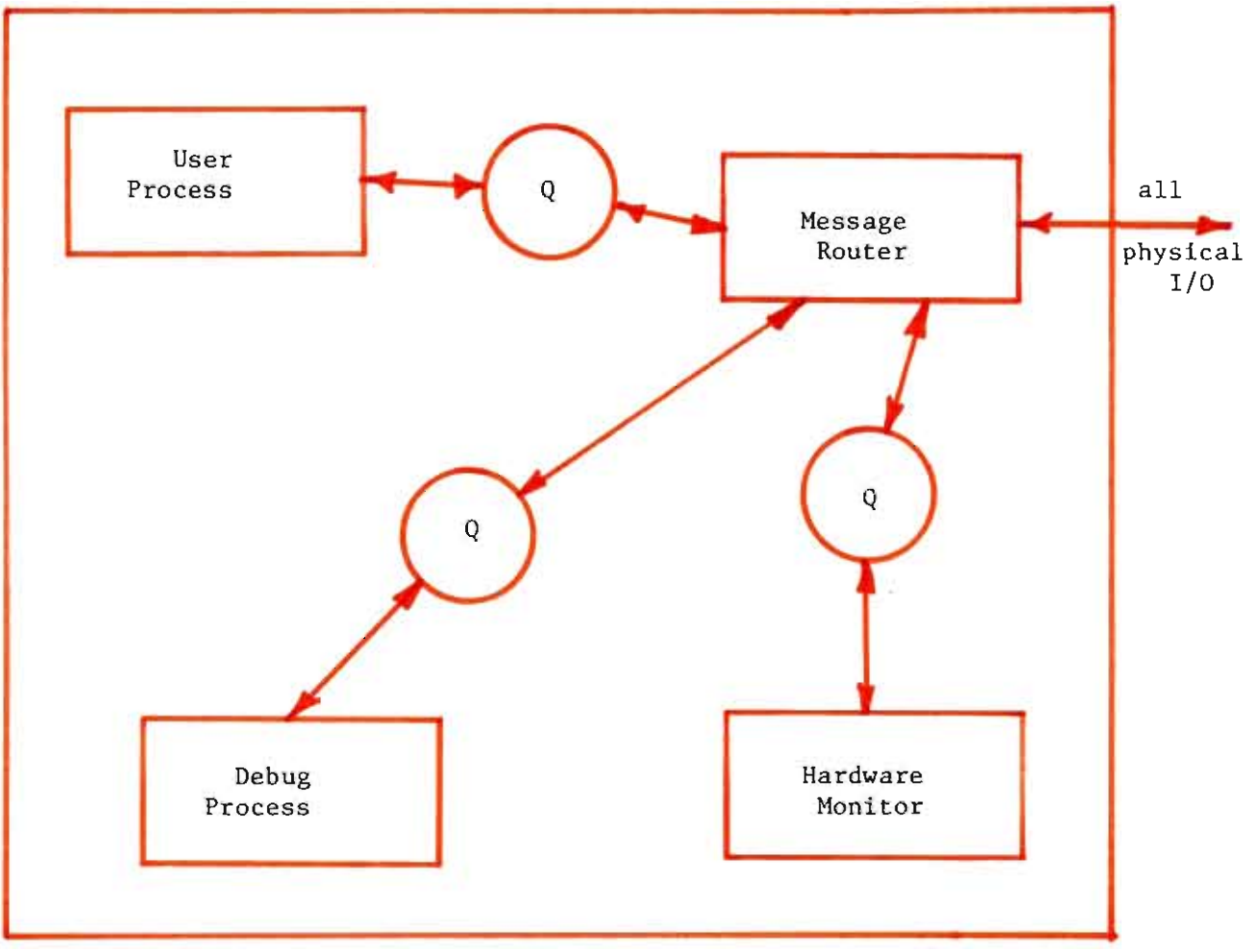
Phase 1 Machine Model

Features: Storage Allocator
Process Scheduler
Interprocess Communication
Physical I/O



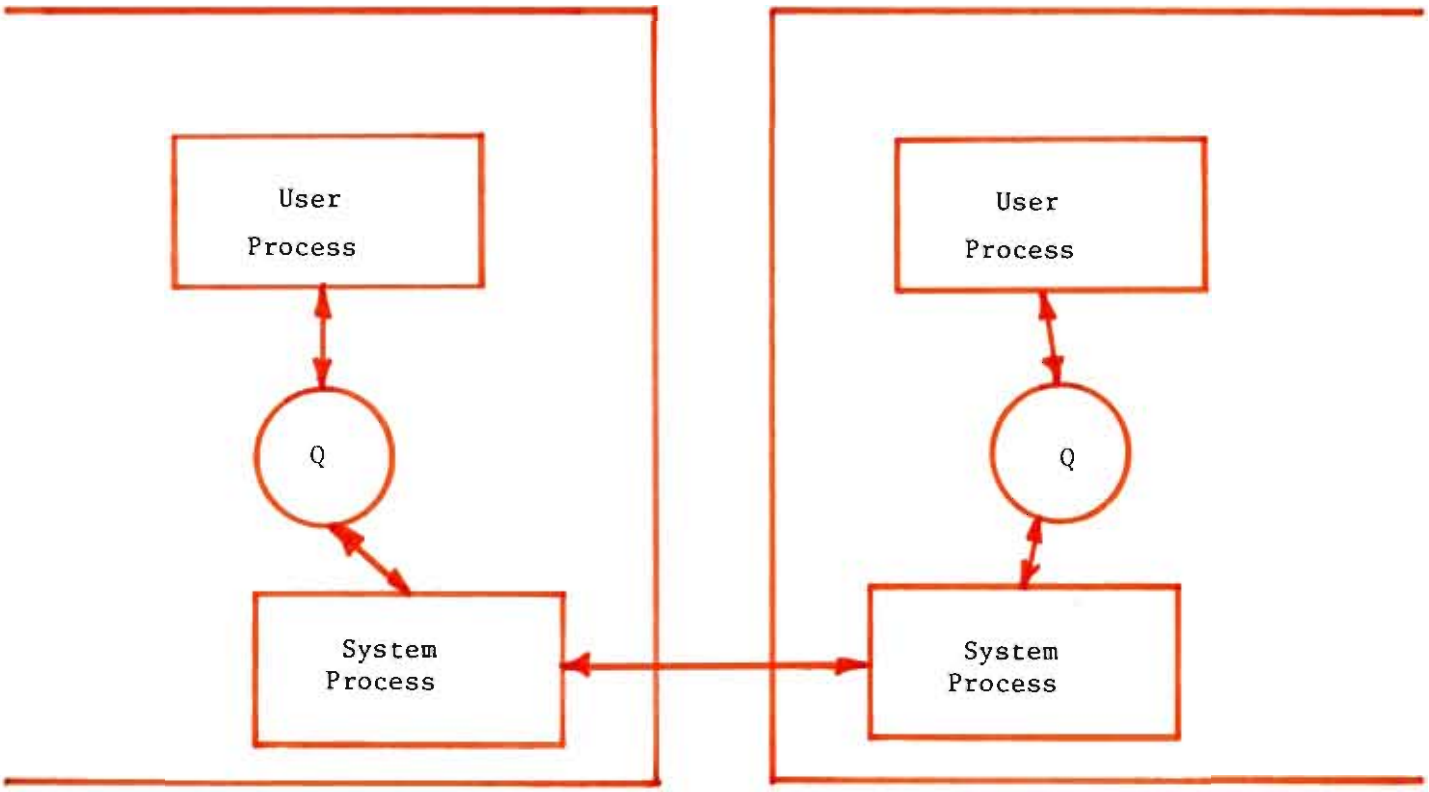
Phase 2 Machine Model

Features: Message Routing
Debug Aids
Diagnostics



Phase 3 Machine Model

Features: Dynamic Process Movement
Global Pointers



Engineering/Science Applications

(Except architecture research)

* PDEs

* Matrix operations

* Sparse matrix operations

* Computer chess

1/2 Simulation (circuits)

Design rule checking

Weather prediction

Oil exploration

"The Deal"

You get: 10:1 increase in available computing resource.

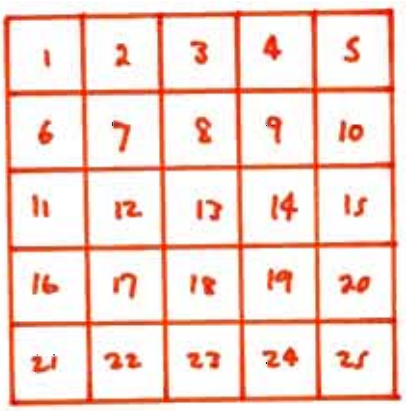
Cost: Different programming style.

Computer Science gets:
Algorithms research and exprience in concurrent computing for only the cost of guidance.

Spatial Decomposition of Problems

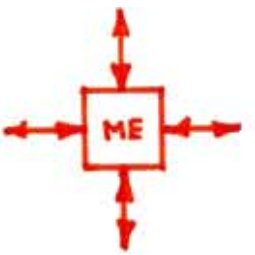
PDE:

conventional



update lattice points sequentially

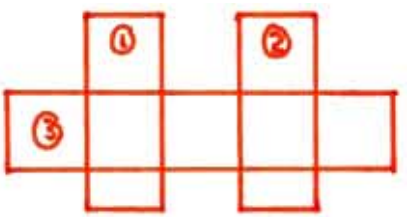
spatial



update yourself

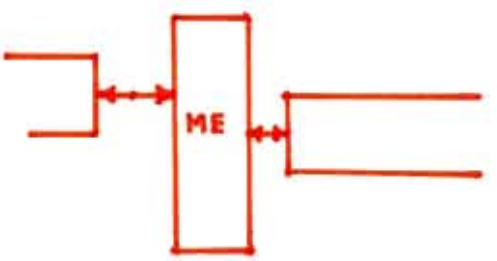
DRC:

conventional



for every pair of polygons
check all design rules

spatial

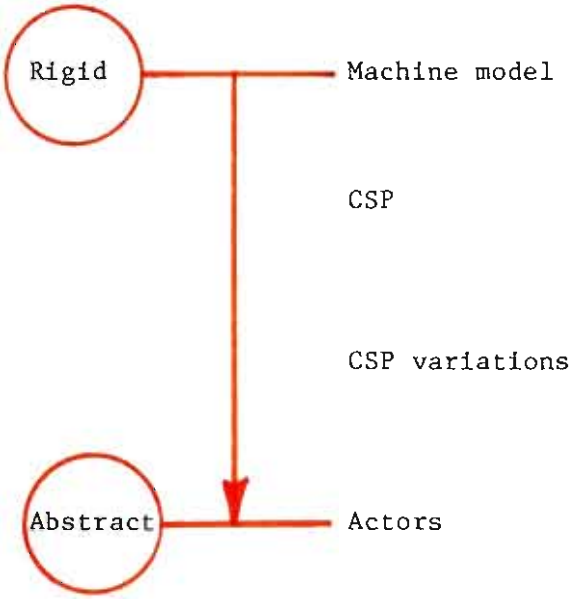


(1)sort self
(2)make sure neighbors are far enough away

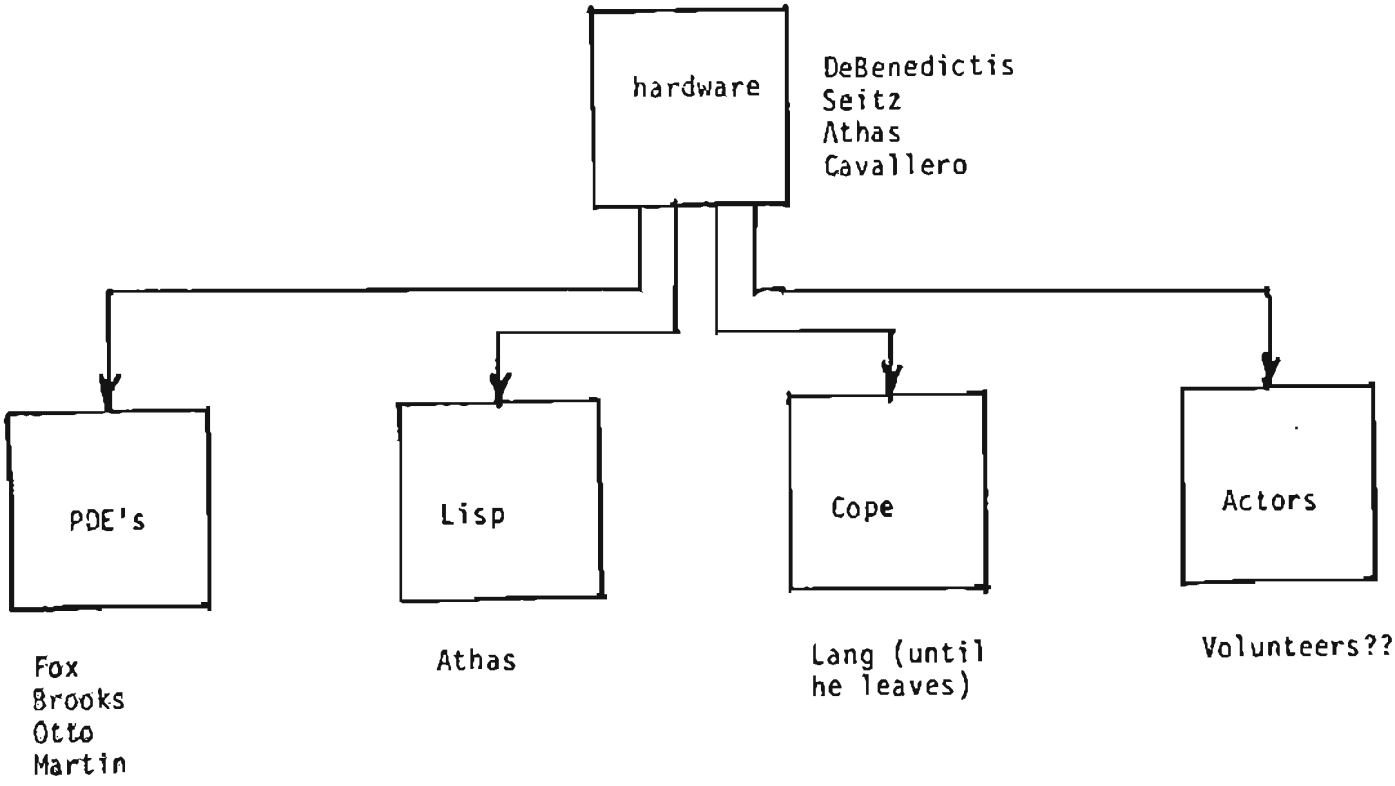
Limitations

Programming styles fall into two catagories:
(1) those that allow general abstraction, and
(2) those that are implemented and are efficient.

(mutually exclusive)



Project Schedule



Motivations For Homogeneous Machine

1) User community exists before the machine is built.
2) Cost/performance of the machine will be considerably less than existing machines (performance leverage).

System Construction

System Step 1: * Nearest Neighbor Communication. (Read/Write to any physical link, and guards.)

* Loader.

User Step 2: Systolic Applications.

System Step 2: * Storage Allocator. (Unix style MALLOC/FREE.)

* Processes.

* Internal Ports.

* External Ports. (Message Routing.)

User Step 3: CSP Applications.

System Step 3: * Process Movment between CPUs.

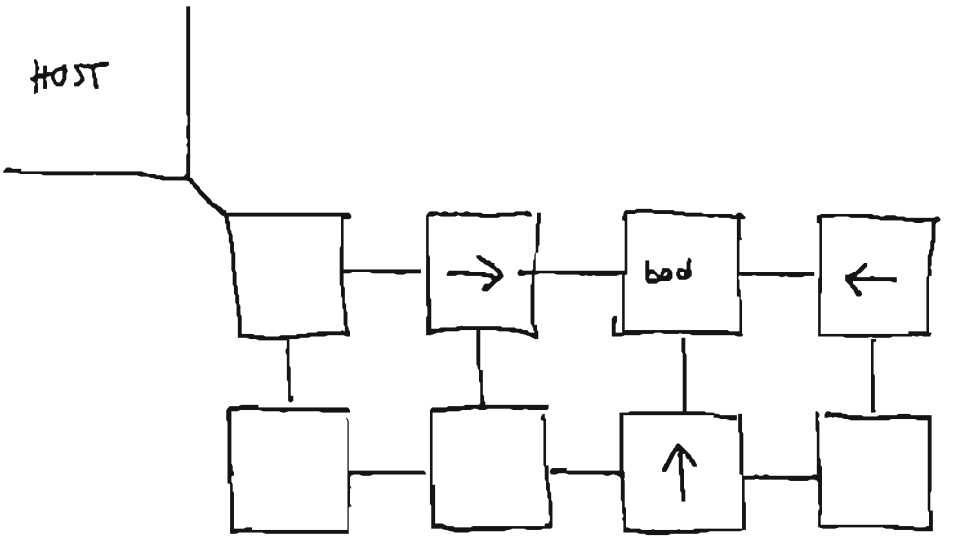
* Global Pointers.

User Step 4: Object Oriented Applications

Actor Systems.

Hardware Diagnostics

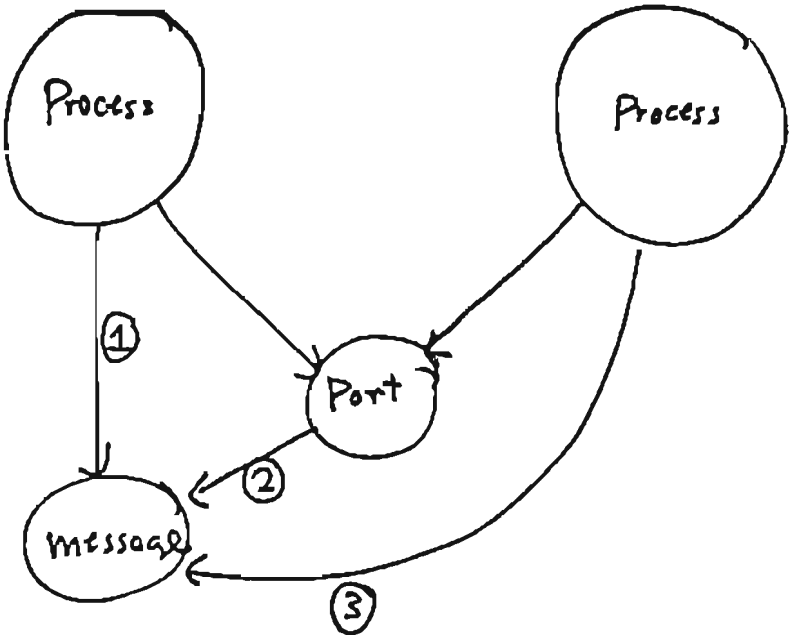
Self replicating diagnostic programs for CPUs and I/O links.



Software Diagnostics

* Provide some secure software, either in ROM or by using a supervisor/user mode feature. Software includes:
    * I/O primitives capable of distinguishing monitor messages from user messages.
    * Debug software. All that is required is deposit/examine memory and process control.
* Provide a debug program in a host with a user interface.

Process / Port / Message System



A process is an independent program except:

(1) Port reference data type.

(2) "Create port" capability (monitor call).

(3) Knows the names of other processes (like knowing how to invoke another procedure in PASCAL)

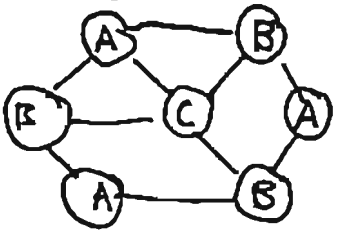
(4) "Create process" capability.

(5) Can send and receive messages via a port reference.

Applications to Systolic Programming

Given a Systolic Program consisting of:

Topology:



Program A:

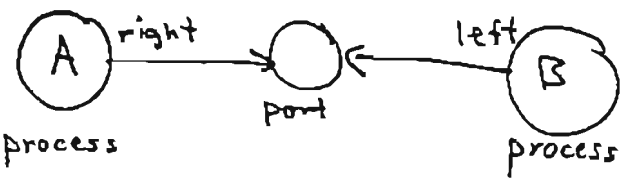
Begin ~ End

Program B:

Program C:

Implementation:

"Master Setup Program" uses the topology specification to create ports and processes.



Applications to CSP Programming

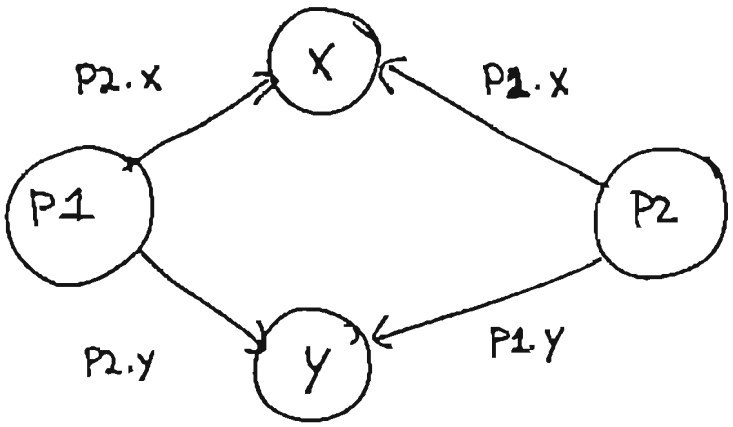
CSP has a nearly static topology (only arrays of processes are allowed), but allows communications between processes, not ports:

P2! X(6); P1? X(j);

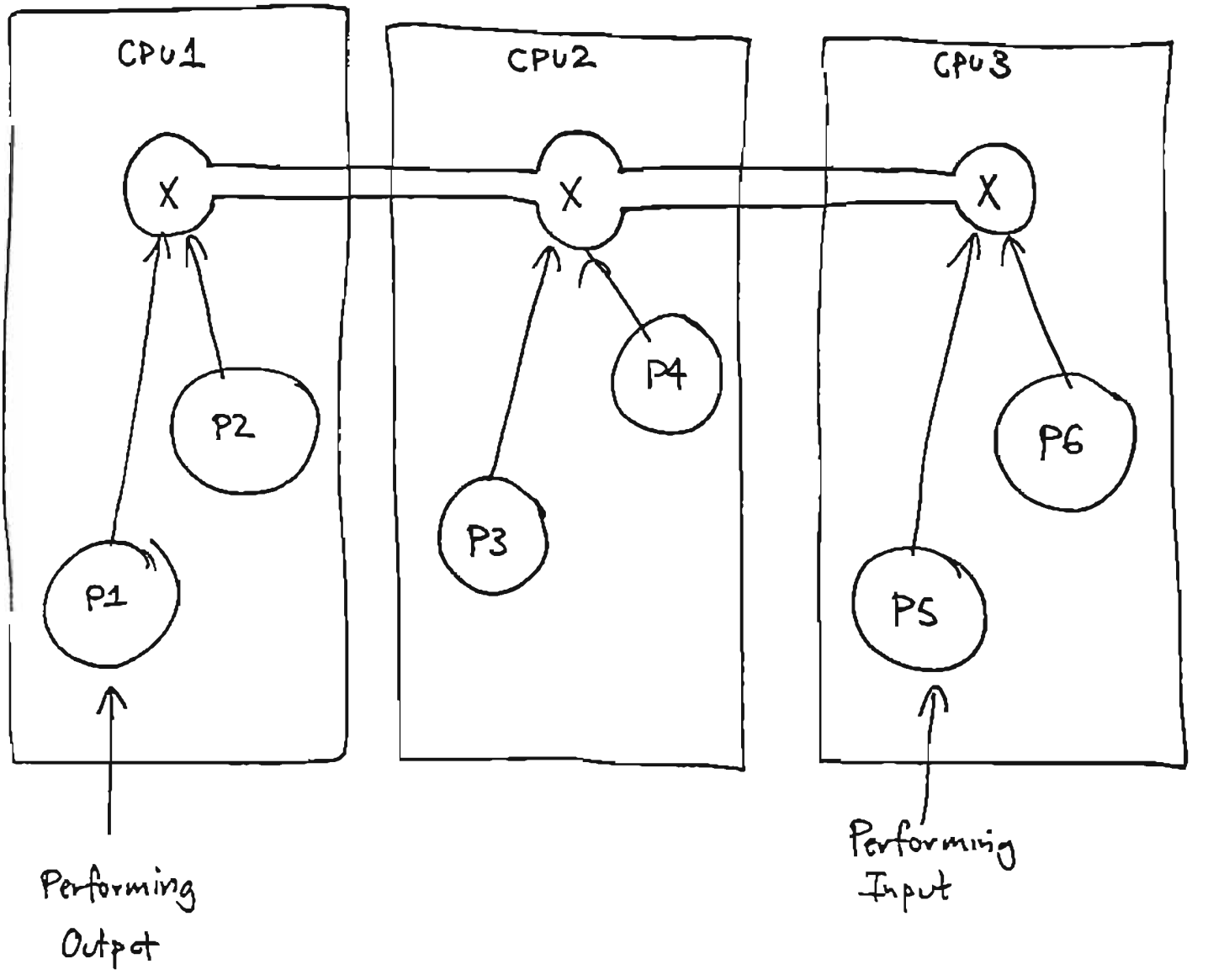
P2? Y(j) P1! Y(8)

Implementation:

Master setup program does following:



Actor / Object Oriented



Hardware Design Considerations

Minimize $\dfrac{\text{Cost}}{\text{Performance}}$!!

Performance Measures:

Maximum size of problem ⟺ Total memory in all processors.

Speed of solving problem ⟺ # of CPUs available to user.

(note about overhead, routing processors.)

CIT 6-cube:

64 CPUs, 1/8 mb / CPU, 1/4 VAX / CPU (@ 10 MHz)

⇒ 8 Mb & 32 VAX

Designable Parameters:

Size of memory

CPU speed / communication time

Size of Memory

We choose: 65K words (16·65K RAM chip)

Too big?? 65K BYTES better?!

Reasons:

Each CPU must be large enough to hold:

operating system	(2K bytes)	
user program	(8K bytes)	(modular)
user data	(48K bytes)	(smaller-the-better)

Experience: HEP people always want more memory, but after some persuasion they will ~~settle~~ settle for more CPUs with equal memory.

Exceptions: Some real programs can run into substantial amounts of code. It remains to be seen if this code can always be modularized. Actor-operating systems are presently huge.

Psychology: We want to encourage concurrent programming. With large amounts of memory naive users adopt a "multiple Von-Neuman" style. By keeping the memory at the lower limits of the useful range, we will coerce users to concurrent programming sooner.

CPU Speed / Communication Time

We choose: 20 μs / 64 bit message transfer
20 μs / floating point ... about equal

To be competetive with conventional VNMs we must have a factor-of-10 cost/performance advantage. We can only maintain this advantage by skimping on the communication.

Application	Communication Overhead
Systolic	<10%
⋮	⋮
Actors	Embarassingly high

COMMUNICATION CHIP!!