Caltech NNCP Project

(Nearest Neighbor Concurrent Processor)

Erik DeBenedictis

The NNCP project has two (almost) separate parts:

* Social experiment in concurrent programming
* Architectural experiment in machine design.

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.

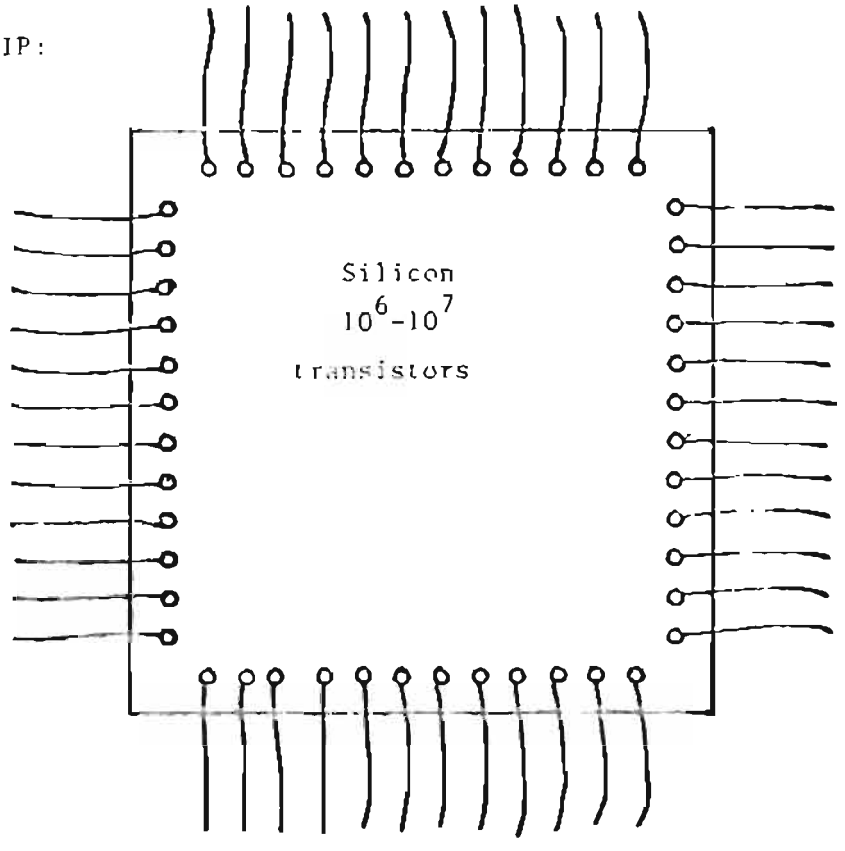
Cost: Different programming style.

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

PHYSICAL MODEL

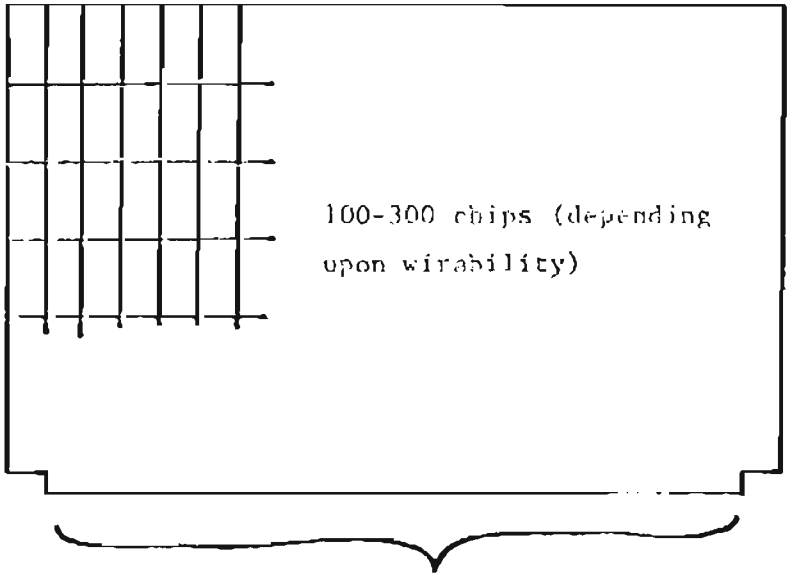
1990

CHIP:

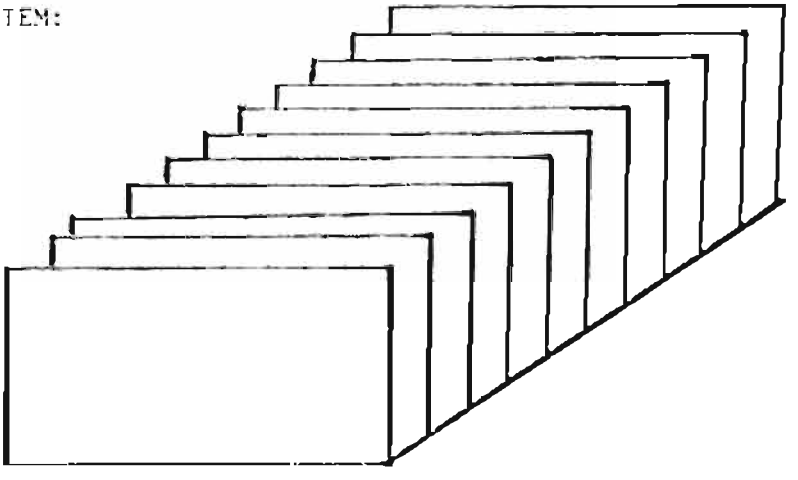


16-40 connections (maybe more?)

BOARD:


100-300 chips (depending upon wirability)
edge connector, 20"@20 connections/ inch = 400 connections

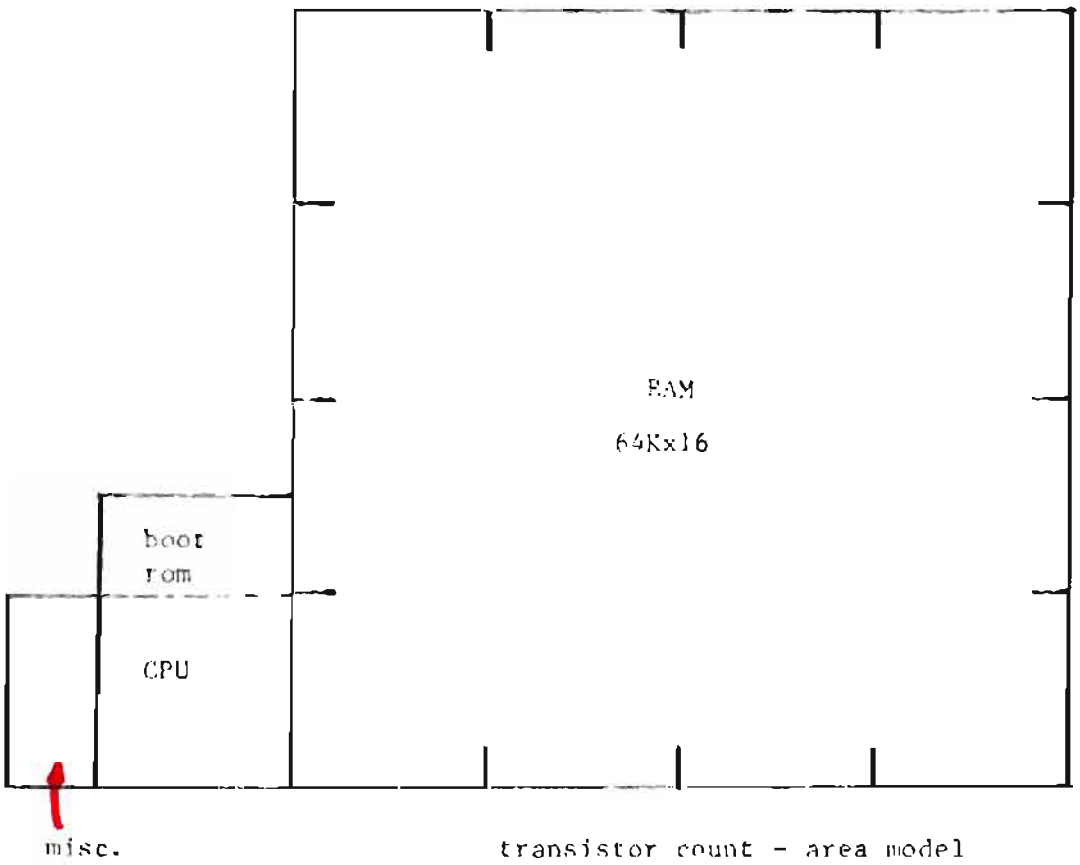
SYSTEM:



100 boards

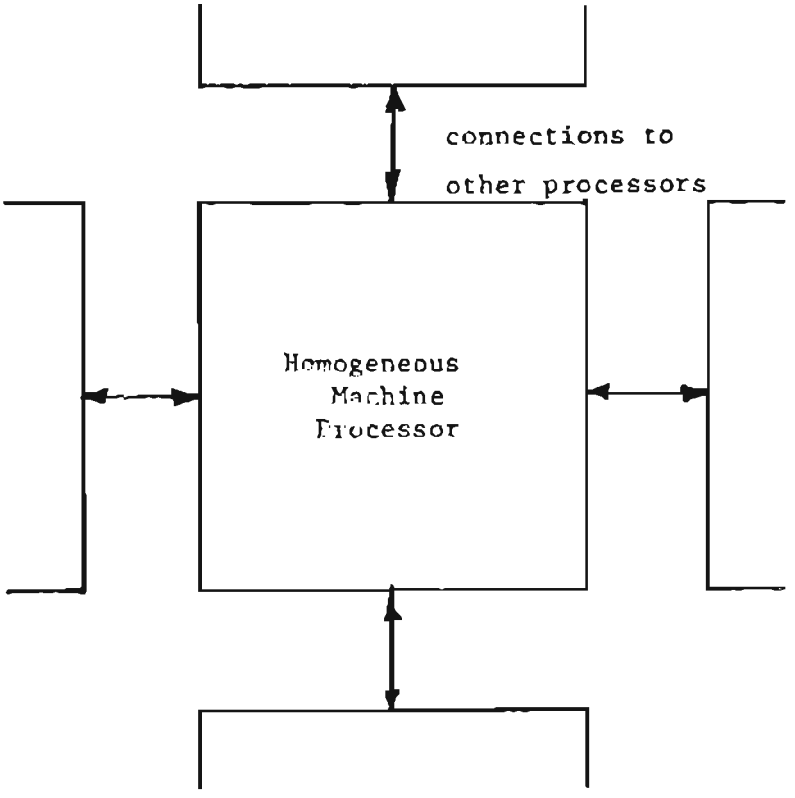
I postulates: Total size (CPU+Memory) less than 10^6-10^7 transistors

Number of pins less than 40 (64?)



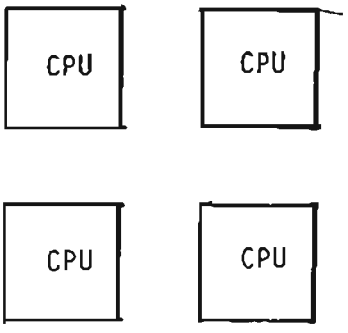
transistor count - area model

Homogeneous System Design

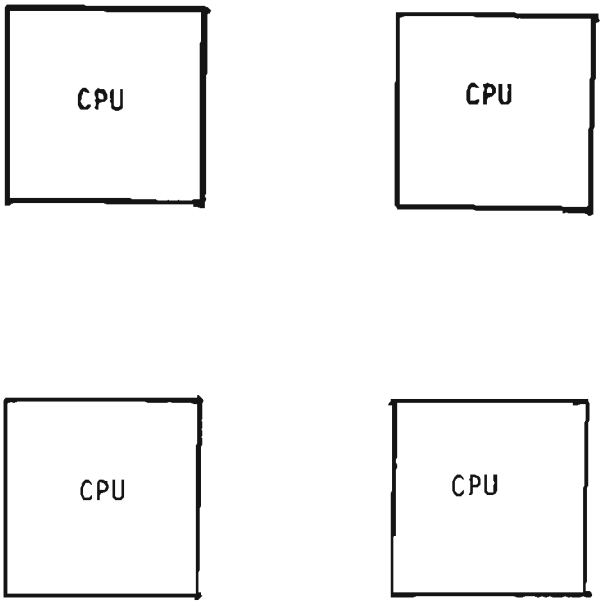


Multiple Von Newman vs Non Von Newman Concurrent Architecture

Multiple Von Newman Architecture:

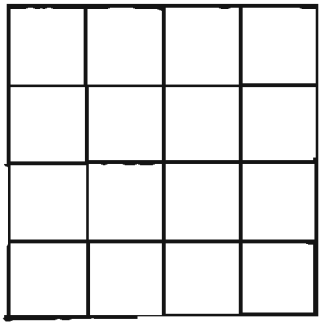


four Von Newman machines

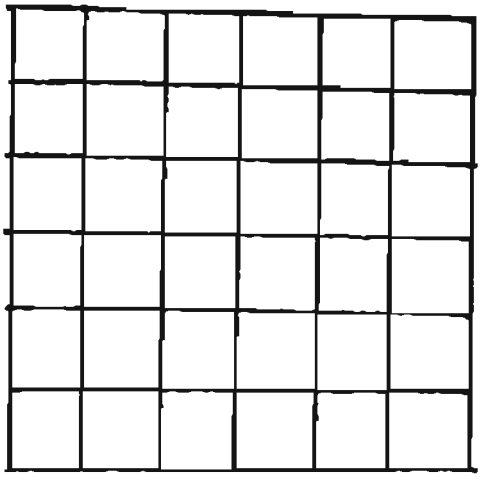


four larger Von Newman machines

Non Von Newman Concurrent Architecture:



16 processors

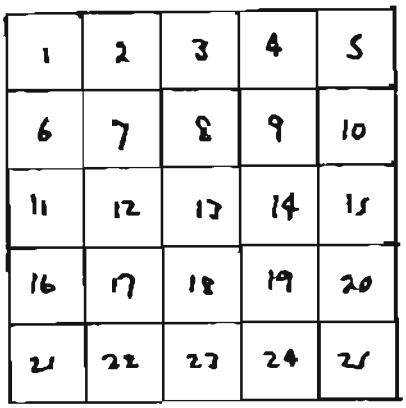


36 processors

Spatial Decomposition of Problems

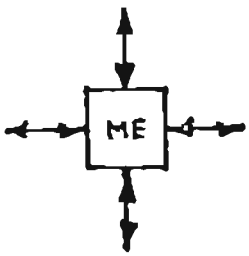
PDE:

conventional



update lattice points sequentially

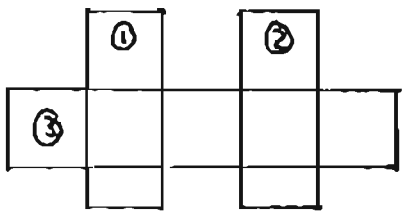
spatial



update yourself

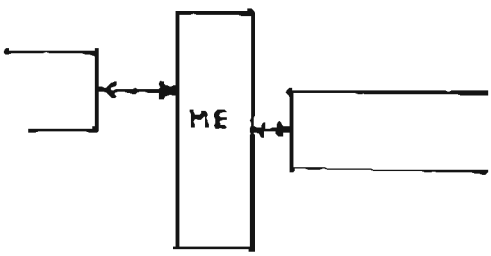
DRC:

conventional



for every pair of polygons
check all design rules

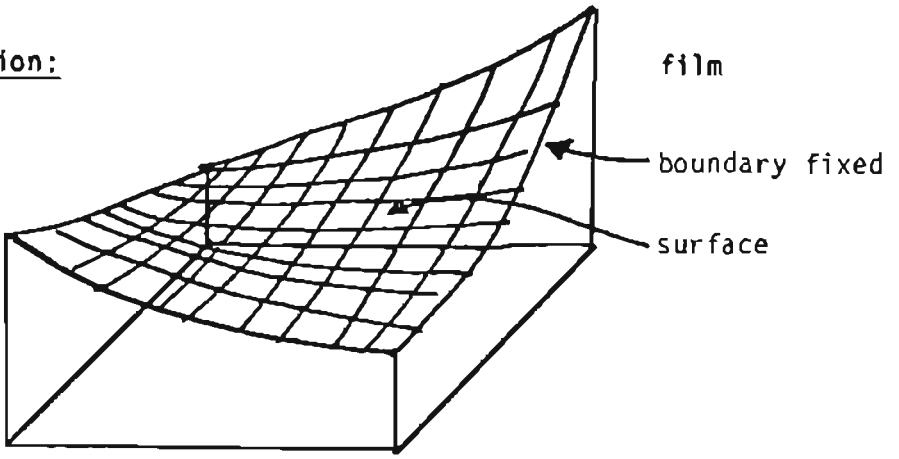
spatial



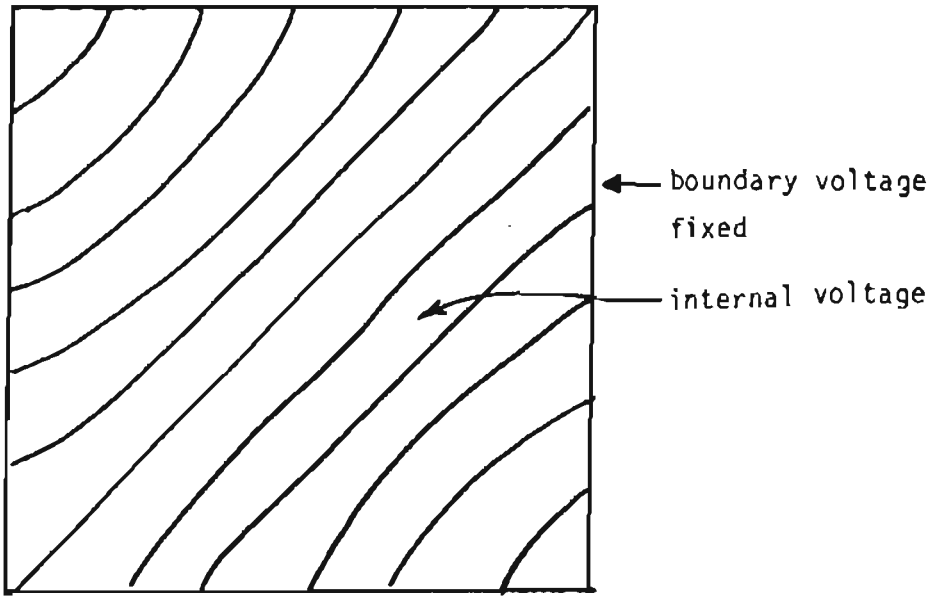
(1) sort self
(2) make sure neighbors are far enough away

Partial Differential Equations

Laplace's Equation:



surface described by: height = f(x,y)

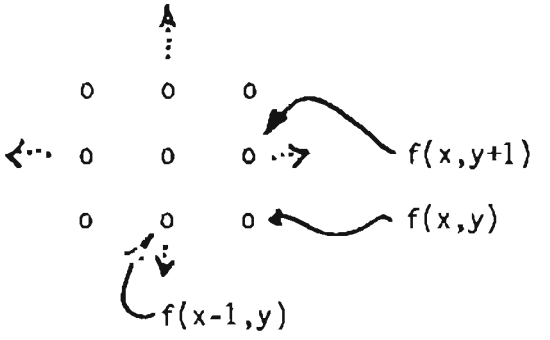


voltage is: v=f(x,y)

Equation:

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right) f = 0$$

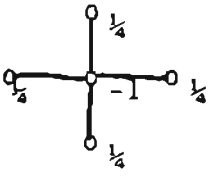
Grid Point Approximation:



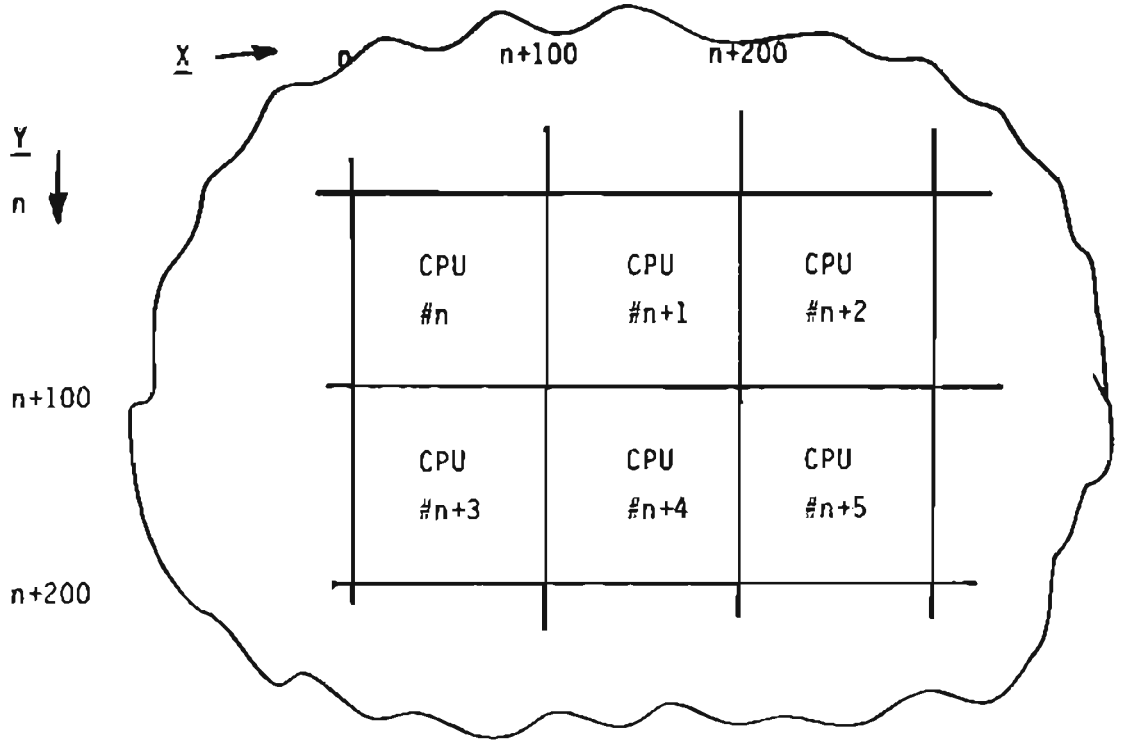
First Order Solution:

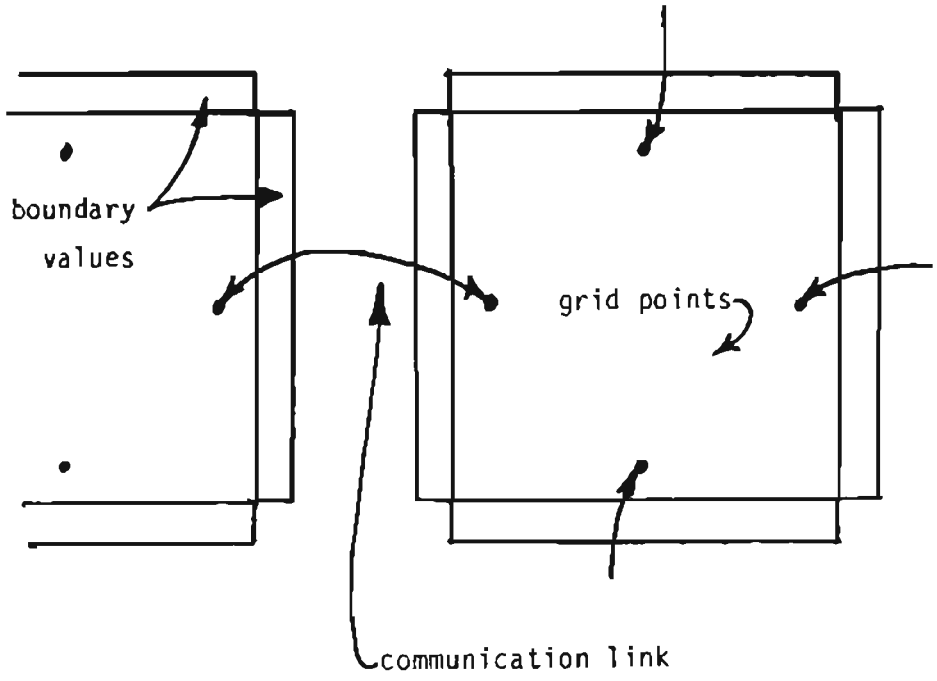
$$f(x,y) = \tfrac{1}{4}\left[f(x,y-1) + f(x-1,y) + f(x,y+1) + f(x+1,y)\right]$$

'Stencil':



Multiprocessor Solution:





Iteration:

1) Transfer boundary values to neighbors.
2) Compute new values for grid points.

Laplace's Equation: Problem Characteristics

1) The number of grid points per processor must be chosen to fit in the available memory. Example:

   100K bytes memory / 4 bytes per point ⇒ 25,000 points

   try grid of 150 x 150 = 22,500 points

2) Time per iteration is determined by compute time + communication time. Example:

   Compute Time: 150 x 150 x time per grid point
   150 x 150 x 150 uS ⇒ 3 seconds

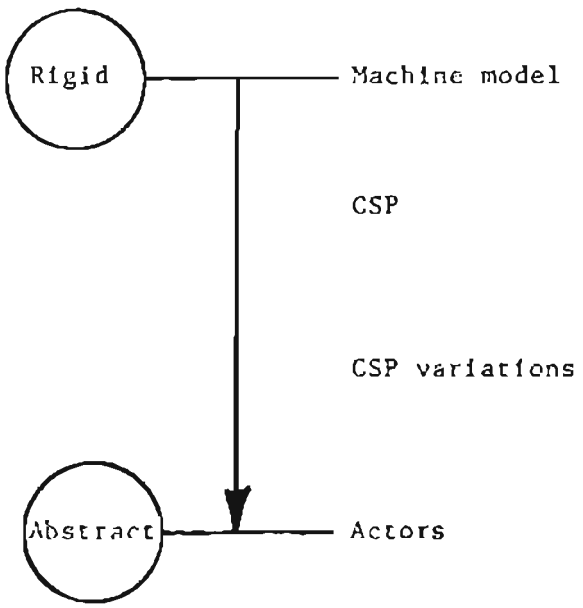
   Communication Time: 4 x 150 x time per point
   4 x 150 x 50 uS ⇒ 30 mS

   Total Time: 3.03 seconds per iteration

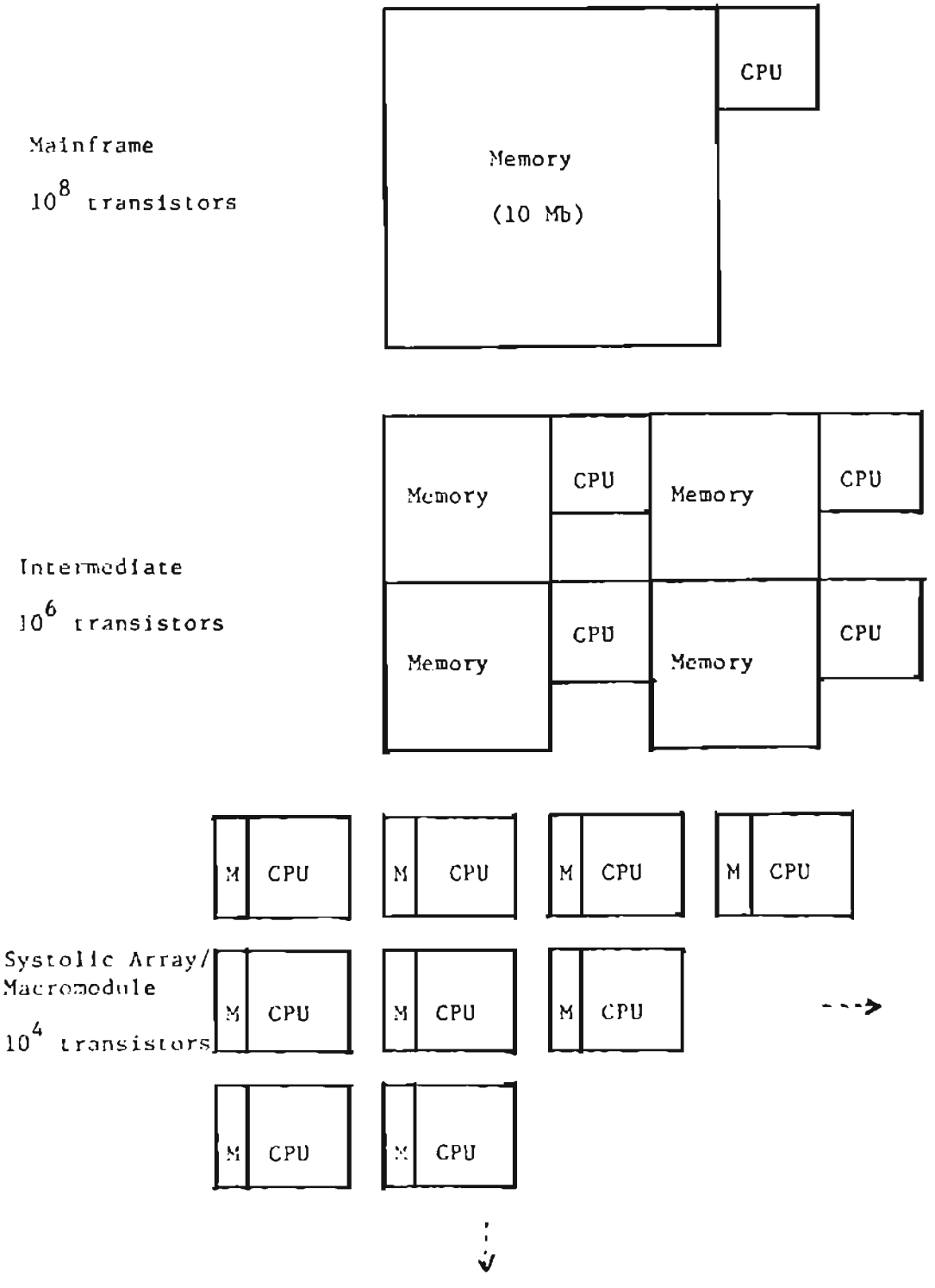
Limitations

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

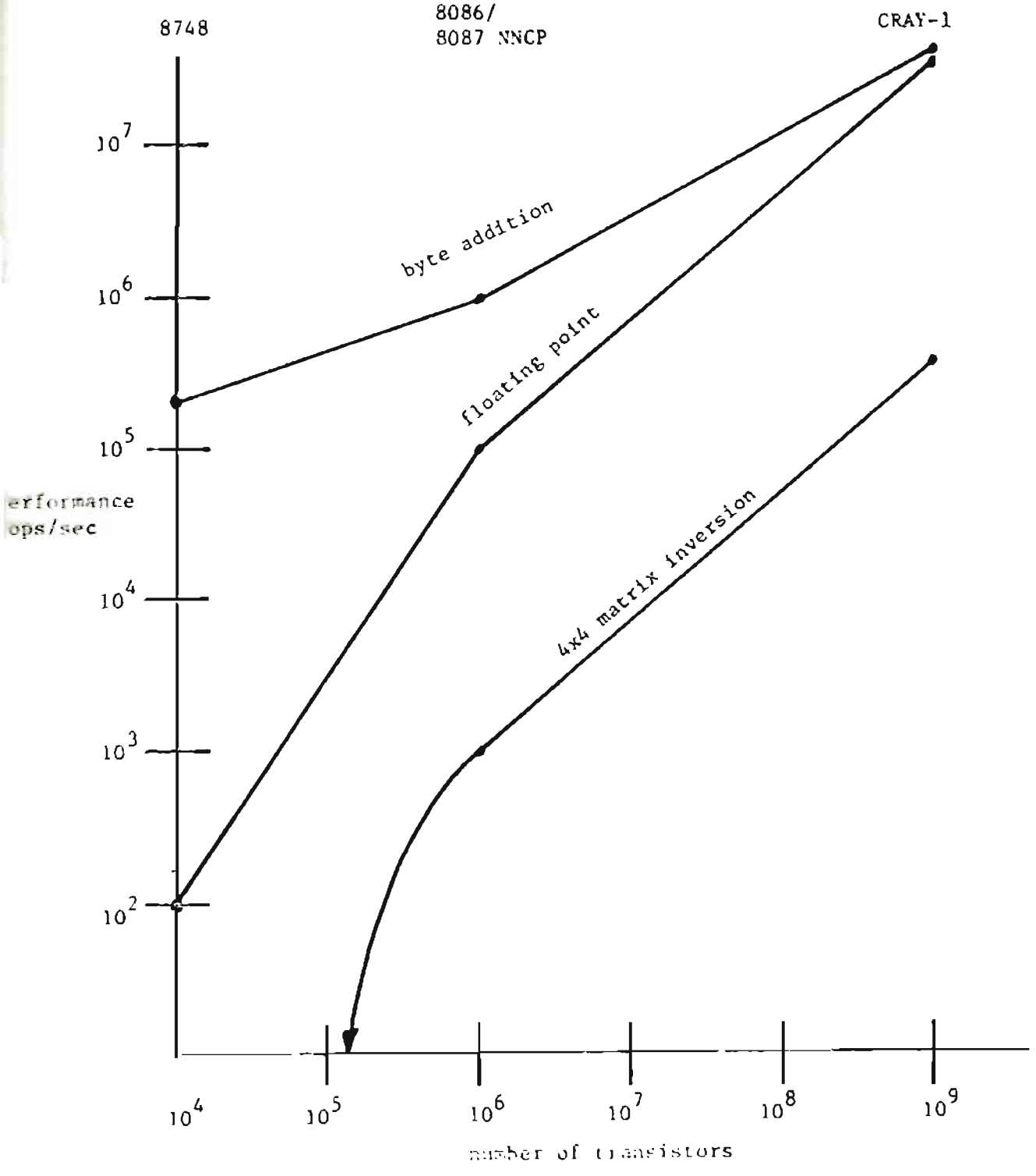
(mutually exclusive)



Processor Characteristics



System Performance

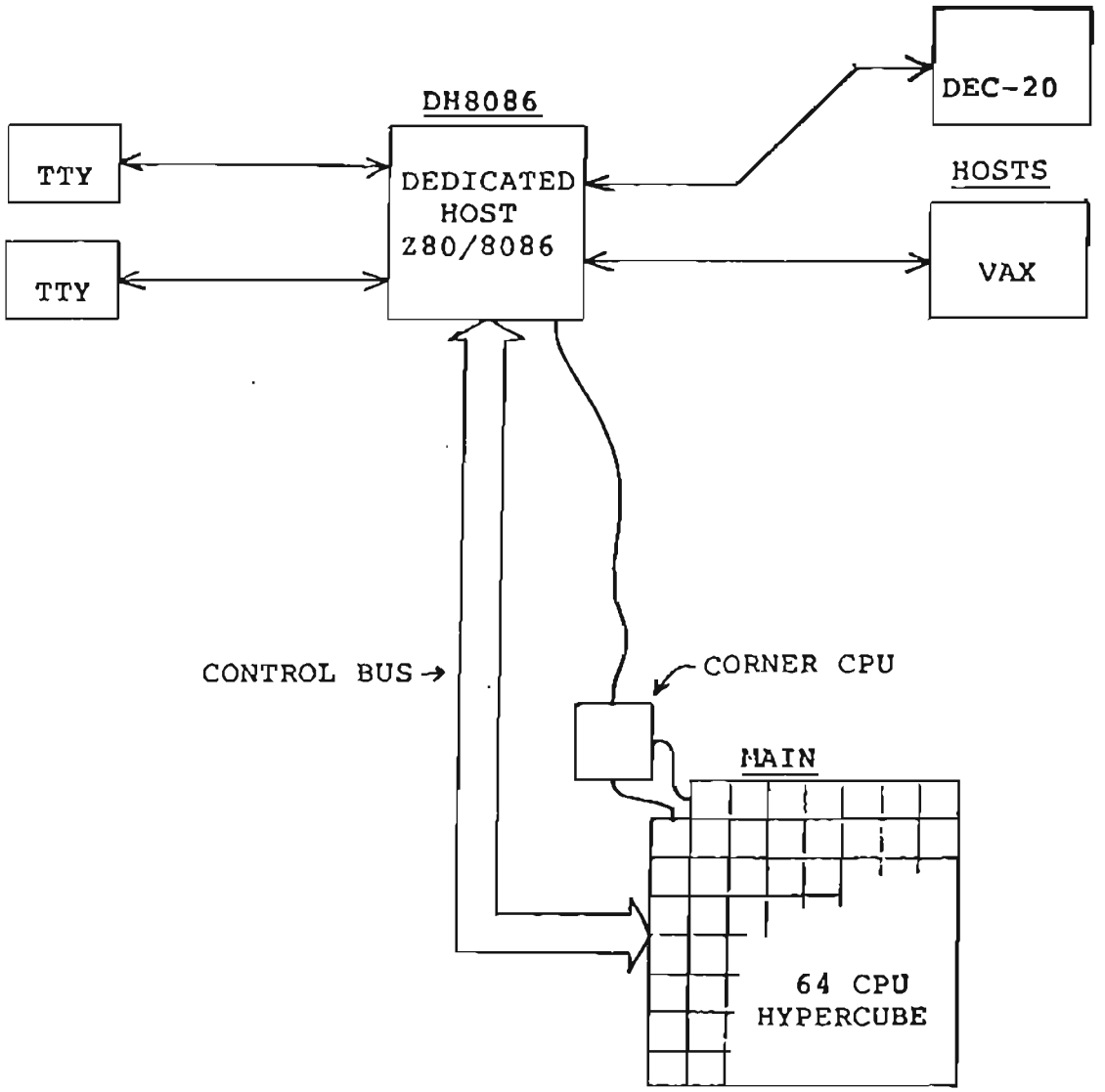


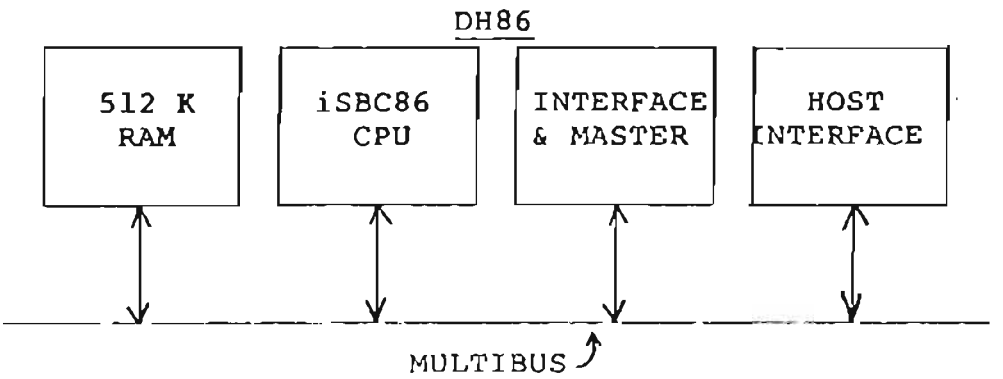
Comparison of Three Types of Processors

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

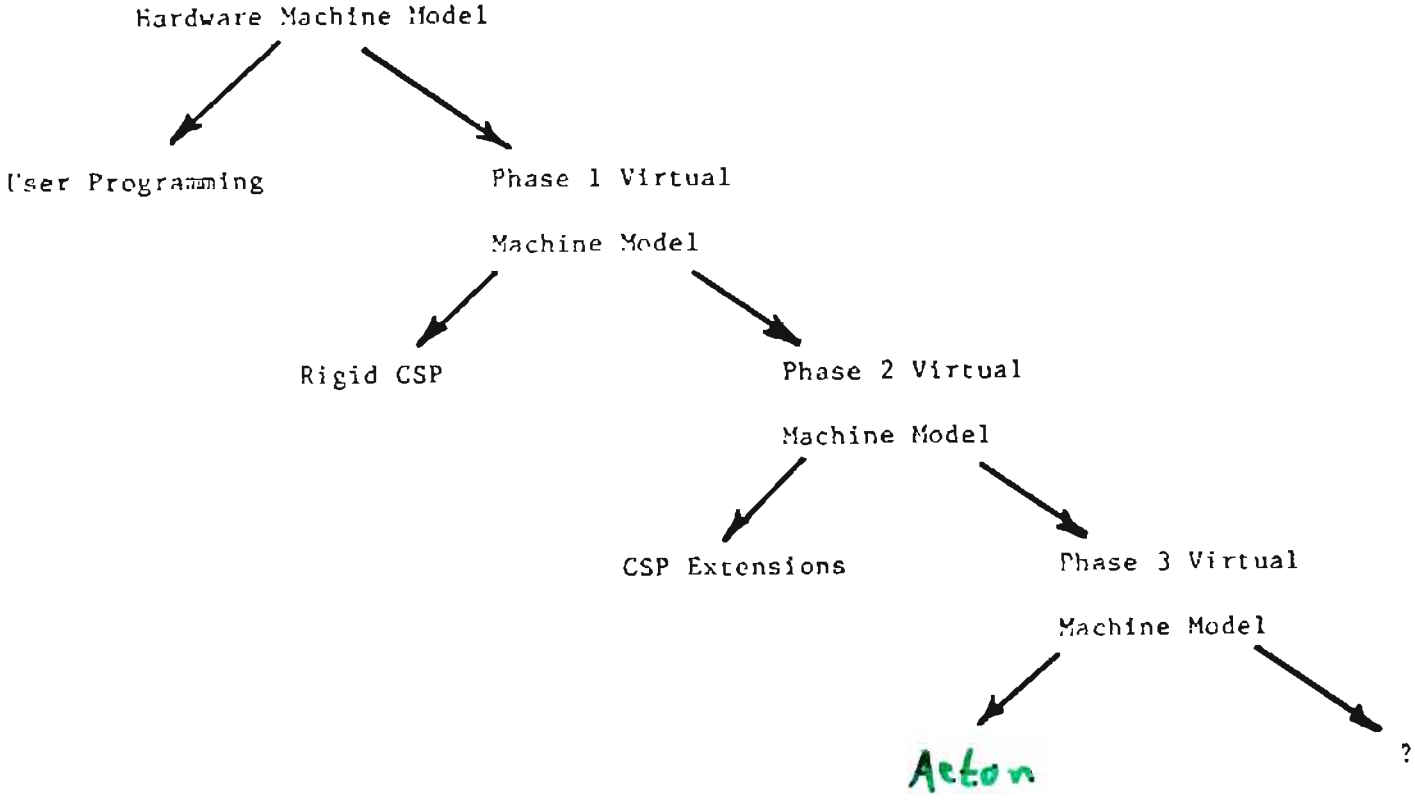
processor	sequential steps	time/step	total time	size	#chips	cost	cost*time
mainframe	10^9	1 uS	10^3 sec	1 Mb		$200K	$2x10^8$
10x10 NNCP	10^7	10 uS	100 sec	100 bds		$200K	$2x10^7$
1000x1000 systolic	10^3	100 uS	.1 sec	10^6 10^4	chips/ boards	$20M	$2x10^6$

HOMOGENEOUS MACHINE

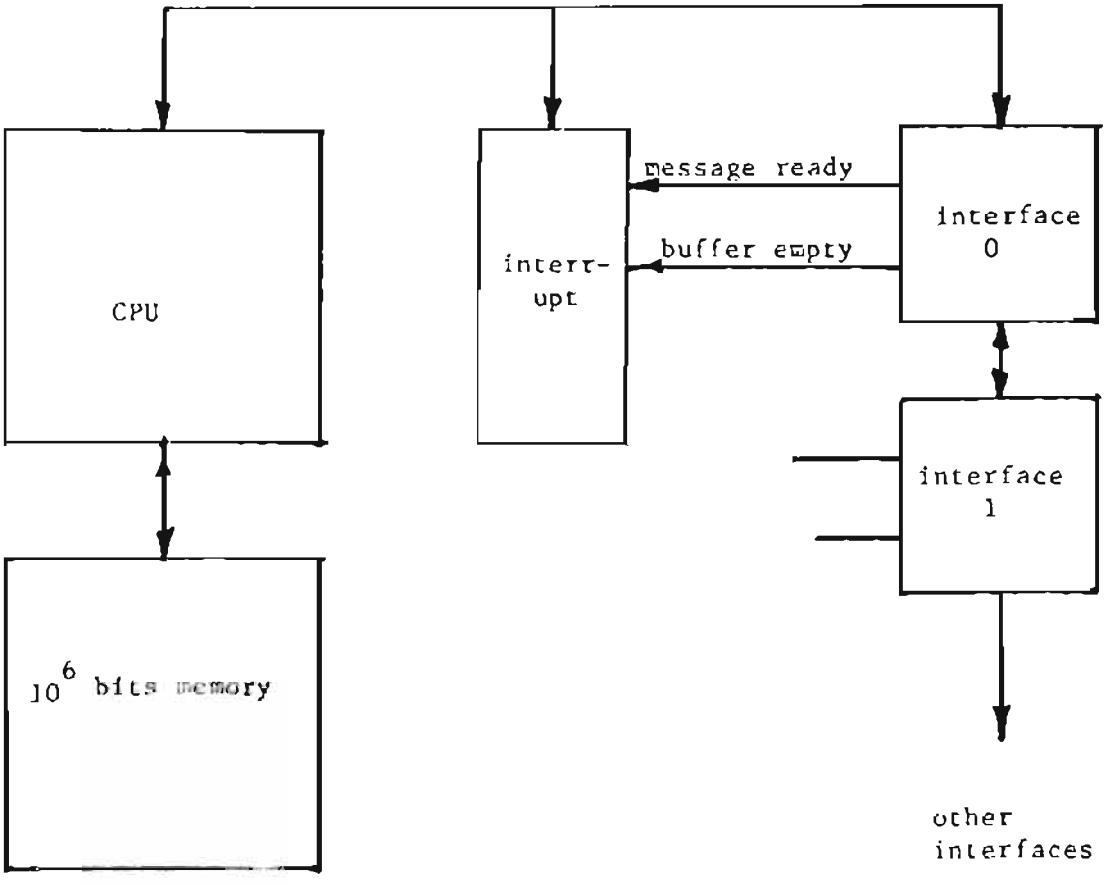




Software Development Plan

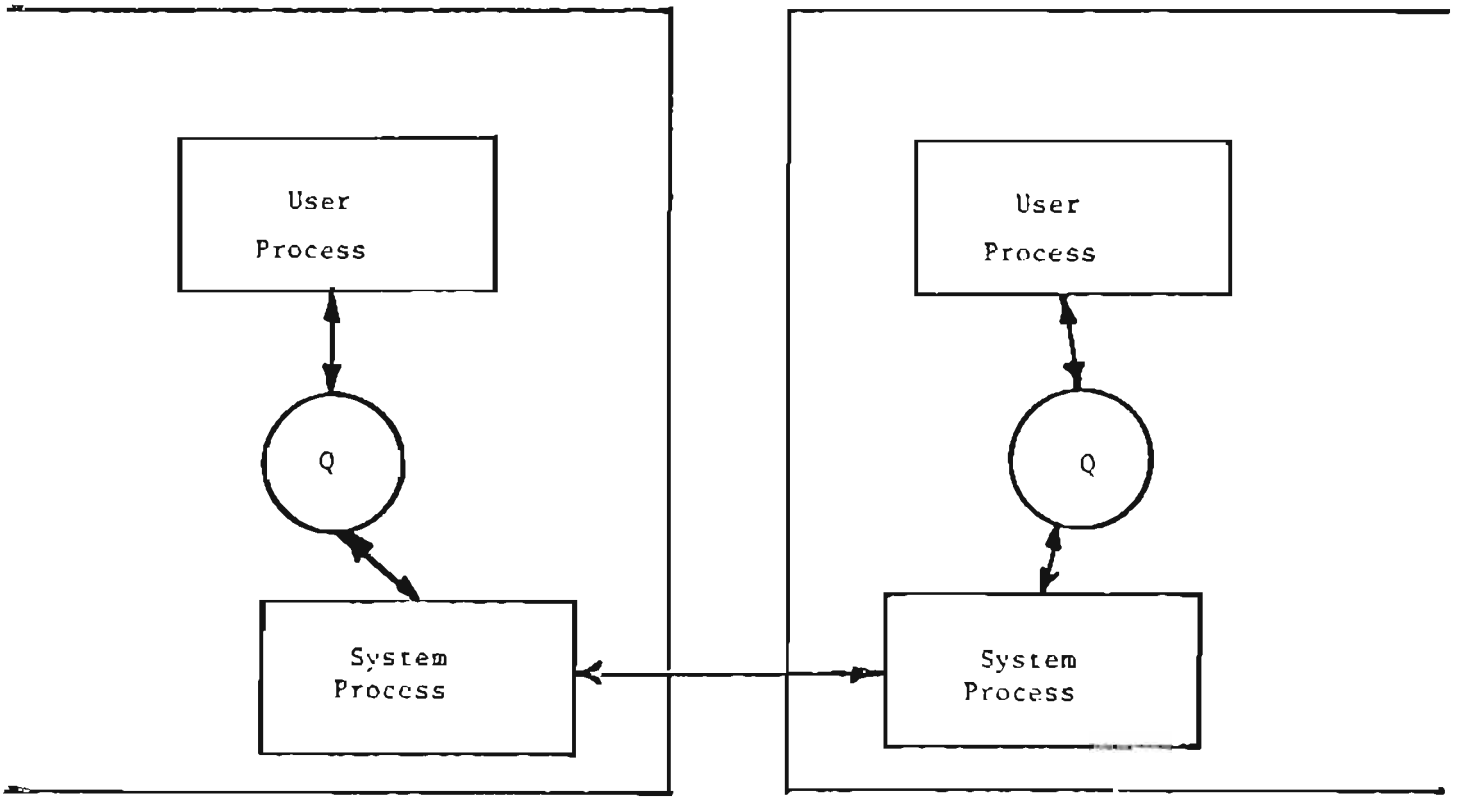


Hardware Machine Model



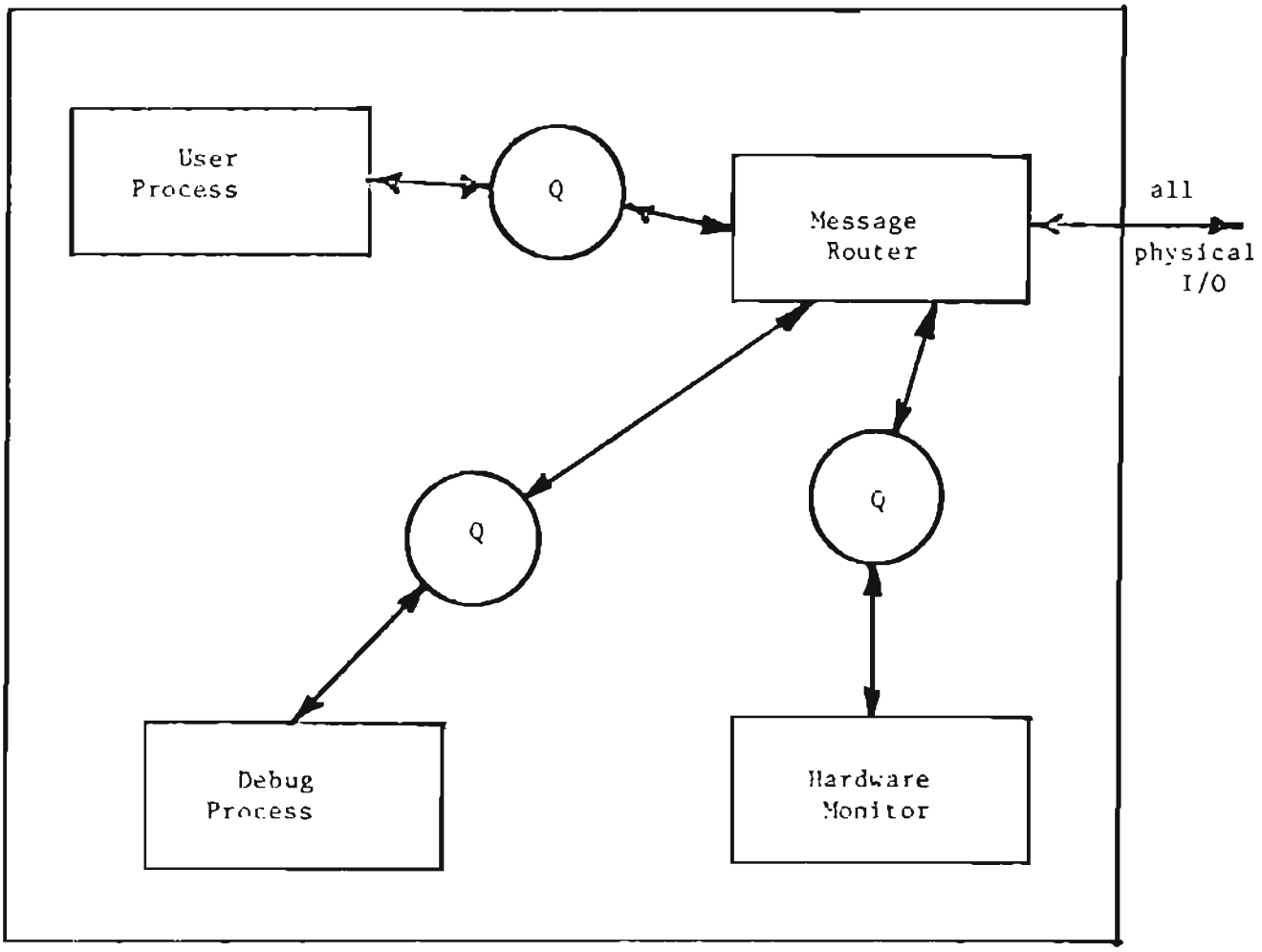
Phase 3 Machine Model

Features: Dynamic Process Movement
Global Pointers



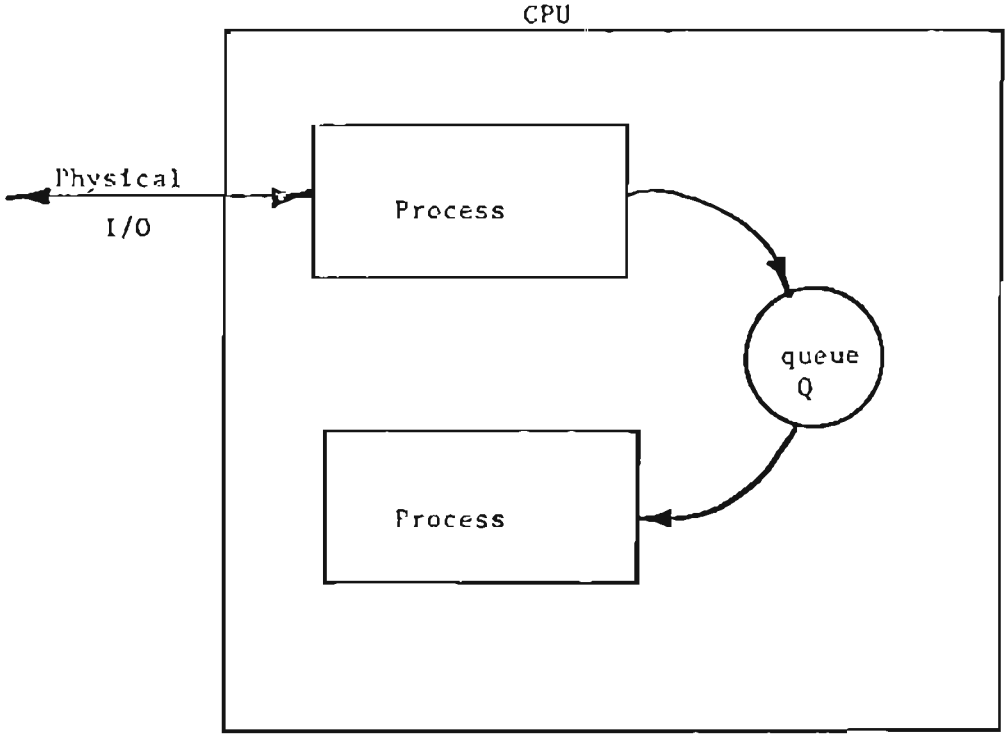
Phase 2 Machine Model

Features: Message Routing
Debug Aids
Diagnostics



Phase 1 Machine Model

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



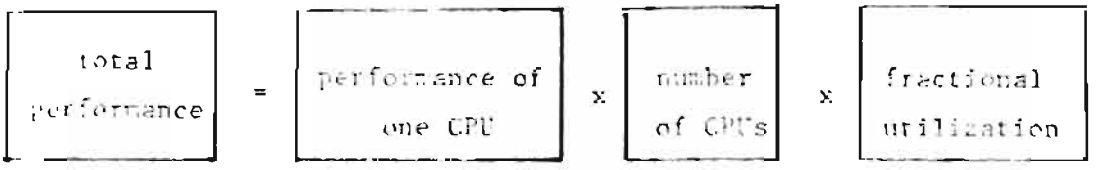
Caltech 8086 Homogeneous Machine

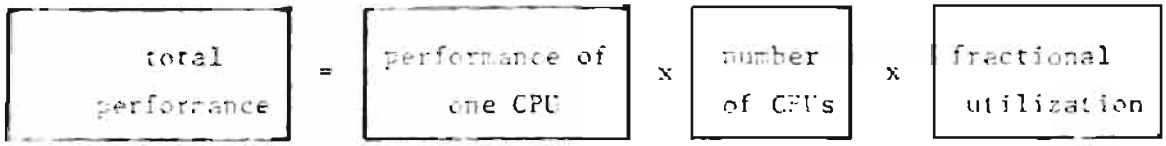
* 78 chips
* 2^{17} bytes storage
* 1.25×10^{6} transistors
* 30 us multiply/add (double floating)**
* 30 us communication time (64 bit message)

**(using 5 MHz clock, half for 10 MHz clock)

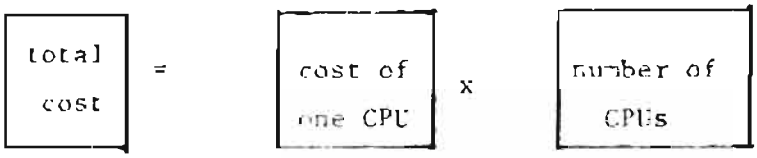
Potential Performance

(normalized to VAX 11/780)





6.4 VAX 1/8 VAX 64 0.8



$54k $1000 64

10:1 cost/performance improvement over VAX (PDE problems)

20:1 with (speculative) full speed parts

Number of CPUs

64

Fractional Utilization

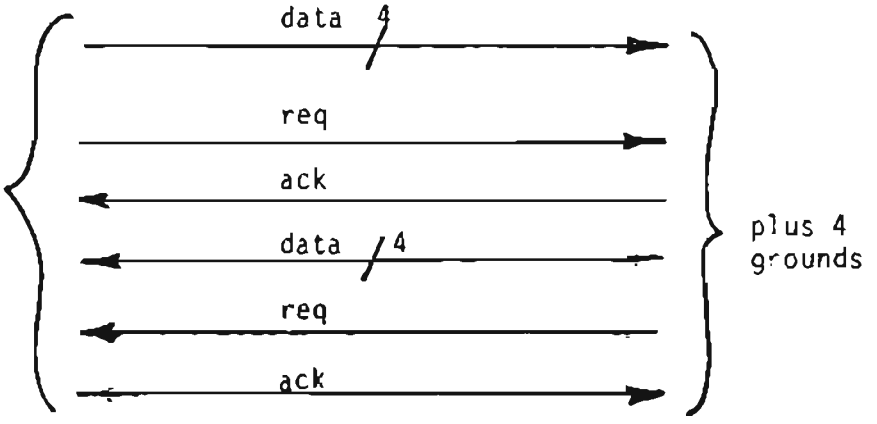
Most PDE problems: greater than 0.8

Other problems: unknown but probably less

Communications

A Main:

1) Fully asynchronous
2) 4 bit serial
3) 64 bit software/hardware message format
4) Flow control (1 message queue)
5) Full duplex:



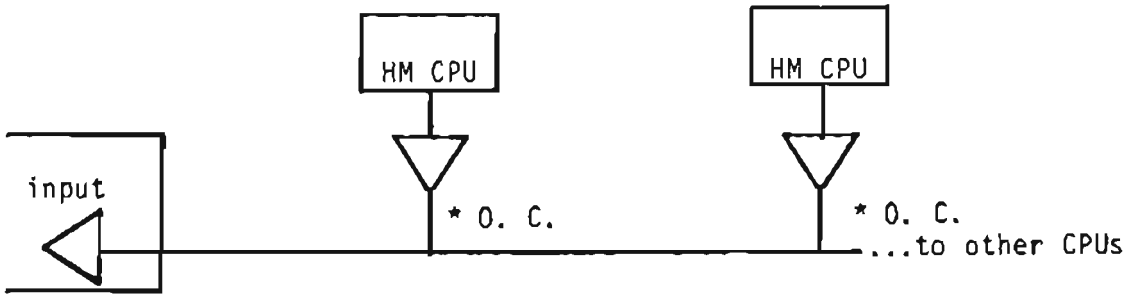
6) Interrupt or poll conditions:
   * receive buffer #n full
   * transmit buffer #n empty
7) Input/Output of message as 4 word IO operations

8 Global:

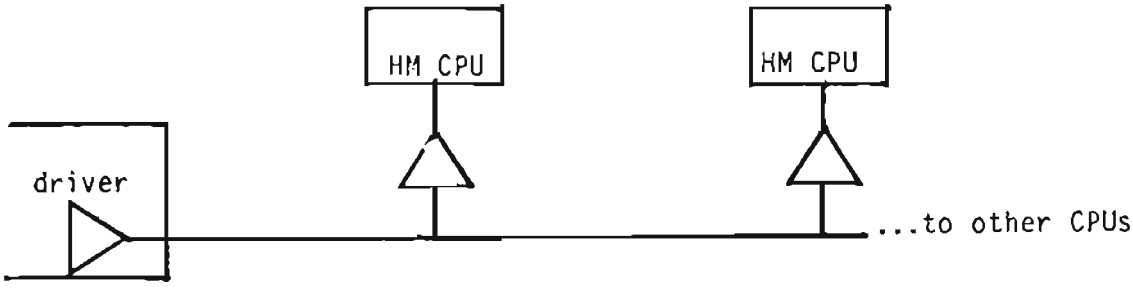
Global bus: (power)
Clock
Reset
Refresh Interrupt
Global Open Collector (0-3)
Global Data (0-3)

Ram Refresh: Non maskable interrupt signal is asserted every 4mS to all processors to invoke a routine that refreshes dynamic RAM.

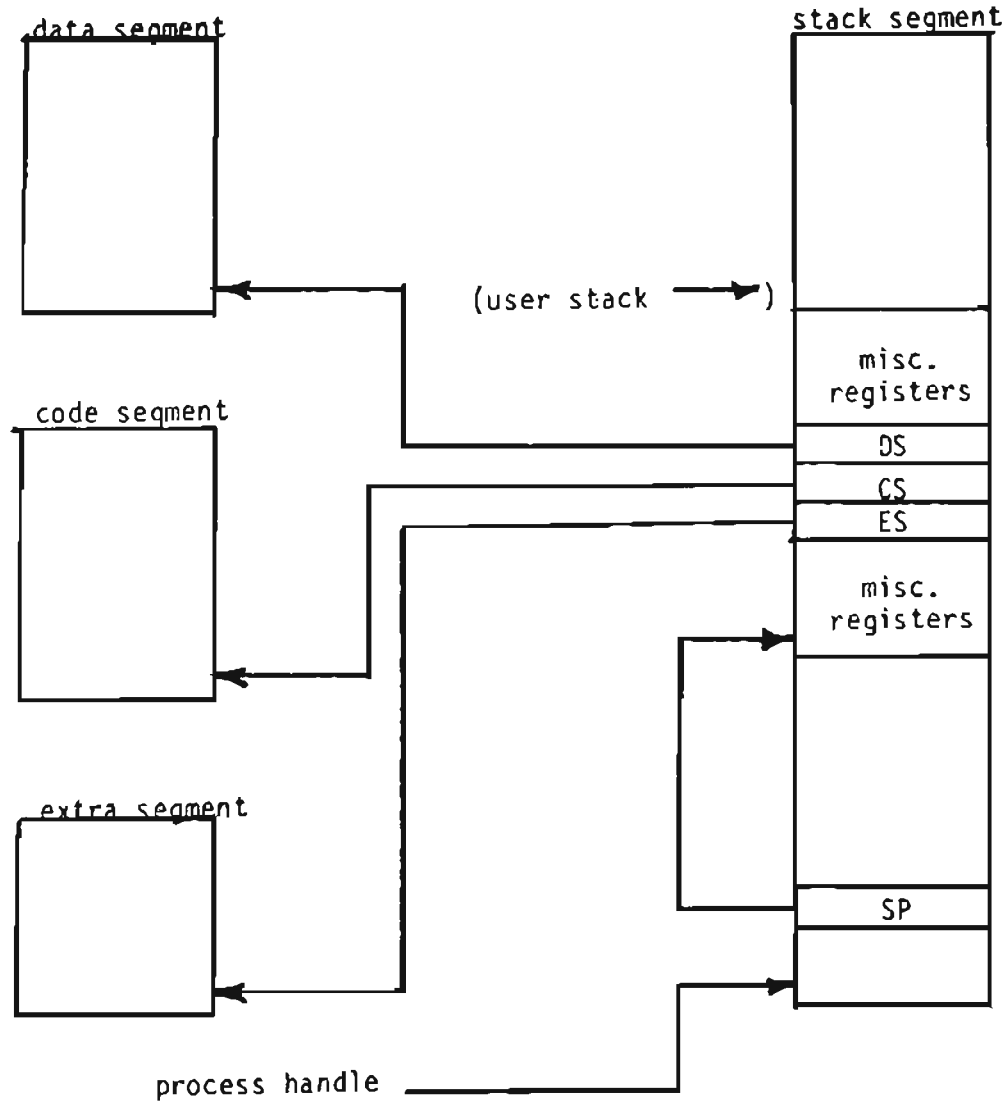
Global Open Collector:



Global Data:



8086 Process Control



Segment registers are either in the CPU or saved in a static process record. The segment registers and segments form a set of objects and pointers that can be relocated or garbage collected.