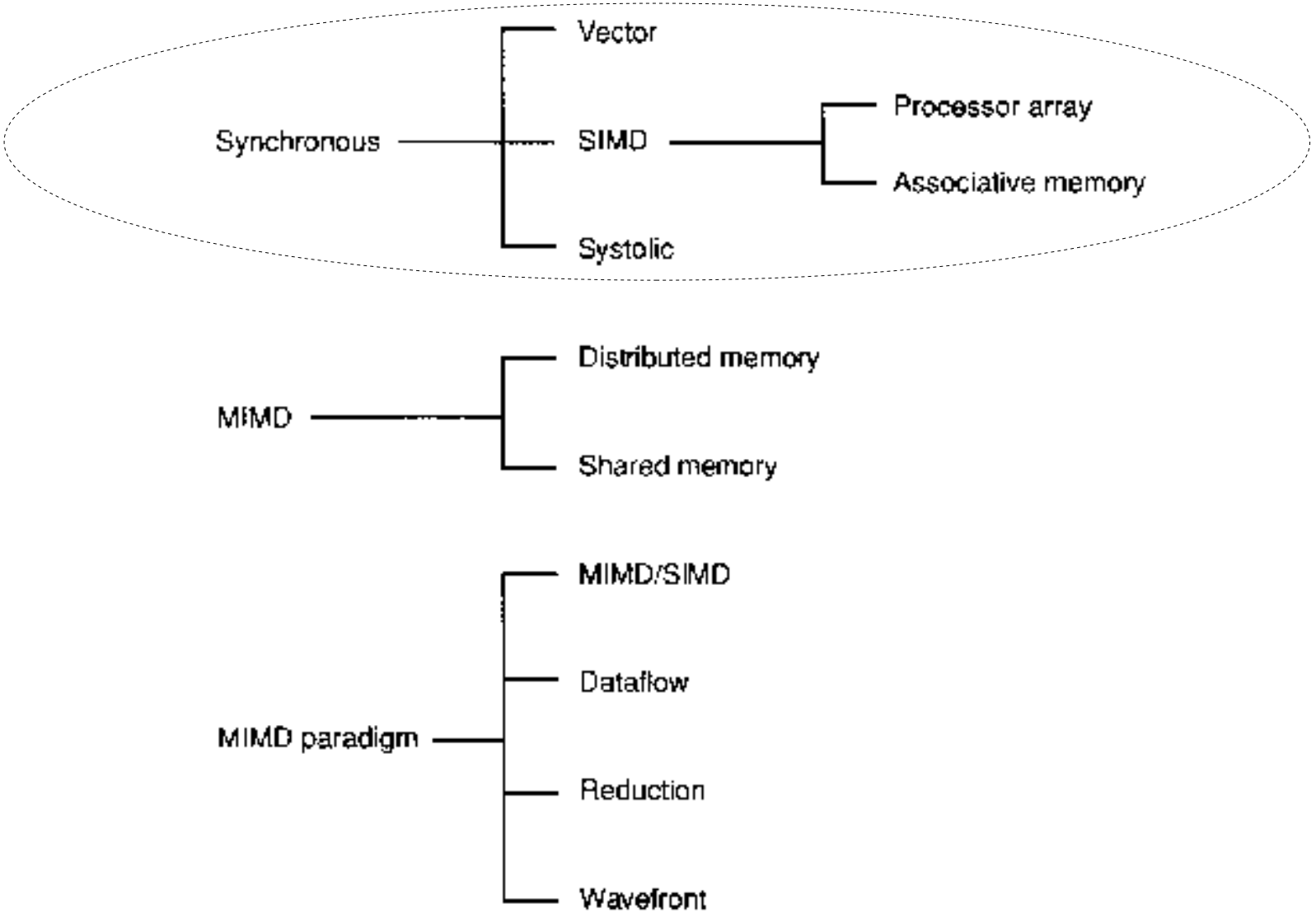
"A Survey of Parallel Computer Architectures"

Ralph Duncan

Journal of Parallel Computer Architectures, Vol. 23, Issue 2, pages 5-16, 1990.

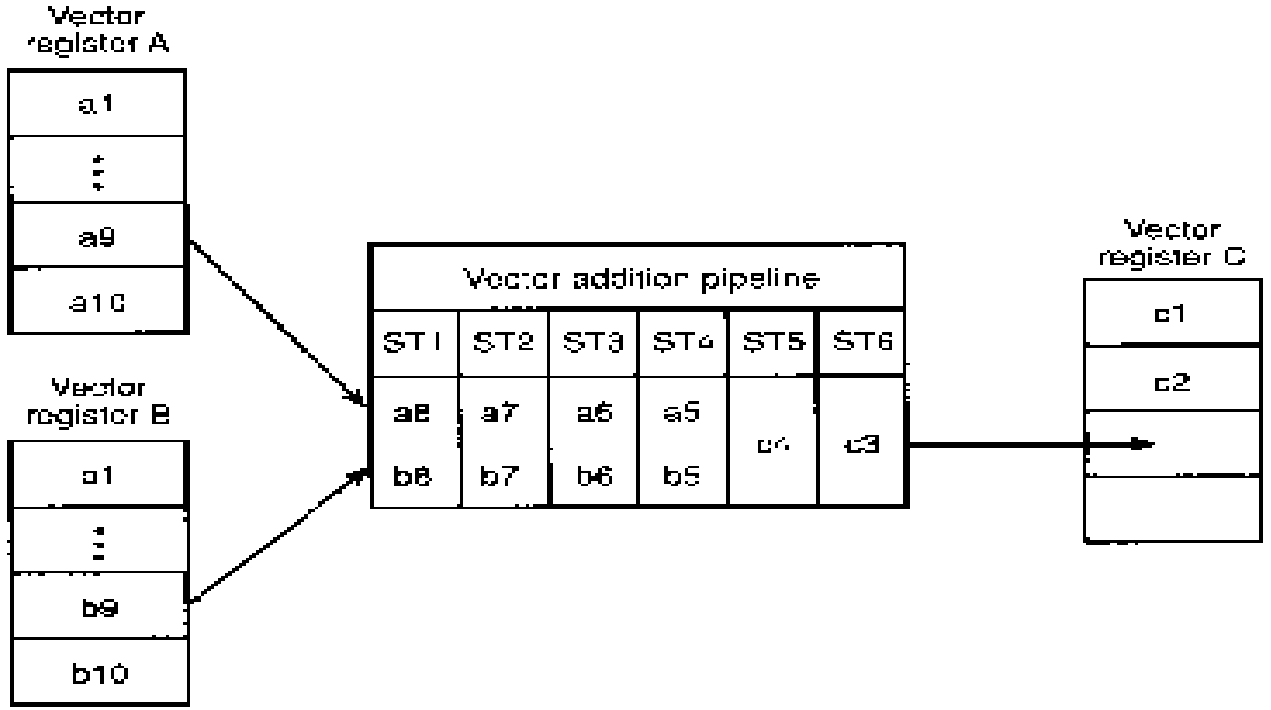
Presentation and slides created by Brad Peterson, bradpeterson@gmail.com

Various architectures organized:



Vector computers

- Specially built hardware which works with each element of a vector or vectors, sends it through a pipeline.

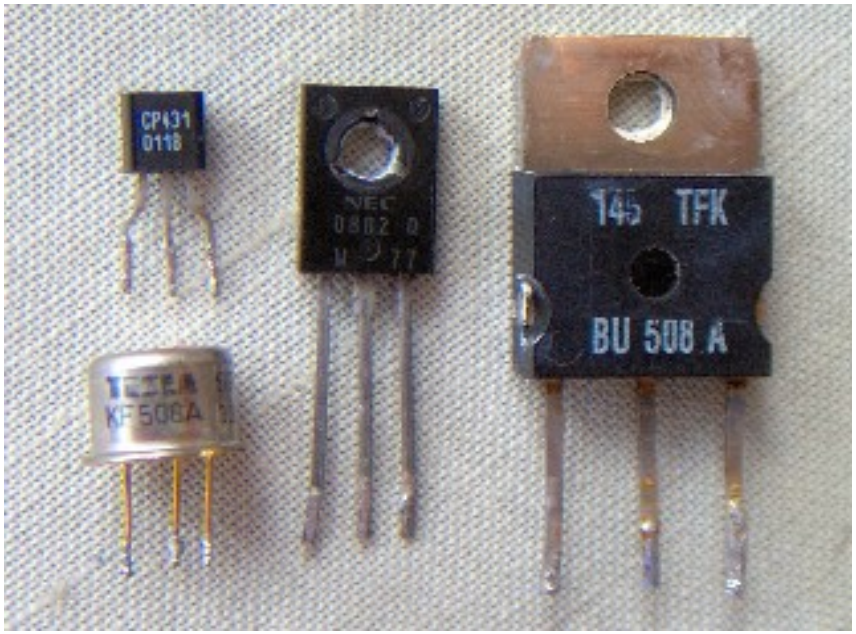


- Seems similar to modern day CPUs. Not parallel.
- Later models added more and made this MIMD and shared memory.

SIMD – Processor Array

- Easier to understand. Classic SIMD machine. A bunch of simple computing units.
- One instruction unit to control all the computing units in unison.
- Works great for working with data that has vectors. The paper mentions image processing and nuclear energy modeling.

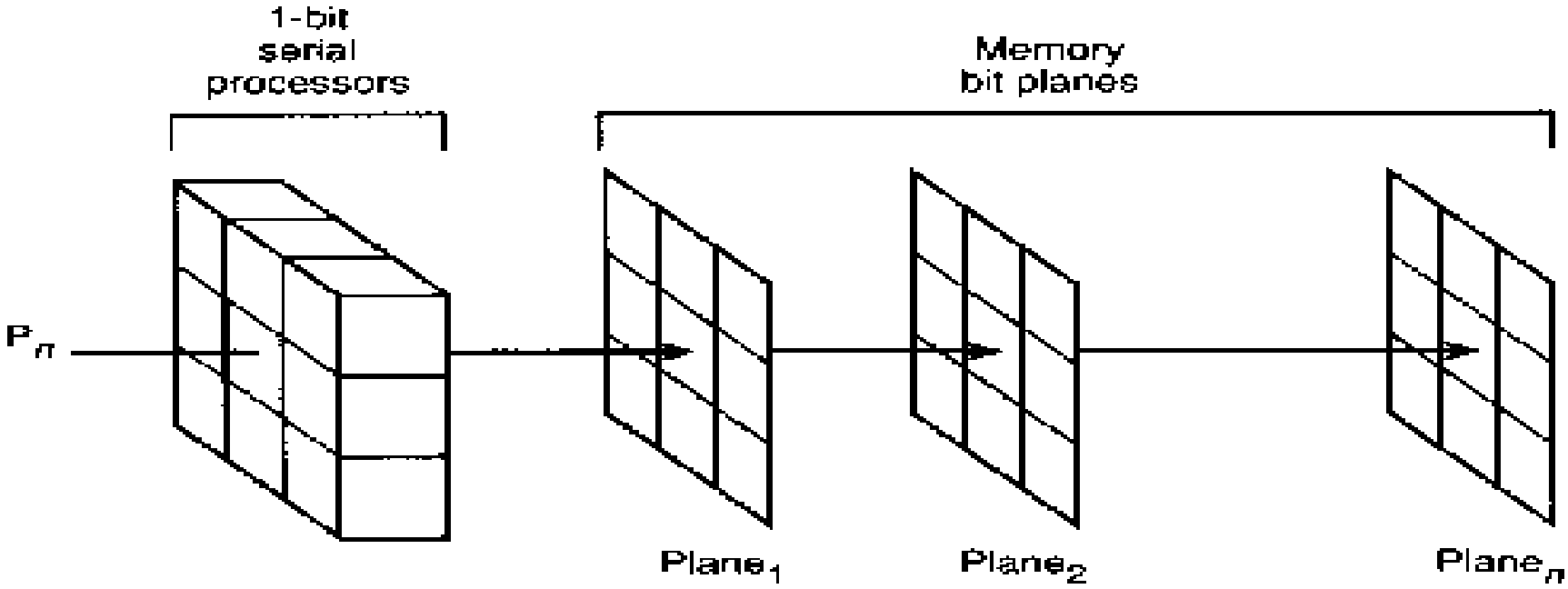
When men were men



- Concepts like integrated circuits, ALUs, multi-bit processors, RAM, weren't at all like what we think of today. It was often just transistors and soldering wire.
- So, how do you build supercomputers with transistors and soldiering irons?

SIMD – Processor Array Variant

- Treat your data as containing 1 bit planes.
- Have a bunch of 1 bit processors (or better called, processing element), each working on its own plane.



SIMD – Associative Memory

- Because this was the 1960s, even the concept of RAM as we think of it today wasn't common.
- These computers often used associative memory (or content addressable memory, or CAM for short).
- RAM is about finding memory fast based on an address. But if you want to search for all matches of a piece of data, that can be rather sequential.
- CAM is about finding memory fast based on the content in the memory. Provide it a short bit phrase and in one cycle it will return a bunch of data associated with that.
- One-bit processors and content addressable memory (CAM) often go together. But it doesn't scale well!

Associative memory (contd.)

- This is effectively searching multiple memory words and signifying which words match the comparison bits. Great for searching.

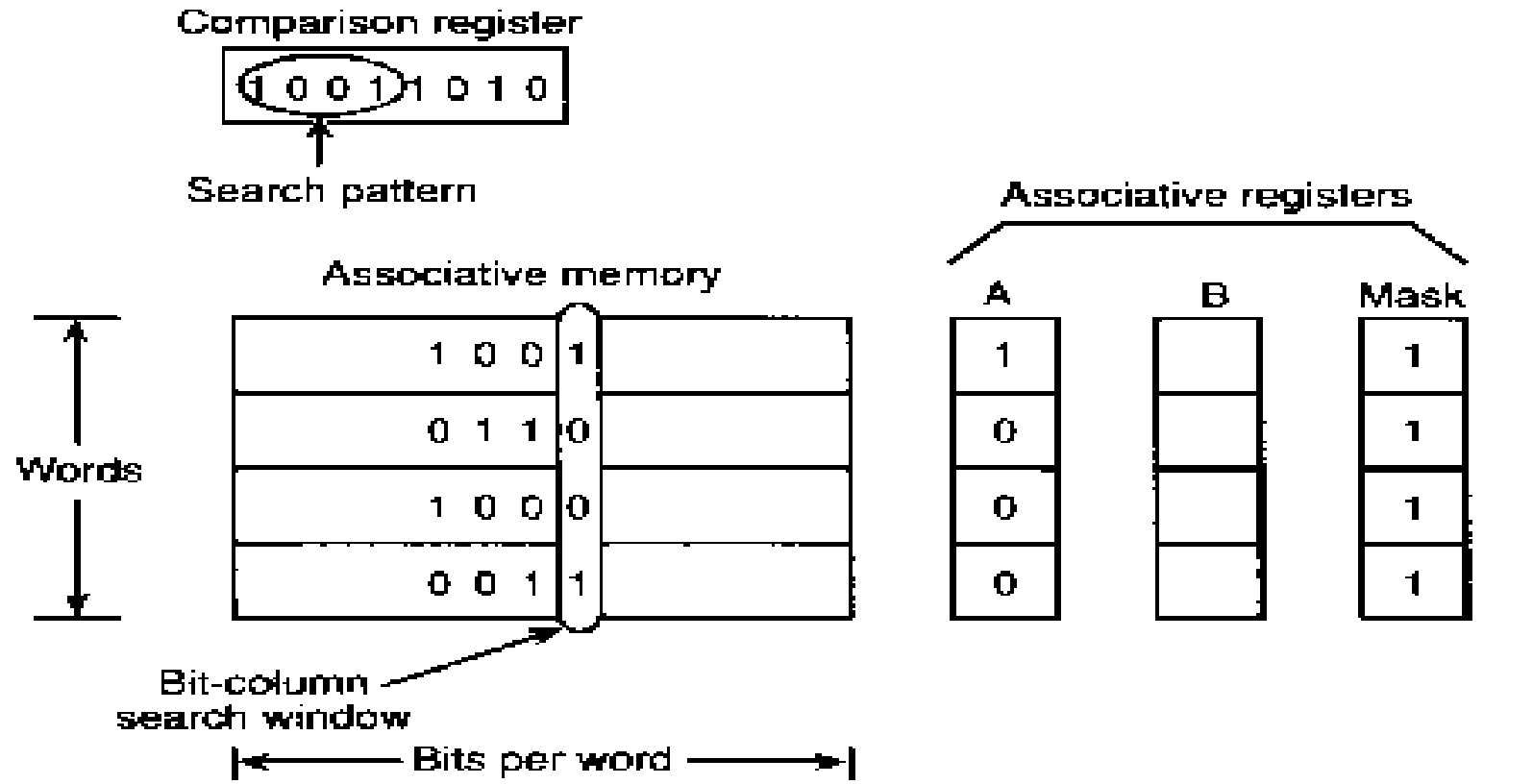


Figure 6. Associative memory comparison operation.

Associative memory (contd.)

- A bitwise OR operation (good for merging data).

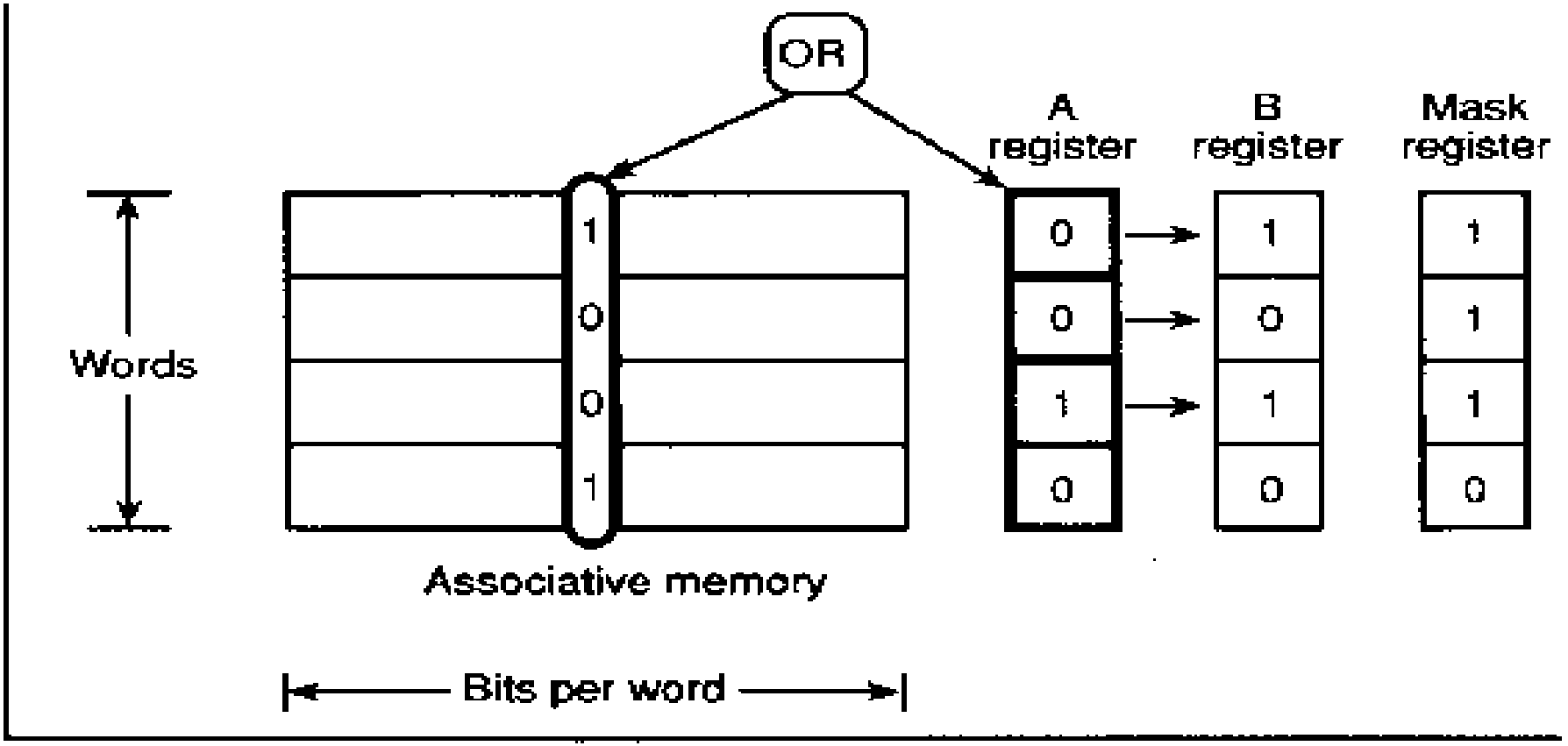


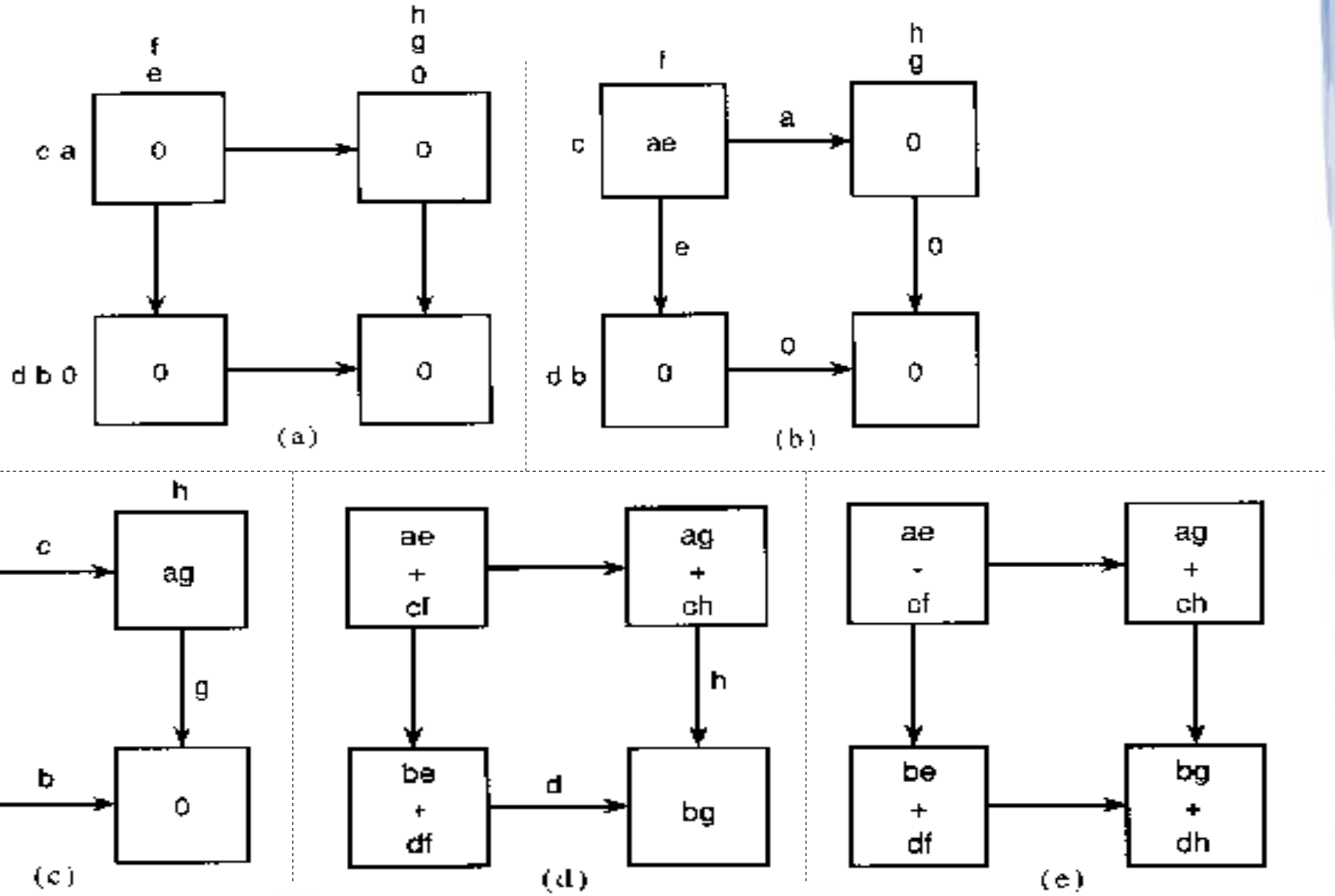
Figure 7. Associative memory logical OR operation.

Systolic architectures

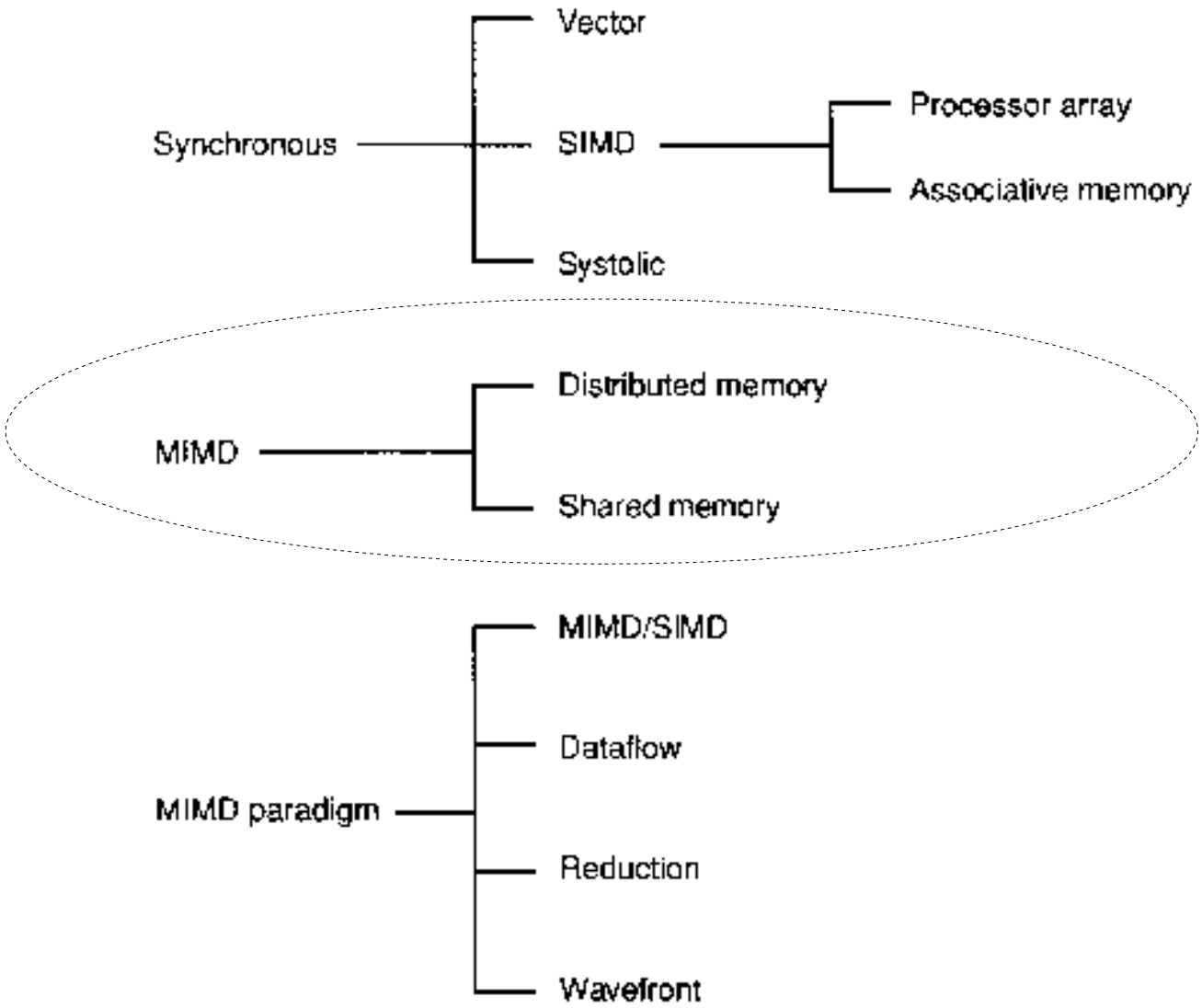
- Overall goal, I/O is costly. Keep the data in processors as long as possible.
- A rather unusual and specialized form of parallel computing.
- Almost like a 2-D pipeline.
- Meant for specialized applications.
- Modern day architectures just use fast cache, shared memory, or registers to avoid I/O.
- I highly doubt we'll see systolic architectures make a comeback.

Systolic computer example (continued)

- Compute $\begin{bmatrix} a & c \\ b & d \end{bmatrix} \cdot \begin{bmatrix} e & g \\ f & h \end{bmatrix} = \begin{bmatrix} ae+cf & ag+ch \\ be+df & bg+dh \end{bmatrix}$



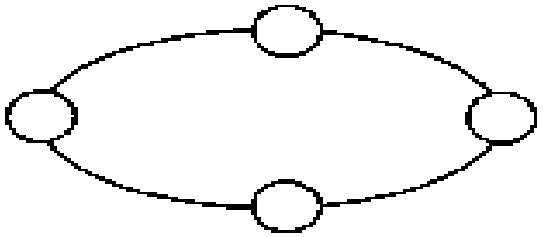
MIMD


Synchronous
Vector
SIMD
Processor array
Associative memory
Systolic
MIMD
Distributed memory
Shared memory
MIMD paradigm
MIMD/SIMD
Dataflow
Reduction
Wavefront

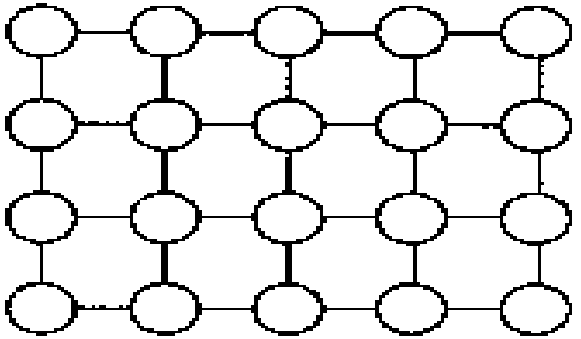
MIMD

- As we've been discussing in class.
- Autonomous processors. Communicate through message passing and coordinate with barriers.
- In order to scale these to large numbers, various connection topologies are used.

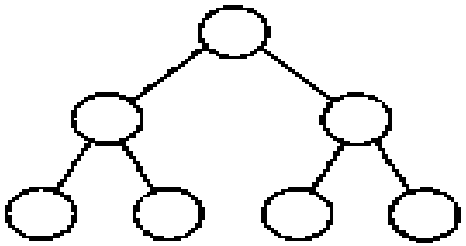
MIMD – Distributed Memory Schemes



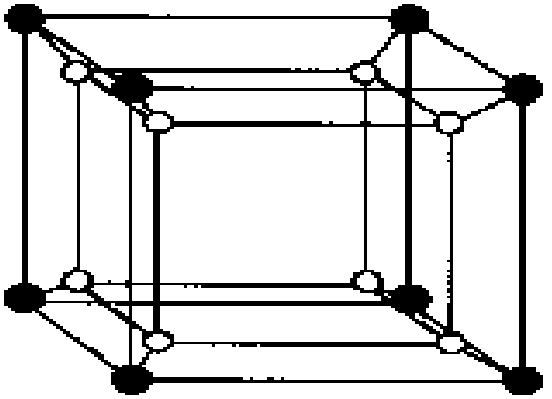
(a)



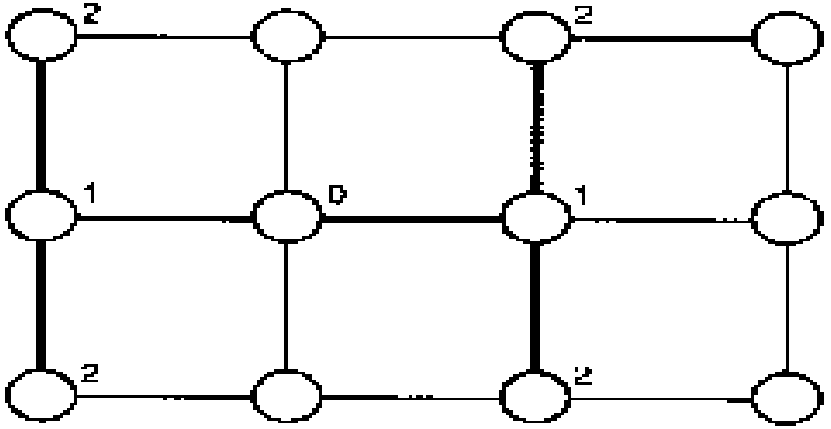
(b)



(c)



(d)



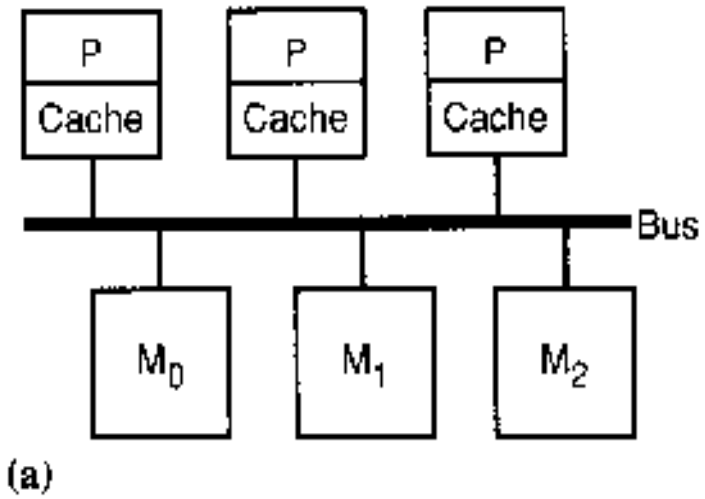
(e)

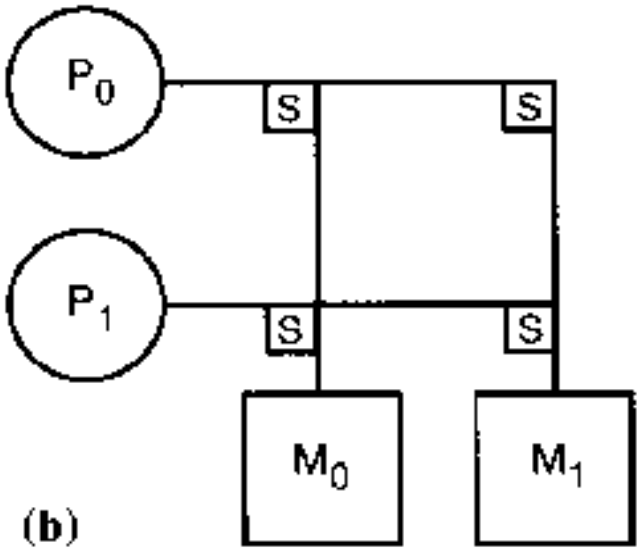
0 = root

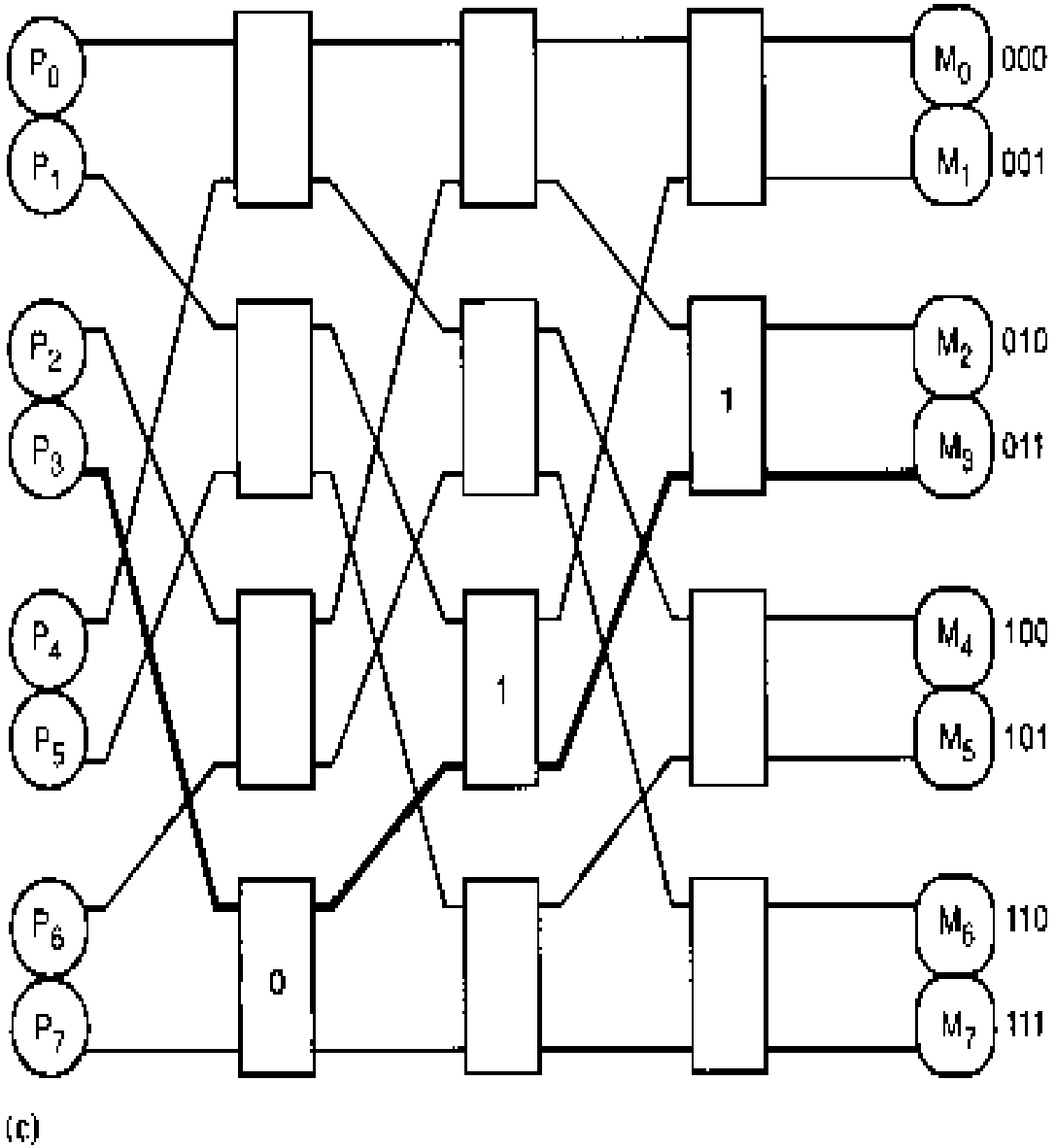
1 = level 1

2 = level 2

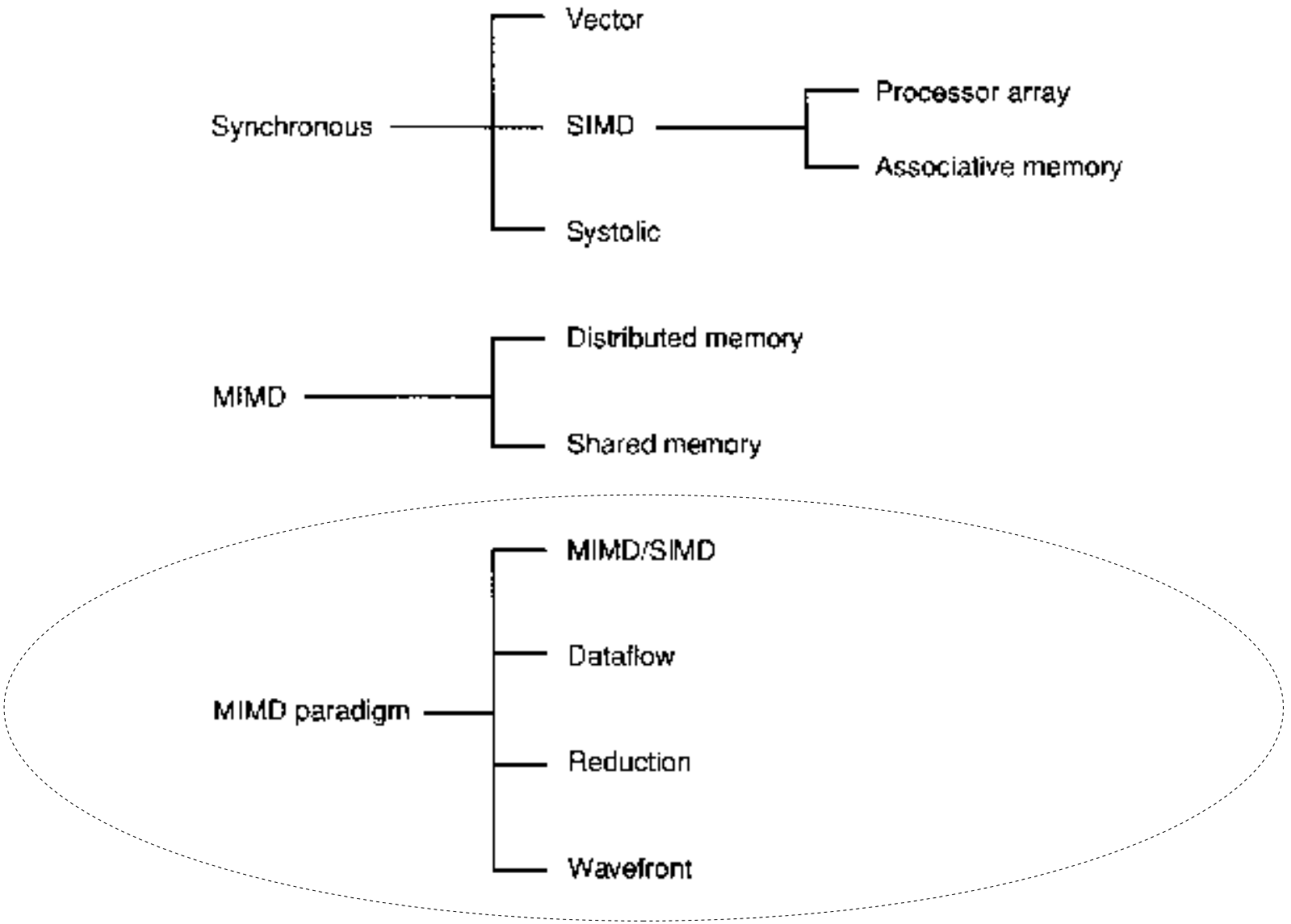
MIMD – Shared Memory Schemes







MIMD paradigm

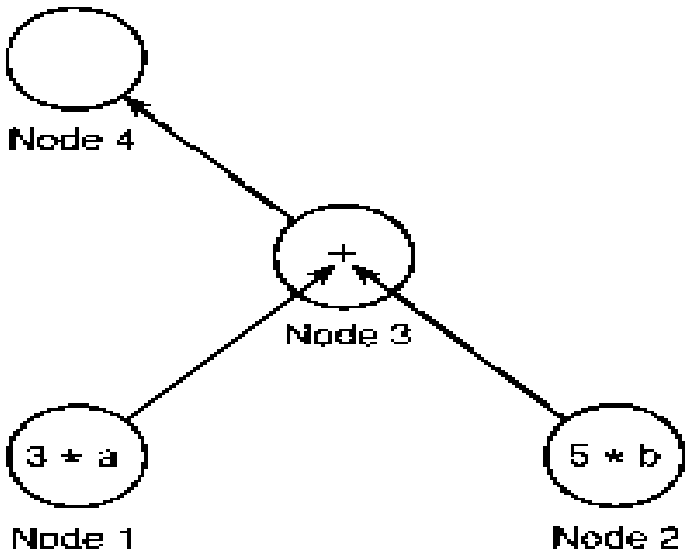


Mixing MIMD/SIMD

- A hybrid idea that started in the 1980s
- An idea we're turning to: CPUs + GPUs = APUs

Dataflow architectures

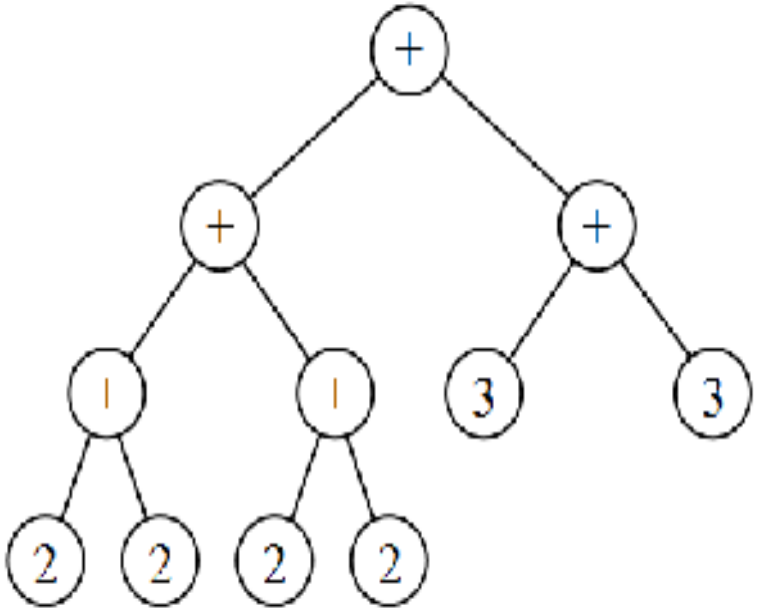
- Goal is to provide an architecture that works well with heavy data dependent problems.
- Suppose you have a huge amount of dependencies, and processes can't continue until values become available.
- Also suppose that once data becomes available, the machine knows how to quickly process that node.



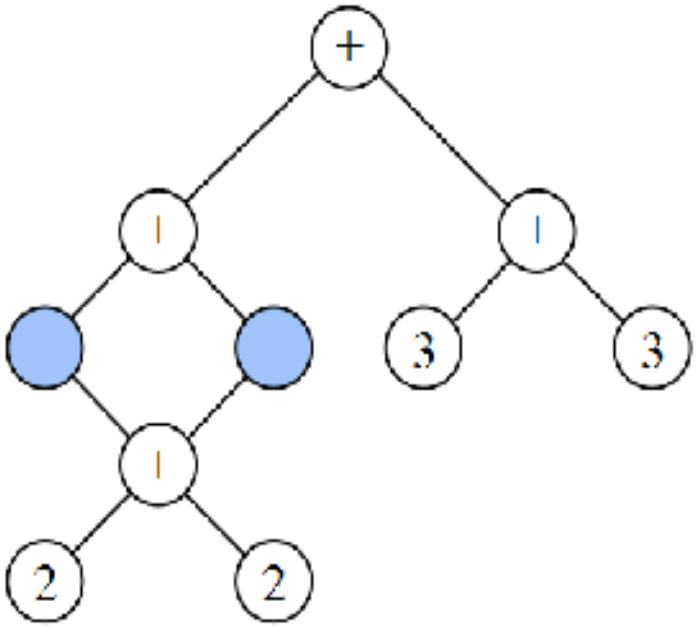
Reduction architectures

- Similar to dataflow architectures. But it takes advantage of graph reduction theory.
- For the problem ((2+2) + (2+2)) + (3+3)

Dataflow way (5 operations)



Reduction way (4 operations)

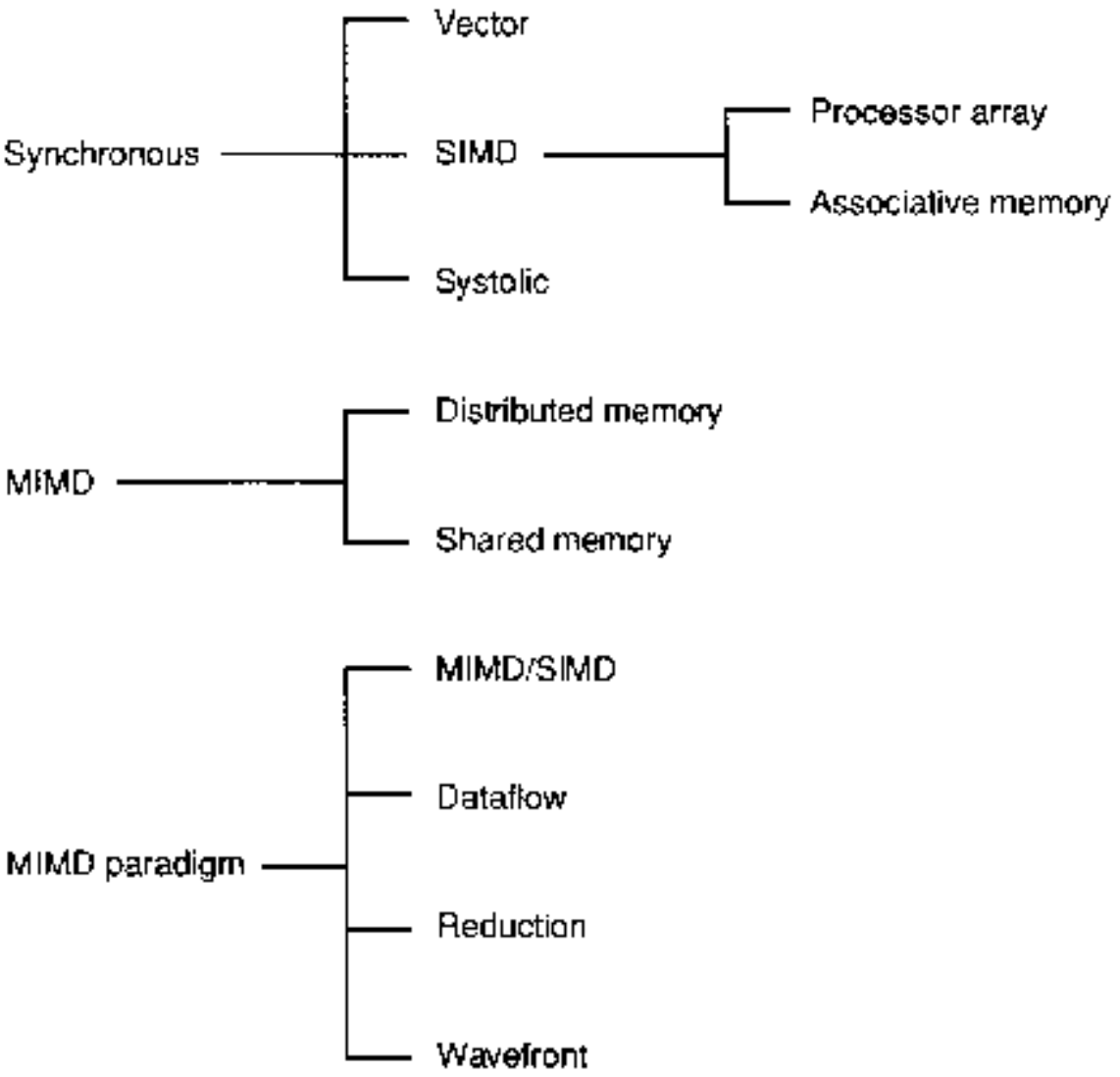


(images shamelessly stolen from Wikipedia)

Waveflow architectures.

- Very, very similar to systolic architectures.
- The difference is that instead of a clock synchronizing every step, each step work asynchronously, with message passing between processors.
- The idea of a wave is that you can almost visualize a ripple of waves as data moves through the machine.
- It seems to scale better than regular systolic architectures.

That's it!


Synchronous
Vector
SIMD
Processor array
Associative memory
Systolic
MIMD
Distributed memory
Shared memory
MIMD paradigm
MIMD/SIMD
Dataflow
Reduction
Wavefront