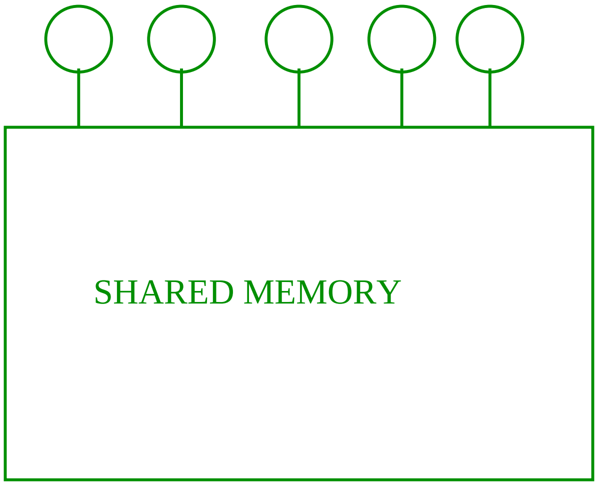
Shared memory Model

- Shared memory model:
  - natural model for expressing pointer based algorithms
- Fantasy model :
  - Algorithm design is convenient
  - Shared memory programs can be simulated on network models

Shared Memory: Parallel Random Access Machine (PRAM)



- Unit time parallel access
- Synchronous operation
- Parallel Step:

Phase 1: Read memory ($O(1)$ words)

Phase 2: Compute using standard instruction set using data fetched from memory

Phase 3: Write Memory

Exclusive/concurrent access

EREW PRAM: Exclusive Read Exclusive Write.
- Locations accessed in every phase must be distinct

CREW PRAM: Concurrent Read Exclusive Write
- Locations accessed in every write phase must be distinct

CRCW PRAM: Concurrent Read Concurrent Write
-Several processors may read/write to/from same location in any phase.
Resolution rules for Concurrent Write

1. All processors writing to same location in any phase must write identical value
   - Violation of above rule: ill formed program
2. Arbitrary:
   - Some one of the value written succeeds
   - Program must work correctly independent of which value succeeds

3. Priority: value written by the lowest numbered processor - succeeds
4. Combining(+):
   - Sum of values gets written
   - Other operators may be allowed

Theorem 1 *Any single step of an N processor PRAM (any variety) - can be simulated in $O(N)$ steps on a uniprocessor*

Theorem 2 *Any single step of an N processor bounded degree network can be simulated in $O(1)$ steps on an N processor EREW PRAM*

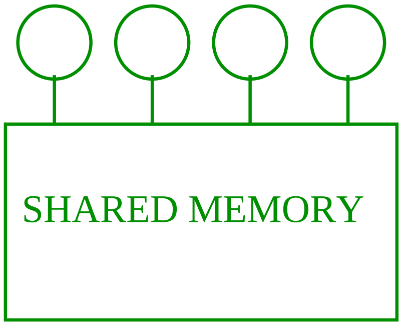
Implications of Theorem 2

$$
\begin{aligned}
& \text{Time}\begin{bmatrix}\text{Prefix of } p\log p \text{ ele-}\\ \text{ments using } p \text{ processor}\\ \text{PRAM}\end{bmatrix} \\
\leq\ & \text{Time}\begin{bmatrix}\text{Prefix of } p\log p \text{ ele-}\\ \text{ments using } p \text{ processor}\\ \text{Tree}\end{bmatrix} \\
=\ & O(\log p) \qquad (1)
\end{aligned}
$$

$$
\begin{aligned}
& \text{Time}\begin{bmatrix}p \text{ element FFT using } p\\ \text{processor PRAM}\end{bmatrix} \\
=\ & O(\log p) \qquad (2)
\end{aligned}
$$

Similarly for matrix multiplication and dense graph algorithms

PRAM MODEL

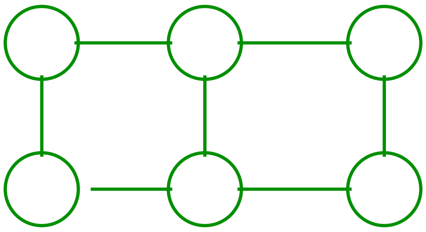


" WHICH PROCESSOR

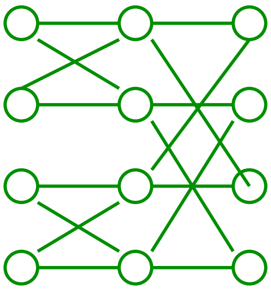
DOES WHICH

OPERATION ?"

NETWORK MODELS



" WHICH PROCESSOR
DOES WHICH
OPERATION ?"



"WHICH PROCESSOR
STORES WHICH DATA ?"

"HOW IS DATA MOVED

FROM ONE PROCESSOR TO
ANOTHER ?"

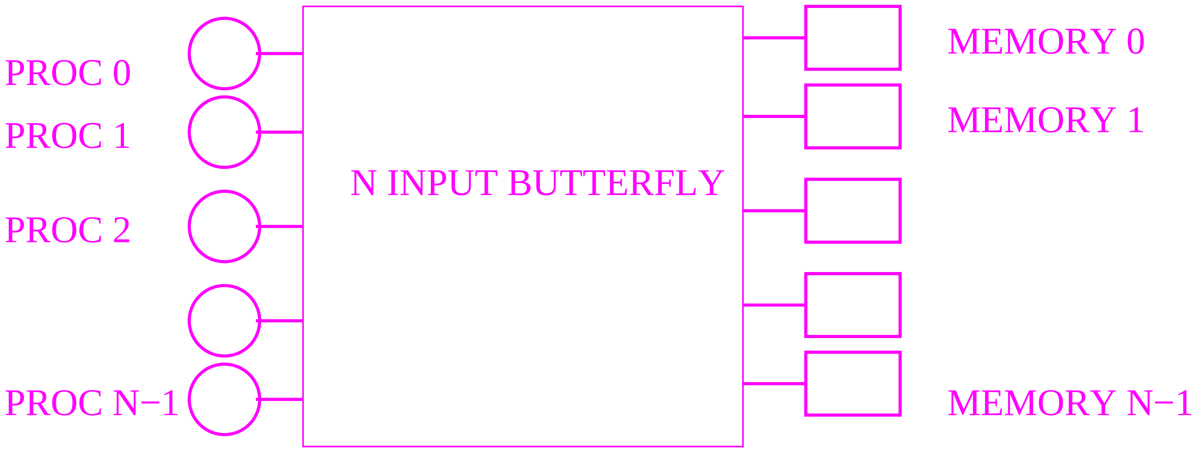
Motivation for PRAMS

- Natural representation for pointers
- Next few lectures: Algorithms on
  - lists, sparse graphs, computational geometry, algorithms, data structures

"Direct" Hardware implementations of PRAMs

- Multiport Memories: Prohibitively expensive
- Simulation of PRAMs using Butterfly/Mesh/ deBruijn...
  1. Distribute PRAM data-structures among local memories
  2. "Write x into location y $\equiv$ "Send x to processor holding location y and store"...

Simplified PRAM simulation



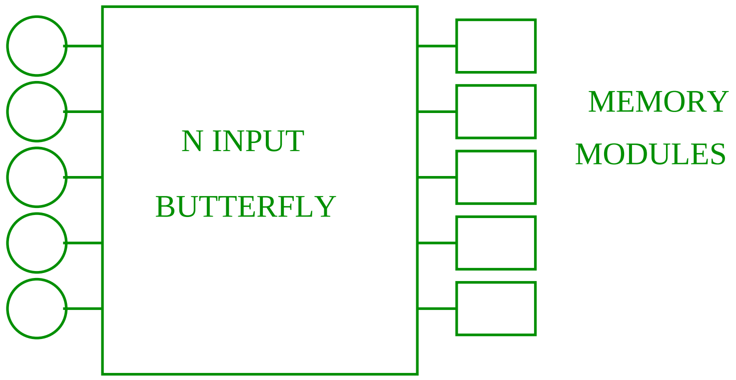
How to distribute PRAM data among memories

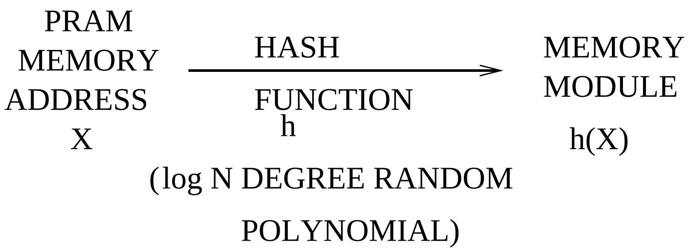
Attempt 1:

PRAM address x $\longrightarrow$ Local Address $\lfloor \frac{x}{N} \rfloor$ in memory $x \bmod N$

Maximum time to simulate 1 PRAM instruction=?

Simplified Shared Memory Simulation (Contd.)





Path selection: Direct

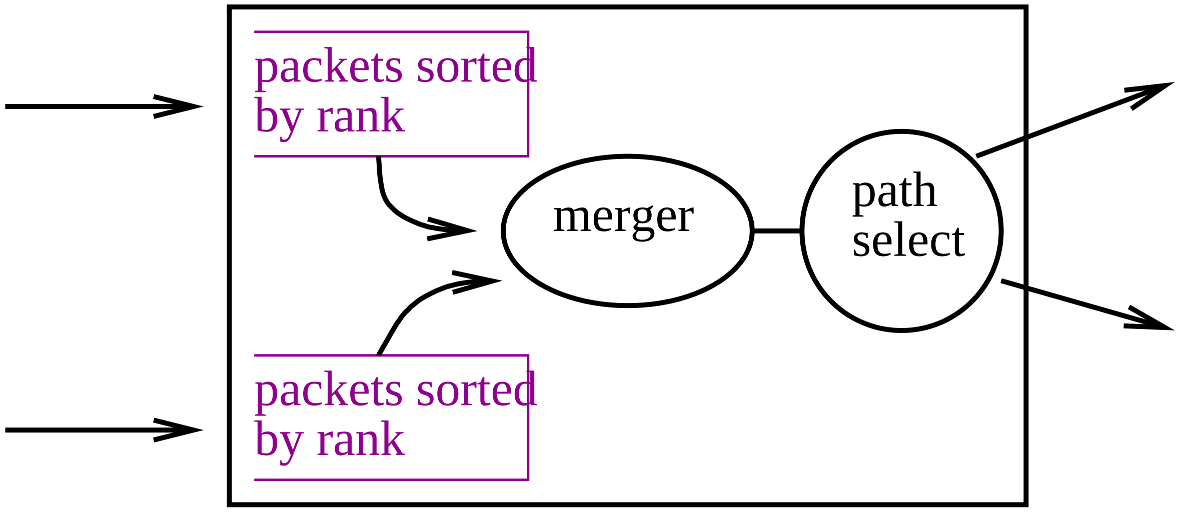
EREW congestion: $O(\log N)$ W.H.P.

Time(write instruction): $O(c + L + \log N) = O(\log N)$

Concurrent Write Simulation

"write x into location y"
Packet Rank $= g(y)||y$ where g is another hash function

$\Rightarrow$ Rank(P_i) = Rank(P_j) iff P_i and P_j write to same address.



Merger finds packets with equal rank
$\Rightarrow$ Packets at head of queues write to same address

$\Rightarrow$ Combine: 1. Throw away one packet
2. Combine data. e.g., sum, "or"
3. $\vdots$

Concurrent Read Simulation

- Rank $= g(y)||y$
- Merger forwards only one message and throws away the other
- merger keeps record of combining:
  - when data returns, it is replicated as necessary
  - $O(\log N)$ bits of additional memory is sufficient

Pram Simulation

Any single step of a P processor PRAM can be simulated in time $O \log P$ on a P processor Butterfly, $O(\sqrt{P})$ on 2D mesh W.H.P.

Problem	PRAM time	time of simulated algorithm on Butterfly	Time for best Butterfly algorithm
Sum of $P \log P$ numbers	$\log P$	$\log^2 P$	$\log P$
$P \times P$ matrix mult	P^2	$P^2 \log P$	P^2
P input FFT	$\log P$	$\log^2 P$	$\log P$
$P \log P$ input list ranking	$\log P$	$\log^2 P$	$\Omega\left(\log^2 P\right)$