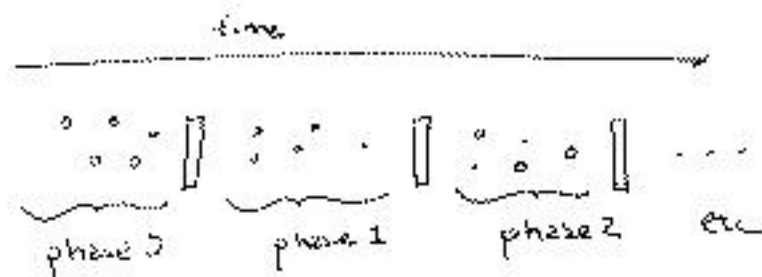


Barrier Synchronization

Often-used technique for scientific
parallel programming

Idea: n processes work together
in "phases"



No process should start phase $i+1$
until all processes have finished
phase i .

How to do this in Linda?

- text book shows
"counter" solution

(one tuple contains number of
unfinished processes in a phase)

Check is easy:

Ad(("count", 0)) ← block until count = 0

How to implement Barrier Synchronization?

1. Shared Counter - as in text

Upon finishing phase

$v \rightarrow \text{In} ("count", \text{None})$

~~write~~

~~write~~ $\text{Out} ("count", v[i] - 1)$

$\text{Rd} ("count", 0)$

2. Simulated message

(when j is finished with phase)

for i in range (n):

$\text{Out} ("Done", j, i)$

for i in range (n):

$\text{In} ("Done", i, j)$

3. Simulated broadcast

$\text{Out} ("Done", j)$

for i in range (n):

$\text{Rd} ("Done", i)$