Exam Parallelism Fabrice

For i=1 to N do if T[i]=1 return i

// there is a 1 in the array /

T = O(N)

FirstOne

Array B.

prelim = For i=1 to N in parallel do
each
$$B[i] = A[i]$$

modif . [For all i,j such that i<j in //
if B[i]=1 & B[j]=1
then B[j]=0]

a) if B[i] and B[j] are =1 too, this means
that B[j] (A[j]) is not
the first one occurrence

so at the end of the for all loop,
if A[k] = B[k]=1 , there is not any
l < k such that A[l]=1, so A[k] is
the cell having the first 1
And, in all other B[x] having A[x]=1, there is now a 0

Which PRAM do we need for this code?
for each i,j, i<j do in parallel
// time = O(1)   if (B[i] =1 & B[j]=1)
then B[j]=0

while a proc reads a cell, eg B[l]
another may read also that same cell B[k]
where the k may be = to the l.
So .E R is needed , O(n²) procs but more precisely
only $\frac{(N)(N+1)}{2}$ 1 2 3 4 .. N

•CW arbitrary is needed   2 . N
writing all 0   N+(N-1)+(N-2)...+1   3 .. N
if a write $= \sum_{i=1}^{N} i = \frac{(N \times (N+1))}{2}$ (N-1)
to B[j] is triggered

A=
0001 0 1 0 1
1 2 3 4 5 6 7 8

Ex, i, j
(4, 5)
(4, 8)
ou
6, (8) i=8

B[8]=0
written in
// by P₄ and P₆

final op.

POSITION variable is a var in global shared mem.

for all i in // do

  if (B[i] = 1) !POSITION=i, exit.

$T// = O(1)$,   EREW needed because we know
 $O(N)$ procs   only one cell in B has value 1

In total: No work optimal algorithm, because we have
for first one: $O(N^2)$ proc, time $O(1)$, but
we know seq. work = $O(N)$.

Reducing number of procs

    var shared in global memory
first one v.2. there is a one on a CRCW PRAM
   for all i, (1..N) in // do

$O(1)//$ if (A[i] == 1)   //there is a one
time.   there is a one = 1

  CW PRAM arbitrary mode is needed
because many procs may find
at the same time a cell in A that contains 1
       still, they all want to write 1

1) Ex: $(0,0,0,0,1,1,0)$ with $x = 2$

 $C = [0 \quad 0 \quad 1 \quad 1]$, then exec first one on C. CRCW O(1)
         //time also, using $O(N/x)^2$ procs.
  It returns 3
2) Given the index in C showing 1, being k (here 3)
  it corresponds to $\{A[(k-1)*x+1] .. A[k*x-1]\}$
  a "1" stored
   within subarray: So, in //, take X PRAM procs
to work on this subarray, and apply final-op on this {subarray}
       //time $O(1)$ on an EREW

3). It takes $O(1)$ time using $O(N/X)^2$ proc at maximum

4). If $x = \sqrt{N}$, $O\left(\frac{N}{X}\right)^2 = O\left(\frac{N}{\sqrt{N}}\right)^2 = O\left(\frac{\sqrt{N} \cdot \sqrt{N}}{\sqrt{N}}\right)^2 = O(N)$. processors
of a CRCW PRAM. As //time is $O(1)$, work si $O(N)$
which is now of the same order as sequential work