Improved upper and lower time bounds for parallel random access machines without simultaneous writes.

A PRAM or parallel random access machine consists of an infinite number of processors that operate in parallel and share an infinite memory. At each time step each processor reads a value from the shared memory, changes to a new state, and writes this new state back into the shared memory. Simultaneous read operations of a single shared-memory cell by many processors are allowed but simultaneous write operations into a single cell are not. An input to a PRAM $M$ consists of a finite sequence $I = (x_0, x_1, \ldots, x_{n-1})$ of natural numbers, of which $x_i$ is placed in cell $i$ of the shared memory at the start of the computation of $M$. When the processors have terminated, the contents of cell 0 constitute the output computed by $M$. The time required by $M$ is the maximum over all inputs of size $n$ of the number of steps needed to process that input, considered as a function of $n$.

Using a communication argument, the authors improve by a constant factor known lower and upper bounds on the running time for PRAMs. More precisely, they show that any PRAM that computes a so-called critical function—i.e., a Boolean function for which there exists an input $I$ with the property that changing a single bit $x_i$ from $I$ changes its output—requires at least time $0.5 \log_2 n - O(1)$, and that there exists a critical function which can be computed in time $0.57 \log_2 n - O(1)$.

Peter R. J. Asveld (NL-TWEN-C)