PARALLEL ALGORITHMS. In a PRAM (Parallel Random Access Machine), processors communicate by writing into and reading from memory cells that are accessible to all. Depending on how read and write conflicts are resolved, there are variants of the PRAM. In an Exclusive Read Exclusive Write (EREW) PRAM, no concurrent reads or concurrent writes are permitted. In a Concurrent Read Exclusive Write (CREW) PRAM, concurrent reads are permitted but concurrent writes are prohibited. In a Concurrent Read Concurrent Write (CRCW) PRAM both concurrent reads and concurrent writes are allowed. Concurrent writes can be resolved in many ways. In a Common CRCW PRAM, an arbitrary processor gets to write in cases of conflicts. In a Priority CRCW PRAM, write conflicts are resolved on the basis of priorities (assigned to the processors at the beginning).

We presented a Common CRCW PRAM algorithm for finding the Boolean AND of n given bits in O(1) time. We used n CRCW PRAM processors. As a corollary we gave an algorithm for finding the minimum (or maximum) of n given numbers in O(1) time using $n^2$ Common CRCW PRAM processors. The following result was also proven: The maximum of n arbitrary numbers can be found in $\tilde{O}(1)$ time using n CRCW PRAM processors.

We also discussed a CREW PRAM algorithm for the prefix computation problem. This algorithm uses n processors and runs in $O(\log n)$ time on any input of n elements. (For the prefix computation problem the input is a sequence of elements from some domain $\Sigma$: $k_1, k_2, \ldots, k_n$ and the output is another sequence: $k_1, k_1 \oplus k_2, \ldots, k_1 \oplus k_2 \oplus k_3 \oplus \cdots \oplus k_n$, where $\oplus$ is any binary associative and unit-time computable operation on $\Sigma$.)

We also proved the following theorems: 1) Prefix computation on n elements can be done using $\frac{n}{\log n}$ CREW PRAM processors in $O(\log n)$ time; 2) If a parallel algorithm runs in time $T$ on a $P$-processor PRAM, it can be simulated on a $P'$-processor PRAM in time $O(PT/P')$ as long as $P' \leq P$; 3) The selection problem on n elements can be solved in $\tilde{O}(\log n)$ time using $\frac{n}{\log n}$ CREW PRAM processors; 4) We can sort n given arbitrary elements in $\tilde{O}(\log n)$ time given n arbitrary CRCW PRAM processors; 5) We can sort n integers in the range $[1, \log n]^c$ in $O(\log n)$ time using $\frac{n}{\log n}$ CREW PRAM processors, c being any constant; 6) We can sort n integers in the range $[1, n(\log n)^c]$ in $\tilde{O}(\log n)$ time using $\frac{n}{\log n}$ arbitrary CRCW PRAM processors, c being any constant; 7) We can sort n arbitrary elements in $\tilde{O}\left(\frac{\log n}{\log \log n}\right)$ time using $n(\log n)^c$ arbitrary CRCW PRAM processors, c being any constant $> 0$; and 8) We can sort n integers in the range $[1, n(\log n)^c]$ in $\tilde{O}\left(\frac{\log n}{\log \log n}\right)$ time using $\frac{n(\log \log n)^2}{\log n}$ arbitrary CRCW PRAM processors.

Chernoff Bounds. These bounds can be used to closely approximate the tail ends of a binomial distribution.

A Bernoulli trial has two outcomes namely success and failure, the probability of success being $p$. A binomial distribution with parameters $n$ and $p$, denoted as $B(n, p)$, is the number of successes in $n$ independent Bernoulli trials.

Let $X$ be a binomial random variable whose distribution is $B(n, p)$. If $m$ is any integer $> np$, then the following are true:

$$\text{Prob.}[X > m] \leq \left(\frac{np}{m}\right)^m e^{m-np};$$

(1)

$$\text{Prob.}[X > (1+\delta)np] \leq e^{-\delta^3 np/3}; \text{ and}$$

(2)

$$\text{Prob.}[X < (1-\delta)np] \leq e^{-\delta^3 np/2}$$

(3)

for any $0 < \delta < 1$. 