EFFICIENT SIMULATIONS BETWEEN CONCURRENT-READ
CONCURRENT-WRITE PRAM MODELS

B. S. Chlebus†, K. Diks†, T. Hagerup‡, T. Radzik†

† Instytut Informatyki, Uniwersytet Warszawski
FKIN, p. 850, 00-901 Warszawa, Poland.
‡ FB Informatik, Universität des Saarlandes
D-6600 Saarbrücken, West Germany.

Abstract. We give several simple and efficient algorithms for
simulations of stronger CRCW PRAMs on weaker ones. The models that we
consider are the well-known PRIORITY, ARBITRARY and COMMON PRAMs, and
COLLISION and COLLISION+†, defined by the property that a special
collision symbol is stored in each memory cell into which more than
one processor attempts to write, or more than one value is attempted
to be written, respectively, in a given step. Our results are the
following, where n denotes the number of processors of the simulated
PRAM:

1) A O(1)-time simulation between any pair of models, provided
that the simulating machine has Θ(n log n) processors;

2) Two n-processor simulations: of PRIORITY on ARBITRARY with
O(log log n) slowdown, and of PRIORITY on COLLISION+ with
O((log log n)²) slowdown.

1. Introduction

The Parallel Random Access Machine (PRAM) has been generally
accepted as the model of a synchronized parallel computer with a
shared global memory (cf. [G,FW]). Various PRAMs have been introduced,
differing in whether they allow read/write conflicts, i.e., attempts
by several processors to access the same memory cell in the same step
of computation. In this paper we consider only concurrent-read
concurrent-write (CRCW) PRAMs, i.e., PRAMs that allow simultaneous
reading from as well as simultaneous writing to each cell by arbitrary
sets of processors. Simultaneous writing is not immediately logically meaningful, and further stipulations concerning the write conflict resolution rule employed are needed. Of interest to us are the following models:

**ARBITRARY**: If several processors simultaneously attempt to write to the same cell, then one of them succeeds and writes its value, but there is no rule assumed to govern the process of selection of this processor.

**COLLISION** (FRW): Whenever two or more processors simultaneously attempt to write to the same cell, a special collision symbol is written in the cell.

**COLLISION +**: Whenever at least two distinct values are attempted to be written to the same cell, a special collision symbol is written in the cell.

**COMMON** (KJ): If two or more processors simultaneously attempt to write to the same cell, they must be writing the same value, which then gets stored in the cell.

**PRIORITY** (G): If several processors simultaneously attempt to write to the same cell, then the lowest-numbered processor writes its value to the cell.

Much effort has been devoted to determining the relative power of these various models. Some relations are obvious: for instance, ARBITRARY is stronger than COLLISION + in the sense that one step of an n-processor COLLISION + can be simulated by O(1) steps of an n-processor ARBITRARY. If we express such a fact by "COLLISION + \leq ARBITRARY", then the following relations are easy to verify:

\[
\text{COLLISION} \leq \text{COLLISION}^+ \leq \text{ARBITRARY} \leq \text{PRIORITY}
\]

Grochowski and Ragde [GR] proved that COMMON is not comparable in the sense of this relation '≤' with COLLISION. Kučera [K] showed that if the simulated machine uses n processors and the simulating one is allowed to use \(n^2\) processors, then any of the considered models can be simulated on an arbitrary weaker one with only a constant slowdown (actually Kučera [K] did not consider COLLISIONs and ARBITRARY, but the statement holds true also for these models). However, a brutal squaring of the number of involved processors is seldom acceptable in efficient algorithms, and later work has concentrated mainly on the case when the machines have the same number of processors. We continue