

DETERMINISTIC ROUTING ON THE ARRAY WITH RECONFIGURABLE OPTICAL BUSES

SANGUTHEVAR RAJASEKARAN* and SARTAJ SAHNI†
*Department of CISE, University of Florida,
Gainesville, FL 32611, U.S.A.*

Received October 1996
Revised March 1997
Accepted by R. Miller

ABSTRACT

In this paper we present efficient deterministic algorithms for various classes of routing problems on the array with reconfigurable optical buses (AROB).

Keywords: Reconfigurable Networks, Packet Routing

1. Introduction

In any fixed connection network, a single step of interprocessor communication can be thought of as a packet routing task. The problem of routing can be stated as follows: There is a packet of information at each node that is destined for some other node. Send all the packets to their correct destinations as quickly as possible making sure that at most one packet crosses any edge at any time. Packet routing is equivalent to the random access write operation first defined by Nassimi and Sahni [6]. The *run time* of any packet routing algorithm is defined to be the time taken by the last packet to reach its destination. The *queue size* is the maximum number of packets that any processor will have to store during the algorithm.

The problem of *partial permutation routing* is the task of routing where at most one packet originates from any node and at most one packet is destined for any node. Any routing problem where at most h packets originate from any node and at most h packets are destined for any node will be called *$h-h$ routing* or *h -relations* [10]. In this paper we study routing problems on the AROB.

An AROB [8, 9] is essentially an $m \times n$ reconfigurable mesh in which the buses are implemented using optical technology. This model has attracted the attention

*Research of this author was supported, in part, by an NSF Grant CCR-9596065

†Research of this author was supported, in part, by the Army Research Office under grant DAA H04-95-1-0111

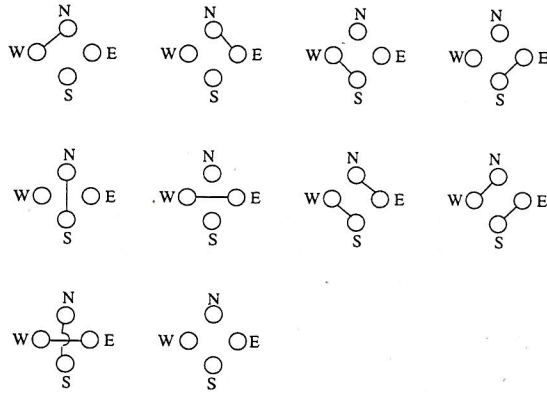


Figure 1: Possible Switch Connections

of many researchers in the recent past owing to its promise in superior practical performance. The allowable switch settings of the processors are the same as those in the RN model of [2]. These are shown in Figure . A bus link connects two adjacent processors x and y and has two associated wave guides. One of the wave guides permits an optical signal to travel from x to y and the other permits signal movement from y to x . By setting processor switches, bus links are connected together to form disjoint buses. On each bus, we need to specify which orientation of the waveguide on each link of the bus is to be used. The time needed to transmit a message on a bus is referred to as one cycle. A cycle is divided into slots of duration τ and each slot can carry a different optical signal. τ is the time needed for an optical pulse to move down one bus link. Pavel and Akl [8] have argued that for reasonable size meshes (say up to 1000×1000), the number of slots in a cycle may be assumed to be n for an $n \times n$ mesh. Further, the duration of a cycle may be assumed constant and comparable to the time for a CPU operation. A linear AROB (LAROB) is a $1 \times n$ AROB [8]. More details of the model can be found in [8, 9].

1.1. Known Results

In [8], the AROB model has been defined. Similar models have been employed before as well (see e.g., [7]). A related model known as the *Optical Communication Parallel Computer (OCPC)* has also been defined in the literature (see e.g., [1], [3], [10], [4]).

In an OCPC any processor can communicate with any other processor in one unit of time, provided there are no conflicts. If more than one processors try to send a message to the same processor, no message reaches the intended destination. [7] considers the problem of selection on a mesh with optical buses.

In [8], algorithms for such problems as prefix computation, routing on a linear array, matrix multiplication, etc. have been given for the AROB. On the other hand, [7] considers the problem of selection on a mesh with optical buses.

The proof of the following Lemma can be found in [9]:

Lemma 1 *Let \mathcal{L} be a LAROB of size n . Consider a routing problem where $O(1)$ packets originate from any node and $O(1)$ packets are destined for any node. This problem can also be solved in $O(1)$ cycles.*

The following result is due to [8]:

Lemma 2 *Say there are k elements arbitrarily distributed (at most one per processor) in a two dimensional AROB of size $\sqrt{n} \times \sqrt{n}$. We would like to 'compact' them in the first $\lceil \frac{k}{\sqrt{n}} \rceil$ rows. This problem can be solved in $O(1)$ cycles.*

Several packet routing algorithms for the OCPC model can be found in the literature. Anderson and Miller have shown that a special case of $\log n$ -relations on an n -node OCPC can be routed in $\tilde{O}(\log n)$ time [1]. Also, [10] and [3] have presented efficient algorithms for h -relations. An algorithm for arbitrary h -relations with a run time of $\tilde{O}(h + \log \log n)$ has been given by Goldberg, Jerrum, Leighton, and Rao [4]. In [9], Rajasekaran and Sahni show that any h -relation can be routed in $\tilde{O}(h)$ time on a LAROB. In contrast, the best known algorithm for the OCPC model has a run time of $\tilde{O}(h + \log \log n)$ [4]. They also present an algorithm for h -relation routing on a $\sqrt{n} \times \sqrt{n}$ AROB with a run time of $\tilde{O}(h + \log \log n)$. The best known algorithm for the same problem on the OCPC model has a run time of $\tilde{O}(h + \frac{\log n}{\log \log n})$ [3]. Their deterministic algorithm for h -relations runs in time $O(h \log n)$ on a $\sqrt{n} \times \sqrt{n}$ AROB as well as on an n -node LAROB.

Definition [BPC Permutations][5]: In a network \mathcal{N} with N nodes, $N!$ permutations are possible. An important subset of these permutations is called *Bit Permute and Complement (BPC)*. Assume that N is an integral power of 2 (i.e., $N = 2^p$ for some integer p). Each node of \mathcal{N} can be labeled with a p -bit sequence $a_{p-1}, a_{p-2}, \dots, a_1, a_0$. Any BPC permutation, π can be described with a vector $(\pi_{p-1}, \pi_{p-2}, \dots, \pi_1, \pi_0)$, where $\pi_i \in \{\pm 0, \pm 1, \dots, \pm(p-1)\}$ and $(|\pi_{p-1}|, |\pi_{p-2}|, \dots, |\pi_1|, |\pi_0|)$ is a permutation of $(0, 1, 2, \dots, p-1)$. Under this permutation, if the origin of a packet is $a_{p-1}, a_{p-2}, \dots, a_1, a_0$ then its destination will be $d_{p-1}, d_{p-2}, \dots, d_1, d_0$, where $d_{|\pi_i|} = a_i$ if π_i is non negative and $d_{|\pi_i|} = \bar{a}_i$ otherwise.

Consider a network \mathcal{N} with 8 nodes, for example. Let the permutation under concern be $\pi = (-2, 0, 1)$. Under this permutation a packet originating from the node (a_2, a_1, a_0) will be destined for (\bar{a}_2, a_0, a_1) . For instance a packet from node $(1, 1, 0)$ will have $(0, 0, 1)$ as its destination.

Many important permutations such as matrix transpose, bit reversal, perfect shuffle, etc. belong to the class of BPC permutations. For example the perfect shuffle can be characterized with the vector $(0, p-1, p-2, \dots, 2, 1)$. The vector $(p/2-1, p/2-2, \dots, 1, 0, p-1, p-2, \dots, p/2)$ corresponds to matrix transpose. Bit reversal is described by $(0, 1, 2, \dots, p-1)$, and so on. The number of permutations in the BPC class is $2^p p!$ where the network size is $N = 2^p$.

2. Matrix Transpose Routing

In this section we show how to perform matrix transpose routing in $O(1)$ cycles

