Efficient Realization of Frequently Used Bijections on Cube-Connected Cycles

Liu Zhiyong (刘志勇), Liu Qun (刘群) and Zhang Xiang (张祥)
Institute of Computing Technology, The Chinese Academy of Sciences, Beijing 100080
Received December 12, 1993.

Abstract
CCC has lower hardware complexity than hypercube and is suited for current VLSI technology. LC-permutations are a large set of important permutations frequently used in various parallel computations. Existing routing algorithms for CCC cannot realize LC-permutations without network conflict. We present an algorithm to realize LC-permutations on CCC. The algorithm consists of two periods of inter-cycle transmissions and one period of inner-cycle transmissions. In the inter-cycle transmissions the dimension links of CCC are used in a "pipeline" manner and in the inner-cycle transmissions the data packets are sorted by a part of its destination address. The algorithm is fast (O(log₂ N)) and no conflict will occur.

Keywords: Hypercube, cube-connected cycles, linear complement permutation, routing algorithm, conflict, complexity.

1 Introduction

Hypercube is a promising topology for parallel and distributed processing systems. In a hypercube of size $N = 2^n$, $n$ links are connected to each node. The hardware complexity of hypercube increases quickly as $n$ increases. CCC (Cube-Connected Cycles) is a feasible substitute of hypercube, in which constant(3) links are connected to each node[1]. A set of algorithms (Descend and Ascend) on hypercube can be simulated on CCC without significant degradation of performance[10].

Permutation is the communication pattern in a multiprocessor system by which each processor communicates with one and only one processor[4,5]. The traditional hypercube routing algorithm is the naive routing algorithm, which uses the $n$ dimensions of hypercube in a low to high or high to low order, and the $n$ bits of the destination address are compared with the $n$ bits of the current node address in order to determine if the message should be sent or just remain in the same node. Naive routing algorithm can be easily simulated on CCC. However, permutations which can be passed by the naive algorithm on a hypercube or CCC without conflict are very limited. In such a permutation the destination addresses on each subcube must constitute a complete residue system (CRS)[4,5]. LC-permutations are a large

This research is supported by China National "863" High Technology Project.
set of permutations frequently used in various parallel computation tasks in image processing, pattern recognition, numerical analysis, signal processing, and other scientific and engineering computations, and they include BPC-permutations as a subset\cite{3,5-7,11}. As an instance, the data alignment requirements in the non-linear parallel storage schemes, known as XOR-schemes, are LC-permutations\cite{3,6,11}. Unfortunately, although CCC is less complex in hardware than hypercube, existing routing algorithms (Descend or Ascend) for CCC cannot realize such a large set of LC-permutations without conflict.

If no conflict occurs in the transmission process, the communication will be more efficient in terms of both message delay time and hardware utilization. Various algorithms have been proposed for LC (as well as BPC) permutations on hypercubes and multistage interconnection networks (MINs)\cite{1,2,5,7-9,12-14}. In this paper, we give a conflict-free routing algorithm for LC-permutations on CCC, based on a corresponding routing algorithm on hypercube\cite{5}, with the same time complexity. This is the first conflict-free routing algorithm for such a large set of permutations (LC-permutations) on CCC.

2 Some Definitions

Consider a multiprocessor system with \(N = 2^n\) nodes. Each node \(M\) has an address \(m_{n-1}m_{n-2}\ldots m_1m_0\).

In a hypercube, there is a link between node \(A\) and node \(B\) if and only if the addresses of \(A\) and \(B\) differ in exactly one bit position \(k\) \((0 \leq k < n)\). We call this link the \(k\)-th dimensional link. And we call the collection of all the \(k\)-th dimensional links the sheaf \(k\).

In CCC, the address of node \(M\) is divided into two parts:

\[M = (M_u, M_y), \quad M_u = m_{n-1}\ldots m_{n-u}, \quad M_y = m_{y-1}\ldots m_0,\]

where \(u + y = n\), and \(y\) is the smallest integer for which \(y + 2^y \geq n\). We call the \(u\) bits \(M_u\) the cube-address of node \(M\), while the \(y\) bits \(M_y\) the cycle-address. Each node \(M\) of CCC has three ports: \(F, B,\) and \(L\) (Forward, Backward, and Lateral). \(F\) is connected to node \((M_u, (M_y + 1)\mod 2^y)\), \(B\) is connected to node \((M_u, (M_y - 1)\mod 2^y)\) and \(L\) is connected to node \((M_u \oplus 2^y, M_y)\). The last link does not exist when \(M_y \geq u\). All the nodes with the same \(M_u\) are linked as a cycle by \(F-B\) links, and all the cycles are linked as a \(u\)-cube by the \(L-L\) links.

In a Linear-Complement (LC) permutation, the source address \(S = s_{n-1}\ldots s_0\) and the destination address \(D = d_{n-1}\ldots d_0\) can be expressed as follows:

\[D^r = T \times S^r \oplus C^r\]

where \(T\) is a nonsingular \(n \times n\) binary matrix:

\[
T = \begin{pmatrix}
  t_{n-1,n-1} & t_{n-1,n-2} & \cdots & t_{n-1,0} \\
  t_{n-2,n-1} & t_{n-2,n-2} & \cdots & t_{n-2,0} \\
  \cdots & \cdots & \cdots & \cdots \\
  t_{0,n-1} & t_{0,n-2} & \cdots & t_{0,0}
\end{pmatrix}
\]