Research Article

Embedding $N \times N$ Crossbar Switches into ILLIAC$(N, N - 1)$ Torus Networks for Optical Interconnection Networks

P. Selin and H. Obara

Graduate School of Electrical and Electronic Engineering, Akita University, 1-1 Tegata Gakuen, Akita 010-8502, Japan

Correspondence should be addressed to H. Obara, obara@ee.akita-u.ac.jp

Received 27 September 2012; Accepted 24 October 2012

Academic Editors: M. D. Hoogerland and G. Montemezzani

Copyright © 2012 P. Selin and H. Obara. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A focus of research for crossbar switches has been to reduce the number of $2 \times 2$ switch elements, each of which not only occupies a relatively large area but also costs a lot when implemented with optical devices. We point out that conventional $N \times N$ crossbar switches, where $N$ is the switch size, actually function as $(N + 1) \times (N + 1)$ switches and thus have redundant $2 \times 2$ switch elements to be removed. We show that ILLIAC$(N, N - 1)$ torus networks, which have a less number of $2 \times 2$ switching elements than crossbar switches of an identical switch size, are equivalent to true $N \times N$ crossbar switches. We describe in detail how the crossbar switches can be embedded into the equivalent torus networks. We also discuss switch control complexity of the equivalent torus networks and show that it can be reduced to $O(1)$.

1. Introduction

The crossbar switch (XBS) is the practical choice for implementing switching fabrics in optical networks [1]. There are two typical configurations of conventional optical XBSs [2]. The first type, which uses $1 \times 1$ transfer switches and constitutes a full-mesh network, is strict-sense nonblocking but suffers from a large power loss, crosstalk, and wiring area to be out of the scope in this paper. We consider the second type of optical XBSs that is composed of $2 \times 2$ switching elements, each of which has two connection states (i.e., bar and cross), and is wide-sense nonblocking. We refer to the $2 \times 2$ optical switching element, typically implemented with an optical directional coupler, as a cell for simplicity. An XBS has up to $N^2$ cells, where $N$ is the switch size. A focus of research has been to reduce the number of cells (or simply crosspoints) required for optical XBSs, because each individual optical cell occupies a relatively large space, consumes power, and costs a lot [3]. In 1968, Kautz et al. devised different switches with fewer than $N^2$ cells, all of which were derived from XBSs, for example, triangular switches with $N(N - 1)/2$ crosspoints [4], while they are rearrangeably nonblocking unlike XBSs. In 1988, Smyth gave a pruned XBS with $N^2 - 3$ cells, preserving wide-sense nonblocking capability [5]. In 2008 Obara suggested in a letter a new pruned XBS with $N(N - 1)$ cells embedded into ILLIAC$(N, N - 1)$ torus networks [6]. In his discussion, however, the question of how conventional XBSs can be transformed into the ILLIAC torus networks has been left open. In this paper, we begin with some definitions and notations for XBSs necessary to later discussion. Then we describe the transformation process and discuss switch control complexity of the equivalent torus networks.

2. Definitions and Notations for XBSs

We consider a conventional $N \times N$ XBS composed of $N^2$ cells, as shown in Figure 1, where $i$ and $j$ represent its input and output port numbers, $0 \leq \{i, j\} \leq N - 1$. $(i, j)$ denotes the cross-point of the $i$th row and the $j$th column, and $c(i, j)$ denotes the cell on the cross-point. The cell has two connection modes, that is, cross and bar, as shown in Figures 2(a) and 2(b). All the cells are initially set to the cross-state (i.e., the default state). When the $i$th input corresponds to the $j$th output, only a single cell at $(i, j)$ will be set to the bar state. Upon releasing the connection, the same $c(i, j)$ will be set back to the cross-state. Thus, the switch control for XBSs...
Input

Output

Figure 1: Conventional $N \times N$ crossbar switch composed of $2 \times 2$ cells.

Figure 2: Two connection patterns of a cell.

Figure 3: Decomposition of a cell into a pair of $1 \times 2$ and $2 \times 1$ SWs.

Figure 4: Deformed XBS with rotated port positions in diagonal direction.

3. Procedure of Transformation from XBS to Torus Network

It is readily seen in Figure 1 that one of the input or output ports of the cells in the top row and the rightmost column is left unused. We know that these unused ports are reserved for scaling the switch size. If we add an XBS with three other XBSs of the same size, we have an XBS of double size. However, if the switch size is fixed, all the idle ports are of no use, and the idle $1 \times 2$ and $2 \times 1$ SWs in the cells can be eliminated, and thus we have a new $N \times N$ XBS with fewer than $N^2$ cells. The following discussion holds for any $N$, but we depict all figures for the case of $N = 4$ for simplicity and clarity. We begin with an XBS shown in Figure 4 that is deformed by the vertical and horizontal rotations, where $h_i$ and $v_j$ are given by

$$h_i = N - 1 - i \quad (0 \leq i \leq N - 1),$$

$$v_j = j \quad (0 \leq j \leq N - 1).$$

Equation (1) means that both input and output ports are aligned diagonally right down. Input ports are on the diagonal cells, while output ports are set on the cells next to the diagonal cells. Output port 1 (i.e., $j = 1$ in (1)), for example, looks to be shifted upward by $N - 1$, but it is really shifted downward by 1 in accordance with the definitions...
described in Section 2. Note that \( c(0, 3) \) and \( c(3, 3) \) in Figure 4 lie next to each other through a vertical wraparound link, although they seem to be far apart. In Figure 4, we see that plain squares at \( c(i, N - 1 - i) \) for \( i = 0 \) to \( N - 1 \), each of which has an input port \( i \), function as \( 1 \times 2 \) SWs. We also see that shaded squares at \( c(N - 2 - j, j) \) for \( j = 0 \) to \( N - 2 \) and \( c(N - 1, N - 1) \), each of which is adjacent to the plain squares, function as \( 2 \times 1 \) SWs. Recall that a pair of \( 1 \times 2 \) and \( 2 \times 1 \) SWs constitutes a cell as shown in Figure 3. As a result we have a modified XBS in Figure 5, where all the pairs of adjacent \( 1 \times 2 \) and \( 2 \times 1 \) SWs in Figure 3 are merged into cells, which are shown by the shaded cells. Note that dashed cells show empty cells corresponding to the original \( 2 \times 1 \) SWs that are merged into the shaded cells. Finally, move all the cells under the shaded cells upward by one and also move \( c(0, N - 1) \) to \( (N - 1, N - 1) \). As a consequence, the bottom line of cells in Figure 5 disappears, and we have an ILLIAC\((N, N - 1)\) torus network as shown in Figure 6, where all the cells are interconnected in the horizontal direction to constitute a global loop, while each set of vertically aligned cells constitutes a local loop [7]. We refer to it as a torus-embedded XBS (TE-XBS). The number of cells in the \( N \times N \) TE-XBS is \( N(N - 1) \), which is \( N \) less than \( N^2 \) of conventional XBSs.

4. Switch Control Complexity

We assume a TE-XBS shown in Figure 7, of which input ports are renumbered so as to have a number identical to the output port number in each column. We label every cell with a pair of numbers, that is, \((p, q)\). \( p \) increases from left to right for each row, and thus it is identical to the output port number in the column \( (0 \leq p \leq N - 1) \). Note that \( q \) starts from the cell that has an input port and ends in the cell that has an output port within each row \( (0 \leq q \leq N - 2) \). We assume that all the cells are initially set to the cross-state, where an input port corresponds to the output port with the identical number. It is easy to see that each column is devoted to the route to a corresponding output port. Now we show how an internal route is provided when a pair of source and destination numbers, \( i \) and \( j \), is given, that is, a call arrives at \( c(i, 0) \) destined for an output \( j \). We have the following simple switch control algorithm:

\[
\begin{align*}
\text{if } i &= j \\
&\quad \text{Set } c(i, h) \text{ to the cross-state for } h = 0 \text{ to } N - 2. \\
\text{else} & \\
&\quad k = (i - j + N - 1) \mod N, \\
&\quad \text{Set } c(i, 0) \text{ and } c(j, k) \text{ to the bar state,} \\
&\quad \text{Set } c((i + s) \mod N, N - 1 - s) \text{ to the cross-state for } s = 1 \text{ to } N - k, \\
&\quad \text{Set } c(j, k + t) \text{ to the cross-state for } t = 1 \text{ to } N - k - 2, \\
\end{align*}
\]

end if

(2)

In other words, when \( i = j \), a route is provided vertically within a column. When \( i \neq j \), a route starting from the cell with the \( i \)th input port goes along the global loop to the \( j \)th column and then diverts its route to exit the \( j \)th output port straight through the column. In either case the internal route is provided over one or two sets of contiguous cells aligned in the horizontal or vertical direction. When the connection is released, the set of cells once fixed in its setup will be left unchanged unlike conventional XBSs, unless another input port corresponds to the idle output port. Consequently, the number of cells to be set in the TE-XBS is distributed between \( N - 1 \) and \( 2N - 2 \), while that in XBSs is only one. Processing time for the switch control becomes \( O(N) \) when conventional CPUs are used for the switch control circuit. However, the switch control algorithm is so simple that the sets of cells given in (2) can be identified simultaneously by a certain digital circuit of a parallel processing capability, and thus the switch complexity can be reduced to \( O(1) \).
5. Conclusions

We have presented how conventional $N \times N$ crossbar switches with $N^2$ cells can be embedded into ILLIAC($N, N - 1$) torus networks with $N(N - 1)$ cells. The key idea is to rotate input and output ports in horizontal and vertical directions and to merge adjacent half-idle cells. Although the idea is simple and straightforward, we believe that it is published for the first time. The number of cells to be switched for one call in the torus-embedded crossbar switch is distributed between $N - 1$ and $2N - 2$, while that in the crossbar is only one. Here we can see a trade-off relation in performance between the number of cells and switch control complexity. The switch control complexity for the torus-embedded crossbar switches, however, can be reduced to $O(1)$ when a certain parallel control circuit is used.

References
