Routing Algorithms on the Bus-based Hypercube Network

Lee-Chuan Fan and Chang-Biau Yang
Department of Applied Mathematics
National Sun Yat-sen University
Kaohsiung, Taiwan 80424
cbyang@math.nsysu.edu.tw

Abstract

In this paper, we study the properties of the bus-based hypercube, denoted as $U(n,b)$, which is a kind of multiple-bus network (MBN). $U(n,b)$ consists of 2^n processors and 2^b buses, where $0 \leq b \leq n-1$, and each processor is connected to either $\lceil \frac{b+2}{2} \rceil$ or $\lceil \frac{b+1}{2} \rceil$ buses. We show that the diameter of $U(n,b)$ is $\lceil \frac{b+1}{2} \rceil$ if $n \geq 3$. We also present an algorithm to select the best neighbor processor via which we can obtain one shortest routing path. In $U(n,b)$, we show that the fault diameter $DF(n,b,f) \leq b+1$, where f is the sum of bus faults and processor faults and $0 \leq f \leq \lceil \frac{b-3}{2} \rceil$. Furthermore, we also show that the bus-fault diameter $DB(n,b,f) \leq \lfloor \frac{b}{2} \rfloor + 3$, where $0 \leq f \leq \lceil \frac{b-1}{2} \rceil$ and f is the number of bus faults.

Key words: multiple-bus network, hypercube, routing algorithm, diameter, fault tolerance.

1 Introduction

One of the most important components of a parallel processing system is the interconnection network. The method of connecting processors is also an important consideration for design and performance of the system. Various interconnection networks have been proposed and studied. Notable examples are the *crossbar switch* nd *multiple bus* systems, *multi-stage* networks, and point-to-point (direct) connection schemes.

In this paper, we consider a kind of interconnection network called the *multiple-bus network*(MBN) [3,5,7, 9,16–19]. The MBN is an extension of the single bus network. An MBN consists of a set of processors and a set of buses. Any pair of processors can communicate via the buses which they both are connected to, and only one message may be transmitted on a bus during a time step.

MBNs have several advantages over point-to-point networks. Some of them are listed as follows: (1) In a point-to-point network, a communication link is connected to a pair of processors. In an MBN, a cluster of processors share a single communication bus, therefore it can be used more efficiently. (2) The number of buses is independent of the number of processors. Hence the network designer can make trade-off between the performance of the architecture and the physical resource. (3) Broadcasting is easy in an MBN. (4) An MBN is easy to be extended to a larger system.

One of the disadvantages of MBNs is that it is difficult to implement very large systems. As the length of a bus increases, the system will operate slowly. Some overlapping connectivity networks were proposed to solve the problem, and bandwidth formulas for these networks were derived by using probabilistic analysis methods [9]. Vaidyanathan and Padmanabhan proposed a bus-based hypercube network, which can perform uniform hypercube algorithms optimally [16]. Many practical hypercube algorithms are uniform. Some special cases of uniform hypercube algorithms have been studied and some applications of them have been identified [1].

Many other MBNs have been proposed. Dighe et al. proposed a class of MBN called *bus-connected ring trees*(BRTs) and *bus based trees*(BBTs) [3]. Ali and Vaidyanathan presented the exact lower bounds on running ASCEND/DECEND and FAN-IN algorithms on synchronous MBNs [1]. Multiple buses have also been used in some synchronous reconfigurable systems [11,13,15]. Some modified network topologies have been proposed to enhance communication performance [4,5,14]. Ishikawa proposed a modified hypercube with multiple buses which used a bypass routing method to reduce the diameter to two [6].

The rest of this paper is organized as follows. In Section 2, we shall review the construction of the network. In Section 3, we shall give the terms and notations we shall use in this paper. In Section 4 and Section 5, the distance properties between two processors will be investigated. We show that the diameter of the n-dimensional network without faults is $\lceil \frac{n}{2} \rceil$. Accordingly, we design the shortest path routing algo-

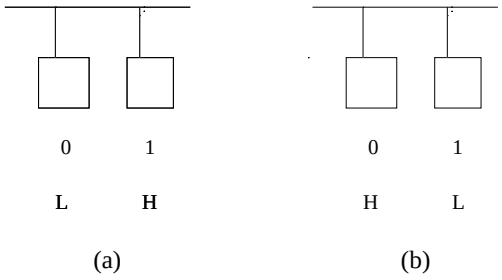

Figure 1: Initial structure of U. (a)$U(1,0)$. (b)$\overline{U}$(1,0).

rithm for the network, which has never been designed before. In Section 6, We shall present some fault tolerant properties of this network. By finding disjoint paths, we obtain the diameters with some bus faults and with some bus and/or processor faults. These results improve significantly the previous result. Finally, the conclusion will be given in Section 7.

2 Construction of the Bus-based Hypercube Network

In this section, we shall describe how to construct the bus-based hypercube network [16]. The bus-based hypercube network $U(n,b)$ consists of 2^n processors and 2^b buses, where $b < n$. The fan-out of each processor, the number of connections to buses, is either $\lceil \frac{b+2}{2} \rceil$ or $\lceil \frac{b+1}{2} \rceil$. A processor with fan-out $\lceil \frac{b+2}{2} \rceil$ is called a *high* processor and with $\lceil \frac{b+1}{2} \rceil$ is called a *low* processor.

The construction of $U(n,b)$ is similar to that of the hypercube. In $U(n,b)$, each processor has a unique identifier between 0 and 2^n-1, and each bus also has a unique identifier between 0 and 2^b-1. $U(n,b)$ can be represented by a Boolean matrix $M_{n,b}$ of size $2^n \times 2^b$, where the entry $(i,j) = 1$ if and only if processor i is connected to bus j. $U(n,b)$ can be defined by the following recursive way:

(1) The initial structure of the bus-based hypercube network, $U(1,0)$ and $\overline{U}(1,0)$, are shown in Figure 1. In the figure, L and H are used to denote a low processor and a high processor respectively. $\overline{U}(1,0)$ is an inverse of $U(1,0)$. In $U(1,0)$ or $\overline{U}(1,0)$, the bus is called the *host bus* of these two processors, because the processors are connected to the bus first.

(2) The construction of $U(n+1,b+1)$ can be explained by the Boolean matrix $M_{n+1,b+1}$, which is divided into four quadrants, as shown in Figure 2. Each quadrant is a matrix of size $2^n \times 2^b$. Initially,

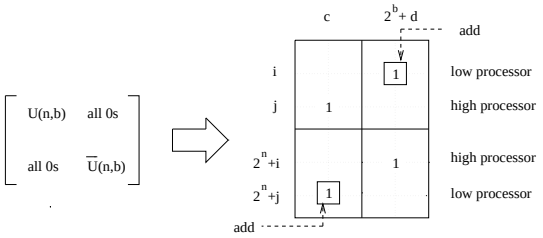

Figure 2: Construction of $U(n+1,b+1)$.

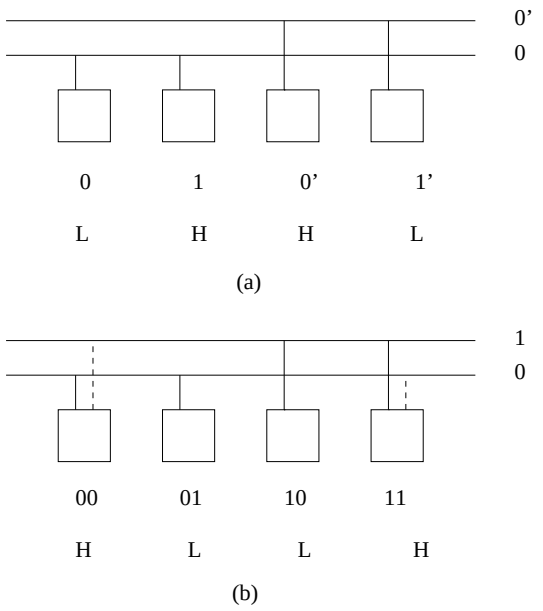

Figure 3: Construction of U(2,1). (a) Combining $U(1,0)$ and $\overline{U}(1,0)$. (b) Adding connections(dashed lines) in *low* processors in $U(1,0)$ and $\overline{U}(1,0)$. Then change the state from low to high and the state from high to low.

the first and third quadrants are all 0's, the second and fourth quadrants are $U(n,b)$ and $\overline{U}(n,b)$ respectively. Assume i and j, $0 \le i,j \le 2^n-1$, are a low and a high processor in $U(n,b)$ respectively. Then i and j are high and low processors in $\overline{U}(n,b)$ respectively. As shown in Figure 2, a connection is added from a low processor i of $U(n,b)$ to the host bus d of the high processor i of $\overline{U}(n,b)$. Here the bus d is called a *guest bus* of i of $U(n,b)$. And, similarly, a connection is added from a low processor j of $\overline{U}(n,b)$ to the host bus c of the high processor j of $U(n,b)$, and the bus c is a guest bus of j of $\overline{U}(n,b)$.

(3) To obtain $U(n+1,b)$, two $U(n,b)'s$ are combined, but the buses are not doubled. The buses in the two $U(n,b)$'s are overlapped.

As examples, the constructions of $U(2,1)$, $U(3,1)$ and $U(3,2)$ are shown in Figure 3, Figure 4 and Figure 5 respectively.

The following theorem gives some properties of

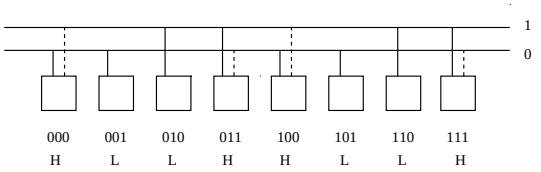

Figure 4: The construction of $U(3,1)$. $U(3,1)$ consists of two $U(2,1)$'s, and the buses are shared to each other.

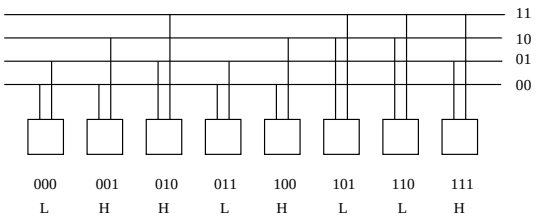

Figure 5: The bus-based hypercube $U(3,2)$.

$U(n,b)$ and $\overline{U}(n,b)$.

Theorem 1 *[16] For all $0 \leq b \leq n-1$,*

(1) $U(n,b)$ can run any step of a uniform hypercube algorithm optimally.

(2) Each bus is connected by $(b+2)2^{n-b-1}$ processors.

(3) Processor i, $0 \leq i \leq 2^n-1$, is a high(low) processor of $U(n,b)$ if and only if processor i is a low(high) processor of $\overline{U}(n,b)$.

(4) Each bus j of $U(n,b)$ or $\overline{U}(n,b)$ is connected to at least two processors i_1 and i_2 such that $0 \leq i_1 \leq 2^{n-1}-1$ and $2^{n-1} \leq i_2 \leq 2^n-1$.

Vaidyanathan and Padmanabhan also showed that $U(n,b)$ has a low diameter and is highly resilient to bus faults [16]. The bus-fault diameter $DB(n,b,f)$ denotes the diameter of $U(n,b)$ with any f bus faults.

Theorem 2 *[16] For all $n > b \geq 0$, for all $0 \leq f \leq \lceil \frac{b-1}{2} \rceil$, $DB(n,b,f) \leq b+2f+1$.*

3 Definitions and Notations

We need some notations as follows.

- $Bin(i,n)$: n binary bits for representing i, where $0 \leq i \leq 2^n-1$. The rightmost and leftmost bits of $B(i,n)$ are counted as bits 0 and $n-1$ respectively.
- $Zero(i,n)$: number of bit positions in $Bin(i,n)$ having value 0.
- $One(i)$: number of bit positions in $Bin(i,n)$ having value 1.
- $Even(i)$: number of even bit positions, excluding bit 0, in $Bin(i,n)$ having value 1.
- $Odd(i)$: number of odd bit positions in $Bin(i,n)$ having value 1.
- $B(i)$: the set of buses connected to processor i.
- $H(i,j)$: the distance from processor i to processor j, which is the number of hops in the shortest path from i to j.
- R_{ij}: the bit-wise exclusive-or operation on i and j, i.e. $R_{ij} = Bin(i,n) \bigoplus Bin(j,n) = (r_{n-1}, r_{n-2},..., r_0)$.
- $SP(i,j)$: the processors on one shortest routing path from processor i to processor j.
- $State(i)$: the state of processor i, either high or low.
- $High(i)$: a Boolean function. If processor i is high, then $High(i)=1$, and $High(i)=0$ if otherwise.
- $D(n,b)$: the diameter of $U(n,b)$.
- $DB(n,b,f)$: the bus-fault diameter of $U(n,b)$ with any f bus faults.
- $DF(n,b,f)$: the fault diameter of $U(n,b)$, where f is the sum of bus faults and processor faults.

4 Distance Properties of $U(n,b)$ where $b = n-1$

We shall present some distance properties of $U(n,b)$, with which we can route message efficiently. We discuss the simple case that $b = n-1$ in this section. The properties of $U(n,b)$ when $b < n-1$ can be easily extended and will be discussed in the next section.

Theorem 3 *In $U(n,b)$, $b = n-1$, $Zero(i,n)$ is even if and only if i is a high processor and $Zero(j,n)$ is odd if and only if j is a low processor.*

Theorem 4 *In $U(n,b)$, $b = n-1$, processor i=(c_{n-1}, c_{n-2},..., c_0) is connected to the buses*

(1)$(c_{n-1}, c_{n-2}, ..., c_1)$, which is the host bus of processor i, and

(2)$(c_{n-1}, c_{n-2}, ..., \overline{c_t}, ..., c_1)$, where $(n-t)$ is odd if i is a high processor, and $(n-t)$ is even if i is a low processor.

Suppose $i = (c_{n-1}, c_{n-2}, .., c_0)$ and $j = (d_{n-1}, d_{n-2}, ..., d_0)$. Let $R_{ij} = Bin(i,n) \bigoplus Bin(j,n) = (r_{n-1}, r_{n-2}, ..., r_0)$, where $r_x = c_x \bigoplus d_x$, $0 \leq x \leq n-1$.

Theorem 5 *In $U(n,b)$, $b = n-1$, let B denote the set of buses to which both processors $i=(c_{n-1}, c_{n-2}, ..., c_0)$ and $j=(d_{n-1}, d_{n-2}, ..., d_0)$ are connected, i.e. $B = B(i) \cap B(j)$. Each of the following is true.*

(1) $H(i,j)=1$ and $State(i)=State(j)=high$ if and only if $One(R_{ij})=2$, $r_s=r_t=1$, where $s > t$, and $(n-s)$ is odd, $(n-t)$ is odd or $t=0$, and $B=\{(c_{n-1}, c_{n-2}, .., \overline{c_s}, .., c_1)\} \cup B'$, where

$$B' = \begin{cases} (c_{n-1}, c_{n-2}, .., \overline{c_t}, .., c_1) & \text{if } t > 0. \\ (c_{n-1}, c_{n-2}, .., c_1) & \text{if } t = 0. \end{cases}$$

(2) $H(i,j)=1$ and $State(i)=State(j)=low$ if and only if $One(R_{ij})=2$, $r_s=r_t=1$, where $s > t$, and $(n-s)$ is even, $(n-t)$ is even or $t = 0$, and $B=\{(c_{n-1}, c_{n-2}, .., \overline{c_s}, .., c_1)\} \cup B'$, where

$$B' = \begin{cases} (c_{n-1}, c_{n-2}, .., \overline{c_t}, .., c_1) & \text{if } t > 0. \\ (c_{n-1}, c_{n-2}, .., c_1) & \text{if } t = 0. \end{cases}$$

(3) $H(i,j)=1$ and $State(i)=high$ and $State(j)=low$ if and only if one of the following two subcases are true.

(i) $One(R_{ij})=1$ and $r_s=1$, where $0 \leq s \leq n-1$, and $B =$

$$\begin{cases} (c_{n-1}, c_{n-2}, .., \overline{c_s}, .., c_1) & \text{if } (n-s) \text{ is odd, or} \\ (c_{n-1}, c_{n-2}, .., c_s, .., c_1) & \text{if } (n-s) \text{ is even or } s=0. \end{cases}$$

(ii) $One(R_{ij})=3$ and $r_0=r_s=r_t=1$, where $n-s$ is odd and $n-t$ is even, and $B = \{(c_{n-1}, c_{n-2}, .., \overline{c_s}, .., c_1)\}$.

Theorem 5 can be used to guide the routing from a source processor to a destination processor. If one of case (1) or case (2) of Theorem 5 is applied in one routing step, we call it is a *2-bit routing step*. If case (3)(i) and case (3)(ii) of Theorem 5 are applied in one routing step, we call them are *1-bit* and *3-bit routing steps* respectively.

To send messages from source i to destination j via one shortest path, we must select a neighbor processor from all connections of processor i properly. Before presenting how to select one best neighbor processor of the source processor, we give the distance between two processors.

Lemma 1 *In $U(n,b)$, $b = n-1$, the distance between processors i and j is $H(i,j) \leq \lfloor \frac{One(R_{ij})}{2} \rfloor + 1$.*

Lemma 2 *In $U(n,b)$, $b = n-1$, suppose i is the source processor and j is the destination processor. There exists a neighbor h of processor i, where $One(R_{hj}) \leq One(R_{ij})$, such that for any neighbor k of processor i, if $One(R_{ij}) \leq One(R_{kj})$, then $H(h,j) \leq H(k,j)$.*

Theorem 6 *In $U(n,b)$, $b = n-1$,*

(1) $H(i,j)=2$ if $Odd(R_{ij})=r_0=1$ and $Even(R_{ij})=0$ and $n + High(i) =$ even, or if $Even(R_{ij})= r_0=1$ and $Odd(R_{ij}) =0$ and $n + High(i) =$odd.

(2) $H(i,j) = \lceil \frac{Even(R_{ij})}{2} \rceil + \lceil \frac{Odd(R_{ij})}{2} \rceil$ if $r_0=0$.

(3) $H(i,j) = \lfloor \frac{Even(R_{ij})}{2} \rfloor + \lfloor \frac{Odd(R_{ij})}{2} \rfloor + 1$ if otherwise.

Theorem 7 *In $U(n,b)$, $b = n-1$, the diameter is $D(n,b) = \lceil \frac{n}{2} \rceil$ if $n \geq 3$, and $D(n,b)=n$ if $n \leq 2$.*

Since the case that $One(R_{ij}) \leq 3$ can be obtained in the proof of Theorem 6, we present only the case that $One(R_{ij}) > 3$ as follows. Our shortest path routing algorithm is described in the following corollary.

Corollary 1 *In $U(n,b)$, $b = n-1$, suppose i is the source, j is the destination processor and $One(R_{ij}) > 3$. The following rules can be used to select the best neighbor of processor i.*

(1) $r_0 = 0$: a 3-bit routing step, a 2-bit routing step, or a 1-bit routing step which changes bit x, where x is even if $Even(R_{ij})$ is odd and x is odd if $Odd(R_{ij})$ is odd.

(2) Case 2: $r_0 = 1$, $Even(R_{ij})$ and $Odd(R_{ij})$ are even: a 3-bit routing step, a 2-bit routing step, or a 1-bit routing step.

(3) $r_0 = 1$, $Even(R_{ij})$ and $Odd(R_{ij})$ are odd: a 3-bit routing step, or a 2-bit routing step excluding changing bit 0.

(4) $r_0=1$, $Even(R_{ij})$ is even and $Odd(R_{ij})$ is odd: a 3-bit routing step, a 2-bit routing step excluding bit 0, or a 1-bit routing step which changes an even bit.

(5) $r_0=1$, $Even(R_{ij})$is odd and $Odd(R_{ij})$ is even: a 3-bit routing step, a 2-bit routing step, or a 1-bit routing step which changes an odd bit.

5 Extension to an Arbitrary Value b of $U(n,b)$

In this section, we shall extend the properties of $U(n,b)$ to the case $0 \leq b \leq n-2$. By the constructing method of $U(n,b)$, we apply the fully doubling method in b times, and the partially doubling method in $(n-b-1)$ times. Therefore, there are 2^{n-b-1} $U(b+1,b)$'s in $U(n,b)$, and each $U(b+1,b)$ shares the common buses to each other. The processor i= $(c_{n-1}, c_{n-2},..., c_1, c_0)$ in $U(n,b)$ can be viewed as the processor i'=$(c_b, c_{b-1},..., c_1, c_0)$ in $U(b+1,b)$, and the bits $c_{n-1}, c_{n-2},..., c_{b+1}$ of processor i can be viewed as the binary index of the $U(b+1,b)$ which processor i belongs to. Hence, the extension method is to reassign the processor identifier by excluding bit $(b+1)$ through bit $(n-1)$, and all properties in the previous section can also be obtained.

By the above extension method, the diameter of $U(n,b)$ is the same as that of $U(b+1,b)$. Thus, we can extend Theorem 7 to the theorem as follow.

Theorem 8 *In $U(n,b)$, $0 \leq b \leq n-1$, the diameter is $D(n,b) = \lceil \frac{b+1}{2} \rceil$ if $b \geq 2$, and $D(n,b)$=$b+1$ if $b \leq 1$.*

6 Fault Tolerance of the Bus-based Hypercube

Hypercube networks have some good fault tolerance properties. Much interest has been paid on the hypercube network [2, 8, 10, 12]. Therefore, we are also interested in the fault tolerance properties of the bus-based hypercube network.

In the following, we shall list the lemmas and theorems, and omit their proofs due to the pages limitation.

Lemma 3 *In $U(n,b)$, $b = n-1$, let i and j are two processors. If there exist k paths whose buses are all distinct and the number of difference bits between every two adjacent buses on one path is one, then the k paths are processor disjoint.*

Theorem 9 *In $U(n,b)$, $b = n-1$, $b \geq 2$, there are $(m-1)$ bus disjoint and processor disjoint paths between any pair of processors where m is the minimun value of fan-outs of the two processors.*

Then, we have the following corollary.

Corollary 2 *In $U(n,b)$, $b = n-1$, $b \geq 2$, for all $0 \leq f \leq \lceil \frac{b-3}{2} \rceil$, the fault diameter $DF(n,b,f) \leq b+1$, where f is the sum of bus faults and processor faults.*

Theorem 10 *In $U(n,b)$, $b = n-1$, $b \geq 4$, there are m bus disjoint paths between any pair of processors where m is the minimun value of fan-outs of the two processors.*

For example, in $U(8,7)$, assume processors i =(00000000), j =(11111110) and k = (1111111). Then we have $B(i)$ = {0000000, 0000001, 0000100, 0010000, 1000000}, $B(j)$ = {1111111, 1111101, 1110111, 1011111}, and $B(k)$ = {1111111, 1111110, 1111011, 1101111, 0111111}. There are 4 bus disjoint paths between processor i and j. They are $B_0(i,j)$ = {B0000000, B0001010, B0101010, B1111010, B1111111}, $B_1(i,j)$ = {B0000001, B0101001, B0111101, B1111101}, $B_2(i,j)$ = {B0000100, B0100110, B0110111, B1110111}, and $B_3(i,j)$ = { B0010000, B0011010, B0011111, B1011111}. There are 5 bus disjoint paths between processors i and k. And they are $B_0(i,k)$ = {B0000000, B0001010, B0001111, B1011111, B1111111}, $B_1(i,k)$ = {B0000001, B0000011, B0000010, B0010110, B1010110, B1111110 }, $B_2(i,k)$ = {B0000100, B0001100, B0001000, B0011001, B1011001, B1111110 }, $B_3(i,k)$ = { B0010000, B0110000, B0100000, B0100101, B1100101, B1101111 }, and $B_4(i,k)$ = { B1000000, B1101000, B0101000, B0101101, B0111101, B0111111 }.

Then, we also have the following corollary.

Corollary 3 *In $U(n,b)$, $b = n-1$, $b \geq 4$, for all $0 \leq f \leq \lceil \frac{b-1}{2} \rceil$, the fault diameter $DB(n,b,f) \leq \lfloor \frac{b}{2} \rfloor + 3$, where f is the number of bus faults.*

Theorem 9 and Theorem 10 can be extended to the case $b < n-1$ easily. The reason is similar to Section 5. Therefore, by Theorem 9, the diameter $DF(n,b,k)$ in $U(n,b)$ is no more than $b+1$. Similarly, Theorem 10 is also true when $b \leq n-1$.

Theorem 11 *In $U(n,b)$, $1 \leq b \leq n-1$, $b \geq 2$, for all $0 \leq f \leq \lceil \frac{b-3}{2} \rceil$, the fault diameter $DF(n,b,f) \leq b+1$, where f is the sum of bus faults and processor faults.*

Theorem 12 *In $U(n,b)$, $1 \leq b \leq n-1$, $b \geq 4$, for all $0 \leq f \leq \lceil \frac{b-1}{2} \rceil$, the fault diameter $DB(n,b,f) \leq \lfloor \frac{b}{2} \rfloor + 3$, where f is the number of bus faults.*

Vaidyanathan and Padmanabhan showed that $U(n,b)$ is highly resilient to bus faults [16]. Thus, our result has a significant improvement on their result.

7 Conclusion

Given a processor i in $U(n,b)$, by the binary identifier of i, the state of i, high or low, can be determined. Then we obtain the buses which processor i

is connected to. Furthermore, the neighbor processors of i can also be obtained. From the properties of neighbor processors, we present a routing algorithm to transmit messages from the source processor to the destination processor via one shortest path, and the diameter is shown to be $\lceil \frac{b+1}{2} \rceil$ in $U(n,b)$. We also present a method to find $(m-1)$ bus disjoint and processor disjoint paths between the source processor and the destination processor, where m is the minimum value of fan-outs of the source processor and the destination processor. In $U(n,b)$, we show that the diameter $DF(n,b,f) \leq b+1$ where f is the sum of bus faults and processor faults and $0 \leq f \leq \lceil \frac{b-3}{2} \rceil$. We also show that $DB(n,b,f) \leq \lfloor \frac{b}{2} \rfloor + 3$, where $0 \leq f \leq \lceil \frac{b-1}{2} \rceil$ and f is the number of bus faults. The results are independent of the number of faults. Therefore, the result is better than that was proposed by Vaidyanathan and Padmanabhan [16].

References

[1] A. Ali and R. Vaidyanathan, "Exact bounds on running ASCEND/DESCEND and FAN-IN algorithms on synchronous multiple bus networks," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 7, No. 8, pp. 783–790, Aug. 1996.

[2] J. Bruck, R. Cypher, and C. Ho, "On the construction of fault-tolerant cube-connected cycles networks," *Journal of Parallel and Distributed Computing*, Vol. 25, pp. 98–106, 1995.

[3] O. M. Dighe, R. Vaidyanathan, and S. Zheng, "bus-based tree structure for efficeint parallel computation," *1993 International Conference on Parallel Processing*, pp. I–158 – 161, 1993.

[4] P. W. Dowd and K. Jabbour, "Spanning multiaccess channel hypercube computer interconnection," *IEEE Transactions on Computers*, Vol. 37, pp. 1137–1142, 1988.

[5] C. M. Fiduccia, "Bused hypercube and other pin-optimal networks," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 3, No. 1, pp. 14–24, Jan. 1992.

[6] T. Ishikawa, "Hypecube multilprocessor with bus connections for improving communication performance," *IEEE Transactions on Computers*, Vol. 44, No. 11, pp. 1338–1344, 1995.

[7] P. Kulasinghe and A. EI-Amawy, "Optimal realization of sets of interconnection functions on synchronous multiple bus systems," *IEEE Transactions on Computers*, Vol. 45, pp. 964–969, 1996.

[8] Y. Lan, "Adaptive fault-tolerant multicast in hypercube multiprocessors," *Journal of Parallel and Distributed Computing*, Vol. 23, pp. 80–93, 1994.

[9] T. Lang, M. Valero, and I. Alegre, "Bandwidth of crossbar and multiple-bus connections for multiprocessors," *IEEE Transactions on Computers*, Vol. C-31, No. 12, pp. 1227–1234, 1982.

[10] Y. Leu and S. Kuo, "A fault-tolerant tree communication scheme for hypercube systems," *IEEE Transactions on Computers*, Vol. 45, No. 6, pp. 641–650, June 1996.

[11] H. M.Alnuwieri, M. Alimuddin, and H. Aljunaidi, "Switch models and reconfigurable networks : Tutorial and partial survey," *Proc. First Workshop on Reconfigurable Architectures*, 1994.

[12] Y. Saad and M. H. Schultz, "Topological properties of hypercubes," *IEEE Transactions on Computers*, Vol. 37, No. 7, pp. 867–872, July 1988.

[13] J. L. Trahan, R. Vaidyanathan, and R. K. Thiruchelvan, "On the power of segmenting and fusing buses," *Journal of Parallel and Distributed Computing*, Vol. 34, pp. 82–94, 1996.

[14] N. F. Tzeng and S. Wei, "Enhance hypercubes," *IEEE Transactions on Computers*, Vol. 40, pp. 284–294, 1991.

[15] R. Vaidyanathan, "Sorting on prams with reconfiguable buses," *Information Processing Letters*, Vol. 42, pp. 203–208, June 1992.

[16] R. Vaidyanathan and A. Padmanabhan, "Bus-based networks for fan-in and uniform hypercube algorithms," *Parallel Computing*, Vol. 21, pp. 1807–1821, 1995.

[17] B. Wilkinson, "Multiple bus with overlapping connectivity," *IEE Proceedings-E*, Vol. 138, No. 4, pp. 281–284, July 1991.

[18] B. Wilkinson, "On crossbar switch and multiple bus interconnection networks with overlapping connectivity," *IEEE Transactions on Computers*, Vol. 41, No. 6, pp. 738–746, June 1992.

[19] T. Woo, Y. W. Su, and R. Newman-Wolfe, "Resource allcocation in a dynamically partitionable bus network using a graph coloring algorithm," *IEEE Transactions on Computers*, Vol. 39, No. 12, pp. 1794–1801, Dec. 1991.