JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING 14, 390–401 (1992)

Design of Gracefully Degradable Hypercube-Connected Systems*

TZE CHIANG LEE

IBM Corporation, T. J. Watson Research Center, P.O. Box 704, Yorktown Heights, New York 10598

JOHN P. HAYES

Advanced Computer Architecture Lab, Electrical Engineering & Computer Science Department, University of Michigan, Ann Arbor, Michigan 48109-2122

We address the problem of modifying a hypercube computer by the addition of spare nodes and links to improve its fault tolerance, while maintaining a specified level of performance. The hypercube is modeled by a graph in which nodes represent processors and edges represent communication links. A new graph-based measure of performance degradation is introduced. This characterizes a fault-tolerant hypercube as k-fault-tolerant (k-FT) g-step-degradable (g-SD) if the removal of any k nodes reduces the dimension of the largest fault-free subcube by at most g. We show how to construct k-FT g-SD hypercubes for values of k up to 16 and g = 0, 1, or 2. Many of these designs are shown to be link- or degree-optimal. We also propose a construction method that uses small k-FT g-SD designs as seeds to construct k-FT g-SD designs of larger sizes. This results in fault-tolerant hypercubes in which reconfiguration can be first done locally and then easily extended to the entire system. The small number of added links and nodes is shown to be useful not only in increasing the fault tolerance of the underlying hypercube, but also in reducing the average internode distance. © 1992 Academic Press, Inc.

1. INTRODUCTION

Since hypercube architectures are very regular and scalable, programs for these computers are frequently designed with the dimension n of the host hypercube Q_n treated as an input variable, so that the same program can be run without modification on hypercubes of different sizes. This scalability property of hypercube architecture makes it possible to tolerate faults gracefully by confining a program to a fault-free subcube of the host hypercube. The reduction of the effective cube size due to faults is a rough indication of the performance degradation. Based on this observation, we present a new characterization of fault-tolerant hypercube architectures that allows the performance degradation due to faults to be measured. We also develop some methods to modify Q_n by the addition of spare nodes and communication links in order to obtain designs whose performance degrades gracefully in the presence of node failures.

* This work was supported by the Office of Naval Research under Contracts N00014-85K0531 and N00014-90J1860.

Becker and Simon [2] have discussed the problem of finding the minimum number of faults needed to destroy every fault-free subcube Q_k, in Q_n, and give some analytic bounds on these numbers. Graham *et al.* [6] have improved the bounds and obtained exact values for some cases. These results show that for $n > k + 2$, Q_n is a good fault-tolerant realization of Q_k. For example, the numbers of faults required to destroy every fault-free Q_{n-3} and Q_{n-4} in Q_n, for $n \geq 6$, are at least 12 and 24, respectively. However, in the worst case, two faults in antipodal positions destroy every fault-free Q_{n-1} and thus degrade the maximum performance of Q_n by a factor of 4. This leads to an interesting question which is the subject of this paper: How can Q_n be modified efficiently so that its fault tolerance, measured by the number of faults required to destroy every fault-free Q_k, is increased?

We define $Q(n, k, g)$ to be a graph that contains Q_n and is such that the removal of any k nodes from it results in a graph containing Q_{n-g}. It is notationally convenient to use $Q(n, k, g)$ to denote the class of graphs of interest, as well as specific members of this class. The parameter k is the measure of fault tolerance, and we introduce the *dimension loss* g to measure the performance degradation. A $Q(n, k, g)$ is then termed a k-fault-tolerant g-step-degradable (k-FT g-SD) Q_n. The fault-tolerance design problem for hypercubes can now be stated as follows: given n, k and g, find a $Q(n, k, g)$ whose cost is a minimum. This cost may be measured by the number of nodes or links of $Q(n, k, g)$, or by its maximum node degree. Since Q_n itself is a good realization of a $Q(n, k, g)$ for $n \gg g \geq 3$, the cases where $g \leq 2$ are of most interest.

Previous research [1, 11, 12] on fault-tolerant hypercube design has, in terms of our notation, only considered the problem of constructing $Q(n, k, 0)$. Most of the proposed designs, however, suffer from excessive maximum node degrees. In this paper, we study the design of $Q(n, k, g)$, where $0 \leq g \leq 2$. We construct $Q(n, k, g)$ by adding a small, preferably minimum, number of extra nodes and links to Q_n. If a design with a minimum number of nodes and links is not attainable, near-optimal solutions are sought. Another concern is to minimize the

0743-7315/92 $3.00

maximum node degree of the modified hypercube. In a real computer system, the degree of a node represents the number of communication channels attached to it, and can significantly affect system cost. Furthermore, the degree of a node is limited by packaging constraints. Hence, in addition to minimizing the number of nodes and links, we also consider minimizing the maximum node degrees in our designs.

To facilitate the discussion, the following standard notation is used [7]. The minimum degree among the nodes of G is denoted by $\delta(G)$, while the largest node degree is denoted by $\Delta(G)$. If all nodes have the same degree, i.e., $\delta(G) = \Delta(G) = r$, then G is called regular of degree r. The connectivity $\kappa = \kappa(G)$ of a graph G is the minimum number of nodes whose removal results in a disconnected or trivial graph.

2. ZERO-STEP-DEGRADABLE DESIGNS

Yanney [12] presents several fault-tolerant hypercube designs which, in our terminology, correspond to $Q(n, k, 0)$ graphs. One of his approaches is based on the use of global spares, but unfortunately, is not appropriate for the fault-tolerant design of a large graph since it requires that the global spares connect to every node in the graph. It therefore results in an excessively high node degree for the spares. Rennels [11] has proposed the implementation of a global sparing method at the subcube level and the use of crossbar switches to reduce node degree. Yanney [12] also considers the use of local spares. A local spare can only be used to replace a subset of nodes in the graph. A fault-tolerant hypercube can be constructed if every node in the hypercube can be replaced by some spare node(s). However, this approach is also not appropriate for higher-dimensional hypercubes, since the maximum node degree is still intolerably high.

Banerjee *et al.* [1] develop a different $Q(n, 1, 0)$ design, for $n \geq 2$. This design uses 2^{n-2} spares, where one spare is added to each of the 2^{n-2} disjoint Q_2's. Although its hardware overhead is higher (25%) than designs using global spares, the spares in this structure increase the maximum node degree by only 2. However, this design is restricted to the 1-FT case.

In the following, we describe our solutions to the $Q(n, k, 0)$ design problem. The designs of $Q(n, k, 0)$ for small n are obtained by using power graphs of loops. Then we propose a systematic construction method that uses $Q(n, k, 0)$ with small n as a "seed" to construct $Q(n, k, 0)$ for large n. This construction method is based on the product operation on graphs, and can be used for $Q(n, k, g)$, where $g > 0$, as well. The maximum node degrees of our designs are relatively small compared to those of previous designs.

2.1. Minimum Node Designs

We now present several promising $Q(n, 1, 0)$ designs for small n. Figure 1 shows designs for $Q(1, 1, 0)$, $Q(2, 1, 0)$, $Q(3, 1, 0)$, and $Q(4, 1, 0)$, all of which use the minimum number of spare nodes. A $Q(n, k, 0)$ with the minimum number of spares is *degree-optimal* if the maximum node degree is the minimum possible. It can be easily seen that Fig. 1a is a degree-optimal $Q(1, 1, 0)$ design. In general, the complete r-node graph K_r is a degree-optimal $Q(1, r - 2, 0)$, for $r \geq 3$.

The foregoing $Q(n, k, 0)$ designs can all be interpreted as power graphs of loops (cycles). The mth *power graph* G^m of G is constructed by adding edges to G such that every node x is connected to all nodes at distance d from x, for $m \geq d \geq 2$. Let C_n denote a loop with n nodes. Figures 1a, 1b, 1c, and 1d demonstrate C_3^1, C_5^2, C_9^3, and C_{17}^5, respectively. Our use of power graphs of loops as fault-tolerant designs of Q_n is based on two facts: (1) Q_n is a subgraph of $C_{2^n}^{2^{n-2}}$; and (2) removing a node from $C_{2^n+1}^{2^{n-2}+1}$ results in a graph containing $C_{2^n}^{2^{n-2}}$, and hence containing Q_n. Furthermore, the removal of any k nodes from $C_{2^n+k}^{2^{n-2}+k}$ results in a graph containing $C_{2^n}^{2^{n-2}}$.

THEOREM 1. *$C_{2^n+k}^{2^{n-2}+k}$ is a $Q(n, k, 0)$ design.*

The reconfiguration of these $Q(n, k, 0)$ designs in the presence of faults is fairly simple. Suppose that we have a $C_{2^n+k}^{2^{n-2}+k}$ with t faults, where $t \leq k$. Let the nonfaulty nodes be labeled clockwise from 0 to $2^n + k - t - 1$. The subgraph composed of nodes 0 to $2^n - 1$ contains $C_{2^n}^{2^{n-2}}$.

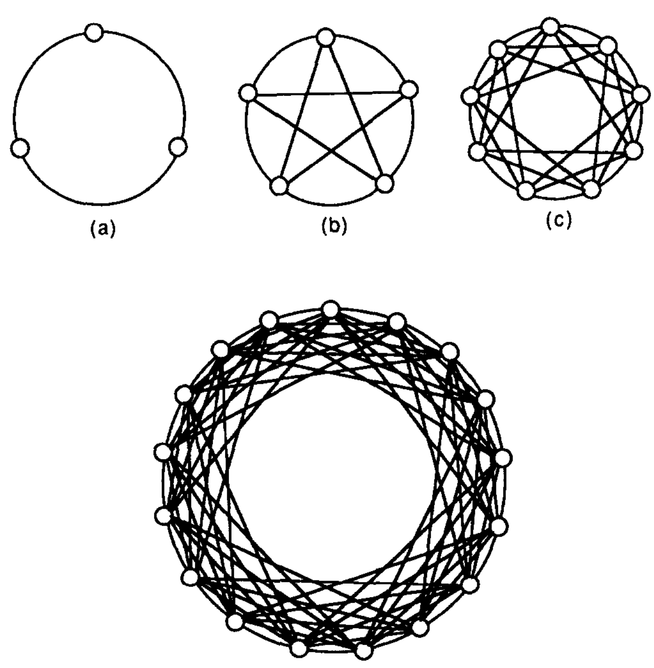

FIG. 1. The $Q(n, 1, 0)$ graphs: (a) C_3^1, (b) C_5^2, (c) C_9^3, and (d) C_{17}^5.

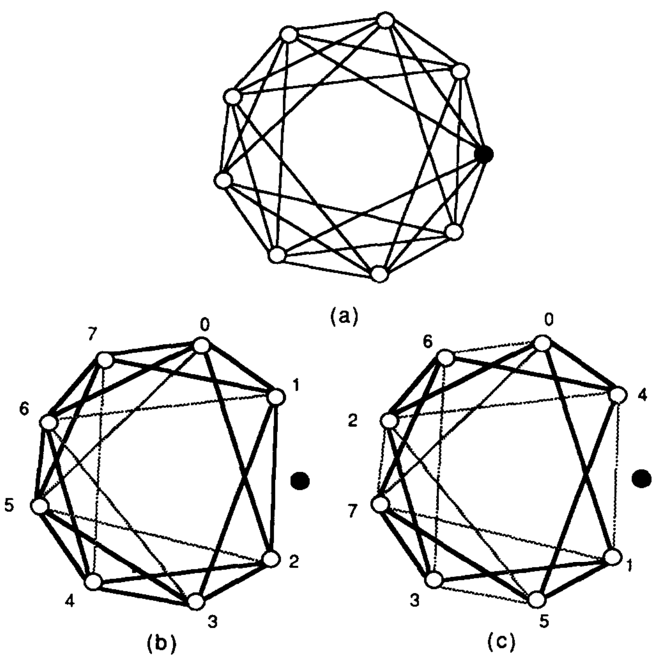

FIG. 2. (a) Proposed $Q(3, 1, 0)$ design with a faulty node. (b) Reconfigured C_8^3. (c) Fault-free Q_3.

Now we relabel each node x to construct a Q_n in $C_{2^n}^{2^{n-2}}$. Let $x = i \times 2^{n-2} + m$, for some $m \leq 2^{n-2}$. For each node x in $C_{2^n}^{2^{n-2}}$, x is relabeled as $2^{m+1} + y$, if $m \neq 0$, or as y, if $m = 0$, where y is 0, 1, 3, and 2 when i is 0, 1, 2, and 3, respectively. To illustrate, Fig. 2a shows $C_9^3 = Q(3, 1, 0)$, with a faulty (black) node present. Following the procedure described above, C_8^2 is recovered as shown in Fig. 2b. A fault-tree Q_3 is obtained accordingly by relabeling the nodes and is depicted in Fig. 2c.

Dutt and Hayes propose similar minimum node $Q(n, k, 0)$ designs using "circulant" graphs [5]. The node degree of their design is similar to ours when n is small, but is smaller when n becomes larger. This reduction in node degree is, however, at the cost of a more complicated reconfiguration process. In the following, we describe a construction method that effectively reduces the required node degree without increasing the complexity in reconfiguration.

2.2. *Graph-Product Method*

Using the power graph of a loop as a fault-tolerant realization of a hypercube has the merit that the resultant graph is regular, and only simple reconfiguration is needed to recover a fault-free Q_n. The number of spare nodes used is also a minimum. However, the maximum node degree increases quickly with the size of hypercube. For example, the 1-FT 0-SD realization of Q_6 by C_{65}^{17} has node degree 34. This may make the design impractical for large hypercubes. To address this problem,

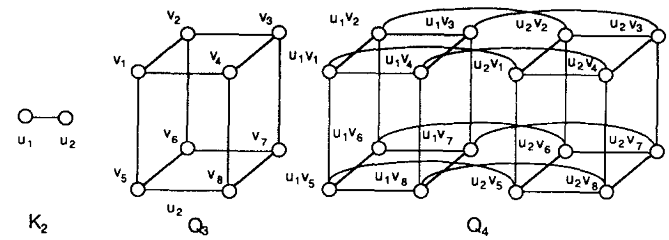

FIG. 3. The graph Q_4 as the product of K_2 and Q_3.

we introduce below a construction method that greatly reduces the maximum node degree by using more spare nodes than the minimum. Specifically, we show that $Q(n + 1, k, 0)$ can be constructed systematically from the product graph of K_2 and $Q(n, k, 0)$. This novel construction method is significant since it allows us to extend designs of smaller n, which are usually quite simple, to large n. Furthermore, as is shown later, fault-tolerant designs constructed by this method can be easily reconfigured when faults occur. This construction method also applies to more general $Q(n, k, g)$ designs, which are discussed in later sections.

The product of G_1 and G_2 is a graph $G_1 \times G_2$ with a node set V. Two nodes $u = (u_1, u_2)$ and $v = (v_1, v_2)$ in V are adjacent if $u_1 = v_1$ and u_2 is adjacent to v_2 in G_2, or if $u_2 = v_2$ and u_1 is adjacent to v_1 in G_1. Figure 3 shows the product of K_2 and Q_3. The resulting graph is, by definition, Q_4. The graph-product construction is characterized in the following theorem.

THEOREM 2. *Suppose that* $|G| \leq 2^{n+1}$. $G' = K_2 \times G$ *is a* $Q(n + 1, k, 0)$ *design if and only if* G *is a* $Q(n, k, 0)$ *design, for* $n, k \geq 2$.

Proof. (1) *If*: Let G' be as shown in Fig. 4, where G_1 and G_2 are two identical copies of G. Let nodes in G_1 be denoted by $x_1, x_2, \ldots,$ and x_m, and nodes in G_2 be denoted by $x'_1, x'_2, \ldots,$ and x'_m. Node x_i is connected to node x'_i, for i from 1 to m. Suppose that the faulty node set is $\{x_{f_1}, x_{f_2},$

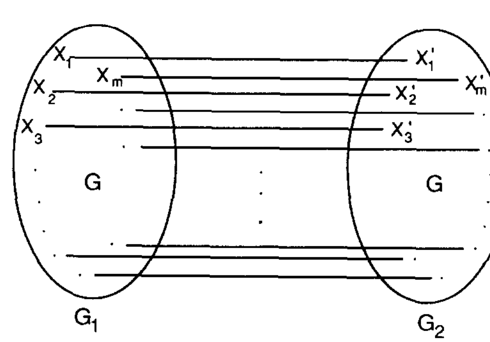

FIG. 4. G', a graph constructed as two copies of G.

$\ldots, x_{f_j}\} \cup \{x'_{f_{j+1}}, x'_{f_{j+2}}, \ldots, x'_{f_k}\}$. Now we generate a "pseudo-fault" set $\{x_{f_1}, x_{f_2}, \ldots, x_{f_j}\} \cup \{x_{f_{j+1}}, x_{f_{j+2}}, \ldots, x_{f_k}\}$, where $\{x_{f_{j+1}}, x_{f_{j+2}}, \ldots, x_{f_k}\}$ is the set of neighbors of $\{x'_{f_{j+1}}, x'_{f_{j+2}}, \ldots, x'_{f_k}\}$ in G_1. Since G is a $Q(n, k, 0)$ graph, we can always recover a Q_n from G_1 so that none of the nodes in the recovered Q_n belong to the pseudo-fault set. Also, in consequence of the definition of the pseudo-fault set, the neighbors in G_2 of the recovered Q_n in G_1 are nonfaulty. Thus combining the two Q_n's, we obtain a fault-free Q_{n+1} in G' for the given k faults.

(2) *Only if*: Suppose that G_1 is not a $Q(n, k, 0)$ graph. We show that k faults can destroy every fault-free Q_{n+1} in G'. We first find a fault set $F_1 = \{x_1, x_2, \ldots, x_k\}$ that destroys every fault-free Q_n in G_1. We can specify a second fault set $F_2 = \{a_1, a_2, \ldots, a_i, a'_{i+1}, \ldots, a'_k\}$, where $a_j = x_j$ and $a'_l = x'_l$, for some i, $1 \le j \le i < l \le k$. F_2 is obtained by moving some of the nodes in F_1 from G_1 to their direct neighbors in G_2. With fault set F_2 present, there can be no fault-free Q_{n+1} in either G_1 or G_2 since each contains only 2^{n+1} nodes. There is also no fault-free Q_{n+1} consisting of a fault-free Q_n in G_1 combined with a fault-free Q_n in G_2.

We claim that there are no other configurations of fault-free Q_{n+1}'s in G' under fault set F_2. To prove this, first suppose that there exists a fault-free Q_{n+1} in the faulty G' that consists of a set S_1 of i nodes in G_1, and a set S_2 of $2^{n+1} - i$ nodes in G_2. Neither S_1 nor S_2 forms a Q_n. Each node in S_1 has at most one link connected to S_2, since each node in G_1 has only one link connected to G_2. At least two nodes in S_1 are connected to nodes in S_2 for $n \ge 2$, since Q_n is Hamiltonian.

Now we show that every node in S_1 must have one link connected to a node in S_2. Suppose that this is not true. It is easy to find a node x in S_1 that connects to a node x' in S_2, and has a neighbor x_1 in S_1 that has no neighbor in S_2. Let the dimension between x and x' be m, and that between x and x_1 be j. We can also find a node x'_1 in S_2 connecting to x' along dimension j, hence x', x, x_1, and x'_1 must form a Q_2. This violates the assumption that x_1 has no neighbor in S_2. Hence each node in S_1 must have a neighbor in S_2. This fact also implies $|S_1| = |S_2| = 2^n$.

Again assume that x in S_1 connects to node x' in S_2 along dimension m. Since neither S_1 nor S_2 forms a Q_n and each node in S_1 connects to a node in S_2, we can find a neighboring node x_1 of x in S_1 such that x_1 connects to x along dimension j, and x_1 connects to node x'_1 in S_2 along a dimension other than m and j, say l. Let x_2 be the neighboring node of x in S_1 along dimension l. It is apparent that x_2 should connect to x'_1 along dimension j. But x'_1 is already connected to x_1, which leads to a contradiction. Hence S_1 and S_2 must form Q_n's and G_1 and G_2 are required to be $Q(n, k, 0)$ graphs. ∎

This theorem implies that we can use $Q(n, k, 0)$ designs of small n as seeds to build up $Q(n, k, 0)$ designs of large n. Figure 5a illustrates a $Q(4, 1, 0)$ that is constructed using $Q(3, 1, 0)$ taken from Fig. 1c as a seed. Larger designs are obtained by applying the graph-product operations several times. Suppose that a $Q(n_1, k, 0)$ is used as a seed and its maximum node degree is d. The maximum node degree of a $Q(n, k, 0)$ produced by the foregoing construction method is then $d + n - n_1$, which is considerably smaller than that of $Q(n, k, 0)$ developed using other methods. The number of nodes and links of the proposed designs and some of their topological properties are listed in Table I. As can be seen, the designs span a broad spectrum with different amounts of redundancy and maximum node degrees.

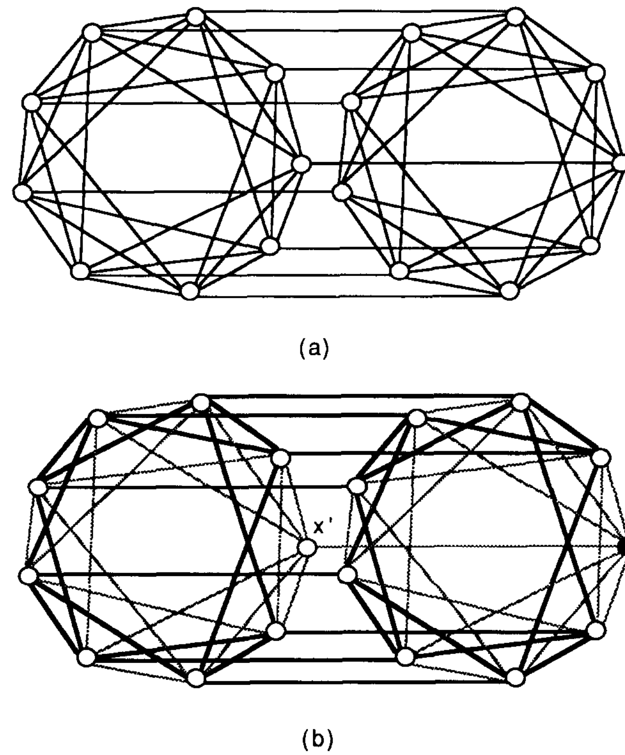

FIG. 5. (a) A $Q(4, 1, 0)$ constructed as the product of K_2 and $Q(3, 1, 0)$. (b) A recovered fault-free Q_4 in the presence of a faulty node x.

Another advantage of the $Q(n, k, 0)$ designs constructed by the graph-product operation is that reconfiguration around faults is fairly simple and efficient. It can first be done locally within a seed and then applied to the whole structure. Suppose that $F = \{x_1, x_2, \ldots, x_j\}$ is the fault set. We generate a pseudo-fault set F' by projecting the faults into a seed $Q(n_1, k, 0)$. Taking the case in Fig. 5b as an example, the product graph $Q(4, 1, 0)$ contains two copies of $Q(3, 1, 0)$ which serve as seeds. We project the faulty node x into the left seed and create a pseudo-fault x'. Now we can easily find a pseudo-fault-tree Q_3 in that seed using the method described in Section 2.1. It is clear that the corresponding Q_3 in the other seed of the $Q(4, 1, 0)$ is also fault-free. Together, the two Q_3's form a fault-free Q_4.

TABLE I
Topological Properties of Some $Q(n, k, 0)$ Designs Constructed by the Graph Product Method

Graph	Seed	Number of nodes	Maximum node degree	Diameter	Number of links
$Q(n, 1, 0)$	$Q(1, 1, 0)$	$2^n + 2^{n-1}$	$n + 1$	n	$3 \times 2^{n-2}(n + 1)$
$Q(n, 1, 0)$	$Q(2, 1, 0)$	$2^n + 2^{n-2}$	$n + 2$	$n - 1$	$5 \times 2^{n-3}(n + 2)$
$Q(n, 1, 0)$	$Q(3, 1, 0)$	$2^n + 2^{n-3}$	$n + 3$	$n - 1$	$9 \times 2^{n-4}(n + 3)$
$Q(n, 1, 0)$	$Q(4, 1, 0)$	$2^n + 2^{n-4}$	$n + 6$	$n - 2$	$17 \times 2^{n-5}(n + 6)$
$Q(n, 2, 0)$	$Q(1, 2, 0)$	$2^n + 2 \times 2^{n-1}$	$n + 2$	n	$4 \times 2^{n-2}(n + 2)$
$Q(n, 2, 0)$	$Q(2, 2, 0)$	$2^n + 2 \times 2^{n-2}$	$n + 3$	$n - 1$	$6 \times 2^{n-3}(n + 3)$
$Q(n, 2, 0)$	$Q(3, 2, 0)$	$2^n + 2 \times 2^{n-3}$	$n + 5$	$n - 1$	$10 \times 2^{n-4}(n + 5)$
$Q(n, 2, 0)$	$Q(4, 2, 0)$	$2^n + 2 \times 2^{n-4}$	$n + 8$	$n - 2$	$18 \times 2^{n-5}(n + 8)$
$Q(n, 3, 0)$	$Q(1, 3, 0)$	$2^n + 3 \times 2^{n-1}$	$n + 3$	n	$5 \times 2^{n-2}(n + 3)$
$Q(n, 3, 0)$	$Q(3, 3, 0)$	$2^n + 3 \times 2^{n-3}$	$n + 7$	$n - 2$	$11 \times 2^{n-4}(n + 7)$
$Q(n, 3, 0)$	$Q(4, 3, 0)$	$2^n + 3 \times 2^{n-4}$	$n + 10$	$n - 2$	$19 \times 2^{n-5}(n + 10)$

3. ONE-STEP-DEGRADABLE DESIGNS

Q_n can tolerate any single fault while maintaining a fault-free subgraph Q_{n-1}. However, two faults in antipodal positions destroy every Q_{n-1} in Q_n. Figure 6 demonstrates the situation where two faulty nodes 0000 and 1111 in Q_4 destroy every Q_3. Therefore Q_n is itself a $Q(n, 1, 1)$ design. In this section, we discuss the design of k-FT 1-SD hypercubes by adding redundant links to the basic Q_n. Our approach is different from that in the previous section where both nodes and links are added to the underlying hypercube structures.

Let Q_n^+ denote an n-dimensional hypercube with extra links. Q_n^+ is said to be *k-fault-tolerant g-step-degradable* if $Q_n^+ - F$ contains Q_{n-g}, where F is any set of k faulty nodes. A k-FT g-SD Q_n^+ is denoted by $Q^+(n, k, g)$. Thus $Q^+(n, k, g)$ is a $Q(n, k, g)$ graph by definition. We construct $Q^+(n, k, 1)$ by adding a small, preferably minimum, number of extra links to Q_n. If a graph with a minimum number of links is not attainable, near-optimal solutions will be sought. Also, for practical reasons, we are interested in finding designs with low maximum node degrees.

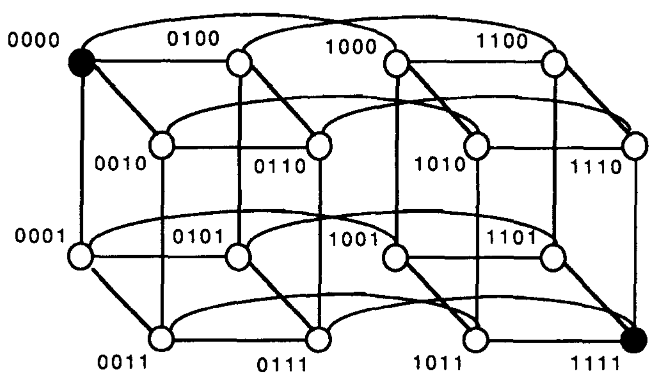

FIG. 6. Example where two node failures destroy every fault-free Q_3 in Q_4.

3.1. $Q^+(n, k, 1)$ Designs for $2 \leq k \leq 4$

We start with designs with small values of n. The following lemma sets lower bounds for the number of links in link-optimal designs for $Q^+(2, 2, 1)$ and $Q^+(3, 2, 1)$.

LEMMA 1. *Link-optimal $Q^+(2, 2, 1)$ and $Q^+(3, 2, 1)$ graphs contain at least 6 and 14 links, respectively.*

Proof. We prove the lemma by considering the minimum number of links that must be added to Q_2 and Q_3 to form $Q^+(2, 2, 1)$ and $Q^+(3, 2, 1)$, respectively. In each case, the number of extra links needed is greater than one. Suppose that, by way of contradiction, only one link is added, and let one of its end nodes be x. We can destroy every fault-free Q_2 by removing nodes x and y, where y is at distance two (three in the Q_3 case) from x. Hence the minimum number of extra links required is two for each of the designs. Since Q_2 and Q_3 have 4 and 12 links, respectively, the minimum numbers of links required for $Q^+(2, 2, 1)$ and $Q^+(3, 2, 1)$ are 6 and 14, respectively. ■

Next, we present $Q^+(2, 2, 1)$ and $Q^+(3, 2, 1)$ with two extra links in each case. Both of these designs are link-optimal by Lemma 1.

THEOREM 3. *The graphs $S = K_4$ and W shown in Fig. 7 are link-optimal realizations of $Q^+(2, 2, 1)$ and $Q^+(3, 2, 1)$, respectively.*

Proof. (1) K_4 clearly implements $Q^+(2, 2, 1)$. K_4 has two links more than Q_2; therefore, by Lemma 1, this design is link-optimal.

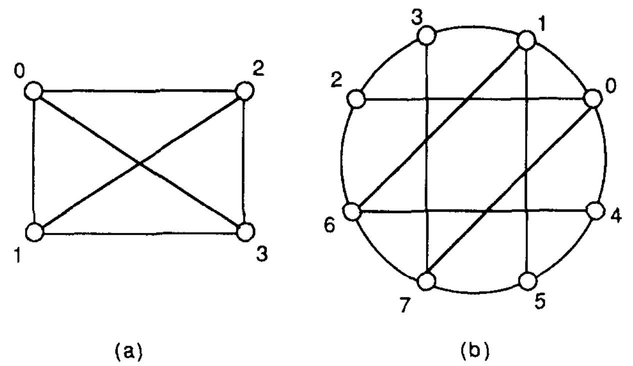

FIG. 7. (a) A link-optimal $Q^+(2, 2, 1)$ graph $S = K_4$. (b) A link-optimal $Q^+(3, 2, 1)$ graph W.

(2) Removing from W any two nodes that are distance two or less apart in the underlying Q_3 will not destroy all fault-free Q_2's in W. Hence there are four pairs of nodes, {0, 7}, {3, 4}, {1, 6}, and {2, 5}, whose removal may destroy each Q_2. We consider the cases for {0, 7} and {3, 4}; the cases for {1, 6} and {2, 5} follow directly from symmetry. The two maximum subgraphs obtained after each node set is removed from W are shown in Fig. 8; the corresponding Q_2's recovered are shown in heavy lines. Hence W is a $Q^+(3, 2, 1)$ design. In addition, since the number of links added to the basic graph Q_3 is two, W is link-optimal by Lemma 1. ■

Both the W and the S designs have the merits of using a minimum number of links and increasing the maximum node degree by one only. W is not regular, however. We next describe two regular and node-symmetric graphs that are also $Q^+(3, 2, 1)$ graphs with the same maximum node degree as W. These two regular graphs also reveal the fact that a degree-optimal $Q^+(n, k, 1)$ is not unique.

LEMMA 2. *Graphs R and U defined in Fig. 9 are degree-optimal regular realizations of $Q^+(3, 2, 1)$.*

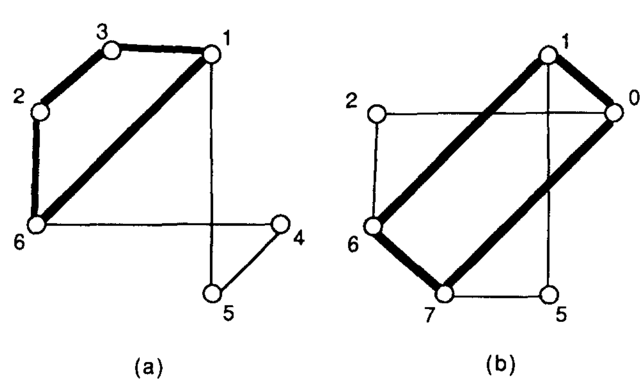

FIG. 8. The maximum subgraphs after removing node sets {0, 7} and {3, 4} from W. The recovered Q_2's are shown in heavy lines.

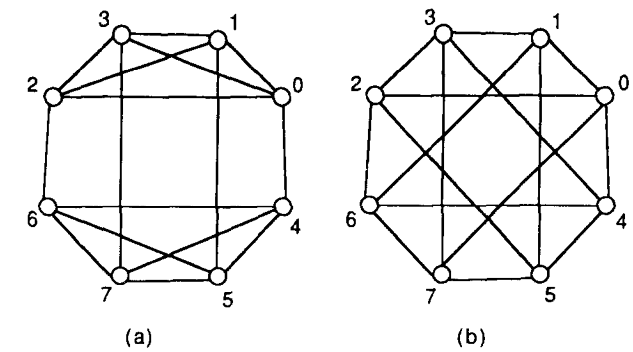

FIG. 9. Two regular degree-optimal $Q^+(3, 2, 1)$'s (a) R and (b) U.

Proof. W (Fig. 7b) is a subgraph of U, hence U is a $Q^+(3, 2, 1)$ graph. The proof for R has two parts: First, if the two faulty nodes in R are not distance three apart in the embedded Q_3, the fault-tolerance property of Q_3 guarantees the existence of a fault-free Q_2. Second, suppose that the two faulty nodes are distance three apart in the embedded Q_3. Without loss of generality, choose nodes 0 and 7 as the faulty nodes. The recovered fault-free Q_2 after nodes 0 and 7 are removed is shown in Fig. 10. The other cases follow from symmetry. ■

In the following, we develop the two fault-tolerant designs $B = Q^+(3, 3, 1)$ and $D = Q^+(3, 4, 1)$ shown in Fig. 11. To prove the desired fault-tolerance properties of B and D, we need the following lemmas.

LEMMA 3. *A four-node connected graph G with $\delta(G) \geq 2$ is Hamiltonian.*

Proof. Let the four nodes be x_1, x_2, x_3, and x_4. If x_1 connects to both x_2 and x_3, and node x_4 is connected to x_2 and x_3, then the four nodes form a cycle. Hence G is Hamiltonian. Other alternatives are that x_4 is connected to x_2 and x_1, or to x_3 and x_1. Consider the case where x_4 is connected to x_3 and x_1. Then x_2 must be connected to either x_3 or x_4 since x_2's node degree is at least two. In either case, there is a cycle of length four in G; thus G is Hamiltonian. The case where x_4 is connected to x_2 and x_1 follows the same reasoning. ■

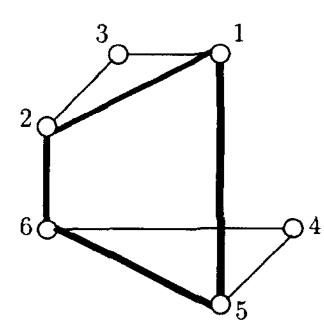

FIG. 10. The recovered Q_2 in R with nodes 0 and 7 removed.

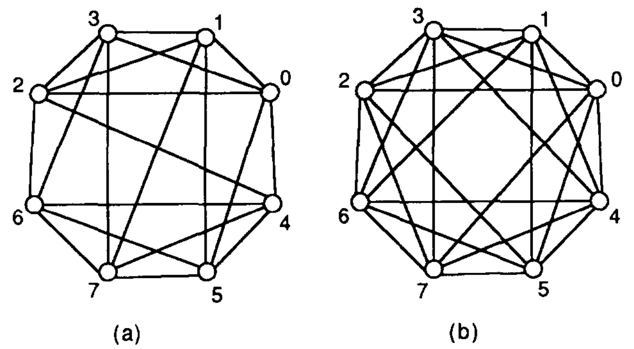

FIG. 11. (a) The $Q^{+}(3, 3, 1)$ graph B. (b) The $Q^{+}(3, 4, 1)$ graph D.

LEMMA 4. *A five-node connected graph G contains Q_2 if $\delta(G) \geq 2$ and at least two of its nodes have degree three or more.*

Proof. Let C_i denote a cycle with i nodes. Suppose that G does not contain Q_2. Then G must contain either C_5 or C_3 since $\delta(G) \geq 2$. Consider the case where G contains C_5. Since two nodes must have degree greater than two, at least one more link not in C_5 must exist. This links must connect two nodes of C_5 resulting in a graph containing $C_4 = Q_2$, hence G contains Q_2. Now consider the case where G contains C_3 but not C_5. Since there are five nodes in G, there exist at least two C_3's in G. The intersection of two C_3's can include one or two nodes, depending on whether the C_3's share a link. When this intersection contains exactly two nodes, then the non-shared links form a Q_2. Suppose that the intersection is a single node, say x. The configuration is shown in Fig. 12. In this case, there is only one node of degree greater than two, hence a link must be added to the graph to meet our initial assumptions for G. It is obvious that adding a link between any nonadjacent nodes results in a graph containing Q_2 as a subgraph. Hence the lemma follows. ∎

Since both B and D contain only eight nodes, it is easy to verify by exhaustion that there exist five (six) node-disjoint paths between any pair of nodes in B (D). By a variation of Menger's Theorem [7], $\kappa(B) \geq 5$ and $\kappa(D) \geq 6$. Therefore removing any three (four) nodes from B (D) results in a connected subgraph.

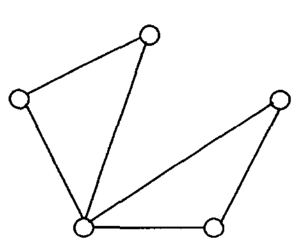

FIG. 12. Two triangles intersect at one node.

Now we prove the basic fault-tolerance properties of B and D using Lemmas 3 and 4.

THEOREM 4. *Graphs B and D defined in Fig. 11 realize $Q^{+}(3, 3, 1)$ and $Q^{+}(3, 4, 1)$, respectively.*

Proof.

Case 1. $\delta(B) = \Delta(B) = 5$. The number of links in B is 20. Let $N(x)$ denote the set of neighboring nodes of x. We observe that for any three nodes x, y, and z in B, $|N(x) \cup N(y) \cup N(z)| \geq 7$. This implies at least two of any three nodes in B are directly connected. As a result, the removal of three nodes can remove at most 14 links from B and produces a graph B' which contains five nodes with no less than 6 links. Note that $\delta(B') \geq 2$ since $\delta(B) = 5$. We now claim that at least two nodes in B' have degree three or more. Suppose that it is not the case. Then there are only three possibilities to be considered: first, the node degree of each node in B' is two; second, four nodes in B' have degree two and one node has degree three; and third, four nodes have degree two and one node has degree four. The first case is ruled out since it requires B' to have only five links. The second is also impossible since there exists a single node of odd degree. The third implies that the three removed nodes have four common neighbors in B. However, one can quickly verify that for any three nodes x, y, and z in B, $|N(x) \cap N(y) \cap N(z)| \leq 3$. The third case is then impossible, hence the claim is true. By Lemma 4, B' contains Q_2, and accordingly B is a $Q^{+}(3, 3, 1)$ graph.

Case 2. $\delta(D) = \Delta(D) = 6$. The removal of any four nodes from D results in a four-node connected graph D' with $\delta(D')$ at least two. Therefore by Lemma 3, D' contains Q_2. ∎

The minimum node degree of any $Q^{+}(3, 4, 1)$ is six. For if the minimum node degree is less than six, we can easily remove four nodes and make the resulting four-node graph contain a node of degree less than two. In that case, it is impossible to form a Q_2. The maximum node degree of D is six, hence we conclude that D is degree-optimal.

3.2. $Q^{+}(n, k, 1)$ *Designs for* $5 \leq k \leq 16$, $n \leq 5$

On examining the designs in the previous section, we see that a $Q^{+}(n, k + 1, 1)$ graph can, in general, be constructed by adding some links to a $Q^{+}(n, k, 1)$ design. For instance, the $Q^{+}(3, 4, 1)$ graph D in Fig. 11b is a supergraph of the $Q^{+}(3, 3, 1)$ graph B in Fig. 11a, and the $Q^{+}(3, 3, 1)$ graph B is a supergraph of the $Q^{+}(3, 2, 1)$ graph R shown in Fig. 9. Based on this observation, we now devise a procedure to generate $Q^{+}(n, k, 1)$ for more

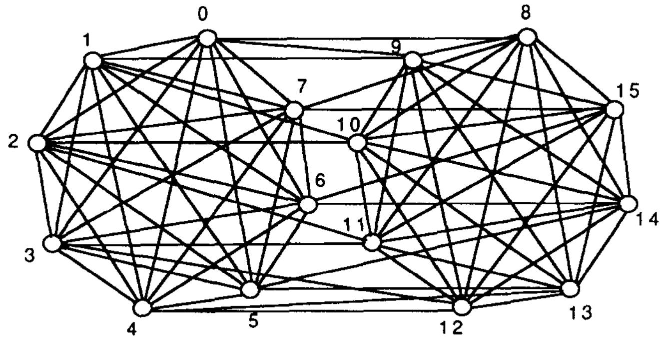

FIG. 13. A $Q^+(4, 5, 1)$ graph generated by ADD_LINKS.

general cases. The algorithm ADD_LINKS given below uses two copies of $K_{2^{n-1}}$, and adds some links between them in a systematic way. Since $K_{2^{n-1}}$ is also a $Q(n-1, 2^{n-2}, 1)$ graph, by connecting the corresponding nodes in the two $K_{2^{n-1}}$'s, a $Q(n, 2^{n-2}, 1)$ is formed, which is a good starting point to construct a $Q(n, 2^{n-2} + l, 1)$, where $1 \le l \le 2^{n-2}$.

ALGORITHM **ADD_LINKS:** {To generate a 1-SD Q_n graph $Q(n, 2^{n-2} + l, 1)$}

(1) Construct two copies of $K_{2^{n-1}}$, denoted S_1 and S_2. Nodes in S_1 are labeled from 0 to $2^{n-1} - 1$; nodes in S_2 are labeled from 2^{n-1} to $2^n - 1$.

(2) Connect each node i in S_1 with all nodes j in S_2, where $j \equiv i + m \pmod{2^{n-1}}$, and $0 \le m \le l$.

Figure 13 demonstrates the $Q^+(4, 5, 1)$ graph generated by ADD_LINKS. Nodes 0, 1, 2, 3, 4, 5, 6, and 7 form K_8, and nodes 8, 9, 10, 11, 12, 13, 14, and 15 form another copy of K_8. Node 0 connects to nodes 8 and 9, node 1 connects to nodes 9 and 10, and so on.

The following are some properties of the interconnection between S_1 and S_2 that are useful in later proofs. Let a be a node in S_1 in a given $Q(n, k, 1)$ graph, where $k = 2^{n-2} + l$. $N(a)$ is the neighboring node set of a in S_2; $NN(a)$ is the fault-free neighboring node set of a in S_2.

PROPERTY P1. $|N(a)| = l + 1$.

PROPERTY P2. *For any given i node in* S_1, $l + i \le |\bigcup^i N(a)| \le 2^{n-1}$.

PROPERTY P3. *Suppose that there are j faults in* S_1 *and* $2^{n-2} + l - j$ *faults in* S_2, $l \le j \le 2^{n-2}$. *For any given i nonfaulty nodes in* S_1, $1 \le i \le 2^{n-2}$, *we have* $|\bigcup^i NN(a)| - (i + j - 2^{n-2}) \ge (l + i) - (2^{n-2} + l - j) - (i + j - 2^{n-2}) = 0$.

Properties P1 and P2 can be easily verified from the procedure itself. P3 states that for any i nonfaulty nodes in S_1 that contain j faults, the union of the fault-free neighboring sets of these i nodes contains at least $i + j - 2^{n-2}$ elements. This property can be derived from P2.

The proofs of several of our results involve some techniques from the theory of systems of distinct representatives (SDRs), which is concerned with finding a distinct representative for each set in a collection of sets. We briefly summarize here the key results we need, following the presentation of [4]. Let B be an arbitrary set and let B_i be a subset of B. A set $\{B_1, B_2, \ldots, B_j\}$ is said to have a *system of distinct representatives* if there exists b_i, $i = 1, 2, \ldots, j$, such that b_i belongs to B_i, and $b_i \ne b_j$ if $i \ne j$.

Condition C1. A set $\{B_1, B_2, \ldots, B_j\}$ is said to satisfy Condition C1 if $B_{i_1} \cup B_{i_2} \cup \ldots \cup B_{i_k}$ contains at least k elements for $k = 1, 2, \ldots, j$, and for each choice of $i_1, i_2, \ldots, i_k$ with $1 \le i_1 < i_2 < \ldots < i_k \le j$.

Condition C2. Let r be a positive integer with $r \le j$. A set $\{B_1, B_2, \ldots, B_j\}$ is said to satisfy Condition C2 if $B_{i_1} \cup B_{i_2} \cup \ldots \cup B_{i_k}$ contains at least $k - (j - r)$ elements for $k = 1, 2, \ldots, j$, and for each choice of $i_1, i_2, \ldots, i_k$ with $1 \le i_1 < i_2 < \ldots < i_k \le j$.

THEOREM 5 (Hall's Theorem). $\{B_1, B_2, \ldots, B_j\}$ *has an SDR if and only if it satisfies Condition C1.*

COROLLARY 1. *Let r be a positive integer with* $r \le j$. *Then there are r sets in the family* $\{B_1, B_2, \ldots, B_j\}$ *which together have an SDR if and only if* $\{B_1, B_2, \ldots, B_j\}$ *satisfies Condition C2.*

Next we show that Algorithm ADD_LINKS can generate $Q^+(n, k, 1)$ correctly for $5 \le k \le 16$.

THEOREM 6. *ADD_LINKS generates* $Q^+(n, k, 1)$, *for* $5 \le k \le 16$, $n \le 5$.

The proof of this theorem is rather lengthy and can be found in [9]. We sketch the proof for a representative case ($k = 10$) here to illustrate how ADD_LINKS is able to generate $Q^+(5, 10, 1)$. In this case, the generated $Q(5, 10, 1)$ contains two copies S_1 and S_2 of K_{16}. Each node x in S_1 connects to three nodes in S_2 and vice versa. Since a $Q(5, 10, 1)$ graph constructed this way always contains $Q(5, 9, 1)$ as subgraph, we will focus on the cases where there are 10 faults in the system.

Suppose that all 10 faults lie either in S_1 or else in S_2. Since S_1 and S_2 are copies of K_{16}, it is clear that either S_2 or S_1 should contain a fault-free Q_4 in this case. Now suppose nine faults are in S_1 and one in S_2. We can find one nonfaulty node a in S_1 which connects to three nonfaulty nodes in S_2. The subgraph induced by node a and the nonfaulty nodes in S_2 contains a fault-free Q_4.

Next, suppose there are j faults in S_1 and $10 - j$ faults in S_2, $j \ge 2$. Let a denote a fault-free node in S_1. By Property P3 above, $|\bigcup^i NN(a)| - (i + 10 - 8) = |\bigcup^i NN(a)| - i + 2 \ge 0$, for any given i nonfaulty nodes in S_1, $1 \le i \le 16 - j$. Therefore by Corollary 1, we can find eight nonfaulty nodes in S_1 each of which has a distinct nonfaulty neighbor in S_2. The graph induced by these

TABLE II
The Increase in the Node Degree in the $Q(n, k, 1)$ Graphs Generated by ADD_LINKS, $2 \le k \le 16$

Fault tolerance k:	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Increased degree:	1	2	3	5	6	7	8	12	13	14	15	16	17	18	19

nodes and their distinct nonfaulty neighbors contains a fault-free Q_4. The cases for other fault patterns follow by symmetry.

ADD_LINKS is not limited to generating the design specified in Theorem 6; it can also generate $Q^+(n, k, 1)$ for some larger k's, as stated in the next theorem. This leads us to the conjecture that an ADD_LINKS-like algorithm can generate $Q^+(n, k, 1)$ for arbitrary k.

THEOREM 7. *ADD_LINKS generates* $Q^+(n, 2^{n-2} + 1, 1)$, $n \ge 2$.

Proof.

Case 1. *There are no more than* 2^{n-2} *faults.* The graph created by ADD_LINKS contains $K_2 \times K_{2^n}$, which is a realization of $Q^+(n, 2^{n-2}, 1)$. Therefore if the number of faults does not exceed 2^{n-2}, a fault-free Q_{n-1} is present.

Case 2. *There are* $2^{n-2} + 1$ *faults, all of which lie in either* S_1 *or* S_2. In this case, S_2 or S_1 contains a fault-free Q_4.

Case 3. *There are* j *faults in* S_1 *and* $2^{n-2} + l - j$ *faults in* S_2. Given i nonfaulty nodes in S_1, by Property P3, $|\bigcup^i NN(a)| - (i + k - 2^{n-2}) \le 0$. By Corollary 1, we can find 2^{n-2} nonfaulty nodes in S_1, each of which has a nonfaulty distinct neighboring node in S_2. The 2^{n-2} nodes in S_1 form a Q_{n-2} and so do their corresponding nonfaulty neighbors in S_2. Together, these two Q_{n-2}'s form a fault-free Q_{n-1}. All other cases follow by symmetry. ■

The node degree of the $Q(n, k, 1)$ graphs generated by this procedure is $n + (2^d + k - d - 2)$, where $2^d < k \le 2^{d+1}$. This node degree increases by $2^d + k - d - 2$ compared to Q_n. Table II lists the increase in the node degree of the $Q(n, k, 1)$ graphs, where $2 \le k \le 16$. It shows that the node degree increases at a moderate rate with the fault-tolerance parameter k.

3.3. $Q^+(n, k, 1)$ Designs for Large n

The discussion so far has focused on $Q^+(n, k, 1)$ for small n, e.g., $n \le 5$. As in the $Q(n, k, 0)$ case, these smaller designs can be used to construct larger ones by the product method, as characterized in the next corollary. The proof of this result is similar to that of Theorem 2 and thus omitted.

COROLLARY 2. $G' = K_2 \times G$ *is a* $Q^+(n, k, 1)$ *graph if and only if* G *is a* $Q^+(n - 1, k, 1)$ *graph, for* $n, k \ge 2$.

The graph-product construction preserves the fault-tolerance property of the seed graph. Suppose that $G \in \{R, U, B, D\}$; we define G_n recursively as follows: $G_3 = G$, and $G_{n+1} = K_2 \times G_n$. By Corollary 2, Lemma 2, and Theorem 4, R_n, U_n, B_n, and D_n are $Q^+(n, 2, 1)$, $Q^+(n, 2, 1)$, $Q^+(n, 3, 1)$, and $Q^+(n, 4, 1)$ graphs, respectively. They are illustrated for $n = 4$ in Fig. 14. A great advantage of this construction method is that the increase in the node degree of a design compared to Q_n is the same

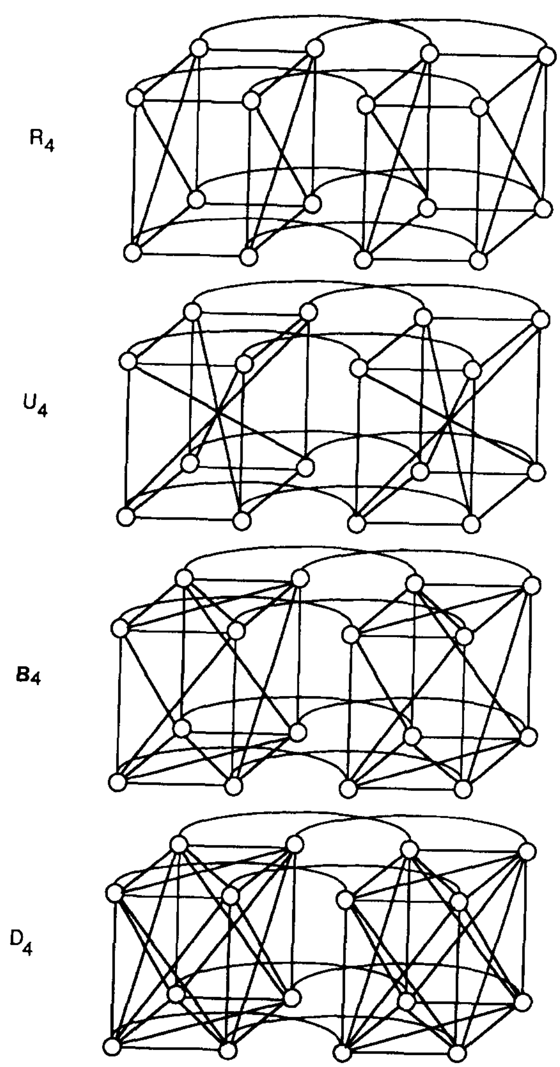

FIG. 14. Examples of $Q^+(4, k, 1)$ graphs: R_4 and U_4 are $Q^+(4, 2, 1)$ designs; B_4 and D_4 are $Q^+(4, 3, 1)$ and $Q^+(4, 4, 1)$ designs, respectively.

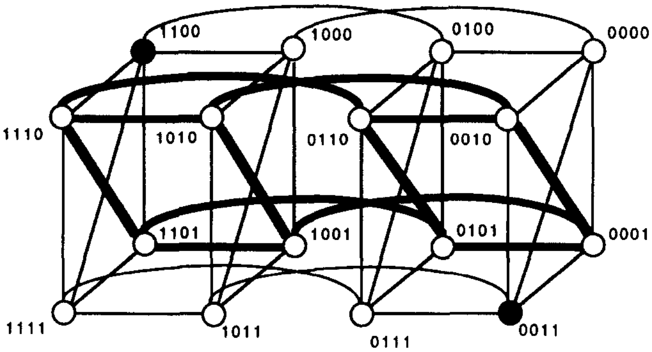

FIG. 15. The recovered Q_3 (heavy lines) in a faulty R_4.

as that of its seed. For instance, the node degree of B_4 is six. The increase in the node degree is two compared to Q_4, and is the same as the node degree increase of B_3, its seed, compared to Q_3.

Designs of this form can be easily reconfigured when faults occur. Consider the case of the $Q^+(4, 2, 1)$ graph R_4 which appears in Fig. 15. Suppose that two nodes 0011 and 1100 are faulty. The subcube 0*** is a $Q^+(3, 2, 1)$ seed. We can project the faults into the seed to obtain a pseudo-fault set $F' = \{0011, 0100\}$. Since the seed is relatively small, it is simple to recover a fault-free Q_2. In this specific example, it is easy to find the four pseudo-fault-free nodes 0110, 0101, 0001, and 0010 which form a pseudo-fault-free Q_2. The neighbors of these four nodes in the other seed 1*** are also pseudo-fault-free. Together they form a fault-free Q_3, which is shown in Fig. 15 by heavy lines.

3.4. Topological Properties

The spare links added to Q_n to construct $Q^+(n, k, 1)$ not only increase fault tolerance with respect to Q_{n-1}, but also improve some useful topological properties of Q_n. These links reduce the diameter and shorten the average internode distance of the graph. They also allow some useful graphs to be embedded compactly in the proposed structures.

The topological characteristics of the proposed designs are summarized and compared with those of Q_n in Table III. It is easily seen that if the diameter of G is d, then the diameter of $G' = K_2 \times G$ is $d + 1$. Using this property together with the observation that the diameter of the graphs R, U, B, and D is two, the diameters of R_n, U_n, B_n, and D_n can be shown to be $n - 1$. The number of links in each graph can be determined from the following result [7]. If $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, then the number of links in graph $G_1 \times G_2$ is $|V_1|\,|E_2| + |V_2|\,|E_1|$.

The *link overhead* of a fault-tolerant design Q_n^+ is defined to be the ratio of the number of redundant links in Q_n^+ to the number of links in the underlying Q_n. Table III shows that the link overhead in each of our designs is small. For instance, the link overhead is $1/n$ in R_n and U_n; it is $5/2n$ in B_n and $3/n$ in D_n.

Let $d(v_1, v_2)$ be the distance between nodes v_1 and v_2. The average distance of $G = (V, E)$ is defined as

$$\frac{\sum_{1\le i\le j\le |V|} d(v_i, v_j)}{\binom{|V|}{2}}.$$

The distance parameters of the proposed $Q^+(n, k, 1)$ designs are also listed in Table III for comparison with Q_n. Note that with a small increase in the number of links and the node degrees, the fault-tolerance capabilities of the proposed structures are increased, while the average internode distance is decreased.

The proposed structures embed more graphs than Q_n. Any cubical graph can also be embedded in the four $Q^+(n, k, 1)$ graphs, since Q_n is a subgraph of R_n, U_n, B_n, and D_n. Furthermore, R_n can embed a cycle C_i for any i, while Q_n can only embed C_i for even i's. B_n and D_n have this property as well, since each contains R_n as a subgraph. In addition, T_n, the full binary tree of height n, can be embedded in R_n, B_n, and D_n more compactly than in hypercubes. It is known that T_n can be embedded in Q_{n+2} but not in Q_{n+1}. Nebesky [10] defines Q_n^{∇} to be the graph $Q_n + rt - s$, where r, s, and t are nodes of Q_n such that rs and st are distinct edges of Q_n. He then shows that T_n is a spanning subgraph of Q_{n+1}^{∇}, for $n \ge 1$. Since Q_{n+1}^{∇} is a subgraph of R_{n+1}, T_n can be embedded in R_{n+1}, and therefore also in B_{n+1} and D_{n+1}.

TABLE III
Topological Properties of Q_n, R_n, B_n, and D_n

Graphs	Fault tolerance	Number of nodes	Number of links	Maximum node degree	Diameter	Average distance
Q_n	$Q^+(n, 1, 1)$	2^n	$n2^{n-1}$	n	n	$\approx n/2$
R_n	$Q^+(n, 2, 1)$	2^n	$(n + 1)2^{n-1}$	$n + 1$	$n - 1$	$\approx n/2 - 1/4$
B_n	$Q^+(n, 3, 1)$	2^n	$(n + 2)2^{n-1}$	$n + 2$	$n - 1$	$\approx n/2 - 1/2$
D_n	$Q^+(n, 4, 1)$	2^n	$(n + 3)2^{n-1}$	$n + 3$	$n - 1$	$\approx n/2 - 5/8$

4. TWO-STEP-DEGRADABLE DESIGNS

We show in this section that the proposed $Q^+(n, k, 1)$'s can serve as 2-SD hypercubes $Q^+(n, k, 2)$ with high levels of fault tolerance. To facilitate the discussion, we use R_3, W_3, and D_3 to denote the $Q(3, 2, 1)$, $Q(3, 3, 1)$, and $Q(3, 4, 1)$ graphs presented previously; see Figs. 9 and 11. The $Q(n, k, 1)$ graphs derived from these graphs using the product method are termed R_n, W_n, and D_n.

To begin, we demonstrate the 2-SD fault-tolerance property of R_3, B_3, and D_3. It is not hard to see that R_3, B_3, and D_3 can tolerate up to five faults, while a fault-free subgraph Q_2 exists. The reason is that each of them contains two copies of K_4. It takes three faults to destroy a fault-free Q_1 in K_4. Therefore it takes at least six faults to destroy every fault-free Q_1. Fault sets that destroy every fault-free Q_1 in R_3, B_3, and D_3 are shown in Fig. 16. Therefore R_3, B_3, and D_3 are $Q(3, 5, 2)$ graphs.

In general, it is not easy to determine the number of faults that a $Q(n, k, 1)$ graph can tolerate while allowing a Q_{n-2} to be recovered. However, if $Q(n, k, 1)$ is constructed by the graph-product method, a lower bound on this number can be obtained, which is characterized in the next theorem.

THEOREM 8. *If G_n is a $Q(n, k, 1)$ graph, then $K_2 \times G_n$ is a $Q(n + 1, 2k + 1, 2)$ graph G_{n+1}.*

The theorem can be explained as follows. G_{n+1} consists of two disjoint copies of G_n. Since G_n is a $Q(n, k, 1)$ graph, it takes at least $k + 1$ faults to destroy every fault-free Q_{n-1} in each copy of G_n. It takes at least $2(k + 1)$ faults to destroy every fault-free Q_{n-1} in G_{n+1}. Thus G_{n+1} can tolerate at least $2k + 1$ faults while a fault-free subgraph G_{n-1} exists. Therefore G_{n+1} is a $Q(n + 1, 2k + 1, 2)$ graph.

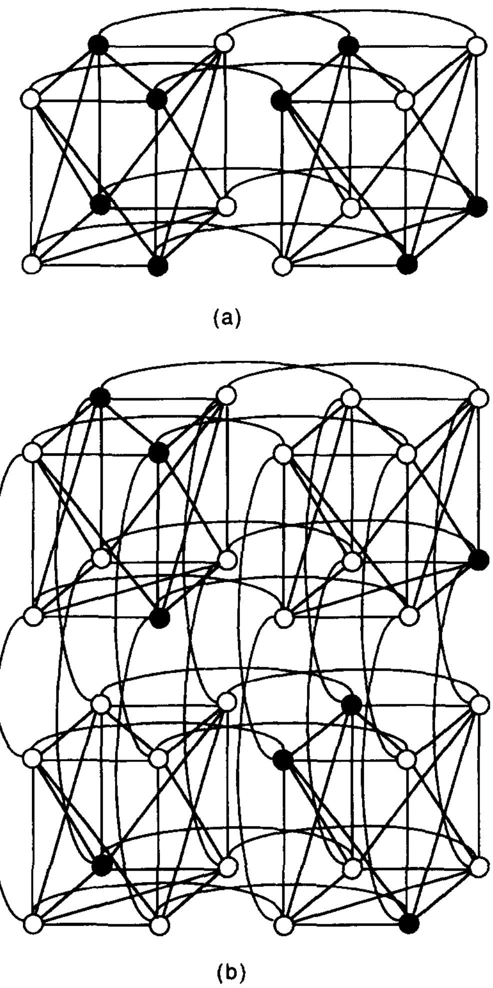

FIG. 17. Fault sets that destroy every fault-free Q_2 and Q_3 in (a) W_4 and (b) W_5, respectively.

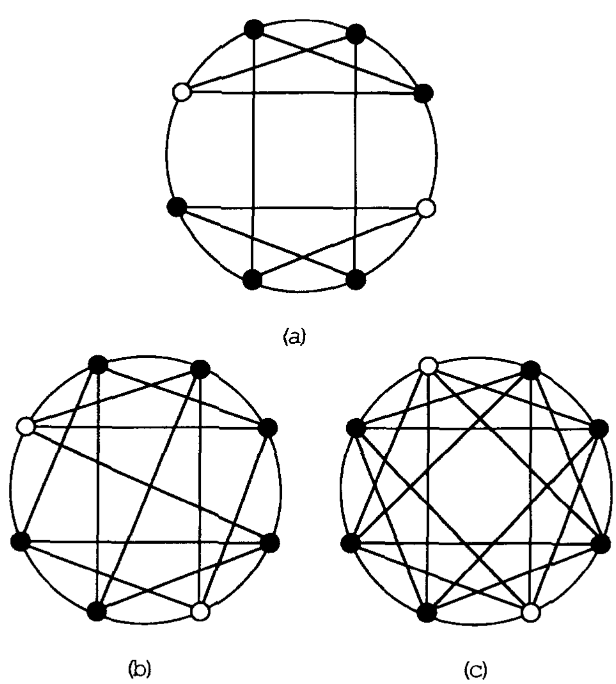

FIG. 16. Faults sets that destroy every fault-free Q_1 in (a) R_3, (b) B_3, and (c) D_3.

It is not clear, however, whether $2k + 1$ faults is the maximum number that G_{n+1} can tolerate while a fault-free Q_{n-1} is present. Nevertheless, in some special cases, $2k + 1$ can be proved to be the maximum by explicitly constructing the fault set. For instance, Fig. 17 demonstrates how eight faults can destroy every fault-free Q_2 and Q_3 in W_4 and W_5, respectively.

Theorem 8 implies that the $Q^+(n, k, 1)$ designs can be taken as two-step-degradable structures with a high level of fault tolerance. For example, it takes at least 8, 9, and 10 faults to destroy every fault-free Q_{n-2} in R_n, W_n, and D_n. Many of the advantages of our previous designs, such as low increase in node degree and ease of reconfiguration, also apply to such designs. We list in Table IV the

TABLE IV
The Fault Tolerance of the $Q(n, k, 1)$ Designs with Respect to Q_{n-2}

Graph	Fault tolerance with respect to Q_{n-2}
$Q(n, 2, 1)$	≥5
$Q(n, 3, 1)$	≥7
$Q(n, 4, 1)$	≥9
$Q(n, 5, 1)$	≥11
$Q(n, 6, 1)$	≥13
$Q(n, 7, 1)$	≥15
$Q(n, 8, 1)$	≥17
$Q(n, 9, 1)$	≥19
$Q(n, 10, 1)$	≥21

corresponding fault tolerance of the $Q(n, k, 1)$ designs used as two-step-degradable structures.

5. CONCLUSIONS

We have presented a novel characterization of fault-tolerant hypercube structures that allows the performance degradation due to faults to be quantified. We developed specific fault-tolerant designs for zero-, one-, and two-step degradation. We showed that by using a graph-product construction, a large fault-tolerant design can be developed using a small design as a seed. Most of our designs are regular and node-symmetric, and are quite efficient in terms of the overhead associated with the spare links and the maximum node degree.

The graph-product construction technique can be extended to other systems defined by product graphs. Suppose that $G = G_1 \times G_2$, and G_1' is a k-FT realization of G_1. Then $G' = G_1' \times G_2$ is a k-FT realization of G. We can illustrate this for an $m \times n$ mesh $M = P_m \times P_n$. C_{m+1}, a loop with $m + 1$ nodes, is a 1-FT realization of P_m. $M' = C_{m+1} \times P_n$ is then a 1-FT realization of M. This method provides a way to construct general fault-tolerant designs efficiently and systematically.

REFERENCES

1. Banerjee, P., Rahmeh, J., Stunkel, C., Nair, S., Roy, K., and Abraham, J. An evaluation of system-level fault tolerance on the Intel hypercube multiprocessor. *Proc. 18th Symposium on Fault-Tolerant Computing*, 1988, pp. 362–367.
2. Becker, B., and Simon, H. U. How robust is the *n*-cube? *Proc. 27th Annual Symposium on Foundations of Computer Science*, 1986, pp. 283–291.
3. Bhuyan, L. N., and Agrawal, D. P. Generalized hypercube and hypercube structures for a computer network. *IEEE Trans. Comput.* **33** (Apr. 1984), 323–333.
4. Brualdi, R. A. *Introductory Combinatorics*. North-Holland, New York, 1983.
5. Dutt, S., and Hayes, J. P. An automorphic approach to the design of fault-tolerant multiprocessors. *Proc. 19th Symposium on Fault-Tolerant Computing*, 1989, pp. 496–503.
6. Graham, N., Harary, F., Livingston, M., and Stout, Q. F. Subcube fault-tolerance in hypercubes. *Inform. and Comput.*, to appear.
7. Harary, F. *Graph Theory*. Addison–Wesley, Reading, MA, 1969.
8. Krishnamoorthy, M. S., and Krishnamurthy, B. Fault diameter of interconnection network, *Comput. Math. Appl.* **13,** 5/6 (1987), 577–582.
9. Lee, T. C. Design of fault-tolerant hypercube computers. Ph.D. thesis, University of Michigan, Ann Arbor, MI, 1990.
10. Nebesky, L. On cubes and dichotomic trees. *Časopis Pěst. Mat.* **99** (1974), 164–167.
11. Rennels, D. A. On implementing fault-tolerance in binary hypercubes. *Proc. 16th Symposium on Fault-Tolerant Computing*, 1986, pp. 344–349.
12. Yanney, R. M. Fault recovery in multiprocessor networks. Ph.D. thesis, University of Southern California, Los Angeles, CA, 1982.

Received December 7, 1990; accepted June 14, 1991

TZE CHIANG LEE received the B.S. degree in electrical engineering from the National Taiwan University in 1982. He received the M.S. degree in computer information and control engineering and the Ph.D. degree in computer science and engineering from the University of Michigan in 1988 and 1990, respectively. He is currently a research staff member at IBM Thomas J. Watson Research Center, Yorktown Heights, New York. His research interests include performance evaluation, computer architecture, parallel and distributed systems, and fault-tolerant computing. Dr. Lee is a member of the IEEE Computer Society and the ACM.

JOHN P. HAYES received the B.E. degree from the National University of Ireland in 1965, and the M.S. and Ph.D. degrees from the University of Illinois in 1967 and 1970, respectively, all in electrical engineering. He has been a professor in the EECS Department at the University of Michigan since 1982. He teaches and conducts research in the areas of computer-aided design and testing, computer architecture, and fault-tolerant computing. Hayes has written several books including *Computer Architecture and Organization* (2nd ed., McGraw-Hill, 1988) and *Hierarchical Modeling for VLSI Circuit Testing* (Kluwer, 1990, coauthored with D. Bhattacharya), as well as over a hundred technical papers. He is a fellow of IEEE and a member of ACM and Sigma Xi.